

Rinobot Team - Equipe de Robótica UFJF

Igor Henrique Silva Barbosa, Tawan Victor Batista de Oliveira, Breno Caldeira de Souza, Juan da Silva Gonçalves, Bárbara Letícia Reis Siqueira, Ramon Ferreira de Paula Matos, Ariza Bertelli Dias, Luiza Helena Ferreira Dias, Vitor Leal de Oliveira Martins, João Pedro Sanseverino, Nicolas de Lima Fonseca, Caio Cedrola Rocha, Guilherme Botelho.

Resumo: Neste artigo serão abordadas informações de todas as etapas sobre a construção dos robôs de futebol da categoria IEEE Very Small Size pela equipe Rinobot.

Palavras-chave: VSSS, Montagem, Visão, Estratégia.

I. INTRODUÇÃO

A Rinobot Team é uma equipe fundada no ano de 2016 e tem como os seus maiores objetivos a construção de robôs para participar de diferentes categorias de competições autônomas e a disseminação da robótica entre as escolas públicas e privadas do ensino médio e fundamental de Juiz de Fora e região.

É uma equipe que visa levar aos seus integrantes vivência prática de sua futura profissão. Atualmente é composta por alunos das Engenharias Elétrica e Mecânica e da Ciência da Computação. Tem como orientadores uma professora do departamento de Engenharia Elétrica e um técnico de laboratório e alguns professores auxiliares de diferentes áreas. Dentre os integrantes da equipe, é subdividida em quatro áreas: Gestão (que é responsável pela parte burocrática, marketing, divulgação e financeira) e as três categorias de competição: Very Small Size Soccer (VSSS), Seguidor de Linha e Sumô Lego. A categoria VSSS é subdividida em, Estratégia, Montagem e Visão Computacional.

Este relatório está dividido da seguinte maneira: A seção II apresenta a descrição dos componentes utilizados para a montagem do robô; a seção III apresenta a descrição do sistema de visão; a seção IV descreve a estratégia utilizada para comandar os robôs; a seção V é uma breve conclusão.

II. MONTAGEM

O trabalho da montagem na equipe tem como objetivo principal aprimorar os robôs utilizados a cada competição, a partir da observação de possíveis melhorias mecânicas e eletrônicas nos robôs durante os testes e jogos. Portanto, existem duas vertentes de trabalho presentes, em que podemos tratar de adaptações em robôs antigos ou então projetos e ideias de novos modelos de robôs. As características do robô utilizado atualmente serão demonstradas com mais detalhes a seguir.

A. Dispositivos Eletrônicos

1) *NRF24*: Utilizamos do módulo de rádio NRF24 para comunicação entre computador e robôs, recebendo dados e transmitindo-os de volta para processamento, tudo isso wireless. O NRF24 possui as seguintes especificações:

- Vcc com 3.3 V;
- Frequência de trabalho de 2.4GHz;
- Corrente nominal de 50mA;
- Alcance de 15 a 60m;



Figura 1. NRF24L01

2) *ESP32*: O microcontrolador utilizado agora é o ESP-WROOM-32. Optamos por esse microcontrolador por possuir uma capacidade de processamento maior que a do arduino, além de ter portas suficientes para demanda de todos periféricos, que incluem: NRF24 e uma ponte H para controle de dois motores DC.



Figura 2. ESP-WROOM-32

3) *Ponte H*: São muito importantes para o desenvolvimento de dispositivos robóticos, consistem em um circuito simplificado capaz de controlar motores de corrente contínua, permitindo desde a definição do sentido de giro até o controle de potência e velocidade. Este controle é feito por meio de modulação por largura de pulso (PWM).

A Ponte utilizada por nós é mostrada na figura 4, a configuração dos pinos para uso, na figura 5, e a modulação por largura de pulso é exemplificada na figura 6.

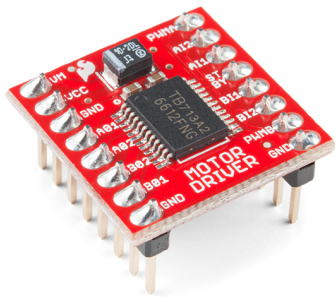


Figura 3. Ponte H TB6612FNG

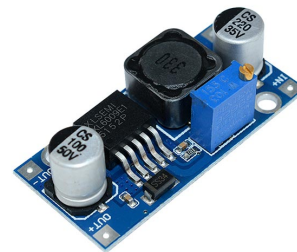


Figura 6. Regulador boost X16009

Input				Output		
IN1	IN2	PWM	STBY	OUT1	OUT2	Mode
H	H	H/L	H	L	L	Short brake
L	H	H	H	L	H	CCW
		L	H	L	L	Short brake
H	L	H	H	H	L	CW
		L	H	L	L	Short brake
L	L	H	H	OFF (High impedance)		Stop
H/L	H/L	H/L	L	OFF (High impedance)		Standby

Figura 4. PWM

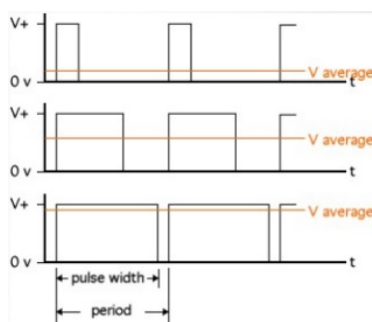


Figura 5. Modulação por largura de pulso

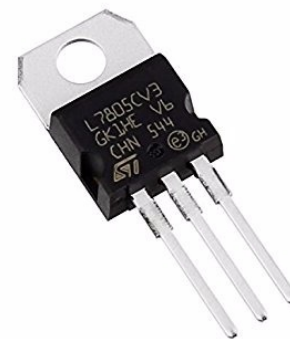


Figura 7. Regulador 7805

7) *Circuito*: A placa foi projetada no programa Protheus em uma face única. Na imagem podemos ver limitações em linhas brancas e amarelas, isso nos mostra os limites do robô e da placa, respectivamente. É necessário respeitá-las para máximo aproveitamento do espaço que temos disponível, além de alinhar as áreas de mecânica e eletrônica.

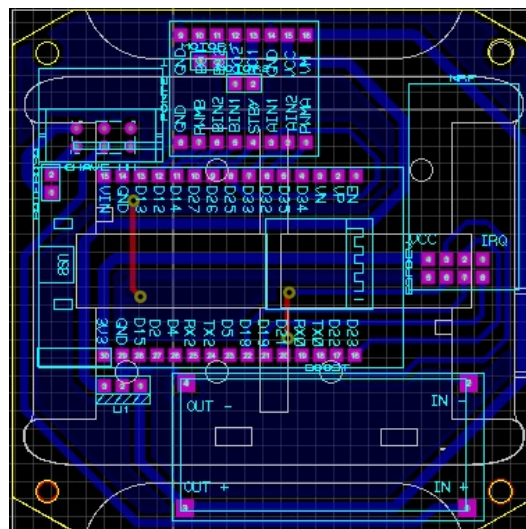


Figura 8. Circuito eletrônico utilizado

4) *Regulador Boost*: Utilizamos um regulador boost X16009, para regularizar a tensão das baterias para 7V constante. Essa tensão é enviada para o pino VM (tensão de referência) da ponte H. A tensão entregue pela bateria pode variar e atrapalhar o funcionamento da ponte.

5) *regulador 7805*: Usamos esse regulador logo após o último (boost), ele abaixa a tensão de 7V para 5V para alimentar o ESP32 e a Ponte H.

6) *Bateria*: As baterias que alimentam os robôs são de polímero de lítio de duas células, e suas especificações técnicas são as seguintes:

- Capacidade de carga: 300 mAh;
- Tensão: 7.4V (duas células de 3.7V cada, ligadas em série);
- Capacidade de descarga contínua: 35C.

B. Sistema de Movimentação

1) *Motor*: Foram utilizados motores de corrente contínua da marca Pololu.



Figura 9. Motor Pololu

Especificações do motor:

- Corrente a vazio de 120 mA;
- Pico de corrente de 1.6 A;
- Torque de 1 Kg x cm;
- Tensão nominal 6V;
- Funciona de 3 a 9V;
- Caixa de redução acoplada que disponibiliza uma velocidade nominal de 625 rpm.

2) *Roda e Pneu*: As rodas e pneus utilizados são da marca Pololu, possuem 32x7mm.



Figura 10. Roda e pneu utilizados

C. Estrutura

1) *Uniforme*: A principal função da carcaça (uniforme) é a de proteger e sustentar os componentes internos do robô, sendo fabricada em impressora 3D. É nela tbm que é fixada as cores para captação do robô. Com o objetivo de aumentar as chances de controlar a bola e facilitar a estratégia de jogo, a carcaça tem duas aberturas para encaixe da bola, na parte dianteira e na traseira.

2) *Base*: A base é responsável pelo suporte dos motores, da bateria e a fixação do uniforme. Além disso, é um componente do robô que está em contato direto com a bola no jogo. Por isso, possui faces côncavas para facilitar a condução da bola durante a partida.

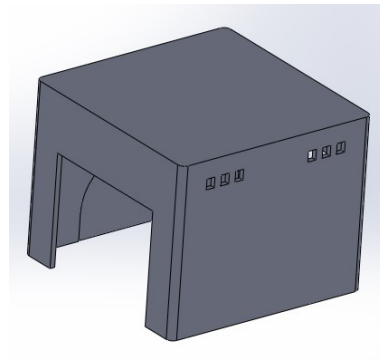


Figura 11. Uniforme

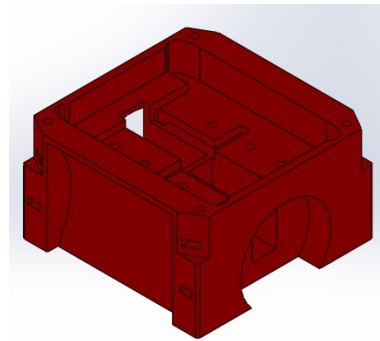


Figura 12. Base

III. VISÃO COMPUTACIONAL

O objetivo desse setor consiste em determinar o posicionamento da bola e de todos os robôs no campo e retornar essas informações com o menor delay e erro em relação à realidade possível, e também de garantir uma parte gráfica de fácil compreensão para interação com o usuário.

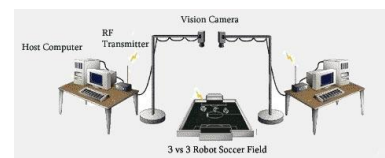


Figura 13. Sistema Visão Computacional

O código vem sendo desenvolvido em C++, usando a biblioteca de código livre OpenCV para processar as imagens obtidas pela câmera utilizando métodos de visão computacional e o sistema do futebol de robôs está rodando no sistema operacional Debian Linux OS. Como existem diversos fatores que podem influenciar a percepção do mundo real após a leitura da câmera, como a iluminação e a incapacidade de o sensor representar o que está captando com perfeição, estratégias computacionais são aplicadas nas imagens obtidas de modo a diminuir esses erros.

O sistema recebe as imagens da câmera e as representa em uma matriz. Primeiramente é feito o processamento dos elementos dessa matriz, alteração de intensidade dentre outras

características básicas que podemos tratar a imagem, logo em seguida são aplicados filtros que suavizam as intensidades dos pixels e diminuímos os ruídos captados pela lente da câmera. Após a etapa de tratamento da imagem é necessário eliminar as cores desnecessárias para o jogo, para isso utiliza-se o sistema de cores HSV, visto que a separação no sistema de cores RGB (adivindo da câmera) não realiza uma boa separação. A próxima etapa consiste em identificar os contornos das figuras encontradas na separação de cores, os mesmos passam por testes de modo a retirar possíveis ruídos capturados e certificar que só será trabalhado com pontos de interesse (o que minimiza o gasto de processamento).

Os pontos obtidos passam por um algoritmo que encontra o centro de cada contorno e logo em seguida por um reconhecimento de robôs, que tem a capacidade de identificar qual combinação de centros formam um robô, sendo ele aliado ou inimigo. Ao final os centros e por consequência o ângulo de cada robô são passados para tratamento da estratégia.

IV. ESTRATÉGIA

A estratégia de jogo consiste em, após ter recebido por meio da câmera e do programa de visão computacional os dados relativos à posição dos robôs (tanto do nosso time quanto do time adversário) e da bola, definir as ações que serão tomadas por cada um dos robôs. Para definir esses comandos, contamos com um algoritmo implementado em C++, que trabalha em tempo real, assim como o programa de visão computacional, enviando aos robôs sinais de saída (velocidade das rodas esquerda e direita dos robôs) de acordo com os sinais de entrada recebidos.

Esse algoritmo pode ser resumido em três etapas, seleção de metas, sistema de navegação, cálculo de velocidades.

A. Seleção de Metas

A seleção de metas, nada mais do que a definição física de onde cada robô deve atingir. Nesse contexto funções de jogos são criadas para a definição desses objetivos (Ex: Atacante, Goleiro, etc..). Uma vez definida o objetivo de cada robô as ações passam a ocorrer no sistema de navegação.

B. Sistema de Navegação

O sistema de navegação da Rinobot Team aborda o uso de Campos Potenciais baseados em Problemas de Valor de Contorno (PVC) [1], mais especificamente Campos Potenciais Univetoriais(CPU).

1) *CPU*: A proposta consiste no uso de uma navegação unificada, que gera a direção de movimento para sair da posição atual e chegar ao destino. Desta forma, invés de gerar todo o caminho e um potencial para todo o campo, gera-se apenas o movimento relativo a posição atual do robô, o que reflete em uma economia computacional em relação a outros métodos de navegação. Esta característica torna o CPU uma ótima alternativa para ambientes dinâmicos e, por consequência, para a aplicação no VSSS.

Idealmente, o CPU utiliza de um vetor de módulo constante unitário e modifica apenas a direção de propagação desse

vetor baseado no comportamento de vetores tangenciais a composição de duas espirais lado a lado. Dessa forma, o caminho gerado converge para meta delimitando a orientação ao atingi-la.

Abaixo, ilustrações do comportamento do CPU.

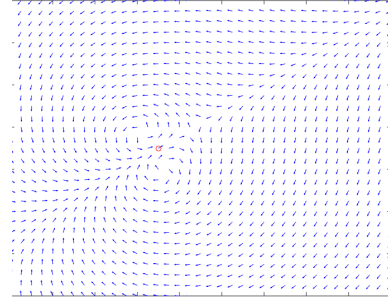


Figura 14. Campos Potenciais Univetorial (CPU)

C. Cálculo de Velocidades

Essa etapa é responsável pelo controle das velocidades dos motores encarregados pelo deslocamento dos robôs autônomos. Esse controle de velocidade é feito por um controlador PD(Proporcional-Derivative) projetado de modo que as ações delegadas ao robô chegue ao seu destino seguindo a trajetória esperada, sem oscilações e erros, no menor tempo possível. Nesse contexto, a velocidade máxima (v_{Max}) de deslocamento dos robôs é fixada, enquanto o controlador age sobre o erro de angulação definido pela orientação do robô e a orientação que ele deve seguir definida previamente pelo sistema de navegação.

$$erro = anguloDoRobo - anguloASeguir \quad (1)$$

Dessa forma, combinando o controlador PD com a equação cinemática acerca das velocidades linear e angular do robô a equação final que rege o controle de velocidade é obtida:

$$v = v_{Max} * modulo(erro) / La + v_{Max} \quad (2)$$

$$w = kp * erro / 180 + kd * (erro - erroAnterior) \quad (3)$$

Onde La é definida para que o agente possa girar em torno do próprio eixo quando $erro \geq La$. Kp e Kd são constantes do controlador ajustáveis para suavizar o movimento do robô.

Tendo definido as velocidades linear e angular, elas podem ser convertidas em velocidade do motor esquerdo (VE) e velocidade do motor direito (VD) pelas seguintes equações:

$$VE = v + w * l \quad (4)$$

$$VD = v - w * l \quad (5)$$

Onde l é o raio das rodas fixas aos motores.

V. ADAPTAÇÃO PANDEMIA COVID-19

Diante da atual situação do Covid-19 adaptações tiveram que ser feitas no contexto do VSSS da Rinobot Team. A primeira dessas adaptações foi a adoção do VSS-SDK, um software livre desenvolvido pela SirLab com a finalidade de simular o ambiente da categoria IEEE-VSSS. Tendo em vista o VSSS-SDK, todo o código relacionado a estratégias utilizadas em um jogo foi convertido para o formato adequado do simulador.



Figura 15. VSS-SDK

Posteriormente, foi adotado a Fira-Sim devido a previsão de uso na Larc-2020, nesse contexto todo o conteúdo foi novamente adaptado. Sendo assim toda a parte que envolve estratégias de jogo foram adaptadas para o Fira-Sim, e toda a parte física dos robôs, embora sofrerão modificações em relação ao ano anterior não teve nenhuma mudança de fato validado devido a falta de acesso dos robôs e o laboratório da equipe na UFJF.

VI. CONCLUSÃO

Este artigo apresenta uma breve introdução do trabalho desenvolvido pela Rinobot Team no futebol de robôs da categoria Very Small Size Soccer (VSSS). Possíveis melhorias serão sempre necessárias, a fim de tornar a equipe cada vez mais competitiva.

O principal objetivo deste projeto é promover e incentivar a pesquisa na área de robótica tanto na Universidade Federal de Juiz de Fora, quanto nas escolas da cidade, e permitir o desenvolvimento técnico dos alunos envolvidos. As expectativas de resultados são boas, visto os resultados alcançados nas últimas competições e ao empenho e dedicação dos membros da equipe.

VII. REFERÊNCIAS

- [1] Yusun Lim*, Seung-Hwan Choi*, Jong-Hwan Kim* and Dong-Han Kim** Evolutionary Univector Field-based Navigation with Collision Avoidance for Mobile Robot, Seoul, Korea, July 6-11, 2008
- [2] OLIVEIRA, Leandro Luiz Resende de; Controle de Trajetória Baseado em Visão Computacional Utilizando o Framework ROS, Dissertação de Mestrado, Universidade Federal Juiz de Fora, Novembro de 2013.

[3] VILAS BOAS, Ana Sophia; Notas de Aula Laboratório de Robótica; UFJF.