

O Time Carrossel Caipira de Futebol de Robôs*

Rodrigo César Barboza Rossetti¹ Leandro Adriano dos Santos¹ João Vitor Mariano Correia¹
Tania Sanai Shimabukuro¹ João Pedro Olimpio¹ Renê Pegoraro¹ e Mateus Batista¹

Resumo—Este artigo apresenta aspectos de software do time Carrossel Caipira, que representa o Departamento de Computação da UNESP, campus de Bauru, na modalidade IEEE Very Small Size de futebol de robôs. Neste ano, como as competições serão realizadas em simulador, o software de controle é composto por um conjunto de módulos que inclui: interface com o simulador, estratégia e controle.

I. INTRODUÇÃO

O Departamento de Computação da Faculdade de Ciências da UNESP, campus de Bauru, participa de competições de futebol de robôs, na modalidade *very small size* (atualmente IEEE *Very Small*), desde 1998, com a realização do 1º Campeonato Brasileiro de Futebol de Robôs – CBRF 98. A pesquisa e o desenvolvimento em futebol de robôs, atualmente pelo Grupo de Integração de Sistemas e Dispositivos Inteligentes (GISDI), mantém o objetivo de incentivar o uso de inovações tecnológicas, no campo da robótica, de baixo custo e com componentes encontrados no mercado nacional. Apesar de diversos períodos de ausência, o time de futebol da UNESP de Bauru é conhecido, desde a primeira edição desta competição no Brasil, como Carrossel Caipira devido sua estratégia de jogo.

O ambiente real do futebol de robôs, nesta categoria, usa uma arena plana horizontal de 150cm x 130cm, onde robôs interagem, tentando conduzir a bola até o gol adversário e, ao mesmo tempo, defender o próprio gol de ataques. No ambiente, os robôs e a bola são identificados através de visão global usando uma câmera, posicionada a 2m sobre do campo e alinhada ao seu centro, que captura imagens do campo. Estas imagens são processadas digitalmente obtendo as coordenadas dos robôs e da bola. Com essas coordenadas, uma estratégia é escolhida e transformada em comandos que são enviados aos robôs por rádio. Os robôs recebem esses comandos e realizam as ações correspondentes, alterando o ambiente do futebol que será capturado novamente pela câmera. A Fig. 1 apresenta uma ilustração do ambiente do futebol de robôs. Durante as disputas presenciais, cada time normalmente participa com três integrantes, um “gerente” que coordena o time, um “técnico” que opera o computador interagindo com o software de controle e um “treinador” que manipula os robôs em campo. Porém, neste período

de pandemia, onde o contato social deve ser restrito, a permanência de três pessoas por time na área de jogo poderia ter consequências indesejadas.

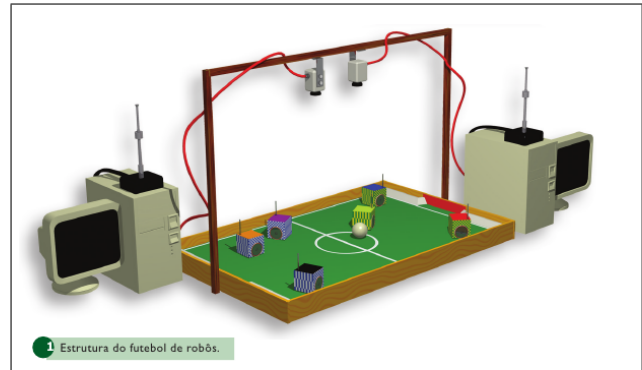


Fig. 1. Ambiente para futebol de robôs.

Para que a segurança dos participantes seja mantida, considerando a evolução da pandemia do novo coronavírus, a organização do *Latin American and Brazilian Robotics Competition* (LARC) manteve, neste ano, todas as modalidades no formato virtual. Assim, o time Carrossel Caipira foi adaptado para este cenário, fazendo que seu código antigo funcione e com a inclusão de algumas melhorias que poderão ser incorporadas ao ambiente com robôs reais para as competições que ocorrerão após esta pandemia.

II. SISTEMA DE SOFTWARE DO FUTEBOL DE ROBÔS

Para o robô locomover-se autonomamente o sistema precisa saber “responder” às seguintes questões: “onde está?”, “para onde vai?” e “como vai?”. Cada módulo do sistema “responde” às perguntas para conseguir a “resposta” e fazer com que o robô entenda o comando final.

O *software* que controla o time pode ser representado através do diagrama apresentado na Fig. 2 que indica as partes principais do processamento. Estas partes são descritas, juntamente com as interações entre elas. A CÂMERA captura e digitaliza uma imagem do campo a cada 33ms que é processada pelo módulo VISÃO, calculando, a partir das cores, as poses (x, y, ângulo) dos robôs e a da bola no campo. Com estas poses, o módulo ESTRATÉGIA determina as poses de destino onde os robôs do time devem estar para realizar a estratégia definida de defesa e ataque. O módulo CONTROLE, transforma as poses observadas dos robôs pelo módulo VISÃO e das poses de destino determinadas pelo módulo ESTRATÉGIA em comandos de acionamento das

*O time Carrossel Caipira foi apoiado pelo Santander Universidades e pela Agência Unesp de Inovação - AUIIN

¹Os autores são do Departamento de Computação, Faculdade de Ciências, Universidade Estadual Paulista “Júlio de Mesquita Filho” - UNESP, Bauru, SP, 17063-170, Brasil. (e-mails: rodrigo.rossetti@unesp.br, leandro.adriano@unesp.br, mariano.correia@unesp.br, tania.shimabukuro@unesp.br, jp.olimpio@unesp.br, rene.pegoraro@unesp.br, mateusbatistasantos@gmail.com)

rodas a serem enviados aos robôs via rádio, fazendo com que os robôs se movam para realizar a estratégia.

Com o objetivo de permitir o intercâmbio dos módulos ESTRATÉGIA e CONTROLE entre o ambiente simulado e o ambiente real, estes módulos usam interfaces para receber as informações de posicionamento dos robôs e bola e enviar comandos aos robôs. Uma representação de cada etapa pode ser vista na Fig. 2.

No ambiente real, Fig. 2, o módulo VISÃO processa as imagens e usa a interface de coordenadas padronizada para a passagem das poses ao módulo ESTRATÉGIA. Uma vez que o ESTRATÉGIA tenha executado, o CONTROLE assume, recebendo as poses e gerando os comandos de acionamento de cada roda dos robôs que são passados pela interface de comandos, de forma padronizada, ao módulo RÁDIO. Assim o módulo RÁDIO recebe as velocidades que devem ser enviadas a cada roda e converte para um único comando serial, que é enviado por um dispositivo de radiofrequência aos robôs.

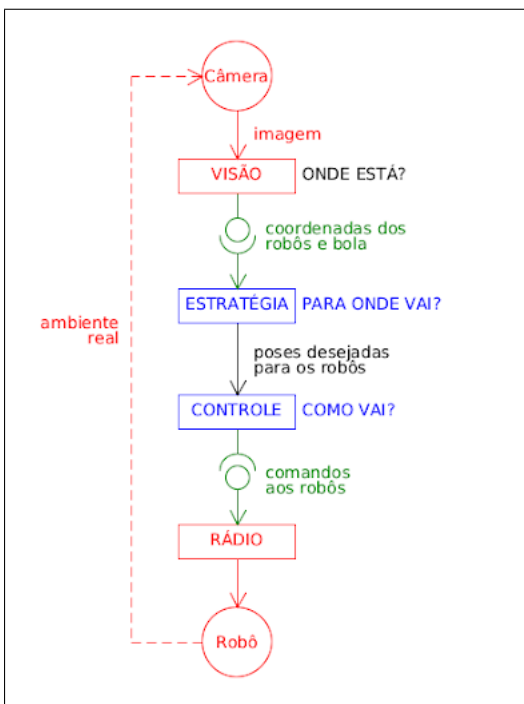


Fig. 2. Diagrama simplificado dos módulos do time Carrossel Caipira no ambiente real.

No ambiente simulado, Fig. 3, os módulos ESTRATÉGIA e CONTROLE, permanecem os mesmos, trocando dados através das mesmas interfaces de coordenadas e de comandos com o módulo CLIENTE que se conecta através de interfaces padronizadas pelo *protobuf*¹ com o SIMULADOR pela Internet.

Assim, o desenvolvimento realizado neste ano para o simulador será aplicado em competições futuras presenciais

¹Os *Protocol Buffers* do Google são mecanismos para serializar dados estruturados de forma padronizada independente da linguagem e da plataforma usadas. Disponível em <https://developers.google.com/protocol-buffers>. Acesso em 22 de setembro de 2020.

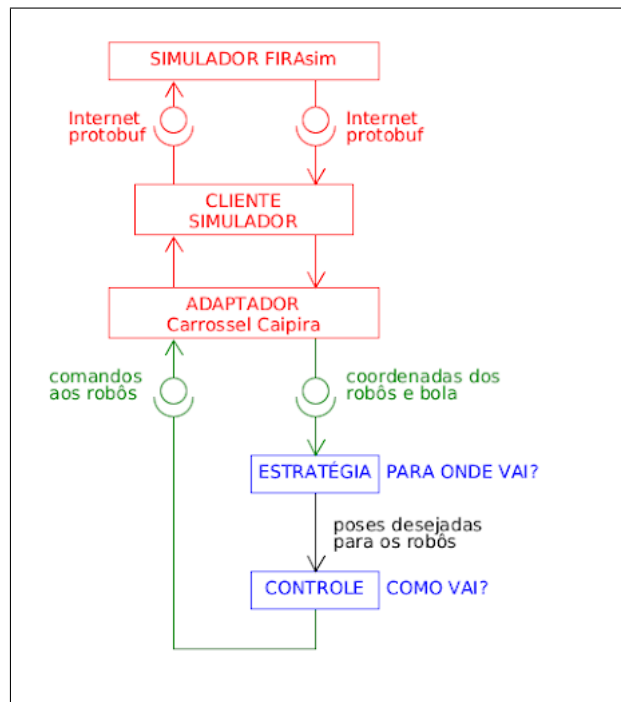


Fig. 3. Diagrama simplificado dos módulos do time Carrossel Caipira no ambiente simulado.

com poucos ajustes pois os módulos ESTRATÉGIA e CONTROLE são intercambiáveis sem alterações entre o ambiente real e o simulado.

A ausência de robôs reais e do processamento de imagens, que trazem desafios importantes e agregam muito ao aprendizado dos participantes desta modalidade, reduz as dificuldades na construção de um time, fazendo os esforços se concentrarem apenas nos módulos ESTRATÉGIA e CONTROLE. Estes módulos do time Carrossel Caipira são descritos a seguir.

A. Módulo ESTRATÉGIA

ESTRATÉGIA é o módulo responsável por definir os objetivos de cada robô. Utilizando as coordenadas atuais da bola e de todos os robôs em campo, este módulo decide qual é a posição mais indicada para cada robô do time, de tal forma que corresponda ao comportamento que os robôs devem desempenhar na partida.

O módulo ESTRATÉGIA foi desenvolvido com base no paradigma de programação orientada a autômatos – Os três comportamentos dos robôs: goleiro, volante e atacante são descritos por autômatos finitos determinísticos (AFD). Cada estado desses autômatos descreve um objetivo para os robôs, as arestas conectando os estados descrevem condições de troca de estados, e em cada instante um único estado deve estar ativo.

Estruturalmente esses três autômatos podem ser considerados “posições” dos robôs, representando super-estados, e a estratégia em si, uma máquina de estados hierárquica que descreve qual a posição ativa de cada robô, com arestas representando condições de troca de posições entre robôs.

Esses quatro AFDs são classes que herdam de uma classe abstrata que representa uma máquina de estados. Utilizando estruturas de seleção (*switches*) e enumerações, os autômatos são implementados proceduralmente através de três métodos herdados:

- *inicia()* – inicia o autômato, ativando um estado em específico.
- *transicao()* – verifica as condições das arestas e realiza transições do estado ativo.
- *acao()* – define o objetivo do robô para o estado ativo.

Os autômatos que descrevem os comportamentos dos robôs são apresentados a seguir.

1) **GOLEIRO:** Quando o jogo tem início, reinício ou a bola se encontra no campo de ataque, o goleiro deve posicionar-se no centro do gol. Caso ocorra um tiro de meta, o robô deve empurrar a bola em direção ao gol adversário. No cenário da bola descrever um movimento em direção ao gol defendido, o posicionamento deve ser feito no ponto (Xgol, Yprev) onde Xgol é a linha de gol e Yprev a estimativa em que a bola irá cruzar a linha de gol. Quando a bola se encontra no campo de defesa mas não descreve trajetória de encontro ao gol, o posicionamento deve ser feito na posição Y da bola caso o mesmo compreenda um valor no intervalo Y do gol, caso contrário posicionar-se no valor Y superior ou inferior da área, a depender da posição da bola. Estando a bola parada dentro da área o comportamento deve ser a fim de retirá-la de dentro da mesma. A Fig. 4 representa o comportamento descrito do goleiro.

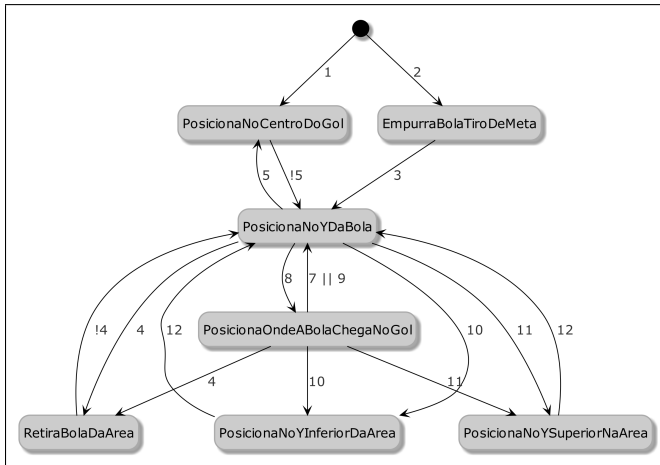


Fig. 4. AFD que descreve o comportamento do goleiro.

2) **VOLANTE:** Como representado na Fig. 5, o volante possui um comportamento híbrido de auxiliar o atacante e servir de linha de defesa contra avanços do time adversário. Com o início do autômato ele é posicionado no meio de campo e espera a bola mover-se. Se o atacante estiver avançando com a bola ele desempenha duas funções auxiliares, a depender: de suporte se estiver no lado adversário, se posicionando um pouco atrás da bola para recuperá-la caso a posse seja perdida, ou de bloqueio se estiver no nosso lado do campo, se posicionando um pouco à frente da bola para impedir que robôs adversários a recuperem, e nesse caso

quando se aproximar do gol adversário, ele sai do caminho para deixar o atacante avançar.

Se a bola passar a linha de defesa do volante, ele tenta recuar para recuperá-la, e se estiver com a bola à frente do atacante, passa para o ataque até que o atacante possa continuar avançando com ela. No caso da bola se encontrar dentro da nossa área de gol, ele se posiciona ao redor da área para defender e recuperar a bola quando ela eventualmente sair da área.

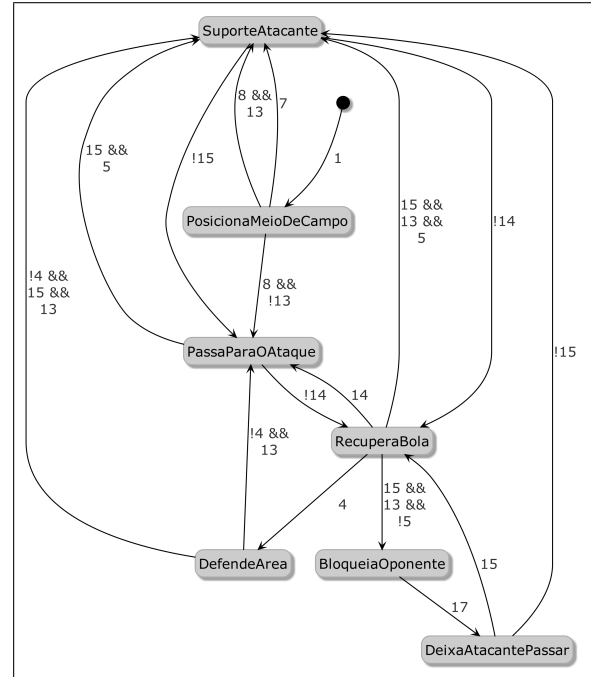


Fig. 5. AFD que descreve o comportamento do volante.

3) **ATACANTE:** A estratégia do atacante está formulada a não interferir no campo de defesa, ou seja, suas ações buscam o posicionamento no meio de campo superior ou inferior até que a bola esteja favorável para o ataque na frente do atacante que a partir disso irá se posicionar ofensivamente se alinhando com a bola, com a finalidade de maximizar as chances de gol.

O jogo se inicia com o atacante se direcionando ao meio de campo, se a bola se manter no campo de ataque, seus posicionamentos serão de alinhar com o Y da bola para a encontrar a posição que proporcione um chute direto ao gol e por fim realize o chute se posicionando no (X, Y) da bola.

Caso a bola recue do atacante ou se encontre no campo de defesa, o estado do autômato será o de posicionamento no campo superior ou inferior, sendo definidos de forma a facilitar o ataque assim que a bola for recuperada. A Fig. 6 representa o comportamento do robô atacante.

Os eventos que disparam as transições entre os estados, representados por números nas Fig. 4, Fig. 5 e Fig. 6, são relacionados com a suas descrições na Tabela I.

B. Módulo CONTROLE

Com as coordenadas detectadas pelo módulo de visão e pelas atribuídas pelos comportamentos do módulo de

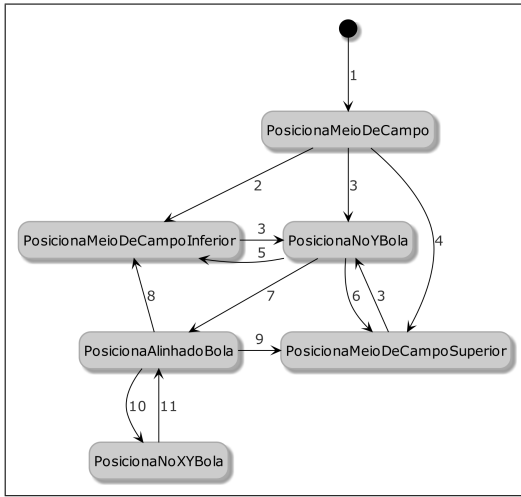


Fig. 6. AFD que descreve o comportamento do atacante.

TABELA I

CONDIÇÕES DE TROCAS DE ESTADOS UTILIZADAS PELA ESTRATÉGIA.

Condição	Descrição
1	Inicia o autômato, ativando um estado inicial
2	Condição de tiro de meta satisfeita
3	Fim do tiro de meta
4	Bola posicionada dentro da nossa área de gol
5	Bola posicionada dentro do campo de ataque
6	Bola posicionada no campo superior
7	Bola indo em direção ao gol adversário
8	Bola vindo em direção ao nosso gol
9	A bola não está em movimento
10	A bola chegará no nosso gol por baixo de fora da área
11	A bola chegará no nosso gol por cima de fora da área
12	A bola chegará no nosso gol pelo centro de fora da área
13	Atacante posicionado à frente do volante
14	Bola posicionada à frente do volante
15	Bola posicionada à frente do atacante
16	O atacante está alinhado com a bola
17	Volante posicionado próximo da área do gol adversário

estratégia, o módulo CONTROLE deve determinar as trajetórias e os melhores comandos a serem enviados aos robôs.

O cálculo de uma trajetória evitando os outros robôs é necessária para levar o robô da sua posição atual até a determinada pela estratégia, nestes casos o campo potencial pode ser empregado. Os campos potenciais partem da ideia de forças imaginárias atuando sobre o robô, ideia proposta por Khatib [1], onde a “força causada” pelos obstáculos é de caráter repulsivo e pela meta, de caráter atrativo. Porém a interferência das “forças” geradas a partir de vários obstáculos podem produzir ótimos locais que atrapalham o desempenho do sistema para encontrar um caminho até a meta para o robô.

Para evitar esta situação, Connolly et al. [2] solucionaram o problema utilizando funções harmônicas para o cálculo do campo potencial de ambientes nos quais as posições das paredes, objetos e metas sejam conhecidas, que é o caso no de futebol de robôs. As funções harmônicas utilizadas são soluções para a equação de Laplace.

Assim é definido um Problema de Valor de Contorno na região de atuação do robô utilizando a condição de Dirichlet² com potencial alto para obstáculos e potencial baixo para a meta. Então, são extraídas as linhas de força, com base no gradiente descendente [2] [3] do potencial, que direcionam o robô para sua meta, desviando de obstáculos.

Uma vez obtido o Campo Potencial, tem-se o ângulo ideal que o robô deve atingir para se deslocar até a meta, o ângulo objetivo, então calcula-se a direção e velocidade de cada motor.

Da mesma forma que os robôs reais, os robôs simulados dispõem de duas rodas de tração, utilizando o sistema de tração diferencial (*differential drive*). A tração diferencial consiste em ter o acionamento de cada uma das rodas do robô por comandos independentes o que permite manobras. Para manter a compatibilidade com os robôs reais do time, as velocidades enviadas aos robôs no simulador dispõem de 7 velocidades para frente e para trás.

Cada uma das faixas da Tabela II, indica qual motor deve ser acionado e em qual direção para que o robô realize o movimento desejado. Na tabela a diferença de ângulo representa a diferença entre o ângulo obtido do campo potencial na célula onde o robô está e o ângulo atual do robô. Para se obter velocidades maiores que 3, um fator é aplicado e por outro lado, se a velocidade precisar ser limitada, um valor é usado como saturação das velocidades. No caso do simulador é aplicado um fator para aumentar a velocidade até um valor desejado.

TABELA II

TRANSIÇÃO DOS ACIONAMENTOS DOS MOTORES.

Diferença de Ângulo	Vel. Motor Esq.	Vel. Motor Dir.
$0^\circ \leq \alpha < 5^\circ$	3	3
$5^\circ \leq \alpha < 31^\circ$	2	3
$31^\circ \leq \alpha < 57^\circ$	1	2
$57^\circ \leq \alpha < 83^\circ$	0	1
$83^\circ \leq \alpha < 90^\circ$	-1	1
$90^\circ \leq \alpha < 97^\circ$	1	-1
$97^\circ \leq \alpha < 123^\circ$	0	-1
$123^\circ \leq \alpha < 149^\circ$	-1	-2
$149^\circ \leq \alpha < 175^\circ$	-2	-3
$175^\circ \leq \alpha < 185^\circ$	-3	-3
$185^\circ \leq \alpha < 211^\circ$	-3	-2
$211^\circ \leq \alpha < 237^\circ$	-2	-1
$237^\circ \leq \alpha < 263^\circ$	-1	0
$263^\circ \leq \alpha < 270^\circ$	-1	1
$270^\circ \leq \alpha < 277^\circ$	1	-1
$277^\circ \leq \alpha < 303^\circ$	1	0
$303^\circ \leq \alpha < 329^\circ$	2	1
$329^\circ \leq \alpha < 355^\circ$	3	2
$355^\circ \leq \alpha < 360^\circ$	3	3

Na Tabela II, nota-se também velocidades negativas, pois o robô pode se movimentar para trás ou para frente, assim,

²A condição de contorno de Dirichlet (ou de primeiro tipo) é um tipo de condição de contorno, nomeada em homenagem a Johann Peter Gustav Lejeune Dirichlet (1805-1859). Quando aplicada sobre uma equação diferencial ordinária ou parcial, especifica os valores que uma solução necessita para tomar-se sobre o contorno do domínio.

dependendo da posição do destino em relação à pose atual do robô, ele pode se mover de ré para que seja feito o menor deslocamento possível para atingir a meta.

Depois de calculado os comandos de movimento de cada roda, eles são enviados através da interface de comandos para que o movimento seja realizado.

REFERENCES

- [1] KHATIB, O.; Command Dynamic dans l'Espace Opérationnel des Robots Manipulateurs en Présence d'Obstacles. PhD thesis, Ecole Nationale Supérieure de l'Aéronautique et de l'Espace, Toulouse. em Francês. 1980.
- [2] CONNOLLY, C. I., BURNS, J. B.; WEISS, R. Path planning using laplaces equation. In IEEE International Conference on Robotics and Automation, 1990.
- [3] CONNOLLY, C. I.; GRUPEN, R. A. On the application of harmonic functions to robotics. Journal of Robotic Systems, 10:931-946, 1993.