

UNIVERSIDADE FEDERAL DO RIO GRANDE DO NORTE
CENTRO DE TECNOLOGIA
ENGENHARIA DE COMPUTAÇÃO

ESTRUTURAÇÃO E DESENVOLVIMENTO DE ESTRATÉGIA PARA FUTEBOL DE ROBÔS

Luiz Henrique R. da Silva

Orientador: Adelardo A. D. De Medeiros

Trabalho de conclusão de curso
apresentado no curso de
Engenharia de Computação como
requisito parcial para sua
conclusão.
Área de concentração: Engenharia
de Computação.

Natal/RN, julho de 2010

*À Deus e minha família pelo
inestimável apoio e paciência
durante estes anos*

Agradecimentos

Valorizo e agradeço a paciência e o apoio dado pelo meu Orientador, professor Adelardo.

Ao mestrando Ellon pela paciência.

Ao companheiro de luta João Paulo.

Aos colegas de turmas e de laboratório.

Resumo

Este trabalho visa apresentar um comparativo entre formas de implementação da etapa de estratégia de um sistema robótico aplicado ao futebol de robôs. Apresenta sistemas de outras comunidades científicas com diferentes soluções para a parte estratégica/tática do futebol. Esta é a principal parte em software relacionada com a aplicação. Conseguir prever o comportamento da bola, analisar melhores trajetórias até a bola, realizar movimentos eficientes, até conseguir levar a bola ao gol adversário, são tarefas próprias da estratégia. Estas tarefas serão estudadas neste trabalho.

Palavras-chave: Sistema robótico, lógica booleana, estratégia, máquina de decisão, tabela.

Sumário

Sumário	i
Lista de Figuras	iii
Lista de Tabelas	v
Lista de Símbolos e Abreviaturas	vii
1 INTRODUÇÃO	1
1.1 Início	1
1.2 Uma classificação	1
2 SISTEMA ROBÓTICO	3
2.1 Funcionamento	3
2.2 Componentes do sistema	4
3 TRABALHOS RELACIONADOS	6
3.1 Competições	6
3.2 Alguns trabalhos similares	6
4 ESTRATÉGIA	9
4.1 A estratégia booleana	9
4.2 A implementação	9
4.3 O zagueiro	10
4.4 O atacante	13
4.5 O goleiro	15
4.6 Os predicados	15
5 ESTRUTURAS	18
6 RESULTADOS E PERSPECTIVAS.....	20
6.1 Experimentos e resultados.....	20
6.2 Propósito da lógica “fuzzy”	20
7 CONCLUSÕES	22
Referências bibliográficas	23

Lista de Figuras

1 Robô universal de rosum.....	1
2 Exemplo de arquitetura de braço robótico	2
3 Exemplo de braço manipulador	2
4 Robô Snookie guiado por sensor “termistor”.....	2
5 Imagem do campo com robôs	3
6 Sistema de coordenadas de mundo.....	5
7 Mirosot	6
8 Estrutura da estratégia de SOTY5	6
9 Formações. No esquerdo simples e lado direito defensiva	7
10 Small Size Robot League	8
11 Estratégia Individual	9
12 Função de contorno, sobreposição de áreas.....	10
13 Situação de gol contra	11
14 Função de sair da área	13
15 Posição desejada para chute	14
16 Etapas de processamento	18
17 Visão geral da etapa de estratégia	19
18 Escopo da estratégia	19
19 Programa simulador, importante no desenvolvimento	20

Lista de Tabelas

1 Funcionamento do goleiro	16
2 Funcionamento do atacante	17
3 Funcionamento do zagueiro	17

Lista de Siglas e Abreviaturas

USB	<i>Universal Serial Bus</i>
XML	<i>eXtensible Markup Language</i>
FIRA	<i>Federation International of Robot-soccer Association</i>
PI	Proporcional-Integrativo (conceito de sistemas de controle)

1. Introdução

1.1 Início

A robótica é uma área de recente desenvolvimento em relação às outras de áreas de ciência. Os primeiros conceitos da robótica foram criados não por cientistas mas por um escritor chamado Karel Čapek que fez a peça “Rossum's Universal Robots”. Foi imaginado todo funcionamento da criatura (figura 1).

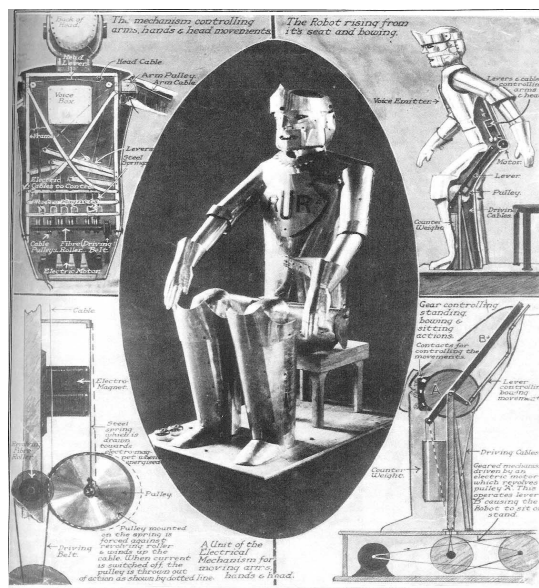


Figura 1: Robô universal de Rossum

Em 1940 o escritor Isaac Asimov aproximou a criatura fictícia do mundo real. Ele criou o termo robótica e as leis que devem reger o comportamento do robô. Podemos associar estes conceitos com a revolução industrial ocorrida há algumas décadas atrás. Nesta época diversas máquinas começaram realizar trabalhos árduos para o ser humano.

1.2 Uma classificação

A robótica é uma área que agrega os conhecimentos científicos de diversas áreas como física, mecânica, computação para desenvolvimento de robôs. Por aplicar conhecimentos científicos para auxiliar a sociedade em suas atividades, se caracteriza como área própria da engenharia.

As principais aplicações da robótica hoje, ainda são encontradas nas indústrias. Os robôs manipuladores são os mais comuns (figura 3 e figura 4). Estes robôs geralmente são fixos, possuindo braço(s) para realizar determinada tarefa. São constituídos de juntas e elos que permitem a movimentação da **mão** do robô. A mão do robô é o lugar onde é colada a ferramenta referente à aplicação do robô. Neste ponto é onde se deseja maior liberdade de movimentação para desvio de possíveis obstáculos ou para um maior espaço de movimentação.

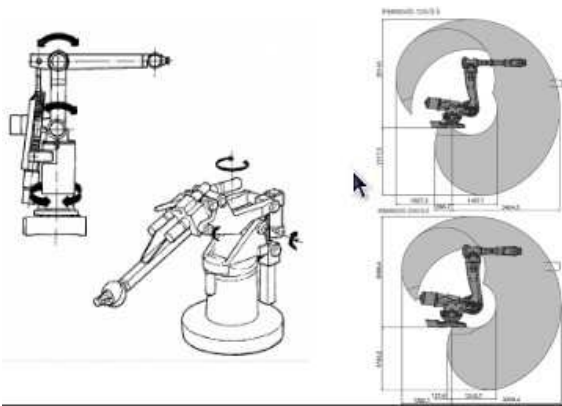


Figura 3: Exemplo de braço manipulador

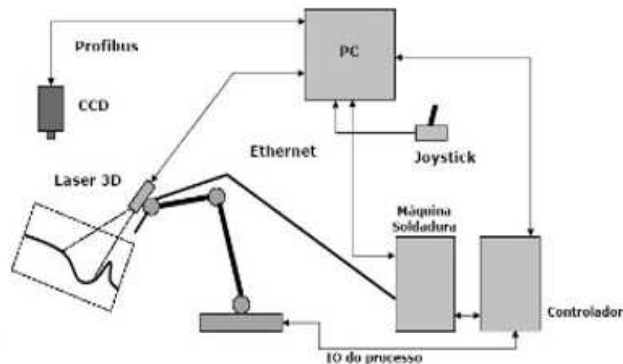


Figura 2: Exemplo de arquitetura de braço robótico

Robôs locomotores possuem também grande generalidade de aplicação. Uma possível aplicação pode ser cortar grama, aspirar poeira de residência. Outra tarefa bem indicada para robôs é exploração de ambientes insalubres para o homem. Estes robôs para cumprirem seus objetivos possuem um meio de locomoção que geralmente caracteriza-os como terrestres, aéreos, ou aquáticos. Nos robôs terrestres existem duas formas principais de locomoção, através de rodas ou de pernas/patas. Robôs aéreos geralmente utilizam hélices, como os aquáticos. Isto não impede que robôs que se movam por dois (ou mais) diferentes meios. A figura 4 mostra um robô de exploração submarina.



Figura 4: Robô de Snookie guiado por sensor “termistor”

De modo geral podemos classificar como robô uma máquina que possui como elementos: sensor, processador e atuadores, logicamente direcionados para a aplicação da máquina.

O sistema em estudo tem como aplicação reproduzir o esporte futebol, mas com os “robôs” sendo os jogadores. Para isso a movimentação de “robôs” é fundamental. Assim os “robôs” são do grupo locomotores. O sistema como todo pode ser classificado como um robô por possuir sensor, unidade processamento da situação e interagir com o ambiente através de atuadores, será referido com o termo “sistema robótico”. Os jogadores que são na realidade os atuadores do sistema, serão referidos como “robô” ou mesmo jogador.

No segundo capítulo será apresentado o sistema em estudo, o terceiro capítulo apresenta alguns sistemas robóticos semelhantes. O quarto e quinto capítulo apresentam a estratégia e estrutura de implementação. O sexto capítulo apresenta o resultados. O trabalho é finalizado com a conclusão.

2. Sistema robótico

2.1 Funcionamento

O sistema descrito aqui tem como aplicação o futebol de robô, está presente no Laboratório de Robótica - DCA. Para que a aplicação tenha um funcionamento satisfatório e coerente, diversos equipamentos e dispositivos são necessários.

Começando pelo ambiente de jogo, temos o campo de futebol, uma estrutura metálica, para sustentar a iluminação e a câmera. Os equipamentos eletrônicos são: câmera de aquisição, o computador para processamento e o transmissor de dados, e por fim os robôs jogadores. Temos o componente lógico também, um programa que realiza todas as etapas de processamento. Estes são os componentes do sistema, mas existem outros (eletrônicos, lógicos) para dar apoio e manutenção dos componentes.

Os robôs possuem rótulos (um tipo de tampa com regiões coloridas) por cima, para serem detectados e identificados na imagem capturada pela câmera, a iluminação serve para melhorar a qualidade da imagem, durante a captura. A identificação de cada elemento do jogo de futebol se dá por cores. O campo tem cor preta e linhas brancas, a bola de golfe utilizada é laranja e os robôs possuem rótulos com determinadas cores. A imagem digital recebida pelo computador é analisada e então são extraídas as informações sobre os jogadores, passando para análise da aplicação, no caso de futebol de robôs é realizada a análise tática da situação. Em seguida são tomadas as atitudes cabíveis. Estas atitudes são traduzidas em sinais compreensíveis ao robô e assim despachadas do computador para o robô, através do transmissor, sem qualquer alteração. Os robôs que são da equipe recebem os dados e então atuam naquele ambiente conforme os dados recebidos. Novamente acontece a aquisição da imagem e realizado processo sequencial mais uma vez. O sistema fica neste ciclo até o termino do jogo.

O sistema descrito se enquadra em duas categorias, *Very Small Size* definida pelo IEEE podendo qualquer organização aproveitá-la, e a categoria originária Mirobot criada pela FIRA. A *Very Small Size* é baseada na Mirobot, seguindo seus regulamentos.

No regulamento é especificado o ambiente de jogo. O campo tem todas as linhas dimensionadas como as cores, superfície do campo e material especificados. A bola utilizada é de golfe e laranja. O jogo não pode ser realizado ao ar-livre, especificam também condição de iluminação. A figura 5 à seguir, mostra o campo utilizado.

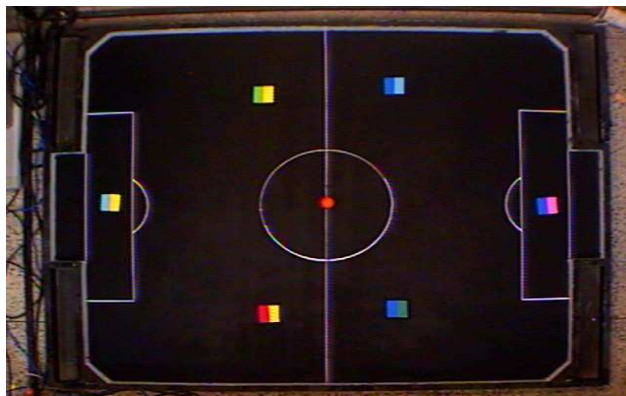


Figura 5: Imagem do campo com robôs

O jogo é realizado por duas equipes cada uma com três robôs jogadores. Cada equipe deve ter seus próprios equipamentos e dispositivos compondo o sistema, sem qualquer compartilhamento com o sistema da outra equipe. Cada robô deve caber num cubo com 7,5 cm de lado, podendo a antena ficar para fora. O rótulo do robô pode ter no máximo duas cores além do preto, para caracterizar a equipe, uma das equipes deve usar a cor amarela e a outra equipe, a cor azul no rótulo. A área da cor da equipe deve ser no mínimo 12,2 cm², uma cor auxiliar pode ser utilizada no rótulo (sem ser a cor da equipe adversária). É permitido substituição e pedir “tempo”. O jogo é dividido em dois tempos de 5 minutos, o intervalo entre eles é de 10 minutos. O jogo pode ser interferido pelo árbitro com os eventos: **pênalti**, **bola livre**, **tiro de meta** e **tiro livre**. Cada um dos eventos tem condições próprias para ocorrerem. A pontuação de uma equipe logicamente ocorre quando a bola entra no gol adversário.

2.2 Componentes do sistema

O robô – O robô utilizado possui duas rodas motoras com acionamento diferencial e caixas de redução. Estas rodas são controladas através de um circuito de potência que por sua vez está ligado à um circuito de recepção e modulação PWM. Os circuitos são alimentados por conjunto de pilhas alcançando 7.2 V e 2500mAh. Cada robô tem um número identificador próprio, isto é caracterizado no microcontrolador (circuito de recepção). O robô é configurável, para jogar numa ou outra equipe.

Câmera - É utilizada à 30 quadros por segundo, com resolução cerca de 12,4 pixels/cm², isto permite uma boa definição do robô como da bola. A imagem gerada pela câmera tem formato RGB e 640 pixels de largura por 480 pixels de altura, a imagem é transportada para outro espaço de cor (HPG) e em seguida passa a ser analisada para extração das informações. Como em outros sistemas de futebol de robô, a câmera rege o número de ciclos por segundo realizado no sistema.

Computador - A organização da parte lógica do sistema robótico é simples, todas as etapas são realizadas dentro do mesmo programa. Desde a aquisição do sensor até transmissão do sinal para o robô. Em outros sistemas, a parte lógica é fragmentada em duas, três partes, geralmente (dois) processos de entrada/saída do sistema robótico se comunicam com (um) processo dedicado à aplicação robótica. Os sistemas operacionais mais utilizados são Linux e Windows, existem sistemas operacionais que foram especializados para a aplicação robótica, garantindo maior tempo de dedicação da máquina para a aplicação, assim é possível fazer processamentos mais elaborados, ou reduzir o tempo de processamento. Os recursos da máquina (processador, memórias) também influenciam no tempo de resposta do sistema robótico.

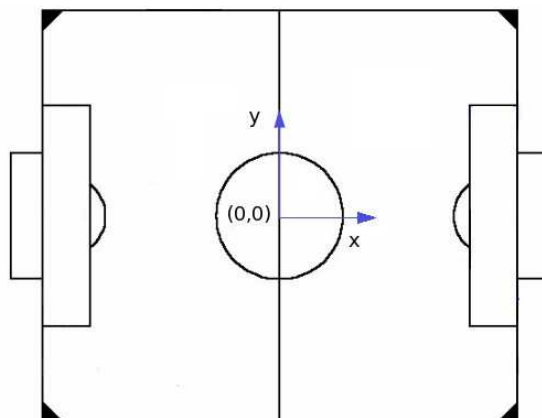


Figura 6: Sistema de coordenadas de mundo

Processos – No sistema robótico em pauta existe um único processo. A parte de análise de dados começa com a busca de cores interessantes na imagem, através de uma amostragem de pixels (a maioria dos pixels não são analisados). Encontrando uma cor interessante é executado um algoritmo de preenchimento (*Seed Fill* melhorado), buscando reconhecer toda extensão de pixels adjacentes com mesmas características, e assim definindo uma região contendo um pixel central e uma cor referente. Assim são obtidas as regiões de cor azul, amarela e laranja na imagem. Em seguida, são buscadas as regiões de cor auxiliar ao redor das regiões de cores principais (amarelo e azul). Encontradas todas as regiões necessárias, pelo centro das duas regiões ditas de um robô é obtido o ângulo do robô e o centro do robô. Para a bola espera-se que exista apenas uma região laranja na imagem isto torna simples o trabalho, a bola terá sua posição sendo o centro da região laranja encontrada anteriormente. A posição do robô é dada em coordenadas retangulares com unidade e metros e ângulo em radianos, estes dados são entradas para a estratégia. A figura 6 elucida as coordenadas.

A parte de transmissão é realizada por um circuito dedicado, ligado ao computador pela porta USB. Os principais elementos do circuito são o microcontrolador e o transceptor. Os dados enviados pelo computador são velocidades da roda esquerda e direita, para cada jogador da equipe. Cada velocidade é codificada num *byte*, sendo um *bit* o sentido de giro e os restantes formam a magnitude de velocidade, existem outros campos na formação do pacote mas são deixados em segundo plano. A função do microcontrolador é fazer a mediação entre dados originados do computador e o transceptor (sincronismo). Existem dois circuitos para transmissão identificados, cada um identifica os dados para sua equipe, de forma a permitir jogos de testes entre duas equipes sem que robôs de uma equipe executem comandos destinados à outra equipe.

Um dos aspectos interessantes nos jogos são as diversidades de equipamentos usados por cada equipe para constituir seu sistema. Cada equipe resolve problemas comuns à todas equipes com soluções originais. Desta forma é possível agregar conhecimentos científicos e tecnológicos nos jogos, comparando as soluções em eficiência, versatilidade, robustez e outros critérios.

3. Trabalhos relacionados

3.1 Competições

Dentro das diversas categorias de futebol de robôs, uma foi mais relevante para o desenvolvimento deste trabalho. A categoria Mirobot, vinculada à entidade FIRA (*Federation of International Robot-soccer Association*). Foi oficialmente fundada em junho de 1997 na Coreia do Sul, a entidade tem objetivo de despertar o espírito de pesquisa nas comunidades científicas pelo mundo, gerando desenvolvimento científico e tecnológico, num ambiente de caráter esportivo.

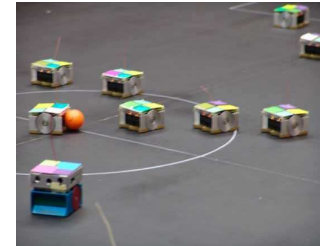


Figura 7: Mirobot

Outras categorias de futebol foram criadas pela FIRA mas a Mirobot continua como sua marca. Um jogo de Mirobot, originalmente, é formado por dois times de três robôs, mas com o tempo, foram adicionados mais jogadores ao jogo, originando duas subcategorias a Mirobot Média (*Mirobot Middle League*) e a Mirobot Grande (*Mirobot Large League*) na figura 7, contendo cinco jogadores cada equipe e onze jogadores cada equipe, respectivamente. Cada categoria tem suas próprias dimensões de campo, mas basicamente em todos outros aspectos (robôs, regras, equipamentos...) são iguais. A atenção para esta categoria se deve pela semelhança. Também pela acentuação do aspecto estratégico na categoria.

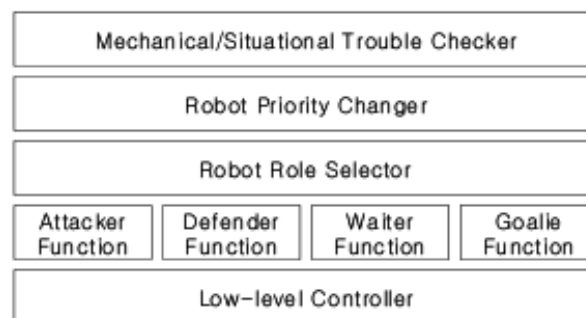


Figura 8: Estrutura da estratégia de SOTY5

3.2 Alguns trabalhos similares

Uma equipe coreana^[1] apresentou uma estratégia estruturada de forma semelhante. Denominaram o sistema de futebol de robôs SOTY 5. Na primeira parte é definido um valor que expressa a integridade e obstrução de cada robô. Na segunda parte é realizada a distribuição das funções disponíveis: atacante, defensor, goleiro e aguardante (*waiter*) que espera a chegada da bola, com base em critérios como posição atual dos robôs, prioridade anterior (cada robô possui seu valor), posição da bola e o critério que envolve tanto defeito mecânico ou obstrução do robô (sintetiza a impossibilidade do robô cumprir tarefas). A terceira parte é atribuir valor da prioridade, para cada robô. A figura 8 ilustra a estrutura da estratégia.

A equipe também incorporou possíveis formações ao time. O time pode usar uma formação ofensiva, ou defensiva, ou simples, ou segmentada. A seguir (figura 9) algumas formações.

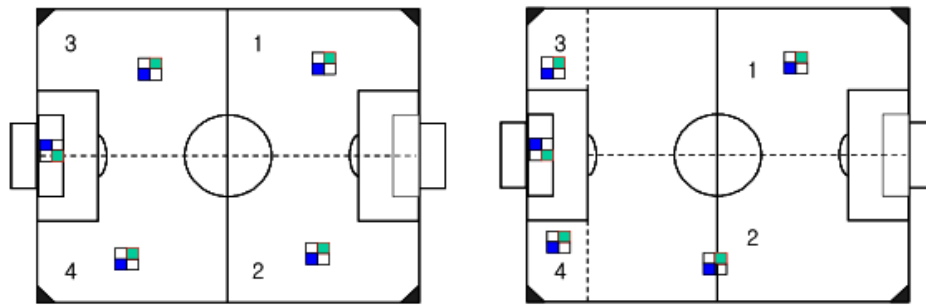


Figura 9: Formações. No lado esquerdo simples e lado direito defensiva

A AUSTRO^[2] da Vienna University of Technology (Áustria) também é uma equipe presente na categoria com grandes semelhanças na parte estratégica. A estratégia segmentada em três camadas, é comandada por uma máquina de estado, a equipe utiliza lógica fuzzy para evitar situações de chaveamento. A estratégia não é implementada de forma direta em linhas de código “if-then-else”, para maior flexibilidade e permitir futuras modificações estruturaram a estratégia em linguagem de descrição derivada do XML (*eXtensible Markup Language*). Uma das versatilidades da linguagem é o uso de termos coloquiais. Desta forma o ambiente que o robô está inserido poder ter seus limites alterados e pode ser incrementado com obstáculos. As camadas da estratégia são nomeadas **estratégia, distribuição de tarefas, ações**. A primeira camada define qual comportamento o time adotará (ofensivo, defensivo), visando alcançar o objetivo maior. A camada de distribuição de tarefas faz a distribuição de funções uma para cada jogador, procurando evitar passar uma mesma função para dois jogadores. A última camada imputa tarefas para cada jogador, tomando como critério de escolha a função do jogador. Também Implementaram tarefas básicas para o futebol de robô. Para execução de uma ação utilizam alguns testes com os predicados *fuzzy* relacionados com a tarefa. O que denominaram tarefa é análogo ao que foi denominado aqui de papéis, e as ações tem analogia com as funções. Também criaram um simulador para verificar o comportamento da máquina de decisão, uma característica interessante é a independência do desempenho da máquina onde é executado, isto é alcançado pela diferenciação de duas escalas de tempo, uma para o modelo físico e outra para exibição de imagem.

A universidade Hradec Kralove^[9], da República Tcheca, apresentou uma estratégia estruturada. Consiste em três níveis de tomada de decisão. No nível mais alto está o reconhecimento da situação do jogo, coordenação do time e seleção de uma reação. No nível intermediário são feitas decisões com respeito a cada robô individualmente, seleção de ação e execução. Na última parte é feito ajuste de parâmetros próprios de cada ação, correção de trajeto, prevenção de colisão, movimentação com bola. O que pode ser enquadrado como etapa de controle do sistema, discutido neste trabalho. Eles utilizam controle PI.

Existem outras equipes^[7,8] que incorporaram mecanismos de aprendizagem na etapa da estratégia. Com diferentes complexidades da etapa, estas equipes geralmente apresentam em um nível inferior da estrutura estratégica, a seleção e utilização de papéis (*skills*). Diversos conceitos de computação já foram aplicados, redes neurais, algoritmo genético, heurísticas adaptativas. A dificuldade na avaliação destas diversas formas é o isolamento da etapa estratégica de cada sistema de futebol de robôs, como a equipe da República Tcheca, a estratégia tem forte vínculo com o controle (acionamento) do robô. Na equipe coreana, a etapa de controle é projetada para o futebol de robôs, reconhecendo limites da área de movimentação. No sistema deste trabalho isto está acima

da etapa de controle.

Existem outras categorias semelhantes, na FIRA, a Narosot é uma categoria derivada da Mirobot. Com robôs de porte um pouco menor. Um evento denominado RoboCup, possui diversos tipos de competições com robôs: coreografias, lutas de robôs, serviços domésticos, futebol de robôs e etc. Uma das categorias do futebol de robôs é chamada *Small Size Robot League* (figura 10). Os robôs tem porte maior que robôs de Mirobot e geralmente apresentam um dispositivo capaz de chutar e segurar a bola. Os jogos apresentam caráter bem estratégico. O planejamento, as ações coordenadas geralmente definem o vencedor do jogo.

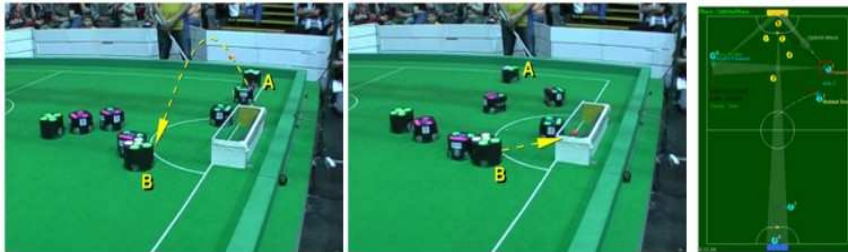


Figura 10:

Categoria Small

Size Robot

League

4. Estratégia

4.1 A estratégia booleana

Na etapa da estratégia são realizadas diversos cálculos para a decisão de qual função é adequada para o jogador na situação. A função escolhida é baseada em predicados booleanos que são considerados mais relevantes para cada papel de jogador (goleiro, atacante e zagueiro). Cada papel de jogador possui seus próprios predicados, por terem diferentes objetivos, assim também, existem funções que são próprias de um papel de jogador.

Podemos considerar que o sistema de decisão tem por entrada os predicados e saída sendo a função adequada para o jogador.

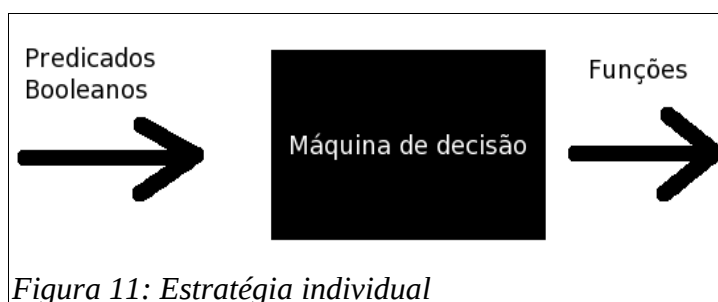


Figura 11: Estratégia individual

A máquina de decisão (figura 11) possui internamente uma tabela verdade que é preenchida na inicialização do programa. O comportamento do jogador é definido por esta tabela, a estratégia da equipe engloba a tabela de cada jogador além da associação de um jogador com um papel no jogo.

Durante o jogo, para cada um dos jogadores, os predicados são calculados, em seguida é consultada a tabela verdade, encontrando a função correspondente. Finalmente é executada a função. No caso de lógica booleana esta máquina pode ser classificada como uma máquina de estado. Onde cada uma das funções é o estado do jogador, e mudando um dos predicados (entrada) teremos um novo estado.

4.2 Implementação

No primeiro momento, o trabalho foi estruturar o código-fonte de forma mais clara para compreensão da estratégia e também tornar mais simples, possíveis alterações na estratégia. Utilizando alguns métodos é possível definir completamente a tabela verdade do jogador e notar se falta a definição de função em algum caso ou se para algum caso foram definidos duas funções diferentes (sobreposição).

A etapa da estratégia implementada consiste em diversos controles de situação. Na ocorrência pênalti, o jogo é parado e jogadores devem se posicionar ou serem posicionados na configuração do pênalti. Existem outras situações que incidem em outras configurações diferentes do pênalti.

Uma das configurações é jogo, nela são tomadas as ações de estratégia no nível de equipe e

em seguida são chamados os procedimentos que caracterizam a estratégia de cada jogador, decidindo a função a ser executada. Todas as funções estão implementadas após a chamada dos procedimentos de estratégia individuais na própria configuração de jogo.

A definição das funções foi realizada através de testes. O goleiro possui duas funções, enquanto o zagueiro e atacante possuem diversas.

A estratégia adota como fato mais relevante o posicionamento da bola, em seguida é verificado se algum jogador do time é capaz de conduzir a bola (através da proximidade da bola).

O papel de zagueiro como de atacante são papéis que necessitam movimentação por grande parte do campo, exigindo maior complexidade para tratamento de situações.

4.3 Zagueiro

O comportamento desejado para este papel precisa ser norteado por algumas características como rapidez para chegar até a bola (quando está no momento atuar), procurar se posicionar entre o gol e a bola, conduzir a bola para longe da área, mesmo com o mínimo sentido de movimento da bola, evitar movimentos que atrapalhem o goleiro ou causem gol contra. Algumas regras do jogo também incidem diretamente no comportamento do zagueiro.

CONTORNAR_ZAGUEIRO : Esta função é deve ser chamada quando o jogador zagueiro está na frente da bola (tomando como orientação o sentido do ataque). É composta de três ações. O primeiro teste verifica se o jogador deve se movimentar para alguma das laterais, afim de evitar toque na bola enquanto volta para trás da bola, em caso verdadeiro o jogador se movimenta para uma das laterais (somente o suficiente para não encostar na bola), em caso negativo já executa a segunda ação que é o movimento para trás da bola, a terceira ação é um alinhamento com a bola, a figura abaixo ilustra melhor esta movimentação.

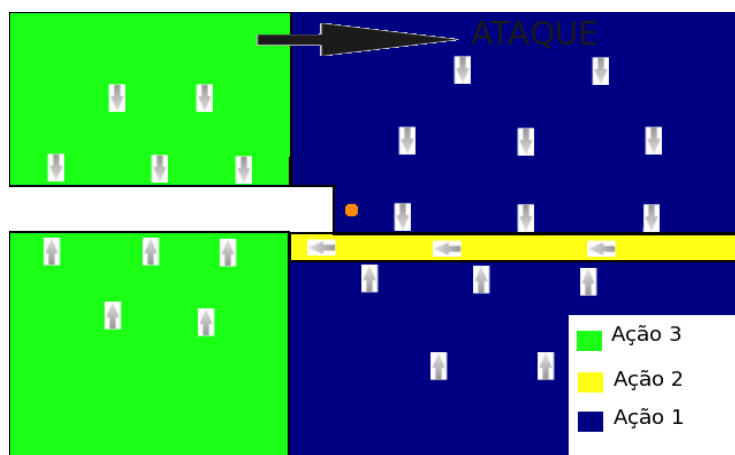


Figura 12: Função de contorno, sobreposição de áreas

Na figura 12, na área azul o robô realiza a primeira ação, na área de cor amarela, realiza a segunda ação e na área verde, a terceira ação. A área em branco é a região desejada para o robô, não realiza qualquer movimento.

TIRAR_BOLA_MEIO : Esta função tem objetivo afastar a bola para as laterais quando ela está numa região à frente da área do time (risco de chute do adversário), tem o princípio de funcionamento parecido com a função acima. O jogador zagueiro possivelmente está na frente da bola, assim procura ficar desalinhado com a bola (evitar choque com bola quando se movimentar para trás). Quando está desalinhado, se movimenta para trás, até que fique lado a lado com a bola,

então gire na direção da bola e finalmente siga em direção da bola, até conduzi-lá para uma distância aceitável, fora da região de perigo. A etapa de giro é importante pois evita que o robô entre dentro da área (não é permitido) quando translada em direção à bola. Neste papel é necessário um efeito de histerese para evitar chaveamento e também tornar a realização da tarefa mais rápida.

DEFENDER : Esta função foca a proteção do gol, em termos mais simples seria “fechar o gol”. Nesta função o zagueiro procura se posicionar entre o gol e a bola (na frente da área), de forma que diminua o espaço de chute do adversário ao gol. Na maioria das situações a posição fica desalinhada com o goleiro (o que é desejável). A função pode ser executada quando não é o momento de atuação do zagueiro, por exemplo, quando a bola estiver do meio-de-campo para frente. É uma função de “guarda”.

RETIRAR_BOLA_PAREDE : É uma das funções mais simples na implementação mas também de fundamental importância, já que a bola encostada na lateral do campo é uma situação muito frequente no jogo. Movimentos em linha reta não ajudam nesta situação, acabam por “travar” o jogador na parede, no máximo podem conduzir a bola para frente ou para trás, permanecendo encostada na lateral. Nesta tarefa o robô realiza uma curva para retirar a bola da lateral. Esta função utiliza apenas a pose do jogador. A partir da pose é calculado a nova referência para ele, a posição da bola não é utilizada, por isso necessita de um prévio posicionamento em relação à bola (adequado), para que não se prenda (e/ou ficar preso) na parede quando executar o papel.

IR_BOLA_ZAG : Nesta tarefa o robô deve se posicionar ligeiramente atrás da bola (já adiantando o posicionamento do robô). Esta função sempre é utilizada para aproximação da bola, devendo ser chamada em grande parte das possíveis situações de jogo. Existem situações onde o robô não pode se posicionar exatamente atrás da bola, para elas é aceitável um posicionamento diferente, mas sempre próximo da bola.

Na execução do programa, após a definição da referência desejada no caso geral, são realizados ajustes (caso necessário) da referência em situações específicas. São realizados alguns testes para evitar referências fora área de movimentação, trajetos por áreas indevidas, e também possíveis toques na bola. Uma situação peculiar e bem prejudicial pode ocorrer na execução deste papel, quando o robô está na frente da bola e tem referência atrás da bola.

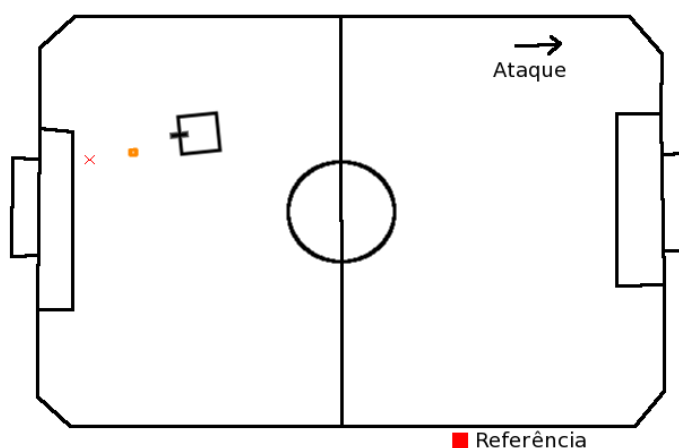


Figura 13: Situação de gol contra

Na ocasião da figura 13, geralmente ocorre o gol contra. Foram realizados diversos testes para tratamento da situação dentro desta função, mas os resultados não foram satisfatórios, não

garantiam segurança. A ineficiência do tratamento mostrou que o problema deveria ser tratado num nível mais alto, na tabela de decisão do jogador. Onde predicados são utilizados para caracterizar esta situação e utilizar outra função mais adequada.

LEVAR_BOLA_ZAG : Dentro desta função o zagueiro conduz a bola “para frente”. Não importando para onde está levando para o próprio gol, para o campo de ataque, para a lateral. A referência do jogador é definida após a bola, assim antes de conseguir chegar à referência estará empurrando a bola. Esta função é própria para afastar a bola do jogador adversário e também conduzi-la até o campo de ataque.

É preciso conhecer a situações em que se executa esta função, para se excluir a situação ilustrada na figura 13, onde o jogador também causaria gol contra.

POSICIONAR_ZAGA: Esta função foi criada para situações onde o zagueiro não possa atuar sobre bola ainda que esteja no campo defensivo, ou situações que possa prejudicar, “atrapalhar” a atuação de outro jogador do time. O jogador fica numa posição definida, próximo da quina da área . A posição da bola é dada por um valor na coordenada X e outro valor na coordenada Y. O valor Y do zagueiro têm mesmo sinal que o valor Y da bola, para ficar mais próximo da bola.

TIRAR_BOLA : Também é uma função de simples implementação mas de muita importância. Consiste em um giro contínuo no próprio eixo do robô, podendo ocorrer no sentido horário ou anti-horário. Esta ação, em testes, cumpriu com êxito a tarefa de retirar bolas nas regiões de quina do campo. As regiões de quina merecem bastante atenção por impor restrições na movimentação do robô. Movimentos nesta região que envolvem translação terminam por prender o robô nas paredes, até movimentos puramente rotativos, por vezes, acabam na mesma situação. O uso da função foi previsto para situações em que o jogador está na frente da bola e que seja muito complexo ou impossível o contorno da bola. Esta função deve ser antecedida por outra função de posicionamento. Um fato peculiar que pode prejudicar o controle do zagueiro é perder as condições de saída desta função (troca de função). Para que esta função seja executada é necessário que os predicados tenham valores específicos, especialmente um predicado que define se o robô está próximo da bola, caso esteja próximo executa esta função (**TIRAR_BOLA**) em caso negativo outra função. Dependendo de como foi definido o predicado de proximidade, o robô pode ser considerado próximo da bola mesmo sem conseguir com o giro, tocar na bola, assim a bola permanece na sua posição como o robô também permanece (girando no seu próprio eixo), o que não altera o valor do predicado de proximidade, isto caracteriza um estado “sem saída”. Para evitar isto é necessário que o predicado de proximidade garanta que se o robô entrar nesta função, consiga tocar na bola, afastando-a do robô e tornando falso o predicado de proximidade, para saída deste estado.

Isto não acontece na função de **RETIRAR_BOLA_PAREDE** justamente por causa da translação que ocorre, tornando falso o critério (predicado) de proximidade com a bola.

POSICIONAR_RETIRADA : É a função que deve anteceder os movimentos de retirada da bola das regiões difíceis (paredes e quinas) para depois realizar as manobras gerais do robô (contorno, conduzir a bola...). A referência definida de forma à conseguir completar o movimento seguinte de retirada da bola. Dentro da função é dividida cada uma destas regiões, onde o robô tem uma pose específica em relação a bola para cada região. É uma função que influencia no bom ou mal desempenho do jogador.

SAIR_DA_AREA : Uma das restrições do jogo, é a permanência de apenas um jogador na área do time, comumente designado goleiro. Se o time infringir a regra, é marcado o pênalti contra a equipe. As funções do papel de zagueiro tem a movimentação muito próxima da área. Para evitar a infração, as funções com movimentação mais próxima da área fazem prevenção de movimentos em

direção à área. Mas durante o jogo é possível um robô adversário empurrar o zagueiro para dentro da área. Por isso é necessário tanto a prevenção como a correção. Esta função visa a correção. Considerando que o zagueiro esteja dentro da área, a posição definida para o jogador é calculada testando a proximidade dele com a linha frontal da área e com as linhas laterais. A referência é projetada para fora da linha mais próxima do robô. Esta é uma forma da referência atingir qualquer ponto limite fora da área e interferir o mínimo possível na movimentação do robô. A figura 14 mostra a função.

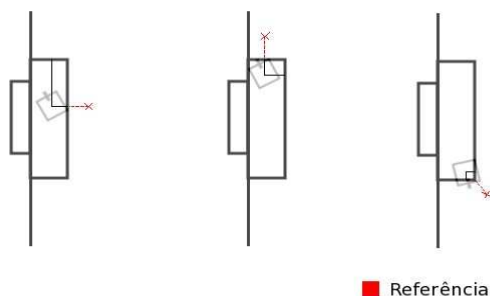


Figura 14: Função para sair da área

4.4 Atacante

O jogador com este papel tem objetivo de levar a bola para dentro do gol adversário. Para alcançar o gol, são necessários cálculos geométricos. Deve considerar também aspectos de cinemática para conseguir se antecipar aos movimentos da bola, levando grande vantagem sobre jogadores adversários. Estes cálculos dependem diretamente da qualidade dos dados obtidos nas etapas de aquisição. Esta função torna visível algumas características gerais do sistema global como precisão dos dados e precisão do controle de pose dos robôs. Tarefas de certa complexidade só poderão ser realizadas baseadas nestas qualidades do sistema.

Este papel utiliza algumas funções já explicadas no papel de zagueiro. O contorno, o posicionamento para retirada de bola, movimento rotativos (`RETIRAR_BOLA_PAREDE`, `TIRAR_BOLA`). Existem outras funções destinadas ao atacante e outras funções semelhantes à funções citadas anteriormente.

POSICIONAR : É uma função para que o atacante aguarde a chegada da bola no campo de ataque. O robô tem uma posição pouco a frente da linha do meio-campo, quando a bola tem o valor Y positivo, fica numa posição com Y negativo, sempre o Y do robô com sinal oposto ao Y da bola.

IR_BOLA : É uma função simplificada de aproximação da bola, sem qualquer tratamento, a referência do robô é a posição da bola.

GIRAR : Uma função para apontar o jogador em direção à bola. A posição de referência é definida como a própria posição do robô (para não se mover), apenas o ângulo de referência é alterado, de forma que o jogador gire no próprio eixo, até que “aponte” para a bola.

ALINHAR : Esta função já é própria de atacante, nela é executado o posicionamento do jogador em relação à bola (figura 15), deixando-o alinhado com a reta definida pela bola e o centro do gol adversário. Esta função deve levar em consideração uma posição futura da bola. A velocidade e direção da bola são valores já disponíveis na etapa da estratégia. A distância do jogador até a bola é constante. Uma forma simples de se antecipar a posição da bola é o posicionamento sobre a reta

que a bola descreve

CHUTAR_GOL : Esta função é o último estado do jogador quando consegue se posicionar em relação à bola, em seguida girar em direção à bola e por fim chutar a bola. Como o robô não possui atuador para chute, o que é realizado é a condução da bola até a linha de fundo do campo de ataque.

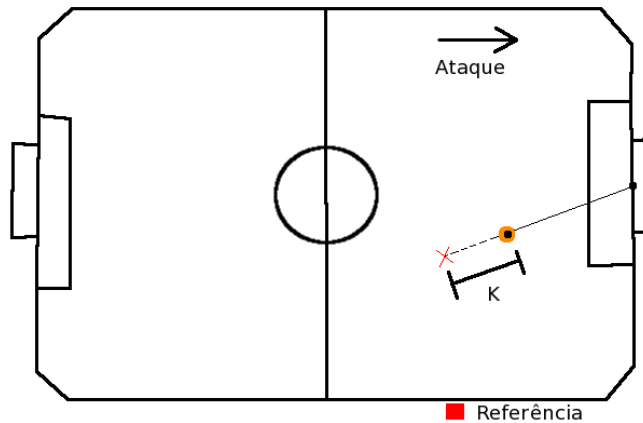


Figura 15: posição desejada para chute

Esta sequência de ações (alinhar, girar e chutar) caracteriza o papel de atacante. Experimentalmente foi verificado um problema de chaveamento de funções nesta sequência. A origem do chaveamento é devido à definição dos predicados relevantes no momento. Para realizar o chute é necessário que o robô esteja orientado para a bola, um predicado definido **esta_orientado** será verdadeiro caso o ângulo do robô esteja dentro de uma faixa de tolerância, tendo como centro (bissetriz) a posição da bola. Intuitivamente esta faixa deve ser minimizada para que o robô, no momento em que validar o predicado **esta_orientado**, consiga atingir a bola e também chegar ao gol adversário com máxima precisão. Mas diminuir a faixa de tolerância torna difícil a validação do predicado **esta_orientado**. A outra situação extrema é aumentar a faixa de tolerância, o problema é que o robô não “acertará” a bola, na trajetória executada. Após conseguir definir estes limites, uma dificuldade que geralmente ocorre é durante a execução do chute, neste instante o robô realiza uma translação, um movimento que experimentalmente não preserva a orientação do robô. Assim acaba tornando falso o predicado **esta_orientado**, podendo também tornar outro predicado (definimos como **esta_alinhado**) assim o robô tem que voltar (ou reiniciar) a sequência de funções.

Estes dois problemas causam o comportamento de chaveamento no robô. O primeiro é inerente à calibração dos predicados (calibração soluciona). O segundo é inerente de atuadores reais, com limitações e imperfeições. A solução encontrada foi a histerese. A histerese implementada é um mecanismo de modificar a definição do predicado, usando uma definição mais restrita quando o robô está no estado (função) associado e uma definição mais abrangente quando o robô está no estado seguinte da sequência. Por exemplo **esta_orientado** tem uma definição mais restrita quando está no estado GIRAR, e uma definição mais relaxada quando entrar no estado CHUTAR_GOL.

A histerese de uma forma mais geral divide um limite entre dois estados em dois limites diferentes, existe um limite para passar de um estado A para um B e existe outro limite para passar do estado B para o estado A. Existem outras sequências de ações que também necessitam de histerese. A histerese em outros contextos é problema, mas neste é solução.

4.5 Goleiro

Este papel é atribuído a um jogador de forma estática. Durante o todo o jogo permanece o mesmo jogador. Diferente dos outros papéis. O goleiro não possui grande movimentação, para ele foram designadas duas tarefas, defender o gol e chutar a bola para fora de sua área. Assim, existem duas funções próprias do goleiro. Deve se antecipar aos movimentos da bola como o atacante.

GOLEIRO : Esta função é para impedir a passagem da bola para dentro do gol. O robô fica posicionado de forma que possa percorrer toda linha do gol, sempre dentro do gol. Para defender verifica se a bola está no campo de defesa, se a bola estiver em baixa velocidade, procura se posicionar “de frente” para a bola. Caso a bola esteja em alta velocidade o goleiro se posiciona na linha onde a bola deve passar.

TIRAR_DO_GOL: Sabendo que os outros jogadores do time não podem entrar na área, necessariamente o goleiro deve “chutar” a bola para fora da área. É necessário que esta situação esteja bem definida para que o goleiro não “abra espaço” para os jogadores adversários. Para isso a bola deve estar parada (ou quase parada) e dentro da área. O chute é definido como uma referência fora da área que tem uma certa distância da bola (suficiente para garantir que a bola fique fora da área), este ponto fica sobre a reta definida pela posição do goleiro e da bola.

4.6 Os predicados

Cada um dos papéis tem objetivos diferentes, em consequência, seus predicados devem “detectar” aspectos importantes do papel.

O goleiro possui 3 predicados importante, já o zagueiro possui dez, o atacante possui 8.

Os predicados do goleiro poderiam ser comprimidos em único predicado por possuir apenas duas funções, mas por organização e facilitar a compreensão, não foram alterados.

esta_na_area: Este predicado testa se o goleiro está dentro da sua área.

bola_parada: Faz um teste com velocidade da bola se for menor que uma constante definida é considerada parada.

bola_na_area: Verifica se a bola esta na área do goleiro.

O zagueiro precisa “conhecer” mais situações que o goleiro, porque tem maior área de movimentação, possui restrições de jogo e para evitar prejudicar o time.

bola_nossa_area: Quando for verdade, o zagueiro nada pode fazer, devendo aguardar a saída da bola ou o gol adversário.

bola_na_zaga: Com este predicado o zagueiro deve começar a atuar e cumprir sua tarefa (conduzir a bola para frente).

bola_na_parede: Este predicado sinaliza quando a bola está proxima das paredes (laterais ou de fundo) do campo. Assim pode fazer uma aproximação mais cautelosa.

alinhado: Este predicado é composto por dois, um verifica se o zagueiro esta numa região (a frente da área) definida como perigosa, o outro predicado é um teste se esta orientado para bola. Caso não

esteja na região perigosa, o predicado composto é verdade, mas se estiver dentro da região perigosa o predicado composto terá valor do predicado de orientação.

bola_comigo: Testa se está próximo da bola.

colado_na_bola: Testa se está ainda mais próximo da bola, a ponto de tocar na bola sem qualquer translação.

atras_da_bola: Para o robô reconhecer se precisa fazer o contorno da bola ou não.

pode_ficar_atras: Existem regiões no campo onde robô não consegue/pode ficar atrás da bola, quando a bola está próxima da parede de fundo ou na frente da área e próxima da área.

alinhadoy: É um teste que analisa o valor Y das posições da bola e robô, um teste de proximidade entre eles.

dentro_da_area: Para correção da posição do robô, caso venha a entrar dentro da área do goleiro, o predicado é verdadeiro.

O atacante tem maior preocupação conseguir uma posição atrás da bola favorável ao chute.

n_bola_no_ataque: Predicado para ativar ou não o atacante, falso se a bola está no ataque.

atras_da_bola: Igual ao predicado do zagueiro.

consegue_alinhar: Testa se a bola está numa posição onde atacante, a bola e centro do gol adversário estejam em linha reta.

consegue_chutar: Analisa se o robô com a orientação atual seguisse em linha reta, chegaria no gol adversário.

bola_na_parede: Verifica se a bola está próxima das paredes laterais do campo.

bola_comigo: Predicado análogo ao predicado do zagueiro.

colado_na_bola: Igual ao predicado do zagueiro.

bola_no_fundo: Verdadeiro quando a bola está na parede de fundo do campo.

Estes foram os predicados considerados mais relevantes para a tomada de decisão. A seguir, as tabelas dos papéis, a tabela 1 referente ao goleiro, tabela 3 referente ao zagueiro e tabela 2 do atacante.

Tabela 1: Funcionamento do goleiro

esta_na_area	bola_parada	bola_na_area	FUNÇÃO
0	X	X	GOLEIRO
1	0	X	GOLEIRO
1	1	0	GOLEIRO
1	1	1	TIRAR_DO_GOL

Tabela 2: Funcionamento do atacante

n_bola_no_ataque	atras_da_bola	consegue_alinhar	consegue_chutar	bola_na_parede	bola_comigo	colado_na_bola	bola_no_fundo	FUNÇÃO
1	x	X	X	X	X	X	X	POSICIONAR
0	1	1	1	X	X	X	X	CHUTAR_GOL
0	1	1	0	X	X	X	X	GIRAR
0	1	0	X	1	0	X	0	IR_BOLA
0	1	0	X	1	1	X	0	RETIRAR_BOLA_PAREDE
0	1	0	X	X	X	0	1	IR_BOLA
0	1	0	X	X	X	1	1	TIRAR_BOLA
0	1	0	X	0	X	X	0	ALINHAR
0	0	X	X	X	X	X	X	CONTORNAR

Tabela 3: Funcionamento de zagueiro

a	b	c	d	e	f	g	h	i	j	FUNÇÃO
X	X	X	X	X	X	X	X	X	1	SAIR_DA_AREA
0	0	X	X	X	X	X	X	X	0	DEFENDER
1	0	X	X	X	X	X	X	X	0	IMPOSSIVEL
1	1	X	X	X	X	X	X	X	0	POSICIONAR_ZAGA
0	1	0	X	X	X	X	0	X	0	TIRAR_BOLA_MEIO
0	1	0	0	X	X	1	1	X	0	GIRAR
0	1	0	1	X	X	1	1	X	0	LEVAR_BOLA
0	1	1	0	0	X	1	1	X	0	GIRAR
0	1	1	1	0	X	1	1	X	0	IR_BOLA_ZAG
0	1	1	X	1	X	1	1	X	0	RETIRAR_BOLA_PAREDE
0	1	0	X	X	X	0	1	0	0	IR_BOLA_ZAG
0	1	0	X	X	X	0	1	1	0	CONTORNAR_ZAG
0	1	1	X	X	X	0	0	0	0	IR_BOLA_ZAG
0	1	1	X	X	0	0	0	1	0	POSICIONAR_RETIRADA
0	1	1	X	X	1	0	0	1	0	TIRAR_BOLA
0	1	1	X	X	X	0	1	0	0	IR_BOLA_ZAG
0	1	1	X	X	X	0	1	1	0	CONTORNAR_ZAG
0	1	1	X	X	0	1	0	X	0	POSICIONAR_RETIRADA
0	1	1	X	X	1	X	0	X	0	TIRAR_BOLA

a: bola_nossa_area
b: bola_na_zaga
c: bola_na_parede
d: alinhado
e: bola_comigo

f: colado_na_bola
g: atras_da_bola
h: pode_ficar_atras
i: alinhadoy
j: dentro_da_area

5. Estrutura

Todas as etapas realizada em computador são condensadas em um único programa, mas todas etapas possuem um procedimento principal (como descrito na figura 16), chamado dentro do laço principal do programa. Assim o interesse neste trabalho é focar o conteúdo da etapa de estratégia.



Figura 16: Etapas de processamento

A etapa de estratégia consiste na chamada de um método classe **Strategy** denominado **strategy** dentro deste procedimento são realizadas as principais tarefas da estratégia. Vale ressaltar que o paradigma de programação é a orientação à objeto.

O procedimento **strategy** pode ser dividido em três trechos principais (figura 17), o primeiro define o jogo de futebol em estados, os estados considerados são **pênalti**, **bola livre**, **chute livre**, **tiro de meta**, **posicionamento inicial** e **jogo**. Apenas o último estado (**jogo**) envolve a execução de tarefas (funções) para os robôs, os outros estados definem posicionamentos adequados quando ocorrer alguma penalidade durante o jogo, penalidades previstas em regulamento.

O segundo e terceiro trechos estão dentro do escopo do estado **jogo**. O segundo trecho envolve a atribuição de funções para cada um dos jogadores, cada um em seu papel (goleiro, atacante ou zagueiro). Este trecho pode ser caracterizado como uma máquina de decisão.

Na terceira etapa são executados as funções atribuídas aos jogadores.

Os principais dados compartilhados na etapa da estratégia, são variáveis que indicam a função de cada robô, uma tabela verdade para cada função de jogador (três ao todo) contendo propriamente a estratégia associada à função. Uma entrada destas tabelas é definida como um conjunto de valores (binários) dos predicados do papel, o que se busca na tabela é a função para uma certa entrada. As tabelas são implementadas em vetores. O tamanho dos vetores vem do número de predicados de cada papel. Foram implementados alguns métodos para facilitar o uso da tabela e também realizar alguns testes. Outro dado importante é o vetor de funções escolhidos para os jogadores. Estes dados são internos à etapa da estratégia. Os dados externos, de comunicação com etapas adjacentes são: o vetor de posições, podemos considerar como entrada desta etapa; e o vetor de referências, como saída desta etapa.

No início o segundo trecho é testado se o jogador está em condição de jogo ou não, pois

algumas situações como travamento na parede (ou com outro jogador), jogador não localizado, são importantes na atribuição de papéis aos jogadores. Daí caso seja necessário ocorre a troca de papéis entre jogadores. Em seguida são chamados alguns métodos da classe **Strategy** para escolha das funções, um método para cada papel. Dentro destes métodos secundários existem algumas variáveis auxiliares, utilizadas em seguida, no cálculo do predicados daquele papel. Finalizando, é feita a consulta da tabela, para definição da função. Isto está ilustrado na figura 18. Existe ainda um momento inicial de preenchimento das tabelas, durante construção do objeto.

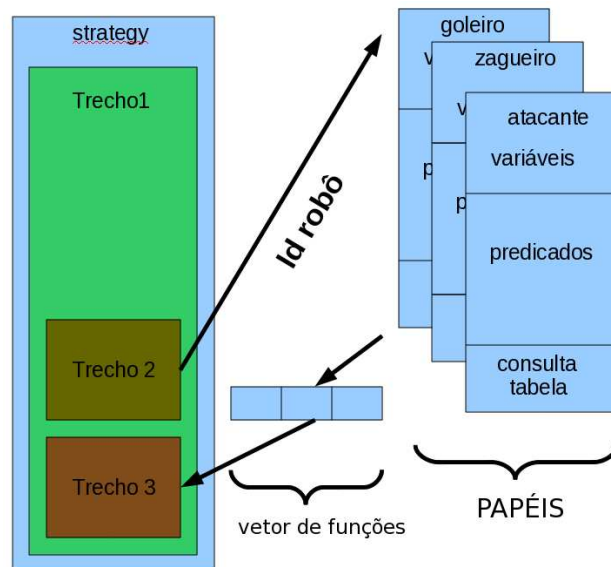


Figura 17: Visão geral da etapa de estratégia

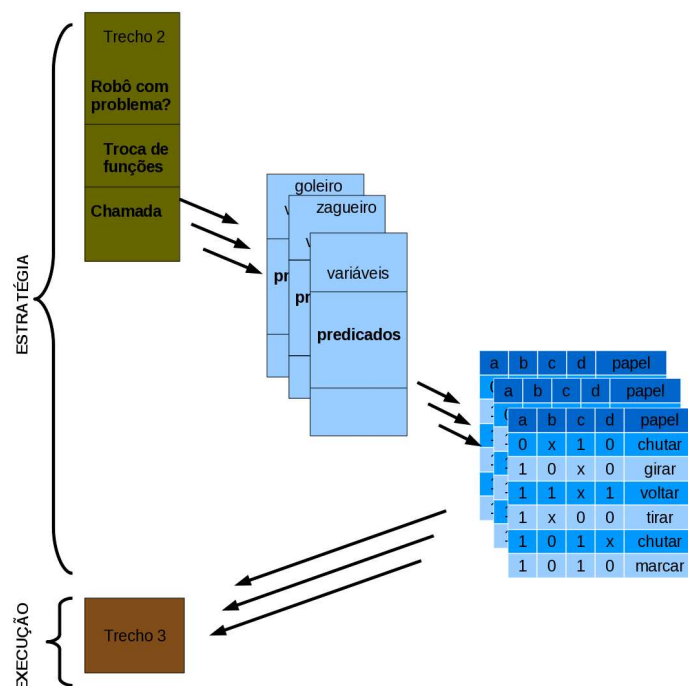


Figura 18: Escopo da estratégia

6. Resultados e Perspectivas

6.1 Experimentos e resultados

Para verificação das funções, das tabelas, primeiro foram testadas as funções acrescentadas, isoladamente, após o funcionamento satisfatório, foi testada a tabela que utilizava as funções. Para testes iniciais foi utilizado um programa simulador disponível, depois do funcionamento correto partia-se para os experimentos reais, os erros encontrados foram corrigidos na mesma etapa onde foram encontrados. Se foi na simulação, na simulação foi corrigido, se foi no experimento real, no experimento foi corrigido.

Os papéis que foram alteradas na parte estratégica foram zagueiro e atacante. O goleiro apresenta já apresenta um comportamento satisfatório, assim não foram alteradas suas funções ou estratégia.

No simulador a estratégia implementada possui bom desempenho, apresentou melhorias, mas há algumas situações onde o robô fica preso na parede, na tentativa de retirada da bola. O que pode ser solucionado com a troca de papéis, o robô zagueiro toma o papel de atacante, e o robô atacante toma o papel de zagueiro. Ainda assim é notável as melhorias no sistema.



Figura 19: Programa simulador, importante no desenvolvimento

6.2 Propósito da lógica “fuzzy”

Durante a etapa de estratégia é definida uma pose (um ponto nas coordenadas do campo e uma orientação para a frente do robô) como referência para o robô. A escolha da referência é feita de acordo com pequenas tarefas (funções) que o robô pode executar. Utilizando predicados binário (verdadeiro ou falso) é possível tomar a decisão de qual a função mais adequada para uma situação. Um fato inconveniente neste ponto é que se um dos predicados está próximo do limiar entre verdadeiro e falso geralmente ocorre um período de chaveamento do valor (entre verdadeiro ou falso) do predicado em certos momentos este predicado pode ser decisivo na escolha da função adequada, caso este dois fatos ocorram no mesmo instante, o robô ficará alternando entre duas funções, pois a velocidade de execução do *loop* do sistema é muito maior que o tempo demandado

para execução de uma tarefa (se trata de um computador controlando um sistema físico). O robô fica “perdido”, com movimentos bruscos. Existe ainda uma possibilidade que robô realize um movimento compreensível, caso as poses de referência resultantes de cada função estejam muito próximas uma da outra (o que é muito raro).

Uma forma de contornar este problema é incluir um mecanismo de histerese na transição de estados (funções), de modo que torne fácil a entrada do robô num estado, mas difícil sua saída. Mesmo em transições de funções que não ocorram o chaveamento mas seja uma única transição, pode mostrar um comportamento brusco na transição.

O uso da lógica “fuzzy” na etapa de decisão de papéis (estratégia) pode solucionar estes problemas de forma mais natural. No caso da mudança de função sem chaveamento pode mudar gradativamente a referência. Esta mudança suave de referência também amortiza o chaveamento devido a proximidade das referências no período de transição, o movimento deve ser mais suave. Além disso também trata os predicados necessários para tomada de decisão de forma mais real.

Foram realizados alguns teste preliminares, a utilização da lógica *fuzzy* mostrou-se viável.

7. Conclusões

Através da pesquisa verificou-se o gradativo avanço dos sistemas robóticos de futebol de robô na análise tática. A etapa da estratégia é uma área de aplicação de conhecimento científico diferente das demais que geralmente envolvem aspectos de tecnologia.

É possível fazer gradativas melhoras no sistema criando mais funções especializadas. Com a utilização das tabelas fica fácil realizar mudanças na estratégia durante a execução do programa. Possibilitando mudanças no comportamento de jogadores durante o jogo. Outra melhoria, é a facilidade de ajustar a estratégia quando os predicados e funções não precisam ser alterados, necessitando de ajustes apenas nas tabelas, que se restringem a poucas linhas do código. Em comparação de tempo com uma estratégia implementada diretamente em código, tem uma pequena desvantagem que é a necessidade de calcular todos os predicados, enquanto numa estratégia implementada diretamente normalmente poucos predicados são realmente calculados.

De forma geral o estruturamento da estratégia possibilita maior versatilidade na etapa da estratégia.

Referências bibliográficas

[1] Han K.-H., Lee K.-H., Moon C.-K., Lee H.-B. & Kim J.-H., **Robot Soccer System of SOTY 5 for Middle League MiroSot**, 2002.

[2] Egly U., Novak G. & Weber D., **Decision Making for MiroSot Soccer Playing Robots**, *Vienna University of Technology*, Vienna, Austria.

[3] Martins D. L., Mendes E. P., Lins C. A., de Araújo M. V., de Medeiros A. A. D., Alsina P. J., **A versão 2007 da equipe POTI de Futebol de Robôs**, Universidade Federal do Rio Grande do Norte, Natal, Brasil, 2007.

[4] FIRA, *Federation International Robot-soccer Association*, disponível em <http://www.fira.net> .

[5] ROBOCUP, *Robot Cup*, disponível em <http://www.robocup.org> .

[6] Bojadziev, G. & Bojadziev, M., **Fuzzy sets, fuzzy logic, applications**, ed. World Sci, 1995.

[7] Huang, H.-P. & Liang C.-C., **Strategy-based decision making of a soccer robot system using a real-time self-organizing fuzzy decision tree**, ed. Elsevier, 2002.

[8] Freedman, H. G. & Mon, G., **How Spiritual Machine Works**, apresentado no congresso 12^o Congresso FIRA 2007, 2007.

[9] Kotzian, J., Kozany, J., Machacek, Z., Srovnal Jr, V. & Tucnik, P., **Complex System Solution for MiroSot soccer's robot**, *University of Hradec Kralove e VSB- Technical University of Ostrava*, República Tcheca.

[10] de Medeiros A. A. D., **Introdução à Robótica**, Universidade Federal do Rio Grande do Norte, apresentado no Encontro Nacional de Automática (SBPC), 1998.