

iBots 2015: Descrição do Time

Alexandre T. R. Silva, Luis A. V. Carvalho, Tanilson D. Santos,
Herinson B. Rodrigues, and Nilo S. Marques

Resumo— Este trabalho apresenta soluções desenvolvidas pela equipe iBots na categoria *RoboCup Simulation 2D*. Descreve o processo de tomada de decisão dos jogadores quando estão com posse da bola, o algoritmo de planejamento de trajetória e o sistema defensivo implementado. Adicionalmente, este trabalho contém testes e os resultados também são apresentados. A equipe iBots usa como time-base o Agent2D 3.1.1.

I. INTRODUÇÃO

Este artigo descreve as soluções implementadas pela equipe iBots, na categoria *RoboCup Simulation 2D*. A equipe iBots foi criada no segundo semestre de 2009 na Universidade Federal do Tocantins. Nesse ano, a partir do time-base UvATrilearn, foi desenvolvida a equipe iBots, que no ano seguinte debutou sua participação em competições. A equipe iBots participou da LARC 2010 [1], em São Bernardo do Campo-SP, e foi eliminada na primeira fase. Para 2011, na Competição Brasileira de Robótica em São João Del Rei-MG, a equipe iBots deixou de utilizar o UvATrilearn, como time-base, e passou a adotar o Agent2D [2] e conquistou a terceira colocação. Desde então, só não participou de competições em 2014. A versão 2015 continua baseada no time-base Agent2D (atualmente utiliza a versão 3.1.1), que depende da biblioteca librcsc. A versão utilizada da librcsc foi a 4.1.0.

II. TIMES-BASE

Alguns times da categoria de *Simulation 2D* da *RoboCup* fornecem uma versão básica de seu próprio código fonte, disponibilizado sob licença livre. Essas versões são chamadas times-base e são utilizadas por outras equipes como frameworks. A utilização de times-base fornecidos por outras equipes permite que o foco do trabalho se concentre no comportamento dos agentes e na estratégia a ser utilizada, uma vez que não é necessário se preocupar com o desenvolvimento de habilidades básicas do agente (percepção espacial, chute, etc.) e com a comunicação com o servidor via conexão UDP ou com a interpretação das mensagens recebidas pelo servidor. Os frameworks mais conhecidos, atualmente, são UvATrilearn, Agent2D e WE-Base.

O UvATrilearn, desenvolvido pela Universidade de Amsterdã [3], é um time-base escrito em C++ com farta documentação, o que o torna bastante utilizado como ponto de partida por muitas equipes iniciantes. É distribuído sob a licença GPL (*General Public Licence*), no entanto ele não é atualizado desde 2003. Por isso, não incorpora recursos surgidos posteriormente no simulador da *RoboCup*. Neste trabalho é importante usufruir dos recursos existentes na versão

mais atual do simulador uma vez que eles surgiram para tornar o simulador com uma modelagem mais próxima à do futebol de humanos.

O Agent2D é também distribuído sob licença GPL e foi desenvolvido em C++ por Hidehisa Akiyama [4]. Esse time-base é atualizado com certa frequência e possui mais ampla cobertura com relação às habilidades básicas se comparado ao UvATrilearn, o que permite ao desenvolvedor centrar seu trabalho na construção de habilidades mais complexas. Ao contrário do framework UvATrilearn, a documentação do Agent2D é escassa em inglês e se restringe a algumas poucas em japonês (incluindo um manual), o que dificulta o entendimento da estrutura de seu funcionamento.

O time WE-Base é escrito em C++ e pertence ao Laboratório Sistemas Multi-agente da Universidade de Ciência e Tecnologia da China. É a escola oficial da equipe WrightEagle, que foi criada em 1999 e ganhou quatro campeonatos mundiais. Tem bastante habilidades básicas implementadas, mais do que o UvATrilearn mas menos do que o Agent2D. Por isso, é um time-base menos usado do que o Agent2D. A documentação do WE-Base é bastante escassa.

III. SOLUÇÕES DESENVOLVIDAS

a. Tomada de Decisão dos Jogadores

O algoritmo de tomada de decisão desenvolvido para este trabalho foi baseado na seguinte política: “se o jogador puder chutar a gol, então ele deve chutar; senão, passar a bola a alguém que possa chutar; senão, passar a bola a alguém que possa passar a alguém que possa chutar; senão...” [5]. Desse modo, a solução é recursiva e, para identificar jogadores melhores posicionados, precisou definir critérios como posicionamento dos jogadores em campo e da qualidade da posição para chute/passe.

O posicionamento ofensivo de um jogador em campo é quantificado pelo método *DistanciaDoGol()*. Esse método tem como parâmetro um atributo da classe *Vector2D*, que recebe como argumento a posição do jogador que está sendo avaliado, utilizada para calcular sua distância em relação ao gol adversário. O método *DistanciaDoGol()* considera como gol adversário a posição 52.5 do eixo x e em relação ao eixo y o gol adversário é discretizado em 15 pontos, separados em uma unidade, e tem como centro a posição 0 (zero). Em suma, isso significa que os pontos verificados pelo método *DistanciaDoGol()* são: (52.5, -7), (52.5, -6), (52.5, -5), (52.5, -4), (52.5, -3), (52.5, -2), (52.5, -1), (52.5, 0), (52.5, 1), (52.5, 2), (52.5, 3), (52.5, 4), (52.5, 5), (52.5, 6) e (52.5, 7).

A. T. R. Silva e T. D. Santos, são professores da Universidade Federal do Tocantins, Caixa Postal 266, Cep 77001-090, Palmas-TO, Brasil (e-mail: arossini@uft.edu.br; tanilsondiasdossantos@gmail.com).

L. A. V. Carvalho, é professor do Instituto Alberto Luiz Coimbra de Pós-Graduação e Pesquisa de Engenharia da Universidade Federal do Rio de

Janeiro, Cx. Postal 68511, Cep 21941-972, Rio de Janeiro-RJ, Brasil (e-mail: LuisAlfredo@ufrj.br).

R. B. Rodrigues e N. S. Marques, são aluno da Universidade Federal do Tocantins, Caixa Postal 266, Cep 77001-090, Palmas-TO, Brasil (e-mail: herinson.br@gmail.edu.br; nilobohn@gmail.com).

Vide figura 1. Assim, quanto menor a distância de um companheiro em relação a um dos 15 pontos considerados do gol adversário, melhor é o posicionamento ofensivo do jogador.

Para a qualidade da posição para chute/passe, é utilizado o mesmo algoritmo descrito na seção 5.8.2 acerca da realização do passe. O jogador não considerará para chute/passe um companheiro que:

- está sendo bem marcado, determinado por uma Rede Neuronal Artificial Perceptron de Múltiplas Camadas previamente treinada;
- houver algum adversário com condições de interceptar o chute/passe, para isto verifica se há algum oponente na área compreendida por 20 graus a mais e a menos do ângulo formado entre o ponto candidato e a bola e algum oponente estiver mais próximo da bola do que o ponto candidato.

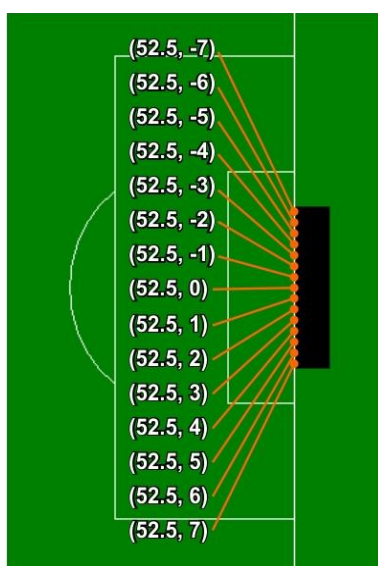


Figura 1 - Pontos considerados pelo método *DistanciaDoGol()* para quantificar o posicionamento de um companheiro de equipe

b. Planejamento de Trajetória

A técnica adotada neste trabalho foi o campo potencial artificial [6], que tem como princípio fundamental a movimentação do robô sob um campo de forças artificiais, geradas pelos obstáculos e pelo alvo. O potencial, seu gradiente, deve ser contínuo. Os obstáculos (outros robôs) e o objetivo (uma posição no campo de jogo) geram campos de repulsão e de atração, respectivamente, obtendo um movimento (seguindo o gradiente) através do qual os obstáculos são evitados e espera-se que o robô atinja seu objetivo. Essas forças são oriundas de uma função chamada de potencial artificial.

O campo potencial artificial é muito popular entre os pesquisadores de Robótica e são comumente adotados no futebol de robôs [7] [8] [9] por ser de baixo custo computacional, algo bastante interessante para aplicações de futebol de robôs, e ser bastante eficiente em ambientes com restrição temporal, quando comparado a outros métodos.

No algoritmo de trajetória implementado, todos os jogadores, adversários ou companheiros de equipe, são

tratados como obstáculos (campos de repulsão) e o objetivo (campo de atração) varia de acordo com a função de cada jogador de acordo com o cenário. Note, na figura 2, que os obstáculos são contornados, o que evita colisões (encontros faltosos) entre jogadores

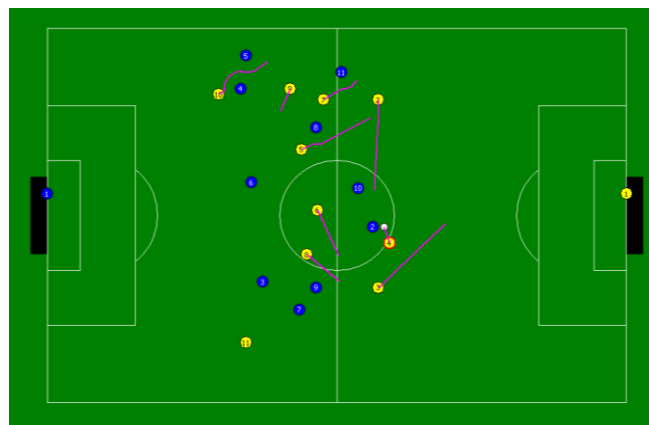


Figura 2 - Exemplo de trajetórias planejadas pelo método Campo Potencial Artificial

Vale atenção especial para a trajetória do jogador dez da equipe amarela (figura 3). Note que para chegar ao ponto de atração, que é marcar o jogador cinco da equipe azul, ele primeiro desvia do jogador quatro e depois desvia do jogador cinco para conseguir chegar ao objetivo.

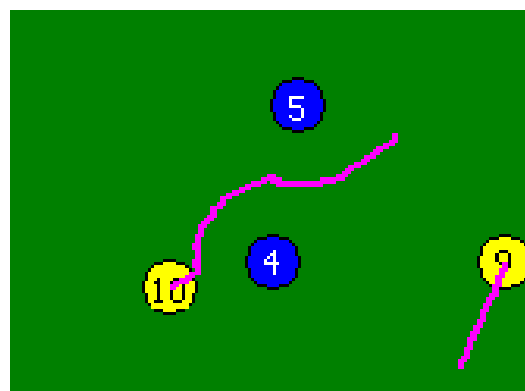


Figura 3 - Destaque ampliado da trajetória do jogador dez da equipe amarela para o cenário da Figura 2

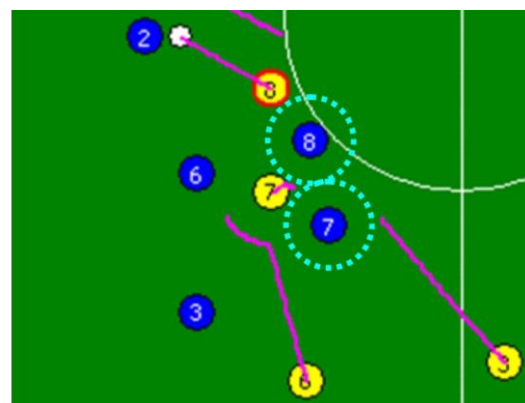


Figura 4 - Exemplo de mínimo local no cálculo do campo potencial artificial

O problema mais conhecido da abordagem por campo potencial é a possibilidade de convergência do movimento para regiões em estado de potencial mínimo local. Um exemplo de mínimo local é apresentado na figura 4. O jogador sete da equipe amarela tem como objetivo marcar o jogador oito da equipe azul. Para isso, a trajetória determinada pelo campo potencial artificial força a passagem por entre os jogadores oito e sete adversários, o que fica impossibilitada uma vez que o campo de repulsão, em destaque tracejado, impede isso. Todavia, há diversas heurísticas propostas para solucionar o problema dos mínimos locais.

c. Sistema Defensivo

Foram desenvolvidas duas estratégias, chamadas: estratégia defensiva 1 e estratégia defensiva 2. Cada estratégia tem o objetivo de explorar situações específicas do jogo quando a equipe estiver sem a posse de bola.

A estratégia defensiva 1 visa defender em unidade ou em bloco, reduzindo o espaço de jogo, com diminuição da amplitude defensiva da equipe na sua largura e profundidade. Assim, a equipe que se defende em bloco reduz os espaços livres que o time adversário poderia utilizar para dar penetração às suas ações ofensivas e, com isto, conseguir se defender de forma mais eficiente.

Já estratégia defensiva 2 foi desenvolvida com o objetivo de induzir a equipe adversária a conduzir a bola para zonas menos perigosas de jogo

IV. TESTES E RESULTADOS

Na rede neuronal e no sistema de inferência fuzzy foram aplicados testes a fim de mensurar a capacidade de indentificar as diferentes situações do jogo quando a equipe estiver sem a posse de bola. Uma base de dados foi montada a partir situações preestabelecidas. Da base de dados utilizada, 60% dela foi destinada para treinamento e 40% para validação. Após treinada a rede neuronal e após serem extraídas as regras fuzzy com a base de treinamento, foi fornecida a base de validação com dados pré-processados.

Os resultados dos testes com a rede neuronal e com o sistema de inferência fuzzy indicaram que a rede neural respondeu mais adequadamente com 90.00% de acertos; o sistema de inferência fuzzy acertou cerca de 81,25% dos casos. Assim, a rede neuronal perceptron de múltiplas camadas foi adotada na equipe iBots.

Posteriormente, a equipe iBots foi colocada em confronto contra as equipes Agent2D 3.1.1, WrightEagle 2009 e FC Portugal 2010. Os resultados sob a perspectiva da equipe iBots estão expressos na tabela I.

TABELA I. RESULTADOS DAS PARTIDAS REALIZADAS PELA EQUIPE IBOTS

	VITÓRIAS	EMPATES	DERROTAS	PONTOS
AGENT2D	48%	32%	20%	58,67%
FC PORTUGAL	64%	32%	4%	74,67%

A fim de obter um parâmetro para medir o desempenho da equipe iBots, foram realizadas 20 partidas em que a equipe Agent2D 3.1.1, que é o time-base de desenvolvimento da equipe iBots, jogou contra ela mesmo e contra a equipe FC

Portugal. Os resultados dessas partidas estão apresentados na tabela II.

TABELA II. RESULTADOS DAS PARTIDAS REALIZADAS PELA EQUIPE AGENT2D 3.1.1

	VITÓRIAS	EMPATES	DERROTAS	PONTOS
AGENT2D	45%	20%	35%	51,67%
FC PORTUGAL	70%	30%	0%	80,00%

É possível observar nos testes que o desempenho da equipe iBots foi contra o Agent2D 3.1.1 superior do que o desempenho do Agent2D 3.1.1 contra ele próprio. Os pontos conquistados foram 7% superior e as derrotas reduziram drasticamente. Além disso, a equipe iBots fez uma média de 1,52 de gols por partida e sofreu uma média 1,08 de gols por partida, ou seja, uma diferença média de 0,44 superior de gols. Já nos jogos da equipe Agent2D 3.1.1 contra ela mesma, a quantidade de gols feitos e sofridos foi igual, com a média de 2,35 gols por partida, ou seja, não houve diferença de gols.

Contra a equipe FC Portugal 2010, a equipe iBots teve perda de desempenho, ficando com 5,33% de pontos a menos do que os confrontos contra o Agent2D 3.1.1. A equipe iBots fez uma média de 1,22 de gols por partida e sofreu uma média 0,08 de gols por partida, ou seja, uma diferença média de 1,14 superior de gols. Já nos jogos contra a equipe Agent2D 3.1.1, a média de gols feitos foi de 1,2 e de 0,3 gol sofridos em, isto resulta em uma diferença de 1,12 gols por partida. Em relação aos gols marcados e sofridos, o desempenho da equipe iBots e Agent2D 3.1.1 foi semelhante.

O que ajuda a entender o melhor desempenho da equipe iBots contra o time Agent2D 3.1.1 e a piora de desempenho contra a equipe FC Portugal é o mapa de calor da bola nos testes (figura 5).

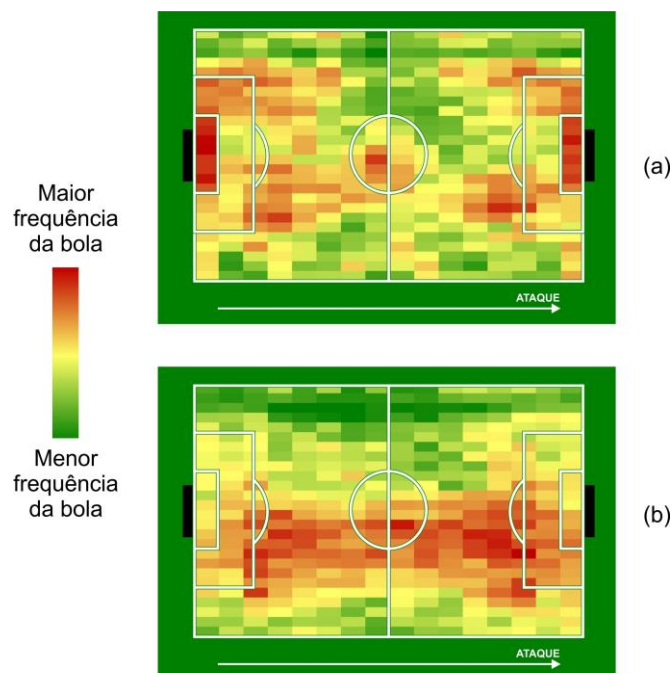


Figura 5 – Mapa de calor da bola nos testes: (a) Agent2D 3.1.1; (b) FC Portugal

Observe que as equipes Agent2D 3.1.1, figura 5a, e FC Portugal, figura 5b, têm características bem distintas de jogo uma vez que a movimentação da bola é muito diferente entre elas.

V. CONSIDERAÇÕES FINAIS

Neste trabalho foram apresentadas as soluções desenvolvidas pela equipe iBots 2015. Foram descritos o processo de tomada de decisão dos jogadores quando estão com posse da bola, o algoritmo de planejamento de trajetória e o sistema defensivo implementado.

No resultado do teste expressos na tabela I e II, constata-se que a equipe iBots melhorou o desempenho contra a equipe Agent2D 3.1.1, mas, por outro lado, piorou em relação à equipe FC Portugal. Isso leva à conclusão de que melhorar o desempenho de uma equipe em relação à outra, não significa melhorar o desempenho contra todas as equipes.

REFERÊNCIAS

- [1] A. T. R. Silva, H. G. Silva, E. G. Santos, G. B. Ferreira, T. D. Santos, and V. S. Silva. iBots 2010: Descrição do time. Latin American Robotics Competition. 2010.
- [2] A. T. R. Silva, G. B. Ferreira, T. D. Santos, V. S. Silva, E. G. Santos, H. G. Silva, C. A. S. P. Rodrigues, and T. S. Arruda. iBots 2011: Descrição do time. Competição Brasileira de Robótica, São João Del Rei, 2011.
- [3] R. Boer, J. Kok. The Incremental Development of a Synthetic Multi-Agent System: The UvA Trilearn 2001 Robotic Soccer Simulation Team. Master's Thesis - Faculty of Science University of Amsterdam, 2002.
H. Akiyama, H. Shimora, and I. Noda, Helios 2009 team description. RoboCup 2009, 2009.
- [4] L. P. Reis. Coordenação em Sistemas Multi-Agentes: Aplicações na Gestão Universitária e Futebol Robótico. Tese de PhD, FEUP, julho de 2003.
- [5] O. Khatib. Real-Time Obstacle Avoidance for Manipulators and Mobile Robots. In International Journal of Robotic Research, v. 5, n. 1, p. 90-98, 1986.
- [6] Y. Nagasaka, K. Murakami, T. Naruse, T. Takahashi, and Y. Mori. Potential Field Approach to Short Term Action Planning in RoboCup F180 League. In RoboCup 2000: Robot Soccer World Cup IV table of contents, 345-350, 2001.
- [7] J. Meyer, R. Adolph, D. Sephan, A. Daniel, M. Seekamp, V. Weinert, and U. VISSER. Decision-making and Tactical Behavior with Potential Fields. In RoboCup 2002: Robot Soccer World Cup VI. Volume 2752 of Lecture Notes in Artificial Intelligence, Springer, 2003.
- [8] A. T. R. SILVA. Comportamento Social Cooperativo na Realização de Tarefas em Ambientes Dinâmicos e Competitivos. Dissertação de mestrado. Rio de Janeiro: Instituto Militar de Engenharia, 2006.