



Universidade do Estado da Bahia
Departamento de Ciências Exatas e da Terra
Bacharelado em Sistemas de Informação

Planejamento e mapeamento de trajetórias em tempo real para navegação de robôs humanóides em ambiente simulado 3D

Alan dos Santos Soares

Rua Silveira Martins, 2555, Cabula, Salvador-BA

13 de Julho de 2014

Alan dos Santos Soares

Planejamento e mapeamento de trajetórias em tempo real para navegação de robôs humanóides em ambiente simulado 3D

Trabalho de Conclusão de Curso apresentado a Universidade do Estado da Bahia como requisito parcial para obtenção do título de Bacharel em Sistemas de Informação.

Orientador:

Marco Antonio Costa Simões

Co-orientador:

Diego Gervasio Frías Suárez

UNEB - UNIVERSIDADE DO ESTADO DA BAHIA
DEPARTAMENTO DE CIÊNCIAS EXATAS E DA TERRA

Rua Silveira Martins, 2555, Cabula, Salvador-BA

13 de agosto de 2014

Alan dos Santos Soares

Planejamento e mapeamento de trajetórias em tempo real para navegação de robôs humanóides em ambiente simulado 3D

Trabalho de Conclusão de Curso apresentado a Universidade do Estado da Bahia como requisito parcial para obtenção do título de Bacharel em Sistemas de Informação.

Data de aprovação :

Prof. Marco Antonio Costa Simões
MSc. em Ciência da Computação
Universidade do Estado da Bahia

Prof. Diego Gervasio Frías Suárez
PhD. em Modelagem Computacional
Universidade do Estado da Bahia

Prof. Josemar Rodrigues de Souza
PhD. em Informática
Universidade do Estado da Bahia

Rua Silveira Martins, 2555, Cabula, Salvador-BA

13 de Junho de 2014

*Dedico esta monografia a minha família,
pelo apoio fornecido e aos meus amigos.*

Agradecimentos

Agradeço primeiramente a minha mãe Alaide Cruz dos Santos Soares que me ensinou e me educou durante toda a minha vida, me dando apoio e amparo nos momentos bons e ruins quando eu mais precisei.

Aos meus professores e orientadores Diego Gervásio Frias Suárez e Marco Antonio Costa Simões pelos sábios ensinamentos, críticas, pelas experiências compartilhadas e pela atenção dada nos momentos em que eu mais precisei.

A todos os meus familiares, irmão e irmãs que contribuíram direta e indiretamente para que eu pudesse chegar ao final do curso, em especial a minha tia Clarice Santos, que sem ela eu não teria conseguido efetuar a minha matrícula no curso.

A todos os meus colegas de faculdade, em especial a Leandreson Ferreira e Leone de Jesus por toda a experiência vivida durante o período acadêmico.

E por fim, a todos os membros do grupo de pesquisa ACSO/BRT/UNEB, pelas experiências compartilhadas, pelas risadas descontraídas, pelo conhecimento compartilhado e por tudo que conquistei estando presente em um dos melhores grupos de pesquisa da Bahia/Brasil.

Resumo

A necessidade da automatização e da construção de agentes robóticos capazes de realizar tarefas complexas, perigosas, delicadas, em ambientes dinâmicos, tornou-se um desafio aos investigadores. Alguns desses desafios incluem o mapeamento de ambientes dinâmicos, planeamento de trajetória e a predição de colisão. Este trabalho propõe um modelo para mapeamento e planeamento de trajetórias com predição de colisão para agentes robóticos em ambiente simulado. O modelo foi desenvolvido utilizando o time base do BahiaRT, desenvolvido pelo grupo de pesquisa ACSO/UNEB em parceria com o FCPortugal(FEUP-LIACC/Univ. Aveiro). A validação do projeto foi realizada utilizando uma aplicação de passe em futebol robótico integrado ao modelo proposto.

Palavras-chaves: Mapeamento. Planeamento de trajetórias. Predição. Simulação 3D.

Abstract

The need for automation and construction of robotic agents capable of performing complex tasks, dangerous, delicate in dynamic environments, it has become a challenge for researchers. Some of these challenges include mapping of dynamic environments, path planning and collision prediction. This work proposes a model for mapping and path planning with collision prediction for robotic agents in a simulated environment. The model was developed using the BahiaRT team base, developed by the research group ACSO / UNEB in partnership with FCPortugal (FEUP-LIACC/Univ. Aveiro). The validation of the project was performed using an application of the pass in robotic soccer integrated in the proposed model.

Key-words: Mapping. Path planning. Navigation. Prediction. Simulation 3D.

Lista de Figuras

2.1	Arquitetura de um agente com capacidade de comunicação.	p. 19
2.2	Garry Kasparov x Deep Blue (Computador da IBM), jogando uma partida de Xadrez.	p. 22
2.3	Simulação do Robô NAO da empresa Aldebaran Robotics feita pelo servidor oficial da liga 3D.	p. 24
2.4	Simulação em tempo real de agentes no RoboViz utilizando um plot gráfico para análise de comportamentos.	p. 25
3.1	Etapas da metodologia do projeto.	p. 30
3.2	Arquitetura do projeto desenvolvido integrado a um modelo de passe desenvolvido para validação.	p. 32
3.3	Imagem do plot realizado pelo Roboviz representando os 5 tipos de mapeamento que é realizado pelo modelo desenvolvido.	p. 34
3.4	Diagrama da classe CellFied utilizado no modelo para representar uma unidade do mapeamento.	p. 35
3.5	Sequência de pontos medidos de um objeto monitorado pelo agente no campo de futebol	p. 37
3.6	Sequência de pontos medidos e estimados de um objeto monitorado pelo agente no campo de futebol	p. 37
4.1	Arquitetura do ambiente de teste utilizado para coletar os resultados. .	p. 43
4.2	Arquitetura da ferramenta Trainer3D.	p. 44
4.3	Cenários de situações de um jogo real de futebol onde um agente tem a possibilidade de efetuar um passe para outro agente aliado em uma posição estratégica.	p. 46
4.4	Resultados dos testes realizados com agentes Dummy nos cenários pre-definidos.	p. 51

4.5	Resultados dos testes realizados com agentes Dummy nos cenários pre- definidos.	p. 51
-----	--	-------

Lista de Tabelas

2.1	Diferenças entre as características dos domínios do RoboCup e Xadrez.	p. 23
2.2	Principais times que participam da RoboCup e sua Universidade. . . .	p. 26
4.1	Configuração dos computadores utilizados no ambiente de teste para coleta dos resultados.	p. 42
4.2	Nível A é considerado fácil, o B é considerado moderado, C é considerado difícil, e o D é considerado muito difícil.	p. 45
4.3	Tabela com o resultados preliminares dos testes realizado com times da liga de simulação 3D	p. 48
4.4	Tabela com o resultado dos testes realizado com times da liga de simulação 3D	p. 49
4.5	Tabela com o resultado das partidas realizadas no processo de validação	p. 50

Lista de abreviaturas e siglas

ACSO Núcleo de Arquitetura de Computadores e Sistemas Operacionais

DARPA Defense Advanced Research Projects Agency

IA Inteligencia Artificial

IAD Inteligencia Artificial Distribuida

OBR Olimpíada Brasileira de Robótica

PEAS Performance Environment Actuators Sensors

SMA Sistemas Multi-Agente

TDP Team Description Paper

Lista de algoritmos

3.1	Implementação da verificação da presença de obstáculos	p.35
3.2	Implementação da predição de colisão	p.41

Sumário

1	Introdução	p. 14
2	Fundamentos de robótica e planejamento de trajetória	p. 17
2.1	Sistemas Multi-Agente (SMA)	p. 17
2.1.1	Comunicação em SMA	p. 19
2.1.2	Coordenação em SMA	p. 19
2.2	Futebol de Robôs	p. 21
2.2.1	Ligas da RoboCup	p. 21
2.2.2	Xadrez vs RoboCup	p. 22
2.2.3	Simulação 3D	p. 23
2.2.4	Simspark	p. 24
2.2.5	Roboviz	p. 25
2.2.6	Times da Simulação 3D	p. 25
2.3	Planejamento de trajetória	p. 27
3	Modelo de mapeamento e planejamento de trajetória	p. 29
3.1	Trabalhos relacionados	p. 29
3.2	Proposta Metodológica	p. 30
3.3	Arquitetura do modelo	p. 31
3.4	Mapeamento das trajetórias	p. 33
3.5	Verificação da presença de obstáculos	p. 35
3.6	Determinação da trajetória de objetos dinâmicos em ambientes ruidosos	p. 36

3.7	Predição de colisão dos obstáculos	p. 40
3.8	Critério do descarte de obstáculos	p. 41
4	Validação e Resultados Obtidos	p. 42
4.1	Metodologia de Teste	p. 42
4.2	Trainer3D	p. 44
4.3	Cenários de Teste	p. 45
4.4	Métricas de Avaliação	p. 46
4.5	Resultados	p. 47
4.5.1	Resultados de partidas utilizando cenário de jogo real	p. 47
4.5.2	Resultados utilizando cenários pré-definidos	p. 50
5	Considerações Finais	p. 52
	Referências Bibliográficas	p. 54

1 Introdução

Um dos principais desafios no campo da robótica, é a forma como os agentes humanóides buscam uma trajetória para chegar a um determinado objetivo. No ambiente em que o agente está atuando existem diversos obstáculos que podem comprometer a sua locomoção no ambiente, principalmente se os obstáculos não forem estáticos.

Assim como o futebol de robôs, o desenvolvimento de veículos autônomos, designado como qualquer veículo com capacidade de transporte de pessoas ou bens sem a utilização de um condutor humano, é um dos desafios que requerem o planejamento e mapeamento de trajetórias em tempo real para evitar colisão com obstáculos móveis da mesma forma que o futebol de robôs.

A construção de veículos terrestres autônomos tem como principais objetivos, reduzir os acidentes de trânsito provocados por fatores humanos, aumentar a produtividade e otimizar os recursos veiculares através da utilização adequada de componentes. Para fomentar e estimular pesquisas no desenvolvimento de veículos autônomos, desafios como o DARPA Urban Challenge e o DARPA Grand Challenge foram criados (DARPA, 2002).

Outro desafio hoje em destaque é o desenvolvimento de robôs de serviço. Estes são robôs autônomos capazes de se mover em um ambiente dinâmico para realizar tarefas úteis para o bem estar de humanos e a preservação de equipamentos físicos, necessitando planejar uma trajetória de forma eficiente e segura, evitando causar acidentes. Segundo Dudek e Jenkin (2010) e Siegwart e Nourbakhsh (2004), o estudo das questões computacionais envolvidas na movimentação não supervisionada de robôs é fundamental para se obter sucesso no desenvolvimento de agentes capazes de atuarem em ambientes dinâmico e contínuo.

Para fomentar a investigação, uma organização internacional chamada RoboCup (ROBOCUP, 2001) foi formada. O objetivo é promover e estimular pesquisas nas áreas de Inteligência Artificial Distribuída (IAD) e Robótica Inteligente, colocando problemas de investigação distintos, mas ao mesmo tempo interrelacionados.

A sub-liga da RoboCup, Simulação 3D (ROBOCUP, 2012a), primeira dentre as simuladas a representar um robô humanóide em sua competição, tem como desafio padrão o futebol de robôs, dispondo de um vasto conjunto de desafios aos investigadores. Tais desafios como desenvolvimento do controle de baixo nível de robôs humanóides e a criação de comportamentos básicos e eficientes como levantar e correr são essenciais para se ter um time competitivo. Algumas das características do domínio e desafios associados mais importantes colocados pelo simulador incluem simulação em tempo real, modelo energético realista, comunicação pouco confiável e com baixa largura de banda, percepção e ações assíncronas, ambiente multi-objetivo, parcialmente cooperativo e parcialmente adverso (REIS, 2001). O futebol de robôs configura-se, portanto, como uma plataforma universal para testes avançados em robótica autônoma com possibilidades em aplicações em diversas áreas.

Este trabalho tem como objetivo propor e validar um modelo de mapeamento e planejamento de trajetórias de agentes humanóides no ambiente simulado 3D no contexto de jogo de futebol de robôs utilizando o time base BahiaRT do grupo de pesquisa ACSO (Núcleo de Arquitetura de Computadores e Sistemas Operacionais) da UNEB. O time possui uma parceria com time FCPortugal (FEUP-LIACC / Univ. Aveiro) de agentes simulados 3D, que tem como objetivo construir agentes totalmente autônomos e inteligentes e vencer a sub-liga 3D da RoboCup.

Para validar o modelo proposto, uma aplicação de passe foi desenvolvida. Este tem como objetivo ser utilizado para gerar situações estratégicas que aumentem a chance de uma jogada bem sucedida. Porém, o sucesso de um passe depende não somente da escolha da melhor trajetória da bola, mais também da predição de interceptação da bola e de comportamentos básicos eficientes como se posicionar e chutar a bola.

O estudo e desenvolvimento do projeto na área de robôs humanóides simulados, atende à necessidade estratégica de continuar criando competências e desenvolvendo *know-how* próprio no estado da Bahia e no país na área de robótica autônoma. Este contribui para não sermos, em um futuro imediato, meros consumidores de tecnologia desenvolvida em outros estados e outros países.

Este trabalho está dividido em 5 capítulos. O Capítulo 2 se refere aos conceitos teóricos do problema e uma descrição do futebol de robôs.

No Capítulo 3 são descritos os trabalhos correlatos, o objetivo do trabalho, a metodologia utilizada e o modelo desenvolvido.

No Capítulo 4 é apresentada a descrição da metodologia de teste utilizada, a arquitetura do ambiente de teste, as métricas utilizadas para avaliar o modelo e os resultados obtidos.

Por último, o Capítulo 5 descreve as considerações finais e os trabalhos futuros.

2 Fundamentos de robótica e planejamento de trajetória

O objetivo deste capítulo é fazer uma contextualização do modelo desenvolvido descrevendo as principais abordagens de Sistemas Multi-Agente (SMA), seus principais benefícios, tanto dentro como fora da área da robótica. Neste capítulo também é feita uma elucidação do conceito de planejamento de trajetória e alguns trabalhos relacionados. Além disso, é feita uma descrição de plataformas utilizadas no desenvolvimento de pesquisa na área de robótica no contexto de futebol de robôs em ambiente simulado.

Na Seção 2.1 é feita uma descrição do conceito de SMA e os principais benefícios de sua utilização. São também introduzidos conceitos de comunicação em SMA na sub Seção 2.1.1. Por fim, na sub Seção 2.1.2 são descritos os tipos de coordenação em SMA e o tipo de coordenação utilizada no BahiaRT para alcançar o objetivo deste projeto.

A Seção 2.2 tem como objetivo realizar uma contextualização do objetivo deste trabalho. Na sub Seção 2.2.1 é feita uma breve descrição da RoboCup e algumas ligas padrões definidas como desafios. Na sub Seção 2.2.2 é realizada uma abordagem do Xadrez computacional e do futebol de robôs. Nas sub Seções 2.2.5, 2.2.4 são descritas as plataformas utilizadas na simulação 3D que é descrita na sub Seção 2.2.3 e posteriormente descrito os principais times campeões dos últimos anos na RoboCup na liga de simulação 3D na sub Seção 2.2.6.

Por último, a Seção 2.3 descreve o conceito de planejamento de trajetória, alguns trabalhos relacionados e os desafios em aberto.

2.1 Sistemas Multi-Agente (SMA)

Um SMA é um sistema composto por múltiplos agentes interagindo em um ambiente compartilhado. Cada agente tem a capacidade de agir de forma autônoma tomando

decisões que o levarão a atingir o seu objetivo. A interação entre os agentes pode ser realizada através de protocolos que são baseados em comportamentos humano. Tal interação tem como objetivo realizar a coordenação, cooperação, competição ou negociação dos agentes (REIS, 2003).

Os SMA tem como objetivo resolver um problema de forma distribuída, aumentando a eficiência da resolução de um problema. Existem diversos fatores que demonstram que a utilização de SMA podem trazer benefícios para resolver determinados problemas (STONE; VELOSO, 1996):

1. O paralelismo, atribuindo diferentes tarefas a diferentes agentes de forma que sua execução seja mais rápida;
2. A robustez, pois utilizam-se diferentes agentes, não existindo desta forma um ponto único de falha no sistema;
3. A escalabilidade, permitindo o aumento de agentes de um determinado sistema aberto;
4. A simplificação das tarefas individuais, dividindo o problema global em subproblemas;
5. A manutenção da privacidade da informação e conhecimentos individuais de cada agente.

Na Inteligência Artificial (IA), a utilização de SMA trazem diversos benefícios, principalmente da rentabilidade de recursos para problemas onde o conhecimento ou atividade é distribuída:

1. Resolução mais rápida de problemas por conta do processamento concorrente;
2. Diminuição da comunicação devido ao processamento estar localizado junto á fonte de informação e a comunicação ser realizada em alto-nível;
3. Aumento da flexibilidade e escalabilidade resultantes da possibilidade de interconexão de múltiplos sistemas com arquiteturas distintas;
4. Facilidade de desenvolvimento do sistema devido a modularidade resultante da decomposição dos problemas e da decomposição dos sistemas em agentes semi-autônomos.

2.1.1 Comunicação em SMA

Um agente possui a capacidade de percepção, processamento e atuação em um determinado ambiente. Desta forma, a capacidade de se comunicar é considerada como um módulo de comunicação, que é dividido em percepção (recepção das mensagens) e de ação (envio de mensagens). Este módulo está ligado diretamente ao módulo central do agente (módulo inteligente), que permite o agente ter acesso as mensagens recebidas do servidor e definir quais as mensagens devem ser enviadas. Uma arquitetura de um agente deliberativo (ou híbrido) com capacidade de comunicação pode ser visto na figura 2.1.

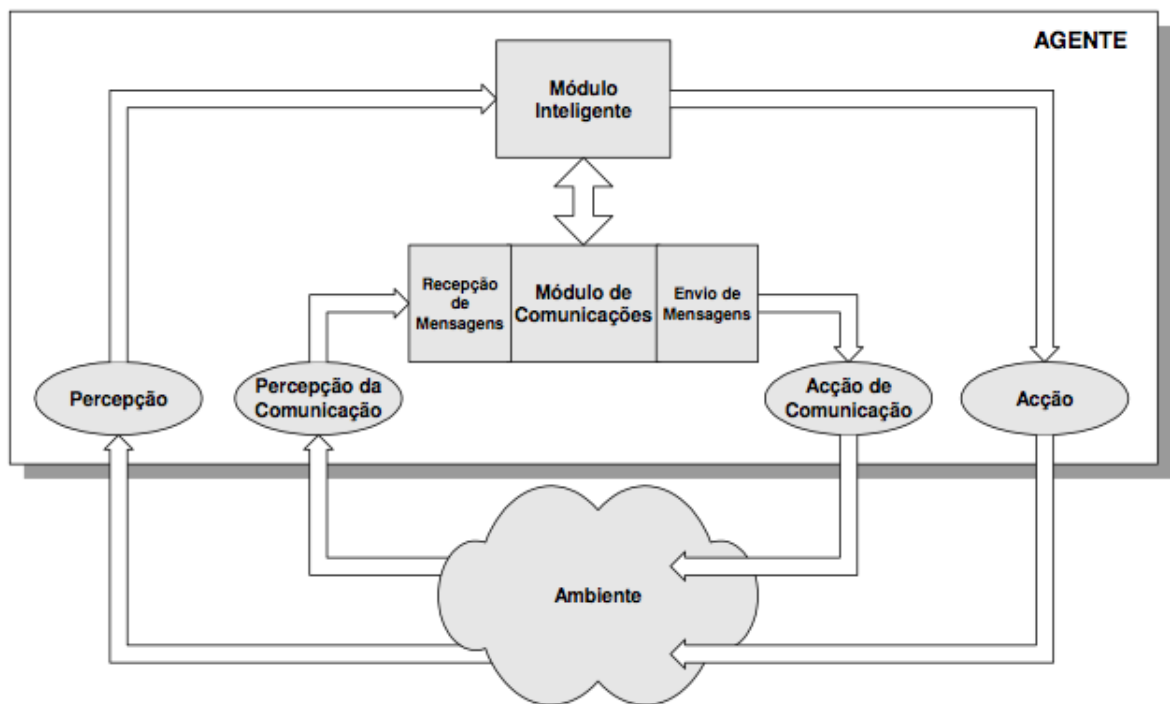


Figura 2.1: Arquitetura de um agente com capacidade de comunicação.

Fonte: (REIS, 2003)

2.1.2 Coordenação em SMA

O Futebol Robótico em geral e a liga de simulação em particular, constituem domínios particularmente adequados a aplicação da investigação realizada em metodologias de coordenação em SMA, nomeadamente no que diz respeito a coordenação para agentes cooperativos (REIS, 2003).

O domínio do futebol robótico é complexo, dinâmico, parcialmente cooperativo e parcialmente adverso. Existem diversos tipos de coordenação aplicados ao domínio do futebol robótico. Neste trabalho só será descrito o tipo de coordenação que está sendo utilizado no

desenvolvimento do modelo, porém a descrição dos diversos tipos de coordenação poderão ser encontradas em Reis (2003). A coordenação pode ser:

1. Coordenação por Comunicação.
2. Coordenação por Percepção Inteligente,
3. Coordenação por Modelização Mútua.
4. Coordenação Estratégica.
5. Coordenação Parcialmente Hierarquica.

O tipo de coordenação utilizada neste modelo é focada na coordenação por comunicação. A coordenação por comunicação utiliza um protocolo que permite que os agentes possam trocar informações acerca do seu estado de mundo e outras informações úteis para a coordenação. Este protocolo é baseado no envio de mensagens através de um canal de comunicação entre o agente e o servidor.

A baixa largura de banda do domínio e as restrições impostas à comunicação, são fatores que tornam o futebol robótico simulado um domínio atraente para a aplicação de metodologias de coordenação por comunicação.

O time BahiaRT do ACSO, além de utilizar coordenação por comunicação, utiliza a coordenação estratégica. Este tipo de coordenação utiliza como premissa a organização dos agentes de tal forma que eles possam definir uma estratégia para um dado jogo, ou situação, composta de táticas e definição de papéis (tipos de jogadores) especificando o comportamento individual e coletivo dos agentes.

A construção de agentes capazes de balancear a reatividade com a capacidade de deliberação individual e social são fundamentais para que os agentes realizem a cooperação. O estado do mundo de um agente estratégico é definido como uma estrutura multi-nível contendo desde informações de baixo nível (incluindo as posições, velocidade e orientações dos objetos presentes no mundo), até informação de nível estratégico (incluindo informação temporal estratégica que irá permitir a seleção de táticas a serem utilizadas). Este estado de mundo é atualizado utilizando informação proveniente da percepção do agente, das comunicações recebidas (mensagens), da dinâmica do mundo e da predição dos efeitos das suas ações e das ações dos outros agentes.

2.2 Futebol de Robôs

A iniciativa RoboCup (KITANO et al., 1995) (KITANO et al., 1997) é um projeto de investigação e educação internacional que tem como objetivo promover a investigação em Inteligência Artificial Distribuída (IAD) e Robótica Inteligente (ROBOCUP, 2001). O projeto tem como base a utilização de um problema padrão – o futebol robótico – onde o desenvolvimento de tecnologias para construir uma equipe de robôs reais ou virtuais seja capaz de participar de um desafio de futebol seguindo regras de jogo pré-especificadas.

Como forma de promover a investigação na área, foi lançado um objetivo de longo prazo:

“No ano de 2050, uma equipe de robôs autônomos humanóides, ser capaz de vencer a equipe campeã do mundo de futebol, em uma partida disputada de acordo com as regras da FIFA.” (KITANO et al., 1997)

Este objetivo é atualmente partilhado como um dos grandes desafios da área da IA e Robótica. Embora este desafio, à luz da ciência e tecnologia atuais, pareça altamente ambicioso, a colocação de objetivos científicos bem definidos de longo prazo, tem sido, ao longo dos anos, uma forma de estimular o desenvolvimento científico (ROBOCUP, 2001). Além disso, os sub-objetivos que são colocados pela RoboCup através das ligas, convergem para alcançar o objetivo final.

2.2.1 Ligas da RoboCup

O futebol robótico inclui diversas ligas que se dividem em dois tipos: ligas robóticas (utilizando robôs pequenos, médios e humanóides) e a liga de simulação. O objetivo é fazer com que cada liga se concentre nos desafios propostos, dando ênfase em determinados tópicos necessários para fazer com que equipes de robôs possam disputar uma partida de futebol. Por exemplo, na liga de simulação, a ênfase é colocada na coordenação em SMA, enquanto na liga de robôs pequenos, a ênfase é colocada no controle rápido e preciso dos robôs e na liga de robôs médios, os tópicos mais importantes incluem a visão computacional, projeto electromecânico e auto-localização dos robôs.

A RoboCup Rescue (KITANO, 1999) e o RoboCup Júnior (SKLAR; EGUCHI; JOHNSON, 2003) são também outras iniciativas associadas ao futebol robótico. O *Rescue* divide-se em robótica física e simulada e tem como objetivo estimular a aplicação da investigação realizada no futebol robótico, a domínios socialmente mais úteis, no caso,

missões de salvamento e resgate em grandes catástrofes (RESCUE, 2001). Já a RoboCup Júnior surgiu como uma forma de estimular os mais jovens a participar da RoboCup. A OBR (Olimpíada Brasileira de Robótica) é um dos eventos que promovem a inclusão de jovens em idade escolar a construir e colocar em funcionamento os seus robôs para realizar diversas tarefas e atraí-los para a CBR (Campeonato Brasileiro de Robótica), que vem atraído a cada ano mais estudantes.

2.2.2 Xadrez vs RoboCup

Um desafio semelhante colocado aos investigadores em IA no decurso das últimas 4 décadas, consistiu em construir um agente (programa) que fosse capaz de vencer o campeão mundial de Xadrez utilizando as regras oficiais da Federação Internacional de Xadrez. Tal desafio mostrou a importância da existência de problemas padrão em que diferentes metodologias e avanços científicos podem ser comparados.

Diversos algoritmos de pesquisa, arquiteturas de computadores e metodologias científicas foram desenvolvidas para este domínio. Em Maio de 1997, o computador Deep Blue da IBM (BLUE, 1997) derrotou Gary Kasparov (o campeão humano de Xadrez), figura 2.2 , utilizando as regras oficiais do Xadrez. Com esta vitória, o desafio de 40 anos da IA utilizando o Xadrez como domínio de aplicação ficou muito próximo de um final com sucesso.



Figura 2.2: Garry Kasparov x Deep Blue (Computador da IBM), jogando uma partida de Xadrez.

Fonte: (FLAMENCOS, 1998)

Uma das características que tornou o Xadrez computadorizado um problema padrão foi a facilidade que este domínio ofereceu para a comparação de abordagens distintas, e

para a avaliação do progresso científico global realizado no domínio. No entanto, com a chegada ao final do desafio associado ao Xadrez, no mundo da IA, novos domínios e problemas padrão mais complexos e estimulantes tornaram-se necessários. Foi neste contexto que se desenvolveu o desafio do futebol robótico como um problema padrão para a IAD e Robótica Inteligente.

As principais diferenças entre o domínio do Xadrez e a RoboCup podem ser visualizadas na tabela 2.1.

Tabela 2.1: Diferenças entre as características dos domínios do RoboCup e Xadrez.

Fonte: (REIS, 2003)

Características / Domínio	Xadrez	RoboCup
Ambiente	Estático	Diamico
Mudança de Estado	Por turnos	Tempo-Real
Acessibilidade ao Estado do Mundo	Completa	Incompleta
Resultado das Ações	Determinístico	Não Determinístico
Leitura dos Sensores	Discreta (Simbólica)	Contínua (não Simbólica)
Utilização dos Atuadores	Discreta (Simbólica)	Contínua (não Simbólica)
Controle	Centralizado	Distribuído

A RoboCup foi desta forma projetada de forma a colocar num mundo limitado, um conjunto elevado de complexidades do mundo real, mantendo no entanto o custo, complexidade global e dimensão do problema, acessível aos grupos de investigação em Robótica e IA. Tais problemas de investigação colocados pela RoboCup de uma forma integrada, cobrem uma vasta área dos domínios da IA e Robótica, incluindo: coordenação, cooperação e comunicação multi-agente, arquiteturas de agentes inteligentes, aprendizagem, planeamento em tempo-real, decisão estratégica e tática, comportamento reativo, visão computacional, processamento e análise de imagem, sistemas de locomoção e atuação, sistemas sensoriais, fusão sensorial em tempo-real, navegação, controle inteligente robótico e outros.

2.2.3 Simulação 3D

A Simulação 3D de futebol de robôs da RoboCup é uma plataforma que tem como objetivo facilitar o desenvolvimento das pesquisas e diminuir o custo com experimentos que os robôs físicos demandam.

A plataforma se esforça para reproduzir os desafios de programação de *Software* enfrentados ao construir robôs físicos reais para esta finalidade. Com a utilização da simulação

3D, a investigação encurta o caminho para conseguir atingir a meta da Federação RoboCup de desenvolver uma equipe de robôs humanóides totalmente autônomos que possa vencer a equipe campeã mundial de futebol humano em 2050.

2.2.4 Simspark

SimSpark Robocup (2012b) é um sistema de simulação multi-agente para agentes em ambiente tridimensional desenvolvido como parte da iniciativa RoboCup. Seu objetivo é fornecer um alto grau de flexibilidade para a criação de novos tipos de simulações. Baseia-se em um quadro de aplicação flexível e esgota a idéia de componentes substituíveis ao longo de sua implementação.

Em comparação com simuladores especializados, os usuários podem criar novas simulações utilizando uma linguagem de descrição de cena. O SimSpark é uma ferramenta poderosa, pois abrange diferentes questões de investigação multi-agente, e é usado como o simulador oficial para a competição *RoboCup Simulation League*.

O *rcsserver3d* é o ambiente de competição oficial para o *Soccer Simulation League* em 3D na RoboCup. Ele implementa uma simulação de futebol, onde duas equipes de até onze robôs humanóides podem jogar uns contra os outros. Esta configuração aparentemente simples, representa um desafio para os implementadores de agentes em vários níveis.

O modelo de robô utilizado na simulação nas competições é atualmente o NAO, figura 2.3. Contudo, a utilização de agentes heterogeneos tem ganhado destaque na área científica, que agora querem criar modelos capazes de se adequarem a qualquer tipo de robô.

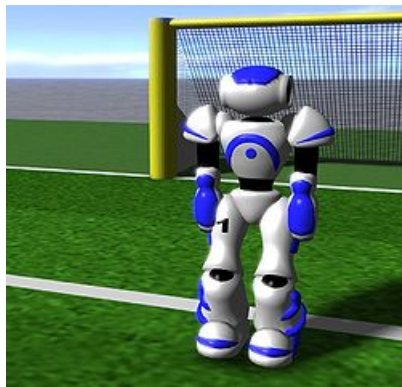


Figura 2.3: Simulação do Robô NAO da empresa Aldebaran Robotics feita pelo servidor oficial da liga 3D.

Fonte: (ROBOCUP, 2012a).

2.2.5 Roboviz

RoboViz Stoecker e Visser (2012) é um programa de *Software* projetado para avaliar e desenvolver comportamentos de agentes em sistemas multi-agente. RoboViz é um monitor interativo que torna o agente e informações sobre o estado do mundo em uma cena tridimensional. Além disso, o RoboViz fornece um design programável e funcionalidade de depuração para os agentes que podem se comunicar através de uma rede.

A ferramenta facilita a visualização em tempo real de agentes em execução simultânea no simulador SimSpark, fornecendo uma análise de nível superior e visualização de comportamentos do agente, figura 2.4, que não está atualmente disponível nas ferramentas já existentes.

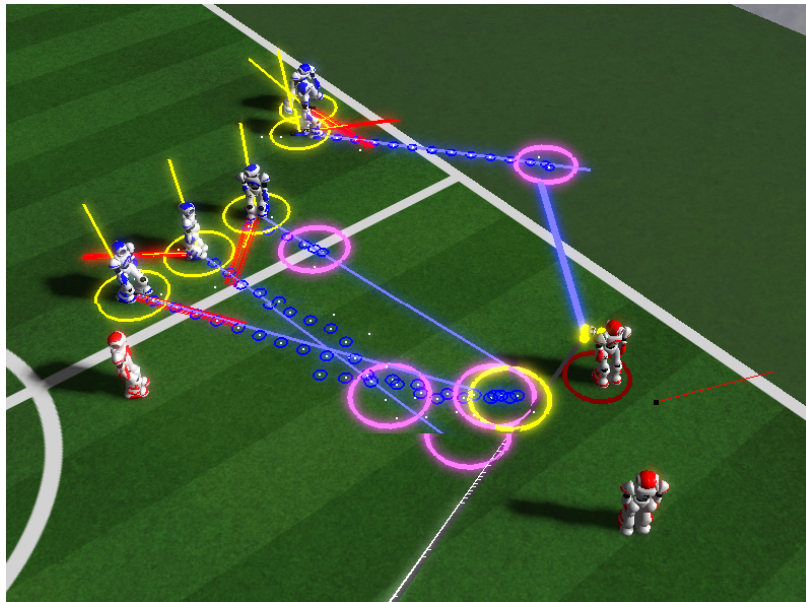


Figura 2.4: Simulação em tempo real de agentes no RoboViz utilizando um plot gráfico para análise de comportamentos.

Fonte: (ROBOVIZ, 2012)

2.2.6 Times da Simulação 3D

A liga de simulação 3D é composta por times que foram desenvolvidos por grupos de pesquisa de diversos países. A quantidade de equipes que possuem times competitivos é pequena, isso se dá pelo fato de que o desenvolvimento de uma equipe requer uma pesquisa aprofundada sobre movimentação de agentes, protocolos de comunicação, planejamento e coordenação multi-agente e IAD.

Um dos fatores que facilitam grupos de pesquisa se inserirem na competição, são

os códigos de times base disponibilizados por equipes para fomentar e estimular a pesquisa no âmbito da robótica autônoma simulada. Como por exemplo, o MagmaOffenburg (MAGMAOFFENBURG, 2014), um time base cedido para a comunidade científica.

Atualmente os principais times que participam da RoboCup podem ser visualizados na tabela 2.2.

Tabela 2.2: Principais times que participam da RoboCup e sua Universidade.

Fonte: (ROBOCUP, 2012a)

Time	Universidade
Apollo3D	Nanjing University of Posts and Telecommunications
‘ AUA3D	Anhui University of Architecture
BahiaRT	Universidade do Estado da Bahia
Bold Heart	University of Hertfordshire
cit3d	Changzhou Institute of Technology
FCPortugal	University of Aveiro/University of Minho/University of Porto
FUT-K3D	Fukui University of Technology
HfutEngine3D	HFUT
ITAndroids	Instituto Tecnológico de Aeronautica
IUIM3D	Iran University of Industries Mines (Tehran Center Branch)
KarachiKoalas	University of Technology, Sydney and Institute of Business Administration, Karachi
KylinSky3D	Hohai University Wentian College / Hohai University
L3M-SIM	Paris8 University
magmaOffenburg	Hochschule Offenburg
Miracle3D	Hefei Normal University
Mithras3D	Farzanegan Tehran
Nexus3D	Ferdowsi University of Mashhad
ODENS	Osaka Electro-Communication University
Paydar3D	Sharif University Of Technology
Rightel	University of Tehran
RoboCanes	University of Miami
Scorpius	AmirKabir University of Technology (AUT)
SEU-Jolly	Southeast University
UTAustinVilla	University of Texas at Austin

Além dos times base, o TDP (*Team Description Paper*), que são descrições de cada time e artigos sobre as metodologias empregadas em seus times, são divulgadas em simpósios e congressos, facilitando o trabalho de outras equipes e fomentando a melhoria contínua de metodologias empregadas para resolver problemas em aberto.

Alguns dos problemas em aberto nesta liga são: como fazer com que os agentes corram, fazer com que agentes efetuem um passe de forma coordenada e planejada, desviar de obstáculos móveis e atuarem com sucesso em ambientes dinâmicos. Um dos times que mais se aproxima da movimentação baseada no comportamento humano é o MagmaOffenburg (GLASER; DORER, 2013), que por sua vez poderá ser um dos primeiros times a conseguir fazer com que o robô consiga correr.

Um outro time que ganhou destaque nas últimas competições foi o UTAustinVilla

(MACALPINE et al., 2013). Desenvolvido por um grupo de estudantes da Universidade de Austin, Estados Unidos, coordenado por Peter Stone, pesquisador renomado na área de robótica. Uma das principais características que determinaram o seu sucesso foi a sua movimentação omnidirecional com capacidade de se manter estável mesmo colidindo com outros agentes.

O time FCPortugal Lau et al. (2013) desenvolvido pelo grupo de pesquisa da Universidade do Porto, Portugal, terceiro lugar na simulação 3D da RoboCup 2013, teve como diferencial o seu super chute, capaz de alcançar até 20 metros, mais da metade do campo utilizado pelo servidor Simspark. Apesar de ser um chute estático, o seu alcance é fruto da otimização do *script* utilizado na perna que realiza o chute.

Na RoboCup de 2013, o time vencedor da simulação 3D foi o Apollo3D (LIU et al., 2013). Uma das principais contribuições que o grupo de pesquisa que desenvolveu o Apollo3D vem dispondo a comunidade científica é o seu código fonte.

A maioria dos times hoje utilizam uma movimentação omnidirecional e dinâmica, onde suas poses são calculadas e alcançadas através de modelos matemáticos utilizando cinemática inversa. Tais modelos substituíram a movimentação estática, que dificultava a realização de jogadas e comportamentos devido o fato de que o ambiente é dinâmico e contínuo.

2.3 Planejamento de trajetória

O objetivo principal do planejamento de trajetória é a construção de algoritmos para automatizar o movimento de robôs, peças e outros ambientes que utilizam objetos geométricos arbitrários. Uma tarefa básica é mover um robô em seu ambiente de trabalho a partir de uma posição e orientação para outra posição e orientação desejada, sem o robô bater em obstáculos. O planejamento de trajetória tem aplicações tanto dentro, como fora da área de robótica.

O problema de navegação mais básico é mover um robô modelado como um ponto no espaço através de um ambiente bidimensional com vários itens proibidos (ou obstáculo). O obstáculo é considerado uma região em que o robô não pode entrar. O modelo é cinemático, o que significa que a única limitação é que o robô tem de se mover ao longo de uma trajetória contínua.

O robô é modelado como um ponto no espaço de configuração, em vez de espaço de

trabalho. Uma configuração é uma especificação do local (localização e orientação) do robô em relação ao meio ambiente e o espaço de configuração é o conjunto de todas as configurações possíveis (CHOSSET et al., 2005).

Realizar o planejamento apresenta um maior nível de complexidade quando o robô tem muitos graus de liberdade e orientação. Um robô guia pode passar por um espaço com uma certa posição e orientação, e atinge um obstáculo se fosse em uma orientação diferente. Isso aumenta a complexidade dos algoritmos para planejamento de trajetória. Além disso, podem existir obstáculos móveis no ambiente, aumentando ainda mais a complexidade da busca de uma trajetória eficiente, sem que haja colisão.

O exemplo clássico de planejamento de trajetória é o problema do mecanismo de enfileiramento (SCHWARTZ; SHARIR, 1983): mover um piano através de uma casa sem colidir com outros objetos na casa (ou a própria casa). Considerando somente a geometria dos objetos e não as forças que atuam sobre o piano, como a gravidade.

Estendendo a analogia, o piano é suposto ter motores infinitamente fortes e infinitamente pequenos. Portanto, o problema do motor de piano só lida a circulação de uma forma geométrica ao longo de um caminho que é contínuo no espaço através de uma atmosfera.

Existem diversos algoritmos que tratam da navegação de agentes autônomos, sendo que um dos primeiros é descrito no livro de Inteligência Artificial de Russell e Norvig, o algoritmo A* (RUSSELL; NORVIG, 2003). Ele busca o caminho em um grafo de um vértice inicial até um vértice final, é a combinação de aproximações heurísticas como do algoritmo *Best-first Search* e da formalidade do Algoritmo de Dijkstra (DIJKSTRA, 1959).

A partir do algoritmo A*, surgiram derivações como o R* (LIKHACHEV; STENTZ, 2008), que depende muito menos da qualidade da função heurística, evitando mínimos locais e resolvendo o problema de planejamento de toda uma série de pesquisas de curto alcance.

3 *Modelo de mapeamento e planejamento de trajetória*

O objetivo deste capítulo é descrever o modelo desenvolvido, apresentando uma descrição de todo o processo envolvido no desenvolvimento, especificando a arquitetura do modelo, a metodologia utilizada para desenvolver o projeto e os trabalhos correlatos.

3.1 **Trabalhos relacionados**

Visando identificar os principais trabalhos desenvolvidos para planejar trajetória, foram realizados alguns estudos no âmbito do planejamento de trajetória e predição de colisão para especificar melhor o domínio no qual o trabalho foi desenvolvido e realizar um estudo comparativo das diferenças do modelo proposto e os pesquisados. A pesquisa não se limitou ao domínio do futebol de robôs e à simulação.

Existem muitas abordagens e modelos para planejamento de trajetória, porém a maioria dos modelos foram desenvolvidos para serem utilizados em ambientes estáticos, e este modelo foi desenvolvido para ser utilizado em ambiente contínuo e não determinístico. Alguns modelos utilizam uma abordagem em cluster, onde utilizam um banco de dados para treinamento e aprendizagem com o objetivo de encontrar trajetórias (JETCHEV; TOUSSAINT, 2009), ou para realizar agrupamentos de trajetórias para identificação de padrões de movimento (SUNG; FELDMAN; RUS, 2012). Na simulação 3D não é viável e eficiente utilizar grandes quantidades de dados e ter tempos de processamento acima de 0.02 segundos, tempo máximo para o agente tomar uma decisão.

Guo et al. (2009) desenvolveu um modelo que capta as imagens e realiza um treinamento para busca do melhor caminho filtrando o erro entre a predição e o ambiente real. O mapeamento do modelo é realizado supondo áreas definidas como quadrados, onde o agente se movimenta de um por um, com o objetivo de evitar colisões.

Algumas abordagens híbridas para planejamento de trajetória utilizam uma combinação de planejamento de passo, algoritmo de busca e checagem de colisão (HAMASAKI et al., 2011)(HORNUNG et al., 2012). Estes modelos utilizam um mapa 2D composto por células de tamanho igual que são marcadas como livres ou ocupadas. Basicamente o modelo primeiro realiza um método recursivo de avaliação de colisão e depois utilizam o algoritmo de busca para encontrar o melhor caminho sem possibilidade de colisão.

O modelo proposto neste trabalho tem como base a busca de uma trajetória livre de colisão, realizando o mapeamento de possíveis trajetórias e verificando possíveis colisões a partir da predição da trajetória dos obstáculos móveis. Além de encontrar a melhor trajetória, foi utilizado uma combinação de um modelo de movimentação omnidirecional do BahiaRT, que já estava implementado, aumentando ainda mais a eficácia da navegação do agente no ambiente.

Por fim, o desenvolvimento e pesquisa de abordagens para ambientes dinâmicos ainda é considerado um grande desafio, pois requer modelos em tempo real, preditivos, com processamento de imagens e de padrões de comportamento eficientes e ao mesmo tempo que garantam a segurança dos envolvidos no ambiente. Por este motivo, o trabalho está sendo desenvolvido utilizando como domínio o futebol de robôs.

3.2 Proposta Metodológica

O desenvolvimento deste projeto foi dividido em 5 etapas apresentadas na figura 3.1.

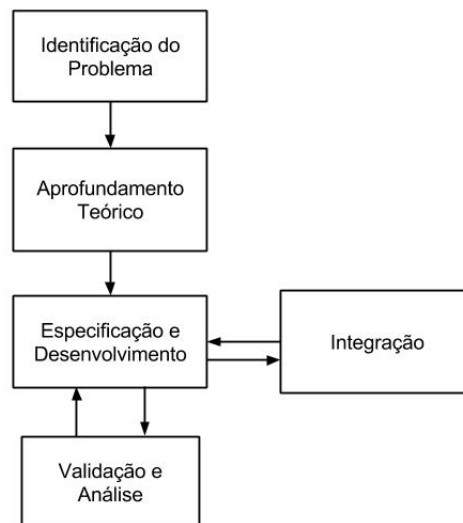


Figura 3.1: Etapas da metodologia do projeto.

Na etapa inicial, definida como "Identificação do Problema", foi realizado um estudo do ambiente simulado 3D, através da especificação PEAS(RUSSELL; NORVIG, 2003) no domínio de futebol de robôs para identificar quais fatores influenciam na tomada de decisão do agente na identificação da melhor trajetória a ser seguida até o objetivo sem que haja colisão.

Na etapa seguinte, "Aprofundamento Teórico", foi realizado um estudo sobre planejamento de trajetória, planejamento multi-agente, coordenação multi-agente, predição de colisão e mapeamento de ambientes dinâmicos. Esta etapa teve como objetivo identificar trabalhos relacionados ao problema identificado na etapa anterior. O resultado do estudo foi descrito na Seção 3.1.

A etapa de "Especificação e Desenvolvimento" teve como base o desenvolvimento do modelo a partir da especificação dos requisitos e da modelagem da solução utilizando o time BahiaRT. Em conjunto com essa etapa, foi realizada a etapa de "Integração", fazendo as modificações necessárias no código do time BahiaRT e integrando o modelo desenvolvido.

Finalmente, na etapa "Validação e Análise", foi realizada uma especificação da metodologia de teste, da instalação do ambiente de teste e executados os testes para validar o modelo desenvolvido. Onde a validação foi realizada utilizando uma aplicação de passe desenvolvido para avaliar a eficácia das trajetórias encontradas, rodando o time com os melhores times da liga de simulação 3D dos últimos anos. E nesta mesma etapa, foram coletados os resultados obtidos, retornando a etapa anterior para corrigir falhas de desenvolvimento.

3.3 Arquitetura do modelo

A arquitetura da aplicação de passe desenvolvida para validar o modelo proposto pode ser vista na figura 3.2.

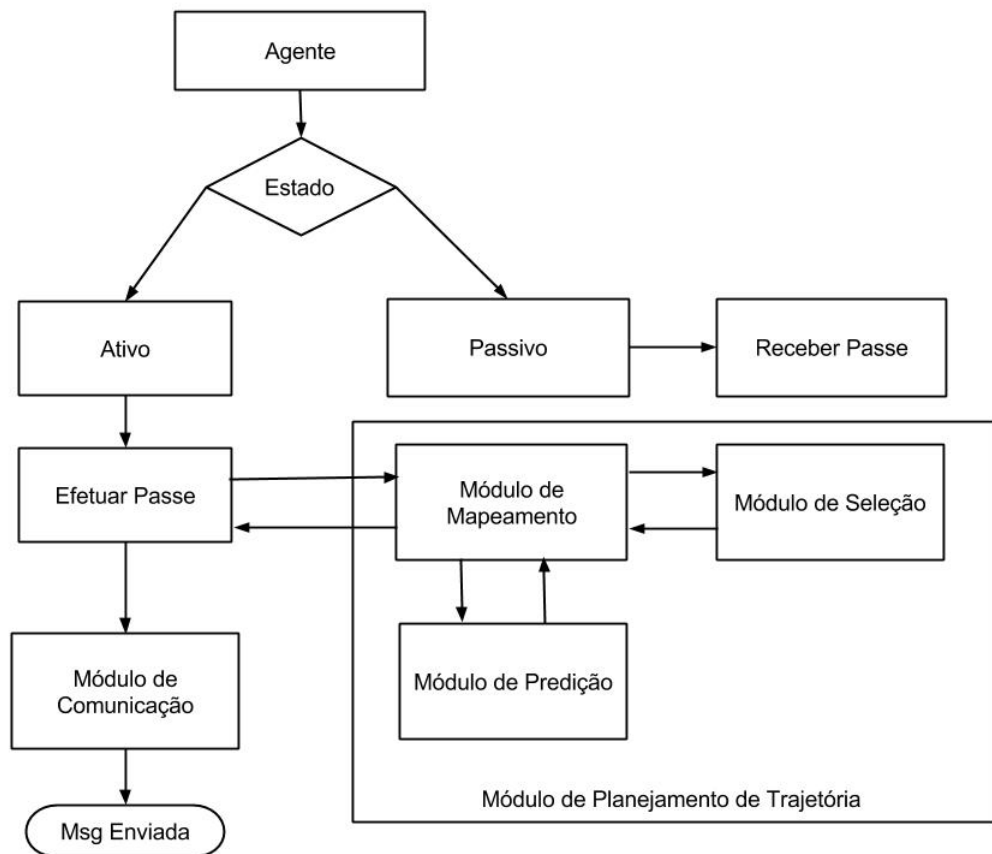


Figura 3.2: Arquitetura do projeto desenvolvido integrado a um modelo de passe desenvolvido para validação.

Nesta arquitetura, um agente pode estar em estado ativo, ou passivo. Um agente em estado ativo, é designado como um agente que está com posse de bola, ou lutando pela posse de bola. Já os agentes passivos são considerados agentes de suporte, que ficam em posições estratégicas. No momento em que um agente está em estado ativo e com posse de bola, ele verifica a possibilidade de efetuar um passe.

O passe só é realizado se obedecer as seguintes regras:

1. O oponente mais próximo deve estar a 4 metros de distância.
2. Deve existir pelo menos 1 agente aliado em uma posição estratégica.
3. O agente aliado que vai receber um passe não deve estar caído.
4. O agente aliado que vai receber um passe não deve estar marcado por agentes oponentes.

5. O agente que vai efetuar o passe deve estar em estado ativo e com a pose da bola.
6. Deve existir uma trajetória confiável para onde o agente vai chutar a bola.

Inicialmente é realizado o mapeamento e a busca das melhores trajetórias para lançar a bola. Depois é realizada uma busca dos melhores jogadores para receber o passe com base nas trajetórias. Em seguida o módulo de passe verifica dentre as trajetórias mapeadas e o posicionamento dos agentes aliados qual é a melhor trajetória para se obter sucesso no passe da bola.

Quando a trajetória é escolhida, o agente envia uma mensagem *broadcast* avisando a posição em que a bola vai ser lançada e quem é o agente aliado que deve se deslocar para receber o passe. Quando os agentes recebem a mensagem, eles verificam se um passe está sendo realizado e quem é o agente escolhido para receber a bola. Por fim, o escolhido se deslocará para a posição estratégica que lhe foi comunicada.

O sucesso do passe depende não somente da aplicação desenvolvida, mas também do chute que vai ser realizado, levando em consideração os fatores tempo (tempo que o agente leva para chutar) e força (força aplicada pelo chute sobre a bola).

3.4 Mapeamento das trajetórias

O mapeamento das trajetórias é realizado a partir da construção de um mapa de grade 2D, composto por células de tamanho igual que é projetado em segmentos em linhas k do centro da trajetória. A construção do mapa leva em consideração as limitações do campo e o posicionamento do agente que vai calcular a melhor trajetória para alcançar o seu objetivo.

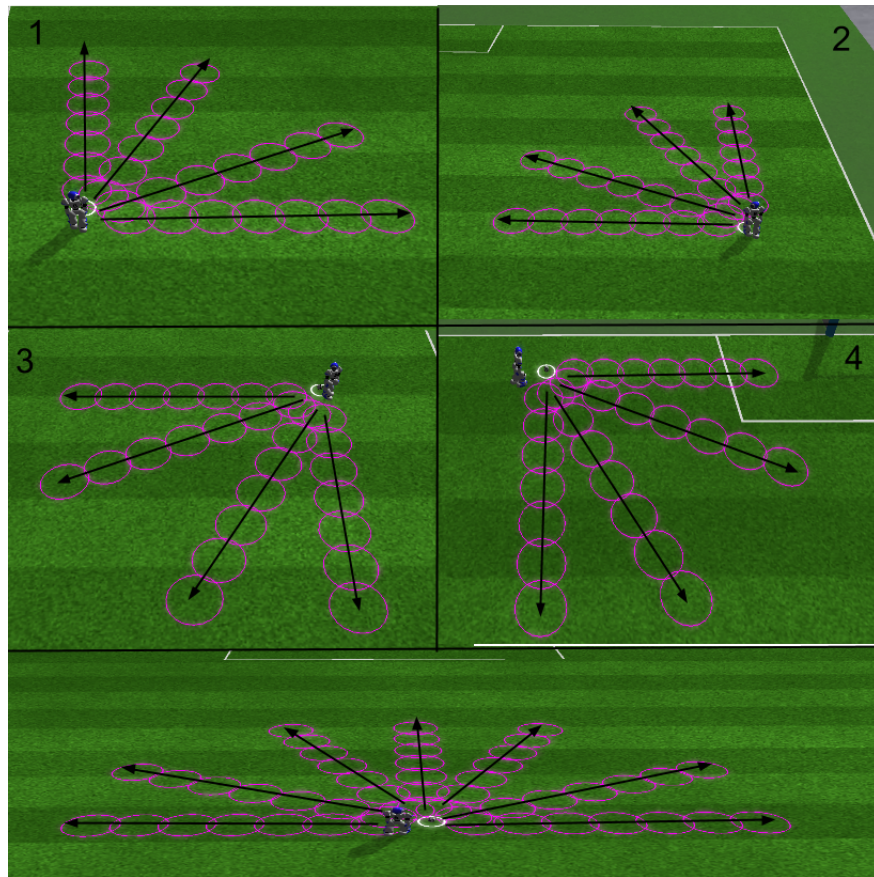


Figura 3.3: Imagem do plot realizado pelo Roboviz representando os 5 tipos de mapeamento que é realizado pelo modelo desenvolvido.

Para cada região do campo, foi definido um tipo de mapeamento, totalizando 5 tipos de mapeamento que são ilustrados na figura 3.3. O objetivo foi evitar que regiões que não estavam dentro do campo fossem mapeadas e a quantidade de células a serem verificadas fosse a menor possível, já que todo o processo de mapeamento e predição de trajetória deve ser feito em menos de 0.02 segundos, que é o tempo de processamento mínimo do agente simulado 3D utilizado para desenvolvimento do modelo.

Cada célula possui um conjunto de informações que são utilizadas no processo de predição da trajetória. No mapeamento de uma célula, apenas a posição e o número da célula são armazenados, sendo que o número representa a trajetória. As outras informações da célula são incluídas no processo de verificação da presença de obstáculos que será descrito na próxima seção.

O diagrama de classe da célula pode ser visto na figura 3.4.

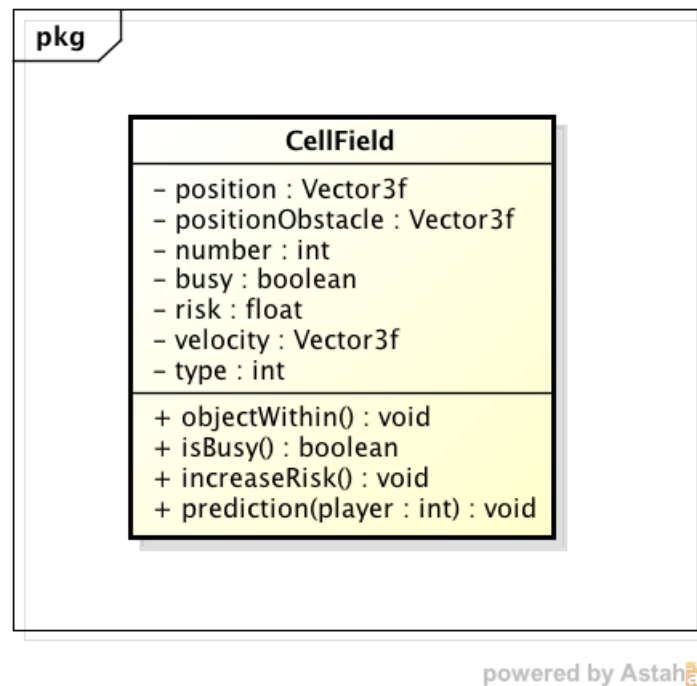


Figura 3.4: Diagrama da classe CellFied utilizado no modelo para representar uma unidade do mapeamento.

3.5 Verificação da presença de obstáculos

A verificação da presença de obstáculos é feita logo após o processo de mapeamento das trajetórias. O modelo levou em consideração que os agentes aliados não são considerados obstáculos por conta do processo de coordenação e planejamento multi-agente que é realizado através do protocolo de comunicação utilizado pelo BahiaRT. Neste processo de coordenação, quando o agente encontrar a melhor trajetória, ele enviará uma mensagem através do canal de comunicação para todos os agentes aliados, informando a trajetória e outras informações que fazem parte do seu estado de mundo. O objetivo é atualizar todos os agentes sobre o estado atual do agente ativo.

No processo de verificação da presença de obstáculos, algoritmo 1, se uma célula estiver ocupada por um obstáculo, as informações desse obstáculo serão armazenadas. Neste processo, a posição do obstáculo, a velocidade e o tipo de obstáculo são armazenados na célula. Como neste domínio, apenas os agentes oponentes são considerados obstáculos, só existirão obstáculos de um tipo, que são os oponentes. Além dessas informações, uma taxa que indica o risco da célula é incrementada para indicar posteriormente se a trajetória é viável.

Cada agente possui informações dos oponentes que são utilizadas para verificar a

Algorithm 1 Algoritmo para verificação da presença de obstáculos

```

para  $k \leftarrow 1$  até  $n$  faça
  para  $i \leftarrow 2$  até  $11$  faça
     $map[k].prediction(agent[i])$  ▷ Realiza a predição de colisão
    se  $!map[k].isBusy()$  então
      se  $map[k].objectWithin(agent[i].position)$  então
        se  $agente[i].conf > minReliability$  então
           $map[k].velocity \leftarrow agent[i].velocity$  ▷ Armazena a velocidade
           $map[k].positionOB \leftarrow agent[i].position$  ▷ Armazena a posição
           $map[k].type \leftarrow OPPONENT$  ▷ Armazena o tipo de obstáculo
           $map[k].busy \leftarrow true$  ▷ Define a célula como ocupada
           $map[k].IncreaseRisk()$  ▷ Incrementa o risco
        fim se
      fim se
    fim se
  fim para
fim para

```

presença dos obstáculos. Estas informações possuem uma taxa de confiabilidade que representa o quanto essas informações estão atualizadas. Neste processo de verificação, apenas os agentes oponentes que possuem no mínimo 80% de confiabilidade são levados em consideração.

3.6 Determinação da trajetória de objetos dinâmicos em ambientes ruidosos

Para calcular a velocidade e direção dos obstáculos foi desenvolvido um modelo que filtra o ruído posicional (oscilações fortes da posição do objeto visto em torno da sua trajetória real) e calcula a velocidade vetorial e a escalar do objeto com poucas oscilações.

Para isso, um vetor é estendido desde a posição filtrada em cada ciclo, para o futuro, para prever a posição futura do objeto. Ao invés de calcular a média das velocidades, foi utilizado uma função que utiliza mínimos quadrados (ROCHA, 2001) para encontrar um valor coerente com o movimento do objeto.

Seja n uma sequência ordenada de N valores numéricos n_i , $i = 1, 2, \dots, N$. Denotado por $S_{a,b}(n)$ o operador que extrai uma subsequência $s = \{n_a, n_{a+1}, \dots, n_{b-1}, n_b\}$ da sequência n , sempre que $a \leq b$. Sejam x_i e y_i as coordenadas medidas de um objeto rastreado num plano no instante t_i , $i \geq 0$. A figura 3.5 demonstra uma sequência de pontos medidos de um objeto monitorado pelo agente no campo de futebol, que foi localizado

quando estava parado na esquerda do gráfico e começou a se movimentar para a direita fazendo uma trajetória complexa, que inclui um giro de 360 graus.

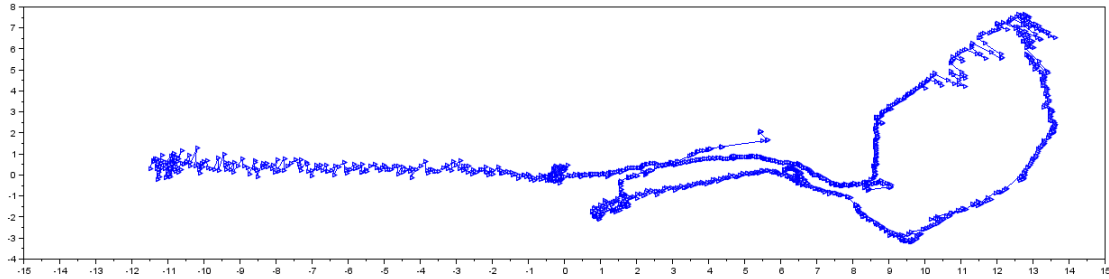


Figura 3.5: Sequência de pontos medidos de um objeto monitorado pelo agente no campo de futebol

Vale notar que independentemente do sistema sensorial utilizado, existirá uma diferença entre o ponto medido $p_i = (x_i, y_i)$ e o ponto onde realmente se encontra o objeto rastreado nesse instante $P_i = (X_i, Y_i)$. Em ambientes simulados é possível realizar experimentos e registrar as posições reais do alvo em intervalos de tempo discretos t_k para $k = 1, 2, \dots, i$, isto é, coletando a trajetória real e medida através de séries temporais $P = \{P_1, P_2, \dots, P_i\}$ e $p = \{p_1, p_2, \dots, p_i\}$. A figura 3.6 demonstra o resultado obtido em um dos experimentos, onde o objeto foi localizado na esquerda do gráfico e começou a se movimentar para a direita. A cor vermelha representa os pontos reais e a cor azul os ponto medidos.

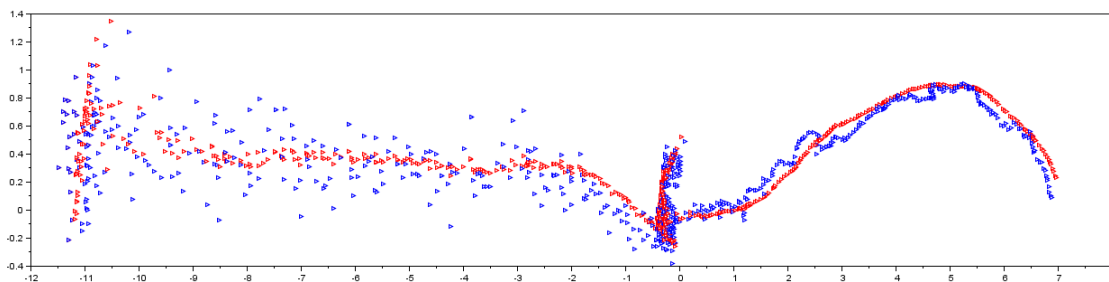


Figura 3.6: Sequência de pontos medidos e estimados de um objeto monitorado pelo agente no campo de futebol

Note que a distância entre a posição medida p_i e a posição real P_i do objeto, resultado do ruído inerente do sistema de medição, varia com o tempo o que indica que o sistema sensorial em teste é autoajustado, mas que mesmo assim nunca desaparece. A intensidade do ruído posicional instantâneo vem dada por $r_i = \|p_i - P_i\|$ onde o operador $\|\cdot\|$ representa a norma euclidiana do vetor argumento.

Desde o ponto de vista de filtragem de ruído, o resultado relevante são os últimos u instantes da trajetória, designados como trajetórias recentes no instante i , e denotados por $\Pi_i = S_{i-u,i}(P) = \{P_{i-u}, P_{i-u+1}, \dots, P_{i-1}, P_i\}$ e $\pi_i = S_{i-u,i}(p) = \{p_{i-u}, p_{i-u+1}, \dots, p_{i-1}, p_i\}$.

Considerando que qualquer trajetória pode ser aproximada por uma sucessão de segmentos de trajetórias lineares, foi utilizado o método dos mínimos quadrados para estimar uma aproximação mais confiável \tilde{P}_i da posição real do alvo no instante i , e com isto poder estimar sua direção e velocidade absoluta utilizando um ponto estimado no passado \tilde{P}_{i-lag} , com $lag \geq 1$. A hipótese principal do estudo é a base provada do método dos mínimos quadrados segundo o qual $\|p_i - P_i\| \geq \|\tilde{P}_i - P_i\|$, em média (ROCHA, 2001).

O procedimento utilizado foi o seguinte:

1. Obter a posição filtrada \tilde{P}_i através de duplo ajuste com mínimos quadrados supondo os resultados com as primeiras derivadas das funções obtidas em cada ajuste, isto é
 - (a) Cálculo dos coeficientes a_{xy}, b_{xy} da reta $y = a_{xy} + b_{xy}x$, utilizando mínimos quadrados com os últimos $u - 1$ pontos filtrados $(\tilde{x}_j, \tilde{y}_j)$, $j = i - u, \dots, i - 1$ e o ponto medido (x_i, y_i) .
 - (b) Cálculo dos coeficientes a_{yx}, b_{yx} da reta $x = a_{yx} + b_{yx}y$, utilizando mínimos quadrados com os últimos $u - 1$ pontos filtrados $(\tilde{y}_j, \tilde{x}_j)$, $j = i - u, \dots, i - 1$ e o ponto medido (y_i, x_i) .
 - (c) Cálculo de dois pontos no presente (instante i), $p_{1,i} = (x_{1,i}, y_{1,i})$ e $p_{2,i} = (x_{2,i}, y_{2,i})$, da forma

$$x_{1,i} = x_i$$

$$y_{1,i} = a_{yx} + b_{yx}x_i$$

$$x_{2,i} = a_{xy} + b_{xy}y_i$$

$$y_{2,i} = y_i$$

- (d) Cálculo do ponto mais provável no presente (instante i) $\tilde{P}_i = (\tilde{x}_i, \tilde{y}_i)$ a partir dos 2 pontos acima, utilizando como peso a primeira derivada, da forma

$$\tilde{P}_i = \left(\frac{1}{b_{xy} + b_{yx}} \right) (b_{yx}p_{1,i} + b_{xy}p_{2,i})$$

2. Obter a direção de movimento mais provável e a velocidade absoluta do objeto utilizando novamente mínimos quadrados e diferenças finitas, isto é

- (a) Cálculo dos coeficientes a_{xy}, b_{xy} da reta $y = a_{xy} + b_{xy}x$, utilizando mínimos quadrados com os últimos u pontos filtrados $(\tilde{x}_j, \tilde{y}_j)$, $j = i - lag, \dots i$.
- (b) Cálculo dos coeficientes a_{yx}, b_{yx} da reta $x = a_{yx} + b_{yx}y$, utilizando mínimos quadrados com os últimos u pontos filtrados $(\tilde{y}_j, \tilde{x}_j)$, $j = i - lag, \dots i$.
- (c) Cálculo de dois pontos no passado (instante $i - lag + 1$), $p_{1,i-lag+1} = (x_{1,i-lag+1}, y_{1,i-lag+1})$ e $p_{2,i-lag+1} = (x_{2,i-lag+1}, y_{2,i-lag+1})$, da forma

$$x_{1,i-lag+1} = \tilde{x}_{i-lag+1}$$

$$y_{1,i-lag+1} = a_{yx} + b_{yx}x_{1,i-lag+1}$$

$$y_{2,i-lag+1} = \tilde{y}_{i-lag+1}$$

$$x_{2,i-lag+1} = a_{xy} + b_{xy}y_{2,i-lag+1}$$

- (d) Cálculo do ponto médio no passado (instante $i - lag + 1$) $\hat{P}_{i-lag+1}$ a partir dos 2 pontos no passado calculados acima, isto é,

$$\hat{P}_{i-lag+1} = 0.5(p_{1,i-lag+1} + p_{2,i-lag+1})$$

- (e) Cálculo de dois pontos no presente (instante i), $p_{1,i} = (x_{1,i}, y_{1,i})$ e $p_{2,i} = (x_{2,i}, y_{2,i})$, da forma

$$x_{1,i} = \tilde{x}_i$$

$$y_{1,i} = a_{yx} + b_{yx}x_{1,i}$$

$$y_{2,i} = \tilde{y}_i$$

$$x_{2,i} = a_{xy} + b_{xy}y_{2,i}$$

- (f) Cálculo do ponto médio no presente (instante i) \hat{P}_i a partir dos 2 pontos no presente calculados acima, isto é,

$$\hat{P}_i = 0.5(p_{1,i} + p_{2,i})$$

- (g) Cálculo do vetor velocidade no presente (instante i), $v_i = (v_{x,i}, v_{y,i})$, tal

$$v_i = \frac{\hat{P}_i - \hat{P}_{i-lag+1}}{t_i - t_{i-lag+1}}$$

- (h) Cálculo da velocidade escalar no presente (instante i), $V_i = \|v_i\|$.

- (i) Cálculo do vetor de direção no presente (instante i), $d_i = \frac{1}{\|v_i\|}v_i$.

Com o algoritmo para filtrar o ruído posicional e calcular a velocidade vetorial e escalar do agente, foi possível estimar os pontos de colisão com maior precisão, auxiliando na escolha da melhor trajetória.

3.7 Predição de colisão dos obstáculos

Neste modelo, além de levar em consideração a presença de obstáculos nas trajetórias, é realizado um processo de predição de deslocamento dos obstáculos que estão em regiões próximas com o objetivo de prever se o obstáculo irá estar na trajetória no mesmo instante em que o agente estiver se deslocando para o ponto objetivo, ou se tiver chutado a bola na trajetória escolhida e ela estiver se deslocando até o ponto objetivo.

O principal objetivo da predição é evitar que o agente ou a bola colida com o obstáculo no momento em que estiver se deslocando em direção ao objetivo. A partir desta análise, se um agente oponente levar um tempo menor para chegar ao ponto P da trajetória, que o agente aliado, ou a bola, então o risco deste ponto aumenta, sendo que cada ponto é considerado como a célula da trajetória, e que para cada célula, são verificados os agentes oponentes próximos desta trajetória que foi mapeada.

O calculo da predição do deslocamento do agente oponente é realizado com a velocidade do agente calculada e a distância entre o oponente e a trajetória:

$$D = A - B \quad (3.1)$$

$$T = \frac{D}{V} \quad (3.2)$$

onde D é a distância do ponto A ao ponto B , V é a velocidade do oponente e T é o tempo que o agente levaria até chegar ao ponto B , levando em consideração que a velocidade é constante.

O cálculo do ponto previsto é descrito como:

$$P = Po + V * T \quad (3.3)$$

onde P é o ponto previsto, Po é a posição atual, V é a velocidade e T é o tempo em que estou estimando a posição. O algoritmo resumido pode ser visto abaixo no algoritmo 2.

Algorithm 2 Algoritmo para predição de colisão

```

função PREDICTION(agent)
  se agent.conf < minReliability então
    devolve
  fim se
  ballTime  $\leftarrow$  distanceBallToCellField/velBall
  meTime  $\leftarrow$  distanceMeToCellField/velMe
  deltaS  $\leftarrow$  agent.distanceToCellField
  agentSpeed  $\leftarrow$  agent.velocity
  se agentSpeed <= 0 então
    agentSpeed  $\leftarrow$  0.1
  fim se
  deltaT  $\leftarrow$  deltaS/agentSpeed
  estimatedPostition  $\leftarrow$  agent.position + agent.velocity * deltaT
  se objectWithin(estimatedPostition) então
    se ((deltaT + TIME_PERCEPTION) < ballTime) || (deltaT +
    TIME_PERCEPTION < meTime) então
      increaseRisk()
    fim se
    se (maxVelocityAgent - agentSpeed) < minVel então
      increaseRisk()
      increaseRisk()
    senão
      increaseRisk()
    fim se
  fim se
fim função

```

3.8 Critério do descarte de obstáculos

O modelo levou em consideração apenas os obstáculos que possuíam informações atualizadas e que estavam próximos da trajetória. O objetivo foi evitar utilizar obstáculos que não possuem nenhum risco para a predição da trajetória.

4 Validação e Resultados Obtidos

O objetivo deste Capítulo é descrever todo o processo de coleta de indicadores e validação do modelo desenvolvido. Este Capítulo está dividido em 4 Seções.

A Seção 4.1 descreve a metodologia de teste utilizada, os componentes utilizados no ambiente de teste e os *frameworks* utilizados para realizar os testes do modelo.

Na Seção 4.3 são descritos os cenários definidos como casos de teste para realizar a validação do modelo.

A Seção 4.4 descreve quais foram as métricas utilizadas para validar o modelo.

Por fim, a Seção ?? descreve os resultados obtidos no processo de validação do modelo, as considerações e a importância dos resultados obtidos com o desenvolvimento do modelo.

4.1 Metodologia de Teste

A realização dos testes se deu através de um ambiente real de competição. O ambiente foi montado utilizando 4 computadores conectadas através de um *switch*. As configurações dos componentes utilizados pode ser vista na tabela 4.1.

Tabela 4.1: Configuração dos computadores utilizados no ambiente de teste para coleta dos resultados.

Programa utilizado	Configuração do componente
Time BahiaRT	Notebook DELL Core i7, RAM-8GB, HD-1TB , Placa de vídeo - 2GB
Time Oponente	Notebook DELL Core i7, RAM-8GB, HD-1TB , Placa de vídeo - 2GB
Simspark	Notebook DELL Core i7, RAM-8GB, HD-1TB , Placa de vídeo - 2GB
Roboviz	Notebook DELL Core i7, RAM-8GB, HD-1TB , Placa de vídeo - 2GB

Para realizar a validação do modelo, foi necessário instalar e configurar o Simspark e o RoboViz, que são plataformas padrões para a simulação 3D do futebol de robôs. O servidor Simspark foi instalado em um computador e o RoboViz que é onde a simulação do ambiente é renderizada, foi instalada em outro computador. Os outros 2 computadores,

que são as máquinas clientes onde os binários dos times foram executados, são máquinas que possuíam apenas o sistema operacional instalado, que neste caso foi utilizado o *Ubuntu 12.04 LTS*. A arquitetura do ambiente pode ser vista através da figura 4.1.

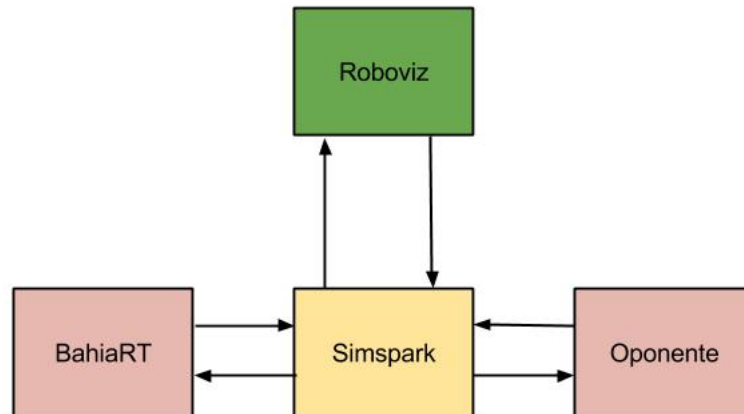


Figura 4.1: Arquitetura do ambiente de teste utilizado para coletar os resultados.

A validação se deu através de dois processos. No primeiro processo, partidas foram realizadas utilizando 4 times da liga de simulação que participaram da RoboCup 2013 e o BahiaRT:

1. UTAustinVilla
2. ITAndroids
3. FCPortugal
4. Apollo3D

O objetivo do primeiro processo foi verificar a qualidade do modelo desenvolvido utilizando a predição das trajetórias dos obstáculos com capacidade de deliberação e a escolha da melhor trajetória para efetuar o passe com base no mapeamento dos obstáculos e na predição dos pontos de colisão de acordo com a velocidade e direção do agente adversário. Esta validação se deu através da análise de situações onde o passe ocorria verificando as métricas de avaliação que foram pré-definidas.

No segundo processo, foi utilizado uma ferramenta para automatizar testes de cenários reais de uma partida de futebol de robôs pré-definidas utilizando agentes Dummy (agente sem capacidade de deliberação, que não se movem).

4.2 Trainer3D

Com o objetivo de automatizar os testes realizados com os agentes da simulação 3D de futebol de robôs, uma ferramenta chamada Trainer3D foi desenvolvida. Esta ferramenta visa facilitar a coleta dos indicadores de desempenho e também ser utilizada no processo de otimização dos comportamentos, como andar ou chutar.

A arquitetura do Trainer3D pode ser vista na figura 4.2.

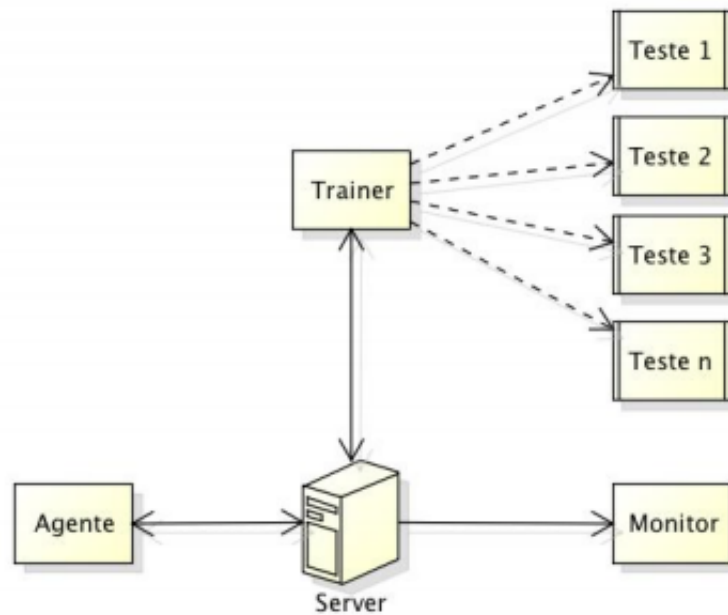


Figura 4.2: Arquitetura da ferramenta Trainer3D.

Quando o Trainer3D é instanciado, ele gera uma comunicação com o servidor SimS-park através de um socket com protocolo TCP/IP que capta as mensagens do estado do agente, bola e campo. A partir da comunicação do trainer com o servidor, as mensagens que o servidor envia são quebradas em tokens que contém informações do ambiente que está sendo simulado e que são renderizadas pelo RoboViz.

Além de captar as mensagens do servidor, o trainer também possui a característica de mudar o estado do ambiente. Para mudar o estado do ambiente, mensagens específicas são enviadas ao servidor. Cada mensagem para alterar o estado de um objeto no ambiente deve conter um identificador que indica qual objeto deve ser atualizado.

As mensagens que o Trainer3D recebe possuem informações como posicionamento da bola e dos agentes que estão no campo. Com essas informações, é possível fazer uma análise de como o agente está atuando, se está caído, qual a velocidade média de deslocamento do agente, se a bola foi chutada, e outras informações.

Com o desenvolvimento da ferramenta, foi possível coletar os resultados dos testes realizados com os cenários pré-definidos de forma automatizada, sem a necessidade de interferência humana.

4.3 Cenários de Teste

O objetivo dos cenários de teste foi diversificar as situações em que o agente (ou robô) pudesse utilizar o modelo para encontrar a melhor trajetória para realizar o seu objetivo, que no caso foi definido como efetuar um passe para um outro agente aliado. Um passe no futebol é fazer com que um agente A chute a bola para um agente B , de modo que o agente B possa receber a bola em uma posição estratégica.

Os cenários foram divididos em níveis de dificuldade. O nível de dificuldade foi definido como a quantidade de agentes oponentes, que são considerados obstáculos móveis que podem comprometer a realização do objetivo do agente que vai efetuar o passe da bola. Os níveis foram mapeados de acordo com a tabela 4.2.

Tabela 4.2: Nível A é considerado fácil, o B é considerado moderado, C é considerado difícil, e o D é considerado muito difícil.

Quantidade de oponentes	Nível de dificuldade
2	A
3	B
4	C
4+	D

Na figura 4.3, são demonstradas as situações escolhidas para realizar os testes utilizando agentes oponentes que não se movem. Nestas situações, o objetivo foi verificar se o agente estava escolhendo a melhor trajetória para efetuar o passe e o executando corretamente.

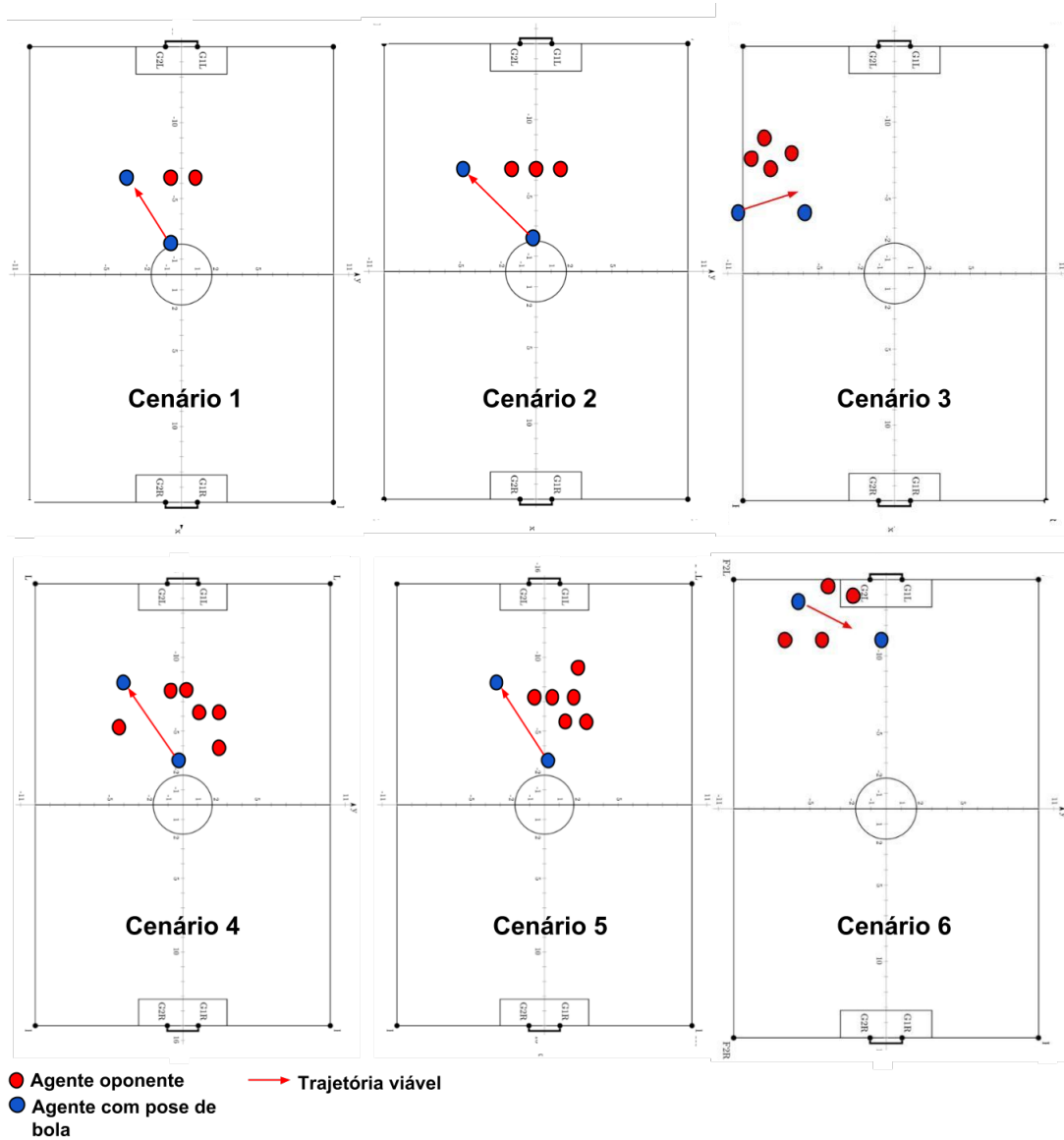


Figura 4.3: Cenários de situações de um jogo real de futebol onde um agente tem a possibilidade de efetuar um passe para outro agente aliado em uma posição estratégica.

4.4 Métricas de Avaliação

Os indicadores escolhidos para validar o modelo proposto foram mapeados levando em consideração aspectos que não são influenciados por problemas que não estão dentro do contexto do problema em questão, que é o planejamento de trajetória em tempo real.

Para o primeiro processo de validação, foram definidas 5 métricas:

1. Quantidade total de passes
2. Quantidade de passes corretos em kickin

3. Quantidade de passes corretos em `goal_kik`
4. Quantidade de passes corretos em `play_on`
5. Quantidade de passes errados

Para o segundo processo de validação, a métrica utilizada para validar o modelo foi a escolha da trajetória com base no cenário de teste executado.

Os testes executados que não estavam de acordo com a especificação da metodologia adotada, foram desconsiderados. O objetivo foi ter resultados bem definidos, que não foram afetados por outros problemas que ainda são lacunas a serem resolvidas em trabalhos posteriores.

4.5 Resultados

O objetivo dessa seção é descrever os resultados obtidos no processo de validação através do ambiente real de jogo e de cenários pré-definidos. Todos os testes foram realizados utilizando o ambiente descrito na figura 4.1. A análise dos resultados do primeiro processo de validação foi feita através do log gerado da partida pelo RoboViz, onde é possível reproduzir todo o jogo, parando, voltando e adiantando quando necessário.

4.5.1 Resultados de partidas utilizando cenário de jogo real

Inicialmente foram realizadas 10 partidas com alguns times da liga de simulação com duração de 10 minutos cada partida. Os resultados preliminares demonstrados na tabela 4.3, mostraram que em mais de 40% das tentativas de passe, onde o agente tinha condições favoráveis, o mesmo era cancelado por conta da demora para se posicionar e realizar o chute, e da aproximação do agente adversário. Em alguns testes, quando o chute da bola alcançava uma distância muito curta, os agentes oponentes conseguiam ter a pose de bola.

Tabela 4.3: Tabela com o resultados preliminares dos testes realizado com times da liga de simulação 3D

Time	Apollo3D	UTAustinVilla	FCPortugal	ITAndroids
Quantidade de passes	57	54	35	105
Quantidade de passes corretos	8.77%	18.52%	31.43%	38.10%
Quantidade de passes que terminaram em gol	0.00%	0.00%	0.00%	2.86%
Quantidade de passes errados que terminaram em gol	3.51%	0.00%	0.00%	0.00%
Quantidade de passes errados com pose de bola do adversário	8.77%	11.11%	15.24%	15.24%
Quantidade de passes desistidos	78.95%	70.37%	43.81%	43.81%

Após a conclusão do projeto, foram realizadas 5 partidas com duração de 10 minutos para cada time. Os resultados obtidos no processo de validação final utilizando o cenário de jogo real demonstraram uma melhora significativa na quantidade de passes realizados com sucesso. Na tabela 4.4 são demonstrados os resultados obtidos.

Tabela 4.4: Tabela com o resultado dos testes realizado com times da liga de simulação 3D

Time	Mithras3d	UTAustinVilla	RoboCanes	Hfutengine
Quantidade total de passes	15	4	6	4
Quantidade de passes corretos em kickin	0	2	4	2
Quantidade de passes corretos em goal_kick	0	2	2	0
Quantidade de passes corretos em play_on	15	0	0	2
Quantidade de passes errados	0	0	0	0

Os resultados obtidos demonstraram que em situações de bola parada, *kickin*, *goal_kick*, onde o agente possui mais tempo para se posicionar e chutar a bola, os passes tiveram 100% de acerto. Porém, em situações onde a bola está em jogo, *play_on*, sendo disputada por outros jogadores, ficou evidente que o principal fator responsável por permitir que o agente consiga efetuar mais passes é a velocidade com que o chute é executado. Contra o time Mithras3d, onde seus agentes possuía movimentação mais lenta, o agente conseguiu efetuar mais passes em modo *play_on*. Porém, contra o RoboCanes e UTAustinVilla, o agente não conseguiu efetuar passes em modo *play_on* devido a velocidade rápida da movimentação dos seus agentes. Ainda assim, nas situações de bola parada, os passes realizados permitiram ao BahiaRT obter vantagem estratégica.

Tabela 4.5: Tabela com o resultado das partidas realizadas no processo de validação

Time	Vitórias	Empates	Derrotas	Gols Marcados	Gols Sofridos	Saldo de Gols
Mithras3d	5	0	0	30	0	30
UTAAustinVilla	0	0	5	0	13	-13
RoboCanes	1	3	1	4	4	0
Hfutengine	5	0	0	22	0	22

Na tabela 4.5 são mostrados os resultados das partidas realizadas no processo de validação. Com o modelo de passe, o BahiaRT conseguiu obter maior aproveitamento, vencendo a maioria das partidas realizadas. Nos jogos realizados contra o time UTAAustinVilla, o BahiaRT perdeu todas as partidas por conta do super chute realizado pelo time oponente. Tais gols não puderam ser evitados, pois foram realizados em situações de bola parada, nas laterais e no meio de campo. Já contra o Mithras3d e Hfutengine, o BahiaRT conseguiu obter maior aproveitamento de gols. Por último, o RoboCanes foi o time mais equilibrado, com forte marcação na defesa, dificultando as chances de passe e marcação de gols.

4.5.2 Resultados utilizando cenários pré-definidos

Os resultados obtidos no segundo processo de validação, onde foram realizados 20 testes com situações pré-definidas demonstraram a eficácia do modelo desenvolvido em ambientes estáticos. Em todos os casos, o agente escolheu a melhor trajetória para efetuar o passe com base na segurança da trajetória e no posicionamento do agente aliado que recebeu o passe. Além disso, ficou evidente que independente do nível de dificuldade descrito na tabela 4.2, o agente consegue mapear e escolher a melhor trajetória.

Para demonstrar os resultados, foram plotados os pontos em que os agentes aliados e oponentes estavam posicionados e a trajetória escolhida para realizar o passe utilizando o modelo desenvolvido. Foram plotadas as trajetórias na cor verde através de retas entre os pontos inicial e final, aliados com pontos em preto e oponentes em cor vermelha. Os resultados podem ser vistos nas figuras 4.4 e 4.5.

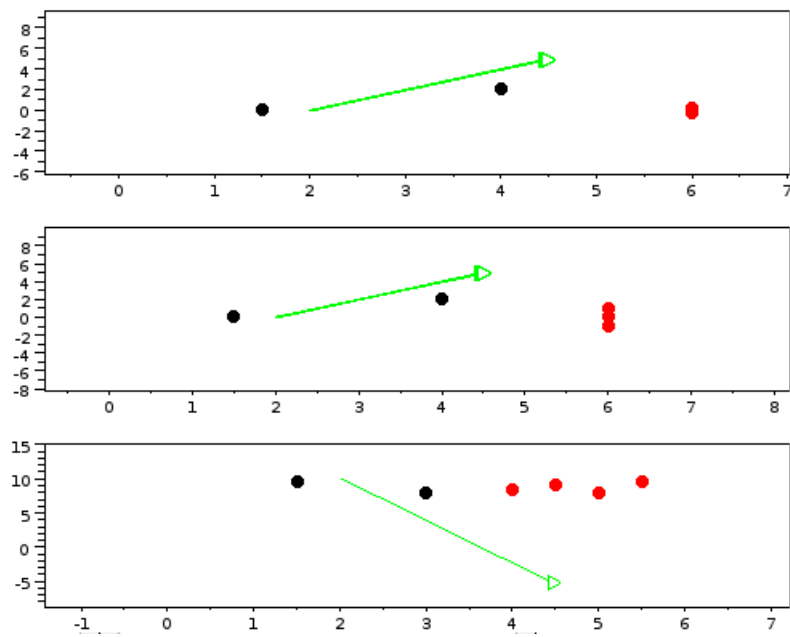


Figura 4.4: Resultados dos testes realizados com agentes Dummy nos cenários pre-definidos.

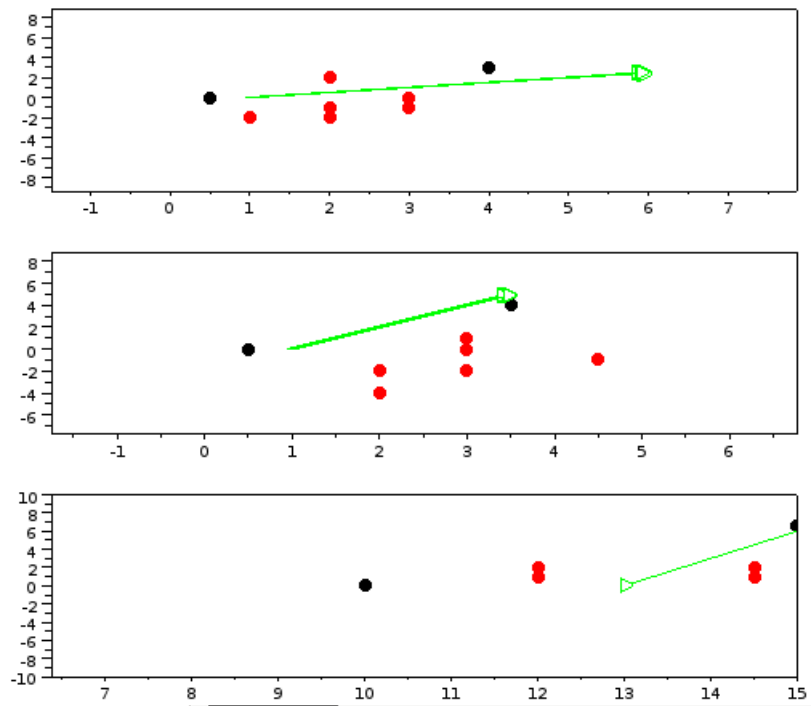


Figura 4.5: Resultados dos testes realizados com agentes Dummy nos cenários pre-definidos.

5 *Considerações Finais*

A construção de agentes bípedes autônomos não é uma tarefa trivial. Além do comportamento de baixo nível que é representado por movimentos como andar, levantar os braços, chutar uma bola, é necessário desenvolver o comportamento de alto nível, onde o agente deve analisar as informações que possui, e a partir delas escolher a melhor ação a ser executada.

Este trabalho apresentou o desenvolvimento de um modelo para realizar o mapeamento e planejamento de trajetórias em tempo real para navegação de agentes humanóides em ambiente simulado 3D. O objetivo do trabalho foi ter um modelo que possa ser utilizado no desenvolvimento de aplicações que requerem uma resposta rápida e ao mesmo tempo segura para guiar um agente através de um ambiente dinâmico. Como exemplo, podemos citar o resgate de pessoas em acidentes de desabamento, um agente inteligente deve ser capaz de se locomover no meio de destroços de concreto e buscar os melhores caminhos para chegar até a vítima de forma segura. Outra aplicação possível é a construção de robôs de serviços que precisam navegar em ambientes onde prestarão serviços diversos aos humanos de forma segura e eficaz.

O modelo foi desenvolvido utilizando a linguagem C++, que é utilizada pelo time BahiaRT. Além disso, outras ferramentas como o Scilab, foi utilizada no desenvolvimento do algoritmo descrito na seção 3.6.

Uma das contribuições trabalho é o Trainer3D. Esta ferramenta poderá ser utilizada por outros times da liga de simulação 3D para realizar testes, coletar indicadores de desempenho e automatizar o processo de otimização.

A validação do modelo se deu através do desenvolvimento de uma aplicação de passe. Esta aplicação foi escolhida para aumentar o nível estratégico do time BahiaRT, proporcionando jogadas que aumentem as chances de gol. Além disso, uma das principais motivações foi o principal campeonato mundial de futebol de robôs, a RoboCup.

Como trabalho futuro, a resolução dos problemas identificados no desenvolvimento da

aplicação de passe se viu necessária. Estas influenciam diretamente a execução do passe no futebol robótico. Dentre os trabalhos, o desenvolvimento de um chute dinâmico é um trabalho imprescindível, pois engloba o rápido posicionamento para realizar um chute para qualquer direção.

Referências Bibliográficas

BLUE, D. *Deep Blue*. 1997. Disponível em: <<http://www.research.ibm.com/deepblue/>>. Acesso em: 20 de abril de 2014.

CHOSSET, H. et al. Principles of robot motion: Theory, algorithms, and implementations. *PRESENCE-CAMBRIDGE MASSACHUSETTS*-, THE MIT PRESS, 2005.

DARPA. *Crashes and Traffic Jams in Military Test of Robotic Vehicles*. 2002. Defense Advanced Research Projects Agency.

DUDEK, G.; JENKIN, M. *Computational Principles of Mobile Robotics*. 2. ed. [S.l.: s.n.], 2010.

FLAMENCOS, R. *Tecnologias do Imaginário*. 1998. Disponível em: <<http://www.research.ibm.com/deepblue/>>. Acesso em: 20 de abril de 2014.

GLASER, S.; DORER, K. Trunk controlled motion framework. In *Proceedings of 8th Workshop on Humanoid Soccer Robots*, 13th IEEE-RAS International Conference on Humanoid Robots(Humanoids), Atlanta, 2013.

GUO, Y. et al. A combination of terrain prediction and correction for search and rescue robot autonomous navigation. *International Journal of Advanced Robotic Systems*, v. 6, n. 3, p. 207–214, 2009.

HAMASAKI, S. et al. Prediction of human’s movement for collision avoidance of mobile robot. In: *Robotics and Biomimetics (ROBIO), 2011 IEEE International Conference on*. [S.l.: s.n.], 2011. p. 1633–1638.

HORNUNG, A. et al. Anytime search-based footstep planning with suboptimality bounds. In: *Proceedings of the IEEE-RAS International Conference on Humanoid Robots*. [S.l.: s.n.], 2012.

JETCHEV, N.; TOUSSAINT, M. Trajectory prediction: Learning to map situations to robot trajectories. In: *Proceedings of the 26th Annual International Conference on Machine Learning*. New York, NY, USA: ACM, 2009. (ICML '09), p. 449–456. ISBN 978-1-60558-516-1. Disponível em: <<http://doi.acm.org/10.1145/1553374.1553433>>.

KITANO, H. *RoboCup-Rescue: Search and Rescue for Large Scale Disasters as a Domain for Multi-Agent Research*. 1. ed. [S.l.]: 1999, 1999.

KITANO, H. et al. *Robocup: The Robot World Cup Initiative*. 1. ed. [S.l.]: 1995, 1995. 19-24 p.

KITANO, H. et al. *The RoboCup Synthetic Agents Challenge*. 1. ed. [S.l.]: 1997, 1997.

- LAU, N. et al. Fcportugal 3d team : Team description paper. *Robocup Soccer Simulation League*, 2013.
- LIKHACHEV, M.; STENTZ, A. R* search. In: *Proceedings of the Twenty-Third AAAI Conference on Artificial Intelligence*. [S.l.: s.n.], 2008.
- LIU, J. et al. Apollo3d team description paper. *Robocup Soccer Simulation League*, 2013.
- MACALPINE, P. et al. UT Austin Villa: RoboCup 2012 3D simulation league champion. In: CHEN, X. et al. (Ed.). *RoboCup-2012: Robot Soccer World Cup XVI*. Berlin: Springer Verlag, 2013, (Lecture Notes in Artificial Intelligence).
- MAGMAOFFENBURG. *Team MagmaOffenburg*. 2014. Disponível em: <<http://robocup.hs-offenburg.de/html/index.htm>>. Acesso em: 24 de abril de 2014.
- REIS, L. P. How to win in robotic? strategy, cooperation and communication to build an artificial football team. In: *WIAA 2001*. [S.l.: s.n.], 2001.
- REIS, L. P. *Coordenação em Sistemas Multi-Agente: Aplicações na Gestão Universitária e Futebol Robótico*. Tese (Doutorado) — Universidade do Porto, Portugal, Junho 2003.
- RESCUE. *RoboCup Rescue Oficial Page*. 2001. Disponível em: <<Http://www.r.cs.kobe-u.ac.jp/robocup-rescue/>>. Acesso em: 20 de abril de 2014.
- ROBOCUP. *3D Simulation League*. 2012. Disponível em: <http://wiki.robocup.org/wiki/Soccer_Simulation_League>. Acesso em: 30 de março de 2014.
- ROBOCUP. *Simspark*. 2012. Disponível em: <<http://simspark.sourceforge.net/wiki/>>. Acesso em: 12 de abril de 2014.
- ROBOCUP, F. *The Goals of RoboCup*. 2001. Disponível em: <<Http://www.robocup.org/overview/22.html>>. Acesso em: 20 de abril de 2014.
- ROBOVIZ. *Roboviz*. 2012. Disponível em: <<https://sites.google.com/site/umroviz/features>>.
- ROCHA, C. D. A. *Algoritmo recursivo dos mínimos quadrados para regressão linear local*. Universidade do Porto. Reitoria, 2001. Disponível em: <<http://hdl.handle.net/10216/9587>>.
- RUSSELL, S.; NORVIG, P. *Informed earch and exploration*. [S.l.: s.n.], 2003. 97-104 p.
- SCHWARTZ, J.; SHARIR, M. *On the piano movers' problem: I. The case of a two-dimensional rigid polygonal body moving amidst polygonal barriers*. [S.l.: s.n.], 1983. 345-398 p.
- SIEGWART, R.; NOURBAKHS, R. *Introduction to Autonomous Mobile Robots*. 1. ed. [S.l.]: 2004, 2004.
- SKLAR, E.; EGUCHI, A.; JOHNSON, J. *RoboCupJunior: Learning with Educational Robotics*. 12. ed. [S.l.]: 2003, 2003. 29-34 p.

STOECKER, J.; VISSER, U. Roboviz programmable visualization for simulated soccer. In: ROFER, T. et al. (Ed.). *RoboCup 2011: Robot Soccer World Cup XV*. [S.l.]: Springer Berlin Heidelberg, 2012, (Lecture Notes in Computer Science, v. 7416). p. 282–293. ISBN 978-3-642-32059-0.

STONE, P.; VELOSO, M. *Multi-Agent Systems: A Survey from a Machine Learning Perspective*. 1. ed. [S.l.]: 1996, 1996.

SUNG, C.; FELDMAN, D.; RUS, D. Trajectory clustering for motion prediction. In: *IROS*. [S.l.: s.n.], 2012. p. 1547–1552.