

MODELAGEM MATEMÁTICA DE UM JOGO DEFENSIVO PARA ROBOCUP F-180 SMALL-SIZE LEAGUE

Monael Pinheiro Ribeiro

Instituto Militar de Engenharia – I.M.E.

Praça General Tibúrcio, n°.80, Praia Vermelha, Rio de Janeiro, R.J., Brasil

mpribeiro@de9.ime.eb.br

Júlio Cesar Silva Neves

Instituto Militar de Engenharia – I.M.E.

Praça General Tibúrcio, n°.80, Praia Vermelha, Rio de Janeiro, R.J., Brasil

sneves@ime.eb.br

Resumo

As equipes de robôs autônomos estão, a cada dia, mais envolvidas com o comportamento social, onde a colaboração em buscar uma meta comum para a equipe é em prol do que cada robô individualmente trabalha. Neste contexto, a competição de Futebol de Robôs da Robocup torna-se um campo vasto para os mais diversos experimentos; vislumbra-se integrar os robôs de cada equipe, formando um verdadeiro time. No presente artigo é proposto, implementado e testado um modelo de programação matemática, com a finalidade de definir a posição dos membros de uma equipe em um jogo defensivo, onde a proximidade da bola e a cobertura de cada adversário são considerados dentro do modelo matemático.

Palavras-Chaves: Modelagem de sistemas, Programação Matemática, Futebol de Robôs.

Abstract

Nowadays, groups of autonomous robots are more and more involved with social behavior, attempting to find a common goal for the group in a collaborative way in order to accomplish the individual task of each robot. In this context the Robocup, a robot soccer competition, becomes a broad field for many different experiments, looking for a integration of the robots from each group becoming a true team. This paper proposes an implemented and tested mathematical programming model, which has the goal of defining the team member position in a game, playing in a defensive way; where the ball position and the action coverage of each opponent are considered in the mathematic model.

Keywords: System modeling, Mathematical Programming, Robocup F-180.

1. INTRODUÇÃO

Para alcançar um bom desempenho, as equipes de robôs autônomos, necessitam superar alguns desafios; devem coordenar as atividades de cada robô de forma a manter-se bem entendido seus comportamentos para atingir os objetivos, mas devem também, responder em tempo real às situações inesperadas. Além disso, a equipe de robôs necessita se adaptar às ações do oponentes. Individualmente, os robôs devem executar seqüências de ações complexas que conduzam ao cumprimento de metas a longo prazo, além de responder, em tempo real, à situações inesperadas. Em outro plano, cada robô deve possuir um repertório de comportamentos e poder executá-los mediante a presença dos adversários, a fim de

construir uma estratégia viável para a equipe, embora, isso mude muito de um aspecto de vários robôs [4] para um de um único robô [6], ademais, a presença dos adversários aumenta significativamente a complexidade do problema. No ambiente apresentado pela competição de futebol de robôs, onde as fragilidades da equipe são exploradas ativamente, o emprego de táticas, habilidades e jogos são importantes para tirar vantagens de tais fragilidades. Desta forma, o controle e a reutilização de políticas através de subproblemas são feitos de modo eficiente, aplicando todo o tempo e recursos disponíveis.

Neste artigo, será proposta a modelagem de um jogo defensivo para controlar um time de robôs; e encontra-se estruturado da seguinte maneira:

Na próxima seção apresenta-se, sucintamente, o problema do futebol de robôs, assim como, a própria competição da Robocup e a liga específica que almeja-se atacar neste trabalho; na seção 3, a arquitetura de controle usada em [1] é exposta e detalhada para que o leitor compreenda o fluxo de informações e o controle de um sistema que coordene os robôs da equipe; na quarta seção, desenvolve-se a proposta deste artigo, apresentando a formulação de programação matemática; a seção 5 descreve brevemente a implementação deste modelo em uma ferramenta computacional e como este pode-se acoplar na arquitetura de controle existente; na seção 6 submete-se o modelo implementado a alguns testes de validação em situações típicas de uma partidas reais; finalmente na seção 7, as conclusões e sugestões para trabalhos futuros são apresentadas.

2. FUTEBOL DE ROBÔS

A proposta de um jogo defensivo é aplicável a uma equipe autônoma de robôs que executam uma tarefa no domínio da competição de futebol de robôs da RoboCup. Mais especificamente, na Liga Small-Size (SSL), uma divisão dentro da RoboCup. Nesta seção, será definido o problema do futebol de robô da SSL.

2.1. LIGA SMALL SIZE DE FUTEBOL DE ROBÔS DA ROBOCUP

O futebol de robôs da RoboCup foi projetado para avançar o estado-da-arte em inteligência robótica através da competição amigável, com o objetivo eventual de conseguir o nível de jogo próximo ao humano por volta do ano de 2050 [2]. RoboCup consiste em equipes de robôs autônomos que competem uns com os outros em jogos de futebol, associado a um simpósio para a discussão de pesquisa na área. Há um número de ligas diferentes dentro de RoboCup, que são projetadas focando as diferentes partes do problema: desenvolvendo a inteligência das equipes de robô; este artigo é focalizado na liga do Small-Size (SSL).

Um jogo de futebol do SSL consiste em duas equipes de cinco robôs cada em um campo de 4.9m x 3.4m, com uma bola de golf da cor laranja [5]. Cada equipe deve ser completamente autônoma durante o jogo, com duração de dois tempos de 10 minutos. Aqui, a autonomia denota não haver nenhum ser humano envolvido em tomada de decisões enquanto o jogo estiver em andamento. Os comandos são transmitidos a cada robô da equipe através de RS-232 usando um protocolo padrão, por meio de um computador que executa o programa servidor [5]. Figura 1 mostra o esquema geralmente usado pelas equipes no SSL, assim como, as medidas do campo. O SSL é projetado para a focalização na autonomia da equipe, conseqüentemente, a visão global é feita mediante câmeras que transmitem as imagens para computadores fora do campo, que podem se comunicar com os robôs através do rádio sem fio.

O futebol de robô do SSL envolve muitas edições de pesquisa; os exemplos de alguns dos desafios da pesquisa incluem: Construir sistemas de controle

autônomos para tarefas dinâmicas com alta performance; Controle da equipe em um ambiente dinâmico, e respostas a oponentes desconhecidas; Limitações provocadas pelos sensores, como oclusão, incerteza, e latência; A manipulação rápida da navegação da bola em um ambiente dinâmico; Rápido, robusto, visão de baixa latência, com fáceis rotinas de calibração; Robôs de desempenho robusto com elevados mecanismos especializados para a manipulação da bola.

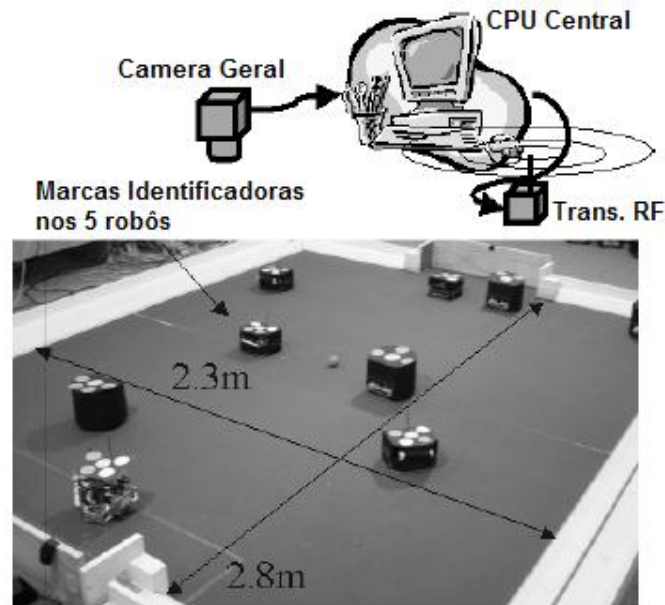


Figura 1: Uma vista geral da equipe small-size de futebol de robô.

Um jogo típico do SSL é altamente dinâmico, onde velocidades da bola é de 3 a 4 m/s^{-1} e velocidades dos robôs de 1 a 2 m/s^{-1} . Com tais velocidades em um ambiente pequeno, torna-se crítica a tradução rápida da informação em ação, para que a equipe seja responsiva aos eventos repentinos; se um robô chuta a bola a 3.5 m/s^{-1} , uma latência de 100 ms, significa que a bola terá movido-se por 35cm antes que os robôs possam responder à observação da bola ter sido chutada. Movimentos de alta velocidade e o impacto da latência no controle são os seguintes: A visão computacional e os algoritmos devem ser modelados de forma a filtrar o ruído e detectar eventos inesperados em tempo mínimo; Mecanismos de predição são requeridos para compensar, para a latência; A equipe e o controle do robô devem adaptar-se rapidamente às mudanças dinâmicas.

O último ponto significa que todas as decisões do controle necessitam de recálculos tanto quanto possível, permitindo o sistema reagir rapidamente aos eventos inesperados. Alguns sistemas recalculam tudo para cada frame, em uma taxa de 30Hz.[3] As decisões de alto nível mudam em uma taxa mais lenta do que decisões de baixo nível. Para uma trajetória aproximada, um jogo dura tipicamente 5-30s, quando uma tática puder operar-se sobre um frame de tempo de 1-30s, e uma habilidade puder operar sobre um frame de tempo 300 ms-5s. Entretanto, alguma decisão em todo o nível pode ser comutada em tempo mínimo de um frame (33ms) e responder a qualquer mudança dinâmica de grande escala.

3. HABILIDADES, TÁTICAS E JOGOS

Uma arquitetura usada para controlar tanto a equipe de robôs como cada robô individualmente, é apresentada em [1] e usada no time *CMDragons* da

Carnegie Mellon University, campeão do Aberto Americano e quarto colocado no Mundial, todos em 2003.

Esta arquitetura denominada STP das iniciais de Habilidades, Táticas e Jogos em inglês (*Skill, Tactics e Play*) apresenta-se na Fig. 2.

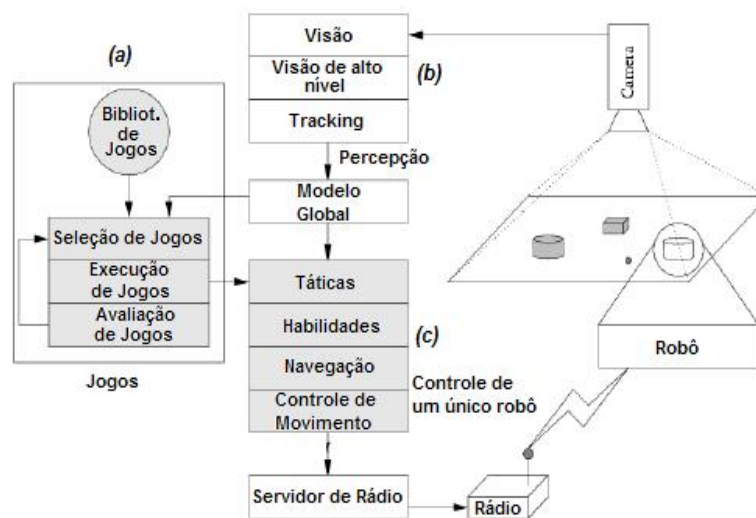


Figura 2 – Diagrama de Blocos da Arquitetura STP.

O ponto chave desta arquitetura é a divisão do comportamento de cada robô individualmente, feito por Fig.2(c), e o comportamento da equipe.

O comportamento da equipe é resultante da execução de uma seqüência coordenada de comportamentos de cada robô, onde jogos, táticas e habilidades interagem com cada robô da equipe.

As Habilidades são codificadas em algoritmos de controle em baixo nível para executar alguma ação complexa em cada robô, almejando alcançar algum objetivo em curto prazo; são as habilidades que dão forma às ações primitivas das táticas.

As Táticas encapsulam o que o robô deve fazer, para alcançar um objetivo específico em longo prazo. As táticas juntamente com as habilidades são responsáveis por avaliar o estado global da partida, de maneira a tomar a decisão do que fazer. Assim, algumas táticas determinam a melhor posição para se mover a fim de receber um passe; outras táticas defensivas avaliam como o robô oponente pode se mover para receber um passe e aonde posicionar o defensor para impedir que o oponente consiga este objetivo.

As táticas podem ser ativas ou não ativas; as táticas ativas são aquelas que envolvem a manipulação direta da bola, como por exemplo, efetuar um chute ao gol, efetuar um toque para um parceiro, entre outros.

Os Jogos, por sua vez, codificam como a equipe de robôs deve coordenar a execução de suas táticas a fim de conseguir os objetivos gerais da equipe. Nos Jogos cada membro recebe uma função; esta função define uma seqüência de Táticas e parâmetros a serem executados por cada robô. A atribuição das funções aos membros da equipe é executada dinamicamente em relação ao tempo e à necessidade apresentada na partida. A cada um dos papéis dados aos robôs, atribui-se uma tática para execução na etapa atual. Os jogos são repleto de táticas e habilidades.

Os jogos, táticas e as habilidades dão forma a uma hierarquia para o controle da equipe; os jogos controlam o comportamento da equipe, as táticas encapsulam ações individuais do comportamento de cada robô com as seqüências de habilidades, que serão executadas por cada um, gerando assim ações realmente úteis.

O controle dos robôs na arquitetura STP consiste em táticas e em habilidades, como foi descrito neste capítulo. O presente trabalho utiliza esta hierarquia de controle para propor um jogo defensivo, ou seja, será formulada uma forma de defesa para ser assumida pelos jogadores da equipe que, facilmente pode ser incorporada a esta arquitetura modular; na seção a seguir, esta proposta será apresentada e discutida.

4. PROPOSTA

Na seção anterior, foi apresentado o trabalho de [1] que propõe uma hierarquia para o controle da navegação dos robôs baseado em Jogos, Habilidades e Táticas, todas já definidas na mesma seção. Agora será proposto um modelo de programação matemática, o qual exprime um jogo defensivo para equipe, abstraindo as táticas e habilidades, pois, essas são inerentes ao papel que um defensor assumirá. Com isso, espera-se conseguir determinar as melhores posições para os jogadores baseados na posição dos adversários e da bola em campo.

Torna-se necessário salientar que o campo de jogo foi dividido em 532 quadros de 18cm², distribuídos em 28 linhas por 19 colunas; escolheu-se o valor de 18cm² devido a Lei 4 das regras da SSL F-180 [7], conhecida por lei do cilindro de 18cm; desta forma denota-se o campo como sendo uma matriz M x N com dimensões como ditas anteriormente.

4.1. ENTENDENDO A PROPOSTA

O primeiro objetivo da proposta é deixar os jogadores da equipe o mais próximo possível da bola. Isto será feito através da função objetivo do problema, apresentada na Eq. 1, empregando a distância Euclidiana entre cada jogador, exceto o goleiro e a bola.

As restrições aderidas ao problema, expressas na Eq. 2 e 3, restringindo as linhas e colunas respectivamente, tratam de garantir que dentro da área circundada por um robô adversário sempre exista pelo menos um robô da equipe exercendo o papel de marcador do robô adversário. Por exemplo, na Fig. 3, seja o quadro vermelho um robô adversário, assim, os quadros marcados entrelaçados são as possíveis coordenadas para posicionar um robô da equipe.

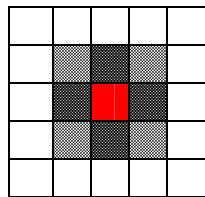


Figura 3 – Posicionamento dos robôs da equipe conforme restrição.

4.2. PROGRAMAÇÃO MATEMÁTICA

A função objetivo do problema é denotada pela seguinte expressão:

$$MinZ = \sum_{k=1}^{k=4} \sqrt{(X_k - X')^2 + (Y_k - Y')^2} \quad (1)$$

Onde:

X_k , representa a coordenada no eixo X do robô k,

Y_k , representa a coordenada no eixo Y do robô k,

X' , representa a coordenada no X da bola e

Y' , representa a coordenada no eixo Y da bola.

Sujeito a:

Restrição 1:

$$\left| X_k - \sum_{i=1}^{i=M} (R_{ijk} \cdot i) \right| \leq 1, \quad (2)$$

para $i = 1$ até M e $k = 1, 2, 3, 4$.

Restrição 2:

$$\left| Y_k - \sum_{j=1}^{j=N} (R_{ijk} \cdot j) \right| \leq 1, \quad (3)$$

para $j = 1$ até N e $k = 1, 2, 3, 4$.

Onde:

X_k , é a coordenada X do robô k no campo

Y_k , é a coordenada Y do robô k no campo

R_{ijk} , será 1 se o robô k do adversário estiver ocupando a área ij do campo.

Caso contrário será igual a 0.

Na próxima seção este modelo será implementado em uma plataforma computacional para posteriormente sofrer testes de cenários de partidas.

5. IMPLEMENTAÇÃO

Este modelo matemático foi implementado em Linguagem C; esta linguagem foi escolhida por se tratar de uma ferramenta bem difundida e facilmente integrada aos sistemas de controle empregados no controle dos robôs na competição.

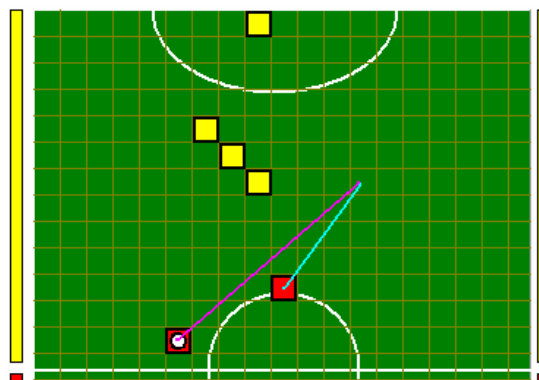
Esta implementação é apenas uma pequena parte da arquitetura sugerida por [1], encaixando-a no diagrama da Fig.2, esta implementação estaria executando dentro do bloco Jogos Fig.2(a), onde as informações recebidas do Modelo Global, que apresentariam as disposições dos objetos em campo, alimentariam as entradas desse procedimento, logo um jogo defensivo seria selecionado através do módulo de seleção de jogos na biblioteca de jogos, em tal caso, esta implementação executaria e daria seus resultados no módulo de execução de jogos, cabendo ao módulo de avaliação de jogos aprovarem seus resultados e os demais módulos fora do bloco Fig.2(a), efetivamente, colocá-lo em prática entre os robôs da equipe. Na Fig. 4 é apresentada a interface que foi construída para a proposta deste trabalho.



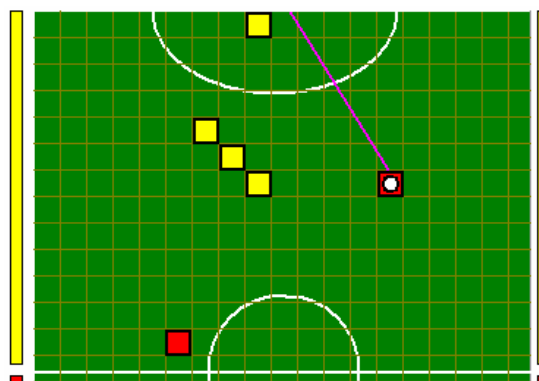
Figura 4 – Interface da Implementação do Modelo

6. TESTES E VALIDAÇÃO DA PROPOSTA

Agora, o modelo será submetido em algumas situações típicas de ataque que resultaram em gol. Os quadros vermelhos são os atacantes, enquanto os amarelos são defensores. Na Fig 5(a) e (b) tem-se uma situação de ataque, onde o sistema detecta um robô com maior possibilidade de alcançar o gol; este cenário foi retirado de uma partida do time japonês ToinAlbatross versus o norte-americano CMDragons.



(a)



(b)

Figura 5 – Contexto de um ataque adversário.

Na Fig.5(a) o robô atacante possui o domínio da bola, entretanto, o sistema detecta que ele não tem possibilidade de efetuar um tiro ao gol, dessa forma, o sistema planeja uma deflexão com o toque (linha magenta) e um deslocamento de um robô melhor posicionado (linha ciano). Já na Fig.5(b) o robô reposicionado e receptor da bola possui visão livre para efetuar o tiro (linha magenta) resultando em gol para a equipe atacante.

Na Fig.6 é apresentado como o modelo de jogo defensivo proposto neste trabalho sugeriria o posicionamento da defesa neste contexto.

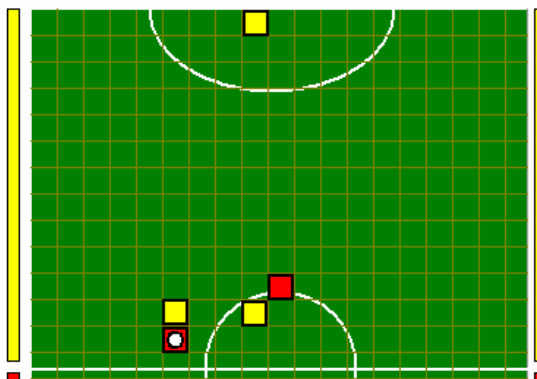


Figura 6 – Posicionamento da defesa conforme o modelo.

Conforme a função objetivo em Eq.1, os robôs devem ficar o mais próximo possível da bola, e obedecendo as restrições da Eq. 2 e 3, onde os robôs atacantes devem ser marcados pelo menos por um robô defensor, como acontece na Fig.6 com o robô atacante que não possui a bola. Ele é marcado por um robô defensor e este robô está na célula mais próxima da bola, o que dificulta o passe. Importante ressaltar que para fins de exemplo não foram considerados os demais robôs, entretanto caso haja a situação onde um robô defensor consiga cobrir a área de um ou mais robôs atacantes, então, o robô que resta irá de encontro à bola ou combater ao robô com a posse, como no exemplo na Fig. 7.

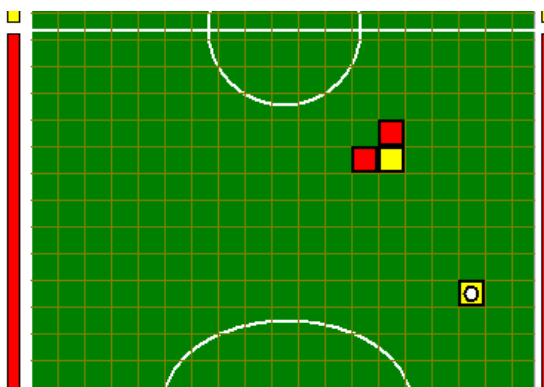


Figura 7 – Área de dois atacantes coberta por um robô defensor, restando um defensor para dominar a bola.

Outra situação típica retirada de uma partida entre RoboDragons versus CMDragons é a expressa na Fig. 8, onde a defesa é atraída por um robô atacante dominando a bola enquanto outro se posiciona e recebe o passe para disparar.

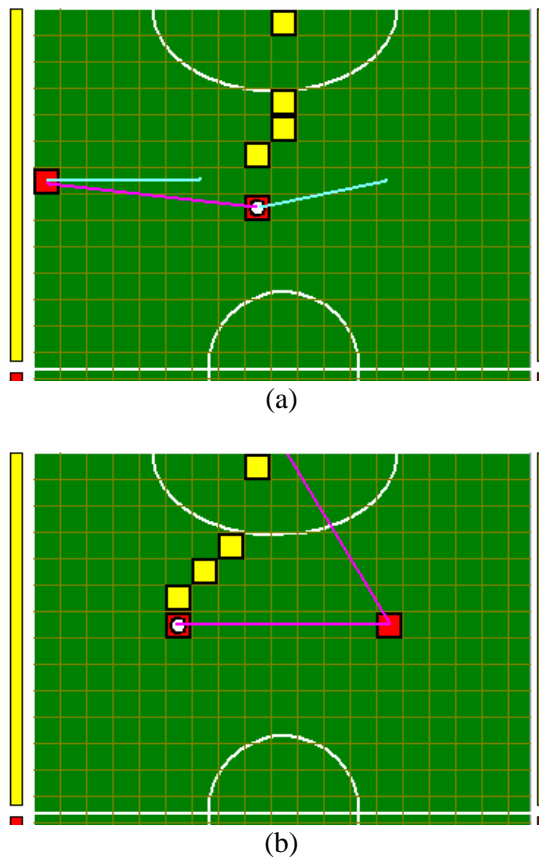


Figura 8 – Outra forma de se empregar a deflexão em uma partida de futebol de robôs.

Na Fig. 9 a defesa é disposta como sugere o modelo deste trabalho, novamente dificultando ao máximo a mudança de traçado da jogada original.

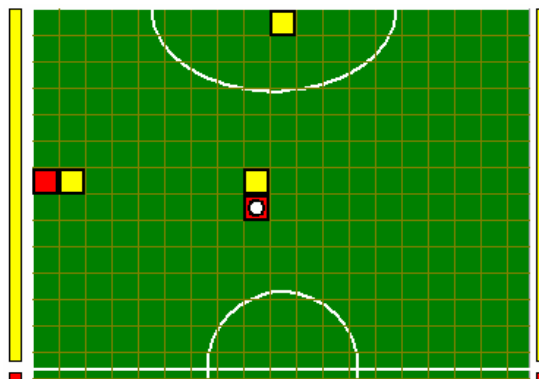


Figura 9 – Posicionamento sugerido pelo modelo defensivo proposto neste trabalho.

7. CONCLUSÃO E TRABALHOS FUTUROS

Este trabalho teve como finalidade modelar um jogo defensivo para uma partida de futebol de robôs da Robocup na categoria F-180 Small Size League, assim como implementar computacionalmente esse modelo, o qual permitisse que alguns cenários reais da competição fossem experimentados e conclusões fossem inferidas.

Na seção anterior, se apresentaram testes com situações reais de jogos que ocorreram nos principais campeonatos de futebol de robôs, onde a equipe atacante obteve sucesso no ataque, enquanto a equipe na defensiva usou técnicas de defesa tradicionais diversas. Entretanto, ao submeter o mesmo cenário à proposta deste trabalho percebeu-se que a equipe atacante teve maior dificuldade para alcançar o sucesso, ou seja, a probabilidade da equipe defensora de sofrer um gol tornou-se

menor.

Este modelo ainda não foi implementado em robôs reais, tão pouco agregado à uma arquitetura de controle como a STP, apresentada na Fig. 2, o que possibilitaria um teste real deste modelo e provaria sua eficiência fora do ambiente simulado. Ademais, outro ponto merecedor de uma pesquisa seria o enriquecimento das restrições do modelo, buscando aprimorar a marcação e otimizar os recursos robóticos.

8. REFERÊNCIAS BIBLIOGRÁFICAS

- [1] Brett Browning, James Bruce, Michael Bowling, and Manuela Veloso. STP: Skills, tactics and plays for multi-robot control in adversarial environments. *IEEE Journal of Control and Systems Engineering*, 2004.
- [2] Hiroaki Kitano, Minoru Asada, Yasuo Kuniyoshi, Itsuki Noda, and Eiichi Osawa. RoboCup: The robot world cup initiative. *Proceedings of the First International Conference on Autonomous Agents (Agents'97)*, ACM Press, Nova Iorque, 1997.
- [3] James Bruce and Manuela Veloso. Real-time randomized path planning for robot navigation. In *Proceedings of IROS-2002*, Suíça, Outubro de 2002.
- [4] L. Parker. Alliance: An architecture for fault-tolerant multi-robot cooperation. *IEEE Transactions on Robotics and Automation*, 1998.
- [5] M. Asada, O. Obst, D. Polani, B. Browning, A. Bonarini, M. Fujita, T. Christaller, T. Takahashi, S. Takokoro, E. Sklar, and G. A. Kaminka. An overview of RoboCup-2002 Fukuoka/Busan. *AI Magazine*, Spring 2003.
- [6] Rodney A. Brooks. A robust layered control system for a mobile robot. *IEEE Journal of Robotics and Automation*, Março de 1986.
- [7] Technical Committee of F180 Robocup 2005. Laws of the F180 League 2005a, Robocup Japão, 2005