Análise do Aprendizado por Reforço Aplicado a Otimização em Tomadas de Decisões Multiagente

André Luiz C. Ottoni
Departamento de Engenharia Elétrica
Universidade Federal de São João del-Rei
São João del-Rei, MG, BRA
andreottoni@ymail.com

Erivelton G. Nepomuceno
Departamento de Engenharia Elétrica
Universidade Federal de São João del-Rei
São João del-Rei, MG, BRA
nepomuceno@ufsj.edu.br

Marcos S. de Oliveira
Departamento de Matemática e Estatística
Universidade Federal de São João del-Rei
São João del-Rei, MG, BRA
mso@ufsj.edu.br

Rubisson D. Lamperti
Departamento de Engenharia Elétrica
Universidade Federal de São João del-Rei
São João del-Rei, MG, BRA
duartelamperti@yahoo.com.br

***Abstract*—O objetivo deste trabalho foi aplicar e analisar os efeitos do aprendizado por reforço na otimização de tomadas decisões de um sistema multiagente cooperativo. É apresentada uma metodologia de modelagem da técnica de aprendizado por reforço para times de futebol de robôs 2D. A implementação da estratégia de aprendizagem consistiu de quatro etapas: definição das ações dos agentes; definição dos estados do ambiente no qual os agentes estão inseridos; definição dos valores dos reforços; implementação no simulador RcSoccerSim da Robocup de futebol de robôs. Os testes estatísticos foram utilizados para verificar o comportamento do time de robôs durante todo o processo de aprendizado. A análise se deu verificando a evolução de desempenho do sistema multiagente como um todo, através de estudos do saldo de gols alcançado em cada jogo. Além disso, a performance individual de cada agente também foi quantificada. Através dos testes de análise de variância e comparações múltiplas foi possível quantificar quais agentes sofreram alterações de performance ao longo do processo de otimização.**

I. Introdução

O Aprendizado por Reforço (AR) é uma técnica de aprendizado de máquina, na qual, o agente aprende por meio de interação direta com o ambiente e seu algoritmo converge para uma situação de equilíbrio [1]. No AR, um agente pode aprender em um ambiente não conhecido previamente, por meio de experimentações. Dependendo de sua atuação, o agente recebe uma recompensa ou uma penalização e, desta forma, o algoritmo encontra um conjunto de ações que levam o agente a percorrer o caminho ótimo. A este conjunto, formado pelas melhores ações, dá-se o nome de política ótima [2].

Em 1950, Turing propôs a abordagem de aprendizado por reforço, escrevendo: "O uso de castigos e recompensas pode ser no máximo uma parte do processo de ensino" [2] apud [3]. Nos animais, nas investigações do comportamento exploratório das abelhas ficou evidente a operação do AR [2]. Já a conexão entre aprendizado por reforço e Processos de Decisão de Markov foi feita por Werbos em 1977 [4]. No entanto, somente no trabalho [5] na Universidade de Massachusetts, teve origem o AR em Inteligência Artificial. Em 1989, Watkins propôs em sua tese de doutorado o Q-learning [6], algoritmo adotado neste trabalho.

Desde a elaboração do Q-learning, pesquisas e publicações vem propondo diferentes aplicações e análises para o AR. Robótica móvel [7], otimização na produção de petróleo [8], tráfego aéreo [9] e controle ótimo de descarregadores de navios [10] são alguns exemplos de aplicações do AR encontrados na literatura. Outras pesquisas atuam na linha de tentar diminuir o tempo gasto para convergência dos algoritmos de AR. Esse é o caso do trabalho de Bianchi [11], que propôs heurísticas para a aceleração do aprendizado. As pesquisas relacionadas ao aprendizado reforço em ambientes multiagente também têm seu destaque [12], [13].

Um ambiente propício para estudos do aprendizado por reforço em sistemas multiagente é a plataforma de futebol de robôs simulado da Robocup[1]. A categoria de simulação 2D da Robocup simula partidas de futebol de robôs autônomos. Nesta liga existem robôs (agentes) virtuais, o qual todo o ambiente é simulado. Um simulador fornece aos agentes todos os dados que seriam obtidos na realidade por meio dos seus sensores e calcula o resultado das ações de cada agente. Cada jogador é visto como um agente individual, e o time como um sistema multiagente totalizando 11 (onze) jogadores por equipe.

Algumas publicações já apresentaram resultados positivos para aplicações do AR na plataforma de simulação da Robocup [14], [15], [16], [17], [18], [19]). No entanto, a literatura ainda carece de uma metodologia de análise sobre os efeitos do AR no futebol 2D.

Baseando-se nisso, o objetivo deste trabalho foi aplicar e analisar os efeitos do AR na otimização de tomadas decisões de um sistema multiagente cooperativo. Como estudo de caso foi adotado o futebol de robôs simulado. Testes estatísticos foram utilizados para verificar o comportamento do time de robôs durante todo o processo de aprendizado [20], [21], [17]. O desempenho do sistema foi verificado através de estudos do saldo de gols alcançado em cada jogo. Além disso, a performance individual de cada agente também foi quantificada. Isso porque, diferentemente do basquete e baseball, onde é comum utilizar a estatística para verificar a contribuição de cada jogador no resultado final, no futebol não é trivial definir a contribuição individual [22]. Dessa forma, foi possível verificar quais robôs mudaram seu desempenho com cada etapa do treinamento do sistema de aprendizado por reforço e a eficiência do processo de otimização.

Este trabalho está organizado em seções. Na seção 2 são

[1]Robocup Federation: http://www.robocup.org.

definidos os Processos de Decisão de Markov. Já na seção 3 é apresentado a modelagem da estratégia de aprendizagem, por meio das definições das ações, dos estados, modelagem de recompensa e implementação no simulador. A análise dos resultados obtidos é apresentada na seção 4. Finalmente, na seção 5 são apresentadas as conclusões.

II. PROCESSOS DE DECISÃO DE MARKOV

Um Processo de Decisão de Markov (MDP - Markov Decision Process) é uma forma de modelar processos, na qual as transições entre estados são probabilísticas.

Uma especificação das probabilidades de resultados para cada ação em cada estado possível é chamada de modelo de transição, denotado por $T(s, a, s')$. $T(s, a, s')$ é utilizado para denotar a probabilidade de alcançar o estado s' se a ação a for executada no estado s [2].

Um MDP é definido pela quádrupla (S, A, T, R) onde [11]:

- S: é um conjunto finito de estados do ambiente;
- A: é um conjunto finito de ações que o agente pode realizar;
- $T : S \times A \rightarrow \Pi(S)$: é a função de transição de estado, em que $\Pi(S)$ é uma função de probabilidades sobre o conjunto de estados S. $T(s_t, a_t, s_{t+1})$ define a probabilidade de realizar a transição do estado s_t para o estado s_{t+1} quando se executa a ação a_t.
- $R : S \times A \rightarrow R$: é a função de recompensa, que especifica a tarefa do agente, definindo a recompensa recebida (ou o custo esperado), ao longo do tempo.

Resolver um MDP consiste em computar a política π: $S \times A$ que maximiza (ou minimiza) alguma função, geralmente a recompensa recebida (ou o custo esperado), ao longo do tempo [11].

A técnica de Aprendizagem por Reforço é fundamentada nos Processos de Decisão de Markov.

III. MODELAGEM DO SISTEMA DE APRENDIZADO POR REFORÇO

A metodologia adotada para o desenvolvimento da estratégia de aprendizagem é dividida em quatro etapas:

1) Definição do conjunto finito de ações que os agentes podem realizar;
2) Definição do conjunto finito de estados do ambiente, no qual, os agentes estão inseridos;
3) Definição dos valores dos reforços, para cada par Estado (S) X Ação (A);
4) Aplicação do algoritmo de aprendizado por reforço Q-learning ao time de futebol de robôs na plataforma de simulação 2D da Robocup.

A. Definição das Ações

Nesta etapa são definidas as possíveis ações de um agente no campo de futebol de robôs simulado 2D. As ações abaixo são apenas para o agente com posse de bola.

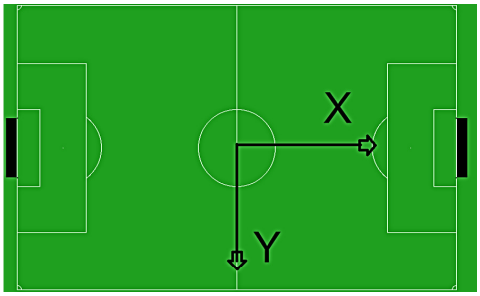

Fig. 1. Sistemas de coordenadas X,Y do campo de futebol simulado 2D para o time que está atacando para a direita.

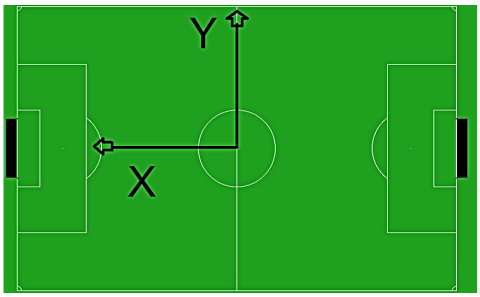

Fig. 2. Sistemas de coordenadas X,Y do campo de futebol simulado 2D para o time que está atacando para a esquerda.

1) Ação: Drible A (Carregar a bola em direção ao gol com drible A);
2) Ação: Drible B (Carregar a bola em direção ao gol com drible B);
3) Ação: Passe A (Tocar a bola para um companheiro com tipo de passe A);
4) Ação: Passe B (Tocar a bola para um companheiro com tipo de passe B);
5) Ação: Lançamento de bola.
6) Ação: Chute (Chutar a bola em direção ao gol).

B. Definição dos Estados

A interação dos agentes com o mundo virtual é interpretado por meio dos estados do ambiente. Nesses estados são definidas as características do ambiente durante uma partida de futebol de robôs. As características levadas em consideração são o posicionamento dos robôs da própria equipe com a posse da bola no plano (X, Y) do campo e a distância dos adversários.

As figuras 1 e 2 apresentam o sistema de eixos (X, Y) na plataforma de futebol de robôs simulado da Robocup. O centro do campo corresponde ao ponto (X=0, Y=0). O eixo X varia de -52,5 a 52,5 de uma extremidade a outra na horizontal e Y de -34 a 34 na vertical.

Para caracterizar o ambiente de atuação dos agentes, o campo de jogo é dividido em cinco zonas. Cada zona, por sua

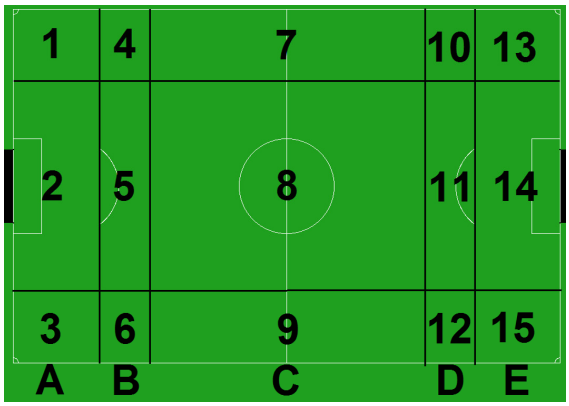

Fig. 3. Esquema de divisão proposto em zonas e células do campo de futebol de robôs simulado. A estrutura é válida para o time atacando da esquerda para a direita.

vez, possui três células, totalizando quinze células no ambiente. As coordenadas X e Y do campo são utilizadas para a definição de cada um desses trechos. Essa estrutura é mostrada na figura 3.

Outra informação levada em consideração para definição do estado do agente com bola é a distância do adversário mais próximo (*dist*). Nesse caso, para *dist* menor que quatro é dito que o adversário está *próximo*. Caso contrário, o adversário está *distante*. É adotado o valor de 4 unidades considerando essa distância igual a soma do diâmetro de dois robôs, visto que, o raio de um agente é próximo de 1 unidade.

C. Definição da Matriz de Recompensas Imediatas

O ambiente do futebol de robôs simulado envolve uma grande complexidade, em termos de número de ações, para que o time de robôs alcance a recompensa principal ao marcar um gol. Um método comum, usado originalmente no treinamento de animais, é chamado de modelagem de recompensa, no qual, fornece recompensas adicionais por ”progressos feitos” [2]. Dessa forma, o objetivo de ”marcar um gol” pode ser desmembrado em ”obter posse de bola”, ”driblar em direção à meta” e ”chutar em direção ao gol”. Reforços intermediários são importantes para acelerar o aprendizado, no entanto, esses reforços devem ter valores inferiores àquele recebido quando o robô atinge o alvo [7].

A partir disso, ao definir as recompensas imediatas, o objetivo é valorizar cada passo necessário para que o time de robôs marque um gol. Ou seja, o objetivo é que o time aprenda uma estratégia de jogo visando um comportamento ofensivo com posse de bola. Essa abordagem é distinta de usar reforços somente quando há gols (recompensas) ou perda de bola (penalizações). Para isso, as recompensas são propostas para aumentarem de valor à medida que o time avance as zonas de divisão do campo, em busca da Zona E e Célula 14. Nesse trecho do campo, o agente estará mais próximo de cumprir a meta de marcar um gol. Dessa forma, para cada Zona é definido um valor de penalidade e um valor de reforço. A penalidade corresponde a um número inferior ao reforço na Zona. Isso porque, o valor de reforço é destinado a execução da ação ”correta” na Zona. No caso da célula 14, a ação ”correta” escolhida é o chute.

A tabela 1 apresenta as penalidades e reforços definidos para cada Zona do campo. Vale notar que o valor do reforço aumenta à medida que o agente com posse de bola está mais próximo do gol adversário (Zona E e Célula14).

TABLE I. VALORES DE REFORÇOS E PENALIDADES PARA CADA ZONA DO CAMPO.

Zona	Penalidade	Reforço
A	-10	-1
B	-1	0
C	0	1
D	1	10
E	10	20
E (Célula14)	10	40

D. Implementação no Simulador

A etapa de implementação da estratégia de aprendizagem por reforço foi realizada no simulador RcSoccerSim de futebol de robôs em duas dimensões da Robocup. O algoritmo algoritmo de AR adotado foi o Q-learning. [23]

1) Algoritmo Q-learning: O método de aprendizagem por reforço Q-learning [23] é um algoritmo que permite estabelecer autonomamente uma política de ações de maneira interativa [19].

A ideia básica do Q-learning é que o algoritmo de aprendizagem aprende um função de avaliação ótima sobre todo o espaço de pares estado-ação $S \times A$. Desde que o particionamento do espaço de estados do robô e do espaço de ações não omita e não introduzam novas informações relevantes. Quando a função ótima Q for aprendida, o agente saberá qual ação resultará na maior recompensa em uma situação particular s futura [24].

A função $Q(s, a)$ de recompensa futura esperada ao se escolher a ação a no estado s, é aprendida por meio de tentativas e erros segundo a equação (1):

$$Q_{t+1} = Q_t(s_t, a_t) + \alpha[r_t + \gamma V_t(s_{t+1}) - Q_t(s_t, a_t)] \quad (1)$$

em que α é a taxa de aprendizagem, r_t é a recompensa, resultante de tomar a ação a no estado s, γ é fator de desconto e o termo $V_t(s_{t+1}) = max_a Q(s_{t+1}, a_t)$ é a utilidade do estado s resultante da ação a, obtida utilizando a função Q que foi aprendida até o presente [24].

```
Para cada s,a inicialize Q(s,a)=0 ;
Observe s ;
while o critério de parada não seja satisfatório do
    Selecione a ação a usando a política de ações ε-gulosa ;
    Execute a ação a ;
    Receba a recompensa imediata r(s,a) ;
    Observe o novo estado s' ;
    Atualize o item Q(s,a) de acordo com a equação (1) ;
    s ← s' ;
end
```

Algoritmo 1: Forma procedimental do algoritmo Q-learning.

A política de escolha ações adotada foi a ε-gulosa(ε-*Greedy*) [2], onde o agente tem probabilidade igual a 1-ε de escolher a ação que maximiza a função valor estado-ação $Q(s, a)$ estando no estado s, e executa uma ação aleatória com probabilidade ε. A forma procedimental do algoritmo Q-learning é retratada no algoritmo 1 [23], [24].

2) Descrição da Implementação: Os agentes foram distribuidos na formação tática da seguinte forma:

- Goleiro: 1.
- Defensores: 2, 3, 4 e 5;
- Meios-de-Campo: 6, 7 e 8;
- Atacantes: 9, 10 e 11.

Os parâmetros do algoritmo Q-learning, a taxa de aprendizagem (α) e o fator de desconto (γ) foram fixados em 0,9 e 0,125 respectivamente. Esse valor para γ é o mesmo utilizado por [11], [14]. Já a definição de α = 0,9 foi baseada nos bons resultados de [15], [14].

Para armazenar as informações aprendidas pelos robôs foi criado um arquivo denominado *q.txt*. Nesse arquivo, a matriz Q de aprendizado foi iniciada com zero para cada par Estado (S) x Ação (A), indicando a inexistência de inteligência no time antes da primeira simulação.

O modelo apresentado visou apenas o aprendizado quando o agente estivesse com a posse de bola. Dessa forma, somente um robô por vez acessa o arquivo *q.txt*, mas o conhecimento de cada um dos agentes fica acumulado na Matriz Q, resultando em uma comunicação entre os jogadores denominada de Quadro-Negro. O Quadro-Negro é uma estrutura comum a todos os agentes, no caso o *q.txt*, onde podem realizar a escrita e a leitura das informações aprendidas por cada robô. Essa estrutura de comunicação visa acelerar o aprendizado dos robôs [11], visto que, as experiências dos agentes se acumulam em uma única estrutura.

Para o processo de aprendizagem (treinamento) o time Aua2D da China [25] foi adotado como adversário. O Time Aua2D participou do campeonato mundial de robótica em 2011 (Robocup 2011).

IV. Resultados

Procurou-se com este trabalho avaliar a evolução do aprendizado do time de robôs. Para isso, foram simuladas 150 partidas de futebol de robôs na plataforma de RcSoccerSim da Robocup. A simulação foi feita adotando o Time Aua2D como adversário para a estratégia modelada. Ao final de cada partida foi armazenado o resultado final do jogo, ou seja, o número de gols feitos por cada time. O saldo de gols (SG) da estratégia modelada no jogo é um fator importante na análise do processo de aprendizado, sendo calculado a partir da diferença de gols feitos (GF) e os gols sofridos (GS), ou como na equação (1),

$$SG = GF - GS. \tag{2}$$

Dessa forma, quanto maior for SG melhor é o rendimento do time. Além disso, o número de toques na bola que os agentes efeturam foram salvos para ponderar a evolução do comportamento individual dos jogadores no decorrer das partidas.

As três análises apresentadas em seguida se referem aos resultados das mesmas 150 simulações. Vale ressaltar que o processo de aprendizado se iniciou na simulação de número 1 e foi finalizado no jogo 150.

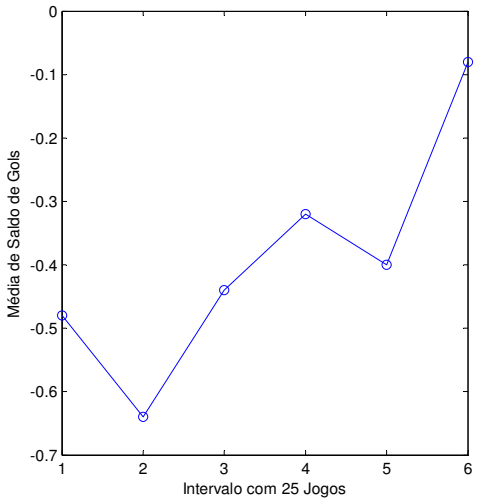

Fig. 4. Média acumulada de saldo de gols para intervalos de 25 simulações.

A metodologia de análise proposta prevê o agrupamento dos jogos em intervalos, a fim de identificar se o time de robôs apresenta alterações nos seus resultados ao longo do processo de otimização.

A. Análise Inicial

A primeira análise dos resultados foi feita dividindo os 150 jogos do processo de aprendizado em seis intervalos com 25 simulações. Para cada um desses intervalos foi calculada a média acumulada de saldo de gols (média entre os 25 jogos no intervalo).

A figura 4 mostra que a média de saldo de gols dos intervalos de 4 à 6 são superiores aos intervalos de 1 à 3. Dessa forma, indicando melhora no rendimento do time de robôs ao longo do processo de treinamento.

B. Análise por meio do Teste T-Pareado

Nessa terceira análise foi utilizado o teste t-pareado [20] para comprovar estatisticamente a evolução do desempenho do time de robôs, durante duas fases de treinamento do algoritmo de aprendizado por reforço:

- Fase Inicial: simulações dos jogos de 1 à 75;
- Fase Final: simulações dos jogos de 76 à 150;

Questionando então, se houve evolução de desempenho do time de robôs na Fase Final em relação a Fase Inicial.

Como foi analisado o comportamento após duas fases da mesma estratégia de aprendizado, verificou-se uma relação de dependência entre as amostras. O teste mais adequado à relação de dependência de duas amostras é o teste t-pareado [26]. O teste t-pareado unilateral foi aplicado utilizando as seguintes hipóteses:

- H_0: a média de saldo de gols na Fase Final é igual a média de saldo de gols na Fase Inicial;
- H_a: a média de saldo de gols na Fase Final é maior que a média de saldo de gols na Fase Inicial;

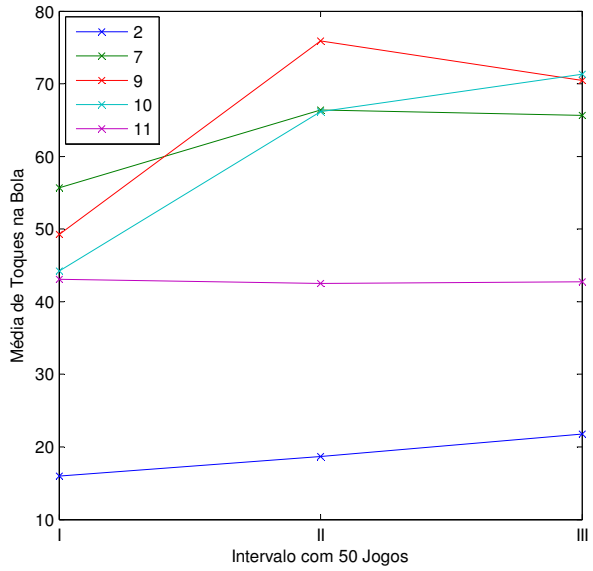

Fig. 5. Média de toques na bola para os robôs 2, 7, 9, 10 e 11.

Foram realizadas todas as etapas de testes no *software* Minitab 14 (versão acadêmica). Inicialmente foi realizado um teste de normalidade para a diferença dos Saldo de Gols entre a Fase Inicial e Fase Final, a fim de verificar a aplicabilidade do teste t-pareado. A suposição de normalidade foi satisfeita a partir do teste de normalidade de Kolmogorov-Smirnov [27], [28].

Se o p-valor (em inglês, *P-Value*) for menor ou igual a 0,05, isto indica que devemos rejeitar H_0 e aceitar H_a, ou seja, o time na Fase Final obteve melhor desempenho que na Fase Inicial. Porém, se o p-valor for maior que 0,05, devemos aceitar H_0, ou seja, o algoritmo de aprendizado não evoluiu da Fase Inicial para a Fase Final. Como o p-valor resultou em 0,018, conclui-se que o comportamento do time de robôs durante a Fase Final de treinamento foi superior estatisticamente em relação ao desempenho na Fase Inicial, com respeito ao saldo de gols.

C. Análise do Comportamento Individual dos Agentes

Nesta etapa foram feitas análises do comportamento individual dos agentes no processo de aprendizado. Desta vez, o treinamento foi dividido em três intervalos para efetuar os testes estatísticos. Sendo que:

- Intervalo I: Jogos de 1 a 50;
- Intervalo II: Jogos de 51 a 100;
- Intervalo III: Jogos de 101 a 150.

O objetivo desta análise foi verificar se havia diferença estatística no comportamento de cada agente de acordo com o Intervalo (I, II, III) do processo de aprendizado. A variável em estudo é o número de Toques na Bola (TB) que cada agente efetuou durante o jogo.

A figura 5 aponta para os agentes 2, 7, 9, 10 e 11 alterações significativas no número médio de toques na bola para os intervalos do processo de aprendizado. Essas alterações de valores indicam uma mudança de comportamento dos agentes de acordo com que o treinamento do algoritmo de AR acontecia. A partir das técnicas de análise de variância (ANOVA) e de comparações múltiplas, foram identificados quais agentes apresentaram diferenças de performace em relação aos intervalos.

O interesse se baseia no teste da igualdade dos efeitos dos tratamentos (intervalos), sendo que t_i é o efeito do i-ésimo tratamento. Dessa forma, o teste de análise de variância foi aplicado utilizando as seguintes hipóteses:

- H_0: $t_1 = t_2 = t_3 = 0$
- H_a: $t_i \neq 0$, para pelo menos um i.

Se o p-valor for menor ou igual a 0,05, isto indica que devemos rejeitar H_0 e aceitar H_a, ou seja, o agente apresentou performance diferente em pelo menos um dos intervalos. Porém, se o p-valor for maior que 0,05, devemos aceitar H_0, ou seja, o agente não apresentou diferença de comportamento entre os intervalos.

Vale ressaltar que a ANOVA possui as seguintes suposições: normalidade, homoscedasticidade e independência. Para os agentes 2 e 8 a suposição de normalidade não foi satisfeito. Já para o agente 9, a suposição de homocedasticidade não foi atendida. Por outro lado, para todos oa agentes verificou-se que não havia violação da suposição de independência. Por fim, para os 3, 4, 5 e 11 não se encontrou significância estatística a partir do ANOVA (ou seja, p-valor acima de 0,05), não sendo necessário a realização do teste de comparações múltiplas para esses casos.

Dessa forma, o teste de comparações múltiplas de Tukey [20] foi aplicado aos dados significantes, agentes 6, 7 e 10, também adotando o mesmo nível de significância de 0,05. Os resultados obtidos a partir do *software* Minitab 14 (versão acadêmica) se encontram na tabela 2.

TABLE II. ANÁLISE DE VARIÂNCIA E COMPARAÇÕES MÚLTIPLAS.

Agente	P-valor	Comparações múltiplas (Intervalos)
6	0,008	I $\neq$ III
7	0,047	I $\neq$ II
10	0,000	I $\neq$ II e I $\neq$ III

V. CONCLUSÃO

Este trabalho teve como objetivo principal modelar e estudar uma estratégia de aprendizado por reforço (AR) no domínio do futebol de robôs em duas dimensões (2D). Para isso, foi adotado o algoritmo Q-learning. Para a aplicação do AR foi adotada uma metodologia dividida em etapas: definição das ações, definição dos estados, modelagem das recompensas e implementação no simulador.

Em seguida, foi feita a análise dos resultados obtidos durante o processo de aprendizado, realizado em 150 simulações. O principal objetivo da análise foi comprovar a evolução da performance do time de robôs.

A análise inicial mostrou uma tendência ao aumento da média acumulada de saldo de gols ao longo do processo de aprendizado, indicando uma melhora no rendimento do time de robôs. Em seguida, na segunda etapa de análise de resultados

adotou o teste estatístico t-pareado. Dentro das condições experimentais utilizadas, o teste sugere efeitos significativos a um nível de confiança de 95%. Já última etapa teve um objetivo diferente, verificar a evolução do desempenho individual dos agentes. Para isso, foi adotada como variável de estudo o número de toques na bola que cada robô efetuou durante um jogo. Através dos testes estatísticos de análise de variância e comparações múltiplas foi possível verificar quais agentes sofreram alterações de performance. Essas mudanças de desempenho indicam que o AR alterou a política de tomadas de decisões multiagente ao longo do processo de otimização. Ou seja, o sistema atuou na busca pela política ótima com as melhores ações para a situação de treinamento.

Vale ressaltar que este trabalho é uma sequência dos estudos realizados por estes autores envolvendo aprendizado por reforço, futebol de robôs e análise estatística. Em 2011, os autores iniciaram as pesquisas sobre a análise dos comportamentos de sistemas multiagentes por meio de testes estatísticos [21]. Já em 2012, nos artigos [16] e [17] foi retratada a metodologia para modelagem e simulação do aprendizado por reforço no futebol de robôs. Além disso, os autores adotaram o teste t-pareado para a análise dos resultados de saldo de gols em [16]. Neste trabalho, com objetivo de melhor representar o ambiente de jogo, foi adotado um novo modelo de AR com mais estados (S). A fim de acelerar o aprendizado, as recompensas foram propostas de forma a valorizar os "progressos feitos" pelos agentes em direção ao gol (objetivo). Outro ponto acrescido neste artigo é a análise do comportamento individual dos agentes ao longo processo de aprendizado, através das técnicas de análise de variância e comparações múltipla.

Nos próximos trabalhos, serão implementados outros algoritmos de AR com o objetivo de comparar o desempenho geral do time e dos agentes em cada implementação. Além disso, serão propostos novos modelos para o problema, adotando outros valores para a Matriz de Recompensas e mais características para definição dos estados.

AGRADECIMENTOS

Agradecemos à CAPES, CNPQ, FAPEMIG e UFSJ pelo apoio.

REFERÊNCIAS

[1] R. Sutton and A. Barto. *Reinforcement Learning: an introduction*. Cambridge, MA: MIT Press, first edition, 1998.

[2] S. J. Russell and P. Norving. *Inteligência Artificial*. Campus, second edition, 2004.

[3] A. Turing. "Computing machinery and intelligence." *Mind*, vol. 59, pp. 433–460.

[4] P. Werbos. "Advanced forecasting methods for global crisis warning and models of intelligence". *General Systems Yearbook*, vol. 22, pp. 25–38, 1977.

[5] A. G. Barto, R. S. Suton and P. S. Brouwer. "Associative search network: A reinforcement learning associative memory". *Biological Cybernetics*, vol. 40(3), pp. 201–211, 1981.

[6] C. J. Watkins. "Models of Delayed Reinforcement Learning". Master's thesis, PhD thesis, Psychology Department, Cambridge University, Cambridge, United Kingdom., 1989.

[7] A. H. P. Selvatici and A. H. R. Costa. "Aprendizado da coordenação de comportamentos primitivos para robôs móveis". *Revista Controle & Automação*, vol. 18, pp. 173 – 186, 06 2007.

[8] G. A. Oliveira. "Uma aplicação da aprendizagem por reforço na otimização da produção em um campo de petróleo". Master's thesis, Universidade Federal do Rio Grande do Norte, 2010.

[9] D. P. Alves. "Modelagem de Aprendizagem por Reforço e Controle em Nível Meta para melhorar a Performance da Comunicação em Gerência de Tráfego Aéreo". Master's thesis, Universidade de Brasília, 2006.

[10] L. A. Scárdua, J. J. Cruz and A. H. R. Costa. "Controle Ótimo de Descarregadores de Navios Utilizando Aprendizado por Reforço". *Revista Controle & Automação*, vol. Vol.14 no.4, 2003.

[11] R. A. C. Bianchi. "Uso de Heurística para a aceleração do aprendizado por reforço." Master's thesis, Tese (Doutorado) Escola Politécnica da Universidade de São Paulo., 2004.

[12] J. S. Waskow. "Aprendizado por Reforço utilizando Tile Coding em Cenários Multiagente". Master's thesis, Universidade Federal do Rio Grande do Sul, 2010.

[13] L. A. Celiberto Jr and R. A. C. Bianchi. "Aprendizado por Reforço Acelerado por Heurística para um Sistema Multi-Agentes". *3rd Workshop on MSc dissertations and PhD thesis in Artificial Intelligence*, 2006.

[14] L. A. Celiberto Jr. "Aprendizado por Reforço Acelerado por Heurísticas no Domínio do Futebol de Robôs Simulado". Master's thesis, Centro Universitário da FEI, 2007.

[15] P. Stone, R. S. Sutton and G. Kuhlmann. "Reinforcement Learning for RoboCup-Soccer Keepaway". *Adaptive Behavior*, vol. 13, no. 3, pp. 165–188, 2005.

[16] A. L. C. Ottoni, R. D. Lamperti, E. G. Nepomuceno, M. S. Oliveira and F. F. Oliveira. "Modelagem e Simulação de um Sistema de Aprendizado de Reforço para Robôs". *VIII Encontro Mineiro de Engenharia de Produção, ISSN 1983 - 0629*, 2012.

[17] A. L. C. Ottoni, R. D. Lamperti, E. G. Nepomuceno and M. S. Oliveira. "Desenvolvimento de um sistema de aprendizado por reforço para times de robôs - Uma análise de desempenho por meio de testes estatísticos". *XIX Congresso Brasileiro de Automática, ISBN 978-85-8001-069-5*, pp. 3557–3564, 2012.

[18] S. E. H. Kerbage, E. O. Antunes, D. F. Almeida and P. F. F. Rosa. "Generalização da aprendizagem por reforço: Uma estratégia para robôs autônomos cooperativos." *Competição Latino Americana de Robótica*, 2010.

[19] J. R. F. Neri, C. H. F. Santos and J. A. Fabro. "Descrição Do Time GPR-2D 2011". *Competição Brasileira de Robótica*, vol. 2011, 2011.

[20] W. W. Hines, D. C. Montgomery, D. M. Goldsman and C. M. Borror. *Probabilidade e Estatística na Engenharia*. LTC, 2006.

[21] A. L. C. Ottoni, E. G. Nepomuceno, F. F. Oliveira and M. S. Oliveira. "Análise do comportamento de sistemas multiagentes cooperativos por meio de testes estatísticos". *X Encontro Mineiro de Estatística*, 2011.

[22] J. Duch, J. S. Waitzman and L. A. N. Amaral. "Quantifying the Perfomance of Individual Player in a Team Activity". *PloS ONE*, vol. 5(6): e10937, doi:10.1371/journal.pone.0010937, 2010.

[23] C. J. Watkins and P. Dayan. "Technical note Q-learning". *Machine Learning*, 1992.

[24] S. T. Monteiro and C. H. C. Ribeiro. "Desempenho de Algoritmos de Aprendizagem por Reforço sob Condições de Ambiguidade Sensorial em Robótica Móvel". *Revista Controle & Automação*, vol. Vol.15 no.3, 2004.

[25] L. Tao and R. Zhang. "AUA2D Soccer Simulation Team Description Paper for RoboCup 2011". *Robocup 2011*, 2011.

[26] W. B. Bussab and P. Morettin. *Estatística Básica*. Saraiva, 7th edition, 2012.

[27] A. Kolmogorov. "Sulla determinazione empirica di una legge di distribuzione". *G. Inst. Ital. Attuari*, vol. 4, pp. 83, 1933.

[28] N. Smirnov. "Tables for estimating the goodness of fit of empirical distributions". *Annals of Mathematical Statistics*, vol. 19, pp. 279, 1948.