

Daniel Tavares Malheiro

Orientador: Prof. Dr. Flavio Tonidandel

Sistema de Estratégia para Posicionamento Eficiente da Defesa no Futebol de Robôs

Relatório Final do Projeto de Iniciação
Científica do Laboratório de Robótica e In-
teligência Artificial da FEI, como requisito para
a conclusão da Iniciação Científica.

São Bernardo do Campo, SP

Novembro de 2010

Resumo

O presente trabalho tem como principal objetivo, o aprimoramento do sistema de estratégia de defesa da equipe de futebol de robôs, categoria F-180. Este trabalho tem como base, algumas das soluções apresentadas pelas principais equipes da categoria. Um dos problemas clássicos apresentados na categoria é a do jogador precisar se antecipar para saber qual a melhor jogada e a que ele terá mais chances de realizar gol, interceptar a bola e principalmente estar preparado para a estratégia da equipe adversária independentemente do nível de desenvolvimento da equipe adversária.

Além disso, uma das principais preocupações deste trabalho é com a utilização de forma eficiente dos jogadores, pois, como por regra, são apenas cinco jogadores incluindo o goleiro, deve-se ter uma equipe cada vez mais dinâmica para que os adversários sejam surpreendidos .

Palavras-chave: sistema de estratégia, defesa, eficiente, filtro de kalman.

Conteúdo

1	Introdução	1
1.1	O Futebol de Robôs	1
1.2	A Categoria Small-Size	2
1.3	O Futebol de Robôs na FEI	4
1.4	A Equipe RoboFEI	4
1.5	Objetivo	6
2	Revisão Bibliográfica	7
2.1	Sistemas de Previsão	7
2.1.1	Previsao Física	9
2.1.2	Método dos Mínimos Quadrados (MMQ)	10
2.1.3	Filtragem de Kalman	12
2.2	Sistema de Posicionamento Eficiente	16
2.2.1	Condições de defesa	17
3	Metodologia	18
3.1	Ferramentas Utilizadas	18
3.1.1	OpenCV	18
3.1.2	Doxygen	20
3.1.3	Simulador	21
3.2	Implementação de Algoritmos	22
3.2.1	Filtro de Kalman	22
3.2.2	Funções primitivas construídas	23
4	Resultados Finais	27
4.1	Análise Comparativa das Técnicas	32

5	Cronograma	36
5.1	Desenvolvimento do projeto	36
5.2	Cronograma mensal	37
5.3	A ROBOCUP 2010	38
5.4	Panorama para a ROBOCUP 2011	38
6	Trabalhos Futuros	39

Lista de Figuras

1.1	Robôs da categoria Small-Size	2
1.2	Sistema de Comunicação da categoria Small-Size	3
2.1	Ciclo de comunicação entre software e robô . Fonte: [Veloso, 1997]	8
2.2	Jogada aérea do robô A para o robô B	14
2.3	Finalização da jogada com o robô B	15
2.4	Probabilidade medida de o atacante acertar o gol: A largura dos corredores desprotegidos para realizar o gol é computada (verde) Fonte: [Blank, 2009]	17
3.1	Simulador Soccerbots. Fonte: [TeamBots, 2010]	22
3.2	Funções DeltaDefence e Block habilitadas. Fonte: [Laue, 2009]	26
4.1	Tela do protótipo implementado para a filtragem de kalman Fonte: Autor.	28
4.2	Gráfico de movimento da bola X versus Y Fonte: Autor.	29
4.3	Gráficos de verificação de erro na previsão de posições Fonte: Autor.	30
4.4	Gráficos de verificação de erro na previsão de posições com erros injetados Fonte: Autor.	31
4.5	Software de estratégia Fonte: Autor.	32
4.6	Análise comparativa entre as técnicas de previsão Fonte: Autor.	33

Lista de Tabelas

3.1	Bibliotecas OpenCV	20
3.2	Funções Primitivas dos Defensores	25
4.1	Simulação da estratégia com filtro preditor	34
4.2	Simulação da estratégia sem filtro preditor	35
5.1	Cronograma Mensal	37

Capítulo 1

Introdução

Neste Capítulo é apresentado um panorama geral do projeto, sobre o futebol de robôs, a categoria Small Size e a Equipe de Competições Robóticas da FEI (RoboFEI).

1.1 O Futebol de Robôs

O futebol de robôs é uma plataforma de pesquisa desenvolvida em diversas universidades do mundo. Tem como objetivo o desenvolvimento de robôs móveis autônomos, capazes de realizar tarefas sem a intervenção humana, entre elas, jogar futebol.

Existem diversas categorias de futebol de robôs, cada uma delas com regras próprias, como quantidade de jogadores, tamanho e tipo dos robôs. A principal comissão organizadora das competições é a [RoboCup, 2010], organização criada em 1993 com o objetivo de incentivar o desenvolvimento de novas tecnologias aplicadas a robótica.



Figura 1.1: Robôs da categoria Small-Size

O futebol é apenas uma forma de motivar e acelerar as pesquisas direcionando os pesquisadores a um objetivo comum. A principal meta da ROBOCUP é a de desenvolver até o ano de 2050 um time de futebol de robôs autônomo capaz de vencer a seleção de futebol campeã do mundo no ano.

1.2 A Categoria Small-Size

A categoria de futebol de robôs denominada Small-Size ou F180, como é conhecida, se depara com o problema da cooperação inteligente multi-agente e de controle em um ambiente altamente dinâmico, com um sistema híbrido centralizado / distribuído.

Em um jogo de futebol da categoria Small Size cada equipe compete com cinco robôs. Cada robô deve obedecer às dimensões especificadas nas regras da categoria F180 em que o robô deve caber dentro de um círculo de 180 milímetros de diâmetro e sua altura não deve ser superior a 15 centímetros, a menos que eles usem um sistema de visão embarcada. Os robôs jogam

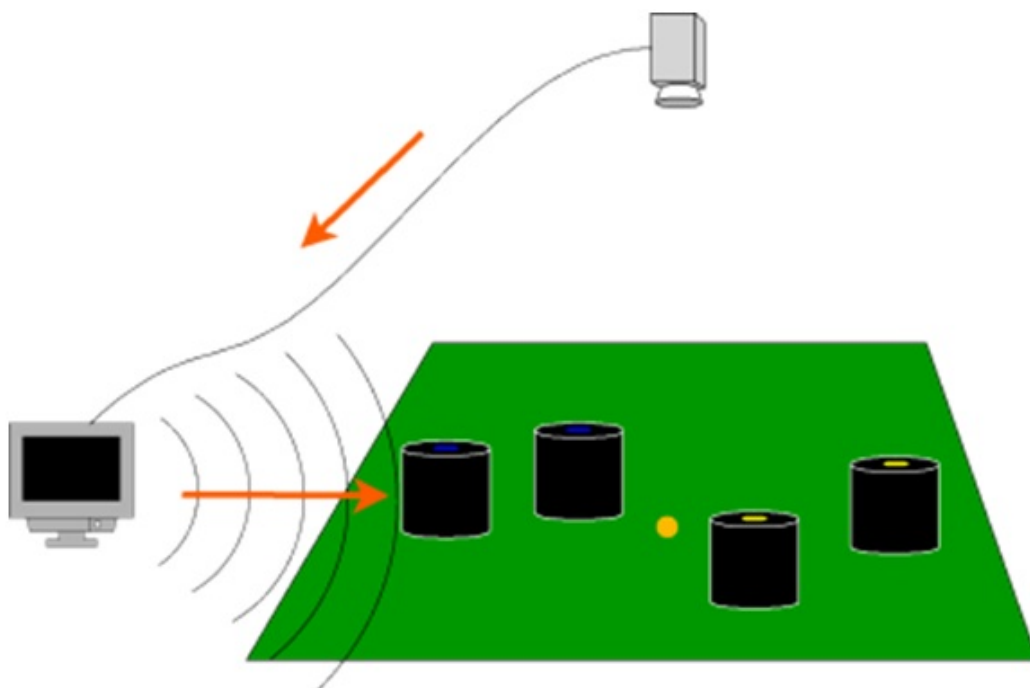


Figura 1.2: Sistema de Comunicação da categoria Small-Size

futebol com uma bola de golfe laranja sobre um tapete verde, o campo tem 6,00 m de comprimento por 4,00 m de largura.

Um PC fora do campo é utilizado para comandos de árbitro de comunicação e, no caso da visão global, a posição de informação para os robôs. Normalmente, cada equipe também possui um PC fora do campo que também executa a maioria, se não todos, os processamentos necessários para a coordenação e controle dos robôs. A comunicação entre os robôs é realizada sem fios, normalmente é utilizado um transmissor FM comercial e receptores dedicados a cada robô.

Construir uma equipe de sucesso requer uma concepção inteligente, implementação e integração de hardware e sub-componentes de software em

funcionamento.

1.3 O Futebol de Robôs na FEI

O projeto do Futebol de Robôs na FEI teve início em 2003 com a categoria Very-Small. Apenas em 2007 foi dado início ao desenvolvimento de um time para outra modalidade mais complexa, a [Small-Size, 2010] da ROBOCUP. Desde o início até os tempos atuais foram desenvolvidos além dos jogadores, Sistemas de Visão, Controle e Estratégia de alto nível, comparados aos dos melhores times.

A equipe [RoboFEI, 2010] possui um ótimo histórico de resultados como o campeonato na categoria Very-Small, nos anos de 2004 e 2006, o vice-campeonato na categoria Very-Small nos anos de 2003, 2007, 2008, e um terceiro lugar em 2005.

Já na categoria Small-Size, a equipe RoboFEI vem competindo na categoria desde 2008 e conquistando excelentes resultados nacionais e internacionais. Em 2010, consagrou-se campeã do Competição Latino Americana de Robótica (LARC), vice-campeão nacional da categoria em 2009. Estando entre as melhores equipes do país na categoria.

1.4 A Equipe RoboFEI

A atual equipe de futebol de robôs da FEI está sub dividida em cinco áreas de desenvolvimento. Apesar desta divisão, vale ressaltar que cada uma das áreas tem relação direta com as outras. Além disso, o trabalho em grupo e

a interdisciplinaridade são fatores determinantes da equipe, justamente por se tratar de robótica.

A Visão computacional é a área responsável pela captura e processamento das imagens do campo e o envio de informações relevantes para o sistema de estratégia. Um dos principais avanços da equipe nesse sistema foi o reconhecimento de objetos preciso e a rápida calibração.

Uma novidade da categoria na ROBOCUP 2010 em Cingapura, foi a utilização de um sistema de visão padrão denominado [SSL-Vision, 2010], em que um computador da organização é responsável pelo envio de todas as informações dos robôs e da bola para as duas equipes simultaneamente.

A estratégia é responsável por receber as informações enviadas pelo sistema de visão computacional e avaliar quais são as melhores jogadas a serem realizadas. Um sistema de identificação de trajetória está sendo implementado, onde são utilizadas as informações da visão e estratégia, para definir a trajetória que cada jogador deve ter em determinada situação, de modo a desviar de obstáculos e interceptar a bola com o menor caminho possível.

O sistema de controle, que com as informações da trajetória, determina como cada robô deve ser acionado, controlando sua velocidade linear e angular de modo que este siga a trajetória previamente definida.

A Mecânica é responsável pelo projeto e construção da estrutura de cada robô e seus componentes, conciliando o espaço e tecnologias necessárias de acordo com as restrições de tamanho do robô.

A Eletrônica é responsável pelo projeto e construção da interface eletrônica de cada robô, necessária para que este receba as informações do controle e possa executá-las dentro do jogo.

1.5 Objetivo

O principal objetivo deste trabalho é desenvolver um novo sistema de defesa que possa ser eficiente e capaz de detectar jogadas e passes até mesmo de equipes mais desenvolvidas na categoria. Através desse projeto, também será desenvolvido um sistema de previsão da bola utilizando algumas técnicas, dando base para um posicionamento eficiente dos jogadores. O sistema de previsão deve também prever quando uma equipe adversária utiliza o chute para cima ("chip-kicking"), prevendo assim o ponto em que a bola tocará o campo novamente.

Capítulo 2

Revisão Bibliográfica

Neste Capítulo, são abordados alguns métodos propostos para a resolução do problema, para isto, foi realizada uma ampla pesquisa entre os métodos propostos por outras equipes internacionais, levando em conta precisão do método e complexidade dos algoritmos. Para a realização desta pesquisa, foi levado em conta os Teams Description Papers que as equipes publicam como pré-requisito para a qualificação na competição internacional, a ROBOCUP.

2.1 Sistemas de Previsão

Sistemas de previsão de posições são amplamente estudados em computação, na área de inteligência artificial. Existem diversas técnicas de previsão que dependem principalmente do tipo de dado que o sistema possui, seu modelo e dinâmica. Na Figura 2.1, é apresentado o ciclo do software de estratégia e a comunicação entre o software e o robô, onde pode-se perceber que do momento de captura do frame até o momento em que o software se comunica

com o robô houve uma perda de tempo e de informações pois o robô já se encontra em outra posição.

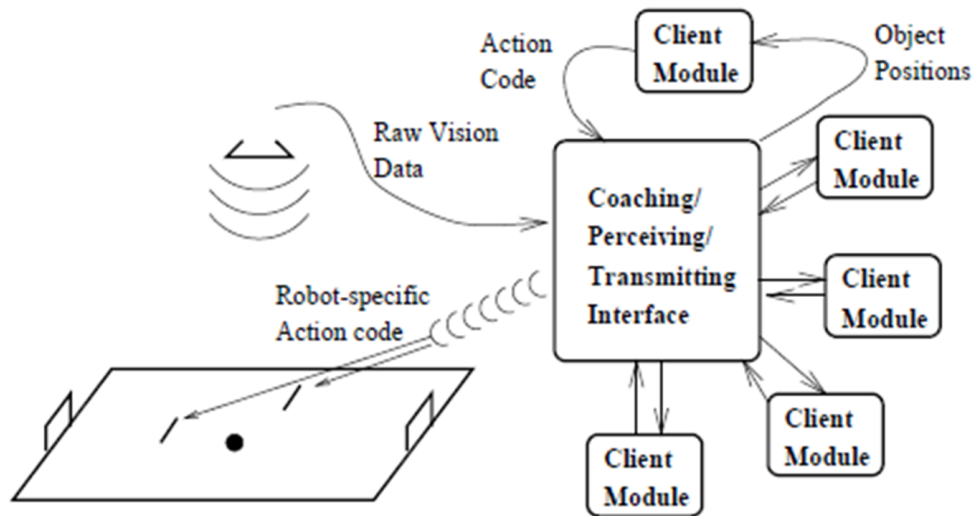


Figura 2.1: Ciclo de comunicação entre software e robô . Fonte: [Veloso, 1997]

Durante este trabalho, foram estudadas três métodos diferentes de se prever as posições futuras que a bola irá percorrer com uma boa precisão. Uma das técnicas é a técnica de previsão seguindo o modelo físico da bola. O segundo método é a Filtragem de Kalman, onde deve-se através das medidas de posição atual prevêr posições futuras baseando-se no movimento da bola e nas medidas anteriores. Já o terceiro método é o de Mínimos Quadrados.

2.1.1 Previsao Física

O sistema de previsão através do modelo físico da bola é bastante simples e foi utilizado pela equipe RoboFEI na ROBOCUP Open Brazil 2009, em Brasília, onde a equipe sagrou se vice-campeã na modalidade. Neste sistema, a estratégia armazenava sempre as duas ultimas posições em que a bola foi detectada pelo sistema de visão computacional, atribuindo a estas duas posições com os nomes de pos_{atual} e $pos_{anterior}$. Com estas duas informações, é calculada a velocidade da bola no eixo X e Y, através da fórmula descrita na Equação 2.1.

$$Vm = \Delta s / \Delta f \quad (2.1)$$

Onde: Δs é a variação de espaço entre dois frames e Δf é a variação de frames, que neste caso é um, levando em conta que são pegos sempre o frame atual e o anterior.

Com a velocidade e a posição anterior, podemos prever onde a bola estará, por exemplo, para prever onde a bola estará em 5 frames, utilizamos a Equação 2.2.

$$S_5 = pos_{atual} + (5 * Vm) \quad (2.2)$$

Onde: S_5 é a posição em que a bola estará contando 5 frames á frente, pos_{atual} é a posição em que a bola está atualmente e Vm é a velocidade média da bola.

Porém, apesar do sistema parecer simples e funcional, ele não leva em conta inúmeros erros que podem ocorrer durante os processos de captura

de imagem e nem sequer as perdas de velocidade da bola como o atrito da superfície. Outro grande problema é se o sistema de visão computacional em algum determinado frame não encontrar a bola e colocar como medida padrão $(0,0)$, ou então o sistema encontrar duas bolas e ter de decidir qual delas realmente é a bola. Se a bola escolhida for a errada, o sistema de previsão mostrará que o deslocamento da bola foi muito grande de um frame para outro.

Mesmo corrigindo alguns erros graves do sistema a precisão não seria garantida, pois só são levadas em conta duas medidas e não é possível verificar o modelo que a bola seguirá quando um jogador chutar a bola para cima, obviamente a trajetória da bola é diferente e com isso, deve-se ter um sistema de equações para descrever todos os modelos da bola. Devido a estes fatores, este tipo de previsão foi excluído do sistema de estratégia, pois a sua precisão é muito baixa e os jogadores ficariam perdidos em campo com medidas diferentes das reais.

2.1.2 Método dos Mínimos Quadrados (MMQ)

Uma técnica de otimização matemática bastante conhecida, é o Método dos Mínimos Quadrados. Esta técnica tem por principal objetivo, tentar encontrar o melhor ajuste para um conjunto de dados tentando reduzir a soma dos quadrados das diferenças entre a curva ajustada e os dados, frequentemente chamados na literatura de resíduos.

Segundo [Isaacson, 1994], para o Método dos Mínimos Quadrados funcionar de forma eficiente, um requisito implícito é que os erros em cada medida sejam distribuídos aleatoriamente com função densidade gaussiana, e que os

resíduos sejam independentes. Para garantir este requisito, utiliza-se o Teorema Gauss-Markov que garante (embora indiretamente) que o estimador de mínimos quadrados (EMQ) é o estimador não-enviesado de variância mínima linear na variável resposta.

A principal aplicação do Método dos Mínimos Quadrados é no ajuste de curvas. Suponha que o conjunto de dados consiste dos pontos (x_i, y_i) com $i = 1, 2, \dots, n$. Nós desejamos encontrar uma função f que descreva y aproximadamente como $f(x)$, ou seja, tal que $f(x) \approx y_i$.

Para se obter tal função, nós supomos que a função f é de uma forma particular contendo alguns parâmetros que necessitam ser determinados. Por exemplo, supor que ela é quadrática, significa que $f(x) = ax^2 + bx + c$, onde a , b e c não são conhecidos. Nós agora procuramos os valores de a , b e c que minimizam a soma dos quadrados dos resíduos:

$$S = \sum_{i=1}^n (y_i - f(x_i))^2 \quad (2.3)$$

No exemplo acima, $f(x) = ax^2 + bx + c$, f é linear nos parâmetros a , b e c . O problema simplifica consideravelmente neste caso e reduz-se essencialmente a um sistema de equações lineares (mínimos quadrados lineares).

Muitas equipes utilizam esta técnica porém com algoritmos computacionalmente eficientes como é no caso da Filtragem de Kalman. Utilizando-se apenas o Método dos Mínimos Quadrados, pode-se ajustar uma função à bola porém, não se tem a precisão exata de onde a bola estará em um instante de tempo futuro.

2.1.3 Filtragem de Kalman

Diversas equipes internacionais utilizam esta técnica para a implementação do sistema de previsão, principalmente devido a sua precisão e adaptação ao modelo físico da bola. Como é o caso da equipe CM-Dragons [Han, 1997], Skuba e ER-Fource que possuem sistemas bastante estáveis de previsão.

Segundo [Welch, 2006], teoricamente, a Filtragem de Kalman [Capparella,] [Mundhra, 2002] [Kalman, 1960], é um estimador para aquilo que é chamado de "Problema Gaussiano Linear Quadrático", que é o problema da estimação dos estados instantâneos de um sistema linear dinâmico perturbado por ruído Gaussiano branco usando-se medições linearmente relacionadas aos estados e também corrompidas por ruído branco.

O filtro de Kalman, na prática, é um conjunto de equações matemáticas que constitui um processo recursivo eficiente de estimação, uma vez que o erro quadrático é minimizado. Através da observação da variável denominada "variável de observação" outra variável, não observável, denominada "variável de estado" pode ser estimada de forma eficiente. Podem ser estimados os estados passados, o estado presente e mesmo previstos os estados futuros. O filtro de Kalman é um procedimento aplicável quando os modelos estão escritos sob a forma espaço-estado.

Quando utilizamos o filtro de Kalman para estimar o estado interno de um processo dada somente uma sequência de observações de ruído, é preciso modelar o processo de acordo com a estrutura do filtro de kalman. Para isto é necessário especificar as matrizes F_k , H_k , Q_k , R_k , e algumas vezes B_k para cada intervalo de tempo k . Dado a Equação 2.4.

$$x_k = F_k x_{k-1} + B_k u_k + w_k \quad (2.4)$$

Onde: F_k é a matriz de transição de estado, que é aplicado ao estado anterior x_{k-1} . B_k é a matriz de controle de entrada que é aplicado ao vetor de controle u_k . W_k é o ruído do processo, que corresponde a uma variável aleatória normal multivariada de média zero e matriz de variância Q_k .

$$W_k \approx N(0, Q_k) \quad (2.5)$$

No tempo k uma observação (ou medida) z_k do estado verdadeiro x_k é realizada de acordo com a Equação 2.6.

$$z_k = H_k x_k + v_k \quad (2.6)$$

Onde H_k é o modelo de observação que transforma o espaço de estados reais no espaço de observações e v_k é o ruído da observação, suposto um ruído branco gaussiano com média zero e matriz de variância R_k .

Supõe-se que o estado inicial e os vetores de ruído a cada passo $x_0, w_1, \dots, w_k, v_1 \dots v_k$ são estatisticamente independentes. Vários modelos dinâmicos reais não se encaixam exatamente neste modelo; porém, como o filtro de Kalman foi construído para funcionar na presença de ruído, um modelo aproximado costuma ser suficientemente bom para que o filtro seja útil.

A filtragem de Kalman é um técnica iterativa, onde a cada previsão realizada é gerada a medida prevista e sua correção. Além disso, sempre são injetados novos dados e o filtro é atualizado. No caso da bola isso é importante pois hora ela está em aceleração e hora em desaceleração. Logo com a

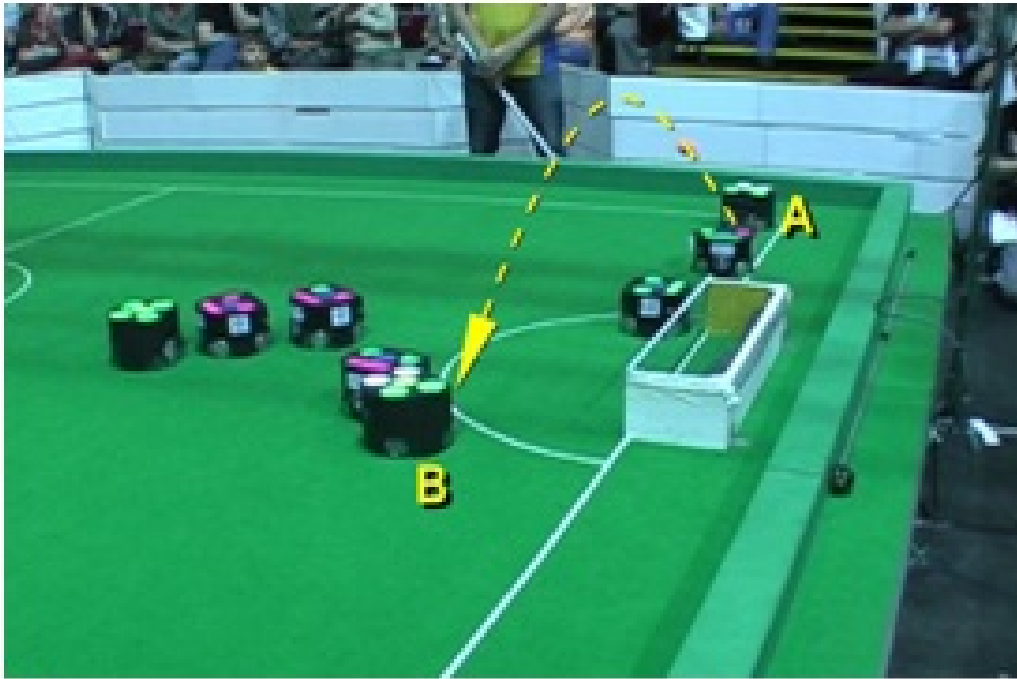


Figura 2.2: Jogada aérea do robô A para o robô B

filtragem de Kalman o filtro constantemente vai se adaptando ao modelo.

Com a filtragem de Kalman, segundo [Han, 1997], [M. Mehrandezh and Benhabib, 2000] e [D. Hujic, 1998] podemos descobrir inclusive características do controle do robô podendo até mesmo prever as futuras posições e os momentos em que o robô pode interceptar uma bola, por exemplo. Além disso, é possível prever jogadas aéreas das equipes e com isso conseguir se defender de maneira mais eficiente qualquer tipo de jogada, como é apresentado nas Figuras 2.2 e 2.3.

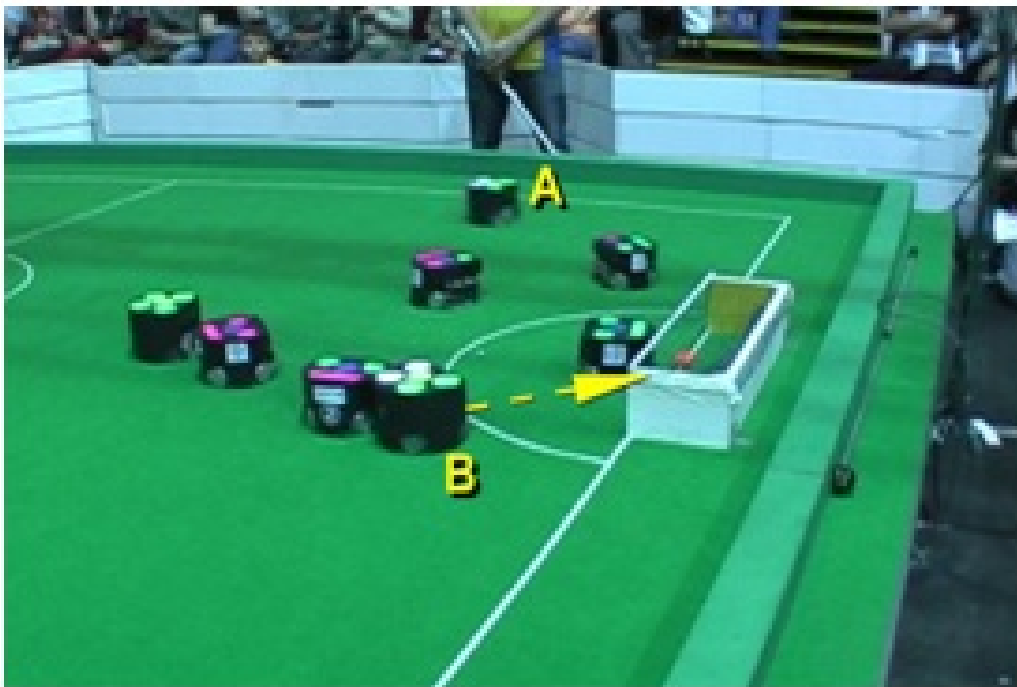


Figura 2.3: Finalização da jogada com o robô B

2.2 Sistema de Posicionamento Eficiente

O sistema de posicionamento eficiente desenvolvido é o fruto de uma extensa pesquisa entre os melhores métodos de defesa eficiente elaborados por diversas equipes da categoria. Dentre todas as equipes, a que mais se destaca internacionalmente é a [CM-Dragons, 2010], que descreve muito bem o seu sistema de estratégia segundo [Velo, 1997].

A idéia principal desta iniciação científica é comparar estatisticamente o novo sistema de posicionamento desenvolvido, em relação ao sistema anterior, principalmente na área defensiva e verificar qual dos sistemas é o que possuirá a defesa mais eficiente. Em um ambiente altamente dinâmico, é muito difícil posicionar os jogadores de maneira eficiente na defesa e ainda sim manter outras funções como o atacante ou o meio campista. Pode-se em um trabalho futuro também pensar no posicionamento eficiente de outras áreas como o ataque ou o meio de campo, levando em conta a mesma idéia da defesa.

Uma técnica bastante utilizada para tentar bloquear o máximo de espaços do gol, é calcular a quantidade de ângulos abertos em que o adversário pode fazer o gol, como é detalhado no Team Description Paper da equipe ER-Fouce, onde é apresentado a Figura 2.4, e tentar posicionar os robôs de forma a minimizar a quantidade de ângulos. Ou seja, quanto menores as aberturas de ângulos na defesa, menor será a probabilidade do adversário realizar o gol.

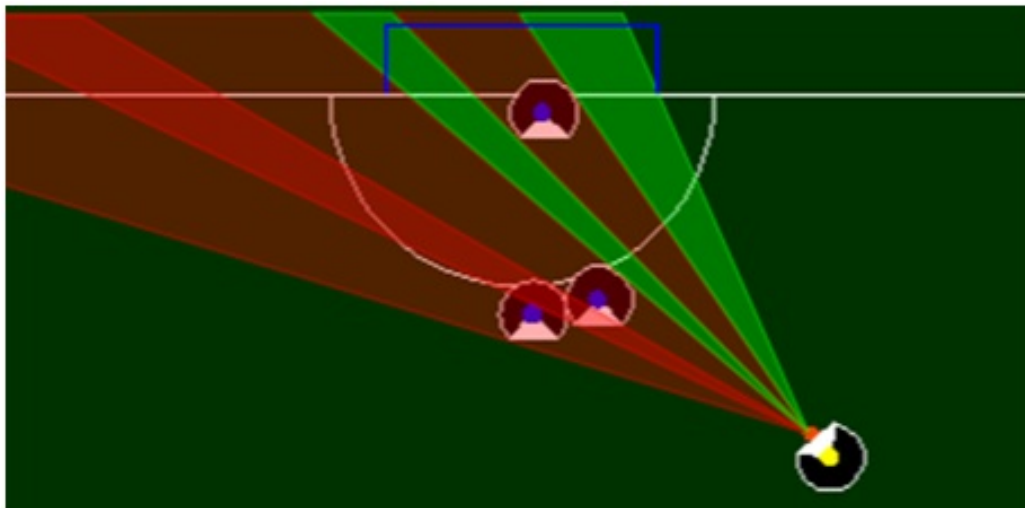


Figura 2.4: Probabilidade medida de o atacante acertar o gol: A largura dos corredores desprotegidos para realizar o gol é computada (verde) Fonte: [Blank, 2009]

2.2.1 Condições de defesa

Levando em consideração o jogo como um todo, decidiu-se construir quatro funções de condição de defesa. Para isso serão levados em conta com quem está a posse de bola e a quantidade de jogadores do time adversário no campo de ataque.

Capítulo 3

Metodologia

Neste Capítulo, serão abordados tópicos referentes a implementação do sistema proposto nesta iniciação científica, assim como as técnicas propostas.

3.1 Ferramentas Utilizadas

Nesta Seção, são descritas algumas das ferramentas utilizadas durante o processo de desenvolvimento desta iniciação científica. Para a escolha destas ferramentas levou-se em conta a complexidade de manipulação da ferramenta e sua eficiência em relação a aplicação proposta.

3.1.1 OpenCV

Para a implementação da Filtragem de Kalman, foram utilizadas diversas bibliotecas do Open Source Computer Vision Library [Bradsky, 2006] para facilitar a codificação. O OpenCV possui uma biblioteca multiplataforma desenvolvida pela Intel, em 2000, totalmente livre ao uso acadêmico e comercial,

para o desenvolvimento de aplicativos na área de Visão Computacional, bastando seguir o modelo de licença da BSD Intel. Ele também possui módulos de Processamento de Imagens e Video I/O, Estrutura de dados, Álgebra Linear, GUI (Interface Gráfica do Usuário) Básica com sistema de janelas independentes, Controle de mouse e teclado, além de mais de 350 algoritmos de Visão Computacional como: Filtros de imagem, calibração de câmera, reconhecimento de objetos, análise estrutural e outros. O seu processamento é em tempo real de imagens.

O OpenCV foi desenvolvido principalmente para a linguagem de programação C/C++, mas também é compatível com outras linguagens como é o caso do Python e do Visual Basic. A biblioteca possui atualmente duas versões, a 1.0 que foi lançada no final de 2006 e a 2.0, lançada em setembro de 2009. Ambas as versões são compatíveis com a aplicação desenvolvida.

Durante o processo de decisão sobre as ferramentas de implementação, levou-se em conta que todo o software de estratégia está implementado em linguagem C++, o que torna o OpenCV a ferramenta mais adequada para a implementação do filtro de Kalman, tendo em vista que o OpenCV já dispõe de bibliotecas adequadas para a implementação.

O OpenCV possui uma estrutura bem organizada, as bibliotecas existentes são detalhadas na Tabela 3.1 .

Tabela 3.1: Bibliotecas OpenCV

Biblioteca	Definição
CV	Agrega funcionalidades e algoritmos de Visão Computacional do OpenCV.
CVAUX	Agrega algoritmos de Visão, porém esta biblioteca possui os algoritmos que estão ainda em fase experimental.
CXCORE	Agrega Estrutura de Dados e Álgebra Linear.
HIGHGUI	Agrega o Controle de Interface e dispositivos de entrada.
ML	Agrega algoritmos de "Machine Learning" é um tipo de máquina de aprendizagem.
ED	Manual de estrutura de dados e operações.

3.1.2 Doxygen

É importante documentar o código desenvolvido para que futuramente o código possa sofrer modificações de forma eficiente, o Doxygen é um sistema que a partir dos comentários de um programa é possível gerar a documentação (em html por exemplo) do software. Para realizar a documentação, apenas devemos seguir algumas regras simples ao comentar o código e depois, basta o utilizador invocar o Doxygen para que seja possível gerar a documentação do programa.

Com isto, todos os integrantes da equipe podem entender o código implementado e possivelmente sugerir futuras implementações a fim de aprimorar ainda mais o sistema de estratégia já existente da equipe.

3.1.3 Simulador

Atualmente, o simulador utilizado pela equipe para a validação da estratégia é o SoccerBots, que é um software distribuído em conjunto com o TeambotsTM [TeamBots, 2010], que é composto de dois aplicativos Java e vários pacotes Java alvo de pesquisa em robótica móvel.

O SoccerBots é executado na aplicação TBSim usando um arquivo de configuração que personaliza o simulador para jogos de futebol. Outros tipos de robôs e tarefas também podem ser simulados em TBSim, incluindo as tecnologias nômades, um exemplo são os robôs Nomad 150 da Georgia Tech que utilizaram este simulador para competir no concurso de robôs móveis AAAI-97 [AAAI-97, 1997].

Este é um software de pesquisa, e foi desenvolvido em conjunto com diversos pesquisadores, através deste software diversas equipes testam sua estratégia e validam suas teorias. Na Figura 3.1, é apresentado a interface do software de simulação utilizado para este trabalho.

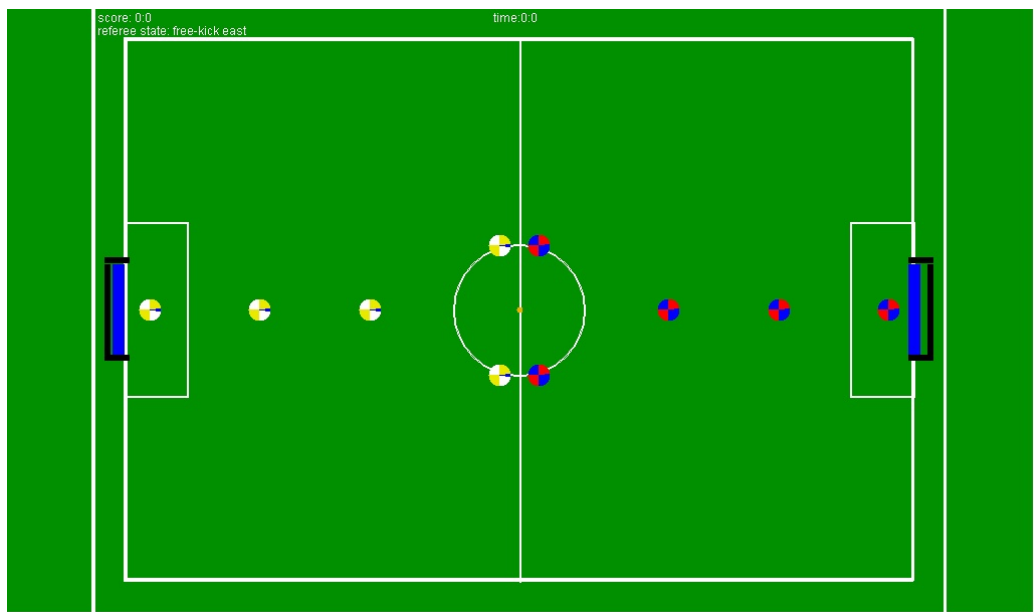


Figura 3.1: Simulador Soccerbots. Fonte: [TeamBots, 2010]

3.2 Implementação de Algoritmos

Nesta Seção, serão apresentados os algoritmos implementados, tanto os algoritmos de previsão, quanto as jogadas primitivas desenvolvidas com a finalidade de otimizar os jogadores e transformar a defesa de forma eficiente mais dinâmica e preparada para qualquer jogada do adversário.

3.2.1 Filtro de Kalman

O algoritmo para a previsão da bola, baseado em [Grewal and Andrews, 1993] e [Sorenson, 1985], foi implementado e testado antes de ser incorporado ao código da equipe. Esta é a principal ferramenta para que a equipe consiga

posicionar os robôs de forma eficiente. Com este algoritmo será possível que o jogador tenha informações mais precisas sobre onde a bola estará e onde são os possíveis pontos que ele pode interceptá-la, evitando falhas de posicionamento e perda da bola caso alguma pacote não chegue no tempo correto da câmera. O Pseudo-código da implementação do filtro de kalman é apresentado em Algorithm 1.

3.2.2 Funções primitivas construídas

Serão construídas três funções primitivas com o objetivo de aprimorar a defesa da equipe e os robôs envolvidos na jogada, conforme é apresentado na Tabela 3.2.

Através destas funções privivas, o módulo de previsão atuará com o objetivo de se antecipar as jogadas dos adversários levando em conta os possíveis ângulos livres entre a bola e o gol, como é apresentado na Figura 3.2, onde a função DeltaDefense é aplicada, segundo a abordagem de [Laue, 2009].

Algorithm 1 Algoritmo Filtro de Kalman

```

1: procedure  $[X] = \text{FUNCTION KALMAN}((duration, dt))$      $\triangleright$  duration =
   length of simulation (seconds)                         $\triangleright$  dt = step size (seconds)
2:    $measnoise \leftarrow 10$                                  $\triangleright$  position measurement noise (feet)
3:    $accelnoise \leftarrow 0.2$                              $\triangleright$  acceleration noise ( $feet/sec^2$ )
4:    $a \leftarrow [1dt; 0]$                                  $\triangleright$  transition matrix
5:    $b \leftarrow [dt^2/2; dt]$                              $\triangleright$  input matrix
6:    $c \leftarrow [10]$                                      $\triangleright$  measurement matrix
7:    $x \leftarrow [0; 0]$                                  $\triangleright$  initial state vector
8:    $xhat \leftarrow x$                                      $\triangleright$  initial state estimate
9:    $Sz \leftarrow measnoise^2$                              $\triangleright$  measurement error covariance
10:   $Sw \leftarrow accelnoise^2 * [dt^4/4dt^3/2; dt^3/2dt^2]$   $\triangleright$  process noise cov
11:   $P \leftarrow Sw$   $\triangleright$  initial estimation covariance  $\triangleright$  Initialize arrays for later
   plotting.
12:   $pos \leftarrow []$                                      $\triangleright$  true position array
13:   $poshat \leftarrow []$                                  $\triangleright$  estimated position array
14:   $posmeas \leftarrow []$                                $\triangleright$  measured position array
15:   $vel \leftarrow []$                                      $\triangleright$  true velocity array
16:   $velhat \leftarrow []$                                  $\triangleright$  estimated velocity array
17:  for all  $t$  do  $\triangleright$  Use a constant commanded acceleration of 1  $foot/sec^2$ 
18:     $u \leftarrow 1$                                      $\triangleright$  Simulate the linear system.
19:     $ProcessNoise \leftarrow accelnoise * [(dt^2/2) * randn; dt * randn]$ 
20:     $x \leftarrow a * x + b * u + ProcessNoise$              $\triangleright$  Simulate the noisy
   measurement
21:     $MeasNoise \leftarrow measnoise * randn$ 
22:     $y \leftarrow c * x + measnoise$                      $\triangleright$  Extrapolate the most recent state
   estimate to the present time.
23:     $xhat \leftarrow a * xhat + b * u$                      $\triangleright$  Form the Innovation vector.
24:     $Inn \leftarrow y - c * xhat$   $\triangleright$  Compute the covariance of the Innovation.
25:     $s \leftarrow c * P * c' + Sz$                          $\triangleright$  Form the Kalman Gain matrix
26:     $K \leftarrow a * P * c' * inv(s)$                      $\triangleright$  Update the state estimate.
27:     $xhat \leftarrow xhat + K * Inn$                          $\triangleright$  Compute the covariance of the
   estimation error
28:     $P \leftarrow a * P * a' - a * P * c' * inv(s) * c * P * a' + Sw$   $\triangleright$  Save some
   parameters for plotting later.
29:     $pos \leftarrow [pos; x(1)]$ 
30:     $posmeas \leftarrow [posmeas; y]$ 
31:     $poshat \leftarrow [poshat; xhat(1)]$ 
32:     $vel \leftarrow [vel; x(2)]$ 
33:     $velhat \leftarrow [velhat; xhat(2)]$ 
34:  end for
35: end procedure

```

Tabela 3.2: Funções Primitivas dos Defensores

Nome da Primitiva	Definição
Defense()	Esta função faz com que um defensor se posicione de forma a bloquear o gol complementando os ângulos que o goleiro não está cobrindo. Ela será utilizada quando a posse de bola não estiver com o time adversário e a bola esteja no campo de ataque, com isso, pretende-se aproveitar melhor os jogadores na área ofensiva para que seja possível realizar jogadas ensaiadas.
DeltaDefense()	Esta função forma uma linha defensiva na frente do goleiro, para ajudar a proteger o gol. A forma geométrica dos defensores e do goleiro é a de um triângulo e por isso é chamada de DeltaDefense. Além disso, os ângulos internos do triângulo mudam de acordo com a posição da bola fazendo com que em qualquer posição do campo o triângulo feche a maior quantidade de ângulos possíveis. experimental.
Block()	Posiciona o robô entre o adversário com a bola e o gol, impedindo que este consiga chutar. Se o adversário não estiver com a bola, se posiciona à frente do adversário, impedindo que este receba a bola. O posicionamento exato leva em conta a região do campo e disposição dos adversários e companheiros.

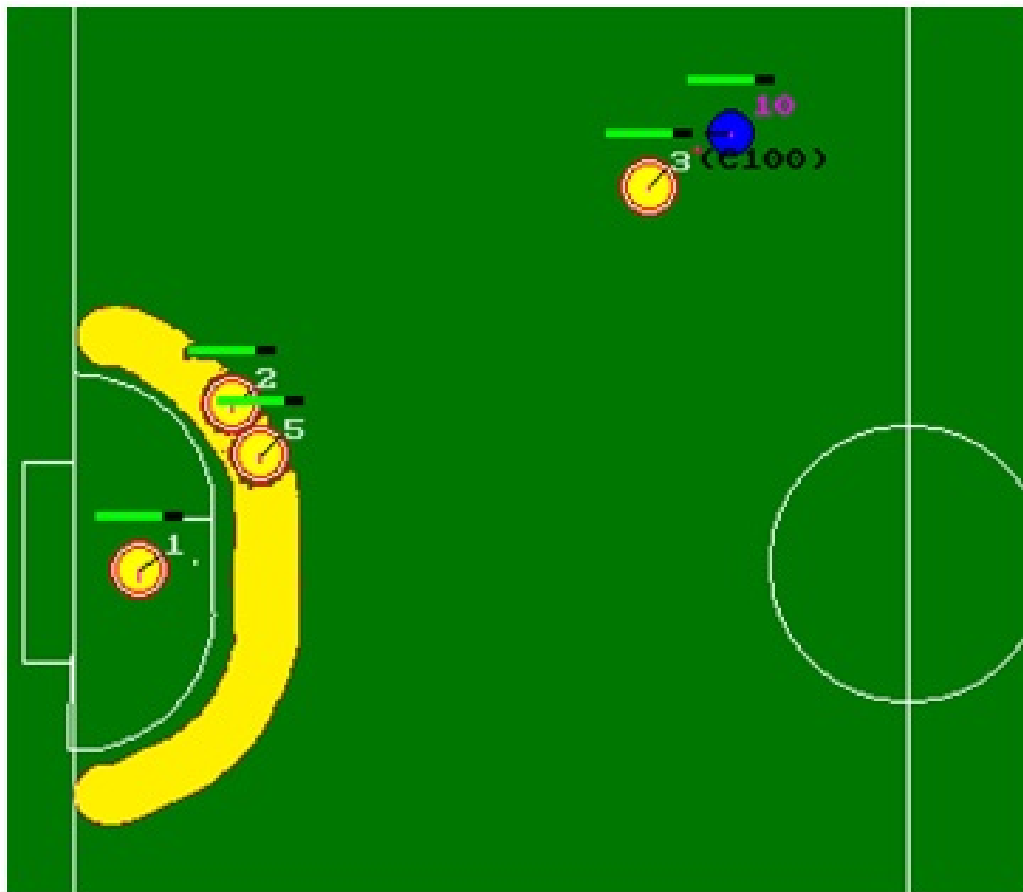


Figura 3.2: Funções DeltaDefence e Block habilitadas. Fonte: [Laue, 2009]

Capítulo 4

Resultados Finais

Para realizar a implementação do filtro de Kalman e verificação se o filtro realmente estava se adequando ao modelo, primeiramente foram obtidas medidas sem nenhum tipo de ruído, através dos log's obtidos da RoboCup 2009, logo depois, foram obtidas medidas com ruídos para analisar o comportamento do filtro. Com isso, obteve-se em todas as amostras uma boa adaptação do filtro e uma precisa previsão de seis frames a frente do frame atual. Na Imagem 4.1, segue a implementação do filtro de kalman em MATLAB.

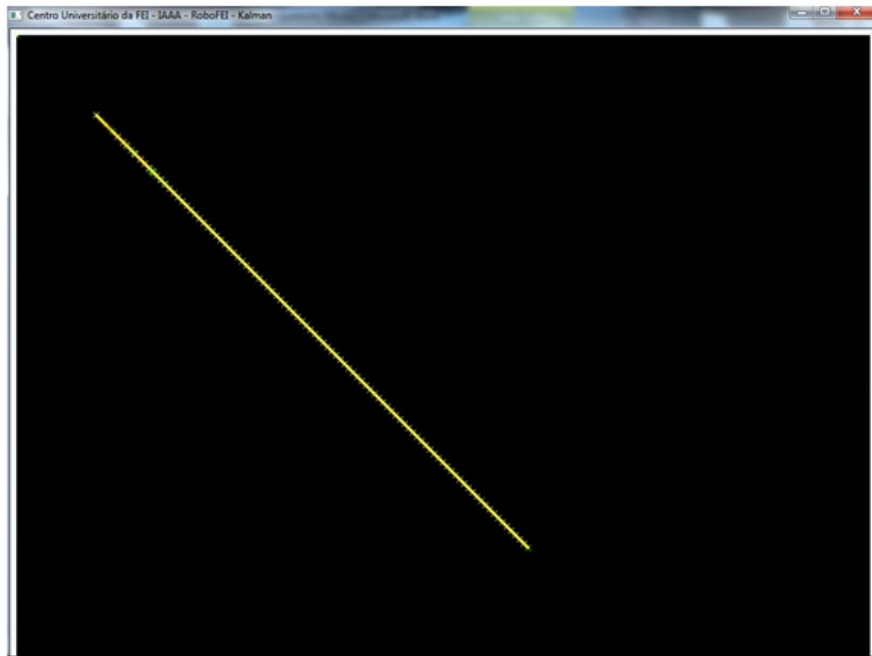


Figura 4.1: Tela do protótipo implementado para a filtragem de kalman

Fonte: Autor.

Porém, foi verificado que o filtro de kalman não é muito eficiente em casos particulares, quando ocorre uma brusca mudança de velocidade e de trajetória da bola, por exemplo, quando a bola bate em um robô adversário ou em um obstáculo estático. Segundo Veloso [Han, 1997], nestes casos é necessário a modelagem física dessas condições. A idéia é aumentar a variância do filtro de kalman quando a bola estiver próxima de algum robô ou de um obstáculo estático, dando pouca confiança a medida e realizar os cálculos físicos necessários para saber onde aproximadamente estará a bola. Com isso, mesmo com pouca precisão sabemos onde a bola estará, e logo quando a bola se distancia dos robos e volta a satisfazer o filtro de kalman a variância

é reduzida fazendo com que o filtro de kalman volte a ser eficiente. Alguns filtros podem ser mais eficientes nestes casos como o H_∞ , porém a implementação deste filtro é mais complexa pois ainda não existem bibliotecas no OpenCV para a utilização deste tipo de filtragem. Existem também técnicas que melhoram a performance do filtro de kalman como a técnica de bootstrap [20] que pode ser utilizado para aprimorar a precisão da previsão estimando os possíveis erros da filtragem de kalman.

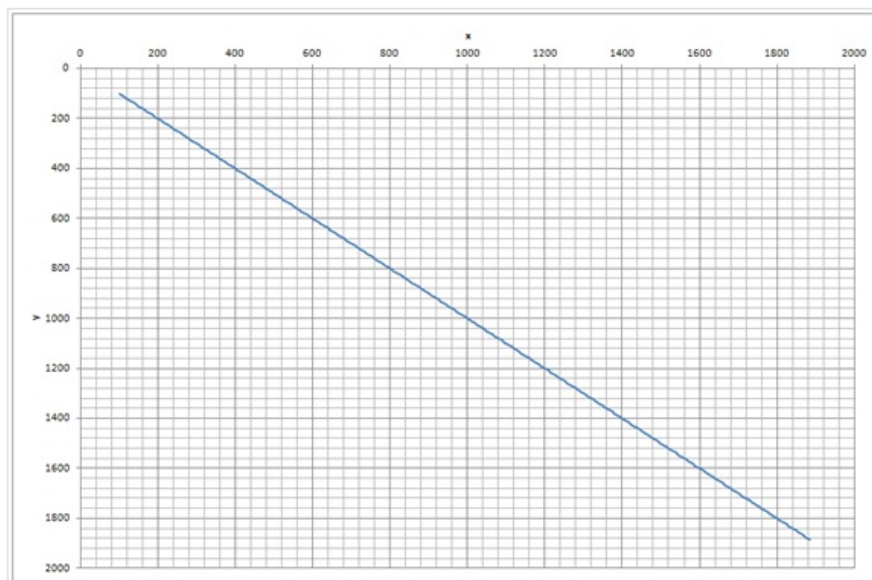


Figura 4.2: Gráfico de movimento da bola X versus Y Fonte: Autor.

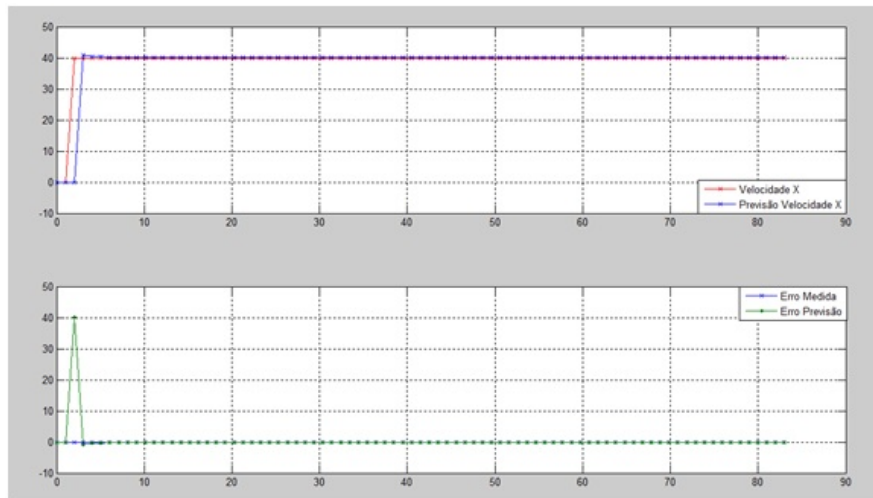


Figura 4.3: Gráficos de verificação de erro na previsão de posições Fonte: Autor.

Com base na análise dos gráficos da Figura 4.3 e 4.2, é possível verificar que a filtragem de kalman só comete um erro consideravelmente grande quando existe uma alteração de movimento, ou seja, quando a bola modifica muito rapidamente sua velocidade. Porém, em pouco mais de dois frames a previsão volta a ficar dentro do modelo e adaptar-se ao movimento, vale lembrar que estas medidas são para a coordenada X da bola, porém temos em vista que o filtro se comporta de forma semelhante também em Y.

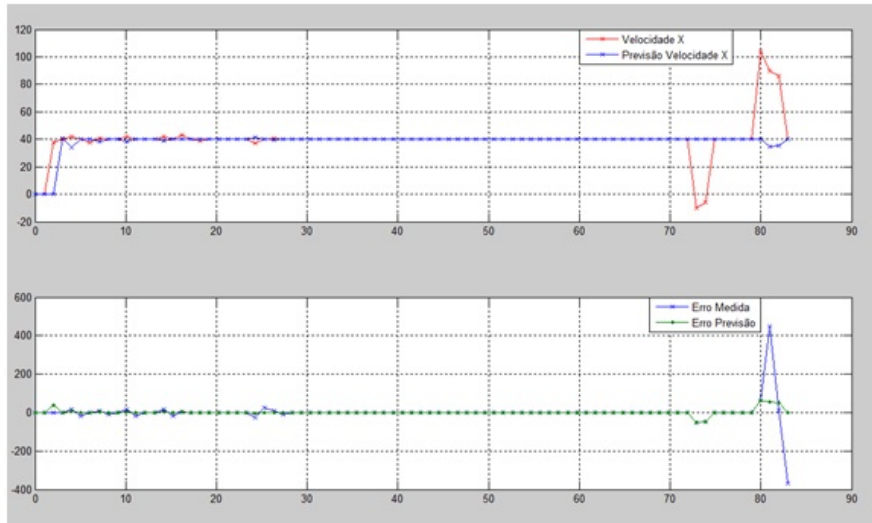


Figura 4.4: Gráficos de verificação de erro na previsão de posições com erros injetados Fonte: Autor.

No gráfico da Figura 4.4, são utilizadas as mesmas medidas do gráfico da Figura 4.3, porém, foram injetadas algumas medidas erradas para verificar a adaptação do filtro ao modelo. Sempre é necessário ter em mente que medidas muito distantes em um curto intervalo de tempo podem ser provavelmente erros e o filtro deve estar preparado para que os erros sejam descartados. Como é possível verificar acima, mesmo com erros injetados o filtro se comporta de forma a ponderar possíveis medidas incorretas.

Partindo dos protótipos, realizou-se a implementação final do filtro de kalman no software de estratégia da equipe e na Seção 4.1 realizou-se uma análise comparativa entre as técnicas utilizadas atualmente e anteriormente a fim de verificar possíveis melhorias no sistema. Na Figura 4.5, pode ser observado que o robô 3 tenta interceptar a bola através do filtro de kalman e a reta amarela partindo do robô é a trajetória que deve-se percorrer para

conseguir interceptar a bola de forma eficiente, ou seja, o local em que será possível o robô 3 pegar a bola levando-se em conta a velocidade da bola e do robô.



Figura 4.5: Software de estratégia Fonte: Autor.

4.1 Análise Comparativa das Técnicas

Nesta Seção, é realizada uma análise comparativa de desempenho entre a implementação de código do filtro de kalman com a implementação sem o filtro de kalman, apenas com o sistema de previsão física descrito na seção 2.1.1, com a finalidade de verificar a importância da implementação no jogo. Na Figura 4.6, pode-se observar que realmente a filtro preditivo auxilia no posicionamento da defesa. Para a obtenção destes dados utilizou-se o simulador citado na Seção 3.1.3.

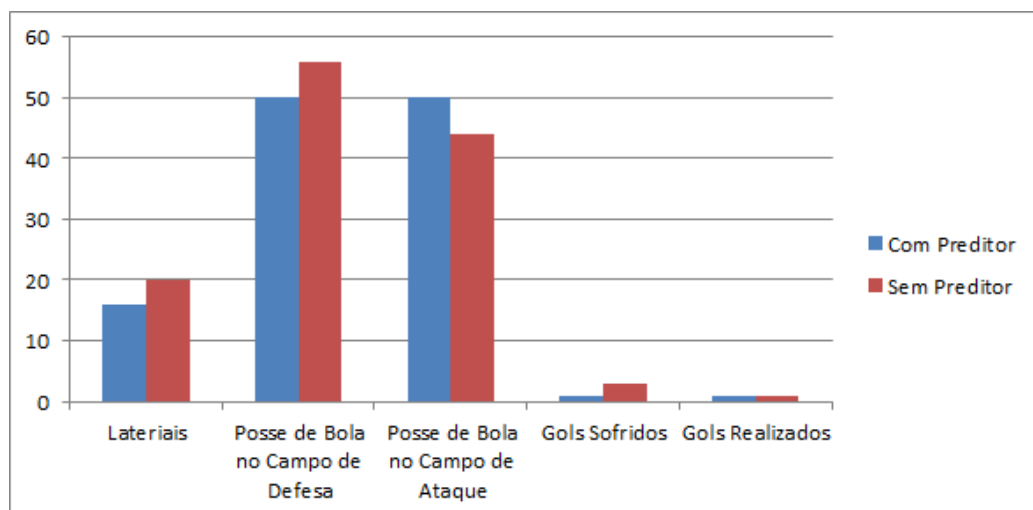


Figura 4.6: Análise comparativa entre as técnicas de previsão Fonte: Autor.

Para a extração dos dados, foram simulados 25 jogos com 4 minutos de duração cada, através dos logs gravados, foram contabilizadas a quantidade de laterais durante o jogo, a porcentagem de posse de bola no campo do ataque e da defesa, gols sofridos e gols realizados. A Tabela 4.1, representa os dados brutos dos jogos realizados com o software aplicando o preditor, já na Tabela 4.2, são apresentados os dados brutos dos jogos realizados sem o filtro de kalman.

Tabela 4.1: Simulação da estratégia com filtro preditor

Jogo	Laterais	Bolas no campo da defesa	Bolas no campo de ataque	gols sofridos	gols realizados
1	6	40	60	0	1
2	5	34	66	0	0
3	44	56	22	0	2
4	45	55	55	1	0
5	34	66	66	0	1
6	87	13	13	2	2
7	17	67	33	0	0
8	18	56	44	3	1
9	36	62	38	0	1
10	22	55	45	1	2
11	16	37	63	2	0
12	17	34	66	0	1
13	20	56	44	1	0
14	14	19	81	0	1
15	20	56	44	1	0
16	10	64	36	0	1
17	32	65	35	1	1
18	20	47	53	1	0
19	22	50	50	1	2
20	12	53	47	2	0
21	16	56	44	3	1
22	14	45	55	2	1
23	15	50	50	0	0
24	15	89	11	0	2
25	18	45	55	2	2

Tabela 4.2: Simulação da estratégia sem filtro preditor

Jogo	Laterais	Bolas no campo de defesa	Bolas no campo de ataque	gols sofridos	gols realizados
1	11	60	40	1	1
2	7	55	45	3	0
3	16	78	22	4	1
4	22	45	55	4	1
5	18	34	66	5	1
6	20	87	13	2	2
7	14	67	33	5	0
8	18	56	44	3	1
9	36	62	38	2	4
10	23	55	45	3	2
11	20	67	33	2	0
12	17	34	66	6	4
13	20	56	44	1	0
14	19	19	81	8	3
15	20	56	44	1	0
16	12	78	22	3	1
17	32	65	35	3	5
18	20	47	53	6	0
19	23	50	50	1	2
20	12	53	47	2	0
21	20	56	44	4	1
22	17	68	32	2	3
23	15	71	29	4	0
24	20	89	11	0	2
25	27	68	32	3	1

Capítulo 5

Cronograma

5.1 Desenvolvimento do projeto

Alguns dos principais algoritmos da pesquisa já estão implementados e em fase de correção, um trabalho estatístico ainda será realizado, comparando a eficiência dos atuais algoritmos em relação aos algoritmos já implementados no código.

5.2 Cronograma mensal

Tabela 5.1: Cronograma Mensal

Mês	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Atividades															
Revisão Bibliográfica	X	X	X	X	X	X	X								
Análise dos Sistemas Existentes	X	X													
Análise Estatística			X	X	X	X									
Definição dos Métodos a serem utilizados			X	X	X	X									
Implementação do Algoritmo					X	X	X	X	X						
Elaboração do Relatório Parcial					X	X									
Testes e Análises dos Resultados								X	X	X	X	X			
Mudanças Finais											X	X	X		
Competição Latino Americana													X	X	
Elaboração do Relatório Final												X	X	X	X

5.3 A ROBOCUP 2010

Para a ROBOCUP 2010 em Cingapura, a Equipe RoboFEI, propôs uma série de reformulações, nas áreas da mecânica, elétrica e de estratégia, porém com o atraso no cronograma de algumas atividades a equipe não conseguiu estreitar os novos robôs em Cingapura e jogou com os robôs sem reformulações elétricas ou mecânica, o que inviabilizou alguns testes que seriam realizados na RoboCup pois os robos já estavam muito desgastados de competições anteriores.

Na área da estratégia, algumas funções já haviam sido implementadas como as primitivas e testadas porém não havia parâmetros para comparar a performance das primitivas atuais com as anteriores, justamente por causa dos robôs estarem desgastados. Com isso, todas as modificações implementadas até agora foram validadas apenas com o simulador.

5.4 Panorama para a ROBOCUP 2011

Uma das principais metas estabelecidas pela equipe de futebol de robôs categoria Small-Size é a de melhorar este ano as marcas alcançadas durante os anos anteriores. Na área de estratégia, o time estará bem mais dinâmico e com uma defesa mais sólida. O sistema de previsão da bola fará com que a equipe possa se preparar para as jogadas dos times adversários, além de montar jogadas de contra-ataque rápidas.

Capítulo 6

Trabalhos Futuros

A partir deste trabalho realizado, pretende-se aprimorar ainda mais a área de previsão dos robôs realizando testes comparativos entre a Filtragem de Kalman, implementada neste trabalho, com filtros de partículas, como por exemplo o filtro H_∞ , uma abordagem detalhada sobre este tipo de filtro pode ser encontrada em [Shen and Deng, 1997] e [Simon and El-Sherief, 1996].

Além disso, recomenda-se a implementação de algoritmos para a tomada de decisões como os algoritmos que partem do princípio de pareto [Kyrylov, 2007], existem alguns trabalhos importantes na literatura que mostram resultados animadores na implementação desta técnica.

Bibliografia

- [AAAI-97, 1997] AAAI-97 (1997). The fourteenth national conference on artificial intelligence (aaai-97). Website. <http://www.aaai.org/Conferences/AAAI/aaai97.php>.
- [Blank, 2009] Blank, P. (2009). Er-force team description paper for robocup 2009. *Chair of Pattern Recognition, Department of Computer Science, University of Erlangen-Nuremberg, Germany*.
- [Bradsky, 2006] Bradsky, G. R.; Pisarevsky, V. B. J. (2006). *Learning OpenCV: Computer Vision with the OpenCV Library*.
- [Capparella,] Capparella, F. Visual servoing of a wheeled mobile robot for inter-cepting a moving object. *Dipartimento di Informatica e Sistemistica, Universit'a di Roma "La Sapienza, Roma, Italy*.
- [CM-Dragons, 2010] CM-Dragons (2010). Equipe small size cm-dragons. Website. <http://www.cs.cmu.edu/~robosoccer/small/index.html>.
- [D. Hujic, 1998] D. Hujic, E. A. Croft, G. Z. R. F. J. K. M. a. B. B. (1998). The robotic interception of moving objects in industrial settings: strategy development and experiment. *IEEE Trans. on Mechatronics*, 3:225–239.
- [Grewal and Andrews, 1993] Grewal, M. and Andrews, A. (1993). *Kalman Filtering Theory and Practice*. 1092-1095. Englewood cliffs edition.
- [Han, 1997] Han, Kwun; Veloso, M. (1997). Physical model based multi-objects tracking and prediction in robosoccer. *In Working notes of the AAAI 1997 Fall Symposium on Model-directed Autonomous Systems, MIT, Boston*.
- [Isaacson, 1994] Isaacson, E., K. H. (1994). Analysis of numerical methods. *Dover*.

- [Kalman, 1960] Kalman, R. E. (1960). A new approach to linear filtering and prediction problems. *Transaction of the ASME-Journal of Basic Engineering*, pages 35–45.
- [Kyrylov, 2007] Kyrylov, M. (2007). Pareto-optimal offensive player positioning in simulated soccer. *LNAI 5001 - RoboCup 2007: Robot Soccer World Cup XI*.
- [Laue, 2009] Laue, T. (2009). B-smart(bremen small multi agent robot team)extended team description for robocup 2009. *Universitaet Bremen, Postfach, Bremen, Germany*.
- [M. Mehrandezh and Benhabib, 2000] M. Mehrandezh, N. M. Sela, R. G. F. and Benhabib, B. (2000). Robotic interception of moving objects using an augmented ideal proportional navigation guidance technique. *IEEE Trans. on Systems, Man and Cybernetics*, 30(A):238–250.
- [Mundhra, 2002] Mundhra, K. (2002). Intercepting a falling object: Digital video robot. *Me-chanical and Aerospace Engineering, Arizona State University*.
- [RoboCup, 2010] RoboCup (2010). Organização internacional de competições robóticas. Website. <http://www.robocup.org>.
- [RoboFEI, 2010] RoboFEI (2010). Equipe de competições robóticas da fei. Website. <http://www.fei.edu.br/robo>.
- [Shen and Deng, 1997] Shen, W. and Deng, L. (1997). *Game Theory Approach to Discrete H_∞ Filter Design*. 1092-1095. April edition.
- [Simon and El-Sherief, 1996] Simon, D. and El-Sherief, H. (1996). *Hybrid Kalman/Minimax Filtering in Phase-Locked Loops*. 615-623.
- [Small-Size, 2010] Small-Size (2010). Oficial robocup small size league. Website. <http://small-size.informatik.uni-bremen.de/>.
- [Sorenson, 1985] Sorenson, H. (1985). *Kalman Filtering: Theory and Application*. 1092-1095. Los alamos edition.
- [SSL-Vision, 2010] SSL-Vision (2010). Small size league vision. Website. <http://small-size.informatik.uni-bremen.de/sslvision>.

-
- [TeamBots, 2010] TeamBots (2010). Software para simulação de um time de futebol de robôs. Website. <http://www-2.cs.cmu.edu/~trb/TeamBots/Domains/SoccerBots/>.
- [Veloso, 1997] Veloso, M. (1997). Cmunitied: A team of robotic soccer agents collaborating in an adversarial environment. *Carnegie Mellon University*.
- [Welch, 2006] Welch, Greg; Bishop, G. (2006). An introduction to the kalman filter. *Department of Computer Science, University of North Carolina at Chapel Hill*.