

Desenvolvimento de Estratégia para Programação do Futebol de Robôs da Mauá

Wânderson O. Assis, Alessandra D. Coelho, Marcelo M. Gomes,

Cláudio G. Labate, Daniel F. Calasso, João Carlos G. C. Filho

**Escola de Engenharia Mauá – Centro Universitário do Instituto Mauá de
Tecnologia (CEUN - IMT)**

Praça Mauá, 1 – CEP 09580-900 – São Caetano do Sul – SP – Brasil

A estratégia de jogo tem um papel fundamental no futebol de robôs. A estratégia é o “raciocínio artificial” do robô desenvolvido por meio de algoritmos de controle e inteligência artificial. Uma das principais dificuldades para as equipes de futebol de robôs é a de conseguir fazer com que os robôs apresentem na prática os resultados que seriam previstos ou desejados com a programação. Devido a um grande número de interferências no processo, o comportamento dos robôs acaba ficando comprometido. Diante disso, a utilização de *softwares* simuladores é uma alternativa interessante porque permite avaliar a eficiência dos algoritmos de controle sem a presença de interferências, tais como: distorções na imagem captada pela câmera, interferência da luminosidade externa, falhas no sistema de transmissão por radiofrequência etc.

Neste artigo apresenta-se, na primeira seção, a utilização do *software* Simurosot como ferramenta para a programação da estratégia do futebol de robôs. Na segunda seção, são apresentados vários aspectos adotados no desenvolvimento do programa que controla automaticamente o time de futebol de robôs da Mauá.

1 - SIMULAÇÃO DA ESTRATÉGIA

A inteligência do sistema futebol de robôs, programada com linguagem de alto nível, funciona em forma de *looping* conforme ilustrado na Figura 1. O algoritmo tem a função de decidir as ações dos robôs com base na identificação da localização de cada componente (robôs e bola). Após a identificação, o algoritmo define as decisões do time (estratégia de jogo) e envia essas informações para o simulador de forma que seja atingido o objetivo.

Para simular a estratégia do futebol de robôs da estratégia foi utilizado o *software* da Simurosot (Figura 2), disponível no site da FIRA (Federação Internacional de Futebol de Robôs): <http://www.fira.net/soccer/simurosot/overview.html>.

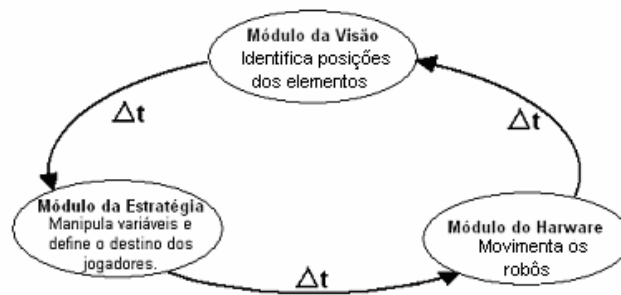


Figura 1 – Estrutura do Algoritmo de Inteligência Artificial



Figura 2 – Simulador

O software baseia-se em funções matemáticas para a movimentação de cinco robôs para serem integradas a um sistema de inteligência artificial que controlará a estratégia de jogo de um time de futebol de robôs.

O programa conta com algumas funções já elaboradas como: Posicionar (posiciona o robô num ponto estabelecido), Girar com bola (para fazer o robô contornar a bola e levá-la para o campo adversário), Marcação (é calculada a posição que deverá ser ocupada pelo robô onde a bola é o ponto médio entre o robô e o robô do adversário que estiver mais próximo da bola), Barreira (os robôs tentam ficar no ponto médio entre a bola e uma média ponderada entre o ponto central, o canto superior, e o inferior do gol, onde os pesos são escolhidos de acordo com a proximidade dos robôs a esses), entre outras funções necessárias para um jogo de futebol de robôs. Foi necessária a inclusão de mais funções para o melhor desempenho dos robôs no campo.

A Figura 3 mostra a tela principal de simulação de partidas de futebol de robôs:



Figura 3 – Tela principal do programa simulador

Uma grande vantagem desse simulador está em monitorar todas as coordenadas dos robôs e posição da bola, como se apresenta na Figura 4.

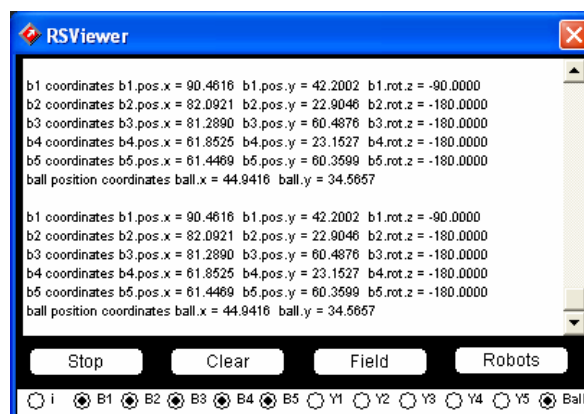


Figura 4 – Coordenadas dos robôs e bola

Outra qualidade do simulador seria a possibilidade de rever a jogada da estratégia (Figura 5), essencial para analisar detalhes:



Figura 5 – Análise em replay

O programa da estratégia desenvolvido para a simulação do futebol de robôs está no **Anexo**. Esse algoritmo pode ser utilizado pelas equipes iniciantes para verificar o funcionamento do futebol de robôs e desenvolver novas estratégias. Para o desenvolvimento desse algoritmo foram considerados os mesmos aspectos utilizados na estratégia real do futebol de robôs da Mauá. A descrição desses aspectos será apresentada na próxima seção.

2 - ESTRATÉGIA REAL

A estratégia do futebol de robôs da Mauá, programada em Delphi®, tem as funções de decidir as ações dos robôs com base em sua posição original informada pelo sistema de visão computacional, e de enviar os comandos de controle para desenvolver a estratégia de jogo.

O diagrama de blocos do algoritmo de estratégia é apresentado na Figura 6. Cada uma das sub-rotinas que compõem o algoritmo serão descritas na sequência.

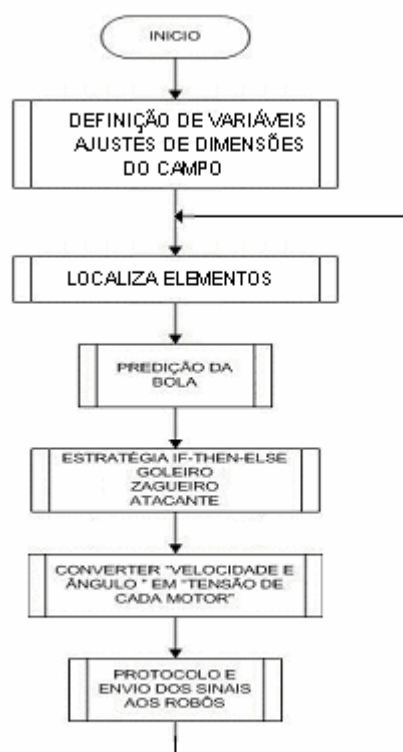


Figura 6 – Fluxograma da Estratégia

Definição de Variáveis, Ajuste de Dimensões do Campo e Localização de Elementos

Para executar a rotina de estratégia é necessário, antes de tudo, conhecer a localização de cada elemento (robôs e bola), seja no simulador, seja no sistema real.

Foram definidas variáveis que identificam ângulo, velocidade, posição anterior da bola, posição atual da bola, posição futura da bola, coordenadas do campo (pontos extremos, gol, área, meio de campo), entre outras.

Foram criadas constantes de dimensionamento do campo, para que haja um ajuste manual, caso a imagem captada com a câmera esteja torcida, ou fora de ângulo (Figura 7), fazendo que os valores captados pelo *Módulo de Visão* fossem diferentes dos reais.

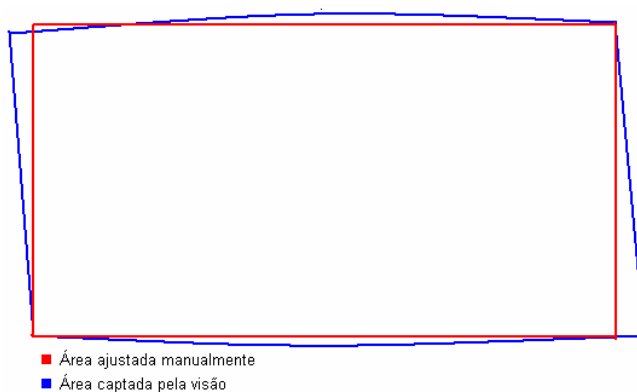


Figura 7 – Ajuste de Dimensões do Campo

Constantes de tolerância de posicionamento foram colocadas para que o destino de posições não seja pontual, mas uma área para que reduza a instabilidade do robô. Por exemplo, como se ilustra na Figura 8, se o destino de um robô for (50,50), é definida uma região em torno do ponto no qual o robô irá parar quando atingir essa região. Podemos definir regiões como $(40 < x < 60)$ e $(40 < y < 60)$.

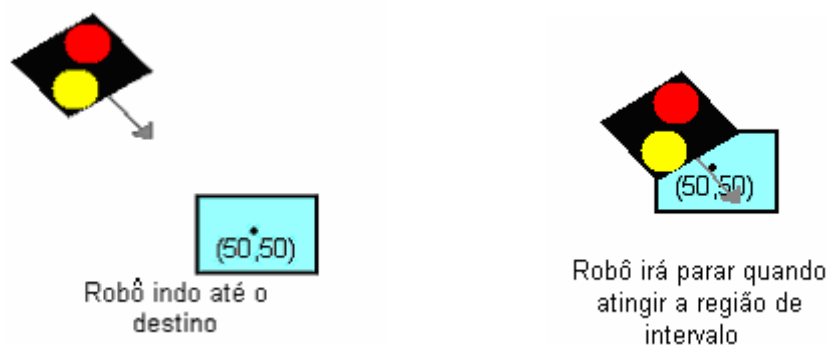


Figura 8 – Utilização de Constantes de Tolerância para o Posicionamento dos Robôs

Predição da Posição da Bola

Cada rotina demora um tempo (certo número de ciclos de máquina) para ser completa. No simulador, os módulos de Visão e *Hardware* são desenvolvidos virtualmente. Como todos os módulos são desenvolvidos diretamente no computador, isso pode ser efetuado muito mais rapidamente que no futebol de robôs convencional onde existe um atraso de resposta Δt considerável em cada etapa do sistema. Ainda assim, o tempo de processamento nesses módulos deve ser considerado. Por isso foi utilizado um algoritmo de predição. Nesta estratégia será apresentada apenas a previsão da bola, contudo deve-se também fazer a predição da localização dos robôs jogadores.

Foi usado um conceito muito simples de semelhança de triângulo (Figura 9):

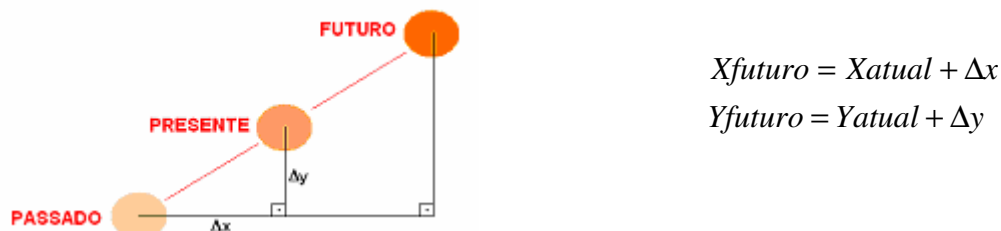


Figura 9 – Algoritmo de Predição

A diferença entre o valor passado e o valor presente é muito pequena. Por isso foi definido um coeficiente que amplia a extrapolação, da seguinte maneira:

$$X_{\text{futuro}} = X_{\text{atual}} + \text{coef} \cdot \Delta x$$

$$Y_{\text{futuro}} = Y_{\text{atual}} + \text{coef} \cdot \Delta y$$

Analisou-se o caso de rebatida como se mostra na figura abaixo:

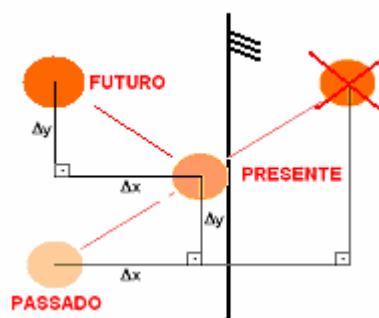


Figura 10 – Predição no Caso de Rebatida

Neste caso, temos:

$$X_{\text{futuro}} = X_{\text{atual}} - \text{coef} \cdot \Delta x$$

$$Y_{\text{futuro}} = Y_{\text{atual}} + \text{coef} \cdot \Delta y$$

Estratégia *IF-THEN-ELSE*

Para explicar a metodologia utilizada na programação da estratégia, apresentam-se separadamente as rotinas que serão desenvolvidas para executar cada uma das seguintes ações:

- posicionamento dos robôs;
- colisão de jogadores;
- tirar o robô de situações de travamento nas bordas do campo (sair dos cantos);
- chute ao gol ou passe a bola;
- estratégias de defesa, ataque e goleiro (individuais para cada jogador).

Posicionamento dos robôs

A lógica de programação do algoritmo de posicionamento apresentada no fluxograma da Figura 11 será aplicado para cada robô, para definir a sua movimentação até o destino estabelecido. Será utilizada uma variável de tolerância (TOL) para garantir a aproximação e o adequado posicionamento do robô.

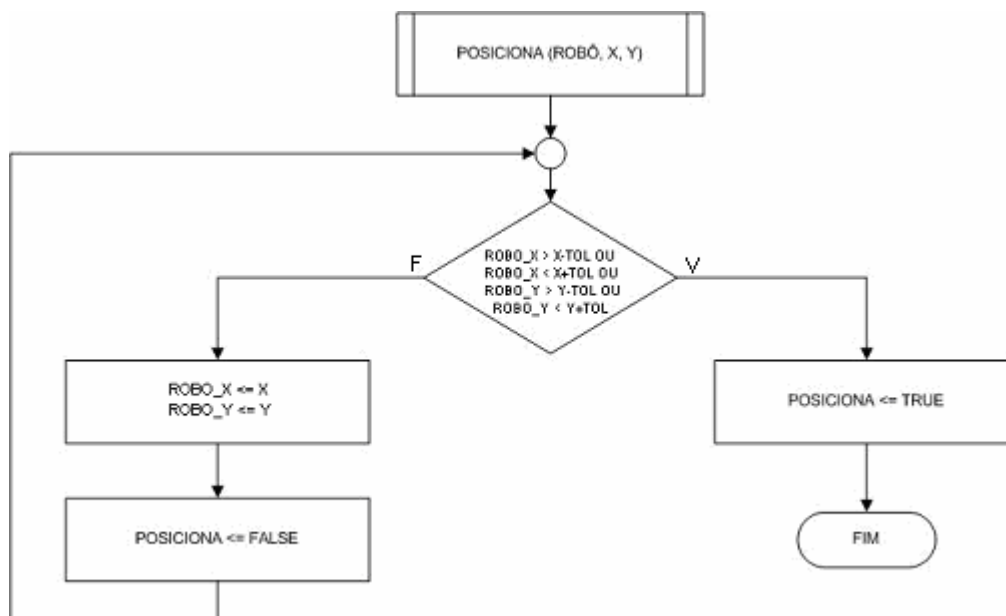


Figura 11 – Algoritmo de Posicionamento dos Robôs

Colisão de Jogadores (afasta robô)

O algoritmo de colisão foi desenvolvido para impedir a colisão entre jogadores. Utiliza-se como conceito a paralisação do robô quando ele se aproxima de uma área pré-definida delimitada com base nas coordenadas dos outros robôs da mesma equipe. A lógica desenvolvida e o fluxograma correspondente são ilustrados nas Figuras 12 e 13.



Figura 12 – Ilustração do Afastamento do Robô para Evitar Colisões

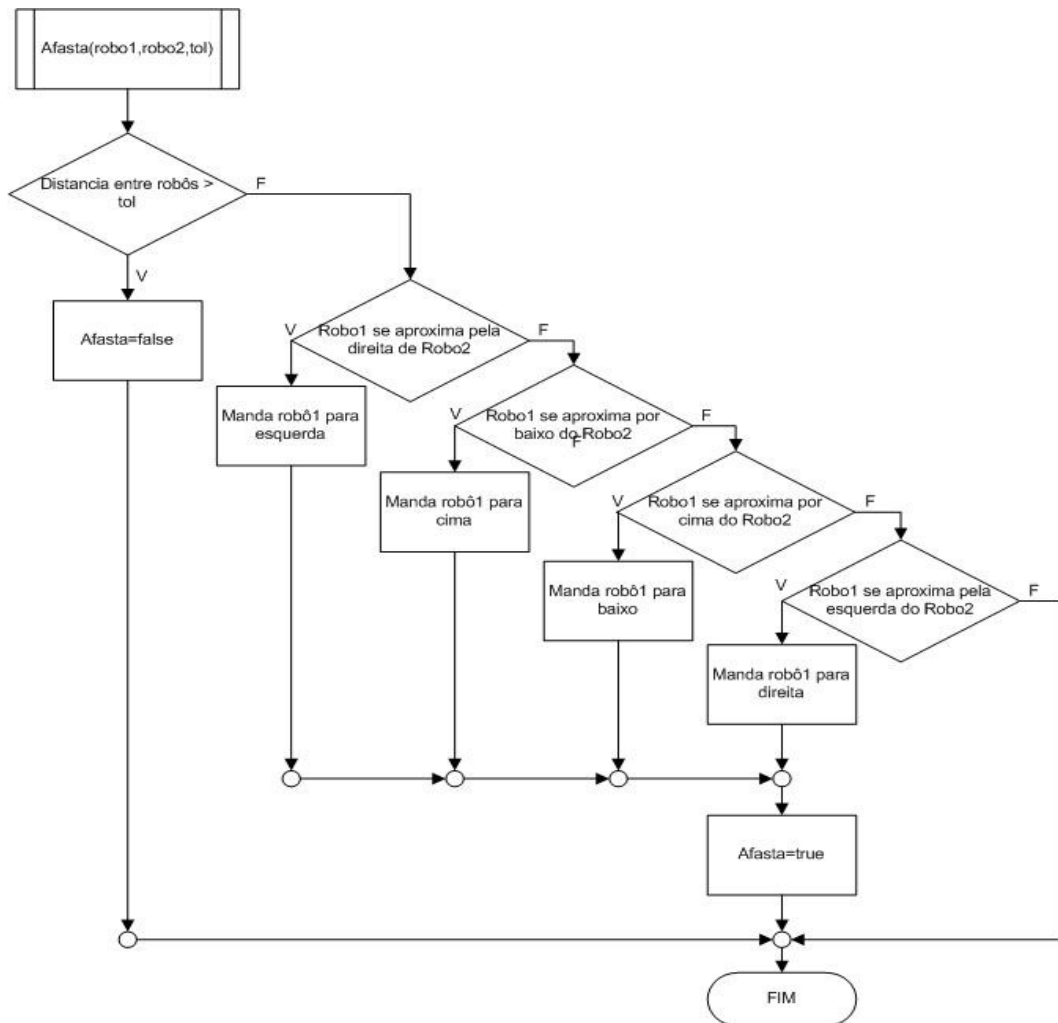


Figura 13 – Fluxograma da Rotina de Afastamento do Robô para Evitar Colisões

Situação de travamento (sai dos cantos)

Esse algoritmo (Figura 14) foi desenvolvido para afastar o jogador da parede quando ele está muito próximo das bordas do campo. Serão consideradas as seguintes situações: travamento próximo à linha do gol de ataque, travamento próximo ao limite inferior do campo (defesa), travamento próximo ao limite superior do campo (ataque) e travamento próximo à linha do gol de defesa.

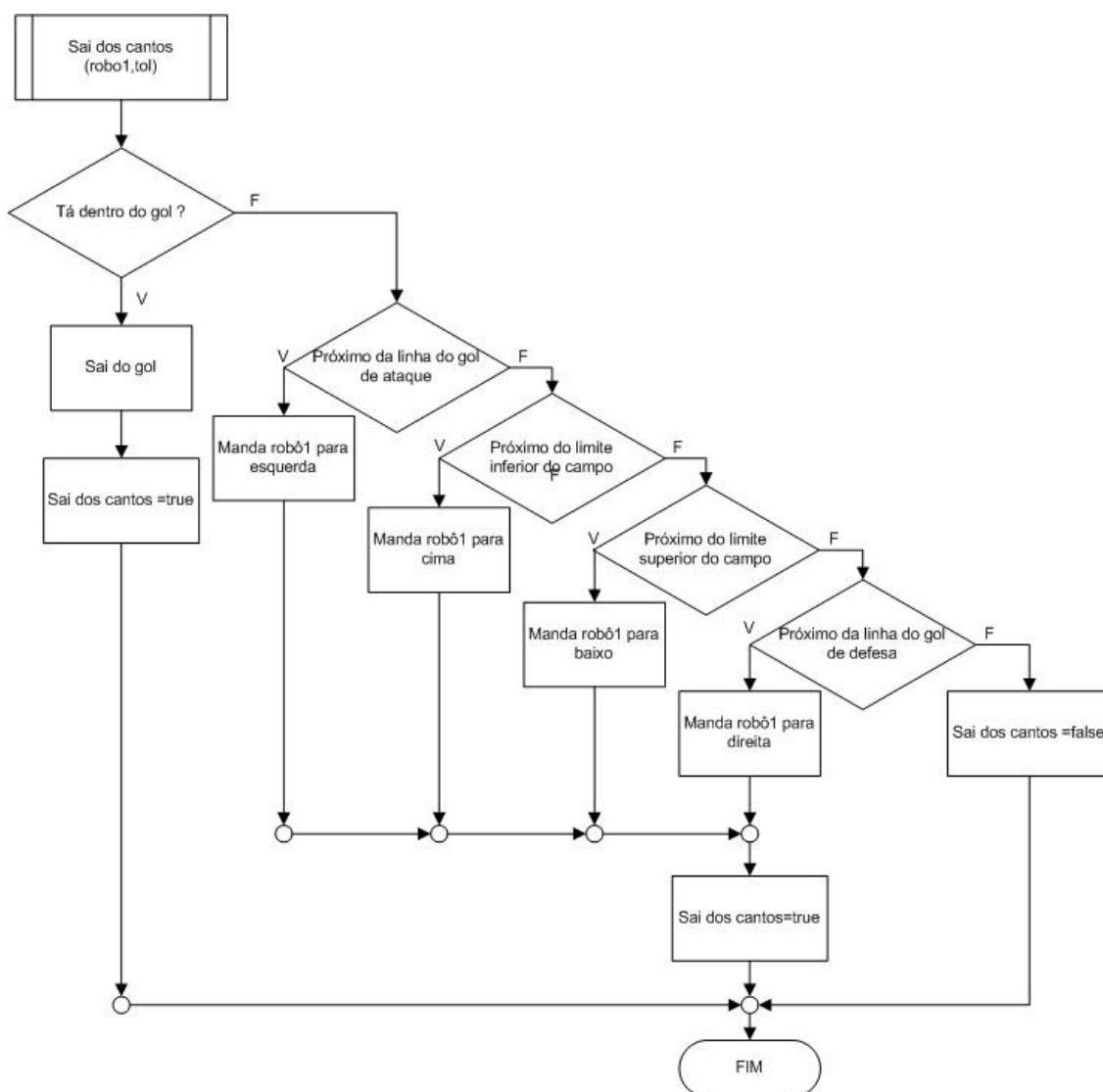


Figura 14 – Fluxograma da Rotina “Sai dos Cantos”

Chute ao Gol ou Passe

Esse algoritmo foi desenvolvido para movimentar o jogador para uma posição estratégica, direcionando-o para chutar e marcar o gol. Esse mesmo algoritmo serve para tocar a bola, por exemplo, do zagueiro para o atacante. O equacionamento foi elaborado dividindo-se a situação em quatro casos, como se mostra na Figura 15.

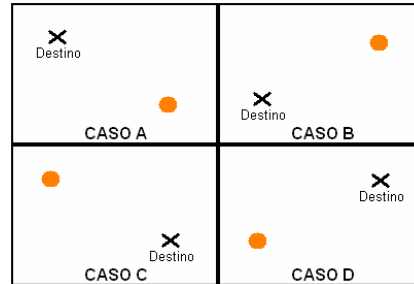


Figura 15 – Demonstração dos Casos considerados no Algoritmo de Chute ao Gol

Vamos estudar apenas o equacionamento do caso A (Figura 16) para demonstrar as equações, já que os outros casos são somente variações de sinais:

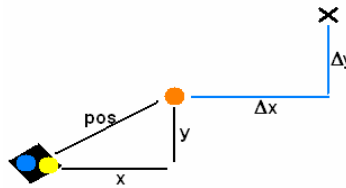


Figura 16 – Estudo do Caso A para Desenvolvimento do Algoritmo de Chute ao Gol

Por semelhanças de triângulos, temos:

$$\frac{\Delta x}{\Delta y} = \frac{x}{y}$$

$$y = x \cdot \frac{\Delta y}{\Delta x} \quad (1)$$

$$x = y \cdot \frac{\Delta x}{\Delta y} \quad (2)$$

A variável *pos* é uma constante definida pelo programador, e corresponde à distância com que o robô se posicionará atrás da bola antes de chutar.

$$x^2 + y^2 = pos^2 \quad (3)$$

Substituindo as equações (1) e (2) em (3) teremos:

$$y^2 \cdot \frac{\Delta x^2}{\Delta y^2} + y^2 = pos^2 \quad x^2 + x^2 \cdot \frac{\Delta y^2}{\Delta x^2} = pos^2 \quad (4)$$

e:

$$y = \frac{pos^2}{\sqrt{1 + \left(\frac{\Delta x}{\Delta y}\right)^2}} \quad x = \frac{pos^2}{\sqrt{1 + \left(\frac{\Delta y}{\Delta x}\right)^2}} \quad (5)$$

Logo se posiciona o robô, para o caso A, da seguinte maneira:

- no eixo y, o robô estará a y (mm) abaixo da bola;
- no eixo x, o robô estará a x (mm) atrás da bola.

Quando o robô estiver posicionado, ele irá chutar a bola. O fluxograma do algoritmo desenvolvido apresenta-se na Figura 17.

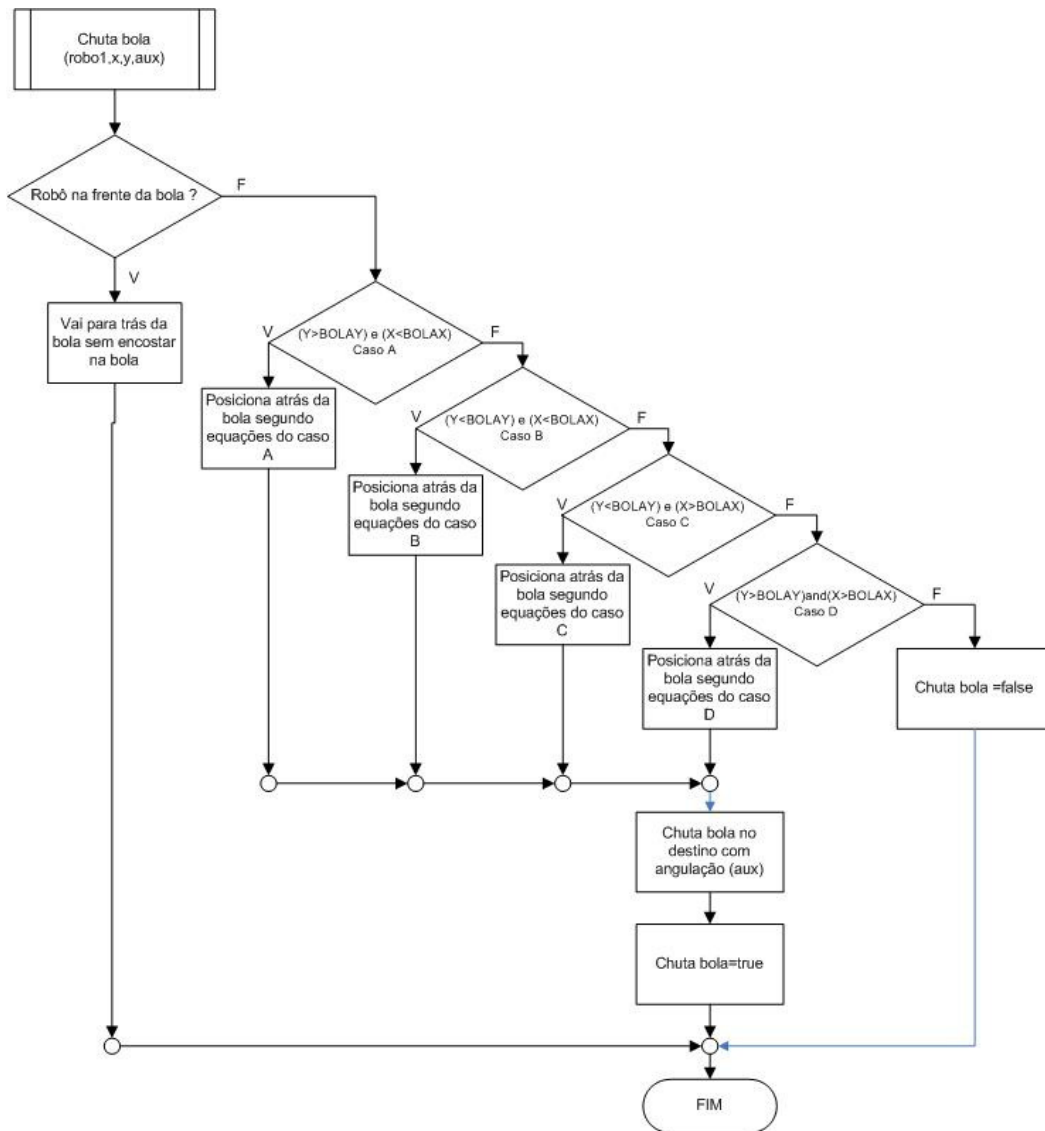


Figura 17 – Fluxograma da Rotina “Chuta Bola”

Algoritmos para Cada Jogador: Goleiro, Atacante e Defesa.

Esse algoritmo foi desenvolvido para definir a estratégia para cada robô da equipe. O goleiro obviamente exige uma estratégia totalmente diferente das outras. Para os robôs atacantes, e de forma similar para os defensores, os algoritmos terão um aspecto similar. Entre jogadores (atacante ou defesa) terá prioridade para movimentar-se em direção à bola aquele que tiver maior proximidade, exceto em situações especiais. Para cada um dos jogadores foram consideradas as seguintes situações:

Goleiro

- a bola está na frente ou atrás do goleiro;
- a bola está fora da área ou dentro da área;
- a bola está ao alcance do goleiro.

Defesa

- a bola está próxima deste jogador de defesa ou de outro jogador;
- a bola está atrás ou à frente do jogador de defesa;
- a bola está na área ou há perigo de gol do adversário;
- a bola está na posição de passe para o atacante ou chute ao gol.

Atacante

- a bola está mais próxima deste atacante ou de outro jogador;
- a bola está atrás ou à frente do atacante;
- a bola está ou não alinhada com o gol (está na posição de chute).

Os fluxogramas dos algoritmos do goleiro, atacante e defesa são apresentados nas Figuras 18, 19 e 20.

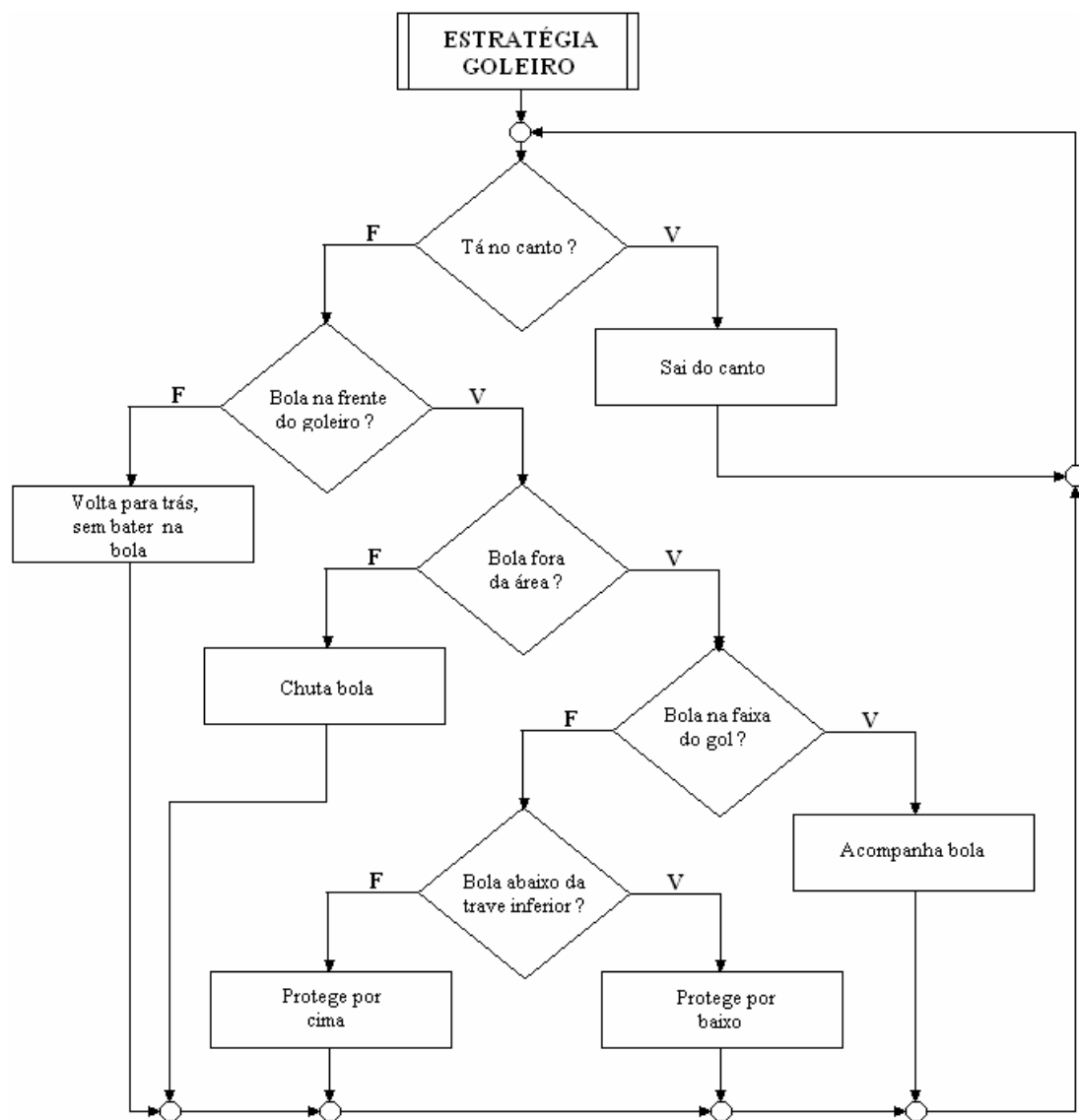


Figura 18 – Fluxograma do Algoritmo do Goleiro

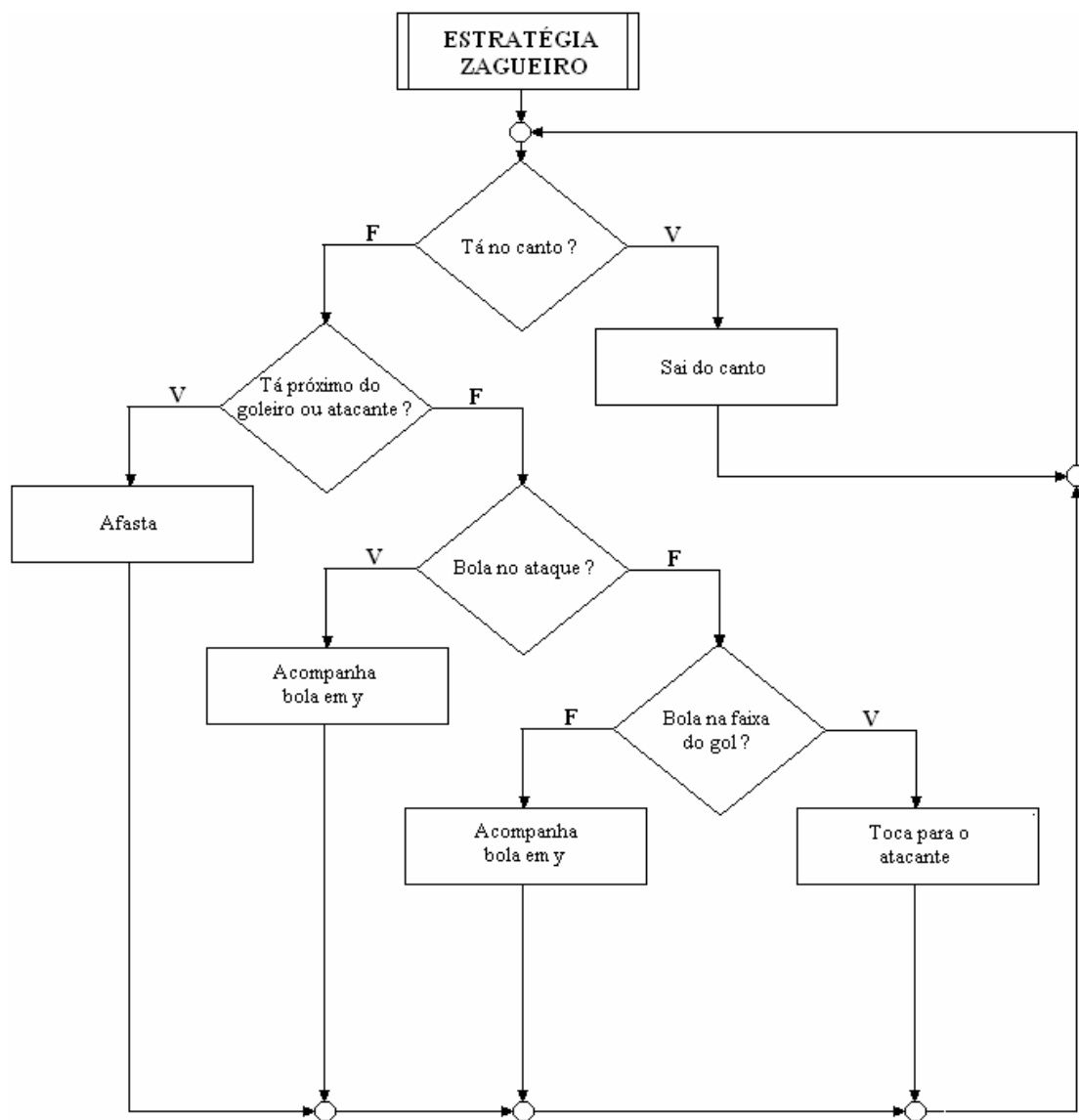


Figura 19 – Fluxograma do Algoritmo do Zagueiro (Defesa)

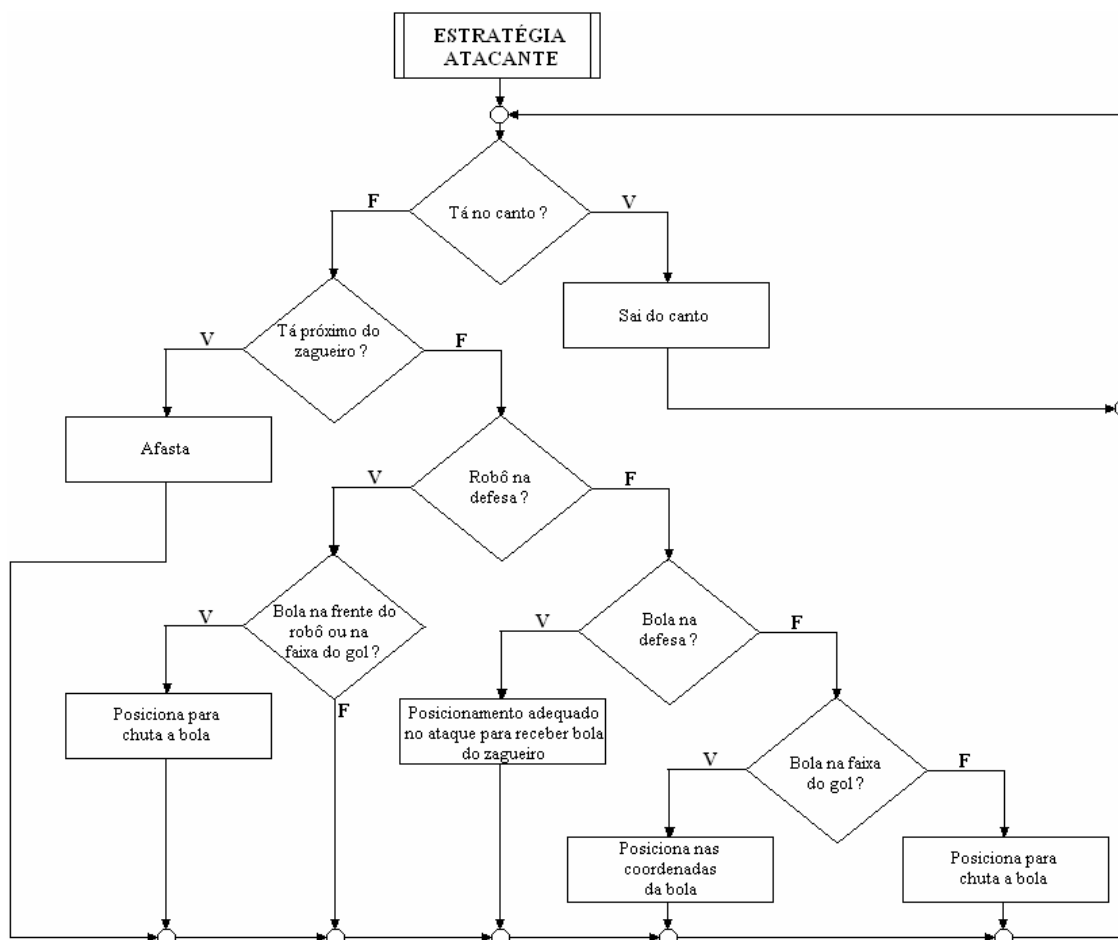


Figura 20 – Fluxograma do Algoritmo do Atacante

Definição das Tensões para Cada Motor e Transmissão dos Sinais de Controle

A última etapa do controle autônomo dos robôs jogadores de futebol consiste em transmitir os sinais de controle. Para isso utiliza-se um algoritmo que converte os comandos de velocidade e ângulo obtidos pelo algoritmo de estratégia para valores de controle para os robôs. Esses valores de controle correspondem à direção e tensão (referência para o PWM) para cada um dos motores de cada robô. Um protocolo de comunicação é então desenvolvido para transmitir os dados através da porta serial do computador. **Informações detalhadas sobre o protocolo de comunicação e os sinais de controle para os robôs podem ser obtidos no artigo “Construção de robôs jogadores de futebol – Parte 2” da edição n.º 29 da revista Mecatrônica Fácil (Editora Saber).**

3 – CONSIDERAÇÕES FINAIS

Várias estratégias adicionais são continuamente avaliadas e introduzidas para garantir melhor eficiência na estratégia de jogo. O projeto deve apresentar uma evolução constante, permitindo cada vez mais a avaliação de novas tecnologias e o desenvolvimento de novos algoritmos de controle.

De qualquer forma, o mais importante é que o projeto seja um incentivo ao aprendizado dos alunos e ao desenvolvimento da inteligência artificial e da programação em linguagem de alto nível.