

## BAHIA2D: DESCRIÇÃO DO TIME

HUGO SILVA<sup>1</sup>, JESSICA MEYER<sup>1</sup>, JAILSA OLIVEIRA<sup>1</sup>, DANILO CRUZ<sup>1</sup>, LEONARDO PESSOA<sup>1</sup>, MARCO A. C. SIMÕES<sup>1,2</sup>, HELDER ARAGÃO<sup>2</sup>, RICARDO LIMA<sup>2</sup>

<sup>1</sup> Núcleo de Arquitetura de Computadores e Sistemas Operacionais (ACSO)  
Universidade do Estado da Bahia (UNEB)  
Rua Silveira Martins, 2555, Cabula. Salvador – BA – Brasil

<sup>2</sup> Grupo de Pesquisa em Computação Inteligente (GPCI)  
Centro Universitário da Bahia (FIB)  
Rua Xingu, n.º. 179, Jardim Atalaia/STIEP. Salvador – BA – Brasil

[hugodaluz@gmail.com](mailto:hugodaluz@gmail.com), [jessy.meyer@gmail.com](mailto:jessy.meyer@gmail.com), [jailsa.oliveira@gmail.com](mailto:jailsa.oliveira@gmail.com), [nilofcruz@yahoo.com.br](mailto:dani-<br/>nilofcruz@yahoo.com.br), [leo.pessoa@gmail.com](mailto:leo.pessoa@gmail.com), [msimoes@uneb.br](mailto:msimoes@uneb.br), [der@fib.br](mailto:hel-<br/>der@fib.br), [ricardolima\\_uneb@yahoo.com.br](mailto:ricardolima_uneb@yahoo.com.br)

**Abstract.** This article presents the initial results of the research group Bahia Robotics Team in the category of simulated soccer 2D. In this TDP are presented the team's architecture and the techniques of Artificial Intelligence used in the player's skills: controllers fuzzy and neural networks. The team Bahia2D was tested in matches in the Robocup 2007 qualification, obtaining success and being the unique Brazilian team qualified for the world championship in the category of simulated soccer 2D. The results of the preliminary rounds (SSIL - Soccer Simulation Internet League) and of the participation in Robocup 2007 are presented, analyzed and compared, showing the team's evolution in the short space of time among the two competitions.

**Key Words:** Fuzzy Logic, Neural Networks, Robots' Soccer, Robocup.

**Resumo.** Este artigo apresenta os resultados iniciais do grupo de pesquisa Bahia Robotics Team na categoria de futebol simulado 2D. Neste TDP são apresentados a arquitetura do time e as técnicas de Inteligência Artificial utilizadas nas habilidades dos jogadores: controladores fuzzy e redes neurais artificiais. O time Bahia2D foi testado em partidas nas eliminatórias da Robocup 2007, obtendo êxito ao ser o único time brasileiro classificado para o mundial na categoria de futebol simulado 2D. Os resultados das eliminatórias (A SSIL - Soccer Simulation Internet League) e da participação na Robocup 2007 são apresentados, analisados e comparados, mostrando a evolução do time no curto espaço de tempo entre as duas competições.

**Palavras chave:** Lógica Fuzzy, Redes Neurais, Futebol de Robôs, Robocup.

### 1 Introdução

O Bahia2D é desenvolvido pelo consórcio Bahia Robotics Team (BRT) formado pelo Núcleo de Arquitetura de Computadores e Sistemas Operacionais (ACSO) e o Grupo de Pesquisa de Computação Inteligente (GPCI) com a finalidade de investigar a aplicação de métodos da inteligência artificial em robôs autônomos, como proposto pela Robocup internacional.

O atual artigo apresenta um estudo, modelagem e aplicação de controladores fuzzy (ZADEH, 1965) em agentes inteligentes jogadores de futebol de robôs simulado em 2 dimensões, além de redes neurais artificiais para auxiliar alguns destes controladores. O objetivo é desenvolver controladores nebulosos para os agentes jogadores de futebol especializados nas diversas posições em campo, de forma a construir uma equipe de agentes inteligentes capaz de disputar as competições da Robocup Federation, constituindo o Bahia2D. Especificamente, serão abordadas as habilidades de chute, passe, marcação, posicionamento com e sem bola a bola, tomada de decisão e controle da carga de bateria disponível (stamina).

A próxima seção apresenta uma breve descrição da modelagem do time Bahia2D. A seção 3 explica a arquitetura adotada pelo time. A motivação de utilizar lógica fuzzy e os modelos fuzzy desenvolvidos pelo grupo para controlar cada um dos jogadores do time estão descritos na seção 4. Já a motivação de utilizar Redes Neurais como técnica auxiliar e os modelos de RNA criados podem ser vistos na seção 5. Por fim, a seção 6 discute os resultados parciais obtidos e os trabalhos futuros.

### 2 Modelagem

A modelagem do time foi feita utilizando-se como base o time UvA Trilearn de 2003 (KOK et al, 2003). Esse time surgiu em 2001 como resultado da dissertação de mestrado de dois estudantes de Amsterdã (BOER e KOK, 2002) e foi o campeão da Robocup 2001 e de outras competições européias. Esse time tem sido usado como base para o desenvolvimento de outras equipes por possuir um mapeamento de todas as percepções enviadas pelo Soccer Server e o seu código fonte estar minuciosamente documentado, fornecendo, dessa forma um framework para o início do desenvolvimento de agentes para a

Robocup. Suas principais características são: a sincronização flexível entre o agente e o ambiente, os métodos exatos para a estimativa de localização e velocidade de objetos e sua hierarquia de habilidades em camadas. Atualmente o UvA Trilearn é utilizado pelas principais equipes do Brasil, podendo citar o Itandroids (MATSUURA et al, 2006), e o Mecateam (COSTA et al, 2006).

### 3. Arquitetura

A arquitetura adotada (COSTA et al, 2006) é dividida em três camadas inter-relacionadas, como pode ser visto na Figura 1.

O Nível Reativo é responsável pela execução das ações. Ele possui os métodos de baixo nível do UvA Trilearn e alguns criados para o Bahia2D. Esses métodos são os responsáveis por transformar as mensagens recebidas do simulador em informações para o nível instintivo e executar as ações resultantes das decisões tomadas.



Figura 1: Divisão em camadas a arquitetura do time.

A segunda camada, o nível instintivo, é responsável pela análise das informações recebidas e pela tomada de decisão com base nessa análise. A terceira camada é o nível cognitivo e ainda está em fase de modelagem/desenvolvimento. Este nível será o responsável pelo planejamento e por gerenciar as metas do time com base nas informações adquiridas durante a partida. Com essa camada o time passará a apresentar inteligência coletiva.

### 4. Controladores Difusos para Robôs Jogadores de Futebol

O Bahia2D utiliza controladores fuzzy e redes neurais para a tomada de decisões, pois as percepções do jogador em uma partida de futebol são imprecisas. O futebol proporciona um ambiente dinâmico, imprevisível e não-determinístico e a força da lógica nebulosa deriva da sua habilidade em inferir conclusões e gerar respostas baseadas em informações vagas, ambíguas, incompletas e imprecisas (FABRI, 2000). Neste aspecto, os sistemas Fuzzy têm habilidade de “raciocinar” de forma semelhante à dos humanos.

Para modelagem das variáveis linguísticas de entrada e saída e das bases de regras fuzzy foi utilizada a ferramenta XFuzzy 3.0 que permite a geração de código da sua máquina de inferência em C. Estes modelos estão descritos nas subseções a seguir.

A formação escolhida para o time foi o 4-3-3 ofensivo, com base nas formações disponíveis na implementação do UvA Trilearn e de acordo com testes empíricos. Dessa forma, o time é composto por um goleiro, dois zagueiros, dois laterais, um meia defensivo, dois meias ofensivos e três atacantes.

#### 4.1 Controlador para Tomada de Decisão de Chute

O Controlador para Tomada de Decisão de Chute tem o objetivo de avaliar a possibilidade de sucesso de um chute à gol. A saída do controlador é a variável Possibilidade de Chute que varia de 0 à 10, possuindo os seguintes termos: Baixa 0 à 3.75, Média 3.75 à 7.5, Alta 7.5 à 10.0.

O agente toma a decisão de acordo com as seguintes percepções: o ângulo em relação ao gol adversário, sua distância em relação ao gol adversário e a quantidade de adversários na direção do gol.

O universo de discurso para ângulo em relação ao gol varia de -180 à 180 graus. Essa definição é utilizada pelo modelo de mundo do UvA Base, sendo que o ângulo 0° é à frente do agente e o ângulo de 180° atrás. Está classificado da seguinte maneira: Ângulo ruim - negativo: -180 à -90, Ângulo ruim - positivo: 90 à 180, Melhor ângulo: -45 à 45, Ângulo bom - positivo: 75 à 90, Ângulo bom - negativo: -90 à -75.

A distância em relação ao gol adversário considerada nesse controlador vai de 0 à 67m e está baseada na distância entre dois pontos (0,-34) à (52.5,7) ou (0,34) à (52.5,-7). Essa é a distância que os atacantes percorrem e devem decidir se vão chutar ou não. Possui como termos linguísticos: Perto: 0.0 à 35.125, Médio: 35.125 à 50.25, Longe: 50.25 à 67.0.

Quantidade de adversários no cone que vai da posição atual do agente ao gol adversário, com raio da base igual à 7.0 m. Seu universo de discurso vai de 0 à 11 e possui os seguintes termos linguísticos: Pouco 0 à 3, Médio 3 à 6, Muito 6 à 11.

Quando este controlador indica uma pequena possibilidade de marcar o gol, ele executa o Controlador para Avaliação de Possibilidade de Passe para todos os companheiros próximos, exceto o goleiro.

#### 4.2 Controlador para Chute

O objetivo do controlador para chute é encontrar o ponto no gol onde a possibilidade de marcar seja a maior possível, sendo a bola chutada da posição atual do agente. Dessa forma, o agente vai escolher o canto do chute de acordo com sua posição em relação ao gol e a posição relativa do goleiro. A variável de saída é a Posição do Chute, que representa a posição no gol em que o agente irá chutar a bola. Seu universo de discurso varia de -7.0 à 7.0, correspon-

dente à posição das traves no eixo Y. O valor do eixo X a ser considerado é o limite do campo, o mesmo da linha do gol, 52.5. A variável possui os termos lingüísticos apresentados na Figura 2, cujo modelo é seguido pelos demais controladores.

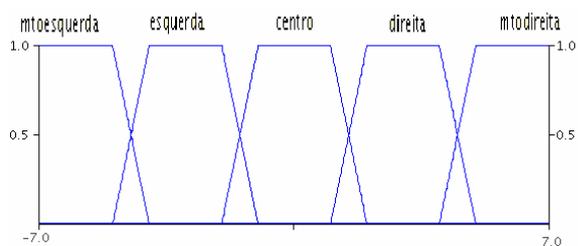


Figura 2: Conjunto fuzzy para as variáveis posição no gol e do goleiro

As variáveis de entrada que influenciam a saída são: Posição do Goleiro, que representa a posição do goleiro no gol e a Posição do Jogador, que representa a posição relativa do agente no campo, com universo de discurso variando de -7.0 à 7.0.

As regras para esse controlador foram criadas a partir da combinação entre as variáveis de entrada, visando a posição no gol mais distante possível do goleiro sem desconsiderar a posição do jogador. O mesmo peso foi atribuído para todas as regras da base. Este controlador foi inicialmente utilizado pelos três agentes atacantes. Posteriormente, o mesmo foi reutilizado também pelos meias ofensivos no momento em que se encontram em situação favorável a chutar para o gol.

#### 4.3 Controlador para Tomada de Decisão de Passe

O Controlador para Avaliação de Possibilidade de Passe tem o objetivo de avaliar a possibilidade de sucesso de um passe para um determinado jogador. A saída dele é a variável Possibilidade de Passe que varia de 0 à 10, possuindo os seguintes termos: Baixa 0 à 3.75, Média 3.75 à 7.5, Alta 7.5 à 10.

O agente toma a decisão de acordo com as seguintes percepções a respeito do jogador candidato ao passe: sua distância à esse jogador, a quantidade de marcadores, a posição desse jogador.

A distância do agente ao jogador candidato ao passe considerada nesse controlador vai de 0 à 30m. Possui como termos lingüísticos: Perto: 0 à 14.25, Médio: 14.25 à 25.5, Longe: 25.5 à 30.

A quantidade de marcadores em um raio de distância de 2m do jogador candidato ao passe. Seu universo de discurso vai de 0 à 11 e possui os seguintes termos lingüísticos: Pouco 0 à 3, Médio 3 à 6, Muito 6 à 11.

A posição do jogador em relação ao agente é calculada pela fórmula:  $posiçãoJogador - posiçãoAgente$ , o resultado é um valor que indica se o jogador candidato ao passe está antes, na mesma linha ou depois do agente com a bola. Seu universo de discurso varia de -30 à 30 e possui os seguintes termos: Antes -30 à 0, Igual 0, Depois 0 à 30.

Se a maior possibilidade retornada pelo controlador for muito pequena, o jogador carrega a bola. Para isso, basta dar um drible com a bola usando 0° de ângulo, pois, independentemente da posição em que o agente estiver no campo, ele irá virar para a frente chutando a bola rente ao corpo, já que o modelo de mundo define o grau 0 à frente e 180 atrás.

A rotina está sendo utilizada dessa forma para os meias e atacantes, os zagueiros e laterais utilizam esse passe, mas quando a possibilidade é pequena demais eles executam um *clear ball* (método para chutar a bola no ângulo com menor quantidade de adversários com exceção do próprio gol).

#### 4.4 Controlador para Tomada de Decisão de Marcação

Este controlador tem o objetivo de proporcionar ao agente a decisão entre marcar um oponente ou se posicionar no campo. Essa decisão é tomada de acordo com as seguintes variáveis: distância da bola ao gol, distância relativa entre a bola e o gol do agente, o número de adversários e companheiros no raio de visão do agente, com quem está a posse de bola, distância até a posição estratégica e a distância do agente até a bola. As regras foram criadas objetivando possibilitar ao agente manter sua posição estratégica, ou seja, ocupar sua zona no campo e também marcar adversários em sua área ou próximo à ela. Quando o agente decide se posicionar ele vai para sua posição estratégica e quando a decisão é marcar ele ativa o controlador de escolha de adversário, que será apresentado a seguir.

#### 4.5 Controlador para Marcação

Este controlador tem o objetivo de escolher o oponente, dentre os próximos ao agente, que tem a maior possibilidade de receber o passe ou avançar em direção ao gol com a bola e não está marcado. Para modelar esta decisão, foi desenvolvido este controlador, que tem como variável de saída a possibilidade que o oponente possui de receber o passe podendo variar de 0 a 10. As variáveis de entrada que influenciam a saída deste controlador são as seguintes: distância do agente ao oponente, uma variável booleana que indica se o oponente já está marcado e a distância da bola ao oponente. A base de regras foi criada com o objetivo de estabelecer uma possibilidade mais alta ao oponente mais próximo do agente, mais próximo da bola e que não está marcado.

#### 4.6 Controlador para Posicionamento dos Atacantes

O objetivo desse controlador é permitir aos atacantes, quando sem a posse da bola, encontrarem uma posição no campo do adversário baseando-se em sua posição atual, na linha de impedimento e na posição da bola. Foi utilizada a abordagem de divisão do campo em zonas (adaptada da proposta de BOER e KOK, 2002), onde a metade do campo do adversário

foi dividida em uma matriz de 9 zonas. A Figura 3 apresenta essa divisão.

A ferramenta *XFuzzy* possibilitou a modelagem de dois controladores para a posição: um para o eixo X e outro para o eixo Y. Esses dois controladores foram integrados a um sistema que possui duas bases de regras independentes. Essa divisão simplificou a criação das regras e não gerou ruído no resultado desejado, pois a variável de saída do eixo X não é influenciada pelas variáveis de entrada no eixo Y e o mesmo ocorre com a saída no eixo Y. Dessa forma, o sistema funciona como um único controlador com cinco entradas e duas saídas.

As variáveis de saída utilizadas pelo controlador são a Posição X e a Posição Y, que representam a posição final para onde o agente deve se deslocar nos eixos X e Y, respectivamente.

As variáveis de entrada são: a posição do jogador e da bola nos eixos X e Y e a posição de impedimento no eixo X.

As regras para esse controlador foram criadas de forma que o agente se deslocasse apenas entre quadrantes adjacentes. Dessa forma, a movimentação ocorre entre distâncias curtas, pois a cada ciclo as percepções podem mudar e, conseqüentemente, a direção a seguir. Por exemplo, um agente posicionado no quadrante 8 só poderá se deslocar até os quadrantes 4, 5, 6, 7, 8 e 9, pois não conseguirá chegar aos quadrantes 1, 2 e 3 no próximo ciclo. Isso foi definido criando-se regras que possuem os mesmos valores para as entradas, mas com saídas e pesos diferentes. As regras para o agente permanecer no quadrante atual, quando a bola está em outro quadrante, possuem peso menor, fazendo com que ele se movimente sempre na direção da bola.

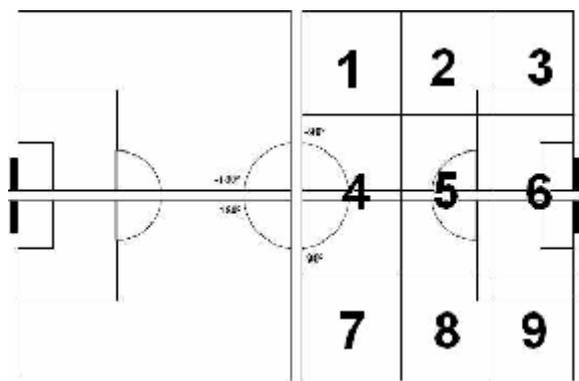


Figura 3: Divisão do campo em zonas, para o time que começa à esquerda. Adaptada de (Reis, 2003).

#### 4.7 Controlador para Posicionamento do Meia Defensivo

O meia defensivo possui um controlador fuzzy cujo objetivo é posicionar estrategicamente o agente na sua área de atuação, levando em consideração a posição da bola e a sua própria posição. Para esse controlador foram definidas duas bases de regras: uma

para o eixo X (na direção de um gol ao outro) e outro para o eixo Y (na direção de uma lateral a outra). A tática para o eixo X é movimentar-se em direção da bola, e a intensidade desta ação depende da posição global da bola e do jogador, ou seja, a depender da posição de cada objeto o jogador tenderá a se aproximar mais ou menos da bola.

Para o eixo Y, a movimentação ocorre de modo similar a do eixo X, só que levando em consideração a coordenada Y. Além disso, para esta base de regras é acrescentada a coordenada do eixo X da bola, que tem um peso expressivo sobre a decisão final de posicionamento. Quanto mais a bola estiver afastada do campo defensivo do time do agente, mais ele tenderá a ficar no centro do campo (em relação ao próprio eixo Y), evitando assim um deslocamento desnecessário, já que a bola está, teoricamente, afastando-se de sua zona de atuação.

#### 4.8 Controlador para Posicionamento do Goleiro com Bola

O controlador para posicionamento do goleiro com a posse de bola possibilita ao goleiro escolher para quem ele deve tocar na reposição de bola.

Para preparar as variáveis de entrada para este controlador, os métodos *escolhaMelhorCirculo*, *escolhaCompanheiroChuteLivre* e *escolhaCompanheiroTiroDeMeta* foram criados. Para a estratégia adotada, o campo à frente do goleiro foi dividido em três círculos iguais, delimitando zonas de reposição da bola. Inicialmente o goleiro escolherá uma das zonas de reposição e, em seguida, escolherá qual o melhor jogador para receber a bola dentro desta zona.

O método da *escolhaMelhorCirculo* recebe como parâmetro informações do modelo de mundo do jogador, trata os dados quantitativos de jogadores em cada círculo e passa para inferência fuzzy a quantidade de companheiros e oponentes em dois dos círculos, retornando como saída o círculo mais viável. Como foram demarcados três círculos, esta função trata primeiramente os círculos da direita e da esquerda, sendo que o escolhido passa novamente pela inferência a fim de ser comparado com o do meio.

Os métodos *escolhaCompanheiroChuteLivre* e *escolhaCompanheiroTiroDeMeta* tratam dados a serem passados para mesma Máquina de Inferência; porém, a depender do evento, tiro de meta ou chute livre, as funções utilizadas do modelo de mundo são distintas. Em ambos os métodos é chamada a inferência *possibCompInferenceEngine*, que recebe como parâmetros: a quantidade de companheiros dentro de um cone, área que vai da posição possível de chute do goleiro até 3m além do companheiro; a distância do goleiro ao companheiro; e a distância do oponente mais rápido ao companheiro. A inferência retornará a variável de saída *possibilidade*, determinando o grau de confiança de cada companheiro dentro da área do círculo previamente esco-

lhido. O jogador que possuir maior grau de confiança receberá a bola do goleiro.

#### 4.9 Controlador para Controle de Stamina

O comando *dash* é usado quando o jogador quer acelerar na direção do seu corpo. Esse comando do Soccer Server recebe apenas um parâmetro: a potência do dash. Essa potência determina a quantidade de aceleração que o jogador, variando de -100 à 100. Quando o jogador acelera na direção contrária o valor passa a ser negativo. O servidor previne que os jogadores permaneçam correndo constantemente em velocidade máxima determinando uma quantidade limitada de stamina para cada um deles. Toda vez que um jogador executa um dash, sua stamina é reduzida de acordo com a potência usada no dash. Se a potência for maior que zero (dash para frente) é reduzida a mesma quantidade, mas se for menor que zero (dash para trás) é reduzida em dobro. O modelo de stamina do Soccer Server está modelado em três partes:

- \* A Stamina: Representa a stamina atual do jogador que possui o valor entre 0 e 4000.

- \* O Esforço: Representa a eficiência do movimento do jogador e varia entre 0.6 a 1.

- \* A Recuperação: Influencia o valor no qual a stamina é recuperada, varia de 0.5 a 1.

A stamina sempre é reduzida quando é executado um dash e recuperada levemente a cada ciclo de acordo com o valor atual da recuperação do jogador. Esse valor de recuperação é reduzido sempre que a stamina chega a um “piso” (atualmente 1200) dificultando a recuperação do jogador. Por esse motivo é crucial controlar o valor de potência enviado ao servidor. Para isso foi criado um controlador Fuzzy que possui como saída a potência que deve ser utilizada pelo jogador no comando de aceleração. As entradas são a stamina atual do jogador, a taxa de esforço, a recuperação e a situação do jogo que é resultante de outro controlador. A situação do jogo possui valores de saída variando de 0 a 10 e com termos que definem a situação como segura, normal ou crítica. Essa definição é resultante das entradas: distância do jogador até a bola, existência de algum adversário próximo à bola, e se a zona em que a bola se encontra é considerada importante (arbitrariamente foram definidas como zonas importantes a defesa e o ataque). A base de regras está definida para que em situações de risco na defesa ou possibilidade de gol no ataque o jogador utilize a potência máxima, mesmo que para isso sacrifique sua recuperação posteriormente. E em situações em que não há disputa de bola o jogador poderá economizar stamina usando uma potência menor para a aceleração.

## 5. Redes Neurais Artificiais para rotina de Marcação

Para a implementação da Rede Neural Artificial (HAYKIN, 2001; Machado, 2005) foi utilizado o Framework NNF 1.1, em C++. A RNA criada foi do tipo MLP, com três camadas, sendo um de entrada, uma intermediária e uma de saída. A rede tem objetivo de decidir entre dois métodos de marcação disponíveis no time base UvaTrilearn, os métodos MARK\_BALL (permanecer a uma distância “d” do oponente na linha entre ele e a bola) e MARK\_BISECTOR (permanecer a uma distância “d” do oponente na bissetriz do ângulo formado pela bola-oponente-gol). As variáveis de entrada para a RNA são: distância da bola ao agente, quantidade de oponentes e companheiros visíveis, distância do agente ao seu próprio gol.

### 5.1 Treinamento

O treinamento consistiu em calibrar os pesos e vieses da RNA. Foram treinadas três RNA's usando o algoritmo backpropagation com taxa de aprendizagem fixada em 0,7. A RNA escolhida possui dois neurônios na camada intermediária e foi treinada usando 20.000 épocas e uma função de ativação degrau que retorna somente -1 caso a saída seja menor que zero e +1 se for maior que zero.

## 6. Resultados Parciais e Trabalhos Futuros

O Bahia2D obteve a 14ª colocação durante as eliminatórias da Robocup 2007 (SSIL) utilizando parte da estrutura de controladores apresentada. Após as eliminatórias, novos controladores foram criados, outros foram melhorados e o uso de Redes Neurais começou a ser adotado. Apesar de ser um time com uma estrutura básica e poucas mudanças em relação ao UvA Trilearn, o Bahia2D venceu todos os times brasileiros que participaram das eliminatórias. Na Robocup 2007, o Bahia2D jogou 10 jogos e perdeu todos. Os melhores resultados foram: Ri-one2007 1 x 0 Bahia2D e YowAI 1 x 0 Bahia2D. Alguns resultados melhoraram com relação às eliminatórias indicando um avanço nas mudanças no time, mas outros não mudaram e alguns até pioraram, mas é importante frisar que a complexidade da simulação faz com que os resultados das partidas sejam influenciados por diversos fatores distintos. Conseqüentemente, derivar conclusões apenas de acordo com resultados das partidas é muito perigoso.

Para avaliar o desempenho dos controladores utilizados no time foi feita a análise dos jogos das eliminatórias e da Copa. Nas eliminatórias foram quatro vitórias em 17 jogos, 28 gols marcados e 161 sofridos.

Na Copa foram 10 jogos, 10 derrotas, nenhum gol marcado, e 172 gols sofridos. A Tabela 1 apresenta

todas as equipes que foram enfrentadas na SSIL, os gols e chutes à gol e o placar no formato Bahia2D:Adversário. Pela análise dos dados pode-se concluir que o time apresentou uma taxa de aproveitamento de chute à gol de 36%. Quando os times adversários possuíam uma rotina de marcação esse valor caiu para 0. Outro problema apresentado nas eliminatórias foi a ausência de rotinas de marcação, que juntamente com a ausência de rotinas para posicionamento e saída de bola do goleiro, prejudicaram bastante os resultados finais contra adversários mais maduros.

**Tabela 1. Resultados obtidos pelo Bahia2D na SSIL 2007.**

Adversário	Gols/Chutes	Placar
HELIOS2007	0/0	0:28
CZU2007	0/0	0:26
NCL07	0/0	0:14
Oxxy	0/4	0:19
Lingdong1214	0/0	0:14
Nemesis	0/0	0:13
FCPortugal07	0/2	0:8
DAINAMITE	0/3	0:10
OPU_hana_2D	0/0	0:7
Incredibles	0/1	0:5
ATH	0/0	0:6
YowAI07	0/1	0:6
Hitro_Jet	0/8	0:3
FURGBOL	1/4	1:0
FEI-Team	3/9	3:2
ITANDROIDS-2D	2/11	2:0
KickOffTUG	22/32	22:0

**Tabela 2. Resultados obtidos pelo Bahia2D na Robocup 2007.**

Adversário	Gols/Chutes	Placar
Oxxy	0/0	0:31
WrightEagle2007	0/0	0:49
ATH	0/2	0:18
ATH	0/0	0:7
Brainstormers07	0/0	0:20
OPU_hana_2D	0/0	0:9
Dainamite	0/0	0:21
Ri-one2007	0/2	0:1
Nemesis	0/0	0:15
YowAI	0/1	0:1

A Tabela 2 segue o padrão da Tabela 1 e apresenta os resultados na Robocup 2007. Ficou evidente nos jogos iniciais que as rotinas de marcação não funcionavam para todos os times e que, além disso, deveriam ser estendidas aos jogadores de meio de campo. Outra falha do time foi na saída de bola do goleiro, que em diversos momentos resultou em gols adversários. A rotina de controle de Stamina criada causou a exaustão dos jogadores que atuavam nas zonas de risco e como não havia um agente técnico para fazer as substituições o time ficou prejudicado diversas vezes. Mudanças no time foram efetuadas a

partir do quarto jogo, estendendo as rotinas de marcação aos meios, e retirando o controle de Stamina, com isso o time passou a se comportar defensivamente e os resultados das partidas melhoraram substancialmente.

Esses resultados expõem as limitações do time e ao mesmo tempo caminhos para trabalhos futuros. Atualmente a utilização de aprendizagem por reforço e computação evolutiva no futebol de robôs estão sendo investigadas. O objetivo é otimizar os comportamentos básicos descritos neste trabalho, criando um Nível Cognitivo para o time.

## Referências

- BOER, Remco de, KOK, Jelle. The Incremental Development of a Synthetic Multi-Agent System: The UvA Trilearn 2001 Robotic Soccer Simulation Team. Amsterdam: Faculty of Science University of Amsterdam, Dissertação de mestrado para Inteligência Artificial e Ciência da Computação. Amsterdã, Holanda, 2002.
- COSTA, Augusto Loureiro da, JÚNIOR, Orivaldo Vieira Santana, SOUZA, João Paulo Rocha de, LINDER, Marcelo Santos. MecaTeam: Um sistema Multiagente para o futebol de robôs simulado baseado no Agente Autônomo Concorrente. ENRI - III Encontro de Robótica Inteligente, Campo Grande, 2006.
- FABRI, José Augusto. Um Sistema Especialista Fuzzy Aplicado a Classificação de Arquiteturas de Computadores. Semana de Informática (SEMINFO) da Universidade Federal da Bahia, Salvador, 09 a 12 de maio de 2000.
- HAYKIN, Simon. *Redes Neurais: Princípios e práticas*, 2ª edição, Editora Bookman, 2001.
- KOK, Jelle, VLASSIS, Nikos, GROEN, Frans. UvA Trilearn 2003 Team Description. Faculty of Science, University of Amsterdam, Amsterdã, Holanda, 2003.
- MACHADO, Fernando Weigert. *Modelagem chuvavazão mensal utilizando redes neurais*. Universidade Federal do Paraná, 2005.
- MATSUURA, Jackson P., XAVIER, Raphael O., BARBOSA, Rodrigo. O Time de Futebol Simulado ITANDROIDS-2D. ENRI – III Encontro de Robótica Inteligente, Campo Grande, 2006.
- REIS, Luís Paulo. Coordenação em Sistemas Multi-Agente: Aplicações na Gestão Universitária e Futebol Robótico. Tese de PhD, FEUP - Faculdade de Engenharia da Universidade do Porto, Porto, Portugal, 2003.
- ZADEH, L. A. Fuzzy Sets – Information and Control. University of California, Berkeley, Califórnia, Estados Unidos, 1965.