

DESENVOLVIMENTO DE UM TIME DE FUTEBOL DE ROBÔS VSSS

Téo Revoredo¹; Luis Felpe Monteiro da Fonte²

Resumo

Competições de robôs propõem desafios relevantes e com paralelo no mundo real que estimulam estudantes de diferentes níveis escolares a desenvolver soluções multidisciplinares que fomentem aprendizado e promovam pesquisa científica. No ensino de engenharia, o desenvolvimento de sistemas robóticos com foco em competições educacionais é prática interessante que potencializa trabalhos de iniciação científica e extensão universitária, especialmente. Nesse contexto, este trabalho apresenta o desenvolvimento de um time de futebol de robôs para competição na categoria IEEE VSSS 3x3, desde a concepção à realização de testes de desempenho, incluindo o projeto e a implementação dos elementos do jogo, circuitos eletrônicos, controladores de baixo e alto níveis e do sistema de visão computacional. Os resultados demonstram a funcionalidade do sistema e indicam direções de melhoria.

Palavras-chave: robôs móveis diferenciais; futebol de robôs; visão computacional.

Abstract

Robot competitions provide relevant challenges closely related to real world applications, stimulating students of different educational levels to develop multidisciplinary solutions that foster learning and promote scientific research. In engineering courses, the development of robotic systems focused on educational competitions is an interesting practice that enhances scientific initiation projects and university extension, in particular. In this context, this study presents the development of a robot soccer team for competition in the IEEE VSSS 3x3 category, from conception to performance testing, including the design and implementation of game elements, electronic circuits, low and high-level controllers, and the computer vision system. The results demonstrate the functionality of the system and indicate directions for improvement.

Keywords: differential robots; robot soccer; computer vision.

1 INTRODUÇÃO

Uma forma eficaz de promover pesquisa científica e estudos de engenharia é estabelecer desafios a serem suplantados a longo prazo, algo comprovado, entre outros, pelo estabelecimento de diversos eventos educacionais na área da robótica, que muitas vezes envolvem competições de robôs com objetivos diversos. Construir robôs que jogam futebol, por exemplo, por si só não produz impacto social e econômico significativo, mas as realizações alcançadas ao longo do seu desenvolvimento levam a grandes conquistas na área. Entre as categorias de futebol de robôs, uma é denominada IEEE Very small size soccer (VSSS), na qual

¹ Doutor em Engenharia Elétrica pelo Instituto Alberto Luiz Coimbra de Pós-Graduação e Pesquisa de Engenharia-COPPE da Universidade Federal do Rio de Janeiro-UFRJ, com doutorado sanduíche realizado na École Nationale de l'Aviation Civile-ENAC, Toulouse-França. É professor adjunto no Departamento de Engenharia Eletrônica e Telecomunicações (DETEL) e Pesquisador no Programa de Pós-Graduação em Engenharia Eletrônica (PEL) da Universidade do Estado do Rio de Janeiro (UERJ). E-mail: teorevoredodo@uerj.br.

² Mestrando do Programa de Pós-Graduação em Engenharia Eletrônica (PEL) da Universidade Estadual do Rio de Janeiro-UERJ. E-mail: luisfelipefonte@gmail.com.

duas equipes de 3 robôs autônomos jogam uma partida comandados por um sistema computacional externo que processa imagens de uma câmera posicionada acima do campo (IEEE, 2025). O seu desenvolvimento envolve, portanto, diferentes áreas da engenharia tais como eletrônica, instrumentação, controle, processamento de sinais e telecomunicações, entre outros. Nesse contexto, este trabalho apresenta o desenvolvimento de um time de futebol de robôs capaz de competir nesta categoria, da sua concepção à realização de testes práticos de desempenho, incluindo a construção dos robôs e o projeto e a implementação dos circuitos eletrônicos, dos controladores de baixo e alto níveis e do sistema de visão computacional.

O trabalho é organizado da seguinte maneira: Na Seção 2 são apresentadas as premissas básicas de projeto com base nas regras que regem a competição de futebol VSSS, na Seção 3 são abordados detalhes da especificação e da fabricação dos robôs, além dos sistemas de comunicação e visão computacional, bem como a interface com o usuário. Na seção 4 é apresentada a modelagem matemática dos robôs e na Seção 5 o algoritmo de controle empregado para a movimentação autônoma. Resultados de validação experimental são discutidos na Seção 6, seguidos das considerações finais sobre o trabalho.

2 PREMISSAS BÁSICAS DE PROJETO

As premissas básicas de projeto são oriundas das regras definidas para a categoria IEEE VSSS de futebol de robôs (CBR, 2025), que incluem um campo de jogo, uma bola, 3 jogadores de cada time, uma ou mais câmeras e sistemas de processamento e transmissão de sinais. São especificadas as dimensões e cores do campo, da bola e dos jogadores para facilitar a identificação dos elementos durante uma partida. A bola deve ter cor laranja com aproximadamente 42,7mm de diâmetro e 46g de massa. Cada robô deve possuir dimensões máximas de 75x75x75mm, podendo vestir uniformes, desde que não ultrapassem 80x80x80mm. É permitido que cada jogador possua um mecanismo para manusear a bola, desde que 70% do seu diâmetro seja visível. Na parte superior dos jogadores são posicionadas etiquetas que identificam o seus time e número, servindo como referências das suas poses para o algoritmo de visão computacional. A Figura 1 ilustra as etiquetas construídas neste trabalho. O retângulo maior identifica os times e os quadrados informam a numeração do jogador. As etiquetas foram fabricadas usando impressora 3D.

Figura 1 - Etiquetas para identificação dos robôs.



Fonte: Elaborada pelos autores.

O campo deve ser fabricado em madeira plana com dimensões de 150x100cm e pintado nas cores preto e branco. Neste trabalho, além de um campo com medidas oficiais, foi construído também um conjunto experimental portátil para realizar ensaios em menor escala. A Figura 2 apresenta as estruturas experimentais desenvolvidas.

Figura 2 – Campo de jogo e estrutura portátil para testes



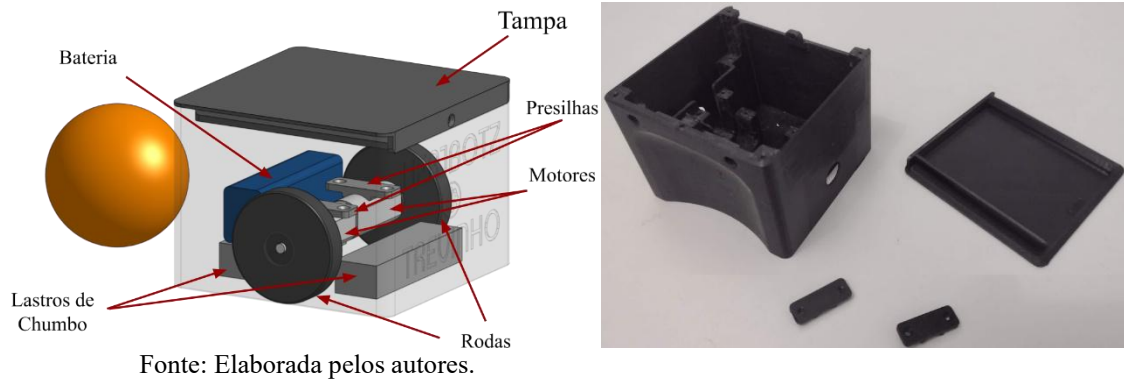
Fonte: Elaborada pelos autores.

3 FABRICAÇÃO DOS ROBÔS

A fabricação dos robôs envolve alguns desafios: montagem compacta limitada a 70 x 70 x 70 mm, velocidade e força para alcançarem rapidamente a bola e vencer disputas pela mesma, comunicação e processamento dos sinais de forma eficiente. Para superá-los adota-se o sistema de locomoção diferencial, que é compacto e tem boa manobrabilidade, permitindo movimentação em qualquer direção. São usados micro motores escovados de corrente contínua (CC) de ímãs permanentes (N20) com caixa de redução, que possuem tamanho reduzido e alta capacidade de torque, além de serem facilmente modelados por sistemas lineares. O modelo adotado opera com 6V de tensão e 500 RPM de velocidade nominal. As partes mecânicas foram

projetadas e impressas em 3D por método de fabricação aditiva. A Figura 3 ilustra as peças impressas e a montagem dos robôs.

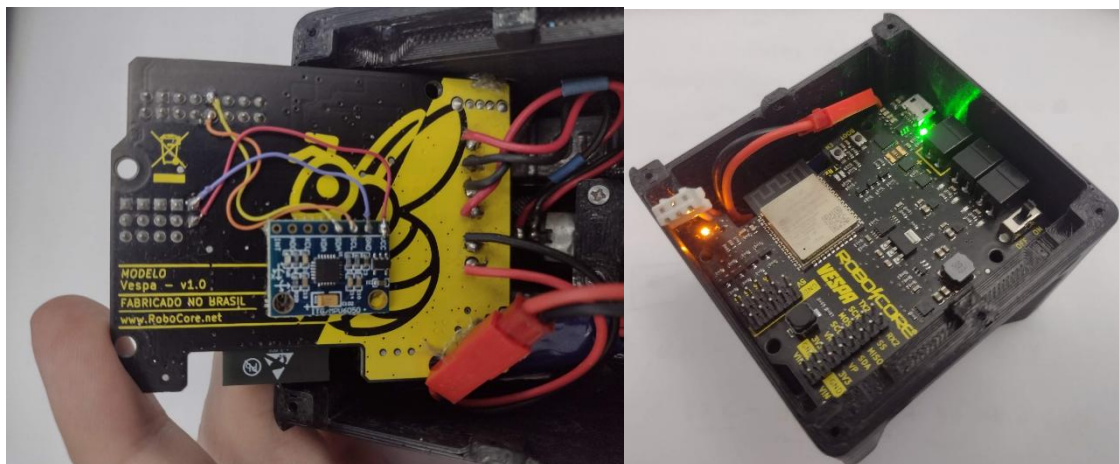
Figura 3 - O robô e suas peças



Fonte: Elaborada pelos autores.

O sistema eletrônico é composto por uma placa “vespa” (Robocore, 2025) que possui saída para dois motores CC e o microcontrolador ESP32 (Espressif, 2025) que além de processar os sinais também recebe sinais de 2.4GHz, um módulo MPU6050 (TDK InvenSense, 2023) de unidade de medição inercial (IMU) fixado com fita dupla-face na placa. A IMU mede a velocidade angular dos robôs, sendo utilizada no controle local dessa grandeza. A Figura 4 ilustra a montagem. A alimentação do sistema é feita com baterias de Polímero de Lítio (LiPo) 2S de 300mAh, dimensionadas para atender aos requisitos de velocidade máxima e duração de dois tempos de jogo com folga.

Figura 4 - Montagem final de um robô



Fonte: Elaborada pelos autores.

3.1 Sistema de comunicação

O computador principal se comunica com os robôs para enviar as velocidades de referência de cada um. A comunicação é realizada usando o protocolo ESPNOW, que permite o pareamento direto entre diversos ESP8266 e/ou ESP32, para envio e recebimento de dados.

Em comparação com outros protocolos como bluetooth e WiFi, o ESPNOW é mais fácil de implementar, com baixa latência e rápido restabelecimento de conexão em casos de desligamento do transmissor ou receptor. A codificação utiliza a biblioteca ESPNOWSerial que permite o uso de comandos de texto legíveis por humanos. A Tabela 1 lista os principais comandos.

3.2 Visão computacional e interface

O algoritmo de visão computacional obtém a configuração dos jogadores e da bola e é implementado em linguagem Python utilizando as bibliotecas OpenCV (Brahmbhatt, 2013) e Numpy (Oliphant, 2006.), devido à sua sintaxe concisa, compatibilidade com portas seriais e ampla disponibilidade de recursos para processamento de imagem e operações algébricas. O código final está disponível em repositório aberto (Fonte, 2024). A execução do algoritmo ocorre nas seguintes etapas: (1) captura; (2) segmentação das cores; (3) detecção de contornos; (4) obtenção das poses dos elementos e encapsulamento.

Tabela 1 – Lista de comandos dos robôs.

Comando	Função
reset	reinicia o robô
bot.stop	freia o robô
bot.move <esq> <dir>	aplica as velocidade em cada roda
pid.on <estado>	liga ou desliga o controlador PID
pid.linear_speed <PWM>	altera a velocidade tangencial
pid.angular_speed <rad/s>	altera o valor da velocidade angular

Fonte: Elaborada pelos autores.

A captura de imagens é feita por uma webcam USB. Embora câmeras de maior velocidade sejam recomendadas, nenhuma estava disponível na realização deste trabalho, o que se apresentou como um desafio interessante a ser suplantado. O processo de aquisição de quadros demanda alguns milissegundos, o que poderia comprometer o desempenho geral do sistema. Para mitigar esse problema, adota-se uma abordagem multi-thread, na qual um processo realiza a captura enquanto outro processa o quadro anterior. A taxa de captura alcançada varia entre 21 e 37 FPS.

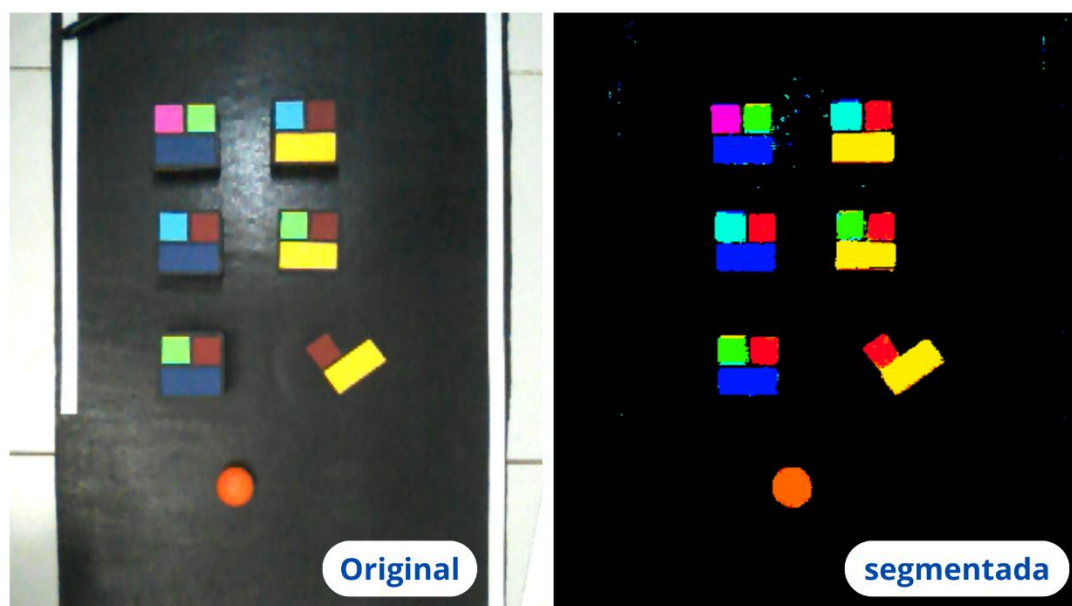
Segmentação por cores permite identificar e diferenciar os elementos do jogo. A bola é indicada exclusivamente pela cor laranja, enquanto as cores amarela e azul distinguem os jogadores e seus respectivos times. Outras cores são usadas para identificar a numeração e a orientação global de cada jogador. A segmentação é feita no espaço de cor HSV (Hue, Saturation, Value) (Hema; Kannan, 2019; Mohd Ali; Md Rashid; Mustafah, 2013), pois o canal

H (matiz) facilita a distinção entre cores. Neste trabalho, os parâmetros HSVmin e HSVmax de cada cor são definidos empiricamente, sendo ajustados conforme as condições de iluminação, tonalidade das cores e características da câmera, uma vez que diferentes modelos apresentam variações na detecção. A Figura 5 apresenta um exemplo de segmentação aplicada.

A segmentação dos objetos é realizada por meio da detecção de contornos, definidos como as bordas de regiões com coloração uniforme. Na biblioteca OpenCV, esse processo é conduzido pela função *findContours*, que retorna os contornos detectados em uma imagem binária. Após a extração, os contornos são ordenados conforme sua área, sendo descartados aqueles que não atingem um limiar mínimo, a fim de eliminar ruídos e elementos irrelevantes.

Para identificar a posição da bola, o maior contorno na cor laranja é aproximado por uma circunferência usando a função *minEnclosingCircle*, que fornece o centro e o raio do menor círculo que contém o contorno. Já as etiquetas são aproximadas por retângulos de área mínima, por meio da função *minAreaRect*, que retorna os vértices do retângulo orientado. A partir desses vértices, calcula-se o centro geométrico conforme a Equação (1).

Figura 5 - Exemplo de segmentação das cores



Fonte: Elaborada pelos autores.

Embora existam métodos mais sofisticados para estimar a posição e a orientação dos objetos, tais como técnicas baseadas em aprendizado de máquina (Ortenzi, *et al.*, 2018; Lee, *et al.*, 2020; Castelli, *et al.*, 2017) a abordagem baseada em contornos simples é eficaz e computacionalmente eficiente para os fins deste trabalho.

$$\vec{P}_{centro} = \sum_i^n \frac{(x_i, y_i)}{4} \quad (1)$$

Na Equação (1), $\vec{P}_{centro}$ é o centro de uma etiqueta, i representa o índice de cada vértice e (x_i, y_i) são as coordenadas de cada vértice. As posições e orientações dos robôs são determinadas como segue:

i. Identificam-se as três maiores figuras nas cores dos times, excluindo-se as filtradas por área, que servem como referência inicial para a posição dos jogadores;

ii. Com base nas dimensões da etiqueta, estima-se as posições onde as menores deveriam estar;

iii. Busca-se etiquetas menores nas cores permitidas próximas das posições estimadas. O conjunto final é selecionado com base no arranjo que obtiver a menor soma de erros entre as posições reais e estimadas. Se a soma for elevada, o arranjo contém apenas a etiqueta do time. As menores agrupadas não estão disponíveis para a próxima etiqueta de time a ser processada;

iv. Se o arranjo possuir etiquetas menores, o centro do robô é calculado pela Equação (2), na qual $\overrightarrow{T1}$, $\overrightarrow{E1}$ e $\overrightarrow{E2}$ são as coordenadas dos centros, sendo $\overrightarrow{T1}$ referente a etiqueta de time. Se o arranjo possuir somente a etiqueta de time, seu centro é considerado o do robô;

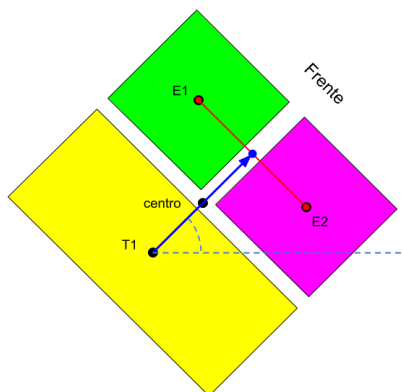
$$\vec{P} = \frac{2\overrightarrow{T1} + \overrightarrow{E1} + \overrightarrow{E2}}{4} \quad (2)$$

v. A orientação do robô é dada pelo ângulo θ do vetor $\vec{v}$ de acordo com as Equações (3) e (4). O vetor $\vec{v}$ aponta para a face do robô e caso o arranjo só possua a etiqueta de time ele é definido como uma das laterais menores da etiqueta. A Figura 5 ilustra o vetor de distância e os centros das etiquetas.

$$\theta = \arctan\left(\frac{v_y}{v_x}\right) \quad (3)$$

$$\vec{v} = \overrightarrow{E1} + \overrightarrow{E2} - 2\overrightarrow{T1} \quad (4)$$

Figura 6 - Vetor de distância e centros das etiquetas.

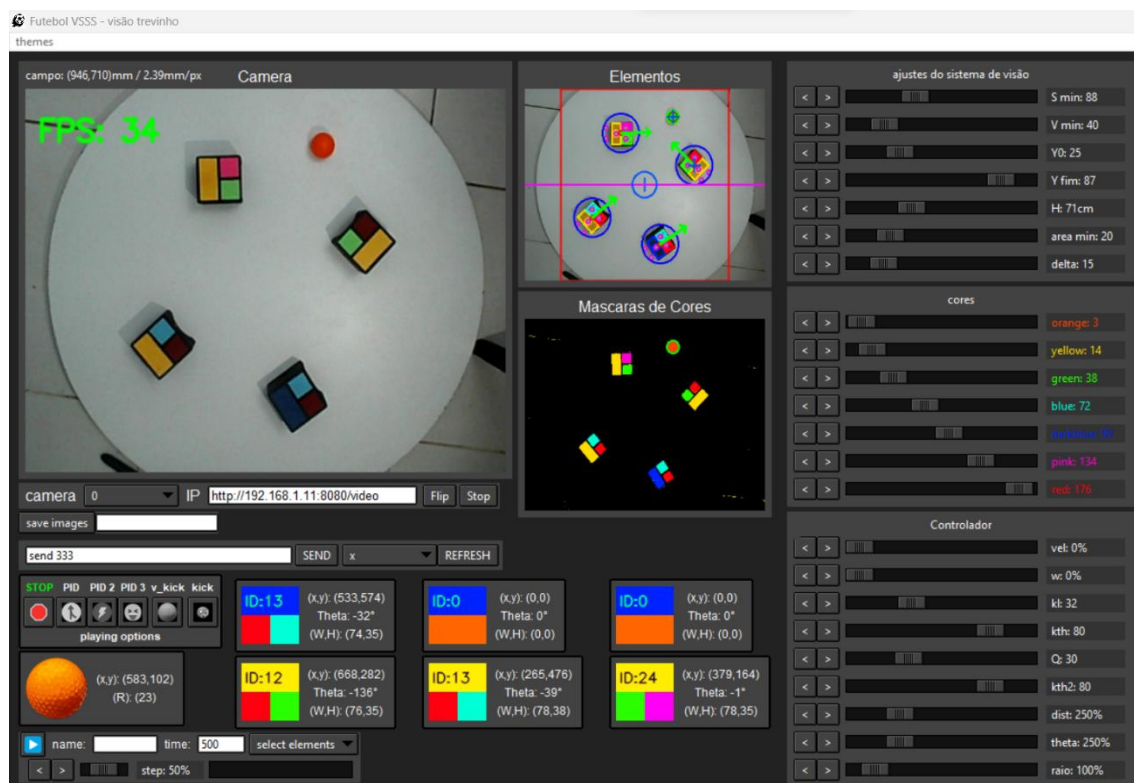


Fonte: Elaborada pelos autores.

3.3 Organização do código e interface com o usuário

O sistema é estruturado em módulos, cada um responsável por uma funcionalidade específica. O módulo Transmissão contém os métodos de comunicação com os robôs. O módulo Visão é responsável pelo processamento dos quadros de imagem e pela obtenção das configurações da bola e dos jogadores. O módulo Controle calcula os sinais de comando para cada robô com base nas poses detectadas e na estratégia de jogo definida. Já o módulo Interface utiliza a biblioteca Tkinter para a construção da interface gráfica com o usuário, além de agrupar outras funcionalidades auxiliares. A interface permite monitorar as etapas de processamento, ajustar os parâmetros do sistema de visão em tempo real e realizar ações como a conexão com os robôs e a aquisição de dados para análises posteriores. Um exemplo da interface é mostrado na Figura 7.

Figura 7 - Interface gráfica do sistema.



Fonte: Elaborada pelos autores.

4 MODELAGEM MATEMÁTICA

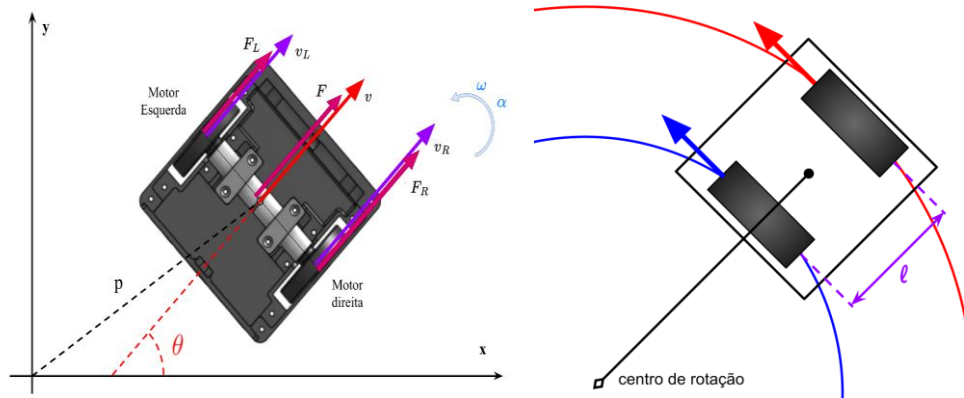
Os jogadores usam sistema de locomoção diferencial, de acordo com as Equações (5) e (6), nas quais l representa a distância entre as rodas e $v_{dir}(t)$ e $v_{esq}(t)$ correspondem às velocidades lineares das rodas direita e esquerda, respectivamente.

$$v(t) = \frac{v_{dir}(t) + v_{esq}(t)}{2} \quad (5)$$

$$\omega(t) = \frac{v_{dir}(t) - v_{esq}(t)}{l} \quad (6)$$

A Figura 8 ilustra as principais grandezas cinemáticas que atuam sobre o robô, bem como o centro instantâneo de rotação (CIR) e as trajetórias individuais percorridas por cada roda em uma curva.

Figura 8 - Forças, velocidades e aceleração angular



Fonte: Elaborada pelos autores.

Manipulando-se as Equações (5) e (6), obtêm-se as Equações (7) e (8), que descrevem as velocidades de cada roda em termos das velocidades linear $v(t)$ e angular $\omega(t)$ desejadas para o robô:

$$v_{dir}(t) = v(t) + \frac{l}{2} \omega(t) \quad (7)$$

$$v_{esq}(t) = v(t) - \frac{l}{2} \omega(t) \quad (8)$$

A atuação sobre as rodas é realizada por motores de corrente contínua cuja dinâmica elétrica é representada pela Equação (9):

$$V_A(t) = R_A i(t) + L_A \frac{di(t)}{dt} + V_{ind}(t) \quad (9)$$

Onde, R_A e L_A representam, respectivamente, a resistência e a indutância do enrolamento de armadura, enquanto $V_{ind}(t)$ corresponde à tensão induzida pela rotação do rotor. Em condições normais de operação, essa tensão induzida é proporcional à velocidade angular do rotor $\omega(t)$ como segue:

$$V_{ind}(t) = \omega(t)k \quad (10)$$

A constante k depende de parâmetros físicos do motor, como o material do núcleo da armadura, número de espiras, geometria do rotor e intensidade do campo magnético. Essa constante também expressa a relação entre o torque $T(t)$ e a corrente consumida $i(t)$:

$$T(t) = ki(t) \quad (11)$$

Somando-se as forças e os momentos gerados pelas rodas, obtêm-se as Equações (12) e (13), que descrevem a dinâmica translacional e rotacional do robô:

$$ma = F_{dir} + F_{esq} \quad (12)$$

$$J\alpha = 2l(F_{dir} - F_{esq}) \quad (13)$$

Aplicando a transformada de Laplace e manipulando as expressões anteriores, é possível obter as forças de tração $F_{dir}(s)$ e $F_{esq}(s)$ em função da frequência complexa s e da tensão aplicada $V(s)$, conforme as Equações (14) e (15):

$$F_{dir}(s) = \frac{s(mV(s) + \frac{J}{l}\Omega(s))}{2} \quad (14)$$

$$F_{esq}(s) = \frac{s\left(mV(s) - \frac{J}{l}\Omega(s)\right)}{2} \quad (15)$$

Os torques resultantes atuantes sobre as rodas são obtidos multiplicando-se essas forças pelo raio das rodas, r .

5 CONTROLE E MOVIMENTAÇÃO AUTÔNOMA

Cada robô é programado para mover-se de forma autônoma, com base em trajetórias planejadas no sistema central e controladas localmente. Para isso, é implementado um sistema de controle distribuído em dois níveis. O primeiro, executado no computador central, utiliza as informações do sistema de visão para calcular trajetórias desejadas e sinais de controle correspondentes. O segundo nível, embarcado em cada robô, é responsável por regular a velocidade angular com maiores frequência e exatidão.

O controle embarcado é feito por um controlador Proporcional-Integral-Derivativo (PID), cuja referência é a velocidade angular desejada em rad/s. Essa velocidade é medida em tempo real por meio do giroscópio da unidade inercial (IMU) presente no robô. Para reduzir os efeitos de ruído no sinal, aplica-se um filtro baseado na Equação (16), com um processo de calibração inicial no qual o robô permanece estático por alguns segundos. Durante esse período, registra-se uma média para definir o ponto zero, e o maior valor absoluto é utilizado como limiar mínimo (definido como 0,02 rad/s). Valores abaixo desse limiar são considerados nulos, minimizando variações indesejadas.

$$\omega = \{\omega - \omega_0, \text{ se } |\omega - \omega_0| \geq \omega_{min}, 0 \text{ caso contrário} \quad (16)$$

A saída do controlador PID é usada para gerar um diferencial de velocidade entre as rodas, corrigindo a orientação do robô. Os sinais modulados em amplitude (PWM) enviados aos motores são:

$$PWM_{dir} = PWM_{medio} + 0,5PWM_{pid} \quad (17)$$

$$PWM_{esq} = PWM_{medio} - 0,5PWM_{pid} \quad (18)$$

O valor PWM_{medio} controla a velocidade linear do robô em malha aberta e as magnitudes são limitadas ao valor máximo permitido pelo sistema de acionamento, evitando saturações que poderiam comprometer a resposta do sistema.

A movimentação autônoma é determinada por um algoritmo de campos potenciais artificiais (Panagou, 2014; Masehian, 2007), adotado por seu baixo custo computacional em comparação a técnicas mais complexas, como árvores aleatórias de exploração rápida (RRT) (Connell, 2017). Neste método, obstáculos exercem forças repulsivas, enquanto o objetivo gera uma força atrativa. A resultante dessas forças, determinada pela Equação (19), define a direção desejada de movimento, cujo ângulo é calculado pela Equação (20):

$$\vec{F}(x, y) = \sum_i^{cargas} \frac{Q_i}{(x_i - x)^2 + (y_i - y)^2} \quad (19)$$

$$\theta = \arctg\left(\frac{F_x(x, y)}{F_y(x, y)}\right) \quad (20)$$

Embora o modelo cinemático defina o comportamento ideal dos robôs, na prática ocorrem desvios devido a diversos fatores, como inércia, desalinhamento do centro de massa, variações entre motores e diferenças de atrito na superfície. Esses fatores tornam o controle em malha aberta inexato, especialmente considerando a limitação na taxa de atualização imposta pelo sistema de visão computacional. A inclusão de um controlador embarcado resolve esse problema, permitindo respostas mais rápidas e exatas. Assim, o controlador de alto nível pode ser simplificado utilizando apenas ações proporcionais com base nos erros de posição e orientação, conforme as Equações (21) e (22):

$$v = k_v erro_d \quad (21)$$

$$\omega = k_{ang} erro_{ang} \quad (22)$$

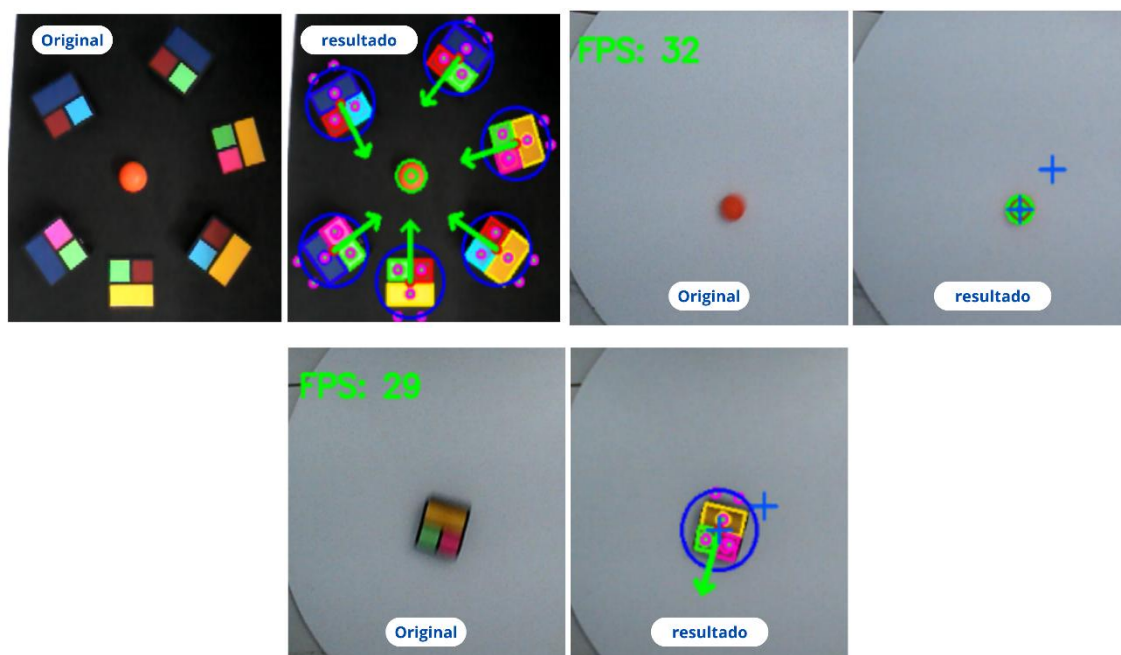
Esse arranjo hierárquico permite conciliar o custo computacional reduzido com boa exatidão de controle, favorecendo o desempenho do sistema de robôs em tempo real.

6 VALIDAÇÃO EXPERIMENTAL

Três experimentos distintos são realizados para validar a operação do sistema de visão computacional, a saber: (i) todos os robôs e a bola em repouso; (ii) a bola em movimento; e (iii) um robô em movimento. Os resultados são apresentados na Figura 9, na qual pode-se observar a capacidade do sistema de identificar e segmentar corretamente os robôs e a bola em cenários

estáticos e dinâmicos. Os vetores verdes indicam a orientação frontal de cada robô, originando-se de seus centros, enquanto a bola é destacada por um círculo verde. Os dados evidenciam que o sistema mantém a capacidade de segmentação e rastreamento mesmo durante deslocamentos, o que é essencial para o contexto do futebol robótico, onde robôs e bola estão frequentemente em movimento.

Figura 9 – Medições do sistema de visão computacional



Fonte: Elaborada pelos autores.

6.1 Controlador embarcado

A validação do controlador embarcado é realizada a partir de testes com diferentes valores de referência para a velocidade angular e para o parâmetro PWM_{medio} . Os resultados são apresentados nas Figuras 9 a 13 e sintetizados na Tabela 2. Os gráficos mostram que o tempo de acomodação do sistema é, tipicamente, inferior a 500 ms, sendo mais rápido quando a velocidade linear é maior (provavelmente porque com mais potência nos motores menor a interferência do atrito). O erro médio é menor que 15%, mesmo para valores diversos de referência. Esses resultados atestam a eficiência da resposta dinâmica do controle implementado.

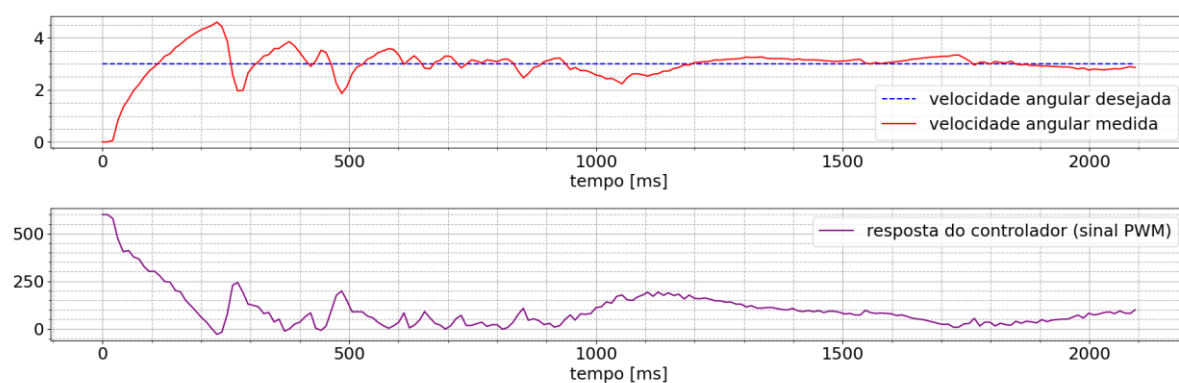
Tabela 2 – Erros médios para valores variados.

$PWM_{med.}$	$\omega_{des.} [rad/s]$	$Erro_{med.} [rad/s]$	$Erro_{med.} [\%]$
0	3,0	0,36	11,9
0	10,0	1,26	12,57
0	20,0	1,86	9,30

500	15,0	1,86	12,43
600	3,0	0,41	13,56
1000	-4,0	0,6	14,90
1000	10,0	0,67	6,68

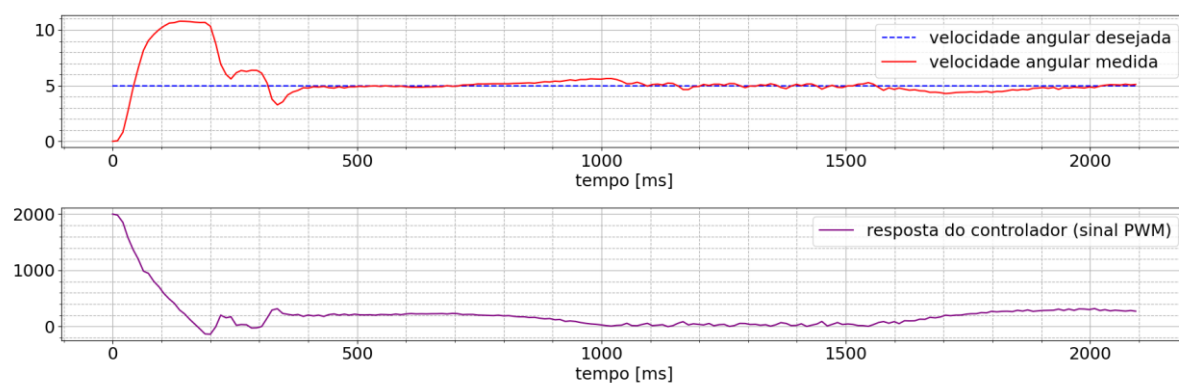
Fonte: Elaborada pelos autores.

Figura 10 – Resultados gráficos 3 Rad/s



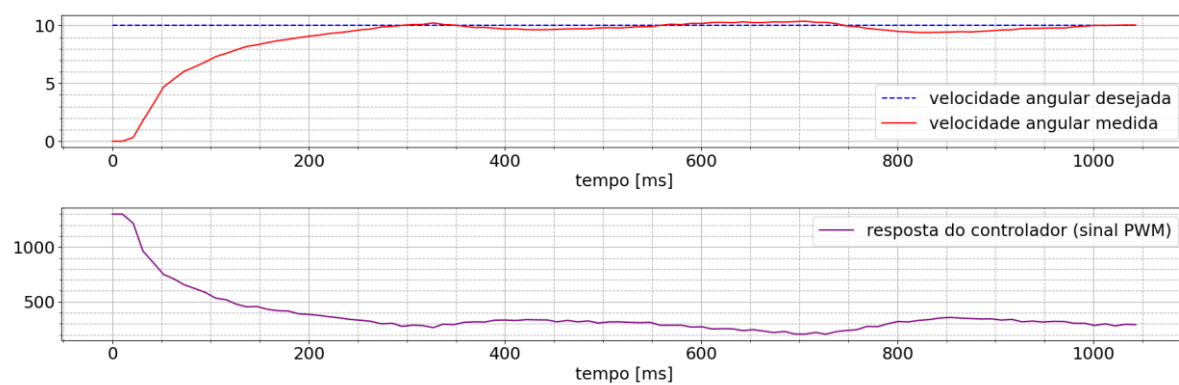
Fonte: Elaborada pelos autores.

Figura 11 – Resultados gráficos 5 Rad/s.



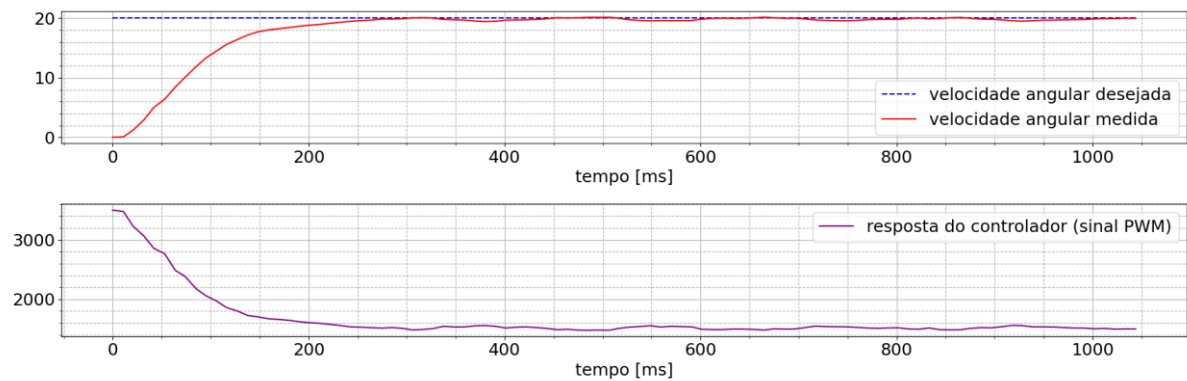
Fonte: Elaborada pelos autores.

Figura 12 – Resultados gráficos 10 Rad/s.



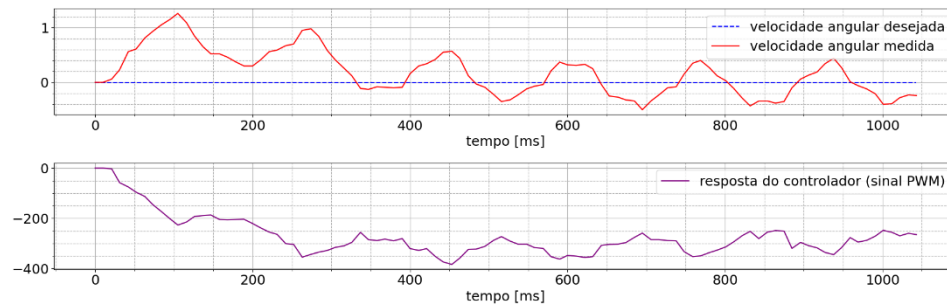
Fonte: Elaborada pelos autores.

Figura 13 – Resultados gráficos 20 Rad/s.

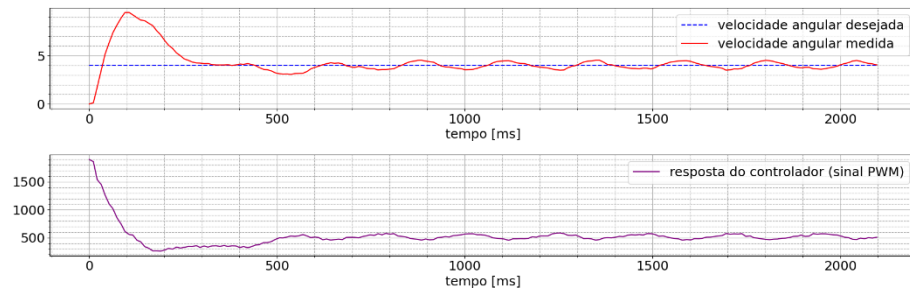


Fonte: Elaborada pelos autores.

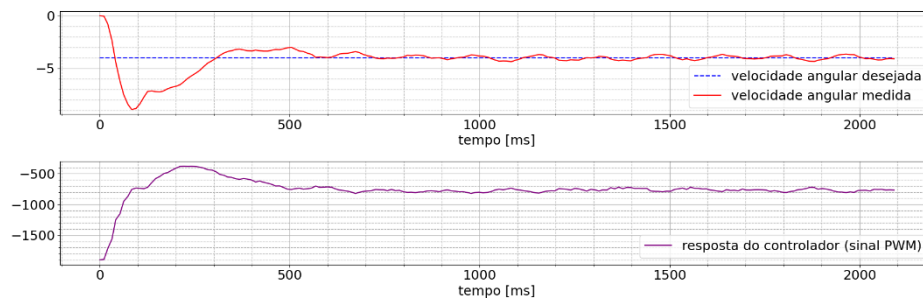
Figura 14 – Resultados gráficos.



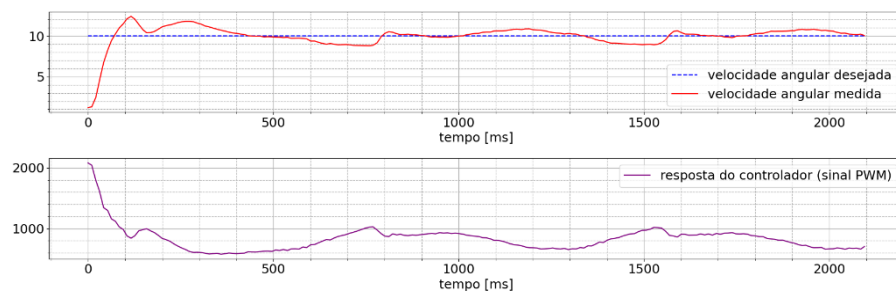
W=0 Rad/s L=1000



W=4 Rad/s L=1000



W=-4 Rad/s L=1000



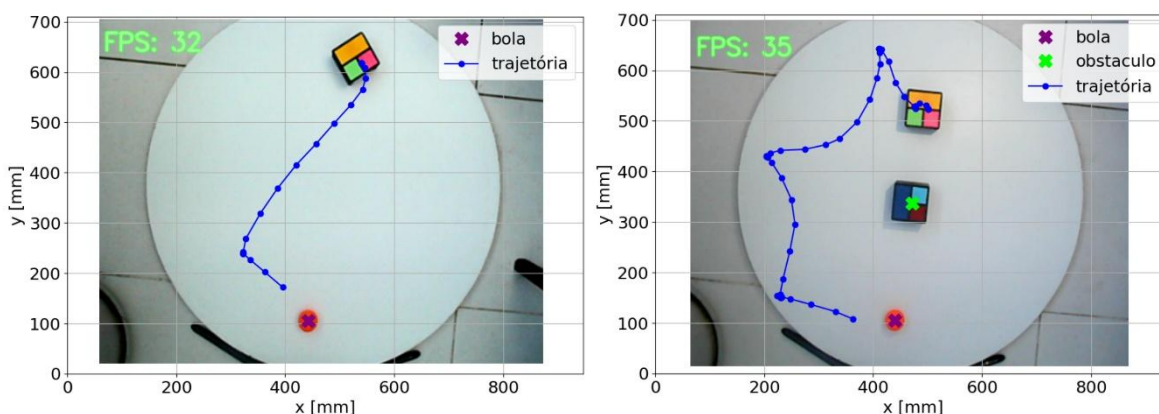
$$W=-10 \text{ Rad/s } L=1000$$

Fonte: Elaborada pelos autores.

6.2 Controlador de alto nível

O controlador de alto nível é testado em duas situações nas quais o objetivo do jogador amarelo é alcançar a bola sem atingir o jogador azul. No primeiro experimento são posicionados em campo apenas a bola e o jogador amarelo. A Figura 14 apresenta a trajetória realizada pelo robô. No segundo experimento um jogador do time azul é incluído na área de movimentação, sendo um obstáculo para o jogador amarelo. A Figura 14 apresenta a posição inicial e a trajetória do robô. Observa-se que o robô do time amarelo alcança o alvo desviando do obstáculo, entretanto, a trajetória resultante possui desvios e oscilações desnecessárias, resultantes das equações adotadas na modelagem do método de campos potenciais. Embora funcionais, estes resultados são pouco eficientes durante uma partida de futebol, sendo importante aprimorá-los. Os autores estão implementando melhorias nesse sentido a serem expostas futuramente.

Figura 15 – Trajetórias desempenhadas com e sem obstáculos.



Fonte: Elaborada pelos autores.

7 CONSIDERAÇÕES FINAIS

Este trabalho apresenta o desenvolvimento de um time de futebol de robôs para a categoria IEEE VSSS desde a sua concepção até a validação experimental e representa o

primeiro passo na direção do estabelecimento de um time competitivo, além de promover a construção da infraestrutura para trabalhos futuros, tais como a fabricação de um campo com medidas oficiais e uma estrutura de testes portátil, além dos robôs jogadores, do sistema de visão computacional e elementos auxiliares. Experimentos simples comprovaram que o sistema de visão computacional identifica os jogadores, mesmo em movimento, de forma consistente. O controlador de velocidade angular embarcado é capaz de manter a velocidade regulada em ampla gama de valores, e com tempo de convergência suficientemente pequeno para a aplicação. O controlador de alto nível, responsável por guiar o robô até o alvo, necessita de melhorias, apesar de atingir o objetivo, a trajetória resultante é muito sinuosa. Entre as inúmeras possibilidades de trabalhos futuros, cita-se a implementação de arquivos de configuração na interface gráfica, o uso de redes neurais na segmentação de cores, melhorias nos algoritmos de controle de baixo e alto níveis usando estratégias de controle mais robustas, com foco especial na movimentação concorrente dos robôs.

REFERÊNCIAS

BRAHMBHATT, S. **Practical OpenCV**. Nova York: Apress, 2013.

CASTELLI, F. *et al.* A machine learning-based visual servoing approach for fast robot control in industrial setting. **International Journal of Advanced Robotic Systems**, v. 14, n. 6, p. 1729881417738884, 2017. Disponível em: <https://journals.sagepub.com/doi/pdf/10.1177/1729881417738884>. Acesso em: 18 mar. 2026.

CBR – COMPETIÇÃO BRASILEIRA DE ROBÓTICA. **Regras do Futebol VSSS 3x3**. Disponível em: <https://cbr.robocup.org.br/wp-content/uploads/2021/05/vssRules3x321.pdf>. Acesso em: 18 mar. 2026.

CONNELL, D.; LA, H. M. Dynamic path planning and replanning for mobile robots using RRT. *In*: IEEE INTERNATIONAL CONFERENCE ON SYSTEMS, MAN, AND CYBERNETICS (SMC), 2017, [S.l.]. **Anais [...]**. [S.l.]: IEEE, 2017. p. 1429–1434. Disponível em: https://ui.adsabs.harvard.edu/link_gateway/2017arXiv170404585C/doi:10.48550/arXiv.1704.04585. Acesso em: 18 mar. 2026.

ESPRESSIF. **ESP32 Series Datasheet**. 2025. Disponível em: https://www.espressif.com/sites/default/files/documentation/esp32_datasheet_en.pdf. Acesso em: 18 mar. 2026.

FONTE, L. **Futebol_VSSS**: repositório de código. GitHub, 2024. Disponível em: https://github.com/luisf18/Futebol_VSSS. Acesso em: 18 mar. 2026.

HEMA, D.; KANNAN, Dr S. Interactive color image segmentation using HSV color space. **Science & Technology Journal**, v. 7, n. 1, p. 37–41, 2019. Disponível em: <https://doi.org/10.22232/stj.2019.07.01.05>. Acesso em: 18 mar. 2026.

IEEE VSSS. **Very Small Size Soccer**. 2025. Disponível em: https://ieeevss.github.io/vss/index_ptbr.html . Acesso em: 18 mar. 2026.

LEE, T. E. *et al.* Camera-to-robot pose estimation from a single image. In: IEEE INTERNATIONAL CONFERENCE ON ROBOTICS AND AUTOMATION (ICRA), 2020, [S.l.]. Anais [...]. [S.l.]: IEEE, 2020. p. 9426–9432. Disponível em: <https://arxiv.org/pdf/1911.09231>. Acesso em: 18 mar. 2026.

MASEHIAN, E.; SEDIGHIZADEH, D. Classic and heuristic approaches in robot motion planning-a chronological review. **World Academy of Science, Engineering and Technology**, v. 23, n. 5, p. 101-106, 2007. Disponível em: <https://www.semanticscholar.org/paper/Classic-and-Heuristic-Approaches-in-Robot-Motion-A-Masehian-Sedighizadeh/7954272d443084a065b08f0dc4b13d1aeca0854f>. Acesso em: 18 mar. 2026

MOHD ALI, N.; MD RASHID, N. K. A.; MUSTAFAH, Y. M. Performance comparison between RGB and HSV color segmentations for road signs detection. **Applied Mechanics and Materials**, v. 393, p. 550-555, 2013. <https://doi.org/10.4028/www.scientific.net/amm.393.550>. Acesso em: 18 mar. 2026.

OLIPHANT, T. **Guide to NumPy**. [s. l.]: Trelgol Publishing, 2006. Disponível em: <https://web.mit.edu/dvp/Public/numpybook.pdf>. Acesso em: 18 mar. 2026.

ORTENZI, V. *et al.* Vision-based framework to estimate robot configuration and kinematic constraints. **IEEE/ASME Transactions on Mechatronics**, v. 23, n. 5, p. 2402-2412, 2018. Disponível em: <https://doi.org/10.1109/TMECH.2018.2865758>. Acesso em: 18 mar. 2026.

PANAGOU, D. Motion planning and collision avoidance using navigation vector fields. In: IEEE INTERNATIONAL CONFERENCE ON ROBOTICS AND AUTOMATION (ICRA), 2014, [S.l.]. **Anais [...]**. [S.l.]: IEEE, 2014. p. 2513–2518. Disponível em: <https://doi.org/10.1109/ICRA.2014.6907210>. Acesso em: 18 mar. 2026

ROBOCORE. **Página da placa vespa**. RoboCore Tecnologia Ltda: Santana do Parnaíba, 2025. Disponível em: <https://www.robocore.net/placa-robocore/vespa>. Acesso em: 18 mar. 2026.

TDK INVENSENSE, folha de dados MPU6050. InvenSense Inc: Sunnyvale-CA, 2023. Disponível em: <https://invensense.tdk.com/wp-content/uploads/2015/02/MPU-6000-Datasheet1.pdf>. Acesso em: 18 mar. 2026.