

UNIVERSIDADE FEDERAL DE JUIZ DE FORA  
INSTITUTO DE CIÊNCIAS EXATAS  
BACHARELADO EM CIÊNCIA DA COMPUTAÇÃO

# IMPLEMENTAÇÃO DO SIMULADOR SIM PARA FÚTEBOL DE ROBÔS

Stephanie Martins Pinto da Costa

JUIZ DE FORA  
FEVEREIRO, 2014

# IMPLEMENTAÇÃO DO SIMULADOR SIM PARA FUTEBOL DE ROBÔS

STEPHANIE MARTINS PINTO DA COSTA

Universidade Federal de Juiz de Fora  
Instituto de Ciências Exatas  
Departamento de Ciência da Computação  
Bacharelado em Ciência da Computação

Orientador: Saul de Castro Leite

JUIZ DE FORA  
FEVEREIRO, 2014

# IMPLEMENTAÇÃO DO SIMULADOR SIM PARA FUTEBOL DE ROBÔS

Stephanie Martins Pinto da Costa

MONOGRAFIA SUBMETIDA AO CORPO DOCENTE DO INSTITUTO DE CIÊNCIAS EXATAS DA UNIVERSIDADE FEDERAL DE JUIZ DE FORA, COMO PARTE INTEGRANTE DOS REQUISITOS NECESSÁRIOS PARA A OBTENÇÃO DO GRAU DE BACHAREL EM CIÊNCIA DA COMPUTAÇÃO.

Aprovada por:

---

Saul de Castro Leite  
Prof. Dr. em Modelagem Computacional

---

Saulo Moraes Villela  
Prof. Dr. em Engenharia de Sistemas e Computação

---

Luciana Conceição Dias Campos  
Prof. Dr. em Engenharia Elétrica

JUIZ DE FORA  
12 DE FEVEREIRO, 2014

## Resumo

O futebol de robôs é uma iniciativa internacional voltada à pesquisa e educação, visando promover desenvolvimentos ligados às áreas de inteligência artificial, robótica móvel, teoria de controle, visão computacional, dentre outras áreas. Uma das áreas de estudo inerente ao futebol de robôs é a simulação computacional, uma vez que permite redução de custos e um leque maior de possibilidades experimentais para criar, avaliar, modificar e explorar novas estratégias. Neste sentido, esse trabalho visa entender como simular o ambiente de futebol de robôs, criar um ambiente próprio e abordar todas as etapas da implementação desse simulador para futebol de robôs denominado SIM. Este simulador foi desenvolvido para ser simples, portátil, eficiente, com interface gráfica opcional à simulação. É voltado para métodos de inteligência artificial e técnicas inteligentes com objetivo de auxiliar o desenvolvimento de estratégias para a categoria MiroSot do jogo real de futebol de robos.

**Palavras-chave:** Simulador, Futebol de Robôs, Inteligência Artificial, Modelos computacionais.

# Abstract

The robot soccer is an international initiative focused on education and research to promote developments related to artificial intelligence, mobile robots, control theory, computer vision, among others. An intrinsic research field related to robot soccer is simulation focused on the simulation of robot soccer environment, as it allows cost savings and a wider range of experimental possibilities to create, evaluate, modify and explore new strategies. In this sense, this work aims to understand how to simulate the robot soccer environment as well as create a proper environment and describe all stages of the development of this simulator called SIM. This simulator was developed to be simple, portable, efficient and to display an optional graphical interface suitable for simulate experiments related to artificial intelligence methods and intelligent techniques to help develop strategies for MiroSot category of the real game of robot soccer.

**Keywords:** Simulator, Robot Soccer, Artificial Intelligence, Computational models.

## Agradecimentos

Agradeço a todos aqueles que de forma direta ou indireta contribuíram para esta parte da jornada.

---

Agradeço a minha mãe que me salvou no dia em que ela veio a vida. No final do dia tudo é sobre você.

---

Agradeço ao Prof. Dr. Saul de Castro Leite que com sua infinita paciência e gentileza me acompanhou nos últimos anos na execução deste trabalho. Agradeço também a Dr. Fernanda Leite que irrompeu sobre um emaranhado de palavras sem nenhuma substância, transformando com seus conselhos, algo sinuoso em algo com forma alinhada e exata. Também reconheço a presença do Lucas que entrou no projeto trazendo novo ânimo e novas perspectivas.

---

Agradeço as minhas amigas, Adriele, Anna, Bárbara, Camila e Isis Lee que retribuíram o azedume despejado a cada dia, com amizade incontestável. Obrigada!

# Sumário

<b>Lista de Figuras</b>	<b>5</b>
<b>1 Introdução</b>	<b>6</b>
1.1 Objetivos . . . . .	8
1.2 Estrutura do Trabalho . . . . .	8
<b>2 Futebol de Robôs e Simulação</b>	<b>9</b>
2.1 Futebol de Robôs . . . . .	9
2.1.1 <i>RoboCup</i> e <i>FIRA</i> . . . . .	10
2.1.2 Categoria <i>MiroSot - FIRA</i> . . . . .	12
2.2 Simulação de Futebol de Robôs . . . . .	13
<b>3 Núcleo da Simulação</b>	<b>15</b>
3.1 Modelo Matemático dos Objetos . . . . .	15
3.1.1 Modelo Matemático do Robô . . . . .	15
3.1.2 Modelo Matemático da Bola . . . . .	17
3.2 Tratamento de Colisões para Corpos Rígidos . . . . .	17
3.3 Detecção de colisão . . . . .	19
3.4 Pseudo-código do Núcleo . . . . .	24
3.5 Estrutura do simulador SIM . . . . .	26
3.5.1 Detalhamento das classes que compõem o simulador . . . . .	27
<b>4 Resultados</b>	<b>31</b>
<b>5 Considerações Finais</b>	<b>35</b>
<b>Referências Bibliográficas</b>	<b>36</b>

## Lista de Figuras

2.1	Imagem de um robô da categoria <i>miroSot</i> (FIRA, 2014) . . . . .	13
2.2	Sistema de controle da categoria <i>miroSot</i> (FIRA, 2014) . . . . .	13
3.1	Modelo do robô definido por Koh e Cho. . . . .	15
3.2	Os quatro tipos de colisões possíveis no ambiente de futebol de robôs: (a) bola - parede, (b) robô - parede, (c) robô - bola e (d) robô - robô. . . . .	18
3.3	Representação de uma colisão entre dois objetos . . . . .	18
3.4	Teste de colisão entre a bola e uma parede (Yamamoto, 2005). . . . .	20
3.5	Projeção do ponto C no segmento $\overline{AB}$ coincidindo com o ponto P. . . . .	21
3.6	Projeção do ponto C fora do segmento $\overline{AB}$ não coincidindo com o ponto P. . . . .	21
3.7	Detecção de colisão robô-bola. . . . .	22
3.8	Demonstração de colisão robô-bola detectada. . . . .	22
3.9	Cálculo dos vértices de um robô (Yamamoto, 2005). . . . .	23
3.10	Representação de uma possível colisão robô-robô . . . . .	23
3.11	Colisão entre dois robôs. . . . .	24
3.12	Sistema completo de simulação . . . . .	26
3.13	SIMi - a tela de visualização do simulador SIM . . . . .	26
3.14	Modelo UML do núcleo libSIM do simulador. . . . .	28
3.15	Modelo UML dos objetos do núcleo libSIM do simulador. . . . .	29
3.16	Modelo UML do SIMi do simulador. . . . .	30
4.1	Teste de colisão robô-robô e robô-parede. . . . .	31
4.2	Teste de colisão robô-robô e robô-bola. . . . .	32
4.3	Teste de colisão robô-robô, robô-bola e robô-parede. . . . .	33
4.4	Teste de colisão robô-robô e bola-parede . . . . .	33

# 1 Introdução

Em 1992, Alan Mackworth propôs em “*On Seeing Robots*” (Mackworth, 1992) a ideia de robôs jogando futebol. Esta ideia até então inexplorada no meio científico foi considerada pioneira (Robocup, 2013). Independente aos trabalhos de Mackworth, em outubro de 1992, um grupo de pesquisadores japoneses, Minoru Asada, Yasuo Kuniyoshi e Hiroaki Kitano, organizou em Tóquio o workshop intitulado “Grandes desafios para a Inteligência Artificial” onde foram debatidas formas de utilizar o jogo de futebol para promover ciência e tecnologia. Motivados pelas propostas e desafios resultantes desse workshop, propuseram em 1993 a *Robot J-League*, a primeira competição robótica de futebol que despertou o interesse de pesquisadores ao redor do mundo tornando-se um projeto internacional de esforço conjunto (Robocup, 2013). Em 1996, o professor coreano Jong-Hwan Kim, conhecido como “*o pai do futebol de robôs*” e pioneiro nas áreas de *Ubiquitous Robotics* e *Soccer Robotics*, criou em Daejeon, Coréia do Sul, o primeiro campeonato coreano de futebol de robôs em contrapartida aos campeonatos organizados pelos japoneses ( Fei (2014), FIRA (2014)).

Atualmente, o futebol de robôs é uma iniciativa internacional voltada à pesquisa e educação, visando promover desenvolvimentos ligados às áreas de inteligência artificial, robótica inteligente, teoria de controle, visão computacional, eletrônica, mecânica, aprendizado de máquina, controle e automação de sistemas, planejamento de caminhos, desenvolvimento de circuitos microcontrolados, planejamento cooperativo, entre outras áreas ( Kitano et al. (1997), Gedson et al. (2001), Gedson et al. (2004), Yamamoto (2005)). Neste sentido, por ser um ambiente multidisciplinar a plataforma do futebol de robôs é favorável a estas e diversas outras áreas de estudo por apresentar característica padronizada, isto é, há um problema padrão a ser resolvido. Sua pesquisa pode ser claramente definida e acompanhada, sendo assim possível testar, adaptar e avaliar várias teorias, algoritmos, arquiteturas e desempenhos ( Costa et al. (2000), Gedson et al. (2004)).

Todos os anos ocorrem campeonatos internacionais de futebol de robôs que são

regulamentados e fomentados pelas duas grandes entidades da competição robótica de futebol, a japonesa *RoboCup*<sup>1</sup> (Chen et al., 2003), e a sul-coreana *FIRA*<sup>2</sup> (FIRA, 2013). Dentro da modalidade futebol de robôs da *RoboCup* ocorrem competições divididas em cinco categorias, *middle size league*, *small size league*, *four-legged league*, *simulation league* e *humanoid league*, sendo estas definidas de acordo com a autonomia, o tamanho, o tipo e número de robôs por time, o tamanho e a função técnica de cada membro na equipe humana e a área de pesquisa. Já a *FIRA* apresenta sete categorias denominadas por *huroCup*, *amireSot*, *miroSot*, *naroSot*, *androSot*, *roboSot* e *simuroSot*, também fixadas a partir das mesmas particularidades encontradas nas especificações da *RoboCup*.

Uma categoria interessante e bem estudada é a categoria *miroSot*<sup>3</sup> da *FIRA*, que será abordada neste trabalho. Esta categoria consiste em times de três robôs cúbicos onde um deles pode fazer o papel de goleiro não havendo intervenção humana durante a partida.

É importante ressaltar que as competições organizadas tanto pela *RoboCup* quanto pela *FIRA* não envolvem somente o embate em si de robôs, mas também um congresso científico voltado a apresentar e discutir os estudos desenvolvidos no ambiente da competição robótica. Uma das áreas pesquisadas e que auxilia diversas categorias dentro das competições é a simulação computacional e, por isso, o uso e a análise de simuladores dentro da plataforma de futebol de robôs tornou-se uma prática constante. Por esses motivos, existem as categorias de competição, tais como a *simuroSot* da *FIRA* e *simulation 2D/3D* da *RoboCup*, que consideram somente o jogo simulado.

A simulação tem um papel importante, visto que aumenta as possibilidades experimentais para teorias muitas vezes inviáveis no ambiente real, mesmo quando são consideradas em categorias, onde são empregados robôs reais tal como a categoria *miroSot*. Contudo, para que exista uma simulação é necessário construir um modelo computacional que corresponda à situação do mundo real (Schulter, 2007).

---

<sup>1</sup>The Robot World Cup Initiative

<sup>2</sup>Federation of International Robot-Soccer Association

<sup>3</sup>Micro Robot World Cup Soccer Tournament

## 1.1 Objetivos

No presente trabalho é proposto a construção de um novo simulador para futebol de robôs denominado SIM. Este simulador tem por finalidade auxiliar o processo de elaboração de estratégias das equipes que competem na categoria *miroSot* da *FIRA*. Dentro dessa proposta o simulador SIM objetiva características como simplicidade, portabilidade e eficiência no planejamento de estratégias envolvendo métodos de inteligência artificial e técnicas inteligentes. Dentre os objetivos específicos deste trabalho destacam-se:

1. Estudar os modelos físicos e matemáticos que regem o futebol de robôs.
2. Estudar como simular esse sistema em um computador definindo a estrutura do simulador, seus elementos e componentes.

## 1.2 Estrutura do Trabalho

Este trabalho está organizado em cinco capítulos. O primeiro se refere a introdução do presente trabalho. No segundo capítulo é feita uma apresentação dos principais conceitos relacionados a futebol de robôs e bem como sua simulação. O terceiro capítulo trata da apresentação do simulador SIM e todos os seus principais componentes. Já no quarto capítulo são apresentados os resultados obtidos. Por fim, no quinto capítulo são feitas as considerações finais e conclusões desta monografia.

## 2 Futebol de Robôs e Simulação

Nesse capítulo são apresentados maiores detalhes sobre competições de futebol de robôs e suas categorias. Além disso, é discutido o papel da simulação voltada para futebol de robôs visando o desenvolvimento de estratégias de jogo inteligente.

### 2.1 Futebol de Robôs

O futebol de robôs, como o próprio nome já diz, é um jogo de futebol disputado por robôs onde dois times competem com o objetivo de ganhar fazendo o maior número de gols possíveis no time adversário. O jogo em si não difere da partida jogada por humanos, havendo assim substituições, penalidades em casos de falta, disputa de pênaltis e prorrogação. Porém cada categoria define condições específicas tanto para os robôs quanto para a equipe humana detentora do times.

Um time de futebol de robôs consiste em uma coleção de robôs autônomos capazes de reconhecer o ambiente onde estão inseridos através de informações oriundas de sensores presentes no próprio robô (Faceli, 2001). Com isso é possível reconhecer o campo de futebol e seus pontos de referência, os objetos pertencentes a este ambiente, a bola e os adversários. Estes dispositivos de reconhecimento devem ainda ser capazes de representar o ambiente, auxiliar o planejamento e o deslocamento dos jogadores. Em se tratando de um jogo de equipe, além do caráter autônomo, os jogadores robóticos devem ser capazes de interagir uns com os outros para estabelecer objetivos coletivos, metas globais, planejar, alocar tarefas aos demais integrantes do time, sincronizar as ações de forma a imprimir ao time um perfil cooperativo (Futvasf, 2008).

Em um jogo real a construção de um time de futebol de robôs precisa integrar diversos tipos de tecnologias, tais como: projeto de agentes autônomos, cooperação em sistemas multi-agentes, estratégias de aquisição de conhecimento, sistemas de tempo real, sistemas distribuídos, reconhecimento de padrões, integração de sensores, aprendizado e robótica móvel (Xbot, 2009). Tudo isso simplesmente para construir um time apto a

competir.

No entanto, algumas limitações surgem, pois os robôs móveis mesmo com toda essa tecnologia, ainda apresentam uma capacidade de navegação reduzida, uma vez que a navegação autônoma requer, entre outras coisas, aprender estratégias de navegação, adaptar-se a novas situações e construir conhecimento a partir das informações obtidas do seu ambiente. Isto é, fatores como luminosidade, transmissão de sinal, variações nos sensores, campo e robos podem contribuir para tornar estes problemas mais difíceis (Cazangi et al. (2003), Sun (2002)). Desta forma, há inúmeros fatores para se controlar, já que um time de robôs é um conjunto de tecnologias funcionando ao mesmo tempo, por exemplo, se um dos sensores que representam o ambiente de um dos robôs falhar, o desempenho do time é comprometido. Logo o alcance experimental do jogo real é mais reduzido e árduo do que o fornecido na simulação computacional (Futvasf, 2008).

### 2.1.1 *RoboCup e FIRA*

Atualmente existem duas federações que regulamentam os campeonatos de futebol de robôs, a coreana *FIRA* e a japonesa *RoboCup*. Tanto a *RoboCup* quanto a *FIRA* não foram criadas somente para promover competições entre robôs. Os jogos fazem parte de um congresso científico no qual são discutidas novas estratégias e tecnologias voltadas para a robótica e inteligência artificial (Ciência Hoje, 2013).

A *RoboCup* dedica-se ao avanço do estado da arte em inteligência artificial e robótica. Os objetivos específicos e as direções de pesquisa promovidos por esta federação são numerosos. O objetivo mais ambicioso a longo prazo é em 2050 construir um time de jogadores robóticos de futebol que possa vencer o time humano campeão do mundo (Xbot (2009), Chen et al. (2003)). Atualmente a *RoboCup* possui cinco categorias: *small size*, *middle size*, *four-legged*, *simulation 2D/3D* e *humanoid*. Essas categorias são definidas de acordo com a área de estudo, por exemplo, a *simulation 2D/3D* envolve o uso de simuladores 2D e 3D para futebol de robôs. Além da área de estudo, a divisão das competições é determinada levando-se em consideração a autonomia, o tamanho, a quantidade e o tipo de robô competidor, como no caso da liga *small size* que compreende dois times de seis robôs cúbicos autônomo em uma partida de futebol sem intervenção

humana, voltada para problemas de robôs inteligentes, multi-agente e cooperativos. A liga *middle size* também é dedicada a partidas sem intervenção humana, porém seus dois times são compostos de cinco robôs cúbicos com foco nas áreas de mecatrônica e design de software. Já as ligas *four-legged* e *humanoid* apresentam competições entre times de robôs humanóides, porém a categoria *four-legged* estipula que todas as partidas só podem ser jogadas por robôs H25 NAO enquanto a divisão *humanoid* engloba qualquer robô com aparência e habilidades aproximadamente humanas, apresentando assim três sub-divisões, *kidSize*, *teenSize* e *adultSize*.

Já a *FIRA* tem por objetivo principal “levar o espírito da Ciência e da Tecnologia ao leigo e à nova geração” Xbot (2009). Seus jogos são realizados a fim de promover o desenvolvimento de robôs autônomos e sistemas inteligentes que possam cooperar uns com os outros e contribuir para o avanço do estado da arte em robótica móvel autônoma e sistemas inteligentes ( Xbot (2009), FIRA (2013)). Hoje em dia, as competições organizadas pela *FIRA* apresentam sete categorias sendo estas, a *simuroSot*, *miroSot*, *naroSot*, *amireSot*, *roboSot*, *androSot* e *huroSot*. A *FIRA* também define suas categorias de acordo com a área de pesquisa e as particularidades dos robôs competidores. Desta forma, a liga *simuroSot* lida com a simulação do jogo de futebol robótico focada em desenvolvimento de estratégias. A divisão *miroSot* corresponde a partidas disputadas por dois times de três robôs cúbicos autônomos onde um deles pode exercer a função de goleiro não havendo intervenção humana no decorrer do jogo. A *naroSot* também representa disputas onde não ocorre intervenção humana e um dos robôs faz papel de goleiro, porém seus dois times possuem cinco robôs cúbicos. As categorias *amireSot* e *roboSot* apresentam robôs de geometria circular. As disputas da *amireSot* não permitem intervenção externa da equipe humana e envolvem times de um único robô circular. A *roboSot* abrange jogos entre dois times de até três robôs circulares autônomos ou semi-autônomos sem limite de peso. A categoria *androSot* envolve robôs semi-automáticos admitindo intervenção humana. Já a *huroSot* é voltada para disputas entre times de robôs humanóides que apresentem flexibilidade, robustez e versatilidade.

No Brasil, existe a Comissão Brasileira de Futebol de Robôs (CBF-R), composta por pesquisadores da área acadêmica, que organiza e promove, periodicamente, o cam-

peonato brasileiro de futebol de robôs. A intenção é que o evento seja sediado, alternadamente, pelas diversas universidades e centros de pesquisa envolvidos na área e filiados a CBF-R. Entretanto, as regras adotadas pela CBF-R devem seguir o padrão dos campeonatos promovidos pela *RoboCup* e pela *FIRA* em suas edições iniciais e, seus avanços devem ser feitos de maneira gradual (Xbot, 2009).

### 2.1.2 Categoria *MiroSot* - *FIRA*

A categoria *miroSot* da *FIRA* consiste em partidas jogadas por dois times formados por três robôs cúbicos autônomos de dimensões 7,5 cm x 7,5 cm x 7,5 cm (Figura 2.1). Em cada time um desses robôs pode exercer a função de goleiro e o jogo é dividido em dois tempos de cinco minutos com um único intervalo de dez minutos entre um tempo e outro. A equipe humana responsável pelo time robótico é formada por um coordenador, um treinador e um técnico, os quais só podem intervir no jogo durante o intervalo ou ao solicitar pedido de tempo com duração máxima de dois minutos (FIRA, 2014).

Todo o monitoramento de uma partida da categoria *miroSot* é feito por meio de sistema de controle (Figura 2.2) composto por um computador dedicado e duas câmeras posicionadas a dois metros de altura, responsáveis pela captura das imagens dos robôs, do campo e da bola. O processamento de imagem é feito a partir de um sistema de cor previamente definido, onde o campo só pode apresentar coloração preta ou verde clara. As linhas e arcos que compoem as marcações do campo devem ser brancas e a bola do jogo, nada mais que uma bola de golfe laranja. Cada time deve apresentar uma combinação de duas cores na parte superior de cada robô, através das quais cada jogador é identificado no decorrer do jogo. Cada equipe tem direito a um computador responsável não só pelo procesamento de imagem, mas também pela execução e o controle da estratégia dos robôs definida antes da partida. Esse computador também controla um sistema de comunicação via rádio, uma vez que a estratégia do jogo precisa ser repassada aos robôs, visto que durante o jogo, os usuários apresentados pela equipe, não interferem durante a partida. Cabe a eles somente a tarefa de escolher e posicionar no campo os jogadores disponíveis em seu time, conforme julguem a estratégia mais adequada (Schulter (2007), Yamamoto (2005), FIRA (2014)).

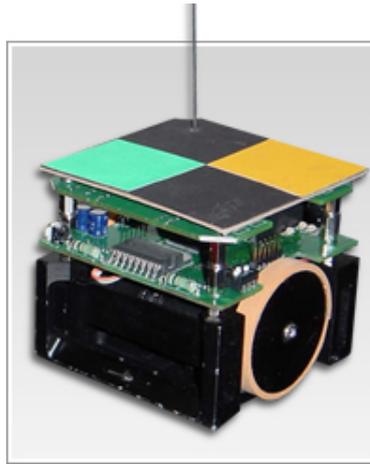


Figura 2.1: Imagem de um robô da categoria *miroSot* (FIRA, 2014)

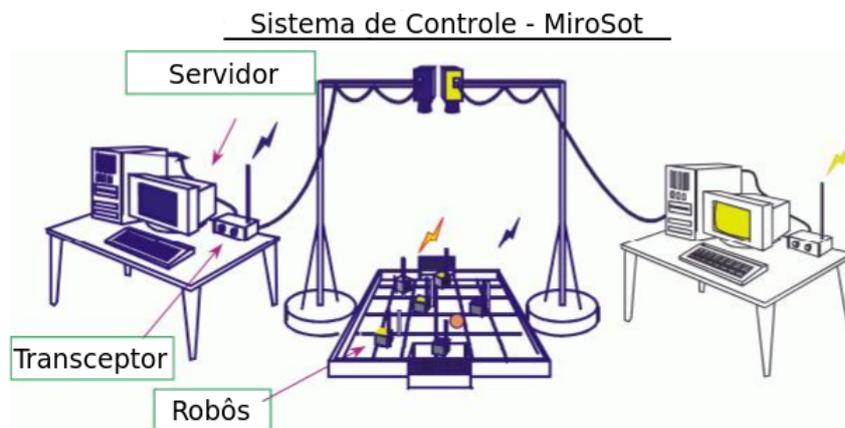


Figura 2.2: Sistema de controle da categoria *miroSot* (FIRA, 2014)

## 2.2 Simulação de Futebol de Robôs

Um simulador de futebol de robôs deve fornecer uma visualização simples e satisfatória de todo o ambiente bem como de todo o comportamento dos jogadores. Para tanto, é necessário que o simulador apresente modelos matemáticos bastante fiéis ao jogo real (Yamamoto, 2005).

O núcleo físico da simulação é o que torna a representação do ambiente virtual adequada ou não ao modelo real. Primeiro, é feita uma observação de todos os eventos físicos presentes no jogo. Após essa observação, é feito um modelo matemático que represente a tradução desses eventos que neste trabalho consistem na modelagem do robô, seu deslocamento e seu comportamento ao colidir (Abreu et al., 1999). O modelo matemático descreve todas as equações físicas relacionadas ao posicionamento, velocidades, acelerações, forças, colisões, momento linear e torque. Posto que todos esses fatores estão

presentes no jogo real e são essenciais para uma manipulação virtual adequada dos jogadores dentro da estratégia e do jogo.

Um aspecto importante das simulações de futebol de robôs envolve basicamente o desenvolvimento de estratégias não só para as categorias exclusivas de simuladores como a *simuroSot* da *FIRA* e *simulation 2D/3D* da *RoboCup*, mas também categorias onde os robôs jogam sem intervenção humana como a categoria *miroSot*.

Existem diversos simuladores para futebol de robôs, dentre eles o simulador fornecido pela *FIRA* denominado *simuroSot*, nome este que também representa a categoria dedicada exclusivamente ao uso e estudo do futebol de robôs por meio de simulação, e o fornecido pela *RoboCup* chamado de *Soccer Simulator*. Além desses, existem outros simuladores oriundos de iniciativas acadêmicas construídos no intuito de auxiliar suas próprias equipes e times de robôs na competição real, como o caso do simulador USPDS da Universidade de São Paulo.

O simulador *Soccer Simulator* da *RoboCup* é um sistema multi-agente desenvolvido para a plataforma GNU/Linux que permite times de onze robôs autônomos de geometria circular competirem em partidas de futebol utilizando métodos de inteligência artificial (Chen et al. (2003), Oliveira (2009)). O *simuroSot* da *FIRA* é um sistema de simulação 3D desenvolvido para a plataforma Windows, onde times de cinco a onze robôs de geometria cúbica disputam partidas de futebol com estratégias criadas pela equipe humana, e cada iteração da simulação gera imagem na interface gráfica 3D (FIRA, 2013). Já o USPDS, uma iniciativa acadêmica da Universidade de São Paulo, é um simulador de interface gráfica 3D independente, desenvolvido para a plataforma GNU/Linux que simula disputas de futebol de robôs da categoria *small size* da *RoboCup*, simulando assim times de três robôs cúbicos com estratégias utilizando métodos de inteligência artificial (Oliveira, 2009).

## 3 Núcleo da Simulação

Nesta seção é descrito o modelo matemático dos objetos físicos que compõem a simulação de futebol de robôs. Além disso, é apresentado algoritmos de detecção de colisões entre os objetos e modelos para o tratamento destas colisões. Por fim, é apresentado um pseudo-código que ilustra a implementação do núcleo principal da simulação.

### 3.1 Modelo Matemático dos Objetos

Os modelos dos objetos de simulação são usados para determinar as novas posições e orientações destes objetos após um passo de tempo da simulação.

#### 3.1.1 Modelo Matemático do Robô

O modelo matemático do robô considerado nesse estudo consiste em um robô cúbico de aresta de 7,5 cm com duas rodas laterais diferenciais (Figura 3.1). O robô pode ser controlado através da escolha das velocidades de rotação de cada roda, dadas por  $\omega_d$  e  $\omega_e$  para a velocidade de rotação da roda direita e esquerda, respectivamente.

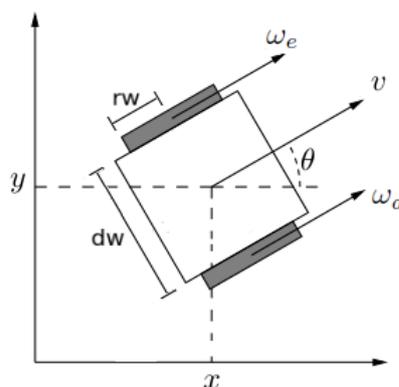


Figura 3.1: Modelo do robô definido por Koh e Cho.

Seja  $v_c$  e  $\omega_c$  a velocidade linear e angular do ponto central do robô. É possível estabelecer a seguinte rotação entre  $(v_c, \omega_c)$  e  $(\omega_d, \omega_e)$  (Koh e Cho, 1994).

$$v_c = \frac{(\omega_d + \omega_e) \cdot r_w}{2}, \quad (3.1a)$$

$$\omega_c = \frac{(\omega_d - \omega_e) \cdot r_w}{d_w}, \quad (3.1b)$$

onde  $r_w$  é o raio da roda do robô e  $d_w$  é a distância entre as rodas do robô.

O modelo para o deslocamento do robô no eixo  $x$  e no eixo  $y$ , dado por  $r_x$  e  $r_y$ , é descrito pela seguinte equação diferencial:

$$\frac{dr_x}{dt} = v_c \cdot \cos \theta, \quad (3.2a)$$

$$\frac{dr_y}{dt} = v_c \cdot \sin \theta, \quad (3.2b)$$

onde  $\theta$  é a direção do robô (Figura 3.1) e ela satisfaz:

$$\frac{d\theta}{dt} = \omega_c. \quad (3.3)$$

Desta forma, durante um passo de tempo  $dt$  na simulação, pode-se calcular a nova posição do robô integrando as Equações 3.2a, 3.2b e 3.3 para obter:

$$r_x^{k+1} = r_x^k + \frac{[\sin \theta^{k+1} - \sin \theta^k] \cdot v_c}{\omega_c}, \quad (3.4a)$$

$$r_y^{k+1} = r_y^k - \frac{[\cos \theta^{k+1} - \cos \theta^k] \cdot v_c}{\omega_c}, \quad (3.4b)$$

$$\theta^{k+1} = \theta^k + \omega_c \cdot dt, \quad (3.4c)$$

quando  $\omega_c \neq 0$ . No caso em que  $\omega_c = 0$ , usa-se a seguinte expressão:

$$r_x^{k+1} = r_x^k + v_c \cdot dt \cdot \cos \theta^k, \quad (3.5a)$$

$$r_y^{k+1} = r_y^k - v_c \cdot dt \cdot \sin \theta^k, \quad (3.5b)$$

$$\theta^{k+1} = \theta^k, \quad (3.5c)$$

O modelo apresentado foi retirado da abordagem definida pelos pesquisadores coreanos Koh e Cho (Koh e Cho, 1994). Existem outros modelos mais complexos para o mesmo robô, como os apresentados por Han e Yamamoto em ( Han e Kopacek (2000), Ya-

mamoto (2003)) e até modelos definidos para robôs mais complexos como no caso apresentado por Zhao (Zhao, 1992). Neste trabalho decidiu-se utilizar a abordagem do modelo definido por Koh e Cho (Koh e Cho, 1994) em razão de ser um modelo simplificado dependente de poucos parâmetros.

### 3.1.2 Modelo Matemático da Bola

O modelo matemático da bola é descrito simplesmente pela posição de seu centro no eixo  $x$  e  $y$ , dados por  $r_x$  e  $r_y$ , respectivamente; pela sua velocidade linear no centro  $\mathbf{v}_c$  e sua direção  $\theta$ . Tem-se que:

$$\frac{dr_x}{dt} = v_b \cdot \cos \theta, \quad (3.6a)$$

$$\frac{dr_y}{dt} = v_b \cdot \sin \theta, \quad (3.6b)$$

$$\frac{dv_b}{dt} = \beta \cdot v_b, \quad (3.6c)$$

onde  $\beta$  é o coeficiente de atrito viscoso linear da bola (Yamamoto, 2005).

O posicionamento da bola é atualizado utilizando as Equações 3.5a, 3.5b e 3.5c, e a velocidade da bola é atualizada a partir da seguinte equação:

$$v_c^{k+1} = v_c^k + \beta \cdot v_c^k \cdot dt = (1 + \beta \cdot dt) \cdot v_c^k. \quad (3.7)$$

## 3.2 Tratamento de Colisões para Corpos Rígidos

No ambiente do futebol de robôs ocorrem colisões elásticas, isto é, colisões onde há conservação de energia cinética e momento linear. Se não houver tratamento para essas colisões, o simulador não será capaz de fornecer uma simulação confiável que represente um jogo da categoria *miroSot*. Desta forma é preciso tratar todas as colisões que podem ocorrer nos seguintes formatos: colisão bola-parede, colisão robô-parede, colisão robô-bola e colisão robô-robô, logo para que o simulador gere uma representação que auxilie o jogo real, é preciso detectar e incluir os efeitos das colisões na simulação (Yamamoto, 2003). A Figura 3.2 ilustra os quatro tipos de colisões presentes no ambiente de simulação. O modelo para tratamento de colisões entre dois corpos utilizado neste trabalho segue o

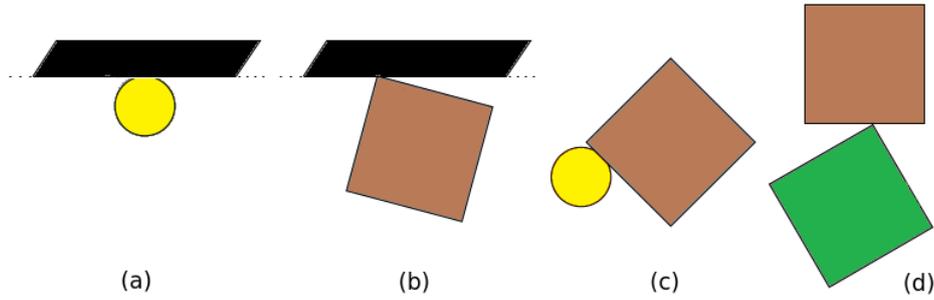


Figura 3.2: Os quatro tipos de colisões possíveis no ambiente de futebol de robôs: (a) bola - parede, (b) robô - parede, (c) robô - bola e (d) robô - robô.

modelo apresentado por (Hecker, 1997), no qual é considerado colisões instantâneas sem atrito.

Suponha que ocorra uma colisão de dois objetos A e B no ponto P. É considerado que o objeto A possui velocidade  $v_i^A$  e o objeto B velocidade  $v_i^B$ . Além disso, o objeto A possui velocidade angular dada por  $\omega_i^A$  e  $\omega_i^B$ . A Figura 3.3 mostra os objetos A e B colidindo no ambiente de simulação. A velocidade linear e angular dos objetos A e B após

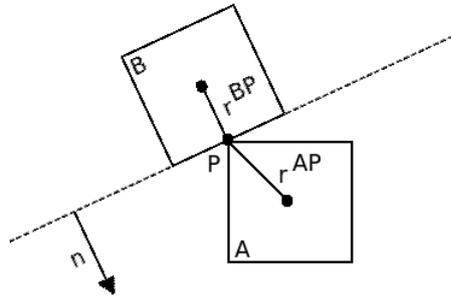


Figura 3.3: Representação de uma colisão entre dois objetos

a colisão são dadas por  $v_f^A$ ,  $v_f^B$ ,  $\omega_f^A$  e  $\omega_f^B$  determinadas através do cálculo do impulso.

Para calcular o impulso proveniente de uma colisão é preciso determinar a velocidade relativa dos objetos no ponto P, dada por  $v^{AB}$ . Esta velocidade é dada por:

$$v^{AB} = v^{AP} - v^{BP}, \quad (3.8)$$

onde  $v^{AP}$  e  $v^{BP}$  são as velocidades dos objetos A e B no ponto P, respectivamente. Estas velocidades são calculadas da seguinte forma:

$$v^{AP} = v_i^A + \omega_i^A r_{\perp}^{AP}, \quad (3.9a)$$

$$v^{BP} = v_i^B + \omega_i^B r_{\perp}^{BP}, \quad (3.9b)$$

onde  $r_{\perp}^{AP}$  e  $r_{\perp}^{BP}$  denotam vetores normais à  $r^{AP}$  e  $r^{BP}$ .

Com esses valores, o impulso  $j$  da colisão é calculado da seguinte forma:

$$j = \frac{-(1+e)v_1^{AB} \cdot \mathbf{n}}{\mathbf{n} \cdot \mathbf{n} \left( \frac{1}{M^A} + \frac{1}{M^B} \right)}, \quad (3.10)$$

onde  $M^A$  e  $M^B$  são as respectivas massas dos objetos A e B envolvidos na colisão,  $\mathbf{n}$  é o vetor normal a colisão (Figura 3.3) e  $e$  é o coeficiente de restituição. Este coeficiente de restituição dos corpos modela a compressão e restituição dos corpos em impacto. O coeficiente  $e$  é um valor escalar que indica quanta energia é dissipada durante a colisão, no qual seu valor pode variar de 0 à 1. O valor escolhido para o simulador foi  $e = 1$  (Hecker, 1997).

Com o impulso calculado, determina-se as novas velocidades linear e angular dos objetos da seguinte forma:

$$v_f^A = v_i^A + \frac{j}{M^A} \cdot \mathbf{n}, \quad (3.11a)$$

$$\omega_f^A = \omega_i^A + \frac{r_{\perp}^{AP} \cdot j \mathbf{n}}{I^A}, \quad (3.11b)$$

$$v_f^B = v_i^B - \frac{j}{M^B} \cdot \mathbf{n}, \quad (3.11c)$$

$$\omega_f^B = \omega_i^B - \frac{r_{\perp}^{BP} \cdot j \mathbf{n}}{I^B}, \quad (3.11d)$$

onde  $I^A$  e  $I^B$  é o momento de inércia dos objetos A e B, respectivamente (Hecker, 1997).

### 3.3 Detecção de colisão

Uma vez tratada as colisões, a etapa seguinte consiste em determinar formas para detectar as colisões definidas na Seção 3.2.

### Colisão bola-parede

A colisão bola-parede ocorre sempre que a distância entre o centro da bola e um segmento de reta que compõe as paredes do campo é menor ou igual ao raio da bola. A Figura 3.4 demonstra o exemplo de verificação de uma colisão bola-parede.

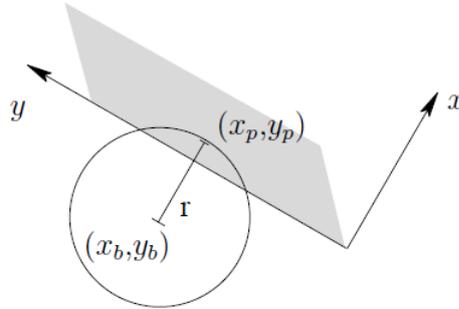


Figura 3.4: Teste de colisão entre a bola e uma parede (Yamamoto, 2005).

Para calcular a distância entre um ponto no centro da bola e um segmento de reta  $\overline{AB}$  da parede, faz-se inicialmente a parametrização do segmento de reta  $\overline{AB}$  da seguinte forma:

$$x(t) = A + t \cdot (A - B), \quad (3.12)$$

onde  $t \in [0, 1]$ .

O ponto D de projeção de C na reta que contém  $\overline{AB}$  é determinado a partir da seguinte equação:

$$(C - D) \cdot (B - A) = 0. \quad (3.13)$$

Substituindo D pela Equação 3.12 acima, pode-se determinar o valor de t correspondente ao ponto D:

$$(C - A - t \cdot (B - A)) \cdot (B - A) = 0, \quad (3.14a)$$

$$(C - A) \cdot (B - A) - t \cdot \|B - A\|^2 = 0, \quad (3.14b)$$

$$t = \frac{(C - A) \cdot (B - A)}{\|B - A\|^2}. \quad (3.14c)$$

Desta forma o ponto mais próximo de  $C$  no segmento  $\overline{AB}$  é dado por:

$$P = \begin{cases} A, & \text{se } t \leq 0 \\ B, & \text{se } t \geq 1 \\ (A + t \cdot (B - A)), & \text{caso contrário.} \end{cases} \quad (3.15)$$

Com o ponto  $P$  determinado, a menor distância entre  $C$  e o segmento  $\overline{AB}$  é :

$$dist = \|C - P\|. \quad (3.16)$$

As Figuras 3.5 e 3.6 representam os casos em que o ponto  $C$  é projetado no segmento  $\overline{AB}$  e fora dele, respectivamente.

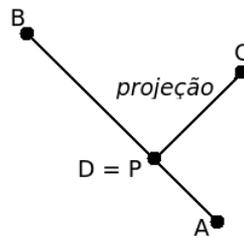


Figura 3.5: Projeção do ponto  $C$  no segmento  $\overline{AB}$  coincidindo com o ponto  $P$ .

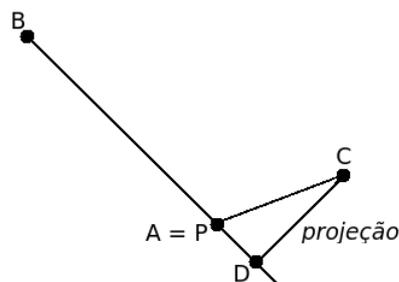


Figura 3.6: Projeção do ponto  $C$  fora do segmento  $\overline{AB}$  não coincidindo com o ponto  $P$ .

### Colisão robô-bola

A colisão robô-bola assemelha-se a colisão bola-parede. No entanto, para verificar a colisão robô-bola representada nas Figuras 3.7 e 3.8, calcula-se a distância entre um ponto definido no centro da bola até os segmentos de reta representados pelos lados do robô. Essas distâncias são comparadas com o raio da bola. Se forem maiores que o raio, a colisão não ocorreu.

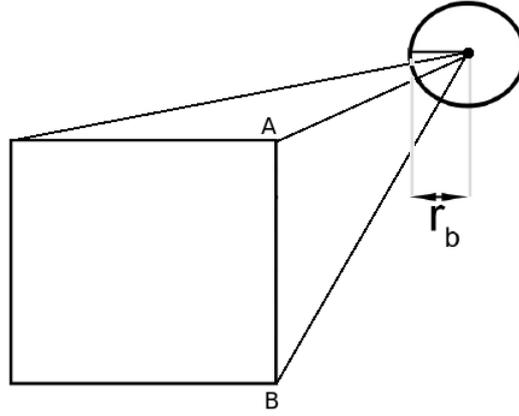


Figura 3.7: Detecção de colisão robô-bola.

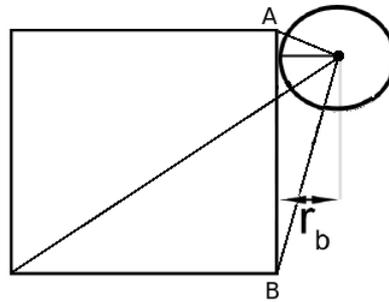


Figura 3.8: Demonstração de colisão robô-bola detectada.

### Colisão robô-parede

Na colisão robô-parede é preciso calcular a posição dos quatro vértices do robô e verificar se pelo menos um deles excede em qualquer uma de suas componentes (x,y) uma das paredes do campo. Deste modo, considera-se que o robô colide apenas com os vértices (Yamamoto, 2005). Para o cálculo dos vértices utiliza-se a seguinte equação:

$$x_j = x_c + \frac{b}{\sqrt{2}} \cos(\theta + 45^\circ + 90^\circ \cdot j), \quad (3.17)$$

$$y_j = y_c + \frac{b}{\sqrt{2}} \cos(\theta + 45^\circ + 90^\circ \cdot j), \quad (3.18)$$

sendo  $j$  o  $j$ -ésimo vértice,  $j = 0, 1, 2, 3, \dots, n$ , onde  $n$  é o número de vértices,  $(x_c, y_c)$  o centro do robô e  $\theta$  a orientação do robô.

A Figura 3.9 abaixo demonstra a explicação do cálculo dos vértices de um robô.

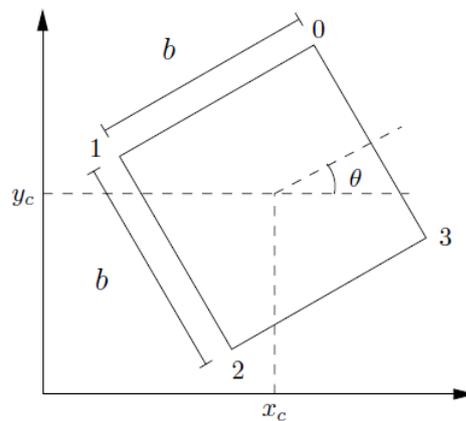


Figura 3.9: Cálculo dos vértices de um robô (Yamamoto, 2005).

### Colisão robô-robô

Na colisão robô-robô primeiro é verificado se as circunferências que circunscrevem os robôs se interceptam. É feito um cálculo rápido medindo a distância dos centros dos robôs até as suas circunferências e, se a distância entre o centro de um robô ao outro for menor que a distância entre o centro do robô e sua circunferência, há uma probabilidade de colisão. Caso isso ocorra, o próximo passo será verificar se um dos segmentos de reta que formam os robôs possuem pelo menos um ponto de interseção. Caso não exista pelo menos um ponto de interseção, a colisão não ocorreu (Gavrilova, 2000). As Figuras 3.10 e 3.11 demonstram a representação do processo de verificação da colisão do tipo robô-robô.

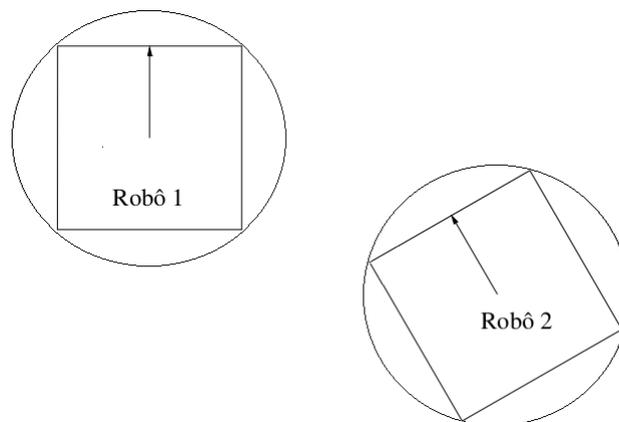


Figura 3.10: Representação de uma possível colisão robô-robô .

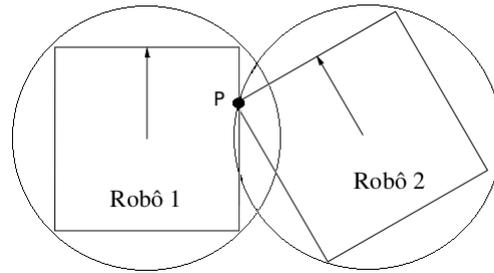


Figura 3.11: Colisão entre dois robôs.

### 3.4 Pseudo-código do Núcleo

Após o processo de modelagem dos objetos (robôs e bola), da cinemática dos objetos e das colisões, a próxima etapa de implementação do simulador é a definição dos algoritmos que o compõe. O Algoritmo 1 representa o funcionamento do laço principal do núcleo do simulador SIM. Esse laço calcula as novas posições dos objetos da simulação após um passo  $dt$  de tempo. Com o tempo de simulação em curso, as novas posições dos robôs e da bola são dadas de acordo com os modelos descritos na Seção 3.1.

Com os objetos em movimento, o Algoritmo 1 verifica a ocorrência de colisões aceitáveis durante a simulação. Uma colisão é dita inaceitável se um dos objetos envolvidos na colisão atravessa o corpo do outro objeto mais do que o definido para a simulação. Caso as colisões detectadas sejam aceitáveis, o tempo  $dt$  da partida é incrementado com mais um passo, ou seja, o jogo prossegue. Com a disputa em curso, é preciso verificar a ocorrência ou não de gols. Na hipótese de gol, o Algoritmo 1 analisa se o gol verificado trata-se do gol direito ou esquerdo. Em seguida, o jogo é reiniciado.

No caso em que haja colisão inaceitável, o estado atual da partida volta ao estado inicial, o passo  $dt$  assume metade do valor inicial e o novo estado é determinado com o passo reduzido. Esse processo é realizado até que não haja mais colisões. Posteriormente ao tratamento da colisão, a simulação prossegue a partir do último passo  $dt$  calculado. O estado atual da partida então prossegue dando sequência a simulação.

O Algoritmo 1 a seguir descreve o funcionamento do laço principal do núcleo do simulador.

---

**Algorithm 1:** Algoritmo de Simulação

---

**Entrada:** dt**Variáveis:** teste, passo, t-total, estado-original;**início**

t\_total = 0.0;

passo = MIN(dt,MAXDT);

teste = verdadeiro;

estado\_original = estado\_atual;

**repita**move todos os objetos de acordo com o tamanho do passo dado,  
verifica se houve colisão e trata a colisão**se colisão aceitável então**

prossigue o tempo de jogo;

t\_total += passo;

passo = MIN(dt\_tot,MAXDT);

estado\_original = estado\_atual;

teste = verdadeiro;

**se ocorrer gol então**

testa se foi gol direito ou esquerdo

reinicia o jogo

**fim se****fim se****senão**

ocorreu colisão inaceitável;

passo = passo/2.0;

estado\_atual = estado\_original;

**se passo < MIN\_DT então**

teste = falso;

passo = MIN\_DT;

**fim se****fim se****até** t\_total < dt;

dt = t\_total;

**Saída:** estado\_atual**fim**

---

## 3.5 Estrutura do simulador SIM

O simulador SIM é composto por três partes independentes: O LibSIM, o SIMi e a *Strategy*, Figura 3.12. O LibSIM é o núcleo físico do simulador capaz de construir uma simulação confiável a partir da modelagem matemática do jogo de futebol de robôs. Ele é definido no simulador SIM como uma biblioteca de classes, a qual contém o núcleo principal do simulador podendo ser chamado por outros programas.

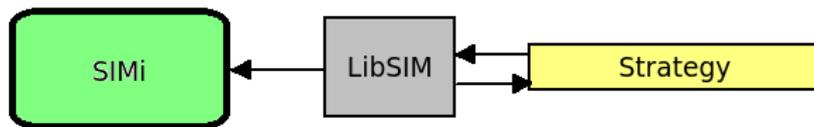


Figura 3.12: Sistema completo de simulação

O visualizador SIMi (SIM interface) é um programa desenvolvido em SDL e OpenGL, responsável por mostrar na tela a cena gerada por cada iteração da simulação. O simulador é independente do SIMi. Deste modo é possível executar várias simulações sem o visualizador. A Figura 3.13 demonstra a imagem da tela de visualização.



Figura 3.13: SIMi - a tela de visualização do simulador SIM

A *strategy* é responsável por definir o que de fato os times farão durante a partida. Trata-se de uma interface definida no LibSIM implementada através de um encadeamento de execução (*thread*) independente do simulador para não interferir com o seu tempo

de execução. Além disso, a estratégia é implementada como uma biblioteca de vínculo dinâmico (.dll no windows e .so em Linux) para que possa ser compilada e carregada no simulador de forma independente.

### 3.5.1 Detalhamento das classes que compõem o simulador

O libSIM é organizado de acordo com as classes mostradas na Figura 3.14, onde a classe *simulator* possui um estado de jogo definido na classe *state* e duas estratégias determinadas na classe *strategy*. A classe *strategy* por sua vez precisa das classes *thread* e *dll* para funcionar corretamente, dado que a estratégia deriva do encadeamento de execução implementado dentro da classe *thread* e a execução da estratégia em si é dada como uma biblioteca de vínculo dinâmico desenvolvido na classe *dll*. Já a classe *state* possui um campo, uma bola e dois times, e por isso estabelece relação com as classes *field*, *ball* e *team*. Por conseguinte, um time possui 3 robôs jogadores tendo assim uma relação entre as classes *team* e *robot*.

Os objetos presentes dentro da libSIM derivam da classe *object* (Figura 3.15) que é responsável por definir os objetos da cena, sendo um desses objetos a bola do jogo. Além de uma bola, a classe *object* define os elementos básicos para representar os polígonos definidos dentro da classe *polygon*. A partir da classe *polygon* deriva-se a construção do campo e dos robôs.

Já a organização da SIMi demonstrada na Figura 3.16 deriva da classe *visualizer* que usa um núcleo gráfico específico para definir a cena do jogo. Na cena é preciso um contador de tempo e um dispositivo gráfico, oriundos das classes *gfx-timer* e *device*, respectivamente. A cena criada no núcleo gráfico e mostrada na tela e é construída a partir das classes *sdl-gfx-engine*, *sdl-gfx-timer* e *sdl-gfx-device*.

Todas essas classes representam a forma como o núcleo físico libSIM e a interface gráfica SIMi estão organizados dentro do simulador SIM.

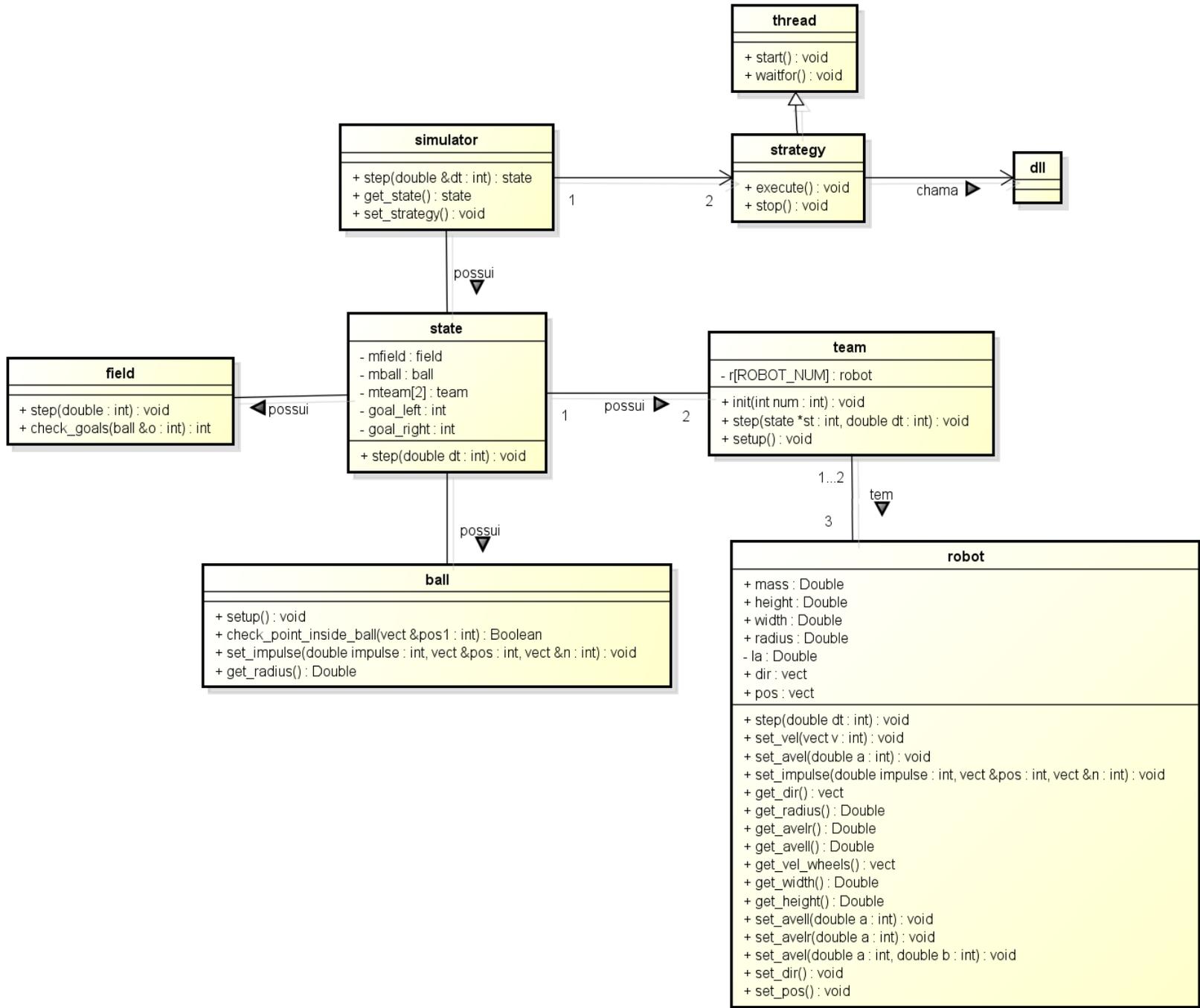


Figura 3.14: Modelo UML do núcleo libSIM do simulador.

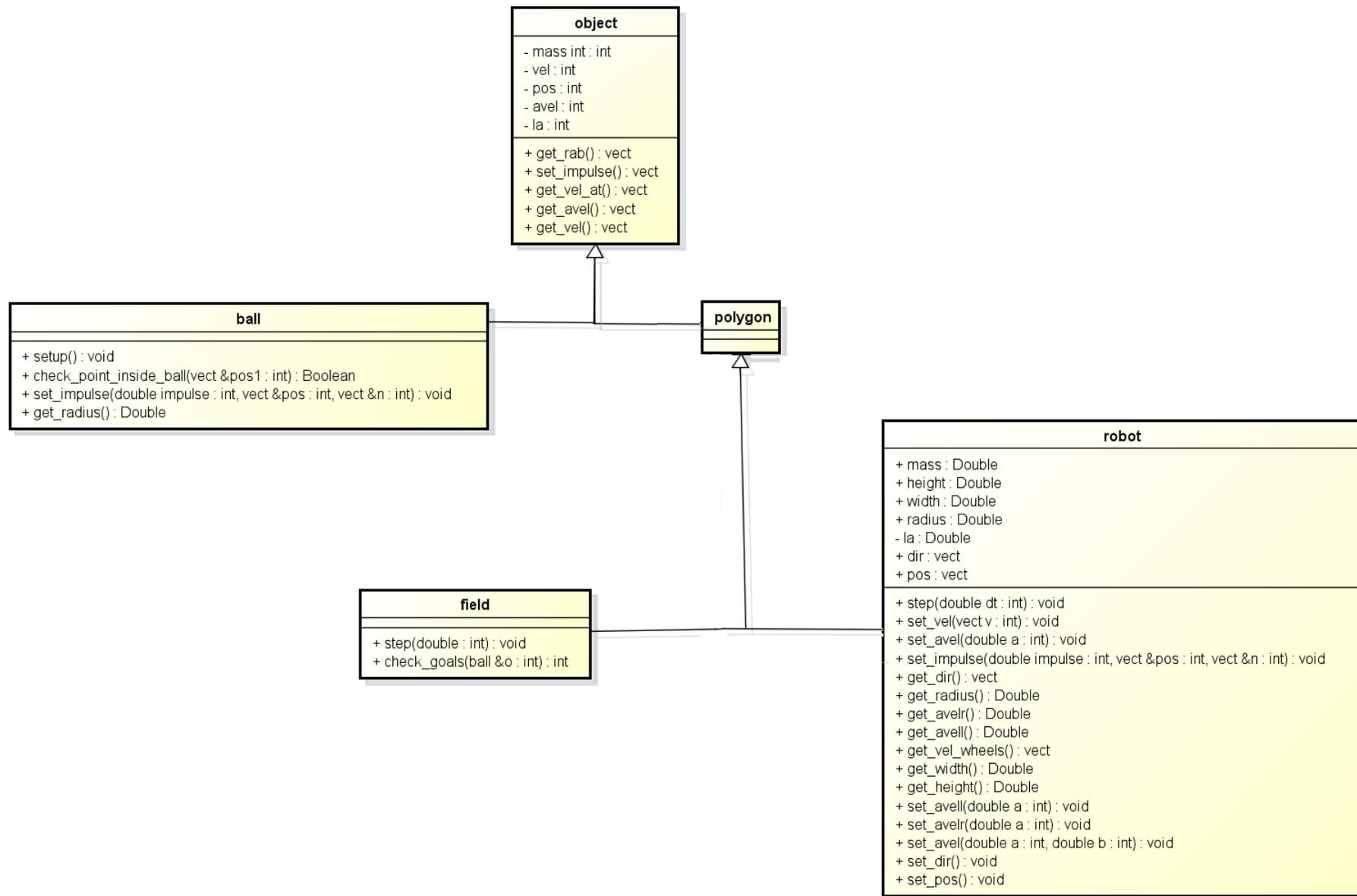


Figura 3.15: Modelo UML dos objetos do núcleo libSIM do simulador.

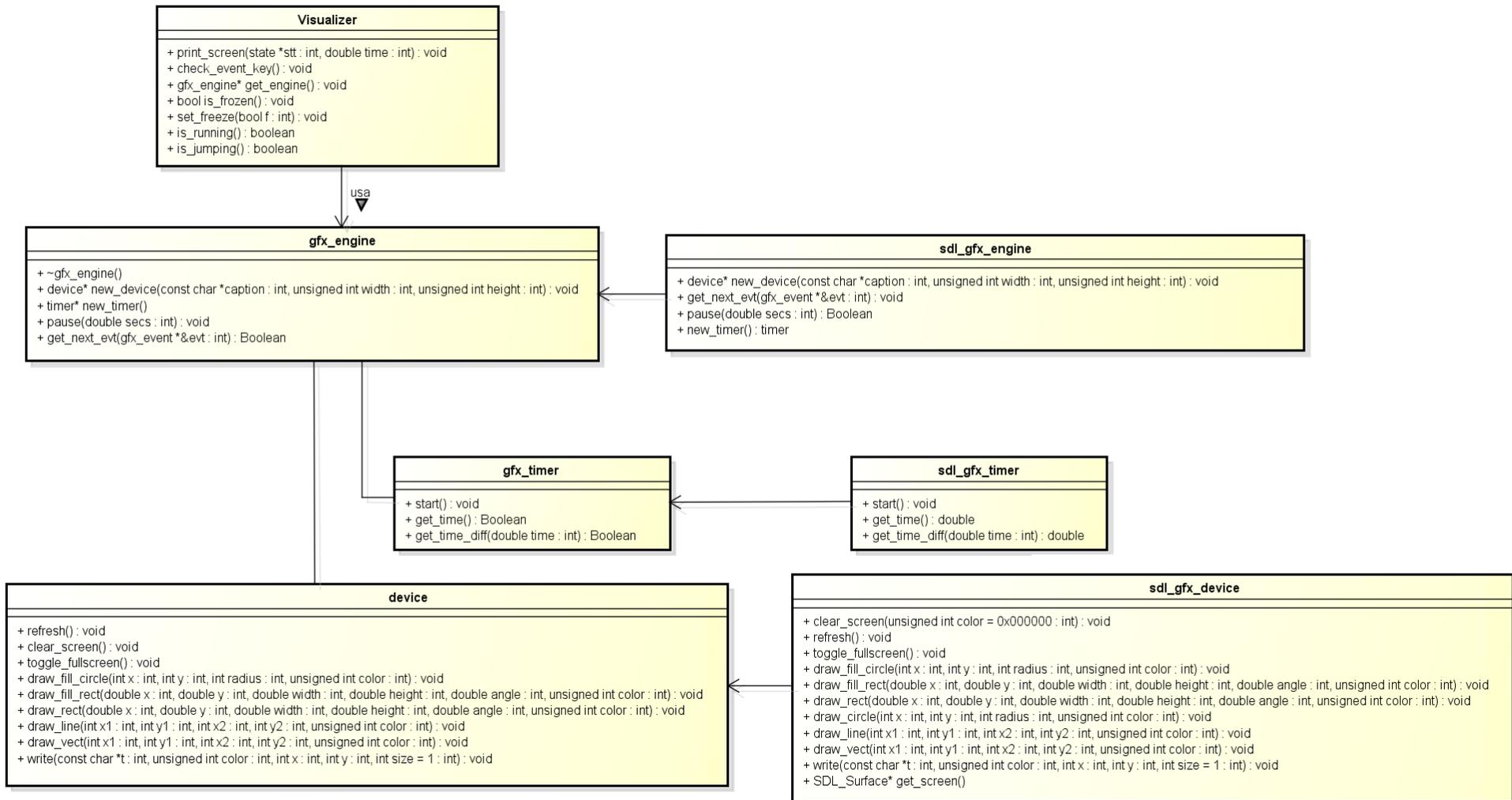


Figura 3.16: Modelo UML do SIMi do simulador.

## 4 Resultados

No intuito de testar o funcionamento do simulador foram analisadas as situações que envolviam os quatro tipos de colisão presentes no ambiente do futebol de robôs, as quais são: colisão bola-parede, colisão robô-bola, colisão robô-parede e colisão robô-robô. A estratégia definida para os testes executados envolveu todos os robôs de ambos os times deslocando-se para o centro do campo em direção a bola. Com essa estratégia foi possível analisar o comportamento dos objetos em uma simulação que continha mais de um tipo de colisão ocorrendo ao mesmo tempo, visto que o simulador SIM visa fornecer uma representação adequada do jogo apresentado na categoria *miroSot*.

O primeiro caso analisado envolveu as colisões robô-robô e robô-parede (Figura 4.1). O tratamento das colisões demonstrado na Seção 3.2 aplicado aos casos, robô-robô e robô-parede, escolhidos para o primeiro teste, denotaram a corretude do Algoritmo 1 definido na Seção 3.4. Desde modo, a cena gerada não apresentou interpenetrações dos objetos envolvidos na cena.

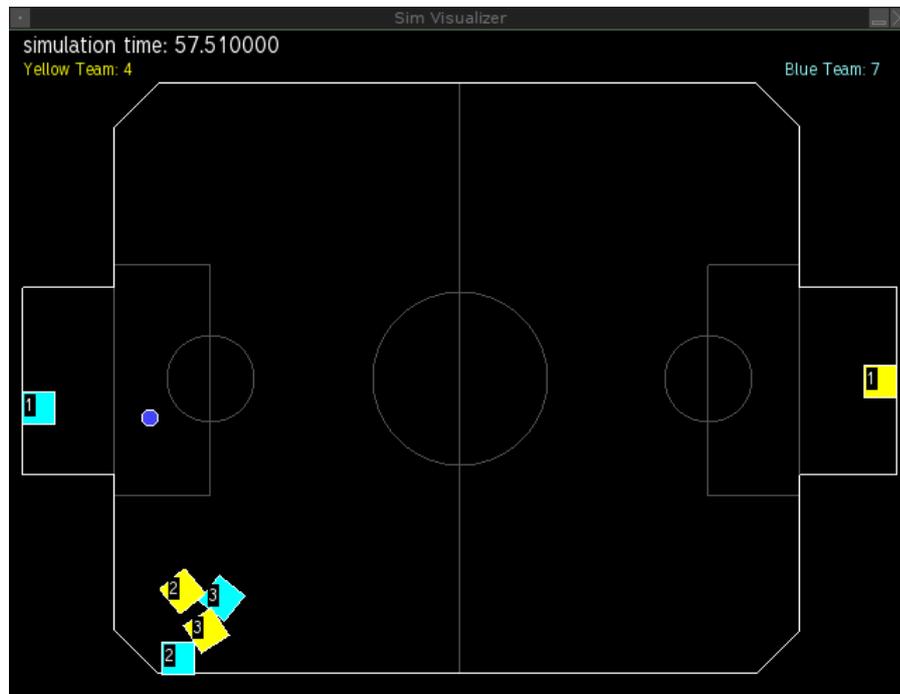


Figura 4.1: Teste de colisão robô-robô e robô-parede.

O segundo caso analisado envolveu as colisões robô-robô e robô-bola (Figura 4.2). A cena exibida também não apresentou colisões inaceitáveis entre os objetos em colisão.

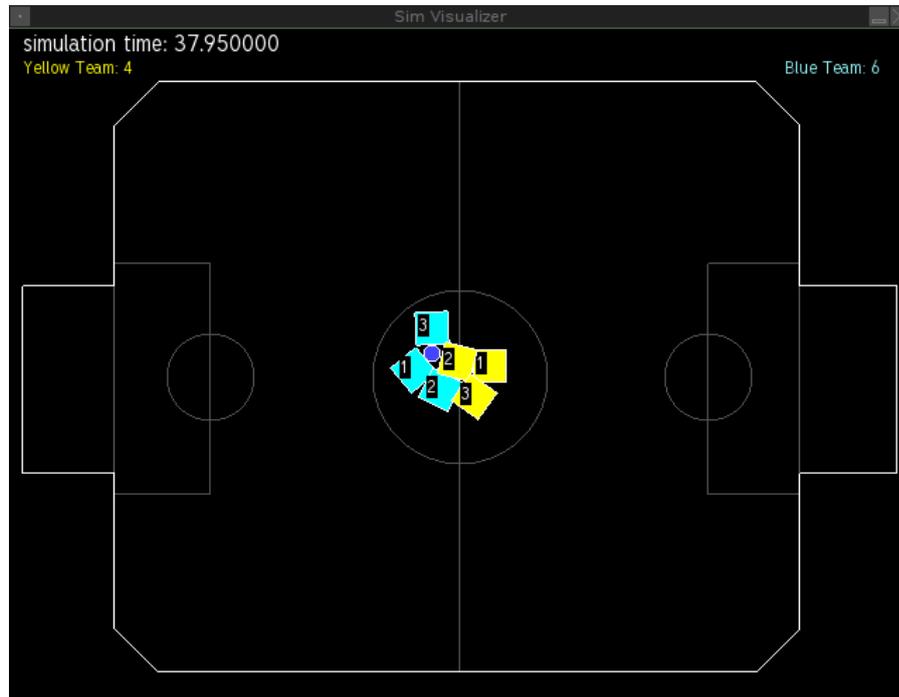


Figura 4.2: Teste de colisão robô-robô e robô-bola.

O terceiro caso explorado abrangeu as colisões robô-robô, robô-bola e robô-parede (Figura 4.3). O SIMi representou as colisões envolvidas sem apresentar falhas na geração das imagens de cada iteração. Logo, o tratamento das colisões foi realizado de forma correta tendo a cena exibido somente colisões aceitáveis.

O quarto caso avaliado compreendeu as colisões robô-robô e bola-parede (Figura 4.4), cuja representação no SIMi exibiu colisões correspondentes aos casos verificados durante uma partida da categoria *miroSot*. Desta forma, os choques entre os robôs e o choque entre a bola e a parede, não foram inaceitáveis. Isto é, não foi exibida nenhuma interpenetração dos objetos ao colidirem.

No decorrer dos testes, a simulação do jogo exibida na tela manteve-se fiel ao jogo real da categoria *miroSot*. Não houve em nenhum momento um comportamento anormal no que tange o deslocamento, a interação das forças e ao tratamento de colisões para os objetos durante a simulação.

A partir dos resultados positivos exibidos na avaliação do jogo simulado em tela, analisou-se a eficiência do simulador SIM. Posto que é desejado uma simulação confiável

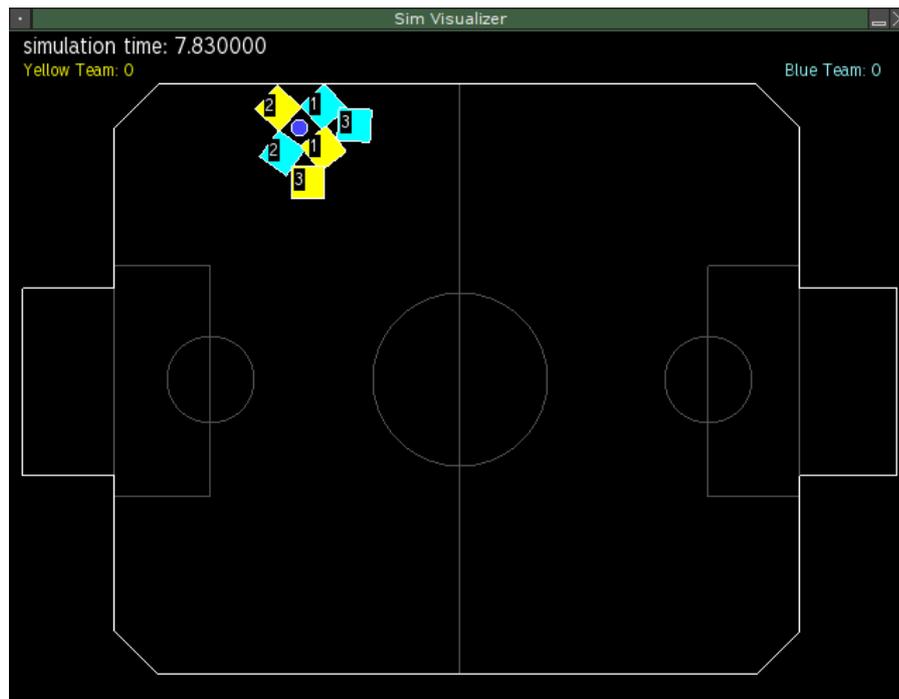


Figura 4.3: Teste de colisão robô-robô, robô-bola e robô-parede.



Figura 4.4: Teste de colisão robô-robô e bola-parede

e eficiente da disputa *miroSot*. Deste modo, o tempo de execução e processamento dos testes expostos anteriormente, apresentou tempo de simulação total igual a 90.819130 segundos. Sendo 90.000000 segundos referentes somente ao tempo de simulação, 0.819130 segundos para iniciar a interface gráfica SIMi e 16.153299 segundos destinados ao tempo computacional. Durante 74.391083 segundos o processo de simulação permaneceu inerte,

---

isto é, não ocorreu processamento desnecessário. Todos os testes realizados no simulador SIM ocorreu em um computador intel core i7 - 2630QM CPU 2.00 Ghz e sistema operacinal Linux.

## 5 Considerações Finais

A conclusão do simulador SIM abre boas possibilidades de estudos futuros. O uso mais imediato para o simulador visa o planejamento de estratégias para a competição robótica de futebol envolvendo uma técnica simples e de baixo custo computacional conhecida por campo potencial harmônico ( Gedson et al. (2001), Gedson et al. (2004)). Uma segunda possibilidade seria aliar essa técnica a métodos de aprendizado por reforço como *Q-learning* associado a redes neurais, entre outras (Riedmiller et al., 2009).

Além do amplo universo de estudos envolvendo aprendizado por reforço para o desenvolvimento de estratégias dentro da plataforma de futebol de robôs, pretende-se realizar otimizações no núcleo físico do simulador na versão Windows para equiparar a eficiência das duas versões (Linux e Windows) do SIM garantindo deste modo a portabilidade visada na construção do simulador.

## Referências Bibliográficas

- ABREU, A. M. M.; RANGEL, J. J. Simulação computacional: uma abordagem introdutória. **UENF - Vértices** 2(1), 1999.
- CAZANGI, R. R.; VON ZUBEN, F. J. ; FIGUEIREDO, M. *A classifier system in real applications for robot navigation*. 2003.
- CHEN, M.; DORER, K. *RoboCup Soccer Server. Users Manual*. Disponível em: <<http://www.robocup.org/>>. Acesso em 3 Fev. 2013, 2003.
- CiênciaHoje. **Ciência e futebol - robôs também disputam copa do mundo de futebol**. Disponível em: <<http://cienciahoje.uol.com.br/especiais/ciencia-e-futebol/robos-tambem-disputamcopa-do-mundo-de-futebol>>. Acesso em 23 Fev. 2013, 2013.
- COSTA, A. H. R.; PEGORARO, R. Construindo robôs autônomos para partidas de futebol: O time guaraná. 2000.
- FIRA. **Federation of international robot-soccer association**. Disponível em: <http://www.fira.net/soccer/simurosot/overview.htm>. Acesso em 3 Fev. 2013, 2010.
- NET, F. **Federation of international robot-soccer association**. "Disponível em: <[http://www.fira.net/contents/sub03/sub03\\_3.asp/](http://www.fira.net/contents/sub03/sub03_3.asp/)>. Acesso em 2 Fev.2014", 2014.
- NET, F. **Federation of international robot-soccer association**. "Disponível em: <[http://www.fira.net/contents/sub01/sub01\\_3.asp/](http://www.fira.net/contents/sub01/sub01_3.asp/)>. Acesso em 2 Fev.2014", 2014.
- FACELI, K. Combinação de métodos de inteligência artificial para fusão de sensores. 2001.
- INDUSTRIAL, C. E. D. F. D. E. **Futebol de robôs**. "Disponível em: <[http://portal.fei.edu.br/pt-BR/pesquisas\\_projetos/projetos\\_institucionais/Robo\\_FEI/futebol\\_de\\_robos/Paginas/futebol\\_de\\_robos.aspx](http://portal.fei.edu.br/pt-BR/pesquisas_projetos/projetos_institucionais/Robo_FEI/futebol_de_robos/Paginas/futebol_de_robos.aspx)>. Acesso em 2 Fev.2014", 2014.
- FUTVASF. **O que é futebol de robôs**. Disponível em: <<http://www.univasf.edu.br/~gari/futvasf/paginas/oQueEOfutebolDeRobos.html>>. Acesso em 23 Fev. 2013, 2008.
- GAVRILOVA, M., R. J. **Reliable line segment intersection testing**. In: Computer Aided Design 32, p. 737–745, 2000.
- FARIA, G.; TEIZEN, L. C. F. ; ROMERO, R. A. F. Controle de trajetórias utilizando campos potenciais aplicado ao futebol de robôs. 2001.
- FARIA, G.; ROMERO, R. A. F.; PRESTES, E. ; IDIART, M. A. P. *Comparing harmonic functions and potential fields in the trajectory control of mobile robots*. p. 762–767, 2004.

- HAN, M.; KOPACEK, P. Neural networks for the control of soccer robots. 2000.
- HECKER, C. *Behind the screen - Physics, Part 3: Collision Response. Game Developer Magazine*. fevereiro/março. Disponível em: <[http://chrishecker.com/Rigid\\_body\\_dynamics](http://chrishecker.com/Rigid_body_dynamics)>. Acesso em 4 Fev. 2013, 1997.
- KITANO, H.; KUNIYOSHI, Y.; NODA, I.; ASADA, M.; MATSUBARA, H. ; OSAWA, H. *Robocup: A challenge problem for AI*. 1997.
- KOH, K. C.; CHO, H. S. A path tracking control system for autonomous mobile robots: an experimental investigation. 1994.
- Mackworth, A. *On Seeing Robots*. 1992.
- MELO, D. O simulador 3d para futebol de robôs uspd. 2009.
- RIEDMILLER, M.; GABEL, T.; PRESTES; HAFNER, R. ; LANGE, S. *Reinforcement learning for robot soccer*. p. 55–73, 2009.
- HISTORY. *RoboCup Soccer*. Disponível em: <<http://www.robocup.org/about-robocup/a-brief-historyof-robocup>>. Acesso em 23 Fev. 2013, 2010.
- SCHULTER, F. Simulador de uma partida de futebol com robôs virtuais. trabalho de conclusão do curso de ciência da computação. 2007.
- SUN, Z. *A Real-time Precrash Vehicle Detection System*. 2002.
- XBOT. **Futebol de robôs**. Disponível em: <<http://www.xbot.com.br/futebol-de-robos/>>. Acesso em 23 Fev. 2013, 2009.
- YAMAMOTO, M.; MEDEIROS, A.; PEDROSA, D. ; ALSINA, P. **Um simulador dinâmico para mini-robôs móveis com modelagem de colisões**. In: VI Simpósio Brasileiro de Automação Inteligente, 2003.
- YAMAMOTO, M.; MEDEIROS, A.; PEDROSA, D. ; ALSINA, P. Planejamento cooperativo de tarefas em um ambiente de futebol de robôs. 2005.
- ZHAO, Y., B. S. Kinematics, dynamics and control of wheeled mobile robots. p. 01–06, 1992.