

UNIVERSIDADE ESTADUAL PAULISTA "JÚLIO DE MESQUITA FILHO"
FACULDADE DE CIÊNCIAS - CAMPUS BAURU
DEPARTAMENTO DE COMPUTAÇÃO
BACHARELADO EM CIÊNCIA DA COMPUTAÇÃO

RAFAEL DE SOUZA SANTOS

**MECANISMO DE PREVISÃO DE POSIÇÃO PARA O TIME DE
FUTEBOL DE ROBÔS DA UNESP – BAURU**

BAURU
2017

RAFAEL DE SOUZA SANTOS

**MECANISMO DE PREVISÃO DE POSIÇÃO PARA O TIME DE
FUTEBOL DE ROBÔS DA UNESP – BAURU**

Trabalho de Conclusão de Curso do Curso
de Ciência da Computação da Universidade
Estadual Paulista "Júlio de Mesquita Filho",
Faculdade de Ciências, Campus Bauru.
Orientador: Prof. Dr. Rene Pegoraro

BAURU
2017

Rafael de Souza Santos

MECANISMO DE PREVISÃO DE POSIÇÃO PARA O TIME DE FUTEBOL DE ROBÔS DA
UNESP – BAURU/ Rafael de Souza Santos. – Bauru, 2017-

33 p. : il.

Orientador: Prof. Dr. Rene Pegoraro

Trabalho de Conclusão de Curso – Universidade Estadual Paulista "Júlio de Mesquita Filho"
Faculdade de Ciências
Ciência da Computação, 2017.

1. Robótica 2. Futebol de robôs 3. Previsão de posição

Rafael de Souza Santos

MECANISMO DE PREVISÃO DE POSIÇÃO PARA O TIME DE FUTEBOL DE ROBÔS DA UNESP – BAURU

Trabalho de Conclusão de Curso do Curso de Ciência da Computação da Universidade Estadual Paulista "Júlio de Mesquita Filho", Faculdade de Ciências, Campus Bauru.

Aprovado em 06/02/2017.

Banca Examinadora

Prof. Dr. Rene Pegoraro- Orientador
Departamento de Computação - FC - UNESP Bauru

Prof. Dra. Simone das G. D. Prado
Departamento de Computação - FC - UNESP Bauru

Prof. Dr. Wilson Massashiro Yonezawa
Departamento de Computação - FC - UNESP Bauru

Dedico esse trabalho a minha família.

Agradecimentos

Primeiramente gostaria de agradecer a minha família que me auxiliou e motivou conseguir concluir essa etapa. Agradeço principalmente a minha mãe e ao meu pai que nesses quatro anos se empenharam para que eu pudesse viver essa experiência de estudar em uma universidade pública e poder ter uma vida confortável fora de casa. Em especial ao meu pai, gratidão pelo apoio e pela companhia nos 800km semanais desse último semestre.

Meu muito obrigado ao orientador, Rene Pegoraro, por ter me passado tantos conhecimentos e por ter permitido que eu fizesse parte do time Carrossel Caipira, time esse que me proporcionou experiências significativas.

À minha namorada, Maria Júlia, por toda compreensão e apoio nesses difíceis quatro anos de convívio com a distância. Mesmo longe se fez muito presente em minha vida.

Agradeço aos meus amigos do Caronas do Barbs, pessoas que convivi a maior parte do tempo em Bauru e que se tornaram a minha família longe de casa.

À todos que tiveram participação nessa fase e que marcaram a minha vida, cada um da sua forma.

*"Many of the truths we cling to depend greatly on our own point of view."
(Yoda)*

Resumo

Este trabalho propõe uma forma de prever a trajetória dos componentes de uma partida de futebol de robôs da categoria *IEEE Very Small Size*. Para que houvesse a previsão, foi integrado ao programa do time Carrossel Caipira um simulador já existente. Foram realizados testes, em ambiente simulado, que possibilitaram afirmar que o método adotado melhora significativamente o desempenho do robô atacante, alvo da implementação.

Palavras-chave: Robótica, Futebol de robôs, Previsão de trajetórias.

Abstract

This paper proposes an approach to predicting the trajectory of the components of a soccer game of robots of the category IEEE Very Small Size. In order to have a prediction, an existing simulator was integrated into the team Carrossel Caipira's software. Tests were performed in a simulated environment, providing the possibility to affirm that the adopted method significantly improves the performance of the attacking robot, the target of the implementation.

Keywords: Robotics, Robot Soccer, Trajectory Prediction.

Lista de ilustrações

Figura 1 – Campo e configuração	13
Figura 2 – Ambiente do futebol de robôs sem previsão (instante de tempo t_i)	14
Figura 3 – Ambiente do futebol de robôs com previsão (instante de tempo t_{i+n})	14
Figura 4 – Ciclo de funcionamento do sistema	16
Figura 5 – Funcionamento filtro de Kalman	19
Figura 6 – Interface gráfica do simulador	22
Figura 7 – Simulador cliente/servidor	26
Figura 8 – Situação de cobrança de pênalti	28
Figura 9 – Mapa de calor do atacante sem previsão	30
Figura 10 – Mapa de calor do atacante com previsão	31

Lista de tabelas

Tabela 1 – Placar final de cada partida.	27
Tabela 2 – Gols contra de cada partida.	27
Tabela 3 – Número de pênaltis marcados por partida.	27
Tabela 4 – Número de gols marcados de pênaltis na partida.	28
Tabela 5 – Número de <i>freeballs</i> por partida.	28
Tabela 6 – Número de gols marcados pelo atacante em cada partida.	29
Tabela 7 – Número de gols contra marcados pelo atacante em cada partida.	29

Sumário

1	INTRODUÇÃO	12
1.1	Objetivo	13
1.2	Motivação	14
1.3	Organização do Trabalho	14
2	FUNDAMENTAÇÃO TEÓRICA	16
2.1	Futebol de Robôs	16
2.1.1	Software	16
2.1.1.1	Módulo da Visão	17
2.1.1.2	Módulo da Estratégia	17
2.1.1.3	Módulo do Controle	17
2.1.1.4	Módulo da Comunicação	18
2.1.2	Hardware	18
2.1.2.1	Câmera	18
2.1.2.2	Sistema de Comunicação	18
2.1.2.3	Robô	18
2.2	Previsão de Posições	19
3	DESENVOLVIMENTO	21
3.1	Simulador	21
3.2	Modularização do Simulador	22
3.3	Chamada do módulo da previsão	23
3.4	O módulo da previsão	23
3.5	Testes e análise dos resultados	25
3.5.1	Simulador cliente/servidor	25
3.5.2	Testes	26
3.5.2.1	Análise dos Resultados	26
3.5.2.2	Análise do Comportamento	29
4	CONCLUSÃO	32
4.1	Trabalhos Futuros	32
	REFERÊNCIAS	33

1 Introdução

O futebol de robôs na UNESP – Bauru surgiu, com o nome de Carrossel Caipira, um ano depois que o professor Jong-Hwan Kim, do Departamento de engenharia Elétrica do KAIST (Korean Advanced Institute of Science and Technology), da República da Coréia (Coréia do Sul), propôs e detalhou como deveria ser um time de futebol de robôs.

Em 1997, o time Carrossel Caipira foi criado e no ano seguinte ocorreu o primeiro triunfo da equipe, sagrando-se vice-campeã da primeira edição do campeonato brasileiro de futebol de robôs. Esse torneio permitiu o credenciamento da equipe do interior paulista a participar do campeonato mundial de 1998 na França. De uma parceria entre POLI – USP e UNESP – Bauru originou-se o time Guaraná, um dos representantes brasileiros na competição da França, onde o time Guaraná obteve o vice-campeonato mundial.

Após esses resultados expressivos o time Carrossel Caipira voltou a ser destaque no cenário de futebol de robôs conquistando o bicampeonato consecutivo no VI SBAI (Simpósio Brasileiro de Automação Inteligente) de 2003, em Bauru, São Paulo, e no VII SBAI em 2005, em São Luís, Maranhão.

Atualmente o time compete na categoria *IEEE Very Small Size* da LARC (Concurso Latino-Americano de Robótica), praticada por três robôs (goleiro, defensor e atacante)¹, cujo objetivo é idêntico ao esporte, conduzir a bola até o gol adversário e, ao mesmo tempo, defender o próprio gol de ataques adversários. Para que isso ocorra, cada robô deve se deslocar no campo de maneira controlada, evitando os outros robôs e procurando alcançar seu objetivo de ataque ou defesa.

Em Mackworth (1993), foi sugerido que o ambiente de futebol de robôs é adequado para avaliar o desempenho de técnicas de robótica autônoma e sua evolução em: Visão Computacional, Inteligência Artificial, Controle, Eletrônica e Mecânica.

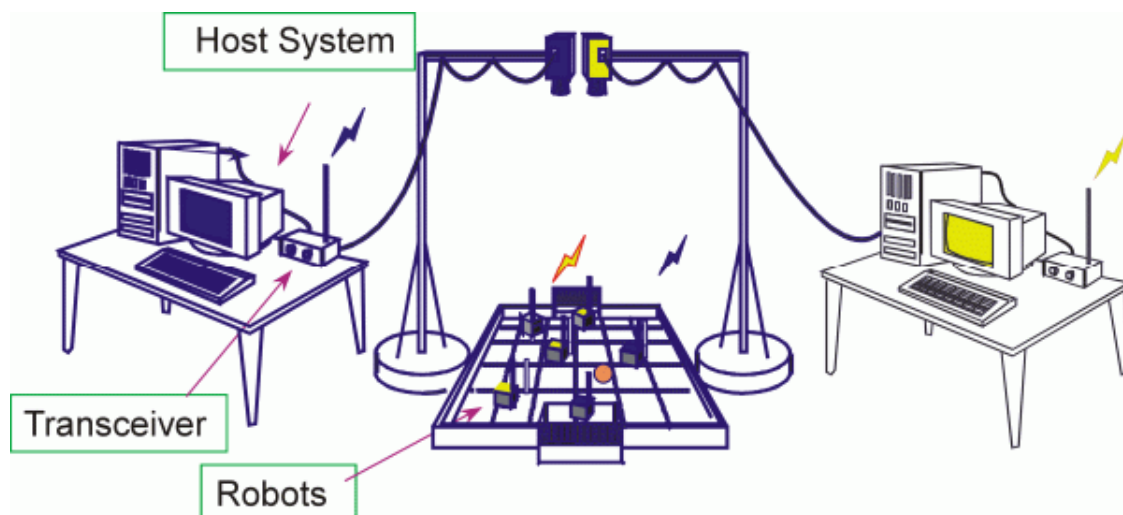
No campo da robótica, o Grupo de Integração de Sistemas e Dispositivos Inteligentes (GISDI), do Departamento de Computação (DCo) da UNESP de Bauru desenvolve iniciativas de baixo custo que auxiliam na consolidação dos fundamentos ensinados nos seus cursos. Além disso, é prestado um serviço a comunidade na forma de desenvolvimento de iniciativas que visam o auxílio ao aprendizado para crianças e adolescentes.

O ambiente de futebol de robôs da UNESP é dividido em 4 módulos principais, que são eles: visão, estratégia, controle e comunicação. Uma câmera é instalada a uma altura mínima de 2 metros acima do campo, para capturar imagens da partida que são transmitidas a um computador responsável por um ciclo de controle bem definido: primeiramente, [no módulo de

¹ <www.cbrobotica.org/wp-content/uploads/2014/03/VerySmall2009_ptbr.pdf>. Visita realizada em 27 de dezembro de 2016.

visão] as imagens são interpretadas, reconhecendo bola e jogadores; a seguir, são decididas [pelo módulo de estratégia] táticas e estratégias de jogo; as ações no domínio são definidas [pelo módulo de controle] em termos de comandos de movimento; e os comandos são enviados aos robôs [pelo módulo da comunicação], via comunicação sem fio (COSTA; PEGORARO, 2000) (Figura 1).

Figura 1 – Campo e configuração



Fonte: <http://www.fira.net/contents/data/MiroSot_Rules.doc> (Acesso em 27/12/2016).

O módulo de estratégia calcula, a partir das posições atuais observadas pela visão, a posição desejada onde cada robô deveria estar para que a jogada fosse realizada, desconsiderando o movimento e a posição futura dos elementos (robôs e bola). Como o futebol é um ambiente dinâmico, os cálculos da estratégia que são constantemente refeitos, podem gerar alterações nas posições desejadas dos robôs, a medida que os elementos se deslocam. Essas alterações das posições desejadas faz com que o robô oscile, uma vez que comandos de deslocamento são enviados aos robôs para que eles cheguem a posição determinada. Não considerar as posições finais dos elementos faz com que o robô seja dirigido para a situação atual da partida, que está em constante alteração, porém se excluir as constantes alterações à posição destino, a estratégia do robô é alcançada.

1.1 Objetivo

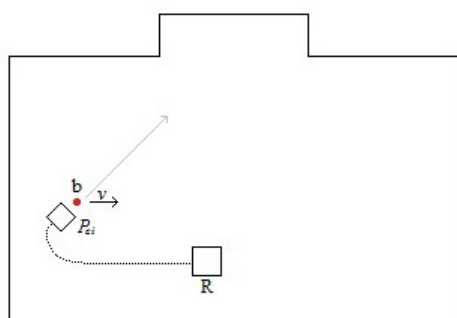
Este trabalho teve como objetivo desenvolver no time de futebol de robôs da categoria *IEEE Very Small Size*, do Departamento de Computação da UNESP de Bauru, um sistema de previsão de trajetórias que sirva de base para o módulo da estratégia decidir sobre as ações dos robôs em campo.

1.2 Motivação

O desenvolvimento e a implementação da previsão permite um avanço significativo ao time da UNESP de futebol de robôs. Além disso, não se observa nos *Team Description Paper*² apresentados nos Concursos Latino-Americanos de Robótica nenhuma equipe com tal desenvolvimento.

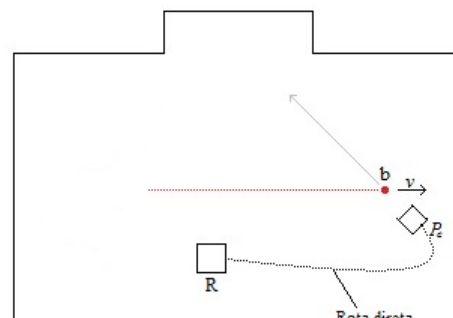
Um exemplo do problema citado acima pode ser observado na [Figura 2](#), onde no instante de tempo (t_i) o robô recebe o comando para ir para a posição de chute (p_{ci}) no início do cálculo da trajetória, porém com o passar do tempo, enquanto o robô se desloca para p_{ci} , a bola (b) vai se movimentando e o módulo de estratégia precisa ajustar a trajetória do robô para a nova posição de chute. Assim, se o módulo de estratégia puder prever a posição onde a bola estará, no momento do chute (t_{i+n}), desde t_i quando traçou a trajetória inicial, ele pode realizar uma rota direta para a posição de chute (p_c), sem sofrer ajustes constantes causados pelo deslocamento da bola ([Figura 3](#)).

Figura 2 – Ambiente do futebol de robôs sem previsão (instante de tempo t_i)



Fonte: elaborado pelo próprio autor.

Figura 3 – Ambiente do futebol de robôs com previsão (instante de tempo t_{i+n})



Fonte: elaborado pelo próprio autor.

1.3 Organização do Trabalho

O restante desse trabalho é dividido em 3 capítulos, sendo organizado da seguinte forma.

O capítulo 2 representa uma revisão bibliográfica sobre o time de futebol de robôs e previsão de posições em diferentes áreas da robótica. É explicado o funcionamento do sistema do time Carrossel Caipira, sua divisão em módulos e a integração das suas partes.

O capítulo 3 é destinado a descrever como este trabalho foi desenvolvido, como os testes foram realizados e por fim é feita uma análise dos resultados obtidos.

² <<http://www.cbrobotica.org/>> Visita realizada em 10 de dezembro de 2016.

No capítulo 4 é feita uma reflexão sobre os resultados dos testes apresentados no capítulo 3. Nele também são apresentados trabalhos futuros que podem complementar esse estudo.

2 Fundamentação Teórica

2.1 Futebol de Robôs

O futebol de robôs é uma iniciativa que visa promover o desenvolvimento da Inteligência Artificial e da Robótica por meio de uma tarefa comum, possibilitando realizar avaliação de teorias, algoritmos e arquiteturas de agente (KITANO et al., 1997).

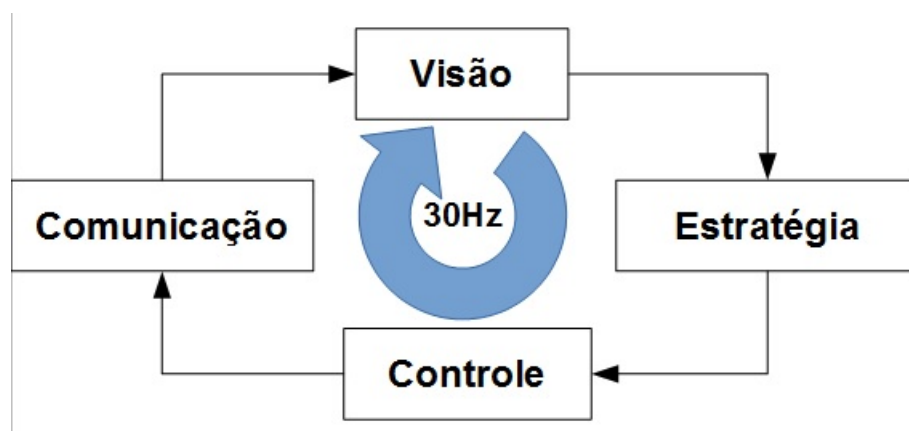
É necessário entender que a robótica surge da união do hardware [que executa as tarefas] e do software [que define as tarefas], e que a harmonia dessas partes, consegue amplificar o funcionamento do sistema.

2.1.1 Software

O software do futebol de robôs tem como objetivo entender o estado do ambiente, definir objetivos e se comunicar com os robôs. Para cumprir essas tarefas o time da UNESP de Bauru adota um modelo de quatro módulos principais, funcionando como um ciclo (Figura 4), que se repete 30 vezes por segundo.

São eles: visão, estratégia, controle e comunicação. A visão tem a função de obter o estado da partida por meio de uma câmera; a estratégia tem a função de entender e definir os objetivos a partir dos dados observados pelo primeiro módulo; o controle define quais comandos são necessários para que o robô chegue a posição idealizada pela estratégia; e, por fim, a transmissão deve comunicar aos jogadores robôs o comando definido nestes módulos.

Figura 4 – Ciclo de funcionamento do sistema



Fonte: elaborado pelo próprio autor.

2.1.1.1 Módulo da Visão

O módulo da visão é o primeiro elemento do ciclo do time Carrossel Caipira. Este módulo tem como objetivo, a partir de imagens globais do campo, processar a imagem capturada por uma câmera e entregar as coordenadas dos objetos para o próximo módulo.

O objetivo do processamento da imagem é fazer um reconhecimento dos objetos no campo – jogadores e bola – baseando-se nas cores da imagem capturada, sendo que um conjunto de *pixels* adjacentes de cor laranja representam a bola; de cor azul, os robôs de um time e de cor amarela, os robôs de outro time (COSTA; PEGORARO, 2000).

Após a identificação dos objetos no campo é necessário transformar toda essa informação em algo que possa ser usado para definir um objetivo. Para isso, a informação proveniente da identificação dos objetos é transformada em coordenadas cartesianas e orientações.

2.1.1.2 Módulo da Estratégia

Com as informações que chegam do módulo da visão, o módulo da estratégia necessita definir a posição desejada de cada robô para que a jogada seja realizada. Como o futebol é um ambiente dinâmico, os cálculos da estratégia são refeitos a cada ciclo do sistema que ocorre 30 vezes por segundo (Figura 4).

Cada robô possui uma função específica no campo de jogo e portanto uma estratégia diferente. O goleiro, deve se mover apenas na área a frente do gol para evitar um gol adversário. O volante e o atacante possuem funções de defesa e ataque, respectivamente, sendo essas funções passíveis de alteração caso um esteja em situação mais favorável que o outro. Dessa possibilidade de rotação de função entre o volante e o atacante surgiu o nome Carrossel Caipira, fazendo referência a seleção holandesa da Copa do Mundo de Futebol da FIFA de 74.

2.1.1.3 Módulo do Controle

Através desse módulo, é calculado o próximo comando que será enviado para os robôs, levando em conta a posição atual e a estratégia determinada, considerando as posições de possíveis obstáculos para a realização da movimentação do robô. Trinta desses comandos são enviados por segundo, sendo que cada um destes comandos é apenas um dos muitos que irão compor o acionamento dos robôs para levá-los à posição desejada pela estratégia.

Uma abordagem que leva em consideração a posição do objetivo e a posição de possíveis obstáculos para definição de um comando, é o planejamento de caminhos por Campos Potencias. Esse conceito, aplicado por Alves et al. (2015), é usado atualmente pelo time Carrossel Caipira e pode ser descrito como métodos capazes de construir uma função potencial, nas quais obstáculos têm potenciais repulsivos e metas têm potenciais atrativos.

2.1.1.4 Módulo da Comunicação

Esse módulo é responsável por transmitir os comandos resultantes do módulo do controle, através do rádio, aos robôs para que eles possam executar a trajetória idealizada pelo módulo de estratégia e controle.

2.1.2 Hardware

2.1.2.1 Câmera

Para a modalidade *IEEE Very Small Size* de futebol de robôs, cada time deve ter uma câmera que esteja posicionada sobre o campo, colocada a 2 metros de altura fornecendo as imagens para o módulo da visão do software.

O sistema de captura de imagem é composto por uma câmera, que recebe a imagem, e um digitalizador, que transforma a imagem em dados para serem processados digitalmente. No caso de uma *webcam*, o digitalizador já está integrado no dispositivo.

A câmera usada tem resolução 640x480 com um intervalo de captura de 1/30 de segundos entre cada imagem. Apesar deste intervalo, o tempo de atraso entre a obtenção da imagem do mundo até a sua entrega ao módulo da visão deve ser considerado. Esse atraso é causado principalmente, pelo tipo e modelo da câmera, a forma de digitalização e a transmissão da imagem para o computador.

2.1.2.2 Sistema de Comunicação

Para cada imagem processada são gerados comandos de acionamento dos motores que devem ser enviados aos robôs. Assim, um sistema de rádio é conectado ao computador que recebe os comandos escolhidos pelo módulo de controle do software e os envia.

2.1.2.3 Robô

O robô do time Carrossel Caipira é composto por uma placa Arduino, sensores de odometria acoplados a cada roda, dois motores e seus acionadores de potência. A estrutura física é formada por placas de acrílico que dão sustentação a todos os componentes da sua estrutura. Seguindo o padrão da categoria, os robôs não podem exceder o tamanho de um cubo de lado 7,5 cm.

Os motores usados são independentes, com caixa de redução de 30:1 e eixos alinhados, configurados em acionamento diferencial (*differential drive*) o que dá ao robô flexibilidade de movimentos.

Os comandos enviados pelo módulo de comunicação são recebidos pelos robôs através de seus rádios e indicam apenas a velocidade esperada para cada motor, para que o robô realize

a tarefa desejada. Internamente ao robô, um sistema de software embarcado escolhe a potência do acionamento dos motores para realizar o seu movimento.

2.2 Previsão de Posições

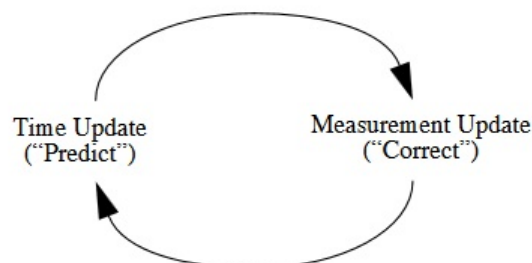
A ideia de prever algo vem da necessidade de identificar uma situação antes que ela aconteça e se preparar para enfrentá-la. A necessidade de prever algo aparece em muitas áreas da robótica, tendo em vista que a previsão é necessária para que o robô consiga completar ações simples como: se locomover, interceptar objetos em um ponto futuro ou esquivar de possíveis obstáculos em movimento.

Em [Andersson \(1988\)](#) é proposto uma abordagem de previsão em três dimensões onde um robô joga ping-pong com uma pessoa. Esta previsão determina onde a bolinha estará no momento do choque contra a raquete, posicionada em um braço mecânico controlado por computadores, permitindo assim a devolução da bola ao jogador humano. Esta abordagem também foi estudada por [Matsushima et al. \(2005\)](#) e [Huang et al. \(2011\)](#).

Em uma categoria muito parecida com a *IEEE Very Small Size*, a F180, a previsão da posição da bola é amplamente usada para mecanismos de passe entre os robôs. Esse tipo de abordagem, pode ser observada no trabalho do time da FEI ([SANTOS, 2013](#)), onde é utilizado um estimador recursivo para ambientes dinâmicos chamado Filtro de Kalman.

A ideia de [Kalman \(1960\)](#) é dividida em duas fases, a fase da previsão e a fase da validação ([Figura 5](#)). Começando com a previsão, é feito um chute inicial para o vetor de estados x'_0 . Após a previsão, ocorre uma propagação para o intervalo de tempo seguinte a fim de obter uma previsão para o vetor de estados, x'_k , no instante k . Assim, no instante k , é feita uma comparação entre o valor previsto e o valor real x'_k , atualizando-o para x_k ([TRAESSEL, 2014](#)).

Figura 5 – Funcionamento filtro de Kalman



Fonte: [Bishop e Welch \(2006\)](#).

Em [Costa e Pegoraro \(2000\)](#), é descrita uma espécie de previsão usada no time Guaraná e que também foi usada no time Carrossel Caipira por muitos anos.

"As posições, orientações e velocidades estimadas da bola e dos adversários são extraídas das imagens atual e anterior, assumindo velocidade constante. As estimativas referentes aos robôs do time são calculadas a partir da situação atual e dos últimos comandos enviados, de acordo com o seguinte modelo:

$$S_{r_{t'}} = f_r(S_{r_t}, c_t, c_{t-1}, \dots, c_{t-n})$$

onde $S_{r_{t'}}$ é a situação estimada, após um atraso Δt ; S_{r_t} é a situação no instante t ; $r = 1, 2, 3$ é o robô considerado; $c_t, c_{t-1}, \dots, c_{t-n}$ são os $n+1$ últimos comandos considerados e $f_r()$ é uma função ajustada empiricamente."

Fonte: [Costa e Pegoraro \(2000\)](#)

Tomando como exemplo o caso do goleiro, ele possui um estado que o posiciona na projeção da posição da bola no momento em que deverá entrar na linha defensiva. Essa projeção é calculada de acordo com a equação a seguir, sendo (x_b, y_b) a coordenada da bola; $(\Delta x_b, \Delta y_b)$ os deslocamentos da bola em x e y , observados nos últimos 33 milissegundos (últimos dois quadros); x_{gol} a linha x onde o robô goleiro deve permanecer para não sair da área nem invadir o gol (linha defensiva) ([COSTA; PEGORARO, 2000](#)).

$$y_{inters} = y_b - \frac{\Delta y_b \cdot (x_b - x_{gol})}{\Delta x_b}$$

Todas essas ideias de previsão podem ser aproveitadas para o futebol de robôs, já que em todos os casos a previsão da posição da bola possibilita a realização da jogada necessária. Assim, uma simplificação bidimensional do método desenvolvido para o ping-pong robótico junto com o processo de estimativa recursivo desenvolvido por Kalman ([KALMAN, 1960](#)) podem ser aplicados para realizar a previsão da posição da bola no ambiente de futebol de robôs da UNESP de Bauru. Porém uma nova abordagem será experimentada nesse trabalho através do emprego do simulador do time Carrossel Caipira já existente, que simula os próximos passos indicando o estado futuro (simulado) da partida.

3 Desenvolvimento

Esse capítulo é dedicado a descrever a inserção de um mecanismo de previsão de posição no time Carrossel Caipira. Para isso é usado um simulador, que tenta prever as posições da bola e robôs em um momento futuro próximo. Uma simplificação e modularização do simulador já existente foi desenvolvida, permitindo inseri-lo no software do futebol de robôs.

Como já existe um simulador funcional e que tem a função de simular o ambiente real para fins de testes, ele é utilizado, pois se encaixa perfeitamente como uma ferramenta que prevê a posição dos componentes da partida. Sendo assim, foi definido que integrá-lo ao programa do time de futebol de robôs possibilitaria alcançar o objetivo proposto por esse trabalho.

3.1 Simulador

Na robótica, o uso de ferramentas capazes de simular um determinado ambiente se faz muito viável para que consiga uma série de vantagens, como as levantadas por [Wolf et al. \(2009\)](#):

"i) economia de recursos financeiros, pois diversos testes podem ser realizados antes de ser implementado fisicamente o robô; ii) economia de tempo, pois podemos realizar um maior número de experimentos através de simulação, nos quais a realização de um experimento requer um menor tempo para configurar o experimento (não há necessidade de recarregar baterias e posicionar equipamentos e objetos), além de se poder "acelerar o tempo" do relógio virtual na execução das simulações; iii) evitar danos ao robô, pois através das simulações pode-se verificar previamente as situações que podem provocar danos ao robô, devido por exemplo a fortes colisões, acionamento indevido dos motores, ou exposição do robô à ambientes perigosos para testes; iv) evitar acidentes e aumentar a segurança, pois através da simulação pode-se realizar diversos testes visando garantir uma maior segurança e robustez do sistema robótico, evitando assim a incidência de acidentes com pessoas e com elementos presentes no ambiente de atuação do robô, permitindo inclusive uma melhor análise de como adicionar novos dispositivos de salvaguarda em hardware e software que permitam aumentar a segurança e confiabilidade dos robôs; v) aperfeiçoamento do hardware e software, uma vez que se pode, através de simulações, testar diferentes configurações de hardware, bem como testar e avaliar novas implementações e ajustes nos parâmetros do software de controle, permitindo assim uma melhoria do sistema como um todo e a otimização do uso dos recursos disponíveis a fim de obter uma maior eficiência do sistema robótico."

Apesar da simulação ser uma importante aliada da robótica móvel, ela tem uma desvantagem muito marcante, não pode substituir totalmente os testes em ambiente real. Apesar de existirem várias ferramentas que simulam a realidade procurando ser o mais fiel possível, seus resultados não são exatos e podem conter falhas ([SANTOS, 1999](#)).

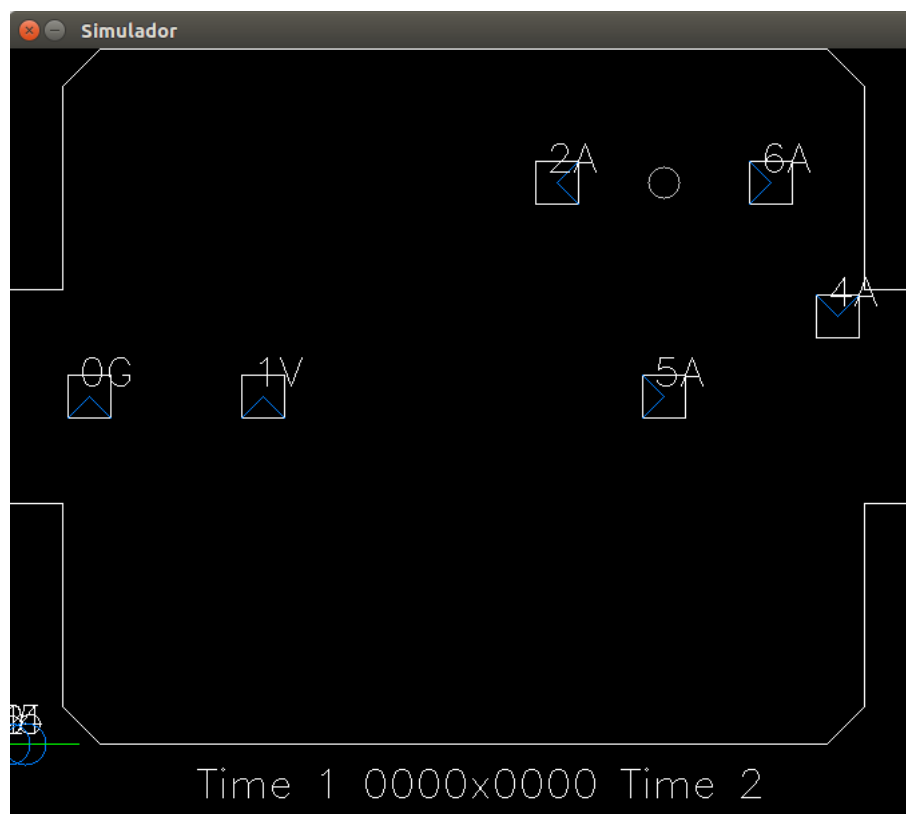
O simulador do time Carrossel Caipira permite que todas as vantagens listadas acima sejam aplicadas, além de poder simular uma partida completa, ele possui comandos interativos que possibilitam que situações específicas sejam testadas.

3.2 Modularização do Simulador

Neste trabalho, o simulador precisou sofrer alterações para viabilizar a sua inserção ao sistema e realizar a função de prever os próximos movimentos dos componentes da partida. Sendo assim, algumas alterações precisaram ser feitas no programa existente do simulador, como a remoção de controles interativos e da interface visual.

Controles e funções interativos não tem serventia nesse caso, pois o simulador tem a função de simular alguns ciclos do sistema e situações muito específicas, que serão tratadas no decorrer desse capítulo. Já a interface visual (Figura 6) não é útil ao sistema automático de previsão, podendo ainda causar atrasos no processamento.

Figura 6 – Interface gráfica do simulador



Fonte: elaborado pelo próprio autor.

Para o simulador se tornar um módulo que possa ser chamado durante a execução do módulo da estratégia, foi necessário a criação de uma biblioteca chamada Simulador. Nela há apenas o escopo da função Simulador que terá sua lógica explicada nas próximas sessões.

A seguir, o Simulador é tratado como um novo módulo do sistema. Tal módulo, será chamado de previsão.

3.3 Chamada do módulo da previsão

Como a previsão serve de base para o módulo da estratégia, a previsão deve ocorrer antes da definição do próximo objetivo. Sendo assim, o módulo foi inserido entre o módulo da visão e da estratégia.

A previsão é realizada apenas para o robô que tem a função de ataque, afinal essa posição possibilita uma análise melhor dos resultados, levando em consideração os gols efetuados. A partir da análise da viabilidade do método adotado, expandir a implementação para os outros robôs se torna algo simples para um trabalho futuro.

Em uma partida existem dois momentos, o de previsão e o de jogo. Esses dois momentos são definidos por estados no programa, sendo eles *emSimulacao* e *emJogo*. Enquanto o sistema está *emJogo*, todos os módulos trabalham para que cada jogador exerça sua função. Já no estado *emSimulacao*, suas funções se restringem em simular o comportamento de todos os componentes por um determinado espaço de tempo, porém sem que haja o acionamento dos robôs.

Apesar de serem dois momentos com características diferentes, o estado *emSimulacao* é uma especificação do estado *emJogo* e está contido dentro dele. Como a previsão precisa simular os próximos acontecimentos da partida, são utilizados os mesmos módulos do sistema para essa tarefa. Assim, o estado *emSimulacao* serve para limitar apenas uma simulação por ciclo, ou seja, quando estiver no estado *emSimulacao* não é possível chamar o módulo da previsão novamente, como pode ser observado no Algoritmo 1.

Algoritmo 1 – CHAMADA DA PREVISÃO

1. **Se** não está *emSimulacao* e está *emJogo*
2. └ **Chama** módulo da previsão

3.4 O módulo da previsão

O módulo da previsão é a parte mais importante desse trabalho, pois é nele que a previsão proposta ocorre.

O sistema de software do futebol de robôs usa estados globais que servem para indicar comportamentos desejados ao programa. A inserção do módulo de previsão, exige que todos

os estados globais usados nesse módulo sejam salvos no início do seu funcionamento, para que ao seu final seja possível retorná-los ao seu estado inicial, caso a previsão não seja considerada satisfatória.

Como o sistema é executado em um ciclo, que se repete 30 vezes por segundo, usa-se uma previsão de 30 interações, que corresponde a 1 segundo. A escolha foi empírica, levando em conta que uma previsão muito longa pode ser difícil de realizar e muito curta não traria vantagens. Acredita-se que é o suficiente para melhorar o desempenho e continuar tendo um previsão realista.

O funcionamento deste módulo é descrito no Algoritmo 2, como é explicado a seguir: no início, são salvos os estados de todos os componentes da partida; após, é executada a previsão, vide Algoritmo 3; com os novos estados resultantes da previsão, é aplicado um teste de aproveitamento para o atacante, vide Algoritmo 4; por fim, os estados são recuperados nos demais componentes para que o sistema siga para o próximo módulo (estratégia).

Algoritmo 2 – ALGORITMO GERAL

1. **Salva** estados
2. **Executa** previsão
3. **Testa** viabilidade para o atacante
4. | **Se** é viável, mantém previsão
5. | **Senão**, descarta os dados previstos e recupera estados anteriores
6. **Recupera** estados anteriores dos demais componentes

A cada início de ciclo, é necessário executar uma vez o módulo da estratégia, utilizando a posição de cada componente, a fim de encontrar posições objetivas iniciais para cada um dos jogadores. Então, em cada uma das 30 interações do módulo da previsão, que ocorre a cada 33 milissegundos, a previsão precisa identificar a posição de cada componente da partida (bola, goleiro, volante e atacante) e atualizá-las, como pode ser observado no Algoritmo 3.

Algoritmo 3 – FUNCIONAMENTO DA PREVISÃO

ENTRADA: Posição de todos os componentes da partida

1. **Muda** variável em Simulacao para verdadeiro
2. **Executa** estratégia
3. **Repete** por 30 vezes
4. | **Para cada** componente da partida
5. | | **Calcula** próxima posição

6. └─ **Posiciona** o componente virtualmente
- 7.
8. **Aplica** condição de aproveitamento da previsão
9. **Muda** variável em Simulação para falso

Sabendo que a previsão se trata de uma ação futura, é necessário entender que o resultado da sua execução será um ponto objetivo. Considerando que em alguns casos o ponto encontrado pode não se tornar realizável pelo sistema, foi desenvolvida uma condição (Algoritmo 4) para saber se esse ponto deve ou não ser aproveitado.

Algoritmo 4 – CONDIÇÃO DE APROVEITAMENTO DA PREVISÃO

1. **Se** o atacante está a um raio de 4cm da bola
2. └─ O objetivo do atacante é igual ao objetivo previsto
3. **Caso contrário**
4. └─ Ignora a previsão e o objetivo do atacante é igual ao estado salvo

A partir do aproveitamento ou não da previsão, todos os estados, com exceção do novo objetivo, precisam voltar para o seu valor inicial para que o sistema siga seu fluxo sem que haja comprometimento.

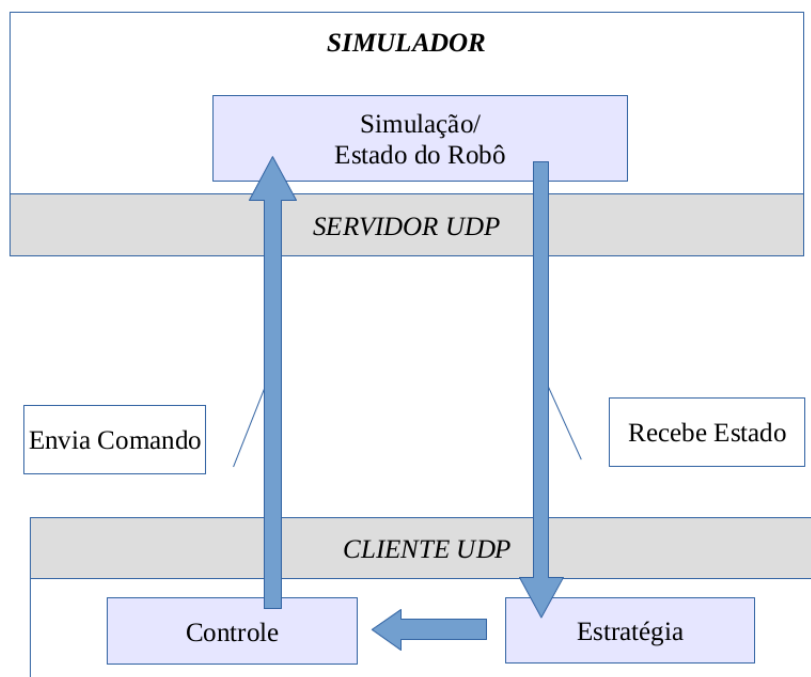
3.5 Testes e análise dos resultados

Para executar os testes foi utilizado um outro simulador já existente, que foi desenvolvido em um trabalho de conclusão de curso do ano de 2015 (SANTOS, 2015). Esse simulador é capaz de se comunicar através da rede por meio do protocolo *User Datagram Protocol* (UDP).

3.5.1 Simulador cliente/servidor

Esse simulador, como mostra a Figura 7, é capaz de se comunicar com dois times clientes, cada um com sua estratégia e controle, com ambos se comunicando com um servidor do simulador, que recebe os comandos enviados por cada cliente, realiza a movimentação dos robôs e da bola, e faz o papel da visão fornecendo o estado de cada robô para os times clientes (SANTOS, 2015).

Figura 7 – Simulador cliente/servidor



Fonte: Santos (2015).

3.5.2 Testes

Tendo um simulador capaz de suportar dois times diferentes, foram idealizadas partidas entre um time com o mecanismo de previsão de posição implementado e um time idêntico, com a mesma estratégia e controle, porém sem a previsão.

Para que não houvesse dúvidas a respeito da eficiência, foi pensado que apenas uma partida com dois tempos de 5 minutos, como é o padrão da categoria, não seria suficiente para uma avaliação com confiabilidade estatística. Sendo assim, foram feitos 5 partidas com a duração de 10 horas cada uma. Em cada partida seus dados foram salvos em arquivos de texto para que houvesse um registro passível de ser analisado.

3.5.2.1 Análise dos Resultados

As tabelas a seguir representam os dados colhidos em cada uma das partidas citadas acima. Para facilitar o entendimento, cada dado será tratado separadamente. Tendo em vista que o critério adotado para avaliar o desempenho é o resultado final de cada partida, pode-se observar na [Tabela 1](#) que o time com previsão conseguiu ser mais efetivo que o time sem previsão, ou seja, o time que mais fez gols no período determinado de teste.

É possível afirmar a partir da análise da [Tabela 1](#), que em média o time com previsão marcou 216.8 gols por partida, contra 112.4 do time sem previsão. Esse número representa um

aumento de 192.88% gols do sistema por partida.

Tabela 1 – Placar final de cada partida.

	Time c/ previsão	Time s/ previsão
Partida 1	195	116
Partida 2	201	123
Partida 3	228	86
Partida 4	210	112
Partida 5	250	125

Apesar de ser possível perceber uma leve melhora no número de gols contra na [Tabela 2](#), os times tem um desempenho quase que equivalente nesse quesito. O mais importante a ser observado é que tanto o time com previsão quanto o time sem previsão marcam muitos gols contra.

Tabela 2 – Gols contra de cada partida.

	Time c/ previsão	Time s/ previsão
Partida 1	94	111
Partida 2	98	103
Partida 3	74	124
Partida 4	97	128
Partida 5	105	139

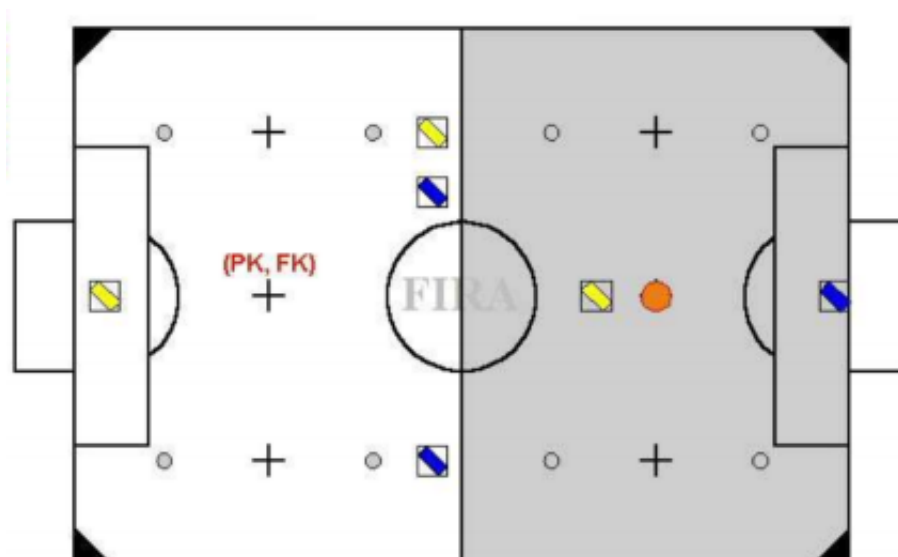
Outros dados importantes a serem notados são o número de pênaltis e a sua taxa de conversão por partida. Essa situação acontece quando a bola fica parada dentro da área de defesa e o goleiro não a retira no período de 10 segundos ou quando a bola entra na área e há mais de um defensor dentro dela. Quando isso ocorre, um tiro livre direto é cobrado de uma marca localizada a 37.5 centímetros do gol, como ilustra a [Figura 8](#).

Os dados contabilizados na [Tabela 3](#) mostram que o time sem previsão comete mais pênaltis que o time com previsão, porém esse dado se justifica pelo fato do time com previsão marcar mais gols e em decorrência disso passar mais tempo no ataque. Quanto a eficiência na cobrança das penalidades pode ser observado na [Tabela 4](#) que se mantém praticamente igual. Assim sendo , não é possível avaliar o desempenho da implementação através desse dado.

Tabela 3 – Número de pênaltis marcados por partida.

	Time c/ previsão	Time s/ previsão
Partida 1	20	38
Partida 2	16	30
Partida 3	11	36
Partida 4	16	30
Partida 5	16	46

Figura 8 – Situação de cobrança de pênalti



Fonte: <www.cbrobotica.org/wp-content/uploads/2014/03/VerySmall2009_ptbr.pdf> (Acesso em 27/12/2016).

Tabela 4 – Número de gols marcados de pênaltis na partida.

	Time c/ previsão	Time s/ previsão
Partida 1	27	18
Partida 2	23	12
Partida 3	30	10
Partida 4	21	16
Partida 5	33	11

Com relação ao número de *freeballs*, representados na Tabela 5, que corresponde a impasses que ocorrem fora das áreas de defesa, é possível afirmar que seu número é muito elevado, em média 2369.8 por partida. Como são muitas as situações que se deve esperar 10 segundos para marcar o *freeball* e parar a partida, na implementação do simulador usado para os testes, subseção 3.5.1, foi reduzido o tempo da condição de parada da partida de 10 para 4 segundos, tornando assim a partida mais dinâmica.

Tabela 5 – Número de *freeballs* por partida.

Partida 1	2367
Partida 2	2384
Partida 3	2370
Partida 4	2378
Partida 5	2350

Como todo o desenvolvimento desse trabalho é feito na posição de atacante, a Tabela 6 e Tabela 7 retratam o alvo da implementação proposta nesse trabalho. Analisando separadamente, pode-se observar que o jogador com previsão teve uma enorme melhora do seu desempenho,

fazendo em média 95.8 gols por partida contra 16.2 do seu adversário sem previsão (Tabela 6). Quanto aos gols contra, Tabela 7, sua vantagem mostra-se mais evidente, chegando a passar três testes sem ao menos um gol contra, enquanto o jogador sem previsão fez em média 61.2 gol contra por teste.

Tabela 6 – Número de gols marcados pelo atacante em cada partida.

	Jogador c/ previsão	Jogador s/ previsão
Partida 1	84	19
Partida 2	98	21
Partida 3	104	11
Partida 4	82	13
Partida 5	111	17

Tabela 7 – Número de gols contra marcados pelo atacante em cada partida.

	Jogador c/ previsão	Jogador s/ previsão
Partida 1	0	55
Partida 2	2	48
Partida 3	0	68
Partida 4	0	63
Partida 5	2	72

3.5.2.2 Análise do Comportamento

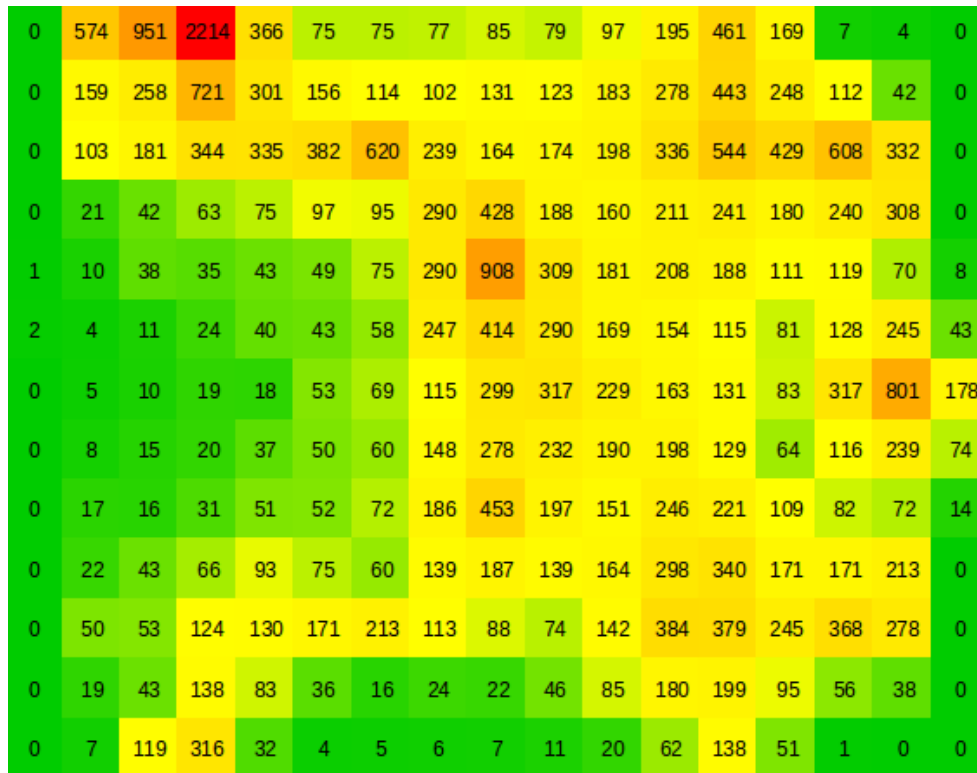
Nessa subseção, será feita uma análise do comportamento do robô atacante durante os testes.

Para que fosse possível realizar essa análise, foi integrado ao simulador, uma função que salva a posição do robô atacante dos dois times durante o teste. Como cada teste teve cerca de 1000000 de interações e que cada uma corresponde a 33 milissegundos, foi decidido que a posição seria salva apenas a cada 1 segundo e não em cada uma das 30 interações que acontecem a cada segundo, para que não fosse comprometida a performance do sistema, tendo em vista que isto não afeta significativamente os dados obtidos, pois em um segundo o robô se desloca pouco.

Na Figura 9 e na Figura 10 os números representam quantitativamente as ocorrências daquela posição no teste realizado. Quanto a cor, é representado em níveis que variam do verde ao vermelho, de acordo com a intensidade da ocorrência, sendo o menor e o maior número de incidências, respectivamente.

A Figura 9 representa o comportamento do robô sem previsão. Nela o robô sem previsão ataca na direção esquerda. Portanto, é possível visualizar que ele permanece a maior parte do tempo no campo de defesa e que também não consegue uma posição favorável de ataque na maior parte do teste.

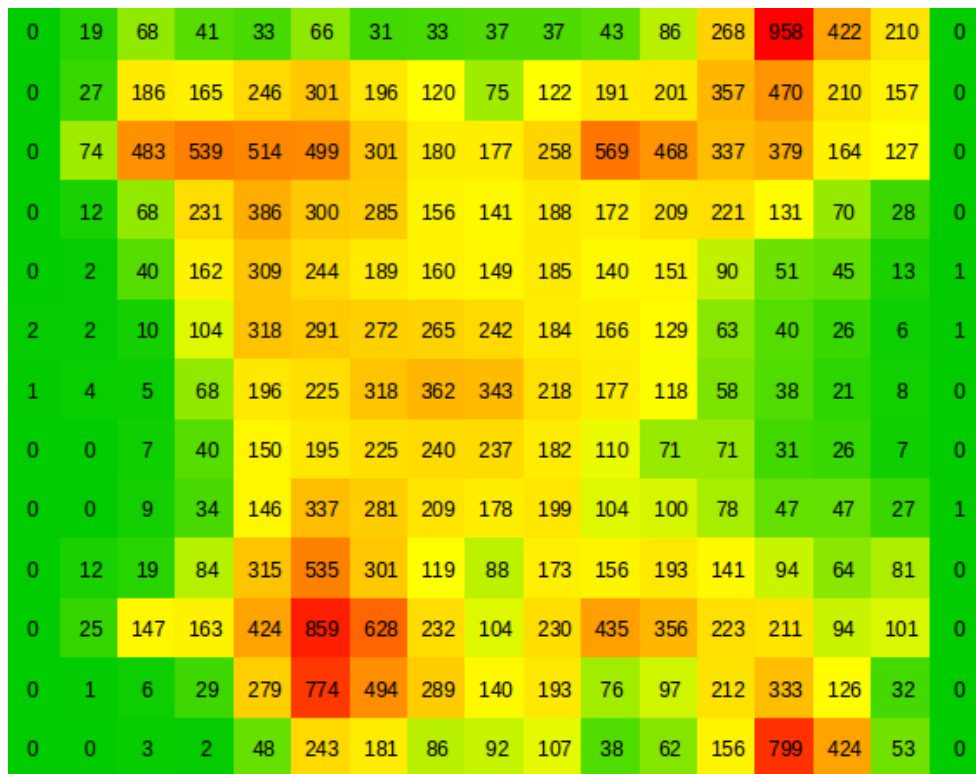
Figura 9 – Mapa de calor do atacante sem previsão



Fonte: elaborado pelo próprio autor.

Já na [Figura 10](#), o robô com previsão ataca para direita e fica claro que ele se posiciona melhor que o seu adversário de mesma função. Apesar de também não passar muito tempo no ataque, ele passa a maior parte do seu tempo mais próximo do ataque, pode-se observar também que ele é muito mais efetivo, como nos dados apresentados na [Tabela 6](#).

Figura 10 – Mapa de calor do atacante com previsão



Fonte: elaborado pelo próprio autor.

4 Conclusão

A partir de todos os dados apresentados, percebe-se que a previsão implementada é uma solução promissora e que melhorou o desempenho do time Carrossel Caipira, mesmo que os testes tenham sido realizados apenas em um simulador. Porém, para uma melhor confiabilidade dos resultados, os experimentos deverão ser repetidos no ambiente real.

A principal dificuldade desse trabalho foi encontrar desenvolvimentos similares ao proposto para que fosse possível ter um maior embasamento, tendo em vista que nenhum time da categoria *IEEE Very Small Size* utiliza um método de previsão de trajetória.

Levando em consideração o objetivo desse trabalho, ele foi alcançado, pois foi desenvolvido uma nova perspectiva de implementação de um mecanismo de previsão de posição para a categoria *IEEE Very Small Size*, do Departamento de Computação da UNESP de Bauru.

4.1 Trabalhos Futuros

Como trabalhos futuros sugere-se que haja uma verificação em ambiente real do método implementado, repetindo os testes com robôs reais. Constatando a eficiência do método adotado, é proposto expandir a implementação para os outros robôs. Sugere-se ainda, que seja feito melhorias no simulador atual do time Carrossel Caipira, para que os resultados em ambiente simulado possam ser mais condizentes com a realidade.

Referências

- ALVES, S. F.; LISBOA, C. de O.; SANTOS, R. de S.; FILHO, H. F.; FRANCISCO, E. K.; SANTOS, M. B.; PEGORARO, R. O time carrossel caipira de futebol de robôs. 2015.
- ANDERSSON, R. L. *A Robot Ping-Pong Player: Experiment in Real-Time Control*. [S.l.]: MIT Press Cambridge, 1988.
- BISHOP, G.; WELCH, G. *An introduction to the kalman filter*. Department of Computer Science, University of North Carolina. [S.l.]: Chapel Hill, NC, unpublished manuscript, 2006.
- COSTA, A. H. R.; PEGORARO, R. Construindo robôs autônomos para partidas de futebol: o time guaraná. *SBA Controle Automação*, v. 11, n. 03, p. 141–149, 2000.
- HUANG, Y.; XU, D.; TAN, M.; SU, H. Trajectory prediction of spinning ball for ping-pong player robot. In: IEEE. *2011 IEEE/RSJ International Conference on Intelligent Robots and Systems*. [S.l.], 2011. p. 3434–3439.
- KALMAN, R. E. A new approach to linear filtering and prediction problems. *Journal of basic Engineering*, American Society of Mechanical Engineers, v. 82, n. 1, p. 35–45, 1960.
- KITANO, H.; ASADA, M.; KUNIYOSHI, Y.; NODA, I.; OSAWA, E.; MATSUBARA, H. Robocup: A challenge problem for ai. *AI magazine*, v. 18, n. 1, p. 73, 1997.
- MACKWORTH, A. K. On seeing robots. 1993.
- MATSUSHIMA, M.; HASHIMOTO, T.; TAKEUCHI, M.; MIYAZAKI, F. A learning approach to robotic table tennis. *IEEE Transactions on robotics*, IEEE, v. 21, n. 4, p. 767–771, 2005.
- SANTOS, E. Sistema de passe para o futebol de robôs. 2013.
- SANTOS, M. B. Planejamento de caminhos para robôs autônomos utilizando campos potenciais: Avaliação de métodos através de simulação para utilização pelo time de futebol de robôs carrossel caipira. 2015.
- SANTOS, M. P. dos. Introdução à simulação discreta. *Rio de Janeiro: UERJ*, 1999.
- TRAESEL, F. A. Desenvolvimento de sistema de estimação de posição empregando filtro de kalman e gps/ins. 2014.
- WOLF, D. F.; SIMÕES, E. do V.; OSÓRIO, F. S.; JUNIOR, O. T. Robótica móvel inteligente: Da simulação às aplicações no mundo real. In: *Mini-Curso: Jornada de Atualização em Informática (JAI), Congresso da SBC*. [S.l.: s.n.], 2009. p. 13.