

Team Description Paper UnBall - Universidade de Brasília

Ian Moura Alexandre¹, Nikson Bernardes Fernandes Ferreira², Hiago dos Santos Rabelo³, André Dornelas Sanches⁴, Gabriel Baliza Rocha⁵, Juliana Araújo Marins⁶, Paulo Henrique Rosa da Silva⁷, Vinicius Campos Silva⁸, Maria Claudia Campos Martins⁹, Luiz Antônio Borges Martins¹⁰

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, controle e estratégia a ser utilizada na Competição Brasileira e Latino Americana de Robótica 2020 (CBR/LARC 2020).

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 este ano. 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 10 alunos de graduação em Engenharia Mecatrônica e Engenharia 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. Para isso, há a subdivisão da 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).

O restante deste documento é organizado como segue. A Seção 2 descreve a estrutura proposta para o sistema de futebol de robôs da equipe UnBall. A Seção 3 descreve de forma a base da estratégia implementada pela equipe. Já a Seção 4, o controle. Já a Seção 5, descreve o projeto e construção dos robôs para competições presenciais. A Seção 6 descreve o simulador utilizado internamente pela equipe. A Seção 7 descreve o sistema de localização por visão computacional. A Seção 8 apresenta algumas considerações finais.

¹Ian Moura Alexandre é estudante de Engenharia Mecatrônica, Faculdade de Tecnologia, Universidade de Brasília, Campus Darcy Ribeiro, Brasília, Distrito Federal, Brasil ianzeba@gmail.com

²Nikson Bernardes Fernandes Ferreira é estudante de Engenharia Mecatrônica, Faculdade de Tecnologia, Universidade de Brasília, Campus Darcy Ribeiro, Brasília, Distrito Federal, Brasil nik.son18@hotmail.com

³Hiago dos Santos Rabelo é estudante de Engenharia Mecatrônica, Faculdade de Tecnologia, Universidade de Brasília, Campus Darcy Ribeiro, Brasília, Distrito Federal, Brasil hiagop22@gmail.com

⁴André Dornelas Sanches é estudante de Engenharia Mecatrônica, Faculdade de Tecnologia, Universidade de Brasília, Campus Darcy Ribeiro, Brasília, Distrito Federal, Brasil andredornelas23@gmail.com

⁵Gabriel Baliza Rocha é estudante de Engenharia Mecatrônica, Faculdade de Tecnologia, Universidade de Brasília, Campus Darcy Ribeiro, Brasília, Distrito Federal, Brasil gabrielbaliza1@gmail.com

⁶Juliana Araújo Marins é estudante de Engenharia Mecatrônica, Faculdade de Tecnologia, Universidade de Brasília, Campus Darcy Ribeiro, Brasília, Distrito Federal, Brasil teco10e20@gmail.com

⁷Paulo 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

⁸Vinicius Campos Silva é estudante de Engenharia da Computação, Faculdade de Tecnologia, Universidade de Brasília, Campus Darcy Ribeiro, Brasília, Distrito Federal, Brasil vinicius848@gmail.com

⁹Maria 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

¹⁰Luiz Antônio Borges Martins é estudante de Engenharia da Computação, Faculdade de Tecnologia, Universidade de Brasília, Campus Darcy Ribeiro, Brasília, Distrito Federal, Brasil labm1997@gmail.com

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.

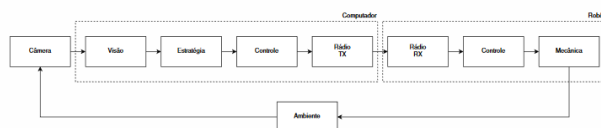


Fig. 1. Sistema UnBall

Primeiramente sobre o módulo "computador". A visão computacional e a estratégia serão discutidas mais a frente, 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.

Depois sobre o módulo "robô", no rádio RX recebe-se as velocidades angulares e lineares referentes a todos os robôs, e são repassadas ao controle as velocidades corretas baseadas em seu próprio identificador. 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 sub-módulo mecânica é onde estão os componentes físicos, que devem alcançar altas velocidades e grande torque.

Para a competição não presencial o sistema torna-se mais simples. A estratégia recebe as posturas e velocidades do simulador FIRASim. Com esses valores a estratégia decide a função, movimento e trajetória de cada robô. O controle retorna a velocidade angular e linear necessária para alcançar o objetivo, se adaptando a possíveis interferências futuras. Depois o controle retorna as velocidades para o FIRASim, que transmite para os robôs simulados.

III. ESTRATÉGIA

O sistema de estratégia tem como objetivo principal regimentar o comportamento dos robôs durante a partida, de forma que estes ajam em equipe de forma a vencer a partida. O fundamento por trás da estratégia da equipe é o conceito de "entidades", a qual "possuem" os robôs nos momentos apropriados.

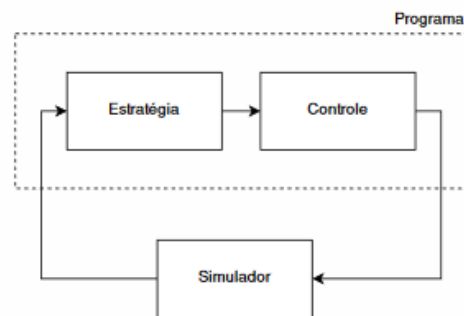


Fig. 2. Sistema UnBall - Não presencial

Cada entidade possui um conjunto de movimentos possíveis, os quais definem uma *pose*, posição linear e angular, a qual o robô deve estar em um futuro próximo.

Existe uma variedade infinita de formas de percorrer o caminho até atingir tal *pose* ou postura. De forma que, define-se campos vetoriais unitários para a navegação;

O algoritmo chamado Univector Field, campo vetorial unitário bidimensional que determina a postura do robô em cada ponto do campo, considerando o objetivo (que é definido pela função que cada robô pode ocupar) e possíveis obstáculos. O objetivo é tratado como um campo atrativo. Junto do campo é feito o controle do sistema, que corrige a trajetória levando em conta as limitações físicas do robô (sendo essas limitações reais ou virtuais).

Tais campos se assemelham com retratos de fase dos sistemas dinâmicos. Dessa forma, para o equacionamento dos diferentes campos, utilizamos equações dinâmicas conhecidas.

IV. CONTROLE

O controle é responsável por garantir que o sistema siga uma referência com o desempenho desejado. Para tal, a equipe optou por realizar um controle em cascata. De forma que, 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 3.

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.

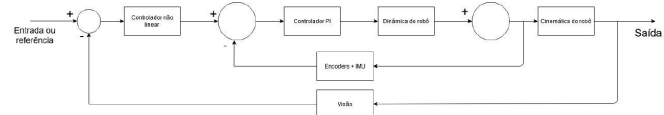


Fig. 3. Esquema ilustrativo de sistema de controle

A. MALHA EXTERNA

A malha mais externa é responsável por alinhar o robô com o campo de vetores unitários e 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 \dot{e} :

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

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

$$\omega = \dot{\theta}_r + k e_\theta \quad (5)$$

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

$$\dot{L} = -k e_\theta^2 \quad (6)$$

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 de forma a respeitar as limitações físicas dos robôs, ao estar com a velocidade angular definida acima.

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, pois é realizável, possui erro nulo a degraus de referência e perturbação e executado em um microcontrolador através de uma equação de diferenças.

O controle dos motores é feito de forma acoplada, sendo alimentado pelo giroscópio e *encoders* 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 em desenvolvimento pela equipe. Não serão utilizados no evento LARC/CBR 2020, mas são uma áreas de atuação importantes para o projeto.

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 foi do modelo 1524b009sr da Faulhaber, com uma caixa de redução 19:1 que já vem de fábrica, a qual 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. Como este ano a competição será a distância não se focou no projeto mecânico, mas na estratégia. Na figura 4 é possível observar estrutura do projeto atual, e na figura 5 uma versão do projeto para próximas competições presenciais, feito no software FreeCAD.

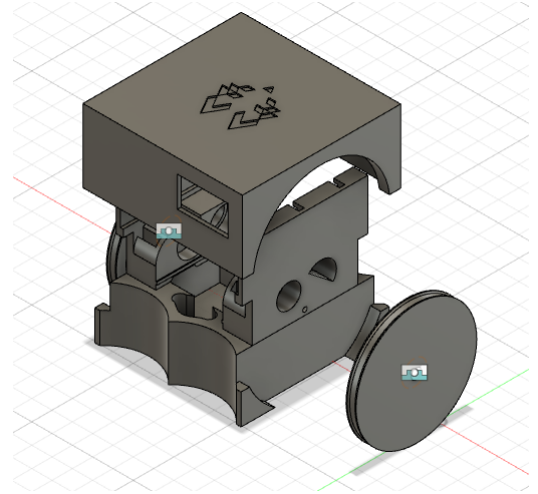


Fig. 4. Estrutura do robô atual

A estrutura do robô é majoritariamente feita em impressão 3D utilizando plástico PLA que fornece resistência mecânica boa o suficiente para a aplicação, e é mais fácil de se trabalhar em comparação com outros polímeros como 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 podem ser alteradas sem que afete o restante da estrutura.

B. Projeto Eletrônico

O projeto eletrônico compreende o acionamento e controle dos motores, a definição e implementação do protocolo de comunicação com o computador central, o projeto da placa

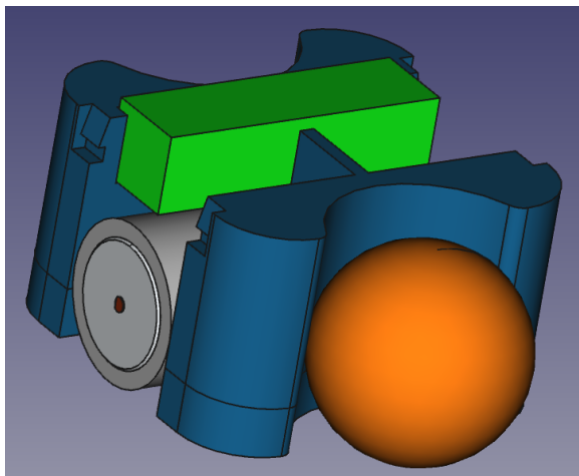


Fig. 5. Projeto para próximas competições presenciais

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 EAGLE - PCB Design Software¹, que permite a adição de diversos componentes eletrônicos e a realização de simulação, para verificar o devido funcionamento dos circuitos.

Utiliza-se o microcontrolador TEENSY 3.0, pela sua velocidade 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, utilizou-se baterias lipo 3s de tensão nominal de 11,1v e carga máxima de 850mah.

Utilizou-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. O Arduino faz a interface da comunicação entre o xbee 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.

Também é utilizado um IMU (Giroscópio e acelerômetro) GY-521 MPU-6050 para correção de posição e velocidade angular, utilizando um controle em baixo nível.

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*. Este simulador além de

permitir escolher quantos jogadores, entre zero e três inclusive, estarão em campo em cada time durante uma simulação, qual a cor do time aliado e se o simulador deve rodar em modo de simulação - utilizada para testes da estratégia - ou como uma interface para visualizar um jogo real em ocorrência, conforme mostra a figura 6, esta plataforma permite também visualizar a trajetória mais recente dos robôs, como se vê na figura 7

Este não é o simulador que será utilizado na competição remota, mas é de grande ajuda no desenvolvimento da estratégia da equipe.



Fig. 6. Tela inicial do simulador

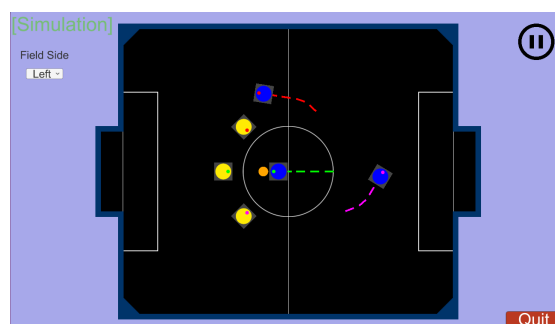


Fig. 7. Tela inicial do simulador

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 a 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.

¹Disponível em: <http://caddesigner.com.br/br/>

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

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

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 três robôs utiliza-se forma. Sobre a camisa são pregadas formas que podem ser: retângulo, triângulo e dois retângulos. Identificadas as formas podemos realizar cálculos geométricos quanto ao centro de massa da forma bidimensional para 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. 8. Localização por visão computacional

Na Fig. 8, 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.

Este ano, em um caso atípico os dados que seriam resultantes da visão são obtidos diretamente do sistema (simulador) que será utilizado durante a competição.

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 sexto 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.

A estratégia está sendo reconstruída, portanto, 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 2020, superando a 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.