

FACULDADE DE ENGENHARIA DA UNIVERSIDADE DO PORTO



FEUP

Desenvolvimento de um Chuto Omnidirecional para um Robô Humanoide

Rui Pedro Gomes Ferreira

Mestrado Integrado em Engenharia Electrotécnica e de Computadores

Orientador: Luís Paulo Reis (Ph.D)

Co-orientador: António Paulo Gomes Mendes Moreira (Ph. D)

Fevereiro 2012

A Dissertação intitulada

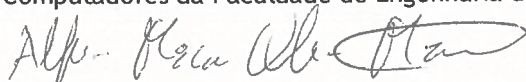
“Desenvolvimento de um Chuto Omnidirecional para um Robô Humanoide”

foi aprovada em provas realizadas em 20 de Fevereiro de 2012

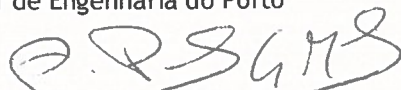
o júri



Presidente Professor Doutor Luís António Pereira de Meneses Corte-Real
Professor Associado do Departamento de Engenharia Electrotécnica e de
Computadores da Faculdade de Engenharia da Universidade do Porto



Professor Doutor Alfredo Oliveira Martins
Professor Adjunto do Departamento de Engenharia Electrotécnica da Instituto
Superior de Engenharia do Porto



Professor Doutor António Paulo Gomes Mendes Moreira
Professor Auxiliar do Departamento de Engenharia Electrotécnica e de
Computadores da Faculdade de Engenharia da Universidade do Porto



Professor Doutor Luís Paulo Gonçalves dos Reis
Colaborador Externo do Departamento de Engenharia Informática

O autor declara que a presente dissertação (ou relatório de projeto) é da sua exclusiva autoria e foi escrita sem qualquer apoio externo não explicitamente autorizado. Os resultados, ideias, parágrafos, ou outros extratos tomados de ou inspirados em trabalhos de outros autores, e demais referências bibliográficas usadas, são corretamente citados.

Autor - Rui Pedro Gomes Ferreira



Faculdade de Engenharia da Universidade do Porto

Resumo

Um dos grandes desafios dos últimos anos é o controlo de um robô bípede, com vários graus de liberdade, com a finalidade de simular comportamentos e características humanas. Este desafio levou vários investigadores nas áreas da biologia, física, eletrotécnica, ciências de computadores e mecânica a debruçarem-se sobre assunto e desenvolver métodos e técnicas capazes de se aproximarem aos comportamentos humanos.

Esta dissertação tem como base o desenvolvimento de um chute omnidirecional para um robô humanoide para a equipa FC Portugal 3D.

Vários métodos foram desenvolvidos para a criação de comportamentos e cada um desses métodos define uma trajetória para cada junta. Um dos métodos de estudo no passado consistia numa sequência de funções degrau que definem o ângulo para cada junta durante o movimento, este método já não é usado pela equipa FC Portugal 3D.

Outros dois métodos são estudados e desenvolvidos. Um baseado na interpolação sinusoidal que consiste em gerar uma trajetória sinusoidal, com transições suaves entre o ângulo inicial e o ângulo desejado, durante um determinado período de tempo. Outro baseado em comportamentos naturais que definem um padrão rítmico para o movimento.

A criação de um comportamento para executar um chute omnidirecional faria com que o robô conseguisse adaptar o movimento às condições dinâmicas do jogo e definisse o valor para as diversas juntas em tempo real ao contrário da execução de movimentos pré-programados.

Existem já vários comportamentos que permitem que uma versão simulada do humanoide NAO caminhe em várias direções, chute a bola, rode, se levante e apanhe a bola. Novos comportamentos e técnicas poderão ser criados usando os métodos mencionados.

Abstract

A major challenge in recent years is the control of a biped robot, with multiple degrees of freedom, in order to simulate human behaviors and characteristics. This challenge has led many researchers in biology, physics, electrical engineering, computer science and mechanical engineering to focus on the subject and develop methods and techniques to get closer to human behavior.

This thesis is based on the development of a omnidirectional kick for a humanoid robot for the team FC Portugal 3D.

Several methods have been developed to create behaviors and each of these methods defines a trajectory for each joint. One of the methods of study in the past was based on series of step functions that define the angle for each joint, during motion. This method is no longer in use by FC Portugal team.

Two other methods are studied and developed. A sinusoidal interpolation based consisting in generating a sinusoidal path, with smooth transitions between the initial angle and angle desired, for a certain period of time. Another based on natural behaviors that define a rhythmic pattern to the movement.

The creation of a behavior to perform a omnidirectional kick would make the robot to adapt the movement to the dynamic conditions of the game and set the value for the various joints in real time instead of executing pre-programmed movements.

There are already several behaviors that allow a simulated version of the NAO humanoid to walk in various directions, to kick the ball, to run, to stand up and to catch the ball. New behaviors and techniques can be created using the methods mentioned.

Agradecimentos

Em primeiro lugar gostaria de agradecer aos meus pais por sempre me motivarem no percurso académico e por me apoiarem incondicionalmente mesmo nos momentos mais difíceis. Também gostaria de agradecer ao resto da família, principalmente BMMRV, por todo o apoio e motivação quando deles precisei.

Em segundo gostaria de agradecer ao meu orientador Professor Luís Paulo Reis e ao meu co-orientador Professor António Paulo Gomes Mendes Moreira por todo o apoio e disponibilidade, sempre com sugestões e correções que me permitiram progredir e melhorar o trabalho.

Agradeço também aos meus colegas e amigos Júlio Ferreira e Seraphin Miranda pelas valiosas contribuições dadas a este projeto e a todos os outros que de alguma forma me ajudaram a completar o trabalho.

Por fim gostaria de agradecer a uma pessoa muito especial na minha vida. Foste um grande pilar, foste mais que um pai, mais que um irmão, mais que um amigo. Todos os caminhos que eu optei por seguir foram influenciados pelo que me ensinas-te e todas as escolhas que escolhi tiveram sempre na base algo fornecido por ti (material ou imaterial). Joaquim, a ti devo tudo.

Rui Pedro Gomes Ferreira

“All things are difficult before they are easy.”

Thomas Fuller

Conteúdo

1	Introdução	1
1.1	Enquadramento	1
1.2	Motivação	1
1.3	Objetivos	2
1.4	Estrutura da Dissertação	3
2	Domínio da Aplicação	5
2.1	RoboCup	5
2.1.1	RoboCup Soccer	6
2.1.2	RoboCup Rescue	9
2.1.3	RoboCup Junior	10
2.1.4	RoboCup@Home	10
2.2	FC Portugal	10
2.2.1	FC Portugal	10
2.2.2	CAMBADA	12
2.2.3	5DPO	12
2.2.4	Portuguese Team	13
3	Simulação	15
3.1	Vantagens da Simulação	15
3.2	Simulador SimSpark	16
3.3	Servidor	16
3.4	Monitor	17
3.5	Modelo dos Agentes	18
3.5.1	HOPE-1	18
3.5.2	NAO	20
3.6	Ambiente e Objetos do Campo	22
3.6.1	Dimensões	22
3.6.2	Regras	22
3.7	Agente FC Portugal 3D	23
3.7.1	Arquitetura	23
3.7.2	Modelo	24
4	Comportamentos	27
4.1	Introdução	27
4.2	Modelos de Comportamentos	28
4.2.1	Step-Based	29
4.2.2	Interpolação Sinusoidal	29

4.2.3	Central Pattern Generator	30
4.3	Conclusões	31
5	Chuto Omnidirecional	33
5.1	Movimentos Rígidos e Transformações Homogéneas	34
5.1.1	Representação de Pontos e Vetores	35
5.1.2	Representação de Rotações	36
5.1.3	Movimentação de Corpos Rígidos	39
5.1.4	Transformações Homogéneas	40
5.2	Cinemática Inversa	41
5.3	Planeamento de Trajetória	45
5.3.1	Introdução	45
5.3.2	Propriedades	47
5.3.3	Construção	47
5.4	Estabilidade	47
5.5	Algoritmo de Execução do Chuto Omnidirecional	48
6	Experiências e Resultados	51
6.1	Cinemática Inversa	52
6.1.1	Experiência da Cinemática Inversa - Ponto alcançável 1	52
6.1.2	Experiência da Cinemática Inversa - Ponto alcançável 2	54
6.1.3	Experiência da Cinemática Inversa - Ponto fora do volume de trabalho	55
6.1.4	Experiência da Cinemática Inversa - Ponto fora dos limites das articulações	56
6.1.5	Análise dos Resultados	57
6.2	Planeamento de Trajetória	58
6.2.1	Experiência Planeamento de Trajetória - Movimento linear	58
6.2.2	Experiência Planeamento de Trajetória - Movimento circular para cima	59
6.2.3	Experiência Planeamento de Trajetória - Movimento circular para o lado	60
6.2.4	Análise dos Resultados	62
6.3	Chuto Omnidirecional	62
6.3.1	Experiência Chuto Omnidirecional - bola em frente aos pés	65
6.3.2	Experiência Chuto Omnidirecional - bola em frente ao pé direito	66
6.3.3	Experiência Chuto Omnidirecional - bola em frente e ao lado do pé direito	67
6.3.4	Análise dos Resultados	68
7	Conclusões e Trabalho Futuro	71
7.1	Conclusões	71
7.2	Trabalho Futuro	72
	Referências	73

Lista de Figuras

1.1	Motivação.	2
2.1	RoboCup - Liga de robôs de tamanho médio [8].	7
2.2	RoboCup - Liga de robôs de tamanho pequeno [10].	7
2.3	RoboCup - Liga de robôs com plataforma normalizada [12].	8
2.4	RoboCup - Competição simulação.	8
2.5	RoboCup - Competição de humanoides [17].	9
2.6	RoboCup - Competição de Salvamento.	10
2.7	RoboCup Júnior [24].	11
2.8	RoboCup@Home [28].	11
3.1	Comunicação entre os agentes, o servidor e o monitor.	16
3.2	Diagrama de comunicação entre o servidor e o agente [42].	17
3.3	SimSpark Monitor em execução [48].	18
3.4	Robô HOPE-1 [42].	19
3.5	Robô NAO [42].	21
3.6	Dimensões do campo simulado [42].	22
3.7	Arquitetura do agente da equipa FC Portugal 3D.	24
5.1	Arquitetura do comportamento a desenvolver.	34
5.2	Dois referenciais de coordenadas, um ponto p e dois vetores v1 e v2 [51].	35
5.3	Orientação de um referencial em relação a outro (2D) [51].	37
5.4	Rotação de um ponto em torno do eixo Z [51].	38
5.5	Orientação de um referencial em relação a outro (3D) [51].	38
5.6	Movimentação de um corpo rígido [52].	39
5.7	Configuração dos referenciais usados na cinemática inversa. Vermelho=eixo-x, verde=eixo-y e azul=eixo-z [53].	42
5.8	Triângulo constituído pelo joelho e pelos membros da perna. Adaptado de [53]	43
5.9	Curvas de Bézier de 1º, 2º, 3º e 4º grau [54].	46
5.10	Definição de letras usando curvas de Bézier [54].	46
5.11	Duas curvas cúbicas de Bézier com diferentes pontos de controlo.	47
5.12	Diagrama do módulo de estabilidade.	48
5.13	Parâmetros necessários para a execução do chute omnidirecional [62].	49
5.14	Algoritmo de execução do movimento.	49
6.1	Algoritmo de execução do programa de testes.	52
6.2	Sequência de pontos percorridos pelo pé direito até atingir a posição final.	53
6.3	Posição do robô simulado com o pé direito na posição final.	53
6.4	Sequência de pontos percorridos pelo pé direito até atingir a posição final.	54

6.5	Posição do robô simulado com o pé direito na posição final.	55
6.6	Sequência de pontos percorridos pelo pé direito até atingir a posição final.	56
6.7	Posição do robô simulado com o pé direito na posição final.	56
6.8	Sequência de pontos percorridos pelo pé direito até atingir a posição final.	57
6.9	Posição do robô simulado com o pé direito na posição final.	57
6.10	Sequência de pontos percorridos pelo pé direito durante a execução do movimento linear.	59
6.11	Movimento linear - Sequência de fotogramas.	59
6.12	Sequência de pontos percorridos pelo pé direito durante a execução do movimento circular no plano XZ.	60
6.13	Movimento circular no plano XZ - Sequência de fotogramas.	60
6.14	Sequência de pontos percorridos pelo pé direito durante a execução do movimento circular no plano XY.	61
6.15	Movimento circular no plano XY - Sequência de fotogramas.	61
6.16	Posições da bola, em relação ao robô, usadas nos testes.	62
6.17	Parte dos parâmetros necessários na execução do movimento.	63
6.18	Exemplos de chutos em várias direções na simulação.	64
6.19	Resultados da experiência com a bola situada à frente dos dois pés.	65
6.20	Dispersão dos pontos finais da bola da experiência com a bola situada à frente dos dois pés.	66
6.21	Resultados da experiência com a bola situada à frente do pé direito.	67
6.22	Dispersão da posição final da bola da experiência com a bola situada à frente do pé direito.	68
6.23	Resultados da experiência com a bola situada à frente e ao lado do pé direito.	69
6.24	Dispersão da posição final da bola da experiência com a bola situada à frente e ao lado do pé direito.	70

Lista de Tabelas

3.1	Configuração das juntas do modelo HOPE-1. Para efeitos de simplicidade apenas foram incluídas as juntas relativas a um braço e a uma perna. Os prefixos 'a' e 'l' no nome significam arm e leg (braço e perna) respetivamente	19
3.2	Configuração das juntas do modelo NAO. Para efeitos de simplicidade apenas foram incluídas as juntas relativas a um braço e a uma perna. Os prefixos 'h', 'a' e 'l' no nome significam head, arm e leg (cabeça, braço e perna) respetivamente	20
5.1	Valores da articulações calculados através da cinemática inversa.	45
6.1	Dados da experiência.	53
6.2	Dados da experiência.	54
6.3	Dados da experiência.	55
6.4	Dados da experiência.	57
6.5	Movimento linear - Dados da experiência.	58
6.6	Movimento circular no plano XZ - Dados da experiência.	60
6.7	Movimento circular no plano XY - Dados da experiência.	61
6.8	Chuto omnidirecional - Parâmetros da experiência com a bola à frente dos dois pés.	65
6.9	Chuto omnidirecional - Parâmetros da experiência com a bola à frente do pé direito.	67
6.10	Chuto omnidirecional - Parâmetros da experiência com a bola à frente e ao lado do pé direito.	68

Abreviaturas e Símbolos

API	Application Programming Interface
DOF	Degrees Of Freedom
FIFA	Fédération Internationale de Football Association
LISP	Locator/Identifier Separation Protocol
ODE	Open Dynamics Engine
UDP	User Datagram Protocol
XML	Extensible Markup Language
UA	Universidade de Aveiro
UP	Universidade do Porto
LIACC	Laboratório de Inteligência Artificial e de Ciência de Computadores
IEETA	Instituto de Engenharia Eletrónica e Telemática de Aveiro
FEUP	Faculdade de Engenharia da Universidade do Porto
INESC-P	Instituto de Engenharia de Sistemas e Computadores - Porto

Capítulo 1

Introdução

1.1 Enquadramento

O trabalho descrito nesta dissertação está relacionado com o estudo e desenvolvimento de um comportamento, para a equipa FC Portugal, de forma a dotar um robô com a capacidade de chutar uma bola em várias direções relativas à sua orientação. O principal objetivo é combinar este comportamento (chuto omnidirecional) a outros já desenvolvidos (correr, rodar, chuto frontal, chuto lateral, etc.) e criar uma equipa de robôs, reais ou virtuais, que seja capaz de jogar um desafio de futebol, utilizando regras semelhantes ao futebol jogado entre humanos.

O trabalho a realizar incidirá em 2 tarefas, são elas:

- Estudo de técnicas e movimentos usados por jogadores de futebol reais.
- Desenvolvimento de um comportamento para um chuto omnidirecional flexível.

1.2 Motivação

A motivação principal para a realização deste trabalho resulta do desafio de criar software que simule comportamentos ou habilidades, existentes no ser humano, e dotar robôs com essas habilidades para conseguirem jogar um jogo de futebol em equipa. Esta equipa de robôs poderá ser capaz de, no futuro, defrontar uma equipa de jogadores reais, e para o efeito é necessário que seja capaz de cooperar, organizar-se e executar estratégias e táticas de futebol real, ou seja, que tenha um complexo grau de inteligência artificial.

O xadrez foi durante muito tempo o domínio principal na investigação de metodologias de Inteligência Artificial. Isso mudou com a vitória da máquina (Deep Blue) sobre, na época, o campeão do mundo (Gary Kasparov), o que resultou na necessidade de levar o estudo da Inteligência Artificial para novos domínios. É neste contexto que surge o futebol robótico.

O futebol robótico é um problema que levanta desafios mais estimulantes e que cobrem uma ampla área de investigação, donde se destaca investigação na área da robótica, física, biologia, eletrotecnia, ciências de computadores e mecânica.

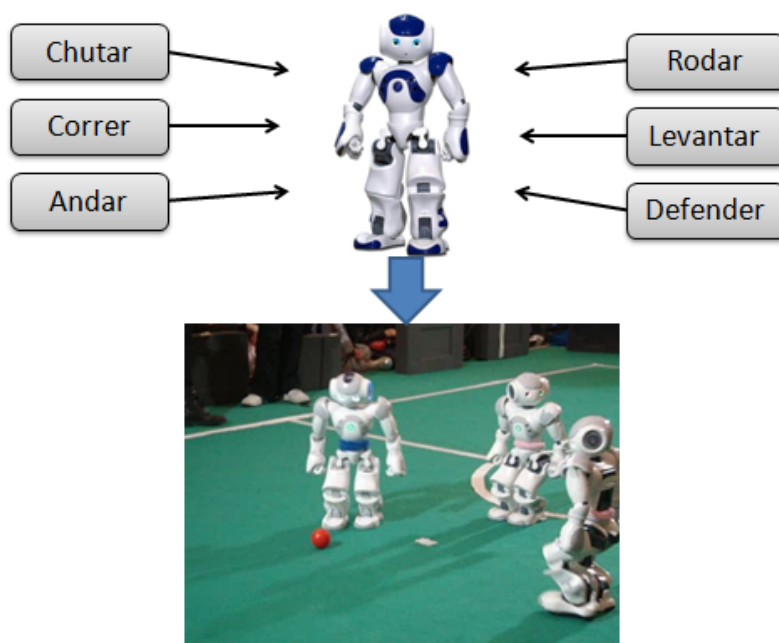


Figura 1.1: Motivação.

O trabalho a ser desenvolvido nesta dissertação poderá ser expandido para a resolução de problemas da vida real, problemas que coloquem a vida humana em jogo. O modo de cálculo, em tempo real, do valor de cada junta de forma a colocar uma parte do corpo (por exemplo: o pé) numa dada posição poderá ser adaptado para um desses problemas. Alguns exemplos onde poderão ser implementados são: salvamento e resgate em catástrofes e na exploração terrestre ou espacial.

1.3 Objetivos

O objetivo de estudo desta dissertação é de desenvolver um comportamento para executar um chute omnidirecional para um robô humanoide. Este comportamento tem como funcionalidade atualizar as várias juntas de movimento do robô para que o pé execute uma trajetória que dote a bola com uma deslocação na direção pretendida.

Este comportamento será utilizado na equipa de simulação da FC Portugal de forma a dotar o robô de uma maior rapidez na execução dos chutes, ou seja, é eliminada a fase de preparação (fase onde os robôs se alinhavam com a bola e a baliza) usada nos diferentes chutes desenvolvidos (laterais e frontais).

Os objetivos específicos são:

- Criação de um módulo de cinemática inversa para movimentação das pernas.
- Criação de um módulo de planeamento de trajetória de forma a dotar o pé com o movimento pretendido.

- Criação de um módulo de estabilidade para equilibrar o robô durante a execução do movimento.
- Combinar os módulos e desenvolver um comportamento capaz de efetuar um chute omnidirecional.

1.4 Estrutura da Dissertação

Capítulo 1, Introdução, breve introdução ao trabalho a desenvolver nesta dissertação, os objetivos, enquadramento e motivação do trabalho.

Capítulo 2, Domínio da Aplicação, descrição da RoboCup, e suas ligas, e da equipa portuguesa de futebol robótico FC Portugal.

Capítulo 3, Simulação, descrição do simulador (SimSpark) usado na liga de simulação 3D da RoboCup, do robô NAO e da arquitetura do agente usado pela equipa FC Portugal 3D.

Capítulo 4, Comportamentos, descrição dos modelos matemáticos usados nos diferentes comportamentos desenvolvidos para o robô.

Capítulo 5, Chuto Omnidirecional, Desenvolvimento do chute omnidirecional e seus módulos constituintes (módulo cinemática inversa, módulo planeamento de trajetória e módulo de estabilidade).

Capítulo 6, Experiências e Resultados, fase de aquisição de resultados através de experiências e análise dos mesmos.

Capítulo 7, Conclusões e Trabalho Futuro, breve resumo da dissertação, conclusões finais e referências a possíveis melhorias e trabalho futuro.

Capítulo 2

Domínio da Aplicação

2.1 RoboCup

O futebol robótico, RoboCup [1], é uma organização internacional de investigação que visa a promoção da investigação em Inteligência Artificial e Robótica Inteligente [2]. Surge na forma de competição que decorre anualmente e que fornece problemas e desafios onde várias tecnologias e metodologias se podem combinar para obter os melhores resultados. A primeira edição decorreu em 1997 em Nagoya no Japão e tem vindo a decorrer anualmente. A próxima competição irá realizar-se em Junho de 2012 no México [3].

Foi proposto o jogo de futebol, como plataforma para a aplicação de pesquisas realizadas nas áreas da inteligência artificial e robótica. Algumas das áreas a desenvolver na inteligência artificial e robótica são: colaboração multi-agente, tomada de decisão em tempo real, modelos cognitivos e planeamento e estratégias [4].

O principal objetivo deste projeto é que, em meados do século XXI, uma equipa de jogadores de futebol humanoides, completamente autónoma, ganhe, jogando de acordo com as regras impostas pela FIFA¹, à equipa de futebol campeã mundial da época [5].

“By 2050, a team of fully autonomous humanoid robot soccer players shall win a soccer game, complying with official FIFA rules, against the winner of the most recent World Cup of human soccer.”

No futebol robótico o mundo é dinâmico e em tempo real, as leituras dos sensores não são simbólicas e o controlo é distribuído. Estes parâmetros fazem com que os desafios impostos pelo futebol robótico sejam bem mais complexos que os desafios impostos pela maioria dos problemas da Inteligência Artificial, tal como o xadrez.

Apesar do futebol ser um bom desafio, a iniciativa RobCup conta também com outros domínios de aplicação. Conta com uma liga de salvamento, uma liga para jovens investigadores e alunos e uma liga para ações do quotidiano. Estas ligas bem como o futebol robótico serão explicadas mais à frente.

¹Fédération Internationale de Football Association.

Cada competição é constituída por duas partes: as competições e o simpósio. As competições dividem-se em três grupos, sendo cada grupo constituído por várias ligas. No simpósio, que decorre após a competição, são apresentados e discutidos diversos trabalhos científicos da área.

No final, as várias equipas podem comparar os resultados obtidos com resultados obtidos por outras equipas e, assim, classificar o trabalho de investigação desenvolvido.

Actualmente, a RoboCup conta com várias ligas [1]:

- RoboCup Soccer
 - Middle Size League
 - Small Size League
 - Standard Platform League
 - Simulation League
 - Humanoid League
- RoboCup Rescue
 - Robot League
 - Simulation League
- RoboCup Junior
 - Soccer Challenge
 - Dance Challenge
 - Rescue Challenge
- Robocup@home

2.1.1 RoboCup Soccer

O futebol foi a grande motivação para o RoboCup visto apresentar desafios científicos bastante importantes. É um jogo coletivo com problemas individuais (identificação, localização e movimentação), e ao mesmo tempo, problemas coletivos (estratégia e movimentos de equipas, por exemplo o passe) [6]. A intervenção humana durante o jogo só é possível para introdução e remoção de jogadores no campo e a colocação da bola em jogo.

- **Middle Size League** — Jogada por duas equipas de 6 robôs móveis autónomos com rodas. Toda a informação relativa ao ambiente é obtida pelos seus sensores e são capazes de comunicar entre eles via wireless [7]. Na figura 2.1 vemos um exemplo de jogo desta competição.



Figura 2.1: RoboCup - Liga de robôs de tamanho médio [8].

- **Small Size League** — Nesta liga os robôs são bem menores e o ambiente é muito mais controlado. A informação obtida por cada robô é fornecida, via wireless, por um computador central que, através de uma câmara de vídeo, faz a aquisição e processamento de toda a informação do campo (posição dos jogadores, da bola, etc.) [9]. Na figura 2.2 vemos um exemplo de jogo desta competição.



Figura 2.2: RoboCup - Liga de robôs de tamanho pequeno [10].

- **Standard Platform League** — Nesta liga, todas as equipas utilizam robôs idênticos fazendo com que cada equipa force as suas atenções apenas para o software. Não é possível visão omnidirecional o que faz com que seja processada informação visual para auto-localização e localização da bola, dos outros jogadores e das balizas [11]. Na figura 2.3 vemos um exemplo de jogo desta competição.

Esta liga veio substituir a muito conceituada four-legged league, jogada com robôs AIBO da Sony, e agora é jogada com humanoides NAO da Alderaban.

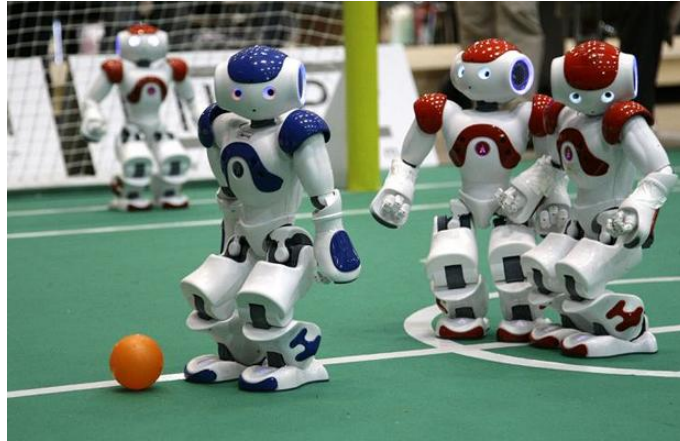
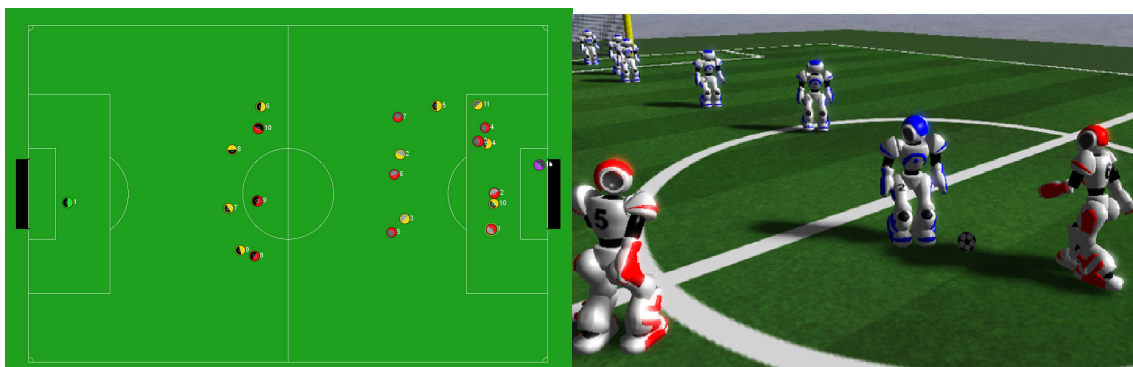


Figura 2.3: RoboCup - Liga de robôs com plataforma normalizada [12].

- **Simulation League** — Nesta liga não existem robôs/agentes físicos, todo o ambiente e agentes são simulados, todo o desenvolvimento é focado na inteligência artificial e estratégia de equipa. Um simulador oferece a todos os agentes os dados que seriam obtidos através de sensores e faz o processamento das ações de cada jogador [13].

Uma das vantagens da simulação é que é possível fazer diversas experiências com o robô sem correr o risco de o danificar o que levava a reparações muitas vezes dispendiosas financeiramente. Orçamentos com o hardware sempre foram uma limitação ao desenvolvimento da robótica. Na figura 2.4 um jogo das duas ligas desta competição.



(a) Simulação 2D [14].

(b) Simulação 3D [15].

Figura 2.4: RoboCup - Competição simulação.

- **Humanoid League** — Nesta liga, robôs autónomos e constituição parecida com a de um humano, jogam futebol. É a liga que apresenta os maiores desafios e de maior complexidade,

tais como, andar, correr e chutar a bola enquanto mantém o equilíbrio, processamento visual e jogar em equipa [16]. Na figura 2.5 vemos um exemplo de jogo desta competição.

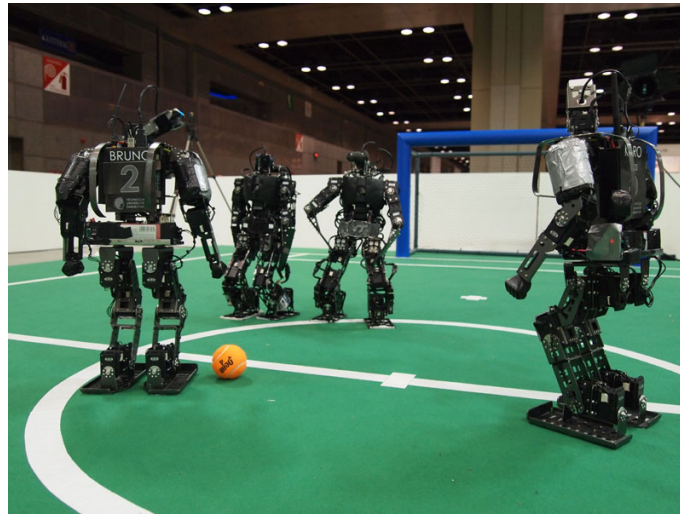


Figura 2.5: RoboCup - Competição de humanoides [17].

2.1.2 RoboCup Rescue

O resgate e salvamento em cenários de catástrofe é uma das áreas onde se prevê grande intervenção de sistemas artificiais no futuro. O objetivo principal deste projeto, como descrito em [18], é promover a pesquisa e desenvolvimento de soluções para esta área, o que leva a uma crescente melhoria de soluções de coordenação de equipas multi-agente. No futuro, muita da tecnologia desenvolvida irá certamente integrar as equipas reais de busca e salvamento [19].

- **Robot League** — Um ou vários robôs movem-se dentro de uma zona de testes que recria um edifício parcialmente destruído onde existem vítimas que esperam ajuda exterior, com zonas de diferentes níveis de dificuldade. O objetivo é procurar as vítimas e construir um mapa do ambiente e transmiti-lo ou trazê-lo até à equipa humana no exterior [20]. Na figura 2.6a vemos um exemplo de cenário desta liga.
- **Simulation League** — Um conjunto de simuladores simula as consequências de um grande desastre. Tenta-se recriar o melhor possível o ambiente de catástrofe de grandes dimensões [21].

Os agentes desenvolvidos pelas equipas irão atuar neste ambiente e irão fazer tudo para salvar o maior número possível de vítimas e minimizar os estragos na cidade. Este ambiente obriga as equipas a pesquisar e apresentar soluções em diversas áreas (planeamento multi-agente, coordenação de agentes heterogéneos, planeamento robusto, etc.). Na figura 2.6b vemos um exemplo de cenário desta liga.

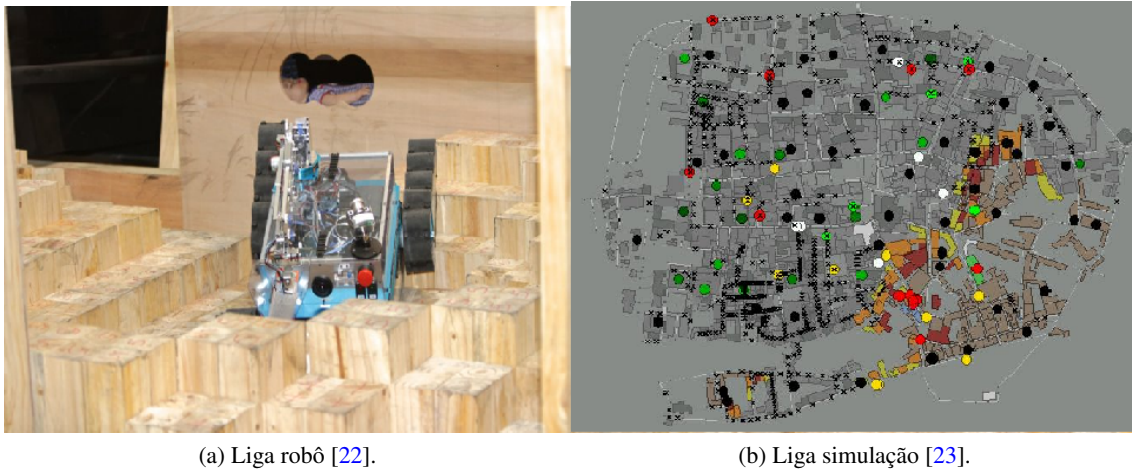


Figura 2.6: RoboCup - Competição de Salvamento.

2.1.3 RoboCup Junior

Liga orientada para jovens estudantes do 3º ciclo e do secundário onde possam participar e estimular o interesse nas áreas da robótica e da Inteligência Artificial. Estão presentes na RoboCup Junior as categorias do Soccer Challenge, Rescue Challenge e Dance Challenge [24]. Os eventos desta competição são de envergadura local e por vezes de envergadura internacional. Na figura 2.7 vemos exemplos competição.

2.1.4 RoboCup@Home

Esta liga tem como objetivo desenvolver robôs autónomos móveis para futuras aplicações na vida quotidiana. O centro das atenções é a interação homem-máquina [25] [26]. Esta liga foi criada (originalmente proposta em [27]) para responder à necessidade de uma cooperação entre robôs e humanos. Na figura 2.8 vemos um exemplo desta competição.

2.2 FC Portugal

2.2.1 FC Portugal

A equipa FC Portugal² [29] [30] foi criada em Fevereiro de 2000 e é um projeto de investigação 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). A equipa participa em diversas competições da RoboCup, incluindo simulação 2D, simulação 3D, competição de treinadores, simulação de salvamento e realidade mista. A equipa tem apostado no desenvolvimento de soluções [29] [30] para posicionamento estratégico dos jogadores e na troca de posições entre jogadores para as ligas de simulação.

²<http://www.ieeta.pt/robocup/>



(a) Competição de futebol.

(b) Competição de salvamento.



(c) Competição de dança.

Figura 2.7: RoboCup Júnior [24].



Figura 2.8: RoboCup@Home [28].

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. O desenvolvimento orientado à investigação fez com que FC Portugal se tornasse uma equipa muito mais competitiva e com obtenção de bons resultados ao longo dos anos (campeã mundial em 2000 e campeã europeia em 2000 e 2001 em simulação 2D, campeã em 2002 e 2º lugar em 2003 e 2004 na competição de treinadores, campeã europeia em 2006 na competição simulação de salvamento, campeã europeia em 2006 e 2007 na simulação 3D, 2º lugar na competição de realidade mista em 2007 e campeã mundial na RoboCup de 2006) [31].

Outras duas equipas, a atuar em competições diferentes, fazem parceria com a equipa FC Portugal: CMBADA (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.

2.2.2 CMBADA

A equipa CMBADA [32] foi criada em Outubro de 2003 e é a equipa de futebol robótico da IEETA/UA [33]. 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 nas áreas de coordenação, jogadas e posicionamento em campo [34] [35]. Na liga que participa já obteve várias vezes o terceiro lugar mas o ponto alto foi a conquista da RoboCup em 2008.

2.2.3 5DPO

A equipa 5DPO [36] foi criada em 1997 para competir na liga de robôs pequenos e começou nos robôs médios em 2000. É formada por estudantes, professores e investigadores da FEUP, Departamento de Engenharia Eletrotécnica e de Computadores, em parceria com o Instituto de Engenharia de Sistemas e Computadores do Porto (INESC-P) [36]. 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 [37].

As principais áreas de investigação são [36]

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

2.2.4 Portuguese Team

A equipa Portuguese Team [38] foi criada em 2011 e é composta por membros de três grandes equipas portuguesas (FC Portugal, CMBADA e 5DPO) mas que já tinham trabalho juntos na equipa FC Portugal, nas ligas de simulação. Foi criada com o intuito de participar na liga de robôs com plataforma normalizada (liga SPL) e de alargar os seus conhecimentos nas seguintes áreas [38]

- Sistemas multi-agente;
- Coordenação em sistemas multi-agente;
- Análise de jogos;
- Modelação de raciocínio estratégico;
- Processamento de imagens em tempo real;
- Desenvolvimento de ligações entre estratégias do futebol real e a sua aplicação no futebol robótico e simulado;

Capítulo 3

Simulação

Neste capítulo iremos detalhar a liga de simulação 3D da RoboCup. Começamos por discutir as vantagens do uso de simuladores e de seguida descrever o ambiente de simulação, incluindo os agentes simulados.

Um programa de computador que tenta simular alguma coisa da vida real é chamado de simulação por computador, ou modelo computacional. Uma simulação pode ser algo da natureza (catástrofes naturais, cosmos, cinemática objetos, etc) ou um objeto criado pelo humano (carros, aviões, computadores, etc.).

3.1 Vantagens da Simulação

Todo o hardware relacionado com robótica, desde manipuladores simples a humanoides realistas, é geralmente caro. O uso de simuladores para investigação, desenvolvimento e testes fornecem muitas vantagens em relação ao uso de robôs reais [39] [40] [41]. As principais vantagens são:

- Custos relativamente baixos em comparação com robôs reais;
- Desenvolvimento fácil de novos modelos de robôs;
- Facilidade de testar novos algoritmos;
- Fase de desenvolvimento e testes mais rápida;
- Possibilidade de adicionar, remover e testar componentes diferentes;
- Todos os testes podem ser realizados sem danificar o robô real;
- Os simuladores são muito realistas o que facilita a transição para o modelo real;
- Controlo sobre o tempo;
- Possibilidade de repetir os testes sem necessidade de reinicializar manualmente o robô a cada iteração.

A grande desvantagem do uso de simuladores é que o simulador nunca é exatamente igual ao sistema real, isto obriga a que seja necessário algumas alterações aquando da transição para o sistema real.

3.2 Simulador SimSpark

O simulador SimSpark¹ é uma plataforma de simulação genérica para sistemas multi-agente em ambientes tridimensionais desenvolvida recorrendo à framework zeitgeist. Pretende simular todo o tipo de situações, desde pequenas experiências de física para fins académicos até a um jogo de futebol para investigação científica. É atualmente usado como simulador oficial na iniciativa RoboCup 3D Soccer Simulation League [42].

Zeitgeist facilita a troca, atualização e desenvolvimento de novas funcionalidades [43].

O simulador é constituído por 3 partes [44] [42] [45] [46]: O servidor, o monitor e os agentes. Na figura 3.1 podemos ver como as 3 partes comunicam com o servidor.

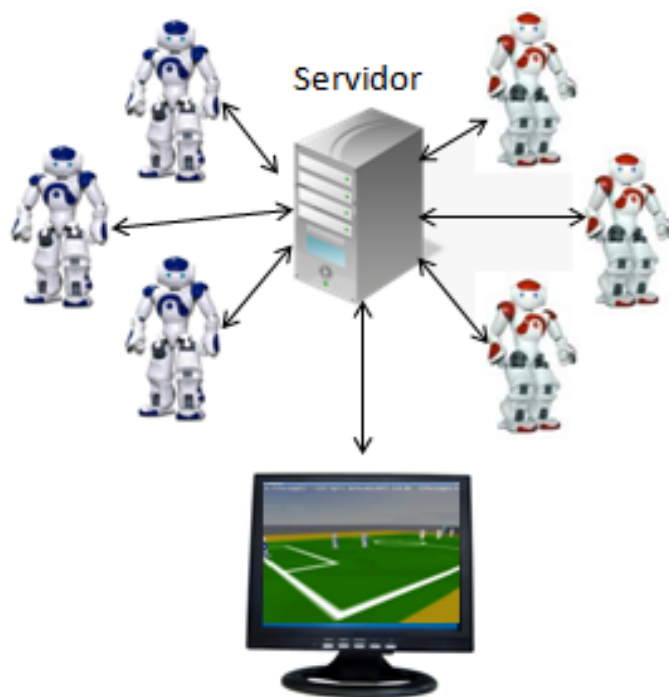


Figura 3.1: Comunicação entre os agentes, o servidor e o monitor.

3.3 Servidor

O servidor é responsável por gerir a simulação, isto é, atualizar o estado da simulação num ciclo contínuo. A arquitetura do servidor é [47]

¹http://simspark.sourceforge.net/wiki/index.php/Main_Page

- **Open Dynamics Engine²** — Módulo que simula a dinâmica do sistema e as propriedades físicas dos objetos simulados, providenciando juntas para os robôs, detecção de colisões com fricção e gravidade. ODE é particularmente útil para objetos simulados num ambiente de realidade virtual. ODE é multi-plataforma e dispõe de uma API escrita em C/C++ de fácil utilização.
- **Zeitgeist** — Conjunto de bibliotecas e classes para gerir dados dos objetos e componentes funcionais do sistema duma maneira uniforme [43]. Segue estritamente o paradigma orientado a objetos e usa a linguagem de programação C++.
- **Motor de simulação** — Representa o núcleo do simulador. Executa as operações de simulação através da troca de mensagens com os agentes com as ações dos mesmos e com informações sobre o ambiente.

Os objetos no ambiente podem alterar as propriedades de outros objetos, tais como a posição e velocidade, mas todos eles estão sobre o controlo de um processo que simula a física de corpos rígidos, isto é, deteta colisões, aplica gravidade, etc.

Outra função do servidor é gerir o processo com todos os agentes conectados. A cada ciclo o servidor recolhe de cada agente informação acerca dos seus atuadores [45] e responde com informação relativa ao estado do jogo. Esta informação inclui estado do jogo (tempo de jogo, resultado e estado) e informação dos sensores do robô (juntas, giroscópio, sensores dos pés e dados da visão).

A comunicação entre o servidor e o agente é feita usando os protocolos TCP ou UDP e as mensagens trocadas seguem um formato parecido ao protocolo LISP. A comunicação entre processos é proibida.

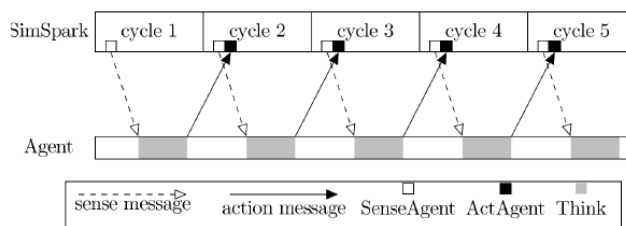


Figura 3.2: Diagrama de comunicação entre o servidor e o agente [42].

3.4 Monitor

Este módulo é responsável por desenhar a simulação numa janela para que os utilizadores possam visualizar o estado do jogo. Conecta-se a uma instância do servidor a partir do qual recebe continuamente atualizações referentes ao estado do jogo e estado de cada agente.

²<http://www.ode.org/>

A simulação pode ser vista em tempo real ou executar uma simulação guardada num ficheiro previamente guardado. O utilizador pode alterar a vista da câmara, mover a bola e outras operações recorrendo a teclas de atalho. A Figura 3.3 representa o monitor em execução.

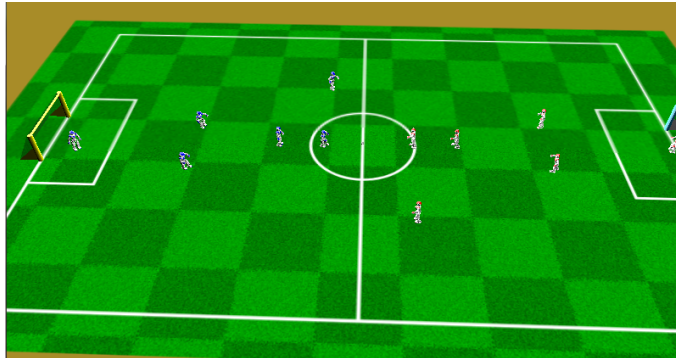


Figura 3.3: SimSpark Monitor em execução [48].

3.5 Modelo dos Agentes

O sistema permite simular diversos modelos de agentes. Os modelos usados na simulação 3D são o modelo HOPE-1 e o modelo NAO [44] (Figura 3.4b e Figura 3.5b respetivamente).

Ambos os modelos estão dotados com um giroscópio no torso, um sensor de força em cada pé que permite medir a força que o pé aplica no solo. Cada junta contém um atuador e um sensor. O atuador permite alterar a posição angular da junta, o sensor permite ler o valor atual da junta

3.5.1 HOPE-1

O HOPE-1 (Humanoid Open Platform Experiment) foi o primeiro modelo usado na simulação 3D, baseado no HOAP-2 da Fujitsu³. Inicialmente as equipas preocupavam-se mais com o equilíbrio e mobilidade básica do robô. Devido a esta preocupação inicial, o robô estava equipado com características não presentes no humano, tais como, câmara omnidirecional montada no peito, cabeça fixa, proporções físicas diferentes das do humano e sem limitações na rotação das juntas [42].

Devido a estas características não era apropriado para simular realisticamente o ser humano e foi uma das principais razões do modelo NAO (explicado na próxima secção) tornar-se tão popular entre as equipas e ter substituído o HOPE-1 na competição. Na figura 3.4 podemos ver o modelo virtual e anatomia do HOPE-1.

³<http://www.fujitsu.com/global/>

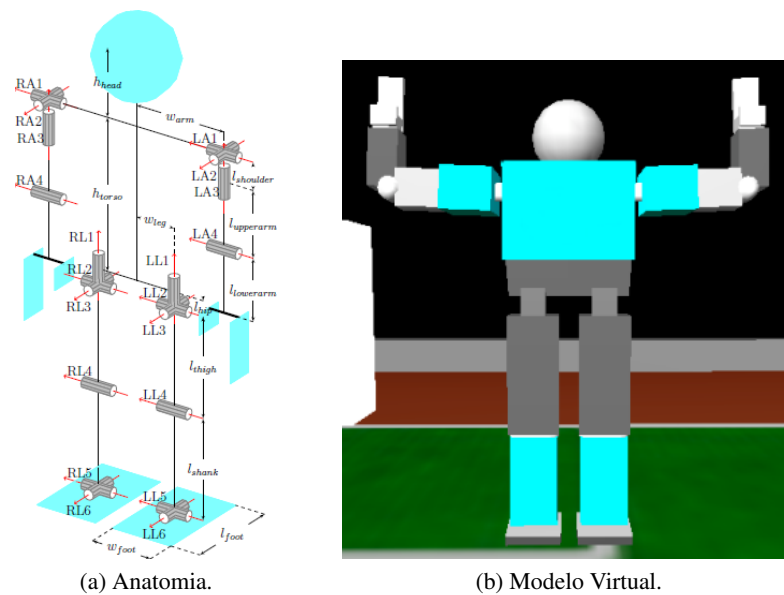


Figura 3.4: Robô HOPE-1 [42].

Descrição

O modelo HOPE-1 tem 3.5 m de altura, pesa 5.50 kg e contém 20 graus de liberdade (DOF), 4 em cada braço e 6 em cada perna [42]. Como as juntas não tinham limites era possível executar movimentos não naturais e como tinha uma câmara omnidirecional o pescoço não dispunha de nenhuma junta.

O giroscópio no torso fornece informação sobre a oscilação do robô e os sensores de força no pé fornecem informação sobre a força que o pé aplica no solo se houver contacto entre os mesmos.

Juntas

Uma configuração detalhada sobre as juntas do robô é apresentada na tabela 3.1 [42]. A tabela apresenta, para cada junta, o nome, tipo, parte do corpo associada e o eixo ao qual a junta pode mover-se em relação ao referencial XYZ.

Tabela 3.1: Configuração das juntas do modelo HOPE-1. Para efeitos de simplicidade apenas foram incluídas as juntas relativas a um braço e a uma perna. Os prefixos 'a' e 'l' no nome significam arm e leg (braço e perna) respetivamente

Nome	Actuador	Parte do corpo	Eixo de rotação
aj12	ae13	Ombro	(1,0,0) e (0,1,0)
aj3	ae2	Braço	(0,0,1)
aj4	ae4	Cotovelo	(1,0,0)
lj1	le1	Anca	(0,0,1)
lj23	le23	Coxa	(1,0,0) e (0,1,0)
lj4	le4	Joelho	(1,0,0)
lj56	le56	Tornozelo	(1,0,0) e (0,1,0)

3.5.2 NAO

Modelo utilizado atualmente na competição. Este modelo foi baseado no robô NAO da Aldebaran⁴ e partilha com este não só as características físicas como também a aparência. Na figura 3.5 podemos ver o modelo real, virtual e anatomia do NAO.

Descrição

O modelo NAO tem 57 cm de altura, pesa aproximadamente 4.5 kg e contém 22 graus de liberdade, 4 em cada braço, 6 em cada perna e 2 no pescoço. As juntas têm limite de rotação, como tal permitem um movimento natural e realista [42] [49].

O giroscópio no torso fornece informação sobre a oscilação do robô, os sensores de força no pé fornecem informação sobre a força que o pé aplica no solo se houver contacto entre os mesmos e conta com uma câmara direcional que simula a visão humana através do movimento realista da cabeça.

Juntas

Uma configuração detalhada sobre as juntas do robô é apresentada na tabela 3.2 [42] [49]. A tabela apresenta, para cada junta, o nome, tipo, parte do corpo associada e o eixo ao qual a junta pode mover-se em relação ao referencial XYZ.

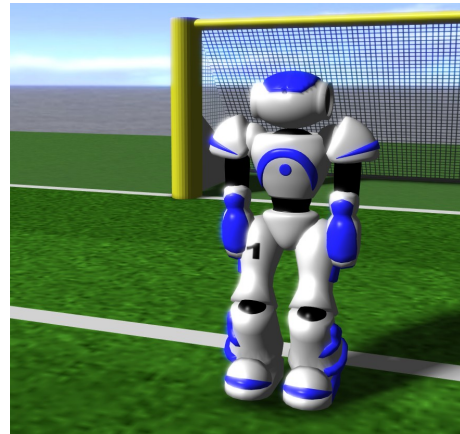
Tabela 3.2: Configuração das juntas do modelo NAO. Para efeitos de simplicidade apenas foram incluídas as juntas relativas a um braço e a uma perna. Os prefixos 'h', 'a' e 'l' no nome significam head, arm e leg (cabeça, braço e perna) respetivamente

<i>Nome</i>	Actuador	Parte do corpo	Eixo de rotação
hj1	he1	Pescoço	(0,0,1)
hj2	he2	Pescoço	(1,0,0)
aj1	ae1	Ombro	(1,0,0)
aj2	ae2	Ombro	(0,1,0)
aj3	ae3	Braço	(0,0,1)
aj4	ae4	Cotovelo	(1,0,0)
lj1	le1	Anca	(-0.7071,0,0.7071)
lj2	le2	Coxa	(0,1,0)
lj3	le3	Coxa	(1,0,0)
lj4	le4	Joelho	(1,0,0)
lj5	le5	Tornozelo	(1,0,0)
lj6	le6	Tornozelo	(0,1,0)

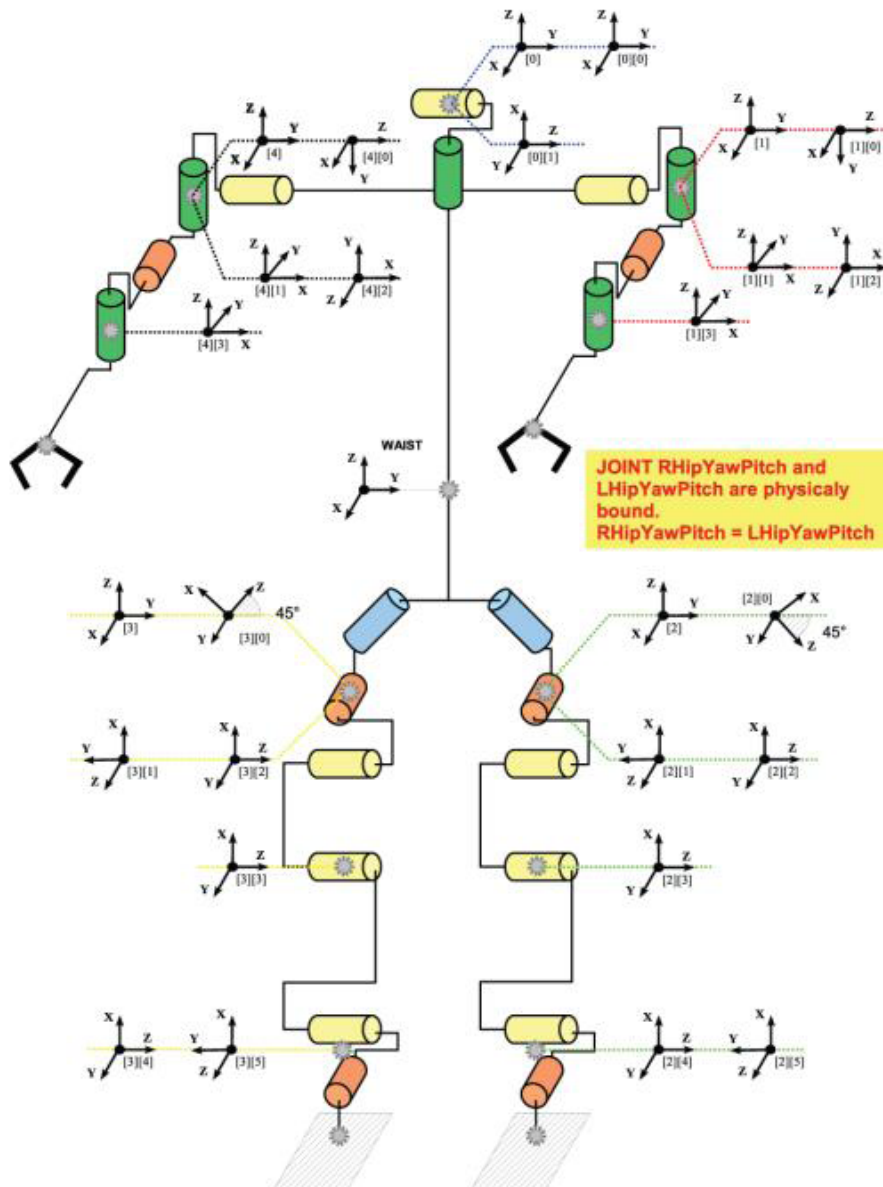
⁴<http://www.aldebaran-robotics.com/>



(a) Modelo real.



(b) Modelo virtual.



(c) Anatomia.

Figura 3.5: Robô NAO [42].

3.6 Ambiente e Objetos do Campo

3.6.1 Dimensões

As dimensões do campo são de 21 por 14 metros. Um campo da FIFA tem aproximadamente 110 por 70 metros, fazendo com que as dimensões do campo simulado sejam 20% do campo real. A área central tem um raio de 1.5 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 3.6.

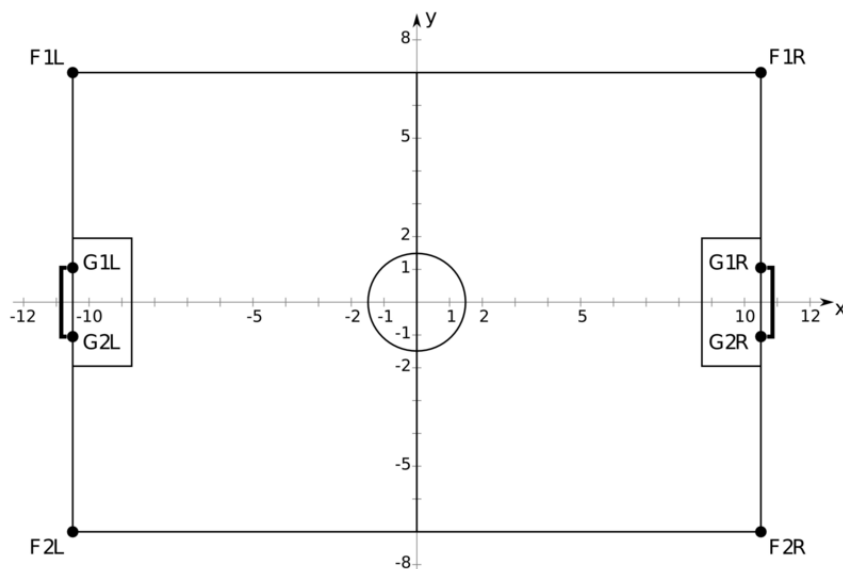


Figura 3.6: Dimensões do campo simulado [42].

3.6.2 Regras

As regras de jogo serão processadas por dois tipos de árbitros [44], 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 a detetar saída da bola da área limite do jogo. O árbitro humano está responsável apenas por iniciar a partida.

3.7 Agente FC Portugal 3D

3.7.1 Arquitetura

O agente da equipa FC Portugal 3D [50] 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 [47] [50]:

- **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 classe para ajudar nas necessidades de geometria. Contém classes para processar vetores, matrizes, polígonos, linhas, retângulos, círculos, etc.
- **Skills:** É neste pacote que são definidas as classes relacionadas com habilidades do robô, tais como andar, chutar, levanta, 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, processamento de API's, etc.
- **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, sensores, etc.
- **Optimization:** Contém classes usadas no processo de optimização.
- **Strategy:** Contém classes com as funções de alto nível do agente.

A figura 3.7 mostra a estrutura da arquitetura do agente.

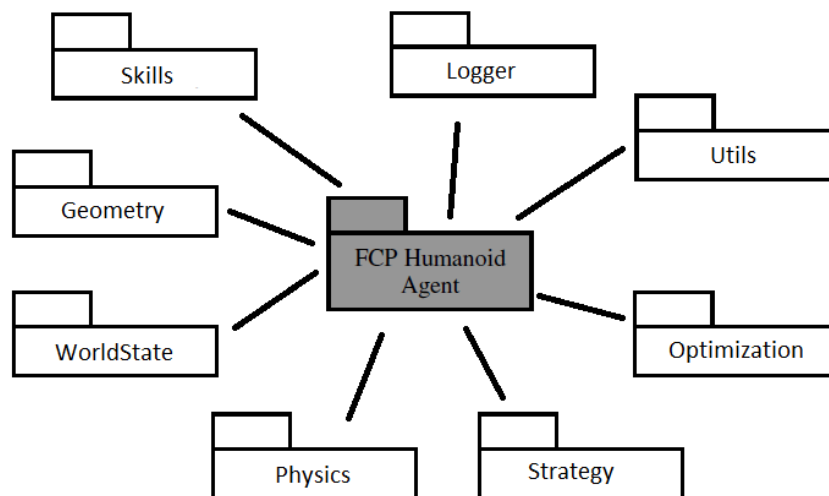


Figura 3.7: Arquitetura do agente da equipa FC Portugal 3D.

3.7.2 Modelo

A construção do modelo tem como base a leitura de um ficheiro XML que contém a estrutura do robô. Neste ficheiro é definido 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:

Listing 3.1: 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>

```

Esta estratégia de guardar informação sobre a estrutura do robô é útil na medida que torna o código mais genérico permitindo, com o mesmo código, definir outro tipo de robôs. Também podemos retirar deste ficheiro parâmetros importantes, como é o caso das matrizes de transformação usadas na cinemática.

Capítulo 4

Comportamentos

Neste capítulo são apresentados modelos de comportamentos que geram valores para as juntas do robô a partir de um conjunto de parâmetros. Estes modelos são usados para criar comportamentos para o robô tais como, andar, chutar, rodar, levantar. Os valores gerados podem ser valores iniciais e finais, sendo os iniciais os valores atuais das junções, ou valores que variam entre os valores iniciais e os finais num dado intervalo de tempo (seguindo uma trajetória).

4.1 Introdução

Um comportamento é uma ação (andar, chutar, rodar, levantar, etc.) que é executada pelo agente. Estes comportamentos são carregados no arranque e podem ser executados a qualquer altura do jogo as vezes que for necessário.

Na equipa FC Portugal estes comportamentos são definidos num ficheiro XML que contém os valores e parâmetros das juntas do robô.

Listing 4.1: Parte de um ficheiro XML (FCP agent - GetupFront.xml) onde é definido um comportamento

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<behavior name="GetupFront" type="SlotBehavior">

  <slot delta=".400">
    <move id="&head1;" angle="0" />
    <move id="&head2;" angle="0" />
    <move id="&lleg1;" angle="0" />
    <move id="&rleg1;" angle="0" />
    <move id="&lleg2;" angle="0" />
    <move id="&rleg2;" angle="0" />
  </slot>
</behavior>
```

```

<move id="&lleg3;" angle="0" />
<move id="&rleg3;" angle="0" />
<move id="&lleg4;" angle="0" />
<move id="&rleg4;" angle="0" />
<move id="&lleg5;" angle="0" />
<move id="&rleg5;" angle="0" />
<move id="&lleg6;" angle="0" />
<move id="&rleg6;" angle="0" />
<move id="&larm1;" angle="-90" />
<move id="&rarm1;" angle="-90" />
<move id="&larm2;" angle="0" />
<move id="&rarm2;" angle="0" />
<move id="&larm3;" angle="0" />
<move id="&rarm3;" angle="0" />
<move id="&larm4;" angle="0" />
<move id="&rarm4;" angle="0" />
</slot>
<slot delta=".200">
  <move id="&larm1;" angle="0" />
  <move id="&rarm1;" angle="0" />
  <move id="&larm2;" angle="0" />
  <move id="&rarm2;" angle="0" />
</slot>

```

Podemos ver que o ficheiro XML é definido como um ficheiro XML normal começando por indicar a versão XML usada seguido do nome do comportamento, modelo do comportamento e definição dos valores finais das juntas. Este exemplo é do tipo SlotBehavior que usa a interpolação sinusoidal para gerar os valores intermédios (entre os valores atuais e os finais definidos no resto do ficheiro) para as juntas.

O único parâmetro usado neste exemplo é o 'delta' que consiste na duração do movimento entre o valor inicial e o final de cada junta associada a esse instante.

4.2 Modelos de Comportamentos

Na criação de comportamentos é necessário dividir o valor inicial e o valor final de cada junta em diversos valores intermédios num dado intervalo de tempo, este conjunto de valores é chamado de trajetória do movimento. Os modelos matemáticos são usados para calcular trajetórias para as diversas juntas.

4.2.1 Step-Based

Este método usa a função degrau para calcular a trajetória. A trajetória é o resultado de várias funções degrau, cada uma definida num dado intervalo de tempo, onde a amplitude do degrau é o valor desejado para determinada junta do robô.

A equação da trajetória para cada junta é [41]:

$$f(t) = \sum_{i=0}^n \theta_i u_{A_i}(t) \quad (4.1)$$

onde n é o número de intervalos, A_i é o intervalo de tempo, θ_i é o valor desejado para a junta nesse intervalo e:

$$u_{A_i}(t) = \begin{cases} 1 & t \in A_i \\ 0 & t \notin A_i \end{cases} \quad (4.2)$$

4.2.1.1 Vantagens e desvantagens [41]

Algumas das vantagens deste método são:

- Simples de perceber e implementar;
- Simples de definir trajetórias.

Algumas das desvantagens deste método são:

- Nenhum controlo sobre a velocidade angular da trajetória;
- A sintaxe do ficheiro de configuração não é "amiga do utilizador";
- O modelo não é flexível.

4.2.2 Interpolação Sinusoidal

A interpolação sinusoidal permite um melhor controlo sobre cada junta durante a trajetória, suavizando a alteração entre o valor atual e o valor desejado da junta num período de tempo. Permite também definir, não só, o valor desejado para cada junta, mas também o tempo em que esses valores serão alcançados, bem como a velocidade angular inicial e final.

A equação da trajetória para cada junta é [41]:

$$f(t) = A \sin\left(\frac{\phi_f - \phi_i}{\delta} t + \phi_i\right) + \alpha, \quad \forall t \in [0, \delta] \quad (4.3)$$

onde δ é a duração entre o instante inicial e o instante final (desse slot), ϕ_i e ϕ_f é a fase inicial e final (controlo da velocidade angular), respetivamente, A é a amplitude, ou seja, o valor para a junta e α é o offset. Para o cálculo do valor de A e do α usam-se as seguintes expressões:

$$A = \frac{\phi_f - \phi_i}{\sin(\phi_f) - \sin(\phi_i)} \quad (4.4)$$

$$\alpha = \phi_i - A \sin(\phi_i) \quad (4.5)$$

4.2.2.1 Vantagens e desvantagens [41]

Algumas das vantagens deste método são:

- Simples de perceber e implementar;
- Período de tempo entre o ângulo inicial e o final é controlável;
- Algum controlo sobre as velocidades angulares;
- Controlo PID para o controlo de picos de valores;
- O modelo é flexível.

Algumas das desvantagens deste método são:

- Não é possível definir estruturas sinusoidais mais complexas;
- As velocidades angulares não são totalmente controláveis.

4.2.3 Central Pattern Generator

Central Pattern Generator são osciladores neuronais biológicos que geram comandos periódicos para movimentos rítmicos, tais como, a respiração, andar, correr, acenar, etc..., sobre o controlo de sinais simples. O estudo e desenvolvimento deste método pode ser visto em [41].

4.2.3.1 Vantagens e desvantagens [41]

Algumas das vantagens deste método são:

- É possível definir estruturas mais complexas;
- Velocidade angular é totalmente controlável;
- A marcha gerada é similar ao comportamento humano.

Algumas das desvantagens deste método são:

- Requerem conhecimentos complexos sobre o comportamento humano;
- As transições levam a movimentos mais lentos.

4.3 Conclusões

Existem diversos métodos para gerar comportamentos, cada um com as suas vantagens e desvantagens. Um determinado método pode produzir melhores resultados para um comportamento mas não significa que produza melhores resultados para todos os comportamentos.

O método 'Step-Based' apesar de ser de fácil implementação trás algumas desvantagens quanto ao controlo dos tempos de execução e velocidades das juntas. Estas desvantagens podem ser eliminadas com o método 'Interpolação sinusoidal'. O método 'Central Pattern Generator' é o mais parecido com o movimento humano e tem as vantagens dos dois métodos anteriores e nenhuma das suas desvantagens, em contrapartida requerem o conhecimento de comportamentos humanos complexos.

Cabe ao programador escolher o método que melhor se adapta ao movimento que pretende criar, ou poderá optar por a criação de um comportamento que combine os métodos estudados.

Capítulo 5

Chuto Omnidirecional

Para um jogo de futebol entre robôs ser bem sucedido muita atenção é dada no desenvolvimento de movimentos rápidos e robustos, isto é, ser flexível nas mudanças bruscas de direção e ser capaz de compensar para distúrbios externos como a colisão com outros robôs e a fricção com o solo.

Para o segundo movimento mais usado no futebol, chutar, também é necessário este tipo de robustez. Uma adaptação online sobre a trajetória do pé torna-se necessária para que o robô possa compensar as perturbações externas, chutando assim a bola com alguma precisão para o destino. Além disso, pequenos distúrbios durante a fase de execução do movimento podem fazer com que o chuto não seja preciso e como tal um bom mecanismo de equilíbrio sobre o pé também é um fator crucial.

Os movimentos de chuto mais comuns baseiam-se no uso de keyframes (quadro-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.

A ideia do uso de um chuto omnidirecional é fazer com que o chuto se torne mais flexível e eliminar a fase de preparação. Para isso, o robô terá que decidir online a direção que deve impor à bola e calcular uma trajetória para o pé que vai chutar para que a bola siga essa direção.

O desenvolvimento deste movimento (chuto omnidirecional) é constituído principalmente por três módulos: módulo de cinemática inversa, o módulo de planeamento de trajetória e o módulo de estabilidade (ver figura 5.1). O módulo de cinemática inversa tem como função calcular o valor de cada articulação do robô em função da posição do ponto que irá tocar na bola (neste caso o pé). O módulo de planeamento de trajetória tem como função calcular uma trajetória para o pé que vai chutar de forma a impor a bola com um movimento na direção pretendida. O módulo de estabilidade tem como funcionalidade a estabilidade do robô (equilíbrio numa perna) durante

a execução do movimento. No final, estes três módulos são combinados para efetuar o chuto omnidirecional.

Neste capítulo também será feito uma breve descrição da álgebra matricial envolvida nas translações e rotações das articulações do robô.

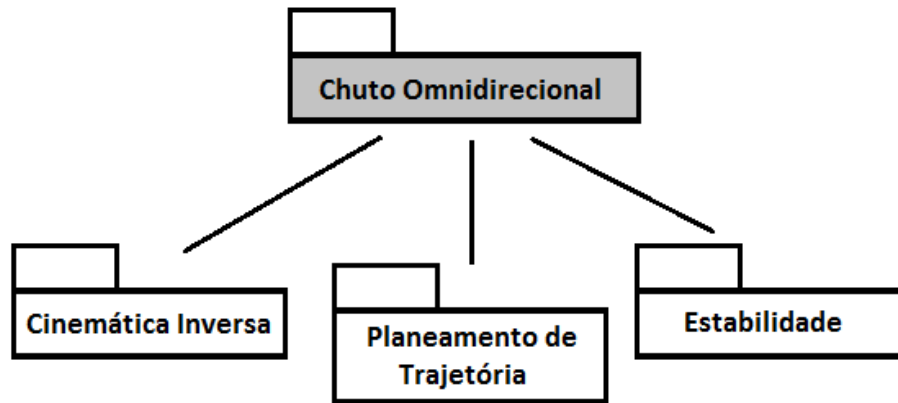


Figura 5.1: Arquitetura do comportamento a desenvolver.

5.1 Movimentos Rígidos e Transformações Homogéneas

Uma grande parte do estudo da cinemática de um robô é a determinação de sistemas de coordenadas para representar as posições e orientações das partes rígidas do robô, bem como as transformações entre referenciais.

Nesta secção estudamos as operações de translação e rotação, e o conceito de transformação homogénea. Uma transformação homogénea combina as operações de translação e rotação numa única matriz e são muito importantes para o cálculo da cinemática direta e inversa de um robô. Os conceitos apresentados foram baseados de [51].

Começamos o estudo com a representação de pontos e vetores num espaço euclidiano, no qual estão presentes diversos referenciais. Esta representação poderá ser vista como uma translação dum determinado referencial para a posição do ponto. Em seguida, é apresentado o conceito de matriz de rotação para representar a orientação de um ponto relativamente a um dado sistema de coordenadas (o ponto está na origem do sistema). No final é explicado a representação de uma translação e de uma rotação através do uso de uma matriz de transformação homogénea. Uma matriz de transformação homogénea além de representar a posição e orientação de um ponto em relação a um determinado sistema de coordenadas pode também representar transformações entre sistemas de coordenadas.

5.1.1 Representação de Pontos e Vetores

Na figura 5.2 está representada uma figura que resume os pontos que vão ser explicados nesta secção. Na figura estão representados dois referenciais, $o_0x_0y_0$ e $o_1x_1y_1$ ¹, diferenciados por uma rotação de 45°. Estes referenciais são chamados de sistema de coordenadas cartesiano e são importantes visto, em robótica, utilizar-se coordenadas cartesianas para definir rotações e translações.

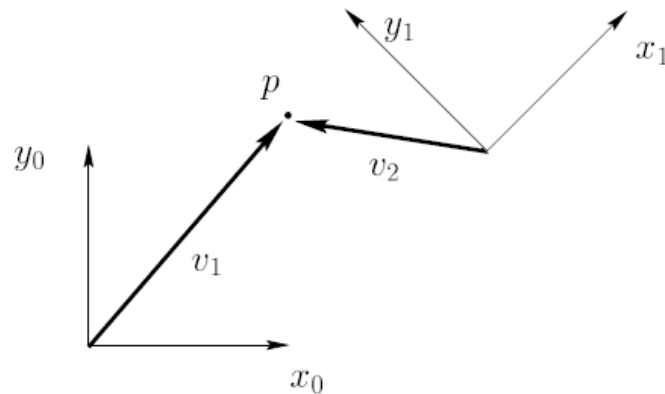


Figura 5.2: Dois referenciais de coordenadas, um ponto p e dois vetores v_1 e v_2 [51].

Considere mais uma vez a figura 5.2. Podemos definir as coordenadas do ponto p tanto em relação a $o_0x_0y_0$ como em relação a $o_1x_1y_1$, e para distinguirmos em relação a que referencial nos estamos a referir, ao longo do capítulo será adotado um subscrito para o efeito. Assim temos:

$$p^0 = \begin{bmatrix} 5 \\ 6 \end{bmatrix}, \quad p^1 = \begin{bmatrix} -2.8 \\ 4.2 \end{bmatrix}$$

onde p representa uma entidade geométrica, um ponto no espaço, enquanto p^0 e p^1 ² representam a localização desse ponto no espaço em relação ao referencial $o_0x_0y_0$ e $o_1x_1y_1$, respetivamente.

Por outro lado, como a origem dum referencial é um ponto no espaço, podemos definir coordenadas para representar a posição da origem dum referencial em relação a outro referencial. Na figura 5.2 temos:

$$o_1^0 = \begin{bmatrix} 10 \\ 5 \end{bmatrix}, \quad o_0^1 = \begin{bmatrix} -10.6 \\ 3.5 \end{bmatrix}$$

onde o_1^0 ³ representa a posição da origem do referencial $o_1x_1y_1$ em relação a $o_0x_0y_0$ e o_0^1 representa a posição da origem do referencial $o_0x_0y_0$ em relação a $o_1x_1y_1$.

Já vimos como representar pontos no espaço, de seguida será mostrado como representar translações. Para haver uma translação é preciso definir-se a direção e a magnitude da translação, estes parâmetros são definidos por um elemento geométrico chamado vetor. Para atribuir coordenadas

¹ $o_i x_i y_i$ representa o referencial com identificador i

² p^k representa a localização no espaço do ponto p em relação ao referencial $o_k x_k y_k$

³ p_i^k representa a localização no espaço do ponto p_i em relação ao referencial $o_k x_k y_k$

a um vetor é usado o mesmo sistema que a atribuição de coordenadas a um ponto, e salienta-se que um vetor é invariante à escolha do sistema de coordenadas mas a sua representação depende diretamente do referencial escolhido. Da figura 5.2 obtemos:

$$v_1^0 = \begin{bmatrix} 5 \\ 6 \end{bmatrix}, \quad v_1^1 = \begin{bmatrix} 7.77 \\ 0.8 \end{bmatrix}, \quad v_2^0 = \begin{bmatrix} -5.1 \\ 1 \end{bmatrix}, \quad v_2^1 = \begin{bmatrix} -2.89 \\ 4.2 \end{bmatrix}$$

5.1.2 Representação de Rotações

Uma matriz de rotação permite-nos representar a orientação de um referencial em relação a um outro referencial. Assim, é possível, recorrendo a uma matriz deste tipo, representar a orientação de um corpo rígido em relação a um referencial.

Para representarmos a posição e a orientação relativa dum corpo rígido em relação a outro, começamos por anexar a cada um dos corpos um referencial e depois determina-se a transformação geométrica entre eles.

Na secção 5.1.1 vimos como representar a localização no espaço da origem dum referencial em relação a outro referencial. Nesta secção iremos ver como representar a orientação de um referencial em relação a outro referencial.

Começaremos o estudo com representação de rotações no plano e depois generalizamos para o caso da representação de rotações no espaço.

Rotação no Plano

Em álgebra linear, uma matriz de rotação é uma matriz que é usada para executar uma rotação no espaço euclidiano. Por exemplo, a matriz de rotação

$$R = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix} \quad (5.1)$$

roda pontos no plano xy , no sentido anti-horário, de um ângulo de θ em relação à origem do referencial. Esta matriz é conhecida como uma matriz de rotação básica.

Na figura 5.3 podemos ver um referencial $o_1x_1y_1$ rodado θ em relação ao referencial $o_0x_0y_0$, sendo $R_1^0 = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix}$ a matriz que representa a transformação geométrica entre o referencial $o_1x_1y_1$ e o referencial $o_0x_0y_0$. Neste caso temos,

$$o^0 = R_1^0 o^1 \quad (5.2)$$

Para executar uma rotação usando a matriz R , a posição de cada ponto deve ser representado por um vetor coluna v , contendo as coordenadas do ponto. Obtém-se o vetor rodado multiplicando a matriz de rotação R pelo vetor v ($v' = R.v$). Matrizes de rotação só podem ser usadas para rotações em torno da origem do referencial.

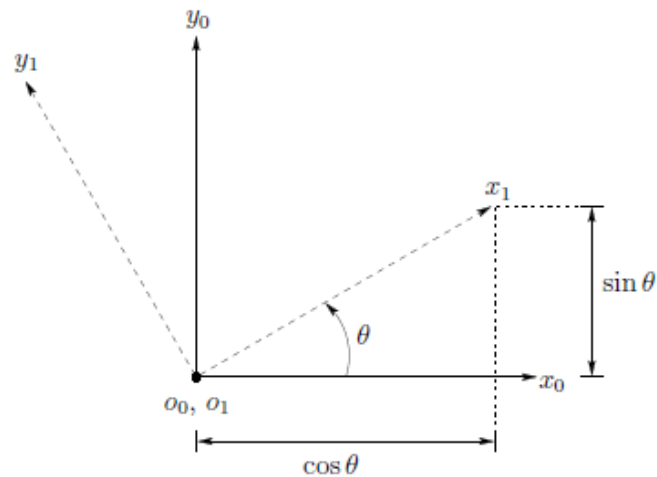


Figura 5.3: Orientação de um referencial em relação a outro (2D) [51].

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:

- $R^T = R^{-1}$ (ortogonal)
- As colunas/linhas de R são ortogonais entre si
- Cada coluna/linha de R é um vetor unitário
- $\det(R) = 1$

Rotação no Espaço

Uma matriz de rotação no espaço executa uma rotação em torno dum eixo que passa pelo centro do referencial. Esta matriz também pertence ao grupo $SO(3)$.

A notação usada neste capítulo para representar rotações no espaço será $R_{i,k}$ que representa uma rotação de k graus em torno do eixo i .

As matrizes de rotação básicas no espaço euclideano são as seguintes:

$$Rot_{x,\alpha} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\alpha & -\sin\alpha \\ 0 & \sin\alpha & \cos\alpha \end{bmatrix} \quad (5.3)$$

$$Rot_{y,\beta} = \begin{bmatrix} \cos\beta & 0 & \sin\beta \\ 0 & 1 & 0 \\ -\sin\beta & 0 & \cos\beta \end{bmatrix} \quad (5.4)$$

$$Rot_{z,\gamma} = \begin{bmatrix} \cos\gamma & -\sin\gamma & 0 \\ \sin\gamma & \cos\gamma & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (5.5)$$

Para rodarmos um ponto em torno de um eixo o cálculo é similar ao já visto anteriormente ($v' = R_{i,k}v$). Na figura 5.4 temos,

$$p_a = R_{z_0,\pi} p_b \quad (5.6)$$

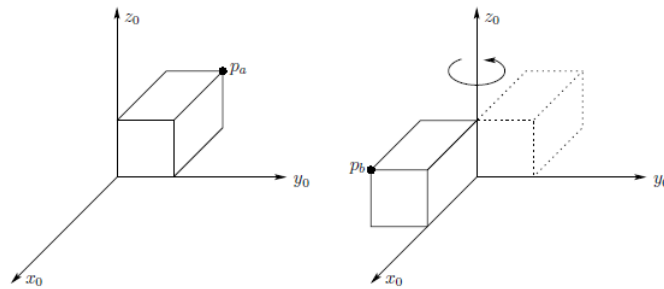


Figura 5.4: Rotação de um ponto em torno do eixo Z [51].

Outra funcionalidade das matrizes de rotação no espaço é representar a orientação de um referencial em relação a outro. Na figura 5.5 podemos ver um referencial $o_1x_1y_1$ que resulta da rotação do referencial $o_0x_0y_0$ θ° em torno do eixo Z. Neste caso temos,

$$o^0 = R_{z,\theta} o^1 \quad (5.7)$$

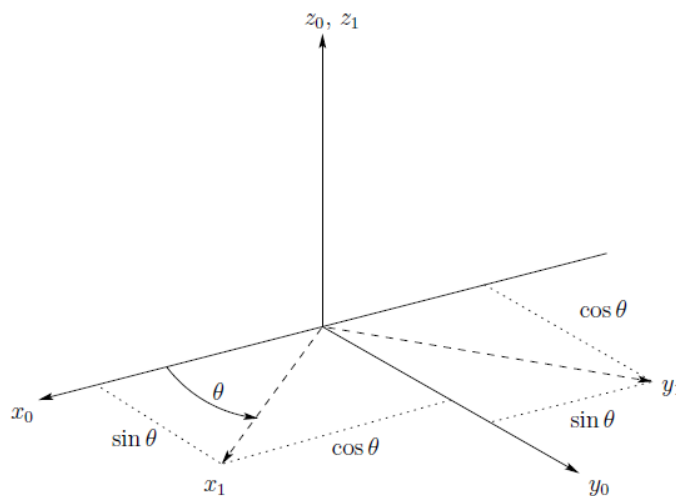


Figura 5.5: Orientação de um referencial em relação a outro (3D) [51].

5.1.3 Movimentação de Corpos Rígidos

Nesta secção iremos combinar os conceitos de translação e rotação, estudados nas secções 5.1.1 e 5.1.2 respetivamente, para assim definir um movimento rígido.

Um movimento rígido é um par ordenado (d, R) onde $d \in \mathbb{R}^3$ e $R \in SO(3)$. O grupo de todos os movimentos rígidos chama-se grupo especial Euclideano (Special Euclidean Group) e é denotado como $SE(3) = \mathbb{R}^3 \times SO(3)$ [51].

Um movimento rígido é constituído por uma translação e por uma rotação puras. Na figura 5.6 vemos que o referencial $o_1x_1y_1z_1$ é obtido através de uma rotação R_1^0 e de uma translação $d_{0,1}$ do referencial $o_0x_0y_0z_0$ e o referencial $o_2x_2y_2z_2$ é obtido através de uma rotação R_2^1 e de uma translação $d_{1,2}$ do referencial $o_1x_1y_1z_1$.

Da figura 5.6 também podemos verificar que se tivermos duas movimentações rígidas

$$p^0 = R_1^0 p^1 + d_1^0 \quad (5.8)$$

e

$$p^1 = R_2^1 p^2 + d_2^1 \quad (5.9)$$

então a sua combinação define uma nova movimentação que podemos descrever substituindo p^1 da equação 5.9 na equação 5.8

$$p^0 = R_1^0 R_2^1 p^2 + R_1^0 d_2^1 + d_1^0 \quad (5.10)$$

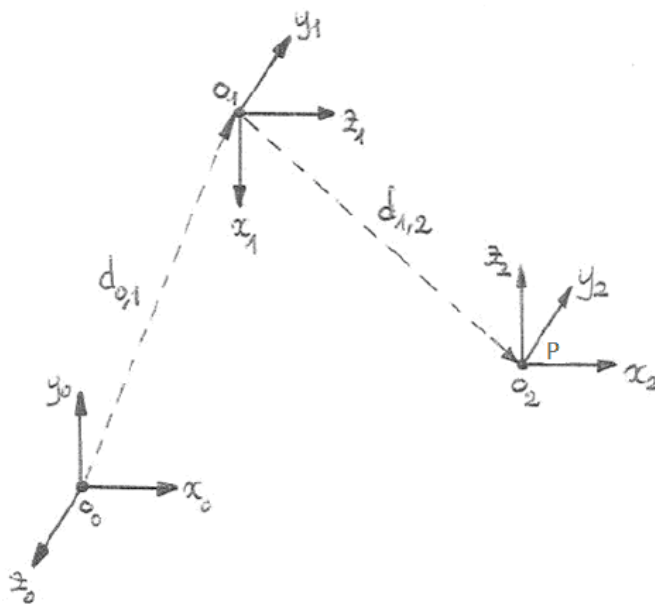


Figura 5.6: Movimentação de um corpo rígido [52].

5.1.4 Transformações Homogéneas

Da equação 5.10 podemos ver que se torna impraticável a determinação de uma movimentação que resulta da combinação de várias movimentações. Nesta secção veremos como uma movimentação rígida pode ser representada sobre a forma de uma matriz fazendo com que a composição de várias movimentações rígidas possam ser reduzidas a multiplicação de matrizes, como foi o caso das rotações [51].

Uma transformação homogénea combina a rotação e a translação numa única matriz e é representada por:

$$H = \begin{bmatrix} R & d \\ 0 & 1 \end{bmatrix} \quad (5.11)$$

onde $R \in SO(3)$, $d \in \mathbb{R}^3$ e 0 é um vetor linha $[0 \ 0 \ 0]$.

Como a matriz R é ortogonal podemos verificar que a transformação inversa H^{-1} é dada por:

$$H^{-1} = \begin{bmatrix} R^T & -R^T d \\ 0 & 1 \end{bmatrix} \quad (5.12)$$

Para representar a transformação dada pela equação 5.8 começamos por adicionar uma quarta componente com o valor 1 aos vetores p^0 e p^1 ,

$$p^0 = \begin{bmatrix} p^0 \\ 1 \end{bmatrix} \quad (5.13)$$

$$p^1 = \begin{bmatrix} p^1 \\ 1 \end{bmatrix} \quad (5.14)$$

de seguida podemos facilmente verificar que a equação 5.8 é equivalente à transformação homogénea

$$p^0 = H_1^0 p^1 \quad (5.15)$$

onde $H_1^0 = \begin{bmatrix} R_1^0 & d_{0,1}^0 \\ 0 & 1 \end{bmatrix}$

Podemos definir um conjunto de transformações homogéneas básicas através de:

$$Trans_{x,a} = \begin{bmatrix} 1 & 0 & 0 & a \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}; \quad Rot_{x,\alpha} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos\alpha & -\sin\alpha & 0 \\ 0 & \sin\alpha & \cos\alpha & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (5.16)$$

$$Trans_{y,b} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & b \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}; \quad Rot_{y,\beta} = \begin{bmatrix} \cos\beta & 0 & \sin\beta & 0 \\ 0 & 1 & 0 & 0 \\ -\sin\beta & 0 & \cos\beta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (5.17)$$

$$Trans_{z,c} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & c \\ 0 & 0 & 0 & 1 \end{bmatrix}; \quad Rot_{z,\gamma} = \begin{bmatrix} \cos\gamma & -\sin\gamma & 0 & 0 \\ \sin\gamma & \cos\gamma & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (5.18)$$

5.2 Cinemática Inversa

O problema da cinemática inversa consiste em determinar os parâmetros de cada articulação de forma colocar um dado ponto do nosso objeto numa determinada localização no espaço. 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 [51].

Existem diversos algoritmos de cálculo de cinemática inversa mas cada caso é um caso e, às vezes torna-se necessário adaptar e combinar métodos para obter uma solução para o nosso problema.

Neste trabalho usaremos um método adaptado de [53] no qual é usada a aproximação geométrica para determinar os parâmetros das articulações de uma perna do robô NAO. Começaremos por identificar os referenciais associados a cada articulação. O resultado está apresentado na figura 5.7.

Antes de começarmos o cálculo é necessário definir a representação da orientação e posição do pé. Esta representação é dada por um matriz de transformação homogênea e a orientação e posição é em relação ao referencial chamado na figura 5.7 de bacia. Assim temos

$$H_{Pe}^{Bacia} = \begin{bmatrix} R_{Pe}^{Bacia} & d_{Pe}^{Bacia} \\ 0 & 1 \end{bmatrix} \quad (5.19)$$

Esta matriz vai servir de entrada de dados para o cálculo da cinemática inversa do robô, como tal terá de ser fornecida manualmente ou automaticamente por outro módulo de mais alto nível.

De seguida temos que saber a posição (no decorrer do processo calcular a posição significa a posição e a orientação) em relação à anca. Determina-se facilmente com uma translação segundo o eixo y ⁴

$$H_{Pe}^{Anca} = Trans_y\left(\frac{l_{dist}}{2}\right) \cdot H_{Pe}^{Bacia} \quad (5.20)$$

onde l_{dist} é a distância entre pernas, ou seja, a distância entre o início de uma perna e o início da outra.

Como a articulação situada na anca do robô físico se encontra rodada 45° em torno do eixo x (ver referencial AncaOrtogonal da figura 5.7) é necessário calcular a posição em relação a este referencial, AncaOrtogonal. Trata-se de uma rotação de 45 graus em torno do eixo x e obtemos a

⁴A notação usada para as transformações homogêneas elementares de rotação e translação é $Rot_{<eixo>}(angulo)$ e $Trans_{<eixo>}(translacao)$, respetivamente

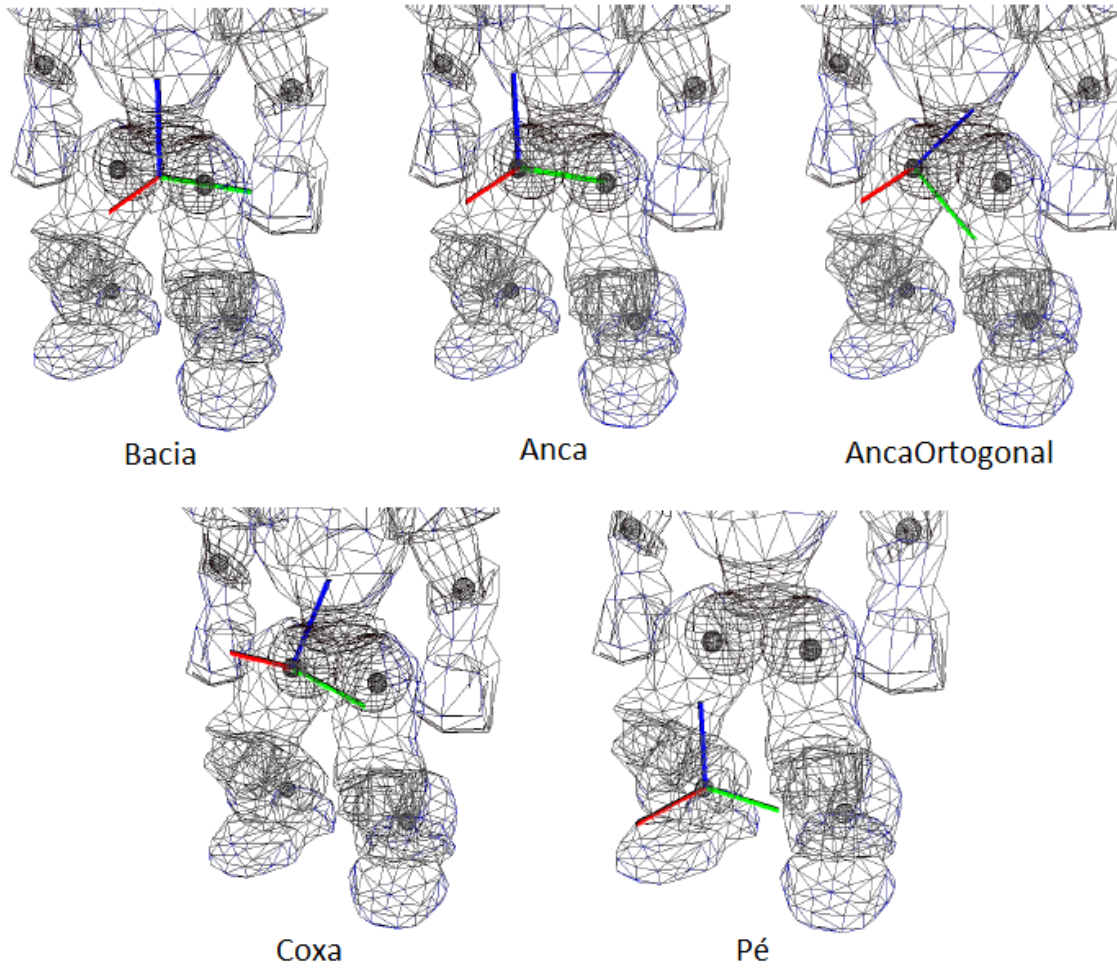


Figura 5.7: Configuração dos referenciais usados na cinemática inversa. Vermelho=eixo-x, verde=eixo-y e azul=eixo-z [53].

transformação homogênea

$$H_{Pe}^{AncaOrtogonal} = Rot_x\left(\frac{\pi}{4}\right) \cdot H_{Pe}^{Anca} \quad (5.21)$$

De seguida iremos calcular os parâmetros da articulação do joelho e do pé. Como podemos ver pela figura 5.8 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 cosenos 5.22 podemos determinar os ângulos do triângulo. O ângulo γ entre $\overline{l_{coxa}}$ e $\overline{l_{tibia}}$ é dado pela equação 5.23

$$c^2 = a^2 + b^2 - 2.a.b.\cos\gamma \quad (5.22)$$

$$\gamma = \arccos \frac{l_{coxa}^2 + l_{tibia}^2 - l_{trans}^2}{2.l_{coxa}.l_{tibia}} \quad (5.23)$$

Através da equação 5.23 determinamos o ângulo do joelho δ_{joelho} . Este ângulo é zero quando

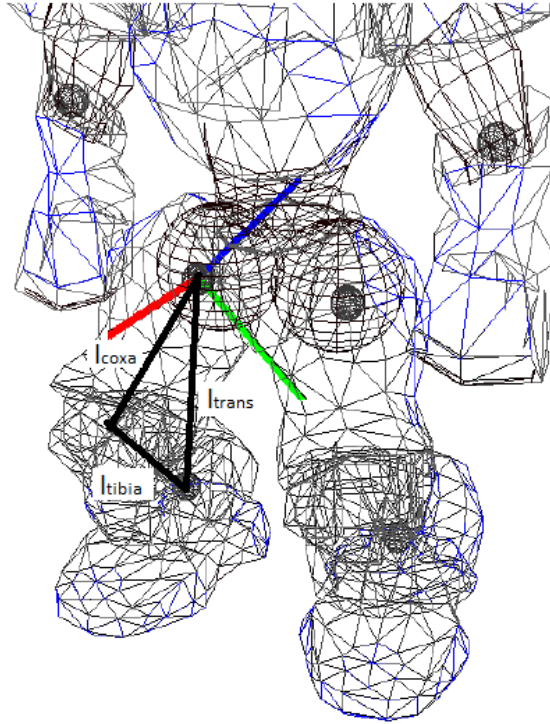


Figura 5.8: Triângulo constituído pelo joelho e pelos membros da perna. Adaptado de [53]

a perna está esticada, logo

$$\delta_{joelho} = \pi - \gamma \quad (5.24)$$

O ângulo oposto a l_{coxa} também tem que ser calculado visto corresponder ao ângulo pitch⁵ do pé:

$$\delta_{ppitch1} = \arccos \frac{l_{tibia}^2 + l_{trans}^2 - l_{coxa}^2}{2 \cdot l_{tibia} \cdot l_{trans}} \quad (5.25)$$

Se combinarmos o triângulo e a articulação do pé veremos que formam uma espécie de unidade pan-tilt⁶. O ângulo pitch e roll⁷ dessa articulação pode ser calculado usando atan2.

$$\delta_{ppitch2} = \text{atan2}(x, \sqrt{y^2 + z^2}) \quad (5.26)$$

$$\delta_{peRoll} = \text{atan2}(y, z) \quad (5.27)$$

onde x, y, z são as componentes do vetor translação da matriz $H_{Pe}^{AncaOrtogonal}$. Como o ângulo pitch do pé é composto por duas partes temos que somar as duas para obter o valor combinado.

$$\delta_{pePitch} = \delta_{ppitch1} + \delta_{ppitch2} \quad (5.28)$$

Com os ângulos do joelho e do pé calculados podemos calcular os ângulos correspondentes

⁵Rotação em torno do eixo y

⁶Sistema usado, por exemplo, pelas câmaras de vigilância.

⁷Rotação em torno do eixo x

à articulação situada na anca. Para calcular estes ângulos adicionamos um referencial auxiliar (chamado de Coxa na figura 5.7) e extraímos esses ângulos diretamente da matriz de transformação homogênea $H_{AncaOrtogonal}^{Coxa}$, que pode ser calculada por

$$H_{AncaOrtogonal}^{Coxa} = (H_{Coxa}^{Pe})^{-1} \cdot H_{AncaOrtogonal}^{Pe} \quad (5.29)$$

onde

$$H_{AncaOrtogonal}^{Pe} = (H_{Pe}^{AncaOrtogonal})^{-1} \quad (5.30)$$

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

$$H_{Coxa}^{Pe} = Rot_x(\delta_{peRoll}) \cdot Rot_y(\delta_{pePitch}) \cdot Trans_z(l_{tibia}) \cdot Rot_y(\delta_{joelho}) \cdot Trans_z(l_{coxa}) \quad (5.31)$$

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 $\cos\delta_x$).

$$H_{AncaOrtogonal}^{Coxa} = Rot_z(\delta_z) \cdot Rot_x(\delta_x) \cdot Rot_y(\delta_y) = \begin{bmatrix} c_y c_z - s_x s_y s_z & -c_x s_z & c_z s_y + c_y s_x s_z \\ c_z s_x s_y + c_y s_z & c_x c_z & -c_y c_z s_x + s_y s_z \\ -c_x s_y & s_x & c_x c_y \end{bmatrix} \quad (5.32)$$

donde concluímos que

$$\delta_x = \arcsin(r_{21})^8 \quad (5.33)$$

O ângulo δ_z retiramos de

$$\frac{-r_{01}}{r_{11}} = \tan\delta_z \quad (5.34)$$

$$\delta_{ancaYaw} = \delta_z = \text{atan2}(-r_{01}, r_{11}) \quad (5.35)$$

e seguindo o mesmo raciocínio

$$\delta_{ancaPitch} = \delta_y = \text{atan2}(-r_{20}, r_{22}) \quad (5.36)$$

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

$$\delta_{ancaRoll} = \delta_x - \frac{\pi}{4} = \arcsin(r_{21}) - \frac{\pi}{4} \quad (5.37)$$

Todos os parâmetros das articulações foram calculados. Está mostrado na tabela 5.1 um resumo desses valores.

⁸O primeiro índice representa a linha, e o segundo a coluna da matriz. Os índices começam em zero.

Tabela 5.1: Valores da articulações calculados através da cinemática inversa.

Articulação	Equação
δ_{joelho}	5.24
δ_{peRoll}	5.27
$\delta_{pePitch}$	5.28
$\delta_{ancaRoll}$	5.37
$\delta_{ancaPitch}$	5.36
$\delta_{ancaYaw}$	5.35

5.3 Planeamento de Trajetória

5.3.1 Introdução

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.

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 iremos determinar uma função que nos devolva uma curva representativa da trajetória que o pé deve seguir. Para o efeito iremos usar 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 ajuda-se no desenho de formas [54].

Uma curva de Bézier é uma curva paramétrica, e um caso especial de uma interpolação Hermittiana, que faz uma interpolação entre um ou mais pontos, no plano ou no espaço. Esta interpolação pode ser de grau 1 (linear), de grau 2 (quadrática), de grau 3 (cúbica) ou de grau maior que 3 [55]. Através da equação 5.38 [56] [57] [54] podemos determinar uma curva de Bézier de grau n que permite determinar uma trajetória com elevado grau de complexidade.

$$b(t) = \sum_{i=0}^n \binom{n}{i} t^i (1-t)^{n-i} P_i, \quad t \in [0, 1] \quad (5.38)$$

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

Na figura 5.9 podemos ver exemplos de curvas de Bézier de diferentes graus.

Este tipo de curvas são amplamente usados no desenho de tipos de letra na área da informática. A figura 5.10 mostra a definição da letra 'g' usando curvas de Bézier [54].

Outra vantagem do uso de curvas de Bézier deve-se a serem acessíveis a todo o tipo de utilizadores, evitando assim toda a complexidade matemática na base do cálculo. Algumas características são [58]:

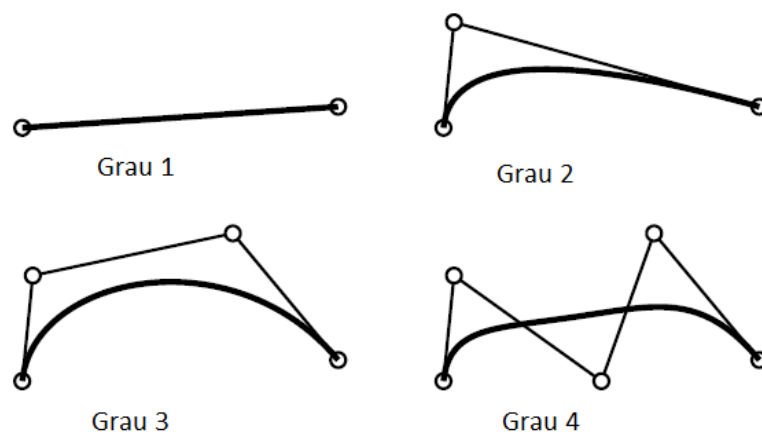


Figura 5.9: Curvas de Bézier de 1º, 2º, 3º e 4º grau [54].

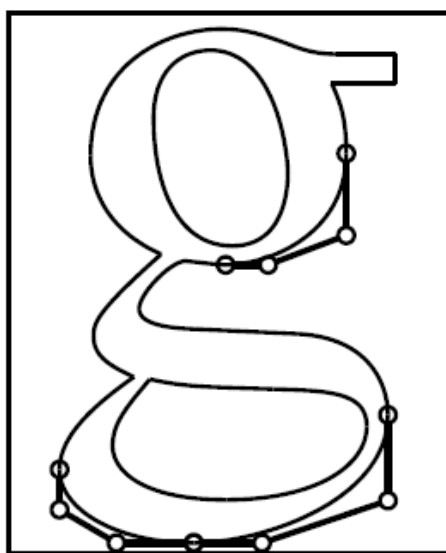


Figura 5.10: Definição de letras usando curvas de Bézier [54].

- 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).

5.3.2 Propriedades

Nesta seção são apresentadas algumas propriedades, importantes para este trabalho, das curvas de Bézier [58] [59].

- O grau de uma curva de Bézier constituída por $n+1$ pontos é n ;
- Uma curva de Bézier começa sempre no ponto P_0 e termina no ponto P_n ;
- A tangente nos pontos P_0 e P_n é dado pelos segmentos de reta $\overline{P_0P_1}$ e $\overline{P_{n-1}P_n}$, respetivamente;
- As curvas de Bézier são simétricas, isto é, se invertermos a ordem dos pontos de controlo obtemos a mesma curva;
- A curva situa-se no invólucro convexo (do inglês "Convex Hull") definido por todos os pontos de controlo.

5.3.3 Construção

Para a construção de uma curva de Bézier de grau n , precisaremos de 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 5.11. Após determinar o grau da curva (n) e os pontos de controlo, usar a equação 5.38 para determinar $b(t)$, que corresponde à localização do ponto no instante t .

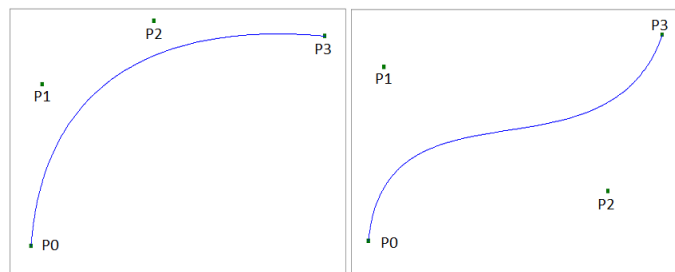


Figura 5.11: Duas curvas cúbicas de Bézier com diferentes pontos de controlo.

Relativamente ao nosso problema queremos 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 5.38 obtemos uma curva de Bézier paramétrica de grau 3 (cúbica) que nos fornece a localização de um ponto num determinado instante. Essa curva é dada por:

$$b(t) = (1-t)^3 P_0 + 3(1-t)^2 t P_1 + 3(1-t)t^2 P_2 + t^3 P_3, \quad t \in [0, 1] \quad (5.39)$$

5.4 Estabilidade

Para a execução de um movimento preciso e realista por parte do robô é necessário que este mantenha o equilíbrio durante a execução do mesmo. O problema da estabilidade do robô esta

ligado ao seu equilíbrio e podemos ver este problema como um problema de cálculo de centro de massa.

Na Física, o centro de massa de um corpo é um ponto do espaço onde, para efeitos de cálculo, toda a massa do corpo está concentrada [60] [61]. No contexto do nosso problema este ponto deverá situar-se no eixo do pé no qual todo o corpo se apoiará enquanto é executado o movimento (caso o chuto seja com o pé direito o ponto será no eixo do pé esquerdo e vice versa) o que faz com que o referencial de origem para o cálculo seja a posição do pé de apoio. Na equação 5.40 temos a fórmula de cálculo do centro de massa.

$$\vec{com} = \frac{\sum_{i=0}^n \vec{parte}_i \cdot m_i}{m_{total}} \quad (5.40)$$

onde $parte_i$ refere-se a cada membro (do corpo) físico do robô (p.e. coxa), m_i refere-se à massa da $parte_i$ e m_{total} é a soma das massas de todas as partes do corpo.

Do resultado deste cálculo concluímos que se o ponto estiver no interior da projeção do pé de apoio no plano do campo (retângulo com largura e comprimento do pé) o robô encontra-se em equilíbrio (estável) caso contrário poderá estar em desequilíbrio (instável). Na figura 5.12 podemos ver o diagrama de execução deste módulo.

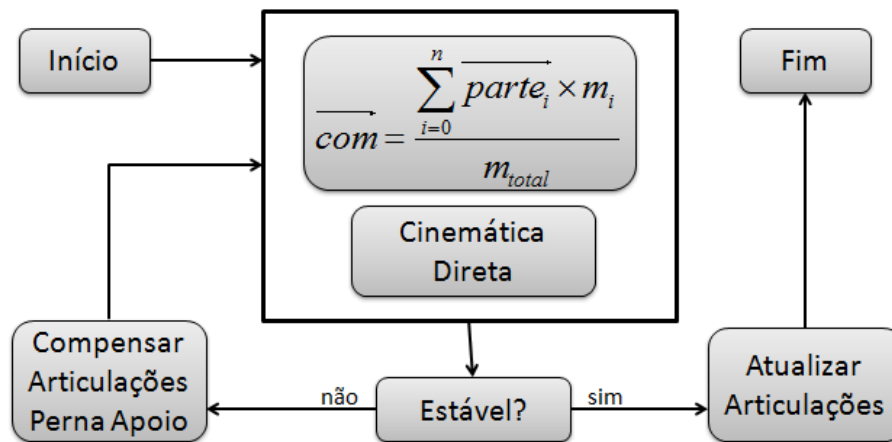


Figura 5.12: Diagrama do módulo de estabilidade.

5.5 Algoritmo de Execução do Chuto Omnidirecional

Nas secções anteriores deste capítulo caracterizaram-se os módulos necessários para executar um chuto omnidirecional. Nesta secção definimos como esses módulos são combinados para criar um comportamento que execute o chuto omnidirecional.

Para além dos módulos é necessário definir os parâmetros usados no comportamento. São eles:

- $dist_{KB}$: distância entre o pé que vai chutar e a bola.
- α : ângulo entre a orientação do pé e a posição da bola em relação ao pé.
- β : ângulo entre a posição da bola em relação ao pé e a posição de destino da bola em relação à bola.

Estes parâmetros podem ser visualizados na figura 5.13.

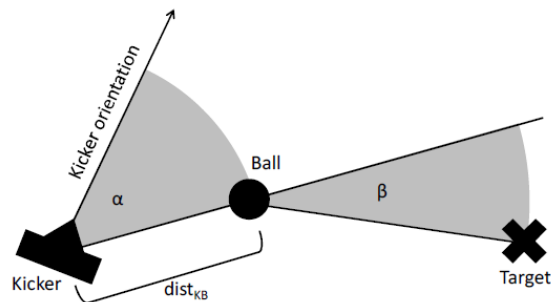


Figura 5.13: Parâmetros necessários para a execução do chute omnidirecional [62].

Após a determinação dos parâmetros é executado o algoritmo de execução do movimento chute omnidirecional em função destes parâmetros e dos módulos necessários. Na figura 5.14 está representado o algoritmo e depois é dada uma explicação do mesmo.

Do algoritmo temos alguns parâmetros que são definidos no início do movimento. São eles a 'ballPos' que é a posição da bola, a 'targetPos' que é a posição para onde a bola deve seguir, o 'nrTrajPoints' define em quantos pontos vamos dividir a trajetória e o 't' que é a duração pretendida para o movimento.

```

P0,P1,P2,P3 ← calcularParametrosBezier(ballPos,targetPos)
left ← escolherPe()
for i = 1 → nrTrajPoint do
  trajPoint ← calcularPontoCurvaBezier(P0,P1,P2,P3,1/nrTrajPoints)
  H ← calcularMatrizRotTrans(trajPoint)
  joints ← calcularCinematicaInversa(H,foot)
  joints ← atualizarCOM(joints)
  slots ← adicionarSlot(joints,t)
end for

```

Figura 5.14: Algoritmo de execução do movimento.

Como podemos ver pelo algoritmo, começamos por determinar os parâmetros (P0, P1, P2 e P3) para calcular a curva de Bézier, ou seja, a trajetória que o pé deve seguir. Os cálculos estão representados nas equações 5.41.

$$\left\{ \begin{array}{l} B = ballPos \\ T = targetPos \\ P0 = B - a \frac{\vec{BT}}{|\vec{BT}|} \\ P1 = \frac{P0+B}{2} \\ P2 = B \\ P3 = B + b \frac{\vec{BT}}{|\vec{BT}|} \end{array} \right. \quad (5.41)$$

onde a e b são as distâncias iniciais e finais, respetivamente, da trajetória em relação à posição da bola.

Começa-se pela determinação do pé que vai chutar e depois o cálculo dos parâmetros da curva de Bézier. A escolha do pé segue duas condições primeiro verifica se algum pé está fora do alcance da bola (caso esteja o pé escolhido é o que está no alcance), segundo, caso seja possível chutar com os dois pés é escolhido o pé que fornece um menor ângulo entre a diferença da orientação do pé para a bola e a orientação da bola para o ponto pretendido.

Depois do cálculo dos parâmetros da curva de Bézier passamos para o cálculo do movimento. O movimento é dado pela execução sequencial dos pontos da trajetória num determinado tempo e para a sua determinação é necessário determinar o valor das juntas para cada um destes pontos. O cálculo do movimento está representado dentro do ciclo for no algoritmo.

Para cada ponto da trajetória começa-se por determinar a sua localização espacial recorrendo ao módulo de planeamento de trajetória (`calcularPontoCurvaBezier(P0,P1,P2,P3,1/nrTrajPoints)`).

De seguida calcula-se a matriz de transformação homogénea (H) desse ponto em relação ao robô. Passa-se para o cálculo dos ângulos das articulações usando o módulo de cinemática inversa (`calcularCinematicaInversa(H, foot)`) e o equilíbrio do robô atualizando essas articulações e recorrendo ao módulo de estabilidade (`atualizarCOM(joints)`). No final guarda-se o valor das articulações de cada um dos pontos da trajetória numa lista (`adicionarSlot(joints, t)`).

Para executar o movimento recorre-se a uma adaptação de um comportamento criado pela equipa FC Portugal, chamado `SlotBehavior`, o qual modifica o valor das articulações, num dado espaço de tempo, usando um elemento⁹ da lista criada previamente.

⁹A lista contém vários elementos e cada elemento contém o valor de uma ou mais articulações.

Capítulo 6

Experiências e Resultados

Neste capítulo iremos efetuar experiências de forma a verificar analiticamente o sucesso do trabalho. Os resultados obtidos nessas experiências serão analisados e, caso a experiência não tenha os resultados esperados, tentar explicar o porquê desses dados obtidos.

Começaremos por fazer testes aos módulos de Cinemática Inversa e Planeamento de Trajetória. Estes testes são de grande importância visto o comportamento depender fortemente destes módulos para o sucesso de execução do movimento pretendido. Depois passaremos para os testes ao comportamento onde são efetuados diversos testes para verificar se o comportamento consegue executar um chute em várias direções e para várias posições da bola.

Para facilitar a execução das experiências bem como a aquisição dos resultados foi criado um programa que efetua o procedimento automaticamente. O objetivo é executar n vezes cada uma das experiências e guardar os dados das mesmas numa forma totalmente automática. Os pontos a ter em conta na criação deste programa foram:

- Gravar num ficheiro os parâmetros do movimento em questão.
- Executar um script com instruções de arranque do simulador e do agente.
- Enviar para o simulador o comando de início de partida para que possa ser executado o movimento.
- Executar um script com instruções para terminar o simulador e o agente.
- Executar n vezes para cada movimento, ou seja, para cada direção.

Por outro lado o agente, aquando da sua execução, terá que ler os parâmetros do ficheiro criado pelo programa de testes e guardar num ficheiro diferente para cada teste os dados pertinentes para a experiência. Na figura 6.1 podemos ver o algoritmo do programa de testes.

```

for  $i = 1 \rightarrow cadaDirecao$  do
  guardarParametrosFicheiro(nomeFicheiro, parametros)
  for  $j = 1 \rightarrow n$  do
    iniciarSimuladorAgente()
    enviarComandoInicioJogo()
    finalizarSimuladorAgente()
  end for
end for

```

Figura 6.1: Algoritmo de execução do programa de testes.

6.1 Cinemática Inversa

Os testes a este módulo têm como objetivo verificar as suas funcionalidades bem como as suas limitações. É necessário que este módulo funcione com o mínimo de erro possível visto estar na base de movimentação das articulações e um valor errado de uma articulação faz com que o pé não efetue a trajetória pretendida que por sua vez faz com que o movimento não seja o pretendido.

As limitações das articulações também originam problemas na trajetória do pé. Se pretendermos que o pé efetue uma trajetória para valores das articulações fora dos limites o movimento não vai ser bem-sucedido.

Começaremos por fazer dois testes de forma a posicionar o pé num dado ponto alcançável pela perna do robô e dentro dos limites das articulações. Depois efetuaremos dois testes para dois pontos inalcançáveis, um porque o alcance da perna não o permite outro porque as limitações das articulações não o permite (em ambos os casos o pé atinge a posição no espaço mais perto do ponto pretendido).

Em todos os testes as posições no espaço estão representadas relativamente à bacia do robô (ver figura 5.7 na secção 5.2), o que significa que a posição $(-0.1, -0.1, -0.1)$ situa-se 0.1 metros em frente da bacia, 0.1 metros à direita da bacia e 0.1 metros abaixo da bacia.

6.1.1 Experiência da Cinemática Inversa - Ponto alcançável 1

Neste teste o objetivo é ver se o pé direito atinge a posição $(0.000, -0.055, -0.100)$. Este ponto situa-se 10 cm abaixo da base da perna o que fará com que o pé efetue um movimento linear para cima até atingir esse ponto (lembrar que a base da perna está situada 5.5 cm em relação à bacia, daí aparecer o valor -0.055 na posição destino).

Na tabela 6.1 temos os dados relativos à experiência e na figura 6.2¹ podemos ver a sequência de pontos que o pé percorre até atingir o ponto final. Por fim, na figura 6.3 podemos ver a posição final do robô na simulação.

¹Percurso de I->x é a fase de preparação na qual o robô levanta o pé e o percurso de x->F é a trajetória que o pé percorre na execução do movimento. Ao longo das experiências é usada esta notação sempre que necessário.

Tabela 6.1: Dados da experiência.

	Posição (m)
Bacia	(-0.208, 0.513, 0.262)
Ponto pretendido	(0.000, -0.055, -0.100)
Ponto final	(0.000, -0.054, -0.097)

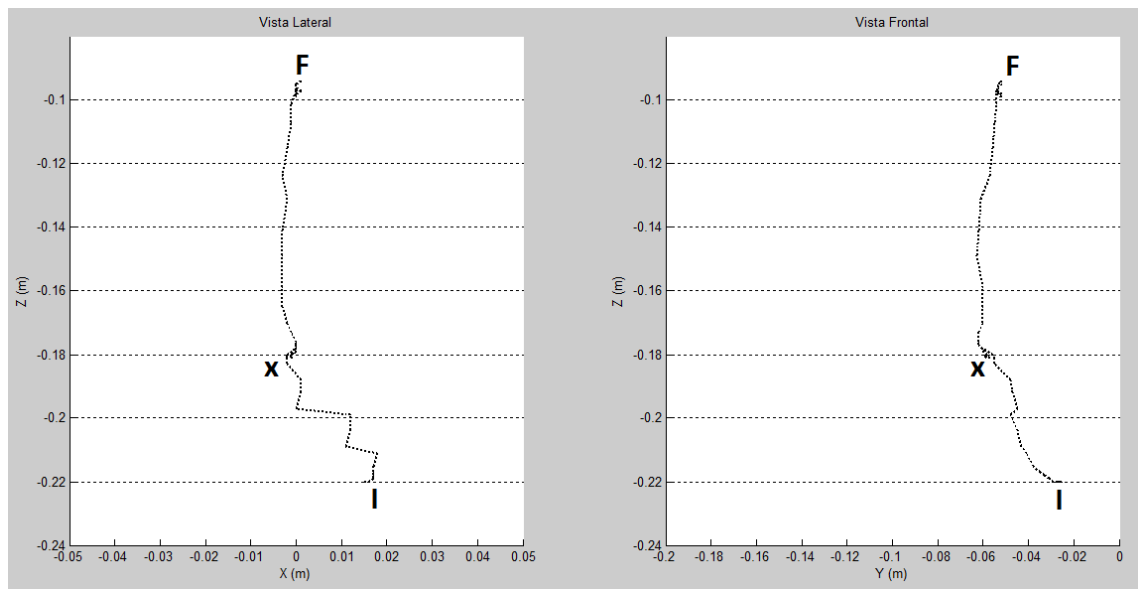


Figura 6.2: Sequência de pontos percorridos pelo pé direito até atingir a posição final.



Figura 6.3: Posição do robô simulado com o pé direito na posição final.

6.1.2 Experiência da Cinemática Inversa - Ponto alcançável 2

Neste teste o objetivo é ver se o pé direito atinge a posição $(0.100, -0.105, -0.100)$. Este ponto situa-se 10 cm abaixo, 10 cm à frente e 5 cm ao lado direito da base da perna o que fará com que o pé fique numa posição diagonal, em relação à bacia, no final do movimento (lembrar que a base da perna está situada 5.5 cm em relação à bacia, daí aparecer o valor -0.105 na posição destino).

Na tabela 6.2 temos os dados relativos à experiência e na figura 6.4 podemos ver a sequência de pontos que o pé percorre até atingir o ponto final. Por fim, na figura 6.5 podemos ver a posição final do robô na simulação.

Tabela 6.2: Dados da experiência.

	Posição (m)
Bacia	$(-0.171, 0.520, 0.266)$
Ponto pretendido	$(0.100, -0.105, -0.100)$
Ponto final	$(0.098, -0.103, -0.105)$

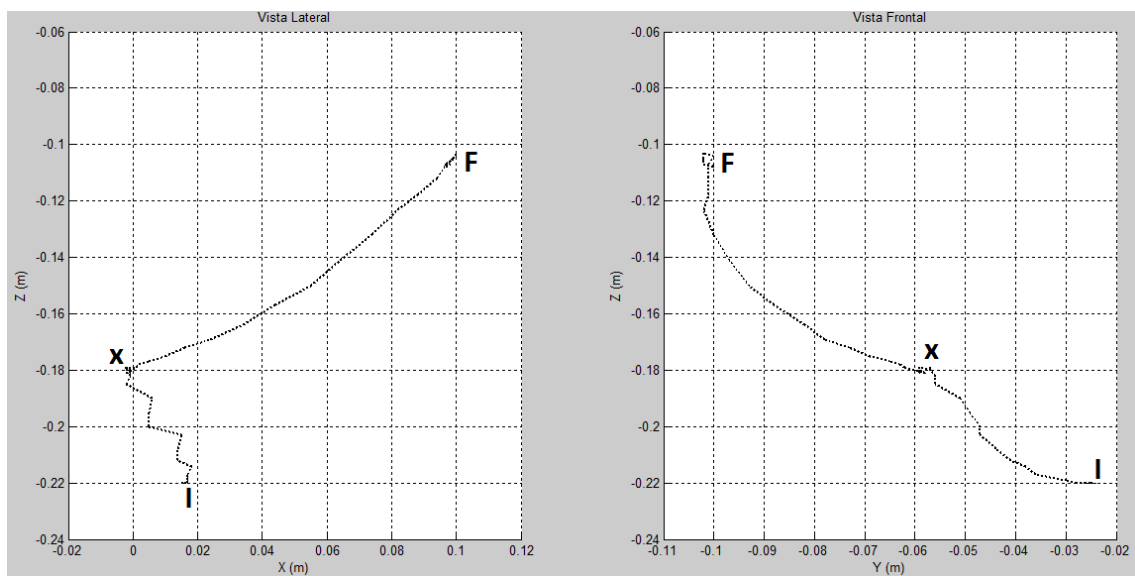


Figura 6.4: Sequência de pontos percorridos pelo pé direito até atingir a posição final.

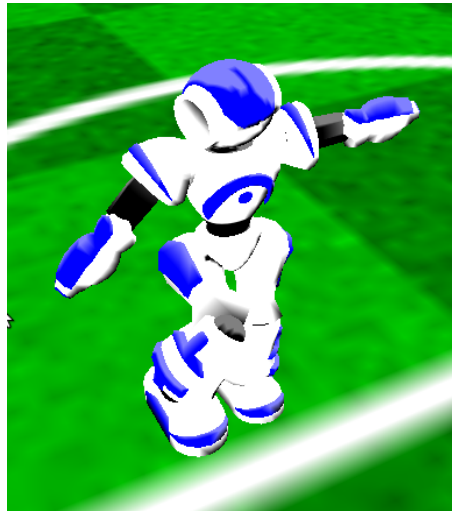


Figura 6.5: Posição do robô simulado com o pé direito na posição final.

6.1.3 Experiência da Cinemática Inversa - Ponto fora do volume de trabalho

Neste teste o objetivo é ver se o pé direito atinge a posição $(0.500, -0.000, -0.000)$. Este ponto situa-se 50 cm à frente da bacia e, como a perna só tem 22 cm de comprimento, não é possível atingir este ponto apenas movimentando a perna. O ponto alcançado é a posição no espaço mais perto do ponto pretendido.

Na tabela 6.3 temos os dados relativos à experiência e na figura 6.6 podemos ver a sequência de pontos que o pé percorre até atingir o ponto alcançado. Por fim, na figura 6.7 podemos ver a posição final do robô na simulação.

Tabela 6.3: Dados da experiência.

	Posição (m)
Bacia	$(-0.222, 0.494, 0.266)$
Ponto pretendido	$(0.500, -0.000, -0.000)$
Ponto final	$(0.236, -0.010, -0.012)$

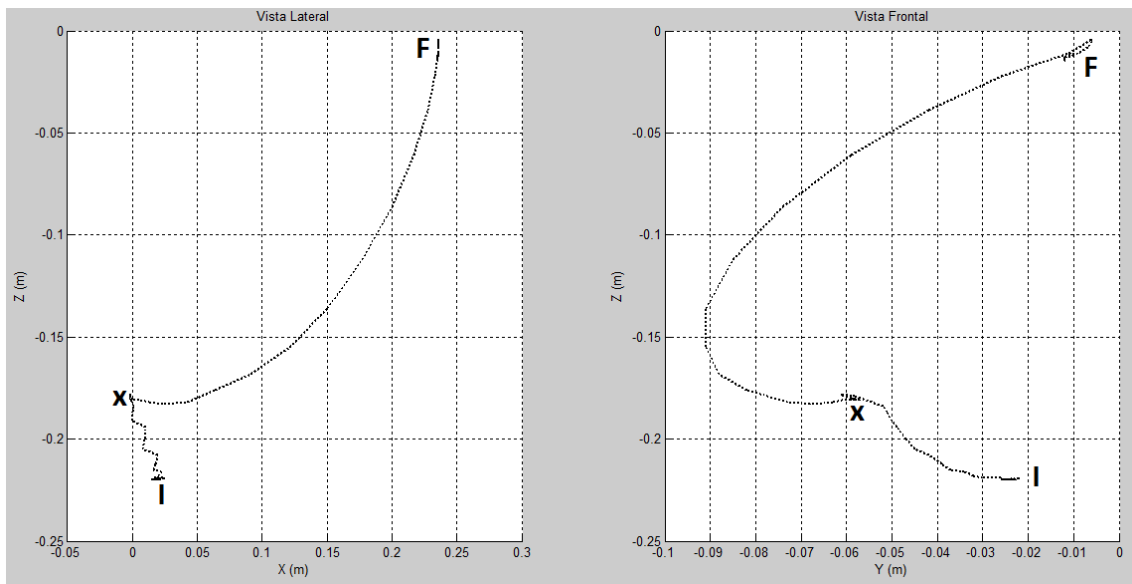


Figura 6.6: Sequência de pontos percorridos pelo pé direito até atingir a posição final.



Figura 6.7: Posição do robô simulado com o pé direito na posição final.

6.1.4 Experiência da Cinemática Inversa - Ponto fora dos limites das articulações

Neste teste o objetivo é ver se o pé direito atinge a posição $(-0.200, -0.055, -0.000)$. Este ponto situa-se 20 cm atrás da base da perna e é um ponto que não pode ser alcançado devido às limitações das articulações ao nível da bacia. O ponto alcançado é a posição no espaço mais perto do ponto pretendido.

Na tabela 6.4 temos os dados relativos à experiência e na figura 6.8 podemos ver a sequência de pontos que o pé percorre até atingir o ponto alcançado. Por fim, na figura 6.9 podemos ver a posição final do robô na simulação.

Tabela 6.4: Dados da experiência.

	Posição (m)
Bacia	(-0.223, 0.487, 0.265)
Ponto pretendido	(-0.200, -0.055, -0.000)
Ponto final	(-0.142, -0.127, -0.085)

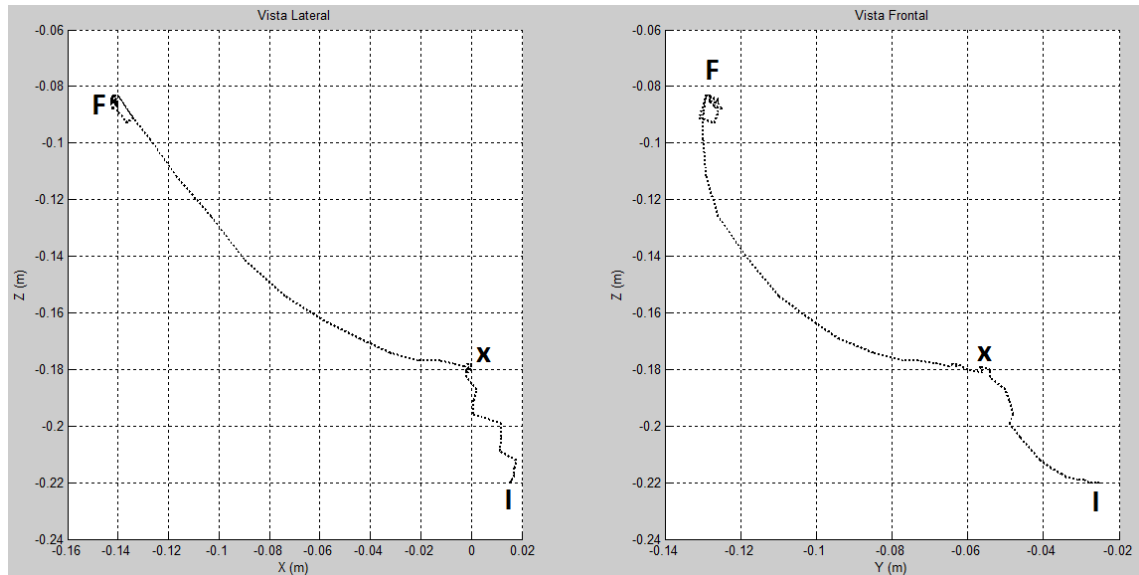


Figura 6.8: Sequência de pontos percorridos pelo pé direito até atingir a posição final.



Figura 6.9: Posição do robô simulado com o pé direito na posição final.

6.1.5 Análise dos Resultados

Como podemos ver pelos resultados das experiências este módulo funciona com uma precisão muito boa. Se não fosse as limitações das articulações seria possível chegar a qualquer ponto

dentro do volume de trabalho.

Um dos grandes problemas deste módulo é a articulação com referencial rodado 45° graus ao nível da base da perna. Este inconveniente faz com que a chegada ao ponto pretendido não seja sempre muito linear.

6.2 Planeamento de Trajetória

Os testes a este módulo têm como objetivo verificar as suas funcionalidades bem como as suas limitações. A importância deste módulo no deslocamento que pretendemos dar à bola prende-se com o facto de ser responsável pela criação de uma trajetória, a ser executada pelo pé que vai chutar, de forma a atingir a bola numa determinada posição e num determinado ângulo. É este ponto de impacto que vai fazer com que a bola se desloque na direção pretendida.

Os testes a realizar para este módulo são a execução de um movimento linear e dois movimentos circulares em planos diferentes. Os dados da experiência são os 4 pontos necessários para o cálculo da curva de Bézier e o número de pontos que a trajetória será dividida (o pé vai dirigir-se para estes pontos sequencialmente).

Em todos os testes as posições no espaço estão representadas relativamente à bacia do robô (ver figura 5.7 na secção 5.2), o que significa que a posição $(-0.1, -0.1, -0.1)$ situa-se 0.1 metros em frente da bacia, 0.1 metros à direita da bacia e 0.1 metros abaixo da bacia.

6.2.1 Experiência Planeamento de Trajetória - Movimento linear

Neste teste o objetivo é verificar o progresso do pé direito ao executar um movimento linear. Este movimento é dado pelos parâmetros P0, P1, P2 e P3 e, para esta experiência, definem um movimento linear segundo o eixo do x ($z=-15$ cm e $y=-5.5$ cm). A posição do pé vai variar entre $-0.080 \leq x \leq 0.170$ cm.

Na tabela 6.5 temos os dados relativos à experiência e na figura 6.10 podemos ver a sequência de pontos que o pé percorre ao efetuar o trajeto calculado. Por fim, na figura 6.11 podemos ver uma sequência de fotogramas com o movimento realizado pelo robô na simulação.

Tabela 6.5: Movimento linear - Dados da experiência.

	Posição (m)
Bacia	$(-0.223, 0.487, 0.265)$
P0	$(-0.080, -0.055, -0.150)$
P1	$(0.000, -0.055, -0.150)$
P2	$(0.080, -0.055, -0.150)$
P3	$(0.170, -0.055, -0.150)$
Nº de pontos	10

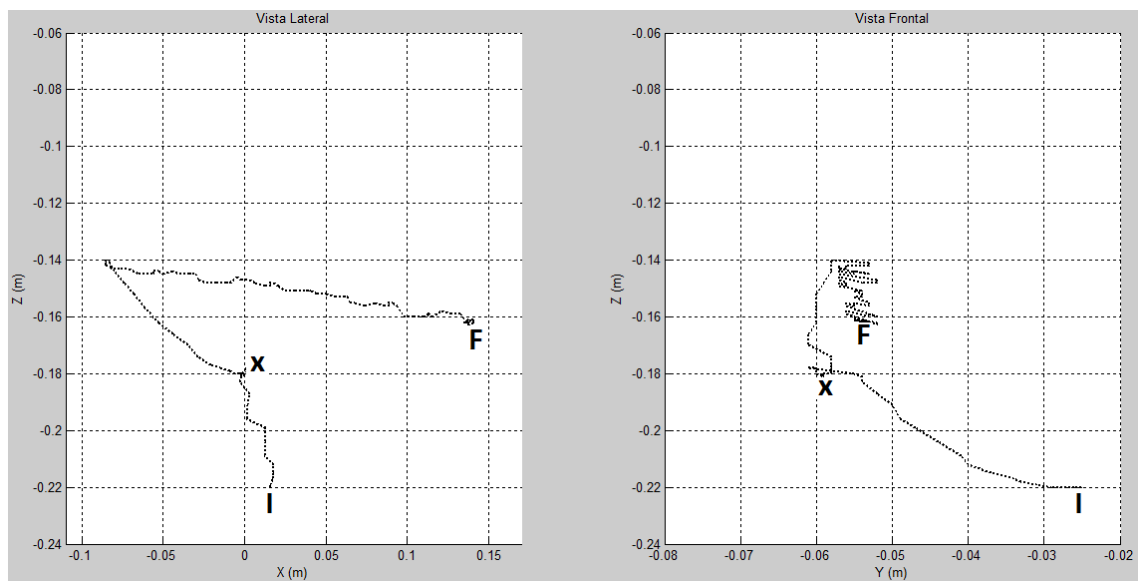


Figura 6.10: Sequência de pontos percorridos pelo pé direito durante a execução do movimento linear.

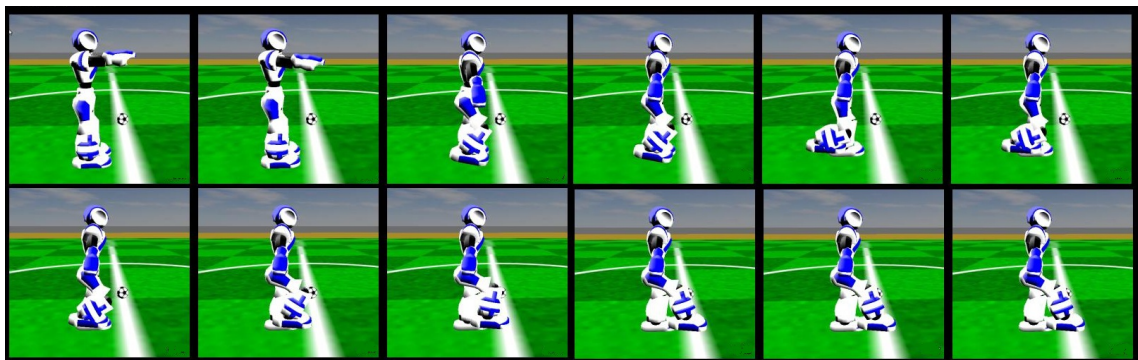


Figura 6.11: Movimento linear - Sequência de fotografias.

6.2.2 Experiência Planeamento de Trajetória - Movimento circular para cima

Neste teste o objetivo é verificar o progresso do pé direito ao executar um movimento circular. Este movimento é dado pelos parâmetros P0, P1, P2 e P3 e, para esta experiência, definem um movimento circular no plano XZ ($y = -5.5$ cm constante). A posição do pé vai variar entre $-0.080 \leq x \leq 0.170$ cm e $-0.180 \leq z \leq 0.0$ cm.

Na tabela 6.6 temos os dados relativos à experiência e na figura 6.12 podemos ver a sequência de pontos que o pé percorre ao efetuar o trajeto calculado. Por fim, na figura 6.13 podemos ver uma sequência de fotografias com o movimento realizado pelo robô na simulação.

Tabela 6.6: Movimento circular no plano XZ - Dados da experiência.

	Posição (m)
Bacia	(-1.193, 3.125, 0.278)
P0	(-0.080, -0.055, -0.180)
P1	(0.000, -0.055, -0.000)
P2	(0.080, -0.055, -0.000)
P3	(0.170, -0.055, -0.180)
Nº de pontos	10

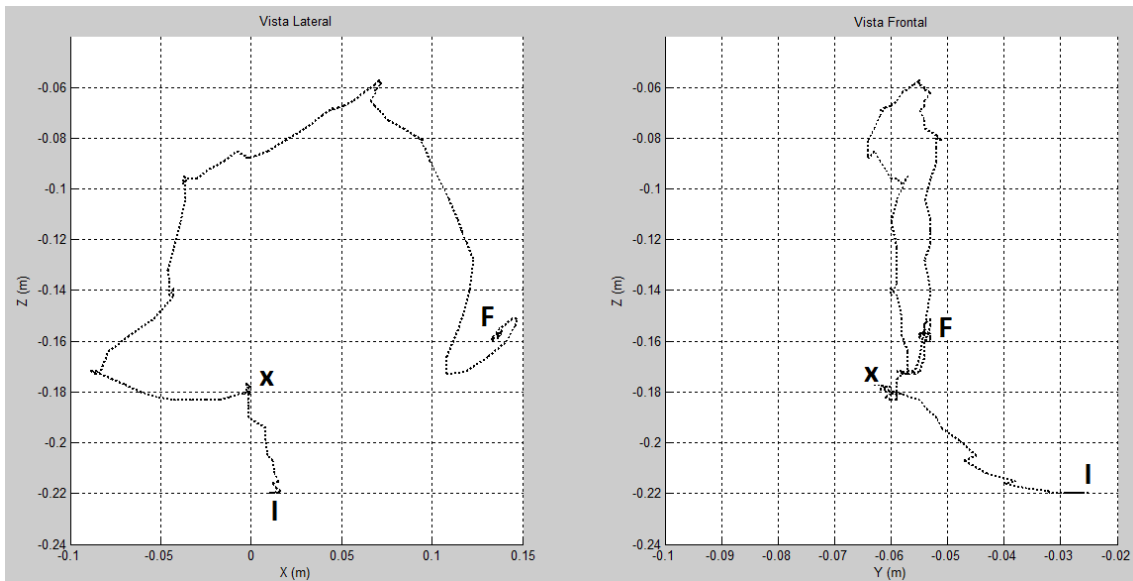


Figura 6.12: Sequência de pontos percorridos pelo pé direito durante a execução do movimento circular no plano XZ.

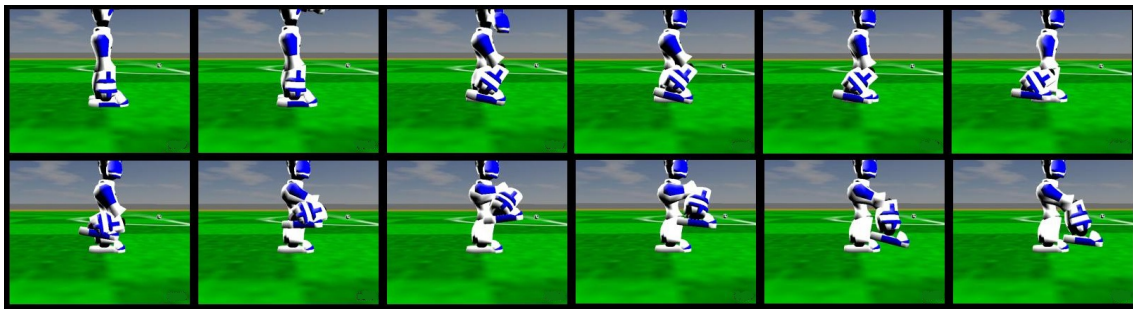


Figura 6.13: Movimento circular no plano XZ - Sequência de fotografias.

6.2.3 Experiência Planejamento de Trajetória - Movimento circular para o lado

Neste teste o objetivo é verificar o progresso do pé direito ao executar um movimento circular. Este movimento é dado pelos parâmetros P0, P1, P2 e P3 e, para esta experiência, definem um movimento circular no plano XY ($z = -0.18$ cm constante). A posição do pé vai variar entre $-0.080 \leq x \leq 0.170$ cm e $-0.050 \leq y \leq -0.3$ cm.

Na tabela 6.7 temos os dados relativos à experiência e na figura 6.14 podemos ver a sequência de pontos que o pé percorre ao efetuar o trajeto calculado. Por fim, na figura 6.15 podemos ver uma sequência de fotogramas com o movimento realizado pelo robô na simulação.

Tabela 6.7: Movimento circular no plano XY - Dados da experiência.

	Posição (m)
Bacia	(-0.175, 5.132, 0.2018)
P0	(-0.080, -0.050, -0.180)
P1	(0.000, -0.300, -0.180)
P2	(0.080, -0.300, -0.180)
P3	(0.170, -0.050, -0.180)
Nº de pontos	10

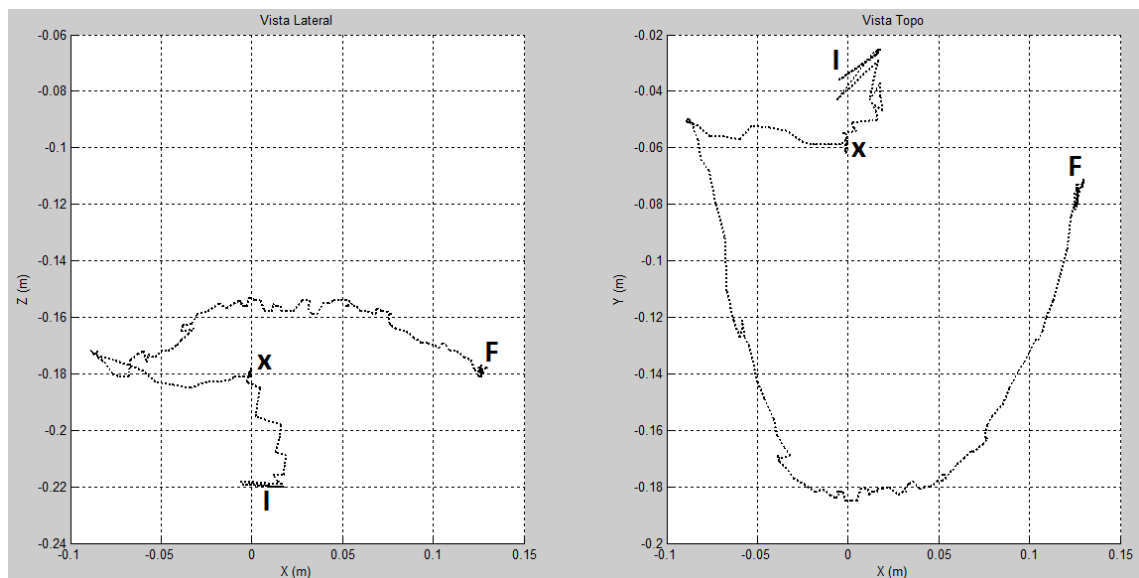


Figura 6.14: Sequência de pontos percorridos pelo pé direito durante a execução do movimento circular no plano XY.

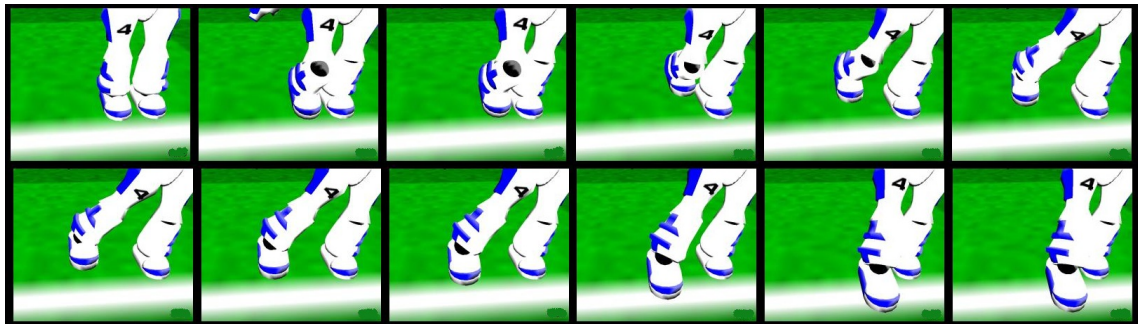


Figura 6.15: Movimento circular no plano XY - Sequência de fotogramas.

6.2.4 Análise dos Resultados

Os cálculos realizados neste módulo são de natureza simples. Usando os quatro parâmetros, P0, P1, P2 e P3, é criada uma curva na qual o movimento vai incidir.

As experiências realizadas fazem forte uso do módulo da cinemática inversa para percorrer os pontos da curva e como este módulo é bastante preciso, a trajetória é executada sem grandes desvios. Como podemos ver pelas figuras 6.10, 6.12 e 6.14 esses desvios são na casa dos mm.

Na figura 6.12 vemos que o movimento circular não é executado de forma totalmente circular. Este problema prende-se com, mais uma vez, a limitação da articulação situada na base da perna a qual só permite que a perna rode -25° no plano YZ, ou seja, só consegue rodar 25° para trás.

6.3 Chuto Omnidirecional

Nesta secção procederemos aos testes ao comportamento na sua totalidade. Iremos testar a execução do movimento para várias posições da bola em relação ao robô e para cada uma dessas posições tentar deslocar a bola em várias direções, nomeadamente a -90° , -45° , 0° , 45° e 90° em relação à orientação do robô. Na figura 6.16 podemos ver as localizações da bola usadas.



(a) Em frente aos pés.

(b) Em frente ao pé direito.

(c) Diagonal com o pé direito.

Figura 6.16: Posições da bola, em relação ao robô, usadas nos testes.

Para que o comportamento consiga direcionar a bola em todas as direções e para várias posições iniciais da mesma é necessário que este consiga determinar os parâmetros apropriados para cada caso. Nestes testes esses parâmetros foram configurados manualmente para que se consiga direcionar a bola nas direções de teste e que esta consiga alcançar cerca de 1 metro. Os parâmetros necessário são (ver figura 6.17):

- Rpos - localização no plano XY do robô em relação ao referencial do campo;
- Rori - orientação do robô em relação ao referencial do campo;
- T - localização no plano XY do ponto pretendido;
- a - distância entre o ponto inicial do movimento e a bola;
- b - distância entre o ponto final do movimento e a bola;

- $hP0$ - Altura do ponto P0 para o cálculo da curva de Bézier;
- $hP1$ - Altura do ponto P1 para o cálculo da curva de Bézier;
- $hP2$ - Altura do ponto P2 para o cálculo da curva de Bézier;
- $hP3$ - Altura do ponto P3 para o cálculo da curva de Bézier;
- $nrPts$ - número de pontos para dividir a trajetória;
- $tMov$ - duração do movimento;
- $footOffset$ - distância entre a articulação do pé e o ponto de impacto do pé na bola. Necessário quando queremos chutar com a ponta do pé, o calcanhar ou o meio da parte lateral.
- $footOrient$ - Se o movimento vai ser executado orientando o pé na direção do ponto pretendido ou não (p. ex.: verdadeiro quando executamos um chuto frontal, falso quando é executado um chuto lateral com a parte lateral do pé).

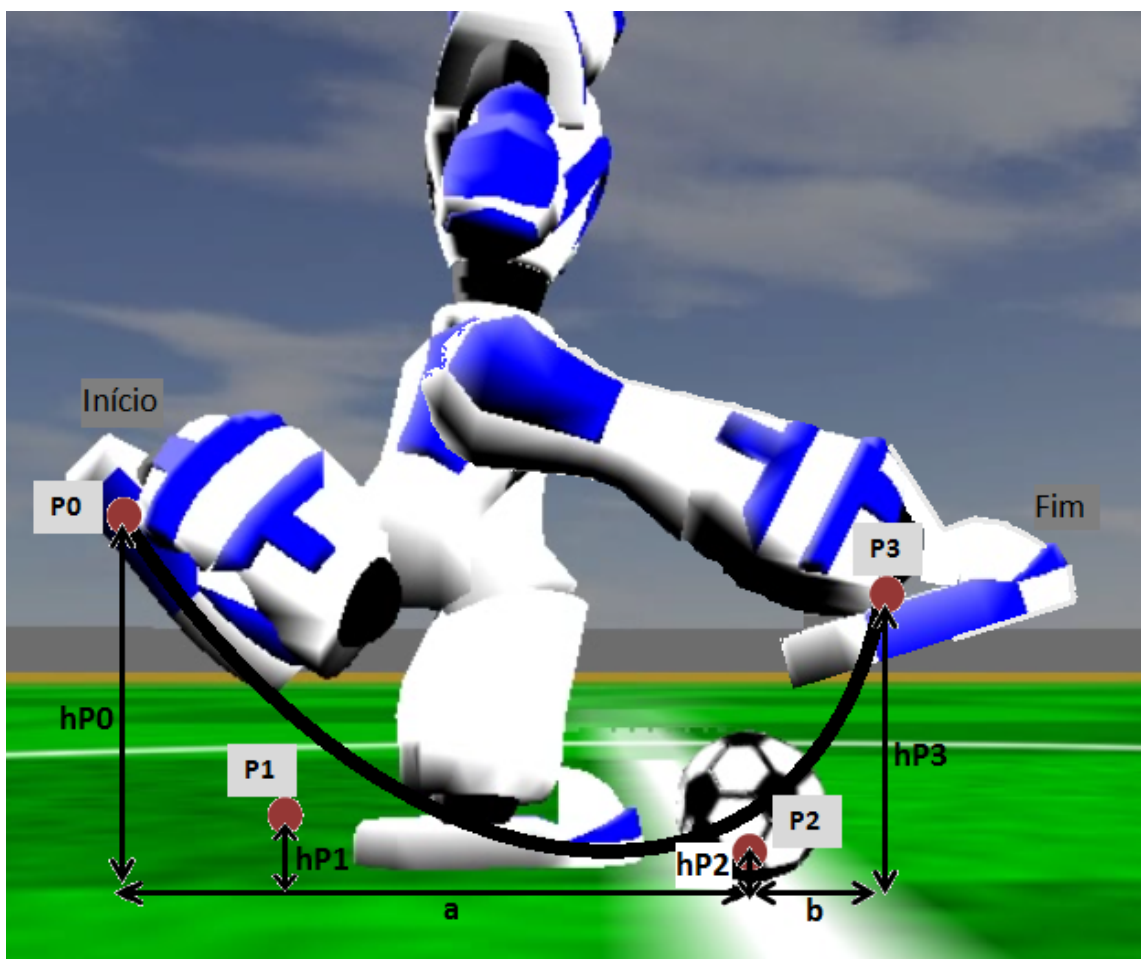


Figura 6.17: Parte dos parâmetros necessários na execução do movimento.

Começamos com os testes com a bola na posição entre os pés, depois passaremos para os testes com a bola em frente ao pé direito e por fim testamos o comportamento com a bola à frente e ao lado direito do pé direito (ver figura 6.16). Poderíamos testar também com a bola localizada em relação ao pé esquerdo mas como o movimento é simétrico os resultados seriam os mesmos.

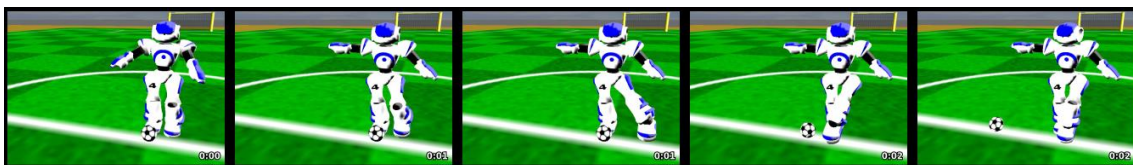
Na figura 6.18 podemos ver uma sequência de fotogramas de alguns chutes realizados nestes testes. Não achei necessário incluir a sequência exaustiva de todos os movimentos realizados nos próximos testes visto grande parte deles serem quase cópias destes movimentos, alterando apenas a posição inicial da bola. Podemos também ver pela figura que o movimento é sempre executado com o pé que melhor poderá executar o movimento.



(a) Chuto frontal.



(b) Chuto lateral interior.



(c) Chuto lateral exterior.



(d) Chuto diagonal interior.



(e) Chuto diagonal exterior.

Figura 6.18: Exemplos de chutes em várias direções na simulação.

6.3.1 Experiência Chuto Omnidirecional - bola em frente aos pés

Nesta experiência o objetivo é verificar se o comportamento consegue direcionar a bola, localizada em frente aos dois pés, nas direções de -90° , -45° , 0° , 45° e 90° . Começou-se por configurar os parâmetros para cada uma das direções e após essa configuração passou-se às experiências em si.

Para cada direção é efetuado o movimento 10 vezes das quais se tiram 10 amostras da posição da bola final. É calculada a média das amostras e no fim calculada a direção média do movimento. Na tabela 6.8 é apresentado os parâmetros usados nas várias direções e na figura 6.19 os resultados obtidos. Na figura 6.20 estão representados todos os pontos atingidos pela bola em cada uma das direções.

Tabela 6.8: Chuto omnidirecional - Parâmetros da experiência com a bola à frente dos dois pés.

	-90°	-45°	0°	45°	90°
Rpos	(-0.19, 0.0)	(-0.17, 0.0)	(-0.17, 0.0)	(-0.17, 0.0)	(-0.19, 0.0)
Rori	0	0	0	0	0
T	(0.0, -1.0)	(1.0, -1.0)	(1.0, 0.0)	(1.0, 1.0)	(0.0, 1.0)
a	0.35	0.2	0.2	0.2	0.35
b	-0.05	0.0	0.4	0.0	-0.05
hP0	0.18	0.18	0.18	0.18	0.18
hP1	0.0	0.02	0.03	0.02	0.0
hP2	0.0	0.02	0.02	0.02	0.0
hP3	0.04	0.04	0.02	0.04	0.04
nrPts	4	8	8	8	4
tMov	0.85	1.05	0.85	1.05	0.85
footOffset	0.04	0.052	0.0	0.052	0.04

	-90°		-45°		0°		45°		90°	
	X (m)	Y (m)	X (m)	Y (m)	X (m)	Y (m)	X (m)	Y (m)	X (m)	Y (m)
Amostra 1	0,007416	-1,06482	1,12929	-1,065	1,015820	0,089043	1,01268	1,1659	0,007717	1,00109
Amostra 2	0,0145	-0,98957	1,22588	-0,93399	1,099890	-0,035131	0,954328	1,13532	0,022954	1,05422
Amostra 3	0,020627	-1,01164	1,07022	-1,11925	1,134680	0,153423	0,931561	0,921923	0,008023	1,08028
Amostra 4	0,02316	-1,0588	1,15707	-1,01849	1,096480	-0,133284	1,03658	0,80097	0,0014	0,941486
Amostra 5	0,014625	-1,00806	1,19109	-1,00816	1,089050	-0,148491	1,26479	1,03271	0,003843	1,0498
Amostra 6	0,044961	-1,00851	1,056609	-1,17818	1,072670	-0,158199	0,840667	1,13107	0,027549	1,00915
Amostra 7	0,01359	-1,10161	1,17687	-1,0468	1,104290	-0,110319	1,25108	0,811474	0,027482	1,01747
Amostra 8	0,010653	-1,05921	1,09849	-1,13168	0,950086	-0,195576	1,24501	0,888301	0,026955	1,00216
Amostra 9	-0,00167	-1,02224	1,00398	-0,8073	1,082720	0,135581	1,09939	1,00886	-0,00969	0,903212
Amostra 10	0,014056	-1,0102	1,09846	-1,06501	1,099980	0,133573	0,944494	1,06948	0,006104	1,05857
Média	0,016192	-1,03347	1,120796	-1,03739	1,074567	-0,02694	1,058058	0,996601	0,012233	1,011744
Posição inicial (m)	0	0	0	0	0	0	0	0	0	0
Posição final (m)	0,016192	-1,03347	1,120796	-1,03739	1,074567	-0,02694	1,058058	0,996601	0,012233	1,011744
Ângulo (°)	-89,1		-42,8		-1,4		43,3		89,3	

Figura 6.19: Resultados da experiência com a bola situada à frente dos dois pés.

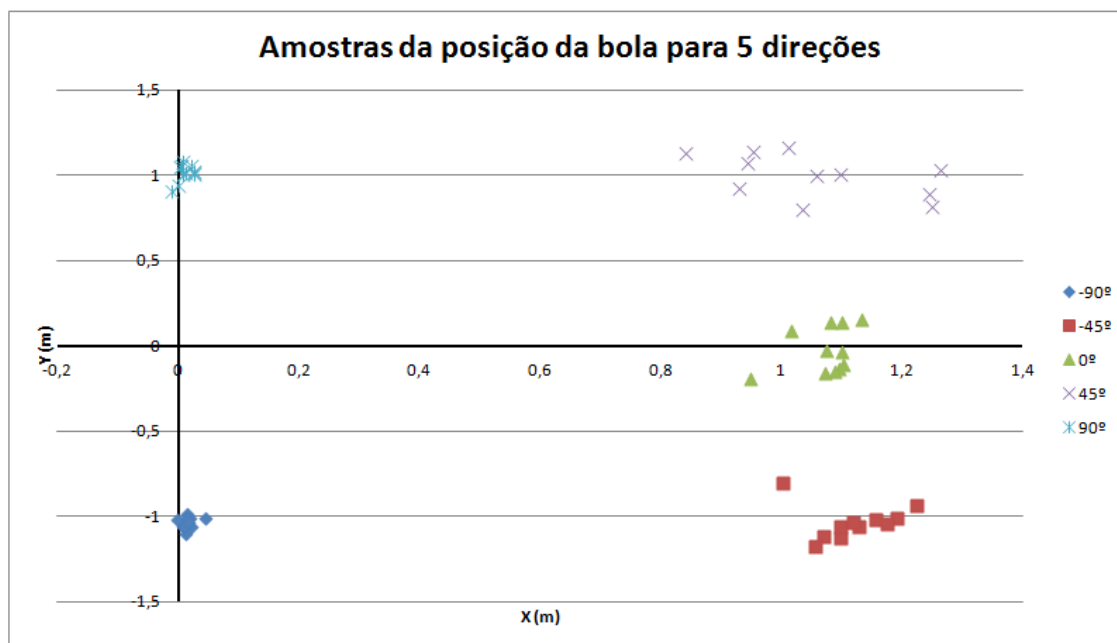


Figura 6.20: Dispersão dos pontos finais da bola da experiência com a bola situada à frente dos dois pés.

6.3.2 Experiência Chuto Omnidirecional - bola em frente ao pé direito

Nesta experiência o objetivo é verificar se o comportamento consegue direcionar a bola, localizada em frente ao pé direito, nas direções de -90° , -45° , 0° , 45° e 90° . Começou-se por configurar os parâmetros para cada uma das direções e após essa configuração passou-se às experiências em si.

Para cada direção é efetuado o movimento 10 vezes das quais se tiram 10 amostras da posição da bola final. É calculada a média das amostras e no fim calculada a direção média do movimento. Na tabela 6.9 é apresentado os parâmetros usados nas várias direções e na figura 6.21 os resultados obtidos. Na figura 6.22 estão representados todos os pontos atingidos pela bola em cada uma das direções.

Tabela 6.9: Chuto omnidirecional - Parâmetros da experiência com a bola à frente do pé direito.

	-90°	-45°	0°	45°	90°
Rpos	(-0.17, -0.055)	(-0.17, -0.055)	(-0.17, -0.055)	(-0.17, -0.055)	(-0.17, -0.055)
Rori	0	0	0	0	0
T	(0.0, -1.0)	(1.0, -1.0)	(1.0, 0.0)	(1.0, 1.0)	(0.0, 1.0)
a	0.35	0.2	0.2	0.2	0.3
b	0.05	0.4	0.4	0.2	0.0
hP0	0.18	0.18	0.18	0.18	0.18
hP1	0.0	0.03	0.03	0.03	0.03
hP2	0.0	0.03	0.02	0.03	0.03
hP3	0.02	0.05	0.02	0.05	0.05
nrPts	8	8	8	8	8
tMov	1.05	1.35	0.85	1.35	1.1
footOffset	0.057	0.057	0.0	0.057	0.04

	-90°		-45°		0°		45°		90°	
	X (m)	Y (m)	X (m)	Y (m)	X (m)	Y (m)	X (m)	Y (m)	X (m)	Y (m)
Amostra 1	0,022955	-0,56339	0,742586	-0,75973	3,067110	-0,002347	1,0234	1,06269	-0,00095	0,737315
Amostra 2	0,029634	-0,72635	0,99228	-1,11968	2,998980	-0,122319	1,01321	1,05337	0,007657	0,739625
Amostra 3	0,017023	-0,76975	0,706447	-0,51977	3,035420	-0,088237	1,08619	1,01756	-0,01878	0,770807
Amostra 4	0,029522	-0,61057	1,10538	-1,08259	3,036300	0,040964	1,11469	1,01017	0,000586	0,736171
Amostra 5	0,008735	-0,76367	1,06673	-0,98838	3,144410	0,040999	1,11024	1,05024	0,003837	0,694625
Amostra 6	0,020861	-0,7555	0,972679	-0,87515	3,064340	0,064553	1,08548	0,993381	-0,0119	0,66261
Amostra 7	0,01303	-0,7623	0,724823	-0,60925	3,114710	-0,245775	1,14354	0,890183	0,012468	0,624943
Amostra 8	-0,00166	-0,7821	0,73155	-0,60494	2,981910	-0,057068	0,935292	1,12008	-0,00173	0,609869
Amostra 9	0,013263	-0,77223	1,07047	-0,877	3,036330	-0,149887	1,07476	1,00008	0,0024	0,682313
Amostra 10	0,007757	-0,76969	1,02859	-1,13892	3,084710	-0,118237	0,940064	1,14384	-0,01262	1,41412
Média	0,016112	-0,72756	0,914154	-0,85754	3,056422	-0,06374	1,052687	1,034159	-0,0019	0,76724
Posição inicial (m)	0	0	0	0	0	0	0	0	0	0
Posição final (m)	0,016112	-0,72756	0,914154	-0,85754	3,056422	-0,06374	1,052687	1,034159	-0,0019	0,76724
Ângulo (°)	-88,7		-43,2		-1,2		44,5		90,1	

Figura 6.21: Resultados da experiência com a bola situada à frente do pé direito.

6.3.3 Experiência Chuto Omnidirecional - bola em frente e ao lado do pé direito

Nesta experiência o objetivo é verificar se o comportamento consegue direcionar a bola, localizada em frente em frente e ao lado do pé direito, nas direções de 0°, 45° e 90°. Começou-se por configurar os parâmetros para cada uma das direções e após essa configuração passou-se às experiências em si.

Para cada direção é efetuado o movimento 10 vezes das quais se tiram 10 amostras da posição da bola final. É calculada a média das amostras e no fim calculada a direção média do movimento. Na tabela 6.10 é apresentado os parâmetros usados nas várias direções e na figura 6.23 os resultados obtidos. Na figura 6.22 estão representados todos os pontos atingidos pela bola em cada uma da direções.

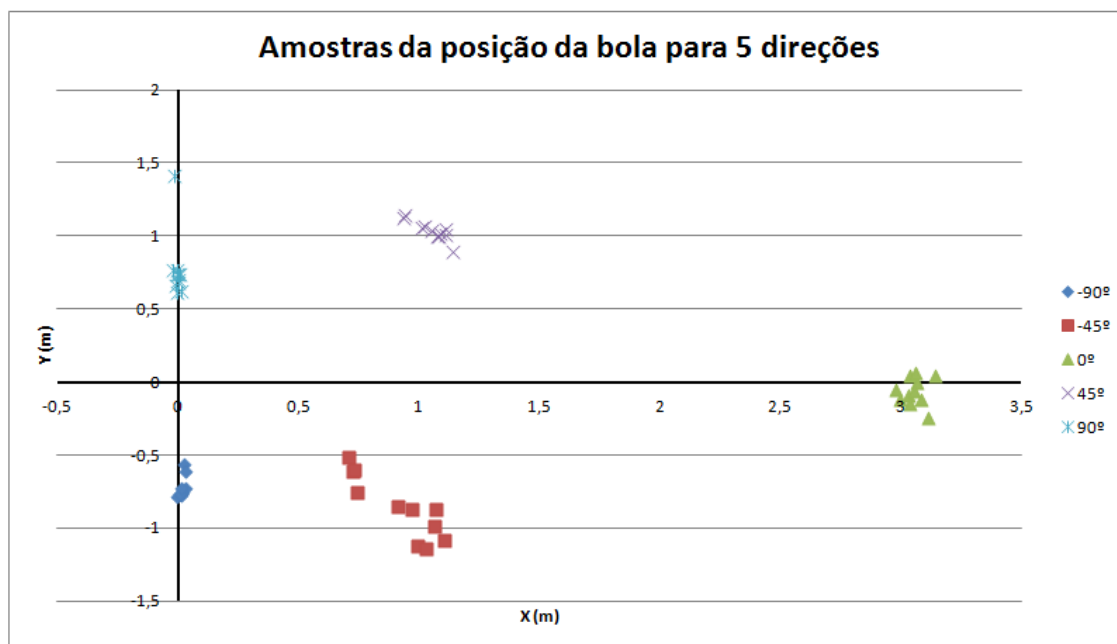


Figura 6.22: Dispersão da posição final da bola da experiência com a bola situada à frente do pé direito.

Tabela 6.10: Chuto omnidirecional - Parâmetros da experiência com a bola à frente e ao lado do pé direito.

	0°	45°	90°
Rpos	(-0.17, -0.15)	(-0.17, -0.15)	(-0.17, -0.15)
Rori	0	0	0
T	(1.0, 0.0)	(1.0, 1.0)	(0.0, 1.0)
a	0.2	0.15	0.1
b	0.2	0.2	0.2
hP0	0.18	0.04	0.07
hP1	0.03	0.03	0.0
hP2	0.02	0.03	0.02
hP3	0.02	0.06	0.02
nrPts	8	6	4
tMov	0.95	1.05	0.85
footOffset	0.0	0.0	0.03

6.3.4 Análise dos Resultados

Pelos resultados obtidos nestas experiências podemos ver que o comportamento é capaz de executar o movimento e direcionar a bola em várias direções. Através da análise das tabelas conseguimos ver que, para o número de amostras, a direção com que a bola se desloca é bastante precisa em relação à direção pretendida e através da análise dos gráficos podemos ver o agrupamento de pontos localizados relativamente perto uns dos outros, o que significa que a posição final da bola não se dispersa com o número de repetições.

	0°		45°		90°	
	X (m)	Y (m)	X (m)	Y (m)	X (m)	Y (m)
Amostra 1	1,592730	-0,218129	0,888489	0,860486	0,126963	0,892148
Amostra 2	1,630530	-0,201397	0,907938	0,913811	0,006861	0,440672
Amostra 3	1,685780	-0,204562	0,877807	0,877299	0,027603	0,654041
Amostra 4	1,678050	-0,206297	0,88706	0,880553	-0,05684	0,77206
Amostra 5	1,582540	-0,246653	0,864034	0,847095	0,00038	0,807083
Amostra 6	1,561960	-0,239541	0,869091	0,887152	0,037382	0,64622
Amostra 7	1,598630	-0,248096	0,882956	0,862786	0,178856	0,766822
Amostra 8	1,651450	-0,221874	0,870164	0,860285	-0,05159	0,618856
Amostra 9	1,668070	-0,229675	0,895509	0,875018	-0,00331	0,851088
Amostra 10	1,699910	-0,166048	0,872237	0,894948	0,076466	1,45873
Média	1,634965	-0,21823	0,881529	0,875943	0,034278	0,790772
Posição inicial (m)	0	0	0	0	0	0
Posição final (m)	1,634965	-0,21823	0,881529	0,875943	0,034278	0,790772
Ângulo (°)	-7,6		44,8		87,5	

Figura 6.23: Resultados da experiência com a bola situada à frente e ao lado do pé direito.

A maior parte dos problemas são possíveis de resolver com a escolha adequada dos parâmetros mas é necessário ter em atenção, aquando do pedido de execução do movimento, para que a bola esteja numa posição alcançável pelo robô, idealmente a distância entre o robô e a bola deveria estar entre 1.5 e 2.0 cm. Se a bola estiver muito perto pode haver colisão precoce entre o pé e a bola, se estiver muito longe o pé pode não conseguir atingir a bola.

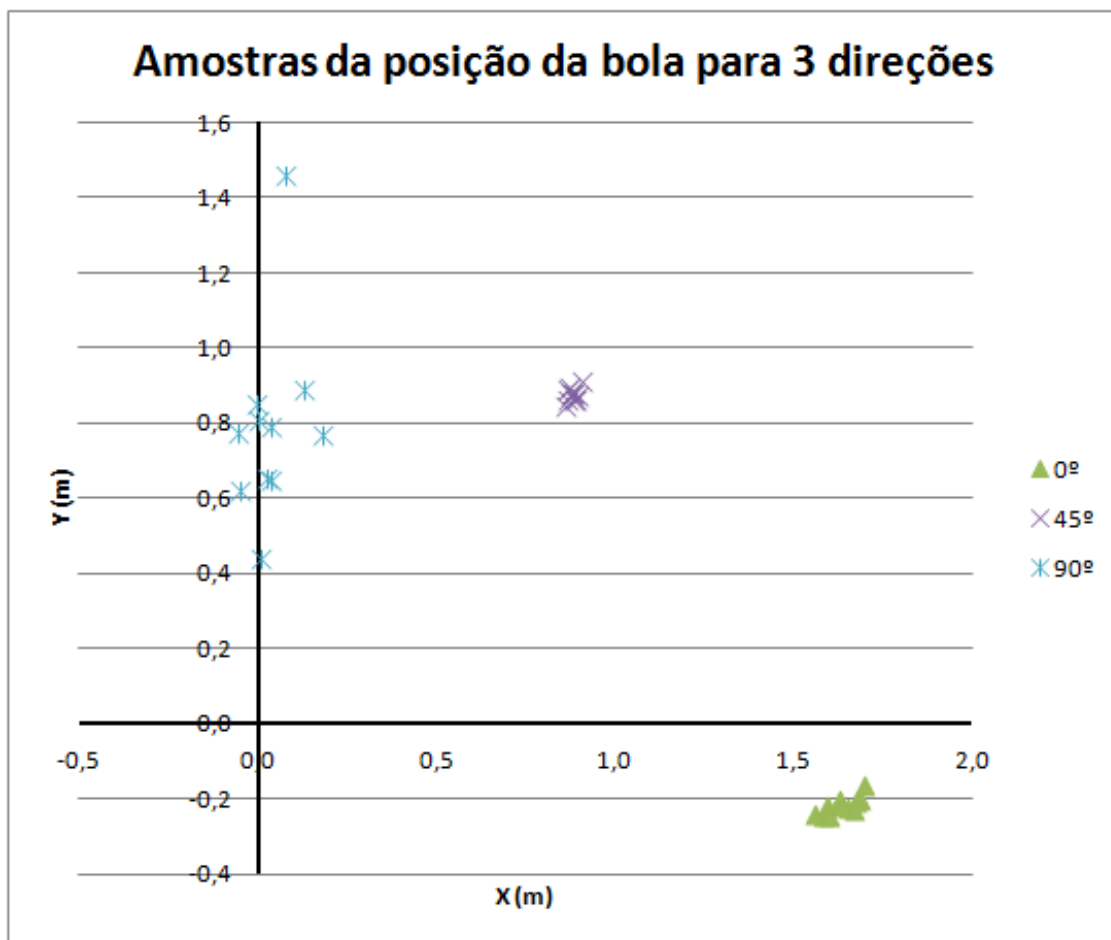


Figura 6.24: Dispersão da posição final da bola da experiência com a bola situada à frente e ao lado do pé direito.

Capítulo 7

Conclusões e Trabalho Futuro

7.1 Conclusões

Neste trabalho desenvolveu-se um comportamento para dotar um robô humanoide com a capacidade de realizar um chute omnidirecional. Este comportamento é totalmente modular (divido em classes) e está apto para ser incorporado na arquitetura de comportamentos da equipa FC Portugal. A nível geral penso que os objetivos foram todos concretizados na medida que se conseguiu desenvolver um comportamento para executar um chute omnidirecional. Pelas experiências provamos que o movimento é bastante preciso e fiável.

Como o comportamento é constituído por diversos módulos foi necessário fazer testes a cada módulo de forma a podermos identificar problemas que possam surgir na execução do movimento final. Como tal procedemos à realização de testes aos módulos individualmente (quando possível) e analisar os dados desses testes.

A análise dos resultados obtidos provou que o comportamento executa com precisão o movimento pretendido. Um grande responsável destes resultados satisfatórios é o módulo de cinemática inversa, é sobre ele que assenta toda a base do movimento.

Alguns problemas apareceram no decorrer do trabalho que condicionaram as funcionalidades do comportamento. Um desses problemas foi o cálculo da estabilidade do robô que tornava o movimento menos preciso e muitas das vezes diferente do pretendido. O cálculo tinha como base a previsão da posição futura e a alteração das articulações de forma a posicionar o pé nesse ponto e era neste cálculo que residia o problema, é que a maior parte das vezes a atualização do posicionamento do robô alterava a posição pretendida para o pé. Devido a este problema, e apesar do módulo ter sido desenvolvido, esta abordagem foi descartada e adotada uma adaptação de um movimento já criado pela equipa na qual o robô se equilibra num dado pé.

Outra falha encontrada foi ao nível do simulador que não verifica as colisões entre partes do modelo. Evidenciou-se isso na execução de um chute frontal com a bola à frente dos dois pés, o pé que ia chutar a bola sobrepunha-se ao outro pé sem qualquer restrição. No gráfico da figura [6.20](#) verificamos que a execução do movimento na direção de 0° foi realizado com algum sucesso mas esse resultado deve-se à existência deste bug.

Neste trabalho não foi possível efetuar testes para o posicionamento da bola atrás ou ao lado do robô visto o sistema de localização do robô não conseguir localizar a bola nestas posições.

7.2 Trabalho Futuro

Apesar dos resultados terem sido satisfatórios este trabalho está longe de ser o que se pretende como resultado final. Foi um trabalho criado de base pode ser melhorado em vários pontos. Algumas dessas melhorias poderão ser:

- Otimização dos parâmetros de forma a executar o movimento mais rápido e impulsionar a bola o mais longe possível ou para um determinado ponto;
- Resolver o problema da estabilidade. Uma abordagem seria efetuar movimentos com o braços, usando curvas de Bézier, para contrabalançar o movimento da perna.
- Expansão do movimento para incorporar chute de calcanhar. O método usado nesta dissertação torna-se ineficaz visto o pé não conseguir chegar ao outro lado da bola de forma a dar de calcanhar. Para a sua execução é necessário compensar a distância com movimentos do corpo.
- Incorporar o movimento num movimento de passo, isto é, executar o movimento em andamento sem que o andamento seja interrompido pelo chute omnidirecional. Uma abordagem seria finalizar o movimento com o pé posicionado numa posição necessária para o andamento.

Referências

- [1] Robocup Federation. Robocup. Disponível em <http://www.robocup.org/>, acessido pela última vez em Janeiro de 2012.
- [2] Hiroaki Kitano, Minoru Asada, Yasuo Kuniyoshi, Itsuki Noda, e Eiichi Osawa. Robocup: The robot world cup initiative, 1995.
- [3] Robocup Federation. Robocup 2012 - méxico. Disponível em <http://www.robocup2012.org/>, acessido pela última vez em Janeiro de 2012.
- [4] Hiroaki Kitano, Minoru Asada, Yasuo Kuniyoshi, Itsuki Noda, Eiichi Osawai, e Hitoshi Matsubara. RoboCup: A challenge problem for AI and robotics. páginas 1–19. 1998.
- [5] Robocup Federation. Robocup objective. Disponível em <http://www.robocup.org/about-robocup/objective/>, acessido pela última vez em Janeiro de 2012.
- [6] Robocup Federation. Robocup soccer. Disponível em <http://www.robocup.org/robocup-soccer/>, acessido pela última vez em Janeiro de 2012.
- [7] Robocup Federation. Robocup middle size league. Disponível em http://wiki.robocup.org/wiki/Middle_Size_League, acessido pela última vez em Janeiro de 2012.
- [8] 5DPO FEUP. Festival robótica 2010 - liga de robôs médios. Disponível em <http://paginas.fe.up.pt/~robosoc/en/lib/exe/fetch.php?w=&h=&cache=cache&media=soccer:2010-20.jpg>, acessido pela última vez em Janeiro de 2012.
- [9] Robocup Federation. Robocup small size league. Disponível em http://wiki.robocup.org/wiki/Small_Size_League, acessido pela última vez em Janeiro de 2012.
- [10] RoboCup. Robocup 2011 small size league - istambul - turkey. Disponível em http://www.robocup2011.org/files/images/photogallery/latestphotos/05_robocup_soccer_small_size_league.jpg, acessido pela última vez em Janeiro de 2012.
- [11] Robocup Federation. Robocup standard platform league. Disponível em http://wiki.robocup.org/wiki/Standard_Platform_League, acessido pela última vez em Janeiro de 2012.
- [12] The Telegraph. Robots warm up for the robocup football world cup 2009. Disponível em http://i.telegraph.co.uk/multimedia/archive/01436/Red-and-blue_1436364i.jpg, acessido pela última vez em Janeiro de 2012.

- [13] Robocup Federation. Robocup simulation league. Disponível em http://wiki.robocup.org/wiki/Soccer_Simulation_League, acessado pela última vez em Janeiro de 2012.
- [14] University of Oxford Department of Computer Science. Robocup 2d football simulation. Disponível em http://www.cs.ox.ac.uk/people/stephen.cameron/pictures/Football_2D.jpg, acessado pela última vez em Janeiro de 2012.
- [15] Tin-Man. Disponível em <http://code.google.com/p/tin-man/>, acessado pela última vez em Janeiro de 2012.
- [16] Robocup Federation. Robocup humanoid league. Disponível em http://wiki.robocup.org/wiki/Humanoid_League, acessado pela última vez em Janeiro de 2012.
- [17] Virginia Center for Autonomous Systems. Robocup 2010 - darwins vs dribblers. Disponível em <http://www.unmanned.vt.edu/news/images/robocupDARwInVsTUD800x600.jpg>, acessado pela última vez em Janeiro de 2012.
- [18] Satoshi Tadokoro, Hiroaki Kitano, Tomoichi Takahashi, Itsuki Noda, Hitoshi Matsubara, Atsushi Shinjoh, Tetsuhiko Koto, Ikuo Takeuchi, Hironao Takahashi, Fumitoshi Matsuno, Michinori Hatayama, Jun Nobe, e Susumu Shimada. The robocup-rescue project: A robotic approach to the disaster mitigation problem. 2000.
- [19] Robocup Federation. Robocup rescue. Disponível em <http://www.robocup.org/robocup-rescue/>, acessado pela última vez em Janeiro de 2012.
- [20] Robocup Federation. Robocup rescue - robot league. Disponível em http://wiki.robocup.org/wiki/Robot_League, acessado pela última vez em Janeiro de 2012.
- [21] Robocup Federation. Robocup rescue - simulation league. Disponível em http://wiki.robocup.org/wiki/Rescue_Simulation_League, acessado pela última vez em Janeiro de 2012.
- [22] RoboCup. Robocup 2011 rescue league - istanbul - turkey. Disponível em http://www.robocup2011.org/files/images/photogallery/latestphotos/11_robocup_rescue.jpg, acessado pela última vez em Janeiro de 2012.
- [23] RoboCup. Robocup - rescue simulation league. Disponível em <http://wiki.robocup.org/images/wiki.robocup.org/d/df/Foligno-ohne.gif>, acessado pela última vez em Janeiro de 2012.
- [24] Robocup Federation. Robocup junior. Disponível em <http://www.robocup.org/robocup-junior/>, acessado pela última vez em Janeiro de 2012.
- [25] Robocup Federation. Robocup@home league. Disponível em http://wiki.robocup.org/wiki/@Home_League, acessado pela última vez em Janeiro de 2012.
- [26] Tijn van der Zant e Thomas Wisspeintner. Robocup@home: Creating and benchmarking tomorrows service robot applications. Em *Robotic Soccer, I-Tech Education and Publishing*, páginas 521–528. 2007.
- [27] Tijn van der Zant e Thomas Wisspeintner. RoboCup x: A proposal for a new league where RoboCup goes real world. Em *RoboCup 2005: Robot Soccer World Cup IX*, volume 4020, chapter 15, páginas 166–172. Springer Berlin Heidelberg, 2006.

- [28] TU-Ilmenau. Robocup@home. Disponível em <http://www2.tu-ilmenau.de/ieee/media/archive1/robots/robocup.jpg>, acessado pela última vez em Janeiro de 2012.
- [29] Luís Reis e Nuno Lau. Fc portugal team description: Robocup 2000 simulation league champion. Em Peter Stone, Tucker Balch, e Gerhard Kraetzschmar, editores, *RoboCup 2000: Robot Soccer World Cup IV*, volume 2019 of Lecture Notes in Computer Science, páginas 29–40. Universidade do Minho, 2001.
- [30] Luís Paulo Reis, Nuno Lau, e Luís Mota. Fc portugal 2d simulation: Team description paper. Em *RoboCup 2010: Robot Soccer World Cup XIV*, 2010.
- [31] Luís Paulo Reis, Nuno Lau, António J. Neves, e António 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. páginas 1 – 3, 2009.
- [32] 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, e J. Vieira. Cambada'2011: Team description paper. 2011.
- [33] CAMBADA Team. Cambada robotic soccer. Disponível em <http://www.ieeta.pt/atri/cambada/index.htm>, acessado pela última vez em Janeiro de 2012.
- [34] Nuno Lau, Luís Seabra Lopes, Gustavo Corrente, e Nelson Filipe. Multi-robot team coordination through roles, positionings and coordinated procedures. Em *2009 IEEE/RSJ international conference on Intelligent robots and systems, IROS '09*, páginas 5841 – 5848. IEEE Press, 2009.
- [35] Nuno Lau, Luís Seabra Lopes, Nelson Filipe, e Gustavo Corrente. Roles, positionings and set plays to coordinate a robocup msl team. Em *14th Portuguese Conference on Artificial Intelligence: Progress in Artificial Intelligence, EPIA '09*, páginas 323 – 337, 2009.
- [36] 5DPO Team. 5dpo. Disponível em <http://paginas.fe.up.pt/~robosoc/en/doku.php>, acessado pela última vez em Janeiro de 2012.
- [37] 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, e A. P. Moreira. 5dpo'2011: Team description paper. 2011.
- [38] A.J.R. Neves, N. Lau, L.P. Reis, A.P. Moreira, A. Trifan, B. Pimentel, C. Sobrinho, e E. Domingues. Spl portuguese team: Team description paper for robocup 2011. 2011.
- [39] Julien Nicolas. Artificial evolution of controllers based on non-linear oscillators for bipedal locomotion. *Master's thesis*, 2004.
- [40] Henrik Hautop Lund e Orazio Miglino. From simulated to real robots. In *International Conference on Evolutionary Computation*, páginas 362 – 365, 1996.
- [41] Hugo Picado. Development of behaviors for a simulated humanoid robot. *Master's thesis*, 2008.
- [42] Joschka Boedecker, Klaus Dorer, Markus Rollmann, Yuan Xu, Feng Xue, Marian Buchta, e Hedayat Vatankhah. *SimSpark User's Manual*. 1.2 edição, January 2010.

- [43] Oliver Obst e Markus Rollmann. SPARK – A Generic Simulator for Physical Multiagent Simulations. *Computer Systems Science and Engineering*, 20(5):347–356, Setembro 2005.
- [44] RoboCup Community. Robocup soccer server 3d users manual. January 2010.
- [45] Joschka Boedecker e Minoru Asada. Simspark - concepts and application in the robocup 3d soccer simulation league. *SIMPAR-2008 Workshop on The Universe of RoboCup Simulators*, CD-ROM, 2008.
- [46] Nuno Filipe dos Reis Almeida. Control agent architecture of a simulated humanoid robot. *Master's thesis*, 2008.
- [47] Hugo Picado. Fcportugal3d - a team of robocup 3d simulation league. página 9, January 2009.
- [48] Nao Team Humboldt. Disponível em <http://www.naoteamhumboldt.de/wp-content/uploads/2010/09/SimSpark.png>, acessado pela última vez em Janeiro de 2012.
- [49] David Gouaillier, Vincent Hugel, Pierre Blazevic, Chris Kilner, Jérôme Monceaux, Pascal Lafourcade, Brice Marnier, Julien Serre, e Bruno Maisonnier. The nao humanoid: a combination of performance and affordability. July 2008.
- [50] Hugo Marques, Nuno Lau, e Luís Paulo 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*, 14:31 –37, April 2004.
- [51] M. Vidyasagar Mark W. Spong, Seth Hutchinson. Robot modeling and control. páginas 29 – 112, November 2005.
- [52] António P. Moreira. Acetatos das aulas teóricas da disciplina rind (mieec) - feup, 2009. Disponível em https://www.fe.up.pt/si/conteudos_service.conteudos_cont?pct_id=83161&pv_cod=50P9a1CaBzaa, acessado pela última vez em Janeiro de 2012.
- [53] Thomas Rofer, Tim Laue, Judith Muller, Alexander Fabisch, Fynn Feldpausch, Katharina Gillmann, Colin Graf, Thijs Jeffry de Haas, Alexander Hartl, Arne Humann, Daniel Honsel, Philipp Kastner, Tobias Kastner, Carsten Konemann, Benjamin Markowsky, Ole Jan Lars Riemann, e Felix Wenk. B-human: Team report and code release 2011. chapter 5. Disponível em http://www.b-human.de/downloads/bhuman11_coderelease.pdf, acessado pela última vez em Janeiro de 2012.
- [54] Thomas 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.
- [55] J. C.; Bartels, R. H.; Beatty e B. A Barsky. Bézier curves. *Ch. 10 in An Introduction to Splines for Use in Computer Graphics and Geometric Modelling*, páginas 211 – 245, 1998.
- [56] Judith Muller, Tim Laue, e Thomas 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*, páginas 119 – 120, Springer 2011.

- [57] Eric W. Weisstein. "bézier curve."from mathworld—a wolfram web resource. Disponível em <http://mathworld.wolfram.com/BezierCurve.html>, acessido pela última vez em Janeiro de 2012.
- [58] C.K. Shene. Introduction to computing with geometry notes. unit 5: Bézier curves. Disponível em <http://www.cs.mtu.edu/~shene/COURSES/cs3621/NOTES/>, acessido pela última vez em Janeiro de 2012.
- [59] David L. Finn. Ma 323 geometric modeling—course notes: Day 14—properties of bezier curves. Disponível em <http://www.rose-hulman.edu/~finn/CCLI/Notes/day14.pdf>, acessido pela última vez em Janeiro de 2012.
- [60] Paul A. Tipler e Gene Mosca. *Physics for Scientists and Engineers Extended Version*. W. H. Freeman, fifth edição, August 2003.
- [61] Tom W. B. Kibble e Frank H. Berkshire. *Classical Mechanics*. World Scientific Publishing Company, fifth edição, June 2004.
- [62] Bruno Pimentel, Nuno Lau, e Luís Paulo Reis. Flexible movement planning in humanoid soccer.