



SPOLM2007

ISSN 2175-6295

Rio de Janeiro- Brasil, 07 e 08 de novembro de 2007.

APLICAÇÃO DE UM MODELO MATEMÁTICA EM UMA EQUIPE DE FUTEBOL DE ROBÔS

Marco Antonio Firmino de Sousa

Instituto Militar de Engenharia

Praça General Tibúrcio, n° 80, Praia Vermelha, Rio de Janeiro - RJ, Brasil
sousamaf@yahoo.com.br

Júlio César Silva Neves

Instituto Militar de Engenharia

Praça General Tibúrcio, n° 80, Praia Vermelha, Rio de Janeiro - RJ, Brasil
sneves@de9.ime.eb.br

Paulo Fernando Ferreira Rosa

Instituto Militar de Engenharia

Praça General Tibúrcio, n° 80, Praia Vermelha, Rio de Janeiro - RJ, Brasil
rpaulo@de9.ime.eb.br

Alexandre Tadeu Rossini da Silva

Instituto Militar de Engenharia

Praça General Tibúrcio, n° 80, Praia Vermelha, Rio de Janeiro - RJ, Brasil
arossini@gmail.com

Resumo

Este trabalho apresenta os conceitos básicos do desafio proposto pela Liga Simulada 2D da RoboCup, que fornece um ambiente dinâmico e desafiador para simulação de partidas virtuais de futebol de robôs. Além de descrever o modelo matemático proposto de uma equipe para este desafio. É realizado a implementação e testes da solução em questão, que possui a finalidade de otimizar a melhor direção para condução da bola.

Palavras-Chaves: Modelagem Computacional; Futebol de Robôs.

Abstract

This paper presents the basic concepts of the challenge propounded for The RoboCup Simulator League (2D), that supplies a simulation environment dynamic and challenger of virtual game of robots soccer. Besides describing the considered mathematical model of a team to stimulating task. It is fulfilled the implementation and tests of the solution, that possess the purpose to optimize the best direction for conduction of the ball.

Keywords: Computational Model; Robot Soccer.

1. INTRODUÇÃO

O trabalho cooperativo entre robôs é uma área de pesquisa que tem atraído muita atenção nos últimos anos. As equipes robóticas com múltiplos robôs fornecem vantagens

sobre sistemas de um único robô. Por exemplo, permitem que uma área seja coberta mais eficientemente e são mais tolerantes às falhas individuais dos robôs [4]. Vejamos a descrição da RoboCup retirada diretamente do site da Organização RoboCup (www.robocup.org):

RoboCup é uma iniciativa internacional de pesquisa e educação. É uma tentativa de promover a Inteligência Artificial - IA e a investigação em robótica inteligente fornecendo um problema padrão onde uma ampla diversidade de tecnologias possam ser integradas e estudadas, bem como ser usada como parte em projetos de orientação educacional.

Ao mesmo tempo em que o âmbito da RoboCup aumentou desde o princípio em 1997, as principais atividades permanecem, as ligas de futebol de robôs: *Small Size*, *Middle Size*, *Four-legged*, Humanóide e Simulação. A liga *Four-legged* consiste em *Sony Aibos* - cachorros robôs - programados para jogarem futebol. A liga Humanóide ainda está em sua infância; o principal objetivo desta liga é demonstrar as habilidades básicas, tais como levantar, caminhar, e chutar a bola, com um robô duas-pernas. As duas ligas *Small Size* e *Middle Size*, exigem a construção real dos robôs que competem na partida de futebol. A diferença essencial entre essas duas ligas, contrariando o que os nomes sugerem, são as tecnologias de detecção. Equipes da liga *Middle Size* são forçadas a usar visão local embarcada, enquanto aos competidores da liga *Small Size* é permitido utilizar sistemas de visão global (além do sistema de visão local, se assim desejarem). A liga de Simulação, como o nome subentende, não envolve material físico, exceto por computadores, seu principal foco é o desenvolvimento de software para cooperação e inteligência artificial.[14]. Na figura 1 é possível visualizar o ambiente robótico para partidas de futebol nas categorias Simulada 2D e *Small Size* em ação.



Figura 1: Tela de visualização do ambiente simulado da liga 2D (esq.). Liga *Small Size* (F-180) durante uma partida (dir.).

Este artigo descreve a estratégia desenvolvida com apoio de modelagem matemática para um time da Liga Simulada 2D. A ênfase deste artigo está no fato de não utilizar técnicas de inteligência artificial, o que o distingue de muitas outras equipes concorrentes, com o intuito de testar uma nova abordagem. Além desta introdução, este trabalho apresenta na Seção 2 uma breve descrição de Sistemas Multiagentes (SMA), na Seção 3, uma explanação da Liga Simulada 2D da RoboCup, em seguida, na Seção 4 é apresentada a proposta deste trabalho, com o modelo matemático e solução abordada, onde é descrito os detalhes da implementação na Seção 5, seguido dos testes realizados na Seção 6, e por fim, os comentários finais na Seção 7.

2. SISTEMAS MULTIAGENTES

Desde o início da década de 1970, os estudos que tentam resolver problemas de modo cooperativo vêm crescendo rapidamente dentro da área de Inteligência Artificial (IA). Desde então, surgiu o termo Inteligência Artificial Distribuída (IAD), uma sub-área da IA, na qual são combinadas as funcionalidades dos sistemas distribuídos com técnicas de IA [15]. IAD trás idéias e conceitos científicos multidisciplinares, incluindo áreas como ciência da

computação, sociologia, economia, organização, gerência, filosofia e entre outras [8] apud [13]. Enquanto a Inteligência Artificial Clássica tem ênfase na representação do conhecimento e métodos de inferências, a IAD baseia seu modelo de inteligência no comportamento social (cooperação), sendo a ênfase transposta para as ações e interações entre entidades denominadas agentes [2]. A IAD acrescenta à Inteligência Artificial Clássica, que tem como metáfora o comportamento humano individual, a metáfora do comportamento social [8].

Conforme Coser [1], as definições de agente encontradas na literatura são geralmente baseadas nas propriedades que diferem um agente de um software convencional. No entanto, é praticamente consenso que características como autonomia, comunicabilidade e reatividade são altamente desejáveis, senão obrigatórias, para que um software desempenhe o papel de agente. A seguir tem-se a descrição de algumas propriedades de agentes.

- Autonomia: agentes devem ser capazes de realizar a maior parte de suas tarefas sem a intervenção direta de humanos ou de outros agentes e devem possuir um grau de controle sobre suas próprias ações e sobre seu estado interno [12].
- Comunicabilidade e Habilidade Social: os agentes devem ter a capacidade de interagir com os usuários e com outras entidades de software, visando resolver os problemas associados às suas próprias tarefas e também auxiliar outros agentes quando for viável [12].
- Reatividade: um agente deve ser capaz de monitorar seu ambiente para conseguir realizar suas tarefas autonomamente [4].

Para resolver o problema do futebol de robôs, o que será caracterizado mais a frente, os sistemas multiagentes (SMA) são utilizados por oferecerem uma solução mais natural e eficiente. SMA é, basicamente, um ambiente povoado por agentes. Em um SMA cada agente pode possuir um objetivo e ações particulares. Quando vários agentes possuem um objetivo global comum, estes são denominados equipe ou time, passando a pertencer a uma sociedade [4].

3. LIGA SIMULADA 2D DA ROBOCUP

RoboCup é um excelente *'test bed'* para desenvolvimento de novas ferramentas e técnicas para sistemas de controle autônomo em ambientes incerto e dinâmico. Em uma perspectiva educacional, é também um grande meio para expor estudantes à projetar, construir, gerenciar, e manter sistemas complexos [14].

A liga de simulação da RoboCup possui simulador (chamado *Soccer Server*) para realização de partidas virtuais, fornecido pela RoboCup e facilmente encontrado no sítio oficial da RoboCup da Liga Simulada (<http://sserver.sourceforge.net/>). Cada partida é composta por dois tempos de 3000 ciclos, sendo que cada ciclo possui 100ms; assim, uma partida possui 600s no total, dividido em dois tempos de 300s. Todavia, vale ressaltar que essa categoria é especialmente voltada para a aplicação de Sistemas Multiagentes (SMA) e coordenação de agentes cooperativos [10].

O *Soccer Server* é um ambiente em tempo real, com restrição temporal, que possibilita a realização de partidas de futebol entre duas equipes compostas de 11 agentes autônomos e, eventualmente, um treinador, que comunicam com um servidor em uma arquitetura cliente/servidor utilizando *sockets* UDP, não orientado a conexão. O ambiente foi implementado para ser inacessível, dinâmico, não determinístico, suscetível à falhas e impor elevadas limitações às capacidades de comunicação dos agentes. Em uma partida de futebol real, essas características são encontradas. Por exemplo, um jogador não é capaz de ter visão completa do jogo e, muito menos, de comunicar-se oralmente com jogadores do outro lado do campo, considerando a distância, e, provavelmente, a todo o barulho da torcida. As formas de

comunicação são abordadas logo em seguida. Este ambiente simulado foi construído para produzir as dificuldades que os futuros robôs reais (físicos) irão encontrar.

O ambiente de simulação da RoboCup é composto, basicamente, por: servidor (*Soccer Server*), visualizador (*Soccer Monitor*), que é utilizado somente havendo necessidade de visualização do jogo, e o registro de jogo (*LogPlayer*), para que possa ser exibido posteriormente as partidas ou utilizá-las para análises. Lembrando que, os clientes não fazem parte do simulador fornecido pela RoboCup e suas implementações ficam a cargo das equipes. A arquitetura de simulação da RoboCup é demonstrada na Figura 2.

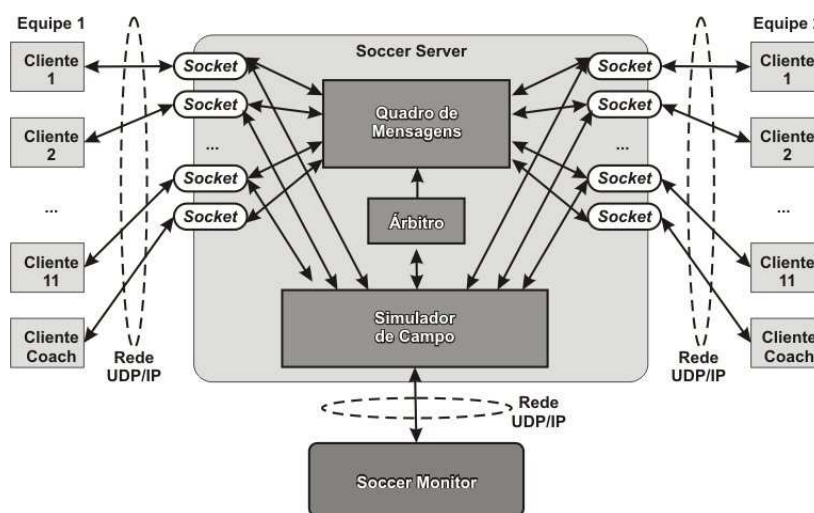


Figura 2 Arquitetura do Ambiente Simulado [10].

O servidor (*Soccer Server*) é constituído por três grandes módulos, que são:

- Quadro de mensagens - controla a comunicação de mensagens entre o servidor e os clientes;
- Árbitro - assegura que as regras do jogo sejam cumpridas;
- Simulador de campo - calcula o movimento dos vários agentes e trata as colisões no campo.

A comunicação dos agentes e o simulador é um ponto muito importante. Através da comunicação são realizadas todas as interações entre agentes e o simulador. Um agente não se comunica diretamente com outro agente, todas as trocas de mensagens são realizadas através do simulador Soccer Server. Essa é uma das limitações existentes no servidor, uma baixa largura de banda disponibilizada no domínio para que todos os agentes se comuniquem. Essas restrições impostas à comunicação, tornam o futebol de robôs simulado particularmente atrativo para a aplicação de coordenação por comunicação, modelos mútuos, comunicação otimizada e visão inteligente [10].

Dentre as formas de comunicação, há três tipos de mensagens possíveis entre os agentes jogadores e o servidor do ambiente simulado. Mensagens de conexão, percepção e atuação. Para estabelecer a conexão entre o servidor e os agentes, deve-se seguir as orientações contidas no manual do *Soccer Server*. Os clientes devem ter a capacidade de comunicar-se com o servidor utilizando o protocolo UDP/IP.

Toda comunicação da percepção é feita através de mensagens, das quais existem três tipos principais entre o servidor e os agentes:

- Percepção Auditiva (*hear*) - Utilizado para comunicação entre os agentes, que obrigatoriamente, comunicam-se através do servidor.
- Percepção Visual (*see*) - Possibilita que os agentes visualizem o ambiente, a bola e os outros agentes jogadores e suas distâncias. De fundamental importância para o futebol de robôs simulado.

- Percepção Sensorial (*sense_body*) - Permite que os agentes tenham os sinais do servidor para controle.

Cada agente deve ser capaz de interpretar as mensagens recebidas de maneira que possa atuar no ambiente, independente de quais tipos forem as mensagens. Para isso, existem definições para as ações dos clientes. Todas as definições existentes são encontradas em [5], com os formatos das mensagens aceitáveis pelo servidor. Os comandos são simples e a maioria não implica em uma resposta do servidor.

Os comandos de atuação são divididos em 4 tipos principais: movimento (*dash*, *turn* e *move*); interação com a bola (*kick*, *tackle* e *catch*); controle da percepção (*turn_neck*, *attentionto*, *change_view*) e comunicação (*say*, *pointto*). Uma restrição importante do ambiente é que um agente não pode enviar, em um mesmo ciclo, mais de uma ação de movimento (comando *move*) ou interação com a bola (*interceptar*, *chutar*). Complementarmente, ocorrendo o envio de mais de uma ação nos casos supracitados, o servidor seleciona apenas um comando para ter efeito.

Todas essas mensagens são designadas de baixo nível. São as interações mais simples possíveis no ambiente simulado, a partir dessas são construídas as outras interações como driblar, roubar bola, efetuar lançamento, tocar bola para companheiro, localizar adversários e localizar a bola.

4. PROPOSTA

O trabalho de Silva [4] é proposto um modelo estratégico separado por objetivos global e local, o qual nos serviu de inspiração para criar um modelo tático semelhante. Assim considerando, temos que o objetivo global é responsável por determinar a estratégia adotada pela equipe, enquanto o objetivo local fica encarregado das ações a serem tomadas.

Como parâmetros para o objetivo global são considerados a distância euclidiana dos jogadores da equipe controlada em relação à bola, que servem para definir o estado de jogo da equipe, os quais são: ataque ou defesa.

Ribeiro [11] considera um esquema de posicionamento defensivo para Liga Small Size (F-180) onde são abstraídas as táticas e habilidades dos jogadores. O ponto forte desse modelo consiste no posicionamento dos jogadores controlados que impedem o ataque dos adversários.

Para este trabalho, com o intuito de estender o de Ribeiro [11], propõe-se um modelo matemático para direcionar uma equipe ao ataque. Fechando assim, um modelo tático mais complexo, portando de estratégias locais especializadas para o momento do jogo que serão escolhidas obedecendo a um objetivo global.

O modelo matemático de ataque tem como objetivos principais as seguintes táticas:

- Minimizar a distância da bola ao gol;
- Maximizar a distância da bola dos jogadores adversários.

Em ambos é utilizada a distância euclidiana. Como restrição para este problema, há a limitação da bola permanecer em campo e também de que apenas um jogador do time fique com domínio de bola.

4.1. PROGRAMAÇÃO MATEMÁTICA

No mundo ao nosso redor é raro concentrar qualquer problema em simples variáveis. Geralmente muitos parâmetros têm que ser analisados antes de encontrar uma solução adequada [16]. Simplificadamente, os dois objetivos foram agrupados na expressão da função objetiva abaixo:

$$MinZ = \sum_{i=1}^{i=11} \frac{\sqrt{(X - Xg)^2 + (Y - Yg)^2}}{\sqrt{(X - Xoi)^2 + (Y - Yoi)^2}} \quad (1)$$

Onde:

X representa a coordenada do eixo X da bola;

Y representa a coordenada do eixo Y da bola;

Xg representa a coordenada do eixo X do gol, considerado como ponto 0 para este problema;

Yg representa a coordenada do eixo Y do gol, considerado como ponto 0 para este problema;

Xo_i representa a coordenada X do robô i ;

Yo_i representa a coordenada Y do robô i .

Sujeito a:

$$\sum_{i=1}^{i=10} Wi = 1 \quad (2)$$

$$Se \sum_{i=1}^{i=10} \sqrt{(X - Xai)^2 + (Y - Yai)^2} < 1 \rightarrow Wi = 1 \quad (3)$$

$$MINX < X < MAXX \quad (4)$$

$$MINY < Y < MAXY \quad (5)$$

Onde:

Xa_i representa a coordenada do eixo X do jogador companheiro i ;

Ya_i representa a coordenada do eixo Y do jogador companheiro i ;

$MINX$ e $MAXX$ representam os limites inferior e superior de X respectivamente;

$MINY$ e $MAXY$ representam os limites inferior e superior de Y respectivamente.

Nas expressões 2 e 3 é feita a consideração de que apenas um jogador da equipe deve permanecer com a bola. Evitando que todos procurem ficar próximos à bola e assim não ocorrendo um aglomerado de jogadores. Já nas expressões 4 e 5, estas impedem que a bola saia de campo.

Para otimizar a solução proposta é utilizado o método do gradiente, que fornece um vetor cuja direção indica a direção na qual o campo escalar aumenta mais rapidamente [9]. Para tanto, adequando ao problema em questão, um vetor em direção contrária ao do gradiente é utilizado, pois este indica a direção onde minimizará a expressão 1.

5. IMPLEMENTAÇÃO

Para validar a solução proposta foi utilizado o código básico disponível pela equipe UvaTrilearn [6], visto não ser o objetivo deste trabalho desenvolver toda uma equipe. Assim, o método do gradiente foi implementado como função para guiar a direção de cada jogador sendo o módulo de alto nível para o código básico da equipe. Como linguagem para desenvolvimento foi utilizada C++ em ambiente GNU Linux.

O método supracitado retorna o ângulo com a melhor direção em relação ao jogador que possui a bola, segundo a função objetiva. Este cálculo é realizado a cada novo ciclo do ambiente. Escolheu-se herdar o esquema de posicionamento da equipe UvaTrilearn [6] por ter qualidades adequadas à sua função.

Uma consideração importante a ser feita é o fato de um jogador em campo não possuir uma visão global do ambiente. Este fenômeno impediria a utilização do método, visto

exigir o posicionamento de cada jogador oponente em campo. Para solucionar este detalhe utilizou-se da última posição vista de cada jogador, esta é atualizada a cada novo ciclo. Caso não ocorra atualização a anterior é preservada. Para a hipótese de não existir nenhuma informação a respeito de um oponente é criada uma posição aleatória dentro de campo, porém, fora do campo de visão do jogador corrente.

6. TESTES

O ambiente simulado disponível pela RoboCup, melhor explanado na Seção 3, foi instalado para realização dos teste. Várias sessões com partidas completas ocorreram contra a equipe campeã de 2003, UvaTrilean. Vale ressaltar que a diferença entre o código básico e a equipe campeã é o fato da segunda possui o módulo de alto nível desenvolvido que permite planejamento tático e estratégico de jogo.

Os resultados das partidas estão na Tabela 1 e um arquivo de log de cada uma delas foi gerado para análise futura. A Tabela 1 contém apenas os resultados das partidas com a melhor configuração da equipe.

Uma descrição de como é o funcionamento de um jogo é encontrado na Seção 3. A princípio, a equipe, apesar de perder o jogo, não está em situação de má qualidade, visto que este mesmo time campeão de 2003 chegou a vencer por 39x0 a equipe *Iranians* (Iran) na primeira fase do torneio de 2003 da RoboCup. É possível notar que após alguns refinamentos o time conseguiu uma evolução. Entretanto, ainda há muito espaço para melhorias e afinamentos no código. Mas algumas conclusões sobre o método utilizado foram alcançadas, estas estão melhores relatadas na Seção 7.

Tabela 1 - Resultados de testes.

Partida	UvaTrilean 2003 vs IMEUva
1 ^a	1x0
2 ^a	2x0
3 ^a	0x3

7. CONCLUSÃO E TRABALHOS FUTUROS

Ao longo deste trabalho foram apresentados conceitos de Sistemas Multiagentes, Liga Simulada 2D da RoboCup, seu ambiente simulado e o modelo matemático desenvolvido para uma equipe desta Liga. Muitas equipes são desenvolvidas com abordagens voltadas puramente para Inteligência Artificial (IA), o modelo proposto demonstra ser eficiente e de fácil utilização para servir de dado de entrada para métodos de IA, em um sistema híbrido.

Entretanto, ainda não esgotada as possibilidades de desenvolvimento deste modelo para a Liga Simulada 2D, pretende-se como trabalho futuro expandir o modelo matemático para otimizar o goleiro e ainda efetuar novas formas de tratar este modelo proposto dando enfoque maior ao toque de bola aos companheiros de equipe e fornecer uma função de avaliação para saber quando a melhor hora para chutar a bola e conduzi-la.

Além de incorporar nesta equipe o modelo de defesa proposto por Ribeiro [11], adaptando o esquema de divisão em quadros por distância em raio de atuação.

Vale ressaltar que o modelo proposto é melhor implementado na Liga F-180 por possuir uma visão global do ambiente de jogo e controle centralizado [4], porém faz-se necessário a construção de não uma, mas de duas equipes de robôs para validar a solução proposta.

8. REFERÊNCIAS BIBLIOGRÁFICAS

- [1] Coser A and Gauthier F. A. O. Utilização de agentes inteligentes no trabalho colaborativo via internet. Master's thesis, Universidade Federal de Santa Catarina, 1999.
- [2] Rocha A. Introdução aos agentes inteligentes e aos sistemas multiagentes. Master's thesis, Universidade Federal de Lavras - UFLA, Lavras, 2003.
- [3] Alper K. C. and Colin G. H. Agent sourcebook. John Wiley & Sons, Inc, 1997.
- [4] Silva Alexandre Tadeu Rossini da and Rosa Paulo Fernando Ferreira. Comportamento Social Cooperativo na Realização de Tarefas em Ambientes Dinâmicos e Competitivos. Master's thesis, Instituto Militar de Engenharia, February 2006.
- [5] ChenMet al. RoboCup Soccer Server for Soccer Server Version 7.07 and later. RoboCup, 2005.
- [6] Kok J.R. Uva trilearn 2003 - soccer simulation team. Technical report, University of Amsterdam, 2003.
- [7] Gasser L. Distribution and coordination of tasks among intelligent agents. In JCAI, pages 189–204. Scandinavian Conference on AI, 1988.
- [8] Wooldridge M. Multiagent systems: A modern approach to distributed artificial intelligence. pages 27–78. The MIT Press, 1999.
- [9] Munem M. A. and Foulis D. J. Cálculo, volume 2. LTC Editora, 1982.
- [10] Reis L. P. Coordenação em sistemas multi-agente: Aplicações na gestão universitária e futebol robótico. Mestrado, FEUP, Julho 2003.
- [11] Ribeiro M. P. and Neves J. C. S. Modelagem matemática de um jogo defensivo para robocup f-180 small-size league. SPOLM2006, Agosto 2006.
- [12] Jennings N. R. and Wooldridge M. J. Software agents. IEEE Review, pages 17–20, 1996.
- [13] Sun R. Cognition and multi-agent interaction from cognitive modeling to social simulation. 2005.
- [14] Raffaello D'Andrea. The Cornell RoboCup Robot Soccer Team: 1999 - 2003. Technical report, Cornell University, 2005.
- [15] Botelho W. T. and Rosa P. F. F. Um sistema de identificação e adaptação pervasivo para a casa inteligente utilizando sistemas multiagentes. Dissertação, Instituto Militar de Engenharia. 2005.
- [16] Coello Carlos Artemio Coello. An Empirical Study of Evolutionary Techniques for Multiobjective Optimization in Engineering Design. PhD Thesis. Tulane University. April 1996.