

Team Description Paper UnBall - Universidade de Brasília

Nikson Bernardes Fernandes Ferreira¹, Hiago dos Santos Rabelo²,

André Dornelas Sanches³, Gabriel Baliza Rocha⁴, Juliana Araújo Marins⁵, Paulo Henrique Rosa da Silva⁶, Maria Claudia Campos Martins,⁷ Ayssa Giovanna de Oliveira Marques⁸, Kawê Takamoto Siqueira⁹

Resumo—Este artigo descreve o desenvolvimento da equipe de futebol de robôs da Universidade de Brasília, UnBall, no que se refere à categoria IEEE Very Small Size. Apresenta-se o processo de projeto e construção dos robôs, da estrutura de software e algoritmos propostos para localização. Apresenta também o controle e a estratégia a ser utilizada na Competição Brasileira e Latino Americana de Robótica 2022 (CBR/LARC 2022).

I. INTRODUÇÃO

A equipe de futebol de robôs da Universidade de Brasília (UnBall) surgiu em novembro de 2009, após a realização da Competição Brasileira de Robótica e do IX Simpósio Brasileiro de Automação Inteligente na Universidade de Brasília. Contudo, a equipe se desestruturou no final do ano de 2012, e foi oficialmente fechada, mas foi reestruturada novamente no início de 2014, com novos integrantes. A equipe competiu com sua nova formação pela primeira vez em 2015, participando da LARC/CBR realizada em Uberlândia. Devido a diversas falhas técnicas e estruturais apresentadas no projeto desenvolvido no ano anterior, a UnBall passou a desenvolver um projeto para os robôs completamente novo, mas agora levando em conta a experiência previamente adquirida, objetivando desenvolver um modelo mais competitivo. Na CBR/LARC 2016, este novo

projeto foi utilizado na competição, ainda não atingindo o resultado esperado. Assim, a equipe continuou trabalhando para aperfeiçoá-lo e torná-lo totalmente competitivo para competições. A equipe continua trabalhando para garantir um desenvolvimento sustentável a longo prazo, para que o projeto não termine com a formatura dos integrantes atuais. Além disso, deseja-se que o grande esforço dispendido pelos membros da equipe seja recompensado por meio de bons resultados obtidos em competições.

A UnBall está voltada para a categoria IEEE Very Small Size e é formada por 9 alunos de graduação em Engenharia Mecatrônica, Engenharia Elétrica e Ciência da Computação, divididos de forma alternada em dois grupos: Hardware e Software. A equipe de Hardware tem como objetivo a construção e manutenção dos robôs utilizados para a competição. Por sua vez, o grupo de Software é responsável pelo software do sistema computacional remoto que os controla. O sistema é subdividido nas áreas de Estratégia, que tem como objetivo a definição de comportamentos para os robôs de acordo com a situação do jogo, e da Visão Computacional, que se dedica aos algoritmos de localização dos robôs a partir das imagens fornecidas por uma câmera fixada sobre o campo.

Dentre os feitos marcantes da UnBall desde sua formação, pode-se ressaltar a elaboração de um tutorial para programação em microcontroladores AVR, voltado para a formação de nossos integrantes. O desenvolvimento de um simulador de futebol de robôs 3D, o desenvolvimento de um simulador de futebol de robôs 2D, a codificação de algoritmos de segmentação e rastreamento de múltiplos alvos por visão computacional e o projeto completo dos robôs utilizados pela equipe. Foi também organizado um ciclo de palestras sobre robótica, para promover a equipe e divulgar as linhas de pesquisa de professores apoiando o projeto, além de diversos cursos e treinamentos internos relacionados às ferramentas utilizadas pela equipe (e.g. OpenCV, AVR, Solidworks, Subversion, C++, Python).

Além disso, diante do contexto da pandemia de 2020, a equipe participou de competições não presenciais, utilizando as plataformas de simulação fornecida pela organização responsável pela competição: FIRASim e VSSReferee. Como os requisitos das competições foram limitados a esses softwares, o sistema da UnBall pôde ser simplificado à estratégia e controle, permitindo a equipe focar todos os esforços para a melhoria dessas áreas do nosso sistema.

O restante deste documento é organizado como segue. Toda a descrição da estrutura proposta para o sistema de futebol de robôs da equipe UnBall é feita na Seção II,

¹Nikson Bernardes Fernandes Ferreira é Engenheiro de Controle e Automação, Faculdade de Tecnologia, Universidade de Brasília, Campus Darcy Ribeiro, Brasília, Distrito Federal, Brasil nik.son18@hotmail.com

²Hiago dos Santos Rabelo é estudante de Engenharia Mecatrônica, Faculdade de Tecnologia, Universidade de Brasília, Campus Darcy Ribeiro, Brasília, Distrito Federal, Brasil hiagop22@gmail.com

³André Dornelas Sanches é estudante de Engenharia Mecatrônica, Faculdade de Tecnologia, Universidade de Brasília, Campus Darcy Ribeiro, Brasília, Distrito Federal, Brasil andredornelas23@gmail.com

⁴Gabriel Baliza Rocha é estudante de Engenharia Mecatrônica, Faculdade de Tecnologia, Universidade de Brasília, Campus Darcy Ribeiro, Brasília, Distrito Federal, Brasil gabrielbaliza1@gmail.com

⁵Juliana Araújo Marins é estudante de Engenharia Elétrica, Faculdade de Tecnologia, Universidade de Brasília, Campus Darcy Ribeiro, Brasília, Distrito Federal, Brasil teco10e20@gmail.com

⁶Paulo Henrique Rosa da Silva é estudante de Engenharia Mecatrônica, Faculdade de Tecnologia, Universidade de Brasília, Campus Darcy Ribeiro, Brasília, Distrito Federal, Brasil rosa.paulo@aluno.unb.br

⁷Maria Claudia Campos Martins é estudante de Engenharia Mecatrônica, Faculdade de Tecnologia, Universidade de Brasília, Campus Darcy Ribeiro, Brasília, Distrito Federal, Brasil mariaclaudia8.0@hotmail.com

⁸Ayssa Giovanna de Oliveira Marques é estudante de Ciência da Computação, Departamento de Ciências da Computação, Universidade de Brasília, Campus Darcy Ribeiro, Brasília, Distrito Federal, Brasil ayssanaara@gmail.com

⁹Kawê Takamoto Siqueira é estudante de Engenharia Mecatrônica, Faculdade de Tecnologia, Universidade de Brasília, Campus Darcy Ribeiro, Brasília, Distrito Federal, Brasil kawetakamoto@gmail.com

incluindo a estrutura usada na competição não presencial. A base da estratégia implementada pode ser encontrada na Seção III. Já a Seção IV, descreve a base do controle. Todo o projeto mecânico e eletrônico é descrito na Seção V. Esta seção também descreve o sistema de localização por visão computacional. Por fim, a Seção VIII apresenta algumas considerações finais.

II. DESCRIÇÃO DO SISTEMA

O sistema utilizado é dividido basicamente em dois módulos, cada um com algumas subdivisões. Conforme é apresentado na Figura 1 esses dois módulos são executados, respectivamente, em um computador central e em cada robô. O módulo executado no computador é responsável pela visão computacional, estratégia, uma primeira malha de controle e rádio (comunicação). O módulo executado no robô é responsável por comunicação, uma segunda malha de controle e pela parte mecânica.

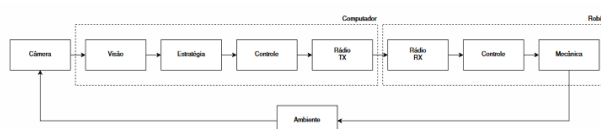


Fig. 1. Sistema UnBall

Primeiramente sobre o módulo "computador". De maneira geral a visão é responsável por, a partir de uma sequência de *frames* oriundos da câmera, obter a postura (posição e rotação) dos robôs e posição da bola, assim como suas taxas de variação. A estratégia, com os resultados da visão, decide o papel de cada robô e o movimento realizado por cada papel. O controle é responsável por emitir as velocidades angular e lineares dos robôs, de forma que o objetivo local de cada robô seja atingido. Por fim, o rádio TX recebe essas velocidades angulares e lineares e as envia aos robôs via rádio 2.4GHz. Essas áreas do módulo computador serão mais detalhadas em seções seguintes deste trabalho.

Sobre o módulo robô: o rádio RX recebe as velocidades angulares e lineares referentes a todos os robôs, e são repassadas ao controle as velocidades devidas baseadas em seus identificadores. O controle recebe as velocidade angular e linear de referência, compara com os dados obtidos pelos sensores (*encoders* e IMU) aplicando a tensão nos motores de forma a alcançar essa referência com o desempenho desejado. O submódulo mecânica é onde estão os componentes físicos, que devem alcançar altas velocidades e grande torque.

III. ESTRATÉGIA

A estratégia tem como objetivo principal regimentar o comportamento dos robôs durante a partida, de forma que estes ajam em equipe para alcançar a vitória. A base da estratégia se encontra no conceito de "entidades" que "posuem" os robôs nos momentos apropriados.

Entidade é uma categoria geral que representa um conjunto de movimentos possíveis, os quais definem uma *pose*,

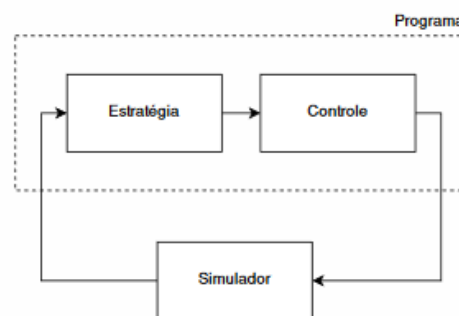


Fig. 2. Sistema UnBall - Não presencial

posição linear e angular, em que o robô deve estar num futuro próximo. Dentro dessa categoria, definimos funções que determinam o conjunto de movimentos desejados, distribuindo papéis (ou subentidades) para os robôs de goleiro, atacante, zagueiro e meio-campo.

A alocação desses papéis é feita de forma variada, a depender do momento do jogo. No início do jogo, as entidades são fixas aos robôs e permanecem desse modo até ocorrer uma situação que precise de troca de papéis ou até entrar em "modo insano" que ocorre quando há uma diferença entre os números de gols aliados e inimigos, sendo que este valor é definido a critério da equipe a cada jogo.

A troca de papéis entre os robôs é feita por meio de uma máquina de estados, que verifica os parâmetros do robô (posição em relação a bola, sua velocidade angular, entre outros) para decidir se o robô deve permanecer no seu papel ou não. Geralmente, na estratégia da equipe, o atacante é aquele que está no campo inimigo e todos seus esforços são para levar a bola até o gol; já o meio-campo é um auxiliar do atacante, que tenta fazer o gol em caso de falha do atacante, sem atrapalhar a movimentação do robô atacante; o zagueiro e o goleiro cuidam da defesa do campo aliado. Uma observação sobre o goleiro: ele é fixado na linha do gol e acompanha a bola para evitar que ela entre no gol.

Para definir o comportamento desses papéis da entidade é ainda construída outra máquina de estados na qual os estados são os comportamentos a serem feitos diante de situações que cada robô pode enfrentar.

Dentro de cada estado dos papéis dos robôs, para que se atinja a *pose* desejada, são definidos campos vetoriais unitários para a navegação. O campo mais utilizado na estratégia atual da equipe é o UVF (Univector Field), sendo, na verdade, um algoritmo que nos entrega um campo vetorial unitário bidimensional; este último é responsável por determinar a postura do robô em cada ponto do campo, considerando o objetivo definido pela função que cada robô pode ocupar, e possíveis obstáculos. Esse campo pode ser visto na figura 3. O objetivo é tratado como um campo atrativo, em

sua maioria das vezes. Junto do campo é feito o controle do sistema, que corrige a trajetória levando em conta as limitações físicas (reais ou simuladas) do robô. Tais campos se assemelham com retratos de fase dos sistemas dinâmicos. Dessa forma, para o equacionamento dos diferentes campos, utilizamos equações dinâmicas conhecidas.

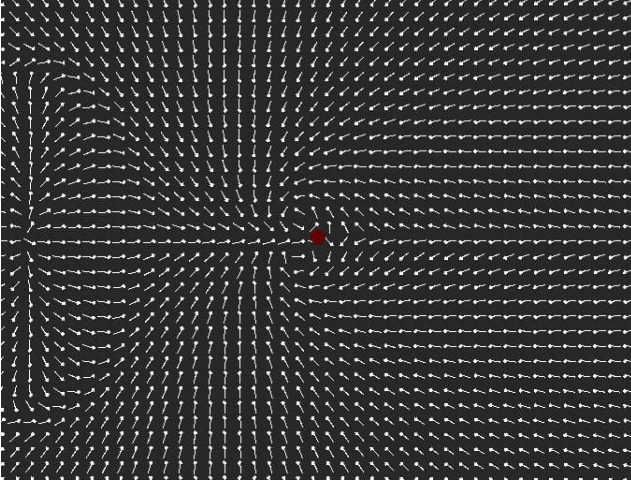


Fig. 3. Campo UVF

A última versão da estratégia segue a ideia descrita anteriormente. Sua implementação foi feita em Python e pode ser vista no repositório da equipe no GitHub, por meio deste link: .

IV. CONTROLE

A modelagem em coordenadas polares é relativamente mais simples que a cartesiana. Tal fato é justificado pela natureza não holonômica dos robôs. Assume-se que a dinâmica do robô possa ser modelada pela equação:

$$\dot{\theta} = \omega$$

Em que θ é o estado que define o ângulo do robô e ω a velocidade angular de referência como entrada do sistema. Claramente trata-se de uma aproximação já que a velocidade angular apresenta uma dinâmica de aceleração.

O controle é responsável por garantir que o sistema siga uma referência de acordo com o desempenho desejado. Para tal, optou-se por utilizar um controle em cascata, onde são utilizadas duas malhas, de forma que a referência do controlador interno é gerada pelo externo. Tal topologia possibilita o projeto de dois controladores cuja construção se torna mais simples que o uso de apenas um controlador para todo o sistema. Assim, uma parte da malha fechada observada na figura 4 fica responsável pela cinemática dos robôs e a outra parte, pela dinâmica que envolve a dinâmica dos motores DC acoplados a cada roda.

Modela-se o sistema de forma que seja composto por dois subsistemas desacoplados em cascata, sendo necessário assim a construção de duas malhas de controle, relativamente mais simples do que seria exigido se considerássemos apenas um sistema mais complexo, como pode ser visto em 4.

Logo, considera-se que um subsistema representa a cinemática dos robôs e o segundo a dinâmica, a qual envolve a dinâmica dos motores DC acoplados a cada roda.

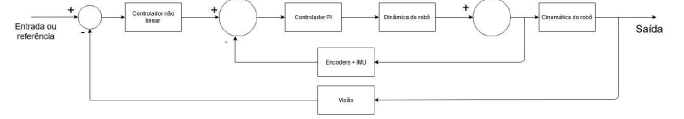


Fig. 4. Diagrama de blocos simplificado do sistema de controle.

A. MALHA EXTERNA

A malha mais externa é responsável por alinhar o robô com o campo de vetores unitários e pelo objetivo final, definidos pela estratégia. Por representar um sistema naturalmente não linear, optou-se por desenvolver e implementar uma lei de controle pelo método de *Lyapunov redesign* a qual define-se uma lei de controle de forma que torne o sistema assintoticamente estável, ou seja, para todo ponto diferente da origem, a derivada da função de Lyapunov é negativamente definida.

Define-se erro de rastreamento como:

$$e_{\theta} = \theta_r - \theta \quad (1)$$

A função positivamente definida 2 como função de Lyapunov:

$$L = \frac{e_{\theta}^2}{2} \quad (2)$$

Sua derivada:

$$\dot{L} = e_{\theta} \dot{e}_{\theta} \quad (3)$$

Expandindo é:

$$\dot{L} = e_{\theta}(\dot{\theta}_r - \omega) \quad (4)$$

Sendo ω o sinal de controle, escolhe-se de forma que $L < 0$, para todo $e_{\theta} \neq 0$:

$$\omega = \dot{\theta}_r + k_w \sqrt{|e_{\theta}|} \cdot \text{sign}(e_{\theta}) \quad (5)$$

Onde k_w é uma constante positiva. Substituindo 5 em 4, tem-se:

$$\dot{L} = -k_w |e_{\theta}| \sqrt{|e_{\theta}|} \quad (6)$$

Tornando \dot{L} negativamente definido, ou seja $\dot{L} < 0$ para todo $e_{\theta} \neq 0$. Ou seja, em aspectos práticos, $e_{\theta} \rightarrow 0$, quando $t \rightarrow \infty$.

A velocidade linear é definida por restrições obtidas empiricamente.

$$v = \min(v_1, v_2) + injection \quad (7)$$

Em que v_1 e v_2 são definidos como:

$$v_1 = v_{bias} + (v_{max} - v_{bias}) \cdot e^{(-k_{path} \cdot |path|)} \quad (8)$$

$$v_2 = k_p \cdot |r - r_b|^2 + v_{ref} \quad (9)$$

onde, v_{bias} é a velocidade mínima desejada durante a trajetória, v_{max} é a máxima. A variável $path$ reflete o quão curvo é o campo a frente e é definida por:

$$path = 10 \sum_k^n |\theta_k - \theta_{k-1}| + e_{th} \quad (10)$$

Onde θ_k é o ângulo do campo em uma posição P_k a frente do robô. A posição P_k é definida pela regra $P_k = P_{k-1} + step \cdot \angle \theta_k$. $step$ e n são definidos empiricamente.

O injection definido em 11 serve para aumentar a velocidade quando a bola está em movimento contrário ao ângulo de chegada:

$$injection = \begin{cases} \frac{2 \cdot |v_b| \cdot 0.10}{|r - r_b|}, & \text{se } v_b \bullet \angle \theta_b < 0 \\ 0, & c.c. \end{cases} \quad (11)$$

B. MALHA INTERNA

A malha interna tem a função de garantir que os robôs sigam as velocidades definidas pela primeira malha com o desempenho desejado.

Por se tratar de um sistema linear, ou linear por partes (considerando a saturação dos motores como uma não linearidade dura), optou-se por projetar um controle linear através do método do lugar geométrico das raízes da planta discretizada. O controlador escolhido foi o PI discreto. É realizável, possui erro nulo a degraus de referência e perturbação e é executado em um microcontrolador através de uma equação de diferenças.

O controle dos motores é feito de forma acoplada. Este é alimentado pelos dados obtidos na visão em conjunto, possibilitando a compensação de deslizamentos da roda, os quais podem ser oriundos da própria aceleração exacerbada, ou por colisões.

V. PROJETO E CONSTRUÇÃO DOS ROBÔS

A seguir serão apresentadas especificações do projeto mecânico, elétrico e de visão computacional utilizado e ainda em desenvolvimento pela equipe.

A. Projeto Mecânico

O projeto mecânico dos robôs é feito no software Fusion 360, no qual é possível prever como será o chassi, as rodas e como os demais componentes serão dispostos no robô. O motor utilizado no projeto da última competição presencial foi do modelo 1524b009sr da Faulhaber, com uma caixa de redução 19:1 que já vem de fábrica, que proporciona um alto torque na saída. O tamanho dos motores apresenta alguns desafios a mais, por isso o desenvolvimento de um novo projeto com motores menores está em andamento. Na figura 5 é possível observar a estrutura do projeto atual. Optou-se por substituir o motor por outro de menores dimensões: modelo N20 da Pololu com redução de 30:1. Assim calculou-se que teríamos uma velocidade acima de 1m/s devido à escolha da roda.

Na figura 6 é possível visualizar uma versão do projeto em construção, feito no software FreeCAD.

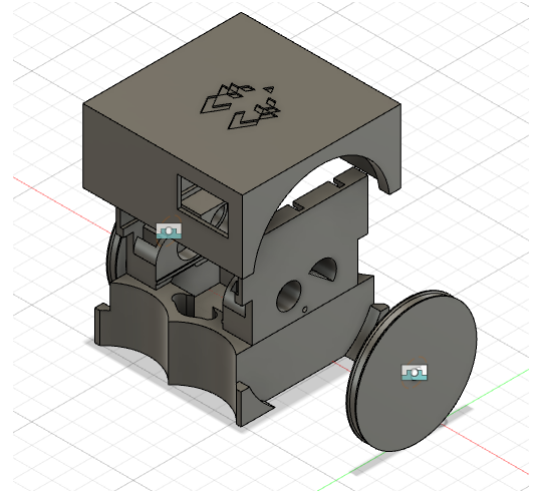


Fig. 5. Estrutura do robô atual

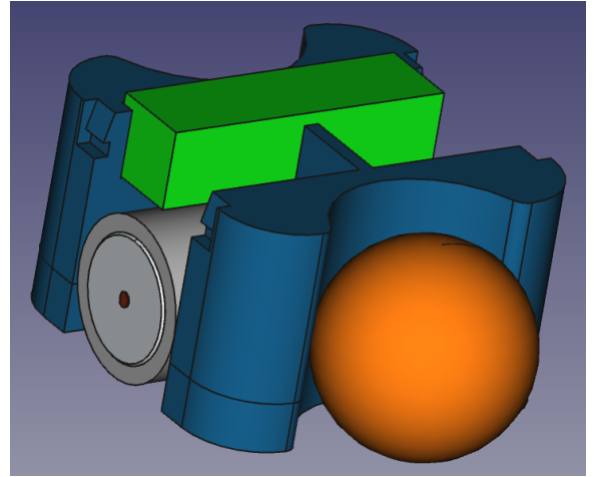


Fig. 6. Projeto para próximas competições presenciais

A estrutura do robô é majoritariamente feita em impressão 3D utilizando plástico PLA que fornece resistência mecânica boa o suficiente para a aplicação e torna o trabalho mais fácil, em comparação com outros polímeros como o ABS usado em anos anteriores. Como já mencionado, a ideia de todo o projeto é sua modularidade para que melhorias em uma parte possam ser efetuadas sem interferência direta nas demais partes. Esta ideia se aplica também ao projeto mecânico separadamente, no qual partes específicas do chassi ou da roda possam sofrer alterações sem que afete o restante da estrutura.

B. Projeto Eletrônico

O projeto eletrônico envolve o acionamento e controle dos motores, a definição e implementação do protocolo de comunicação com o computador central, o projeto da placa de circuito impresso utilizada nos robôs e a codificação do *firmware*.

Para simplificar o processo de prototipagem da placa de circuito impresso, optou-se pela utilização do software de

construção e simulação de circuitos eletrônicos EasyEDA¹, que permite a adição de diversos componentes eletrônicos e a possibilidade de se trabalhar em grupo no projeto online.

Utiliza-se o microcontrolador Wemos D1 Mini, pelo seu tamanho e arquitetura. O *driver* de motor TB6612FNG Breakout v10² foi utilizado para controlar simultaneamente os dois motores do robô. Este *driver* recebe um sinal PWM do Arduino, que indica o quanto de tensão da fonte os motores devem receber. Como fonte de energia para os robôs, utiliza-se baterias lipo 2s de tensão nominal de 7,4v e carga máxima de 450mAh.

Utiliza-se o módulo NrF24I01 Rf *transceiver* de frequência de operação de 2.4 GHz para se estabelecer comunicação entre os robôs e o computador central. Um ESP32 faz a interface da comunicação entre o módulo e os robôs e computador por meio das portas serial RX e TX. O protocolo de comunicação utilizado constitui mensagens de 8 bits, dos quais 2 são de identificação do robô e 3 pra velocidade de cada roda.

VI. SIMULADOR INTERNO

Como a plataforma para simulação acaba sendo um dos desafios da categoria IEEE Very Small e considerando as limitações do simulador 3D previamente desenvolvido pela equipe em anos anteriores, um simulador 2D de futebol de robôs³ foi desenvolvido no Unity e integrado ao ambiente ROS utilizando o *rosbridge suite*. Além de permitir escolher o número de jogadores em cada time (entre zero e três), a cor do time aliado, modo de simulação (utilizada para testes da estratégia) e uma interface para visualizar um jogo real em ocorrência, conforme mostra a figura 7, esta plataforma permite também visualizar a trajetória mais recente dos robôs, como se vê na figura 8.



Fig. 7. Tela inicial do simulador

Outro simulador 2D foi desenvolvido pela equipe em *Python*, podendo ser encontrado neste link: https://github.com/unball/python_simulator. Nele ainda é possível realizar as mesmas tarefas do simulador antigo, controlando a posição dos robôs e da bola, analisando o comportamento da estratégia antes da implementação em hardware, entre outras. Sua execução se dá como por um

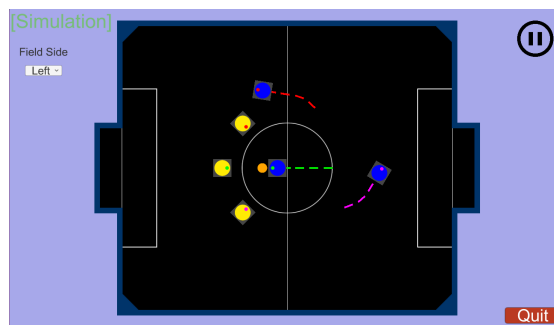


Fig. 8. Tela inicial do simulador

Processo de Decisão de Markov (MDP) e sua física é realizada com a biblioteca Box2D. Na imagem 9 é possível visualizar a interface gráfica desse simulador.



Fig. 9. Interface gráfica do simulador em python.

Devido ao cenário atual, os simuladores desenvolvidos pela equipe não serão utilizados para a competição de 2022.

VII. LOCALIZAÇÃO

A função do módulo de visão computacional é obter a posição e direção dos robôs e a posição da bola em relação ao campo. Estes valores são repassados como entrada para o módulo de estratégia, que os utiliza para especificação do comportamento dos robôs.

A visão computacional é implementada em Python 3.6, utilizando a biblioteca OpenCV na versão 3.2, e segue algumas etapas até a localização e obtenção da velocidade e da pose dos robôs ser possível.

Primeiro é necessário remover o fundo (campo) do que é reconhecido. Isso é feito primeiro delimitando o campo, depois, removendo cores próximas de branco e de preto. Assim, teremos uma imagem tratada contendo apenas o que é relevante: robôs e a bola.

Depois deixa-se apenas a bola. Como a bola é o único objeto laranja em campo, se considera apenas cor, forma não é necessário.

Por fim é necessário identificar a posição dos robôs aliados e diferenciar cada robô. Para identificar o time desejado utiliza-se as cores azul e amarelo, definidas pela categoria, e uma cor complementar. Assim pode-se identificar os robôs desejados e ignorar os robôs adversários. Para diferenciar os

¹Disponível em: <https://easyeda.com/>

²Disponível em: <https://www.sparkfun.com/products/9457>

³Disponível em: https://unball.github.io/simulator_unity

três robôs utiliza-se a forma como referência. Sobre a camisa são pregadas formas que podem ser: retângulo, triângulo e dois retângulos. Identificadas as formas podemos realizar os cálculos geométricos para o centro de massa da forma bidimensional, visando definir o ângulo de referência para o sistema.

Com isso, é possível obter a partir de cada *frame* a velocidade e posição da bola e dos robôs e também projetar movimentos futuros.



Fig. 10. Localização por visão computacional

Na Fig. 10, observa-se o funcionamento do módulo de visão computacional no seu estado atual. No quadro superior estão os resultados da câmera após simples tratamento de cor e iluminação. No quadro inferior é possível observar o sistema em funcionamento. Do lado direito estão os robôs aliados, devidamente reconhecidos e identificados pelos números 0, 1 e 2.

Nessa competição de 2022, os dados antes resultantes da visão são obtidos diretamente do simulador que será utilizado na LARC 2022. Além disso, faz-se notar que essa visão será utilizada para elaboração do Technical Challenge da CBR/LARC 2022 referente à parte de visão computacional.

VIII. CONCLUSÕES

A equipe UnBall, formada apenas por graduandos voluntários da Universidade de Brasília, e contando com o apoio da universidade e de patrocinadores, caminha para o seu oitavo

ano de participação na LARC/CBR. A equipe vem trabalhando para desenvolver um projeto sólido e obter excelentes resultados na competição, podendo assim participar de jogos do mais alto nível, mostrando bom desempenho e jogabilidade.

Os robôs estão sendo construídos em sua versão preliminar, porém, até a data da competição, vários ajustes ainda serão feitos. Portanto, com o empenho e a organização atual, espera-se que a UnBall se destaque entre as melhores equipes na LARC/CBR 2022, superando as marcas atingidas nos anos anteriores.

REFERÊNCIAS

- [1] UnBall, Tutorial Microcontroladores AVR (ATMega8). Universidade de Brasília, 2010. Disponível em: https://equipeunball.files.wordpress.com/2015/08/tutorial_avr_unball.pdf
- [2] R. C. Gonzalez, R. E. Woods, Digital Image Processing. Prentice Hall, 2008.
- [3] G. Bradski, A. Kaehler, Learning OpenCV – Computer Vision with the OpenCV Library. O'Reilly Media, 2008.
- [4] R. Siegwart, I. R. Nourbakhsh, Introduction to Autonomous Mobile Robots. The MIT Press, Cambridge, Massachusetts. 2004.
- [5] Murphy, Robin R. , Introduction to AI Robotics. The MIT Press, Cambridge, Massachusetts. 2000.