RobôCIn IEEE Very Small Size Soccer 5v5
LARC 2022 Team Description Paper

Arthur Silva, Breno Cavalcanti, Carlos Silva, Elisson Araújo, Erick da Silva, Felipe Martins, Geovany Nunes, José Douglas Silva, Leonardo Silva, Luisa Cavalcante, Mariana Barros, Mateus Machado, Matheus Albuquerque, Morgana Galamba, Pedro Coutinho, Pedro Oliveira, Pedro Silva, Ryan Morais, Thiago Souza, Edna Barros[1]

***Abstract*— Este *Team Description Paper* tem como objetivo descrever o projeto desenvolvido pela equipe RobôCIn, do Centro de Informática da UFPE, para participação na competição IEEE *Very Small Size Soccer 5v5*. Neste *Team Description Paper* são descritos os principais sistemas desenvolvidos: o sistema de localização por visão computacional, o sistema de tomadas de decisão, bem como a comunicação entre os robôs, o sistema de controle, e o sistema eletrônico e mecânico do robô projetado.**

I. INTRODUÇÃO

Para participar da competição da categoria IEEE *Very Small Size Soccer* (VSSS), são necessários cinco pilares essenciais: detecção de objetos, tomada de decisões, planejamento de caminhos, controle, mecânica, eletrônica e comunicação. O primeiro estágio do ciclo de execução do projeto é a detecção de objetos. Nesta etapa, a equipe optou por adaptar um módulo já existente da categoria de Small Size League para o VSSS 5v5. Após a etapa de detecção, as informações de posição, direção e velocidade são passadas para o módulo de estratégia. O módulo de estratégia define um caminho a ser seguido pelos robôs, e assim calcula as velocidades a serem enviadas para o módulo de controle do robô (*hardware* embutido no robô). Com as velocidades definidas, o módulo de controle local do robô realiza o controle do robô.

Na seção II, é explicado o módulo de localização dos objetos em campo. Na seção III, é detalhada tomadas de decisões realizadas pelo módulo de estratégia. Em seguida as seções IV e V descrevem a mecânica e eletrônica dos robôs utilizadas pela equipe e a seção VI que explica a comunicação entre os módulos. Em seguida, são apresentadas as seções de conclusão, agradecimentos e referências.

II. VISÃO

O módulo de visão é responsável por coletar a posição do robô em campo. Embora a competição deste ano ocorra em um ambiente simulado, onde este módulo não é necessário, nossa equipe trabalhou neste módulo para nos permitir usar nossos robôs no mundo real.

Ao longo dos anos de participação na competição, desenvolvemos um sistema de visão computacional rápido e robusto para o ambiente VSSS chamado MaggicVision [1]; no entanto, desenvolvemos o sistema para o ambiente 3v3 e atualmente estamos adaptando e testando o sistema para a configuração 5v5.

Para permitir uma transição mais rápida para testes do mundo real, investigamos o uso do software SSL-vision [2] em um ambiente VSSS 5v5 como um módulo de visão provisório. Adaptamos nosso software para receber as informações de visão por meio de mensagens de rede, possibilitando executar o sistema de visão como um modulo separado. Também alteramos nossas *tags* de identificação do robô para um padrão mais adequado aos requisitos do SSL-vision, mostrado na Fig. 1, permitindo reunir as informações de id e ângulo do robô, além da equipe e posição.

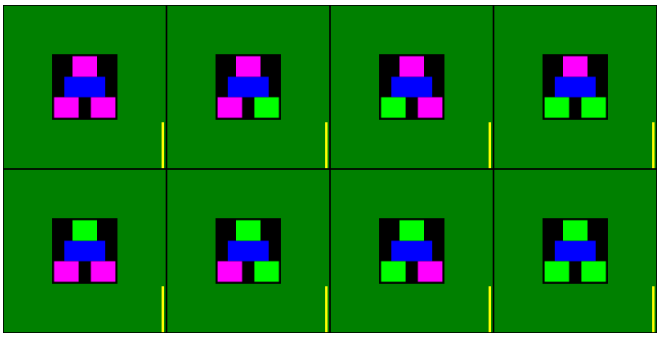


Fig. 1. Novos padrões de *tags* adaptados para o SSL-vision.

III. ESTRATÉGIA

O módulo de estratégia é responsável por determinar as melhores ações e caminhos a serem tomados pelos robôs, utilizando como fontes de informação a posição, orientação e velocidade dos robôs e da bola presentes em campo, informações essas que são fornecidas pelo módulo de visão. É importante que o módulo de estratégia seja rápido e preditivo, de forma a prover uma reação rápida dos robôs em função das situações de jogo e assim garantir uma maior probabilidade de vitória.

Para a competição desse ano, o RobôCIn vai unificar suas equipes de VSSS, a equipe de abordagens determinísticas e a de aprendizado profundo. Na seção III-A descrevemos nossa formação durante todo o jogo, o primeiro modulo no nosso processo de decisão, em seguida descrevemos os nossos comportamentos de atacante de aprendizado por reforço e

[1]Centro de Informática - Universidade Federal de Pernambuco, Av. Jornalista Anibal Fernandes, s/n - CDU 50.740-560, Recife, PE, Brasil. robocin@cin.ufpe.br

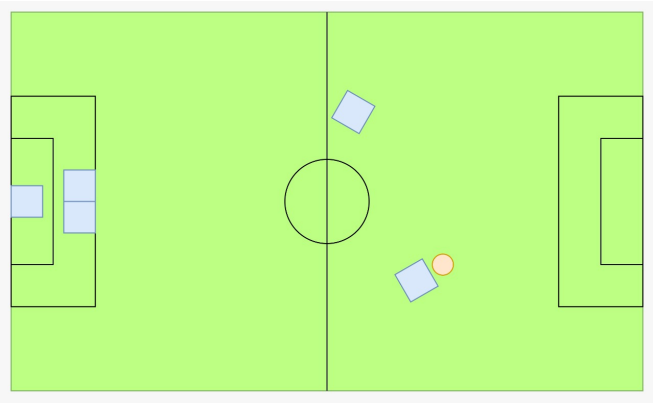


Fig. 2. Formação da equipe RobôCIn.

determinísticos nas seções III-B e III-C respectivamente, na seção III-D descrevemos nossa implementação de posicionamento automático, que pretendemos utilizar pela primeira vez nesta competição para aumentar a autonomia da nossa estrategia.

A. ESCOLHA DE FORMAÇÃO

Para explicar a camada de escolha de formação, é necessário primeiramente introduzir o conceito de comportamento. O comportamento de um robô é definido como o conjunto de relações entre as ações que o robô pode tomar e as situações em que cada ação é tomada. Sua definição é análoga à definição de função de um jogo de futebol. Por exemplo, semelhante a um goleiro de um time de futebol que, ao perceber que a bola está se aproximando de seu gol, deve ir em direção à bola com objetivo de desviar sua trajetória, um robô programado com o comportamento de goleiro deve executar ação semelhante em campo.

Para gerir as diferentes combinações de comportamentos, que denominamos como formações, com pequenas variações no posicionamento de campo e nas tomadas de decisão, adicionamos uma camada extra, pois precisamos escolher primeiro, qual a formação seria o mais eficaz dada a condição atual do jogo.

Para a categoria de VSSS 5v5, adotamos uma formação muito parecida com a formação básica da categoria do *Small Size League* (SSL). A ideia é ter um goleiro, uma zaga fixa com dois robôs, um suporte e um atacante. Ilustramos a nossa formação na Figura 2. Esta decisão de formação foi retirada diretamente das nossas análises de jogos da LARC 2021 e da maioria dos jogos de SSL.

B. ATACANTE DE APRENDIZADO POR REFORÇO

Tendo em vista a ótima performance do nosso agente de atacante treinado com aprendizagem por reforço no VSSS 3v3, teremos também o mesmo agente para um jogo de VSSS 5v5. Para o treinar o agente, teremos que construir um novo ambiente de treinamento para o campo de 5v5. Utilizaremos a *framework* rSoccer [3] para fazê-lo. As recompensas e estados serão os mesmos com a diferença das recompensas serem normalizadas, algo que notamos que melhora o aprendizado do agente em vários outros ambientes [4] [5].

O algoritmo que utilizaremos durante o treino será o Deep Deterministic Policy Gradients (DDPG) [6], por mostrar maior exploração de estados e maior taxa de gols nos experimentos realizados no VSSS 3v3 [3]. Futuros estudos com métodos como Proximal Policy Optimization [4] e Soft Actor-Critic [5] também serão feitos.

Para integrar no *software* de unificação, utilizamos a *framework* Pytorch [7] para salvar e carregar os parâmetros da rede atuante no ambiente que usamos no agente de atacante durante o jogo. Um tutorial geral de como fazemos isto está descrito em `https://pytorch.org/tutorials/advanced/cpp_export.html`.

C. AGENTES DETERMINÍSTICOS

Nesta categoria, teremos três tipos de agentes determinísticos: goleiro, zagueiro e suporte. O goleiro será um agente básico que seguirá a bola no eixo y até os limites do gol. Este comportamento aumentou a taxa de defesas em 80% nos jogos do VSSS 3v3 e pretendemos utilizar o mesmo nesta categoria.

Existirão também dois zagueiros, como mostramos anteriormente na seção III-A. Os zagueiros assumem uma posição apenas defensiva, sem avançar ou trocar de papel com nenhum outro robô. O comportamento desses agentes é semelhante ao do goleiro, porém um pouco mais adiantado e seguindo uma curva parabólica. Veja na Figura 3 o comportamento dos zagueiros em função da posição da bola.

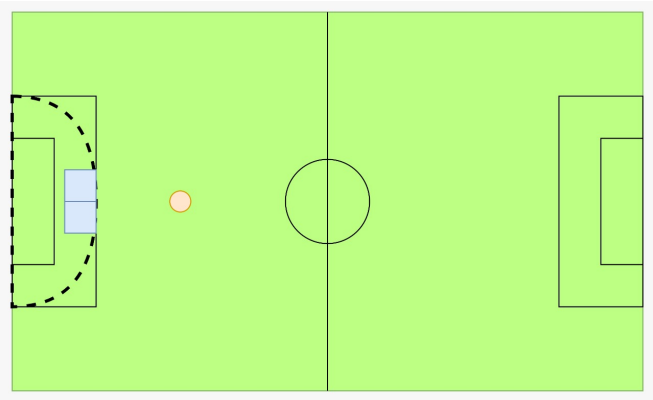
O nosso agente suporte faz o papel de coringa, apenas esperando a bola "sobrar" para trocar de papel com o atacante e tentar fazer um gol. O posicionamento desse jogador é definido pela posição futura da bola em N passos. Faremos um estudo exploratório para saber quantos passos são necessários para ter a troca ótima de papéis entre o atacante e o suporte. Ilustramos na Figura 4 o posicionamento do suporte.

Estes agentes contam com um modelo básico de controle. Dado o objetivo e angulação instantâneos para nosso agente, é necessário mapear tais valores para velocidades de rodas que produzam o resultado desejado. Para tal, é utilizado um controle PD como mostrado nas equações 1, 2, 3, 4, que mapeiam as velocidades necessárias para que o robô saia do ponto R para o ponto O, onde θ_{ref} é o ângulo desejado, θ é o ângulo atual do robô, ϵ_θ é o erro angular, D é a distância euclidiana entre o robô e o ponto objetivo e os valores K_p^θ, K_d^θ, K_p^v, K_d^v são constantes que dependem de fatores físicos do robô.

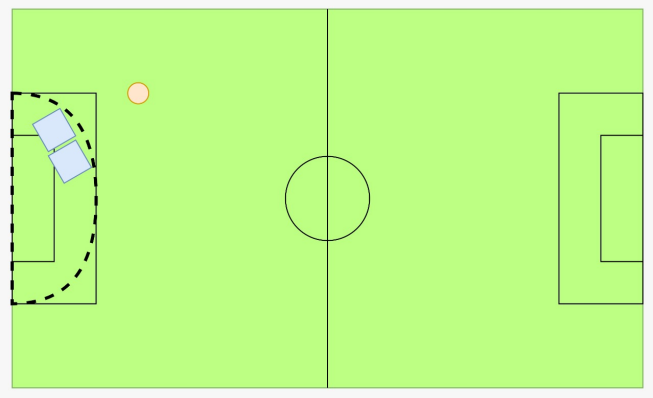
$$\epsilon_\theta[t] = \theta[t] - \theta_{ref}[t] \tag{1}$$

$$\omega = K_p^\theta \epsilon_\theta[t] + K_d^\theta(\epsilon_\theta[t] - \epsilon_\theta[t-1]) \tag{2}$$

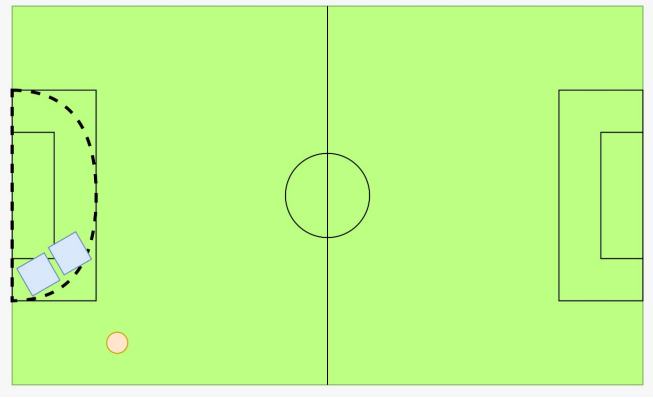
$$D[t] = \sqrt{(R_y[t] - O_y[t])^2 + (R_x[t] - O_x[t])^2} \tag{3}$$



(a)



(b)



(c)

Fig. 3. Comportamento dos zagueiros em função da posição da bola.

$$v = K_p^v[t]\epsilon_v + K_d^v(D[t] - D[t-1]) \quad (4)$$

Para compensar o *delay* da câmera na tomada de decisão, implementamos um sistema de previsão que funciona usando o método de Runge-Kutta. Tal método se baseia em equações que levam em consideração o ângulo que o robô está se movimentando e sua velocidade para predizer onde o robô estará. A equação 5 calcula o ângulo futuro que o robô vai estar com base no seu ângulo atual e sua velocidade angular. As equações 6 e 7 usam a média entre o ângulo atual e o futuro, a posição atual e a velocidade na coordenada para

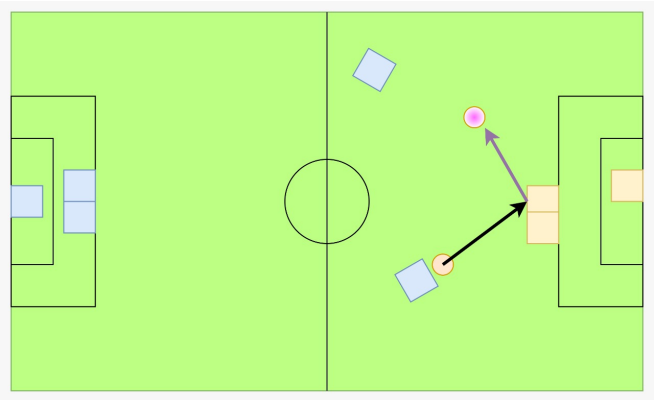


Fig. 4. Posicionamento do agente suporte. O ideal é que o agente esteja numa posição em que seja fácil ir de encontro com o ponto futuro da bola, ilustrado pelo círculo rosa.

prever a posição futura dos objetos em campo.

$$\theta(t + \Delta t) = \theta t + \Delta t.wt \quad (5)$$

$$x(t + \Delta t) = x(t) + \Delta t.\cos((\theta.t + \theta(t + \Delta t))/2).V_x(t) \quad (6)$$

$$y(t + \Delta t) = y(t) + \Delta t.\cos((\theta.t + \theta(t + \Delta t))/2).V_y(t) \quad (7)$$

D. POSICIONAMENTO AUTOMÁTICO

Sendo o VSSS uma categoria de robótica autônoma, a visão é que tenhamos partidas com a menor quantidade de interferência humana possível. Com o objetivo de autonomia em mente, implementamos uma estratégia de posicionamento automático para que os robôs se preparar durante as interrupções da partida na posição ideal para o reinicio, sem que seja necessária uma interferência. Abordamos esse problema como a sequência de decisão de seleção de posicionamento desejado, atribuição de posições para cada robô, planejamento de caminho e controle.

Esta implementação é um exemplo dos benefícios que a unificação de softwares entre as categorias dentro do RobôCIn nos trouxe, o desafio de posicionamento é um problema que abordamos a fundo no *Small Size Soccer*, e conseguimos rapidamente adaptar as soluções já desenvolvidas para esta categoria. A seleção de posicionamento é feita utilizando posições pré definidas para cada situação de jogo, tendo um conjunto de posições objetivo, atribuímos a cada robô uma posição utilizando algoritmo de otimização combinatória *Hungarian* [8]. Por fim, tendo os conjuntos de robôs e posições objetivo, podemos utilizar abordagens de planejamento de caminhos e controle. A abordagem usada pela equipe para o planejamento de caminho consiste em definir o caminho a ser seguido utilizando um campo de vetores unitários [9], que podem ditar a direção de movimento do robô.

IV. MECÂNICA

Durante o hiato de 3 anos de competições presenciais, uma análise sobre aspectos tanto ergonômicos quanto de performance foram realizados, através do estudo dos robôs desenvolvidos na LARC e no MIROSOT [10]. Em suma, os aspectos gerais para o desenvolvimento do novo robô permeiam três fatores: Caixa de redução, sistema de pontos de contato e otimizar as constantes físicas, estas descritas nas subseções IV-A, IV-B, e IV-C, respectivamente.

Na figura 5a, está o protótipo resultante dessa análise composto dos seguintes módulos: chassi de impressão 3D, mancais, rodas 50mm, motores DC 22mm, baterias de 300mAh, placa do sistema embarcado e capa.

A. Caixa de redução

Na mecânica de 2019 [11], o eixo dos motores eram diretamente ligados à roda por conta de sua caixa de redução inclusa. Já o atual por sua utiliza uma transmissão descentralizada do pinhão de 16 dentes do motor para a engrenagem interna da roda, em uma redução de 5:1 (Figura 5b). Essa nova disposição dos motores traz impacto direto no centro de massa do robô.

B. Pontos de contato

Para carregar a bola, o formato elipsoidal se provou mais eficiente em sua função. Com isso, foi proposto um sistema de três toques [12], e dessa maneira, quando a bola tender a sair desse sistema, a elipse inferior gera um atrito que a recentraliza (Figura 6a).

O sistema de três pontos de toque também é utilizado para as tangências do robô no campo, através esferas de rolamento entre as elipses, garantindo estabilidade nos toques junto às rodas durante a movimentação (Figura 6b).

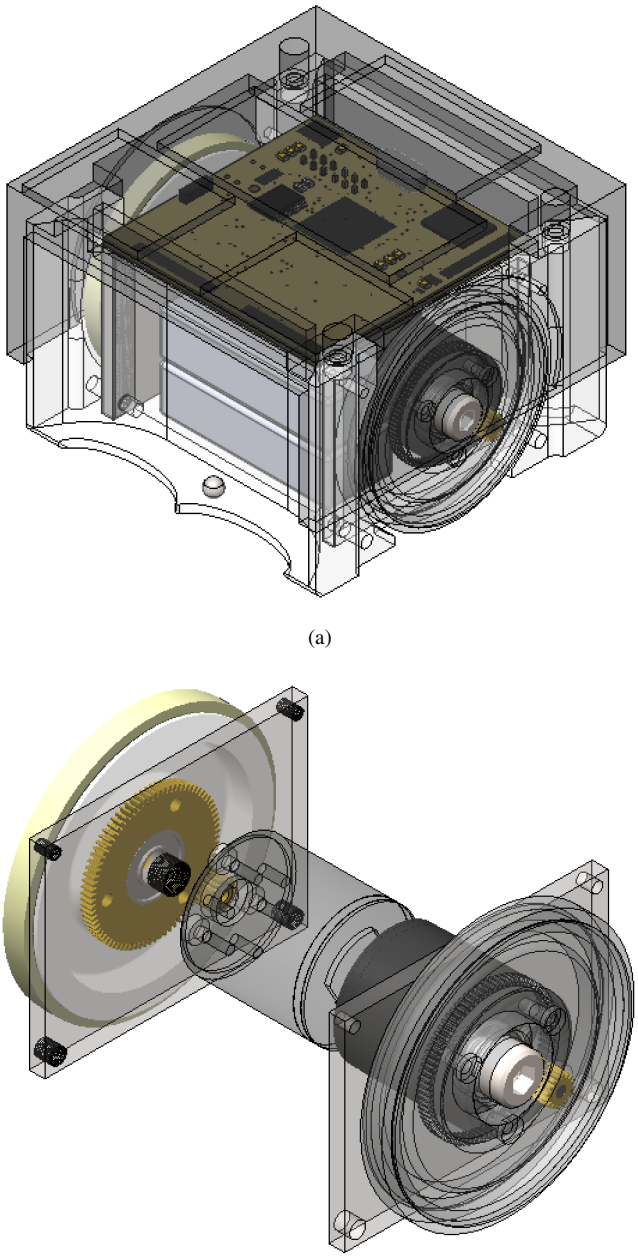
C. Constantes Físicas

O robô da equipe RobôCIn de 2022 permite 1380 RPM com torque de 2,16 Kg-cm. Ademais, é de suma importância o coeficiente de atrito da roda, pois é a única constante manipulável para alterar a aceleração máxima do robô [13]. Esse coeficiente pode ser alterado através da dureza e tempo de cura do material do pneu da roda.

V. ELETRÔNICA

Atualmente, o sistema embarcado desenvolvido para os robôs se baseia no microcontrolador ATmega328. Neste são utilizados dois microcontroladores, sendo um deles configurado como mestre e o outro como escravo. Eles foram escolhidos pela praticidade, tamanho, e capacidade de controlar todos os dispositivos requisitados no robô, no entanto um novo sistema baseado no microcontrolador STM32F446 está em desenvolvimento. A criação deste novo sistema se justifica pelo ganho de desempenho trazido pelo processador do tipo ARM e, também, por este anular a necessidade de serem utilizados dois microcontroladores no mesmo robô.

Para controle de potência dos dois motores DC presentes no robô, é utilizado o *driver* de motor TB6612FNG, que possui limite de corrente contínua de 1A por motor e 3A para corrente de pico. Além disso, os motores contam com *encoders* para que possa ser feito um controle de rotação mais preciso por meio de *software*. Além disso, o circuito do robô também é composto pelo nRF24L01+, um módulo *wireless*, responsável pela comunicação com o computador.

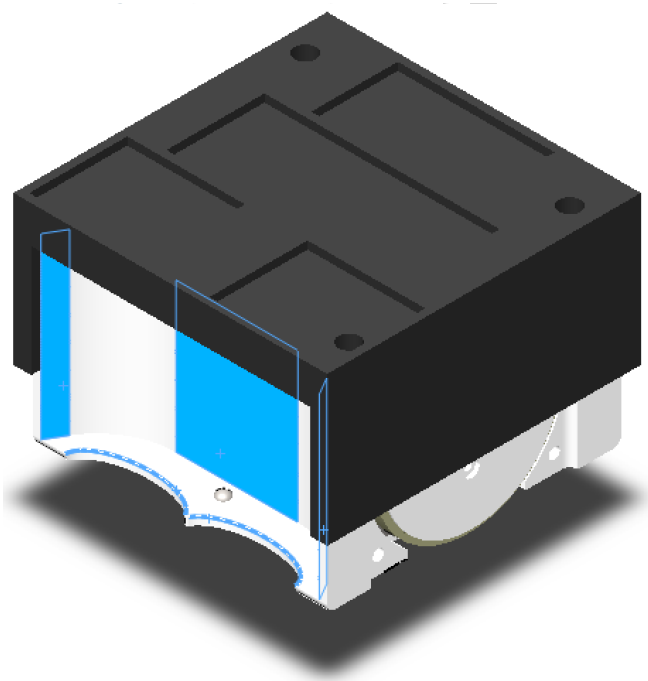


(a)

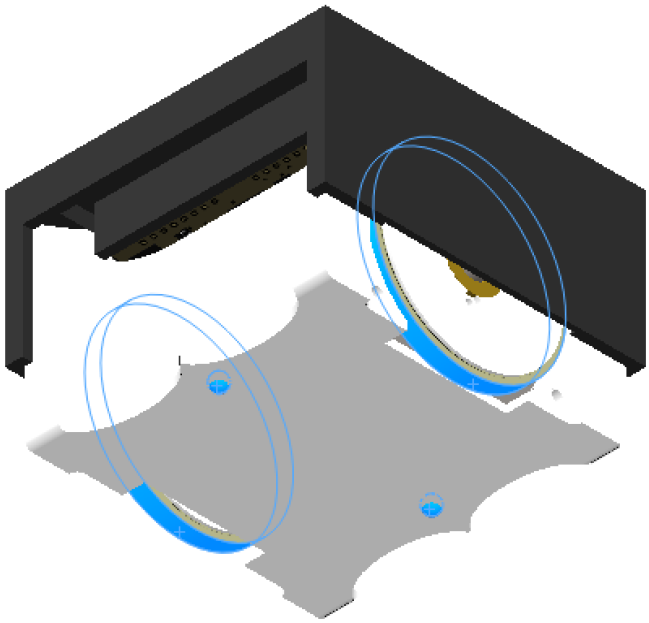
(b)

Fig. 5. (a) Mecânica RobôCIn 2022; (b) Unidade de potência;

A fim de otimizar o espaço do robô, os circuitos foram impressos em placas de dupla camada. Para fazer os layouts dos circuitos impressos foi utilizado o *software* de desenvolvimento de placas de circuito impresso Eagle [14]. A placa principal foi desenvolvida conforme Figura 7 para compor a parte superior do robô.



(a)



(b)

Fig. 6. (a) Toques com a bola; (b) Toques com o campo;

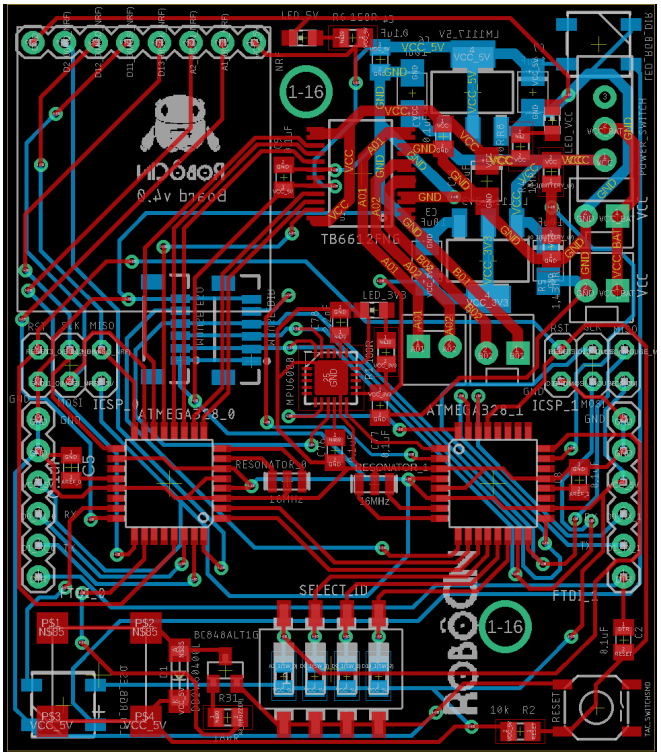


Fig. 7. Projeto do circuito principal feito através do Eagle Circuit CAD.

VI. COMUNICAÇÃO

Para realizar a comunicação entre os robôs e o computador, na competição do VSSS em 2017 o circuito utilizado permitia o uso tanto do xBee Series 2 quanto o do módulo nRF24L01+. Depois de testar o desempenho do módulo nRF24L01+ nos jogos, julgou-se melhor optar pelo uso único dele e anular o uso do xBee Series 2 devido aos problemas dispostos pelo mesmo na competição VSSS de 2016. Assim, a partir da competição de 2018, a equipe utiliza apenas o módulo nRF24L01+ para comunicação.

Ao decidir usar a comunicação do tipo radiofrequência (RF), a equipe implementou um protocolo através da classe *NRF24_Communication*, onde foram utilizadas as funções nativas do *RF24*. A comunicação foi desenvolvida de modo que incluísse não somente o envio de mensagens do computador para o robô, mas também a resposta do robô para o computador.

O módulo utilizado, nRF24L01+, se comunica com o microcontrolador seguindo o protocolo *Serial Peripheral Interface* (SPI). Assim, é utilizado um microprocessador ARM e dois módulos RF para a comunicação por parte do computador, onde um módulo atua como receptor, e o outro como transmissor. Contudo, no robô é utilizado um único módulo RF que recebe os dados que o computador enviou e ao ser solicitado envia mensagens de resposta para o computador.

Para realizar a comunicação, é seguido um fluxo como está mostrado na Figura 8. Inicialmente, o computador processa a informação recebida através de imagens e converte nos movimentos que devem ser realizados pelos robôs. Para

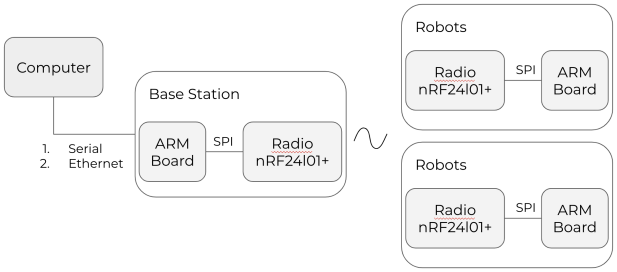


Fig. 8. Arquiteura da comunicação entre computador e robôs.

acessar o módulo nRF24l01+, o computador é conectado a uma "base station" através de uma interface serial ou Ethernet. Esta "base station" consiste de um sistema embarcado composto por uma placa ARM conectada por interface SPI a dois módulos nRF e que permite a sua comunicação com o computador. Em cada um dos robôs, um módulo nRF é utilizado para enviar ao robô as informações do computador sobre o movimento que deve realizar. Caso haja necessidade de resposta, o robô as envia para o computador também pelo seu módulo RF.

VII. SIMULAÇÂO

Como a competição deste ano vai continuar acontecendo em um ambiente simulado, foi necessário adaptar nosso código. Acreditamos que ajustar nossa estratégia e parâmetros serão cruciais para o desempenho geral da equipe.

Adaptar nosso código para se comunicar com o simulador foi simples, pois já o usamos para testar nossos desenvolvimentos. Como já havíamos adaptado nosso código para trabalhar com um sistema de visão por meio de mensagens de rede, ele era intercambiável com o simulador. Aprimoramos também nosso código para robustez de comunicação, revisamos o comportamento do código para tratar as possibilidade em que os pacotes de rede podem ser perdidos ou acumulados.

Para nossos comportamentos de aprendizado por reforço, usamos a estrutura rsoccer [3] para treinamento; ele usa o simulador rsim, uma versão modificada dos simuladores FIRA/grSim [15]. Como possui parâmetros físicos diferentes, usamos técnicas de ajuste fino para adaptar os comportamentos aprendidos para o simulador de competição

VIII. CONCLUSÃO

No presente TDP, é mostrado como foi projetado e desenvolvido o sistema necessário para participar na categoria VSSS. Para isso, foi descrito um sistema de localização e detecção de objetos com objetivo de localizar a bola e os robôs em campo, um planejador de caminhos e decisor de comportamentos a fim de permitir a elaboração de estratégias de jogo, o projeto mecânico e eletrônico dos robôs e a comunicação entre eles e o sistema de controle. Também foi descrito o simulador utilizado para testes e desenvolvimento, e as mudanças realizadas no mesmo para comunicação com o *software* desenvolvido.

AGRADECIMENTOS

A equipe gostaria de agradecer o Centro de Informática da UFPE pelo apoio financeiro e de recursos durante todo o processo do projeto, à todo apoio dado pelos professores Edna Barros e Hansenclever Bassani. Também agradecemos a ajuda dos nossos patrocinadores: CESAR, Microsoft, Moura, HSBS, and Veroli Transportadora.

REFERENCES

[1] L. Maggi, "Maggicvision: A fast and lighting-invariant computer vision system for ieee very small size soccer," Tese de Graduação, Universidade Federal de Pernambuco, 2021.

[2] S. Zickler, T. Laue, O. Birbach, M. Wongphati, and M. Veloso, "Ssl-vision: The shared vision system for the robocup small size league," in *Robot Soccer World Cup*. Springer, 2009, pp. 425–436.

[3] F. B. Martins, M. G. Machado, H. F. Bassani, P. H. M. Braga, and E. S. Barros, "rsoccer: A framework for studying reinforcement learning in small and very small size robot soccer," in *RoboCup 2021: Robot World Cup XXIV*, R. Alami, J. Biswas, M. Cakmak, and O. Obst, Eds. Cham: Springer International Publishing, 2022, pp. 165–176.

[4] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov, "Proximal policy optimization algorithms," 2017. [Online]. Available: https://arxiv.org/abs/1707.06347

[5] T. Haarnoja, A. Zhou, K. Hartikainen, G. Tucker, S. Ha, J. Tan, V. Kumar, H. Zhu, A. Gupta, P. Abbeel, and S. Levine, "Soft actor-critic algorithms and applications," 2018. [Online]. Available: https://arxiv.org/abs/1812.05905

[6] T. P. Lillicrap, J. J. Hunt, A. Pritzel, N. Heess, T. Erez, Y. Tassa, D. Silver, and D. Wierstra, "Continuous control with deep reinforcement learning," 2015. [Online]. Available: https://arxiv.org/abs/1509.02971

[7] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, A. Desmaison, A. Kopf, E. Yang, Z. DeVito, M. Raison, A. Tejani, S. Chilamkurthy, B. Steiner, L. Fang, J. Bai, and S. Chintala, "Pytorch: An imperative style, high-performance deep learning library," in *Advances in Neural Information Processing Systems 32*, H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, Eds. Curran Associates, Inc., 2019, pp. 8024–8035. [Online]. Available: http://papers.neurips.cc/paper/9015-pytorch-an-imperative-style-high-performance-deep-learning-library.pdf

[8] H. W. Kuhn, "The hungarian method for the assignment problem," *Naval research logistics quarterly*, vol. 2, no. 1-2, pp. 83–97, 1955.

[9] Y.-J. Kim, J.-H. Kim, and D.-S. Kwon, "Evolutionary programming-based univector field navigation method for past mobile robots," *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, vol. 31, no. 3, pp. 450–458, 2001.

[10] C. F. Pana, N. G. Bizdoaca, I. C. Rescanu, and M. Niculescu, "Strategy planning for mirosot soccer's robot," 2008.

[11] F. B. Martins, H. R. de Medeiros, L. H. C. Santos, M. G. Machado, P. H. M. Braga, R. de Azevedo Delgado, and E. N. da Silva Barros, "Robocin ia description paper," 2020.

[12] Z. Huang, L. Chen, J. Li, Y. Wang, Z. Chen, L. Wen, J. Gu, P. Hu, and R. Xiong, "Zjunlict extended team description paper for robocup 2019," *arXiv preprint arXiv:1905.09157*, 2019.

[13] D. Otten, "Building mitee mouse iii," *Circuit Cellar INK*, 1990.

[14] Autodesk, Inc., "Eagle."

[15] V. Monajjemi, A. Koochakzadeh, and S. S. Ghidary, "grsim–robocup small size robot soccer simulator," in *Robot Soccer World Cup*. Springer, 2011, pp. 450–460.