

# Descrição da equipe PinguimBots - Categoria Very Small Size Soccer 2020

Alan Carlos Junior Rossetto<sup>2</sup>, Christian Cardoso Gressler<sup>1</sup>, Frederico Bueno da Silva Schaun<sup>1</sup>, Gilberto Kreisler Franco Neto<sup>1</sup>, Guibson Souza dos Santos<sup>1</sup>, Larissa de Vargas da Silva<sup>1</sup>, Leonardo Camargo Jorge Fetter Bordignon<sup>1</sup>, Leonardo da Silva Avila<sup>1</sup>, Lucas Coutinho Freitas<sup>1</sup>, Marcelo Schiller de Azevedo<sup>3</sup>, Marco Aurélio Justiniano Alkimim<sup>1</sup>, Paulo Roberto Ferreira Jr.<sup>2</sup>, Pedro Henrique Diehl<sup>1</sup>, Vitor Leitzke Braga<sup>1</sup>, Vivian Domingues Mattos<sup>1</sup>

**Resumo**—Este TDP descreve brevemente a equipe e apresenta as ferramentas e técnicas utilizadas para a construção de um time de robôs para a categoria Very Small Size Soccer (VSSS). É apresentado como foi organizado e estruturado o código desenvolvido pela equipe, as ferramentas e técnicas utilizadas. Bem como uma descrição dos componentes eletrônicos utilizados e a estrutura física dos robôs que foi desenvolvido pela equipe. Por não ter a competição fisicamente, devida à pandemia, também é explicado o processo de integração com o simulador.

## I. INTRODUÇÃO

O PinguimBots é uma equipe multidisciplinar composta por entusiastas e aqueles que desejam se engajar na área da robótica móvel. Alunos estes dos cursos de Ciência da Computação, Engenharia de Computação, Engenharia de Controle e Automação, Engenharia Eletrônica, Jornalismo e Design Digital. Os membros da equipe foram divididos por áreas buscando uma maior produtividade, sendo elas visão computacional, estrutura, eletrônica, controle e mídias sociais. A integração é feita através da constante comunicação entre os membros das diversas áreas, tanto por reuniões ordinárias quanto informalmente durante o processo de trabalho.

A área da visão computacional é a área responsável pela criação de uma interface amigável de comunicação bem como pela extração de informações do campo durante o jogo e compor esses dados para que possam ser processados posteriormente pelos processos de tomada de decisão da estratégia e controle.

A área de controle é responsável por desenvolver as técnicas que irão processar os dados obtidos para o controle da velocidade linear e angular do robô para uma correta execução de trajetória estipulada pelo módulo de estratégia.

A área da eletrônica é responsável pela confecção do circuito eletrônico, selecionar e manipular os componentes eletrônicos, assim garantindo o correto funcionamento do robô.

A área de estrutura é a responsável pelo desenvolvimento de todo o esqueleto do robô, garantido que cumpra com as regras de medidas estabelecidas pelo edital e ao mesmo tempo garantir a disposição do espaço para que caibam todos os componentes eletrônicos.

A área de mídias sociais é a responsável por divulgar a equipe nas redes sociais e atraindo patrocinadores para apoiarem a equipe, sendo tão importante quanto as outras áreas.

A VSSS é uma categoria que possibilita essa integração de diversas áreas, pela diversidade de conhecimento a ser aplicado para o desenvolvimento de todos os componentes necessários para se ter um time competitivo, proporcionando a aplicação prática do que é visto em sala de aula.

Para um jogo da categoria VSSS, cada time possui uma câmera que é posicionada a aproximadamente 2 metros acima do campo (Figura 1). É a partir dessa câmera que serão extraídos todos os dados necessários como a posição dos robôs e da bola no campo e que serão usados para a definição da estratégia a ser usada.

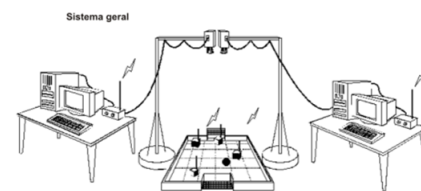


Figura 1. Posicionamento das câmeras no campo

O código da equipe foi dividido em módulos para facilitar a abstração e organização, sendo eles: visão, estratégia, controle e comunicação. Para isso foi utilizado a linguagem de programação C++, por sua característica performática e de fácil organização.

O módulo de visão extrai os dados do campo através da biblioteca de processamento de imagens OpenCV. Esses dados são organizados em uma estrutura e enviados para o módulo de estratégia, que irá definir qual o movimento cada robô deverá executar com base na organização dos componentes do campo. Após, o módulo de estratégia se comunica com o de controle, que define a direção e a velocidade angular e linear correspondente a estratégia escolhida. Por fim, o módulo de comunicação é responsável por codificar essas informações e transmitir via módulo sem fio para cada robô.

A estrutura desenvolvida pela equipe foi pensada para fazer o melhor uso possível do espaço disponível para a organização dos componentes eletrônicos, bem como ser resistente a eventuais impactos durante o jogo.

<sup>1</sup>Graduandos na Universidade Federal de Pelotas, membros do PinguimBots da Universidade Federal de Pelotas (UFPel), Pelotas. Rio Grande do Sul, Brasil.

<sup>2</sup>Professor do CDTec na Universidade Federal de Pelotas.

<sup>3</sup>Professor do IF-Sul de Camaquã.

Esse ano, devido à pandemia, não haverá competição física, sendo a modalidade transferida temporariamente para um meio online e se tornando uma competição simulada. Dessa forma será utilizado apenas a estratégia desenvolvida pela equipe, pois os dados que seriam extraídos pela visão são fornecidos diretamente pelo simulador. Portanto, também é apresentado uma seção sobre a integração com o simulador.

## II. SISTEMAS DE SOFTWARE

O desenvolvimento do sistema de software foi todo feito em C++ por ser uma linguagem que proporciona alta performance e permite uma maior organização, mantendo a simplicidade do projeto devido ao nível de abstração oferecido. Optamos pela separação em módulos para tirar proveito dessa abstração fornecida.

Para o processamento de imagens e desenvolvimento de uma interface gráfica foi escolhido a biblioteca OpenCV. Esta é uma biblioteca de licença BSD que proporciona uma infraestrutura comum para visão computacional[9]. A escolha dessa biblioteca pra trabalho foi devido a larga documentação disponível livremente na internet por causa da sua grande comunidade, tornando simples conseguir suporte, e também a sua robustez, fator oriundo de sua longa trajetória. Ademais, possui uma API de fácil manipulação com uma vasta gama de funcionalidades embutida, o que facilita o desenvolvimento.

### A. Visão

O sistema de visão computacional tem integração direta com o módulo de interface gráfica, que foi desenvolvido usando o HighGUI, biblioteca inclusa no OpenCV que foi desenvolvida principalmente para a depuração. A escolha desta foi devido à sua simplicidade e comunicação direta com a OpenCV, visto que, embora limitada, sirva o escopo do projeto, produzindo um tempo de desenvolvimento reduzido além de menor acoplamento.

A interface gráfica compreende os processos de calibragem, de visualização dos processos internos do módulo de visão e de sinalização de início e final de partida, funcionalidade que é de implementação relativamente trivial devido à simplicidade de biblioteca.

Durante o processo de calibragem será armazenado as cores dos robôs do nosso time, do time adversário e da bola, bem como é feito o mapeamento do campo.

No decorrer do jogo as imagens são capturadas, através de funções fornecidas pelo OpenCV, e então tratadas para remoção de ruídos. Após, as imagens são processadas, sendo feito inicialmente um *threshold* com as cores previamente armazenadas para então ser extraído as coordenadas no plano cartesiano de cada componente móvel em campo e compostas em uma estrutura de dados e então enviada para o módulo de controle.

### B. Controle

Ao receber a estrutura composta das coordenadas cartesianas, o módulo de controle as transforma para o sistema de coordenadas polares [8], obtendo assim o ângulo do robô, que

é um indicativo de sua direção, e a distância em linha reta até a bola.

A partir do processamento desses dados cria-se um modelo cinemático com o controle de Lyapunov[6][7], o qual permite estabelecer relações matemáticas entre os atuadores e as grandezas controláveis, garantindo uma estabilidade mesmo quando as relações entre as grandezas são não lineares. Dessa forma, controlamos a trajetória do robô até a bola e, quando em posse da bola, até o gol.

### C. Estratégia

A estratégia consiste em dividir as tarefas de goleiro, defensor e atacante entre os três robôs [10]. Nesta divisão existem tarefas fixas e flexíveis, onde a única fixa é a do goleiro, e esta será atribuída ao mesmo robô do início ao fim. O comportamento dos outros dois será feito com base em critérios que podem ser diretamente medidos.

Dessa forma, o robô que estiver mais próximo da bola será o atacante, já o outro será o defensor juntamente com o goleiro. As estratégias da defesa e do goleiro se baseiam na orientação da reta que os conecta com a posição atual da bola. Já o atacante terá sua orientação a partir da reta que conecta a bola ao gol.

### D. Comunicação

No final do processo o modulo de comunicação envia esses dados através de comunicação serial para um Arduino para este fazer a transmissão via wifi, utilizando um módulo NRF, para os robôs em campo.

## III. SISTEMA MECÂNICO

O sistema mecânico compreende a estrutura do robô desenvolvida e os componentes eletrônicos utilizados. Neste processo foram antes definidos os componentes a ser utilizados, para a partir de simulações com suas medidas, ser definido a estrutura.

### A. Componentes Eletrônicos

A equipe realizou diversos testes antes de decidirmos quais seriam as melhores opções para utilizarmos no projeto. Após algumas simulações segue a descrição dos principais componentes eletrônicos e suas funções no projeto.

- Motor: É o componente que em conjunto com as rodas permite o deslocamento do protótipo. Em nosso desenvolvimento utilizamos um par de motores Pololu HP com caixa de redução, esse modelo se adequou às nossas necessidades por possuir um tamanho pequeno e grande torque devido a ter incluso a caixa de redução e possuir tensão nominal de 6V.



Figura 2. Motor - Pololu HP[3]

- Ponte H-TB6612FNG: Para ser feito o controle dos motores é necessário o uso de uma ponte H. Ela determina o sentido de rotação de cada um dos motores. O TB6612FNG pode controlar até dois motores CC a uma corrente constante de 1,2A (pico de 3,2A). Dois sinais de entrada (IN1 e IN2) podem ser usados para controlar o motor em um dos quatro modos de funcionamento - CW, CCW, freio-rápido e parada. As duas saídas dos motores (A e B) podem ser controladas separadamente, a velocidade de cada motor é controlada via um sinal PWM com frequência de até 100kHz

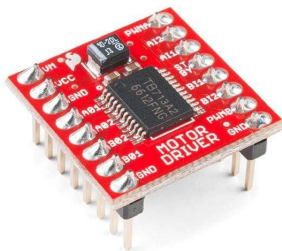


Figura 3. Ponte H-TB6612FNG[5]

- Arduino nano: Este componente é a base de controle do nosso protótipo, um microcontrolador Atmel. O modelo de Arduino que utilizamos é o Arduino Nano, ele possui 8 portas analógicas, 14 digitais e destas digitais 6 podem ser utilizadas como PWM (Pulse Width Modulation), quatro desses pinos são utilizados para o controle dos motores no módulo da ponte H L298 e outros seis pinos são utilizados para o controle do módulo wifi.

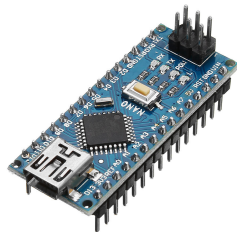


Figura 4. Arduino Nano[1]

- Modulo wifi-NRF24L01: Um módulo wifi é responsável por realizar a comunicação externa do protótipo, para

esse fim utilizamos o NRF24L01. Nosso módulo possui 6 pinos de controle e comunicação de dados, além de 2 para alimentação. Seu alcance pode chegar a 10 metros em ambientes internos e 50 metros em campo aberto.

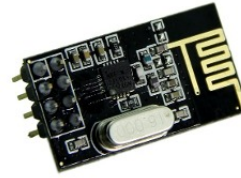


Figura 5. wifi - NRF24L01[4]

- Bateria de 7,4V: Para que todo o hardware do protótipo funcionar ao longo de uma partida utilizamos uma bateria do tipo LiPo (lítio-polímero). A vantagem desse tipo de bateria para nosso uso específico é o ganho com o espaço, pois ela é pequena e comporta dentro do protótipo, ela é uma bateria de 7,4V e 1000mAh.

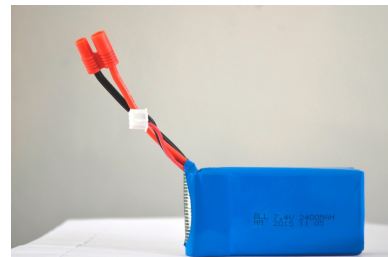


Figura 6. Bateria 7,4V[2]

## B. Estrutura

A estrutura, desenhada no software Inventor, foi pensada para obter melhor desempenho, qualidade e estabilidade. Feita de modo a proteger todos os componentes internos e a parte eletrônica. Utilizando o material ABS por ser resistente a impactos e de fácil acesso.

Foram feitos cortes nas partes frontal do robô para ter um melhor desempenho na posse de bola dos robôs evitando que ela escape para as laterais com a variação da aceleração e direção, consequentemente possuindo um maior domínio sobre a bola.

Outro ponto positivo é o chassi do robô possuir uma pequena distância do chão, logo não há grande variação de altura trazendo maior estabilidade em seus movimentos.

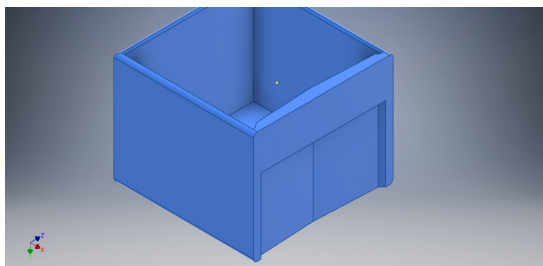


Figura 7. Projeto da estrutura no inventor

### C. Rodas

Desenvolvemos as rodas - que juntamente com os motores permitem o deslocamento do protótipo - com o intuito de ter maior estabilidade e aperfeiçoamento do movimento, pois os modelos prontos não se mostraram eficientes. Foram impressas duas rodas e no chanfro do aro foi colocada uma borracha para obter maior aderência com o chão.

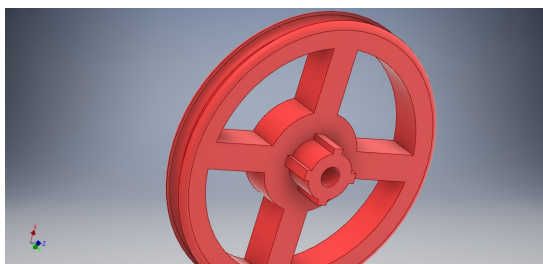


Figura 8. Projeto da roda no inventor

## IV. COMPETIÇÃO SIMULADA

Para a competição simulada foram definidos dois softwares: o FIRASim que é o simulador e de onde virão os dados para a estratégia e o ssl-client, responsável pelo recebimento dos dados do campo e envio dos novos comando de execução para os jogadores em campo. Portanto, foi tirado proveito da estrutura de código já apresentada no ssl-client. A partir do recebimento dos pacotes pela comunicação o mesmo é enviado para um módulo responsável por extrair os dados do pacote e fazer a análise estratégica, gerando as próximas posições que cada jogador assumirá no próximo movimento. As posições são enviadas para outro módulo que irá calcular qual a velocidade que cada roda de cada jogador assumirá para chegar corretamente a próxima posição. Um terceiro módulo é responsável por receber esses dados e enviar de volta para o simulador.

## V. CONCLUSÃO

O trabalho aqui apresentado tem por objetivo demonstrar o projeto de um time de robôs autônomos. Visamos a construção de um time de baixo custo, mas ainda assim competitivo através da utilização de ferramentas de alta performance e da implementação de técnicas bem estabelecidas, tendo em vista o alto nível dos outros competidores.

## REFERÊNCIAS

- [1] Arduino Nano. <https://www.filipeflop.com/produto/placa-nano-v3-0-cabo-usb-para-arduino/>. Acessado: 16/06/2019.
- [2] Bateria 7.4V. <http://www.eletrodrones.com.br/10a07a/bateria-7-4v-2500mah-para-drone-syma-x8w-x8g-x8hw-x8hg-explorer-ydtech-alta-ca>. Acessado: 16/06/2019.
- [3] Motor Polulu HP. <https://www.pololu.com/product/999>. Acessado: 16/06/2019.
- [4] NRF24L01. [https://produto.mercadolivre.com.br/MLB-960136302-modulo-nrf24l01-rf-transceptor-ism-24ghz-pic-arduino-\\_JM](https://produto.mercadolivre.com.br/MLB-960136302-modulo-nrf24l01-rf-transceptor-ism-24ghz-pic-arduino-_JM). Acessado: 16/06/2019.
- [5] Ponte H-L298N. <https://www.curtocircuito.com.br/modulo-driver-ponte-h-l298n.html>. Acessado: 16/06/2019.
- [6] Gabriel Antonio de Araujo Ribeiro; Lívia Chaves Paravidino. Controle de robôs móveis para futebol de robôs. Acessado: 16/06/2019.
- [7] Celso de Sousa Júnior; Elder Moreira Hemerly. Controle de robôs móveis utilizando o modelo cinemático. Acessado: 16/06/2019.
- [8] Walter F. Lages; Elder M. Hemerly. Controle em coordenadas polares de robôs móveis com rodas. Acessado: 16/06/2019.
- [9] OpenCV. <https://opencv.org/about/>. Acessado: 16/06/2019.
- [10] Silas F. R. Alves; Renê Pegoraro; Humberto Ferasoli Filho; Marco A. C. Caldeira; Wilson M. Yonezawa. Carrossel Caipira - O time de futebol de robôs da UNESP de Bauru. Acessado: 16/06/2019.