

RoboCIn 2D Description Paper

Cristiano Santos de Oliveira¹, Lucas Vinicius da Costa Santana²,
Mateus Gonçalves Machado³, Walber Macedo Rodrigues⁴,
Hansenclever de Franca Bassani⁵, Edna Natividade da Silva Barros⁶

Abstract—Este *Team Description Paper* (TDP) tem por objetivo introduzir a equipe RobôCIn na categoria RoboCup Soccer Simulation 2D da Latin American and Brazilian Robotics Competition (LARC/CBR) 2018, competição que ocorrerá de 6 a 10 de novembro em João Pessoa/PB. Serão descritos os primeiros avanços no desenvolvimento de um time competitivo usando estratégias de Inteligência Artificial.

I. INTRODUÇÃO

O futebol de campo tem como essência a tomada de decisões individuais e em grupo, tal característica é herdada para a competição de *Soccer Simulation 2D*, em que agentes autônomos tomam decisões individualmente e são regidos por uma entidade externa ao jogo que conduz o grupo para agir de forma mais coesa.

O RoboCIn surgiu com a proposta de entrar na categoria no final de 2017 e iniciou as atividades em fevereiro de 2018. A ideia de formar uma estratégia de jogo organizada para 11 jogadores é escalável para a equipe que já participa há 2 anos da categoria Very Small Size Soccer (VSSS), em que a estratégia consiste em controlar 3 jogadores.

A equipe de Simulação 2D tem como objetivo utilizar as técnicas mais atuais para um sistema multi agente(SMA), desde árvores de decisão à técnicas de *Deep Learning*, no ambiente do futebol.

Para entrar na categoria, primeiro tivemos que nos ambientar, observar o estado da arte em relação à categoria, localizar interfaces de comunicação do servidor oferecido pela RoboCup e estruturar o código de forma fácil para adequar os algoritmos para goleiro, defesa e ataque.

Este artigo apresenta a descrição do *framework* utilizado por nós, a estruturação do nosso código, a descrição de

papéis de cada agente em campo e os avanços pretendidos até a competição.

II. METODOLOGIA

O desenvolvimento deste projeto foi baseado na utilização do *framework soccerpy* [1]. Escolhemos esse *framework* pela simplicidade de implementação e liberdade de poder criar uma estrutura do início com um bom desempenho. Apesar de poucas citações desse *framework* por outras equipes, ele é de código aberto e não demonstra baixa em velocidade comparado a interface do *Agent 2D* [2], o mais utilizado por outras equipes nos últimos anos.

O *soccerpy* é um *framework* que fornece funções básicas de comunicação com o servidor e ações para os agentes. Essas funcionalidades são suficientes para construção de um time capaz de jogar competitivamente. Por ser implementado completamente em Python, o *soccerpy* permite o desenvolvimento de uma aplicação com alta complexidade. Também existe a possibilidade de utilização de linguagens de mais baixo nível, como C e C++, caso seja necessário um controle maior da execução do código.

Partindo dessa estrutura base, implementamos para cada jogador um comportamento diferente de acordo com seu papel no jogo. Além desses serem definidos por papel, também se diferenciam por situação de jogo.

III. DESENVOLVIMENTO

A. Arquitetura

Mesmo utilizando o *framework soccerpy*, tivemos que construir uma arquitetura bem definida para dar maior suporte ao desenvolvimento do projeto. A estrutura do código segue um fluxo de funcionamento como mostrado na figura 1.

O pilar básico para o funcionamento dessa estrutura são os agentes. Cada agente representa um jogador e uma thread autônoma que comunica diretamente com o simulador. Suas funções essenciais são definidas a partir das classes mostradas no diagrama, como a classe World, que se mostra uma das mais importantes, visto que todas as características do ambiente da simulação são definidas nela. Além disso, cada agente é definido individualmente

¹Cristiano Oliveira Author is with Centro de Informática, Universidade Federal de Pernambuco, Brazil csoc@cin.ufpe.br

²Lucas Santana Author is with Centro de Informática, Universidade Federal de Pernambuco, Brazil lvcs@cin.ufpe.br

³Mateus Machado Author is with Centro de Informática, Universidade Federal de Pernambuco, Brazil mgm4@cin.ufpe.br

⁴Walber Macedo Author is with Centro de Informática, Universidade Federal de Pernambuco, Brazil wmr@cin.ufpe.br

⁵Hansenclever Bassani Author is with Centro de Informática, Universidade Federal de Pernambuco, Brazil hfb@cin.ufpe.br

⁶Edna Barros Author is with Centro de Informática, Universidade Federal de Pernambuco, Brazil ensb@cin.ufpe.br

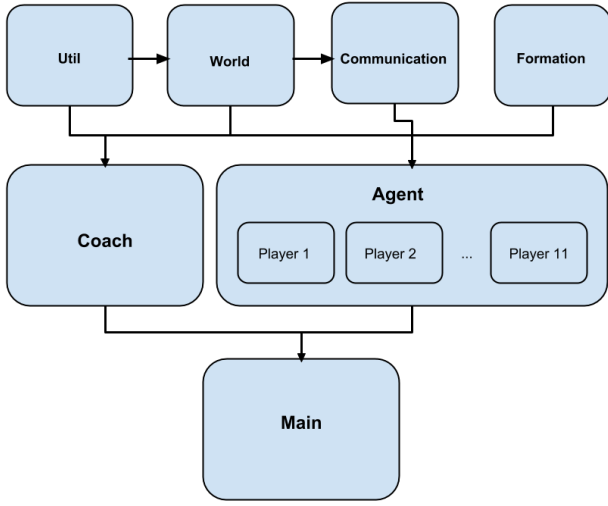


Fig. 1. Diagrama da arquitetura.

baseado no seu papel no jogo, goleiro, atacante etc.

Em paralelo, temos o *Coach*, que realiza suas funções de análise do jogo e substituições, seguindo a função de técnico, limitada pelas regras do simulador.

B. Posicionamento

O posicionamento é parte fundamental na estratégia durante um jogo de futebol e na simulação. Com a posse de bola, o objetivo é que o agente que está com a bola tenha condições de realizar passes durante a maior parte do tempo. Sem a posse de bola, os agentes devem ocupar a maior parte do campo, isso aumenta o tempo de reação do agente caso a bola ou um agente adversário esteja perto.

Visto esse problema, como método para organizar o time em campo foi escolhido o algoritmo *bird swarm*, descrito por Craig W. Reynolds [3] como *boids*. Atualmente existem outros algoritmos que replicam comportamentos de enxame [4] [5].

O algoritmo de enxame [3] é desenvolvido utilizando três movimentos principais, separação, coesão e alinhamento. Tais movimentos se encaixam na estratégia de marcação por zona: os agentes evitam estar muito próximos uns dos outros, respondem em conjunto à movimentação de outros agentes e tendem a se manter sempre ao alcance uns dos outros facilitando a criação de jogadas. Esses movimentos são representados na figura 2.

Cada agente a_i possui um campo de visão V_i , posição p_i e velocidade v_i , representação na equação 1. Como cada agente não tem acesso a velocidade de outro, o cálculo da velocidade entre agentes é dado pela definição de derivada da posição no tempo. A ultima posição vista,

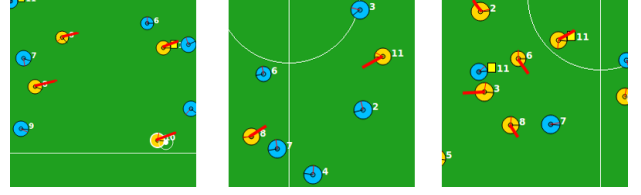


Fig. 2. Esquerda: Movimento de alinhamento. Centro: Movimento de coesão. Direita: Movimento de separação.

$p_{t-\Delta t}$ é subtraída da posição atual p_t e dividida por Δt que representa o intervalo de tempo entre cada vez que um jogador foi visto. A cada 2 segundos a ultima posição vista é esquecida para evitar informações desatualizadas da situação.

$$\vec{v} = \frac{p_t - p_{t-\Delta t}}{\Delta t} \quad (1)$$

O movimento de separação, \vec{s} , é definido pela equação 2 e é essencial para evitar colisões entre os agentes. Cada agente com seu campo de visão consegue identificar os outros no entorno e estimar o movimento para evitar uma colisão. O vetor de movimento de separação é definido pela direção inversa da soma dos vetores entre as posições do agente atual i e os outros agentes de seu entorno.

$$\vec{s}_i = - \sum_{\forall a_j \in V_i} p_i - p_j \quad (2)$$

O movimento de coesão \vec{k} , equação 3, é responsável por manter agentes vizinhos sempre em alcance mesmo com o movimento de separação. Ele é o vetor entre a posição de um agente e o centro de densidade de os outros agentes que estão ao alcance do agente.

$$\vec{k}_i = \sum_{\forall a_j \in V_i} \frac{p_j}{m} - p_i \quad (3)$$

O movimento de alinhamento \vec{m} , definido pela equação 4, assegura que todos os agentes acompanhem a tendência de movimento dos outros agentes ao seu alcance. Esse movimento é calculado a partir da velocidade média dos outros agentes ao seu alcance. Assim, garante que o time consiga responder a mudanças no jogo, a informação de movimento dos agentes irá se propagar até agentes mais afastados.

$$\vec{m}_i = \sum_{\forall a_j \in V_i} \frac{v_j}{m} \quad (4)$$

C. Goleiro

O goleiro é um agente que merece atenção especial e, para uma primeira forma de abordá-lo, escolhemos implementar um algoritmo determinístico, baseado na otimização da sua área de atuação.

A princípio, estamos usando de uma área de atuação determinada por uma reta. O agente fica em paralelo com a bola, isso sendo possível quando a bola está no seu campo de visão. Essa é uma abordagem simples e desconsidera a precisão de onde a bola pode estar em relação ao ângulo de chute de quem contém a bola.

Após analisarmos alguns outros comportamentos possíveis para o goleiro, chegamos a uma evolução desse tratamento em linha reta, fundamentado a partir do livro [6]. Esta abordagem considera as posições de atuação do goleiro com base em um semi-círculo traçado a partir dos limites laterais do gol, como mostrar na figura 3.

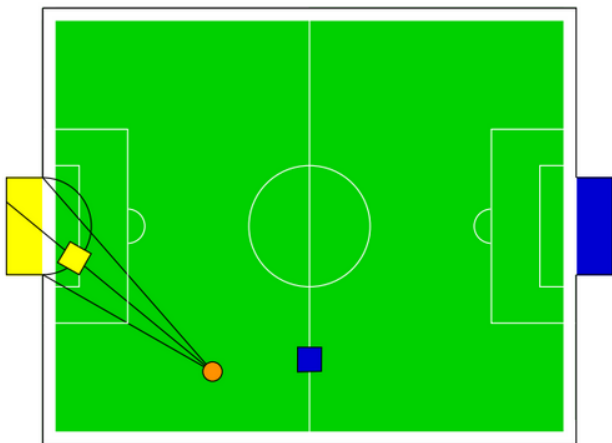


Fig. 3. Posicionamento de atuação do goleiro.

Além de considerar essa outra forma de delimitar as posições de atuação, esse agente também tem noção de onde a bola poderia estar caso fosse chutada para o gol. Isso é possível supondo a partir das retas traçadas da bola, no campo de visão do goleiro, para o goleiro e para os limites laterais do gol.

D. Defesa

Como método de defesa, é escolhido a marcação por zona que se configura quando o time se organiza de tal maneira em que o agente se preocupa em otimizar a o espaço total ocupado pela sua equipe. Como a situação do jogo sempre se modifica, vários parâmetros são levados em conta quando se configura o tipo de marcação por zona, posição dos outros agentes do seu time, a posição da bola e a distância até os gols.

Esse tipo de marcação utiliza zonas de função dinâmica, que modificam durante o jogo o comportamento de cada jogador dependendo da localização da bola em campo. Essas zonas foram definidas por [7], assim, quando um jogador estiver dentro de uma zona específica, seu comportamento será modificado. Por exemplo, quando um adversário estiver na zona de *BA_CrossBlock*, o agente deverá impedir que a bola chegue ao meio de campo na zona *BA_Danger*, ou se a bola chegar nessa zona, o agente deverá reduzir o ângulo em que o atacante adversário deverá bater a bola. A figura 4 indica todas as zonas do campo.

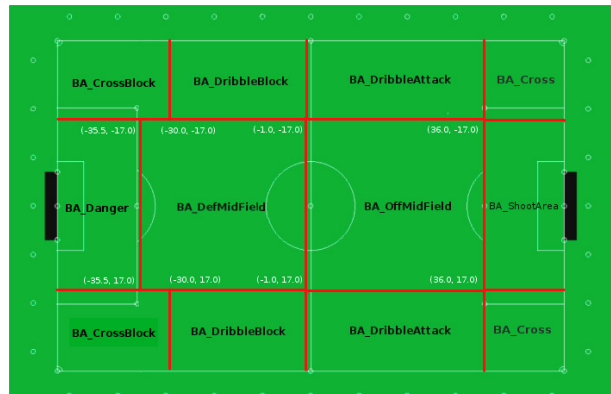


Fig. 4. Zonas de função dinâmica [7].

E. Ataque

De forma parecida com a defesa, o ataque funciona com forças atrativas para jogadores aliados e repelentes para jogadores adversários. O jogador deve passar a bola para o jogador mais perto com menor força repelente, ou seja, com menos marcação. Caso o jogador com a bola seja o mais vantajoso, em relação às forças, ele continua com a bola. Dessa forma, o ataque pode ser feito de uma forma fluida com os jogadores tomando as rotas mais livres para receber a bola.

O passe longo é de muita importância nessa estratégia, uma vez que o jogador mais livre e melhor posicionado possa estar mais distante do jogador atual, porém não é utilizado muitas vezes durante o jogo por conta da *stamina*, ou energia, do jogador que receberá a bola.

F. Técnico

O técnico é o agente que controla algumas ações de campo, por exemplo, substituições e formação do time. Atualmente nosso técnico funciona de forma determinística, a qual substitui os jogadores de acordo com a *stamina* do jogador, e determina a formação do time de acordo com a situação do jogo.

A mudança de formação se adapta de acordo com a situação do jogo. Caso estivermos perdendo, partiremos

para uma formação mais ofensiva, tentando uma melhor aproximação ao gol adversário. Caso estivermos ganhando por mais de um gol de diferença, partimos para uma estratégia mais defensiva. A troca de passes também é importante para o desenvolvimento do jogo. Caso o time não estiver obtendo um acerto de passes acima de 85%, adotamos uma formação com mais jogadores no meio de campo.

G. Abordagens futuras

Visto que estamos participando pela primeira vez dessa categoria, há muito a ser explorado e desenvolvido ainda. Com a experiência obtida nessa primeira participação, já temos um direcionamento maior para investigar novas estratégias e estudar novas abordagens dentro do que está sendo desenvolvido, com intuito de melhorar o desempenho da equipe.

Uma abordagem bastante cogitado por nossa equipe é o uso de *Deep Learning* (DL) para determinar o comportamento dos jogadores, dado que temos integrantes com experiência na área e já estão fazendo algumas experimentações. Outra abordagem é o uso de *Reinforcement Learning* (RL), que algumas equipes brasileiras usaram nos últimos dois anos, através do *Q-Learning* [8]. O *Q-Learning* guarda uma matriz de melhores ações tomadas durante a partida e tenta maximizar o valor da recompensa da próxima ação do agente segundo a equação 5.

$$Q(\text{estado}, \text{ação}) = R(\text{estado}, \text{ação}) + \gamma * \text{Max}[Q(\text{próximo estado}, \text{todas ações})] \quad (5)$$

Onde:

$R(\text{estado}, \text{ação})$: Recompensa da ação x no estado y
 γ : Taxa de aprendizagem

Pretendemos treinar o goleiro com o *Q-Learning*, porém utilizando a matriz Q com alguns valores condizentes com nosso goleiro atual. O tempo de convergência do algoritmo tende a ser menor dessa forma, pois o agente conseguirá uma maior recompensa.

Também usando *Q-Learning*, pretendemos aprimorar nosso técnico para escolha de formações. Para isso, iremos treinar com os times campeões mundiais e colocando algumas formações bases, queremos aprimorar a matriz Q do Treinador para que haja excelência na escolha de formações.

Outro estudo pretendido pela equipe é implementar o algoritmo de defesa descrito por Gabel et al. [6]. Eles aplicaram a técnica de *NeuroHassle* [9], uma técnica de RL que descreve políticas de ações. Os times que

adotaram essa abordagem conseguiram uma aproximação mais agressiva da defesa contra o time adversário. Caso tivermos sucesso, tentaremos fazer um algoritmo misto com o *NeuroHassle*.

IV. CONCLUSÃO

No presente *Team Description Paper* (TDP), é mostrado como foi desenvolvido um sistema completo para a participação na competição *Soccer Simulation 2D*. Foi descrito um sistema de estratégia que define cada papel por jogador e delimita seus comportamentos utilizando uma lógica de enxame com o objetivo de que o time vença a partida.

AGRADECIMENTOS

A equipe gostaria de agradecer o Centro de Informática da UFPE pelo apoio financeiro e de recursos durante todo o processo do projeto. Também gostaríamos de agradecer à todo apoio dado pelos professores Edna Barros e Hansen-clever Bassani ao longo da criação do projeto.

REFERENCES

- [1] "Soccerpy," <https://github.com/jasontbradshaw/soccerpy>, accessed: 2018-07-10.
- [2] "Robocup tools agent2d," <https://pt.osdn.net/projects/rctools/>, accessed: 2018-07-10.
- [3] C. W. Reynolds, "Flocks, herds and schools: A distributed behavioral model," in *ACM SIGGRAPH computer graphics*, vol. 21, no. 4. ACM, 1987, pp. 25–34.
- [4] X.-B. Meng, X. Z. Gao, L. Lu, Y. Liu, and H. Zhang, "A new bio-inspired optimisation algorithm: Bird swarm algorithm," *Journal of Experimental & Theoretical Artificial Intelligence*, vol. 28, no. 4, pp. 673–687, 2016.
- [5] M. Awadel-Bayoumy, M. Rashad, M. Elsoud, and M. El-dosuky, "Fafsa: Fast artificial fish swarm algorithm," *International Journal of Information Science and Intelligent System*, vol. 2, no. 4, pp. 60–70, 2013.
- [6] R. A. Seesink, in *Artificial Intelligence in a multi agent robot soccer domain*, M. Poel, A. Schoute, and J. Zwiers, Eds. Netherlands, Faculty of Computer Science University of Twente: Language, Knowledge and Interaction Group, Faculty of Computer Science University of Twente, The Netherlands, pp. 13, 42.
- [7] "Robocup 2d simulation strategy," github.com/robocup2d/manual/blob/master/sections/Strategy.md, accessed: 2018-07-10.
- [8] C. J. C. H. Watkins and P. Dayan, "Q-learning," *Machine Learning*, vol. 8, no. 3, pp. 279–292, May 1992. [Online]. Available: <https://doi.org/10.1007/BF00992698>
- [9] T. Gabel, M. Riedmiller, and F. Trost, "A case study on improving defense behavior in soccer simulation 2d: The neurohassle approach," in *RoboCup 2008: Robot Soccer World Cup XII*, year="2009, L. Iocchi, H. Matsubara, A. Weitzenfeld, and C. Zhou, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, pp. 61–72.