

Project Neon 2022 Very Small Size Soccer

Alexsandro Francisco dos Santos¹, Ana Julia Monici Orbetelli², Carlos Henrique Lima Santos³,
Eduardo Yukio Makita⁴, Giovanni Rodrigues Dias⁵, Gabriel Mendes de Lima⁶,
Guilherme Carvalho Torres⁷ João Pedro Kayano Leal⁸ Josias Calebe Pereira Rodrigues Gonçalves⁹
Luíza Módolo Gonçalves¹⁰, Marcos Augusto Pacheco¹¹,
Marcus Vinicius Holanda de Lima¹², Matheus Cardoso da Silva¹³,
Thales Machado Fernandes¹⁴, Victor Hugo Lourenço¹⁵,

Resumo—No ano de 2020, a equipe decidiu reescrever o algoritmo de tomada de decisão usado na modalidade IEEE VSSS, tomando uma abordagem mais modular, assim como uma plataforma com uma separação entre seus componentes que permitisse um desenvolvimento mais simples num contexto de um time maior. Em 2021, a equipe teve sua primeira experiência com a modalidade 5v5, com um time muito defensivo. Entre 2021 e 2022, a equipe decidiu repensar as estratégias usadas para o 5v5 virtual, alterando a função e posicionamento de alguns dos robôs com o objetivo de criar um time 5v5 mais ofensivo. Este artigo descreve a evolução do NeonFC como um software de tomada de decisões para a modalidade IEEE VSSS 5v5 virtual.

I. INTRODUÇÃO

A Equipe foi fundada em 2015 por alunos da Universidade Federal do ABC e é atualmente composta por 40 membros de diferentes áreas do conhecimento.

Um dos pilares mais importantes adotados pelo time é o conceito de *open-source*: todo trabalho desenvolvido é aberto, seja hardware ou software, para uso de qualquer um, permitindo uma rede de colaboração e aprendizado aberto.

Desde do início dessa mudança de paradigma causado pela pandemia, a equipe tem usado das modalidades e torneios junto a seu processo interno para se reinventar, de forma a pensarmos em projetos mais bem desenvolvidos, documentados e modulares. De forma que possamos trabalhar de forma assíncrona, nos dividindo em times pequenos e reduzindo a barreira técnica inicial que uma modalidade como esta apresenta.

No ano de 2020, concluímos a primeira iteração do desenvolvimento de NeonFC[1], no qual um dos principais desafios foi criar uma divisão coerente das diferentes áreas

do conhecimento dentro do código. Em 2021, utilizamos essa primeira versão para pensar num desenvolvimento mais paralelo, com mais pessoas atuando de forma contínua no código ao mesmo tempo que os novos membros da equipe aprendem sobre conceitos básicos da computação e mais especificamente da robótica. Por fim, em 2022, o foco passou a ser repensar as estratégias do time para melhorar nosso desempenho no IEEE VSSS simulado, além de garantir que a maior parcela possível do NeonFC poderia vir a ser compatível com os novos robôs físicos.

II. SOFTWARE

O NeonFC é um software desenvolvido em Python 3. Pensado para ser uma plataforma de desenvolvimento para as modalidades IEEE VSSS e IEEE VSSS virtual. O software atua com um esqueleto base, o que chamamos de main loop. main loop define as principais ações que um código de tomada de decisão para esta modalidade deve fazer.

Algorithm 1 Main Loop

```
1: function MAIN LOOP(frame)  
2:   match.update(frame)  
3:   commands ← match.decide()  
4:   communication.send(commands)  
5: end function
```

No algorithmic 1 podemos ver dois elementos relevantes. *match* e *communication*. Esses são dois módulos do NeonFC. esse main loop também recebe o parâmetro *frame* que é um objeto com todos os dados de uma captura enviada pelo ambiente.

Os módulos dentro do NeonFC servem para que possamos trocar o comportamento de forma simples, adaptando o código para diversas ocasiões. *communication* por exemplo, é responsável por enviar os dados de volta para o ambiente, que na circunstância de uma competição online pode ser o FIRASim ou o TRAVESim. Porém a troca desse módulo pode permitir o envio para robôs físicos do mundo real, dando liberdade a equipe facilmente adaptar o código para testar diferentes protocolos de comunicação em etapas futuras do projeto.

Da mesma forma, o *vision* é um modulo que recebe dados do ambiente, e o que, a cada recebimento de dados invoca o

¹ alexsandro2134@gmail.com

² ana.monici@aluno.ufabc.edu.br

³ ryqck@hotmail.com

⁴ makitayukiol@gmail.com

⁵ giovanni.dias@aluno.ufabc.edu.br

⁶ mendes.gabriel95@gmail.com

⁷ g.ctorres09@gmail.com

⁸ jpkleal@gmail.com

⁹ josiascalebe@gmail.com

¹⁰ lulimg2004@gmail.com

¹¹ a.pacheco@ieee.org

¹² marcus.id.apple@gmail.com

¹³ cardoso.matheus@aluno.ufabc.edu.br

¹⁴ thales.machado.fernandes@gmail.com

¹⁵ victorhlo12@hotmail.com

mainloop.match por sua vez encapsula dentro dele todos os objetos relevantes para a tomada de decisão.

Esse ano, a modalidade ira trazer uma competição cinco contra cinco robôs. Nessa modalidade a equipe acredita que novas ferramentas dentro do nosso software afim de customizar a estratégia em diferentes situações ira nos ajudar a conduzir uma quantidade maior de robôs num espaço maior.

Um bom controle e bons algoritmos de tomada de decisão são cruciais, porem alem que isso, conseguir identificar situações especificas e mudar a estratégia sera tão importante quanto. Com isso em mente, pensamos em trazer um Coach que muda o conjunto de estratégias distribuidas para os robôs baseado em condições, e essas transições serem administradas por uma maquina de estados. Chamamos esse sistema de Playbook.

A. Módulos

Atualmente o NeonFC conta com alguns módulos que durante o desenvolvimento foram considerados relevantes para receber esse status. Um módulo nada mais é do que um componente do nosso software que pode ser trocado de forma a se adaptar para outras modalidades.

Nome	Função	Alternativas
vision	Se comunica com o ambiente e recebe os dados de posicionamento do campo	FIRAVision - para o FIRA TraveVision - para o TraveSim SSLVision - para o SSL
comm	Se comunica com o ambiente para enviar os comandos para os robôs	FiraComm - para o FiraSim TraveComm - para o TraveSim PortComm - para envio para porta serial USB UDPComm - para envio para algum endereço UDP
referee	Se comunica com arbitro automatico	VSSSRefereeComm - para o VSSSReferee
field	Encapsula os dados relevantes a geometria do campo	3v3, 5v5
coach	Gerencia a tomada de decisão dos robos	LARC2020 - Coach usado na LARC 2020 IRON2021 - Coach usado na Iron Cup 2021
strategy	Encapsula a tomada de decisão que deve ser feita para um robo	Attacker, Goalkeeper, Midfielder...
algorithm	Encapsula o uso de um algoritmo de tomada de decisão	Astar, PotentialField, DWA...
controller	Recebe os dados de uma strategy (velocidades nos eixo x, y) e transforma em potenciais para as rodas	PID

Fig. 1: Principais componentes modulares usados no NeonFC e suas alternativas

O conceito de um software modularizado também nos ajudou a validar o funcionamento de diferentes algoritmos usados tanto para tomada de decisão do robô, quanto para o sistema de controle, possibilitando uma testagem rápida de diferentes algoritmos. Com ajuda de conceitos de programação orientada a objetos podemos testar diferentes sistemas de controles atrelados a diferentes sistemas de tomada de decisão, que serão descritos na subseção *Fluxo das Entidades*.

Por fim, a facilidade de desacoplar os módulos, também permitiu que membros da equipe com menos familiaridade a esses algoritmos pudessem testar configurações novas e desmembrar as diferentes etapas do fluxo de tomada de decisão do NeonFC. Isso se demonstrou uma decisão acertada dado que diversos algoritmos validados dentro do contexto do nosso software pudessem ser facilmente portados para novos projetos e robôs.

B. Fluxo das Entidades

O módulo Game é o primeiro objeto a ser inicializado na execução do NeonFC e tem como responsabilidade tratar e

armazenar informações que precedem a execução do jogo, mais precisamente quais módulos de visão e comunicação instanciar, quais são seus arquivos de configuração, quantos robôs inicialmente estarão em campo e qual a cor inicial do time (amarelo ou azul). Por fim, essas informações são passadas para os modulo *Vision*, *Comm* e *Match* para que sejam devidamente inicializados.

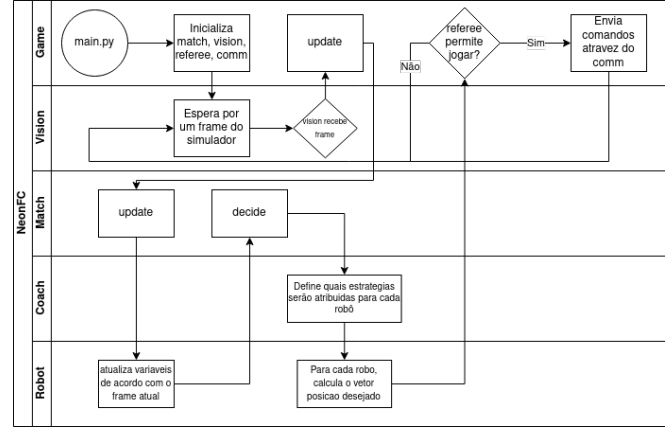


Fig. 2: Iteração principal do NeonFC

O módulo *Comm* apenas encapsula as funções necessárias para o envio do dado para o robô. Sendo que a estrutura da mensagem a ser enviada fica a cargo da entidade robot.

Para que as entidades tomem decisões utilizamos uma entidade chamada coach. O Coach é a entidade virtual que tem como objetivo centralizar a tomada de decisão do comportamento das demais entidades. O Coach faz isso tendo uma lista de estratégias e uma série de regras. Para cada estratégia existe uma regra que deve ser atendida e uma prioridade, a entidade mais apta a receber aquela estratégia passa a desempenha-la.

III. TOMADA DE DECISÃO

A. Sistema Playbook

Com o intuito de dar maior dinamicidade no uso de estratégias, surgiu o Playbook. O Playbook é uma estrutura alternativa para se criar coaches, ele funciona como uma máquina de estados onde cada transição (chamada de trigger) pode ser ativada por alguma flag visível no NeonFC e cada estado (chamado de Play) é um conjunto de estratégias a serem atribuídas aos robôs, fazendo com que possamos, facilmente, trocar de estratégias ao decorrer da partida.

O Playbook também facilita a criação de jogadas de bola parada, que não eram muito bem exploradas. Com as informações que o VSSSReferee envia (como qual o tipo de falta que ocorreu e para qual time a falta foi marcada) é possível utilizá-las como gatilho para uma transição de estratégia. Por exemplo, em uma situação de jogo acontece um lance de bola livre, nesse caso, o VSSSReferee envia a informação “FREE BALL”. Com isso, é possível utilizá-la como um gatilho para uma transição de estratégias, ou seja, se a estratégia atual for uma para quando o jogo está em andamento e acontece uma bola livre, com o gatilho enviado

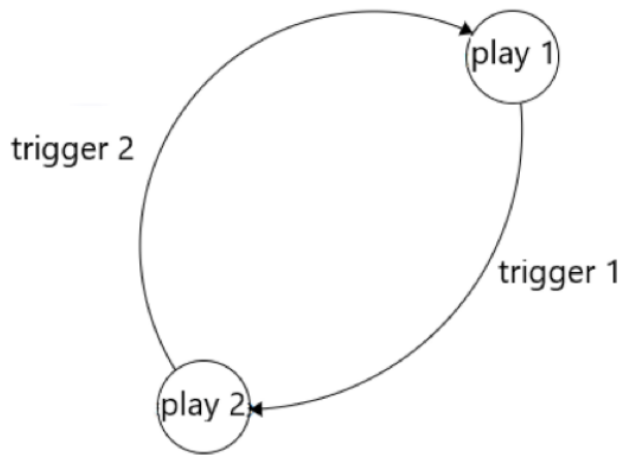


Fig. 3: Um sistema simples de playbook com duas jogadas, sendo alternadas por duas transições.

pelo Referee, é possível mudar esta estratégia para uma de bola parada e, logo em seguida, voltar para a estratégia de jogo em andamento, o que possibilita o melhor aproveitamento das informações que recebemos ao decorrer da partida e cria diversas possibilidades de criação de jogadas.

B. A* (A-estrela)

O A* é um algoritmo de path-planning que busca o caminho mais curto entre dois nós de um grafo. O NeonFC usa esse algoritmo com um grafo de nós representando pontos no campo no qual o robô poderá percorrer. Assim, o A* recebe um grafo, o nó inicial (posição atual do robô) e o nó final (posição do objetivo), então retorna uma lista de posições no campo que o robô deverá percorrer para alcançar o objetivo.

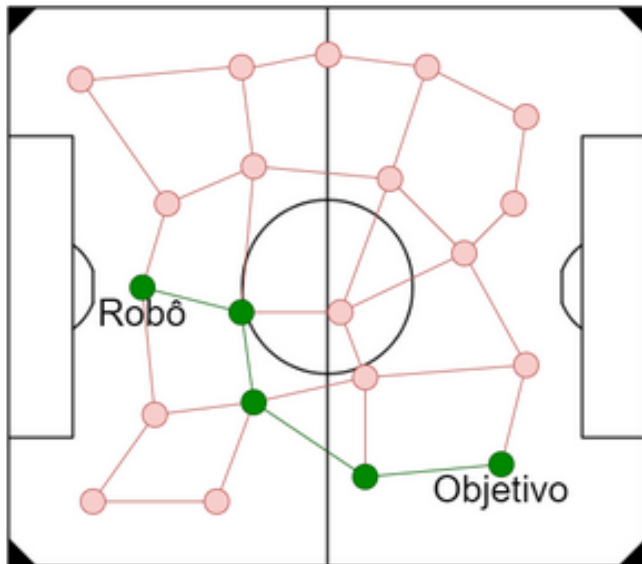


Fig. 4: Os pontos no campo são os nós do grafo. Os nós em verde representam o menor caminho entre o robô e o objetivo.

Uma classe abstrata chamada FieldGraph é usada para criação dos grafos que serão usados pelo A*. Dessa forma, várias estratégias diferentes podem ser criadas a depender dos grafos gerados. Atualmente, o NeonFC utiliza diagrama de Voronoi para gerar o grafo usado pelo A*. O NeonFC usa esse algoritmo com um grafo de nós representando pontos no campo no qual o robô poderá percorrer, então, o A* retorna a lista de nós em ordem para que o robô chegue ao nó objetivo percorrendo a menor distância possível.

C. Campos Potenciais

Campo potencial (potencial field) é um algoritmo de planejamento de trajetória muito usado em robôs móveis. Consiste na criação de vários campos vetoriais, onde um vetor velocidade é atribuído a cada ponto do campo, sendo que diferentes somatórias de diferentes campos geram comportamentos diferentes para o robô usar em determinadas situações.

O NeonFC usa 3 tipos de campos vetoriais: campo de ponto, campo de linha e campo tangencial. No geral, todos os campos têm um ponto central e uma área/raio de atuação, sendo possível determinar a velocidade máxima do robô afetado pelo campo, e se o robô sofrerá aceleração/desaceleração ao se aproximar/afastar do ponto central do campo. O campo de ponto gerará um campo vetorial no qual os vetores apontam em direção a um ponto escolhido, podendo ter sentido para o ponto (atração) ou para longe do ponto (repulsão).

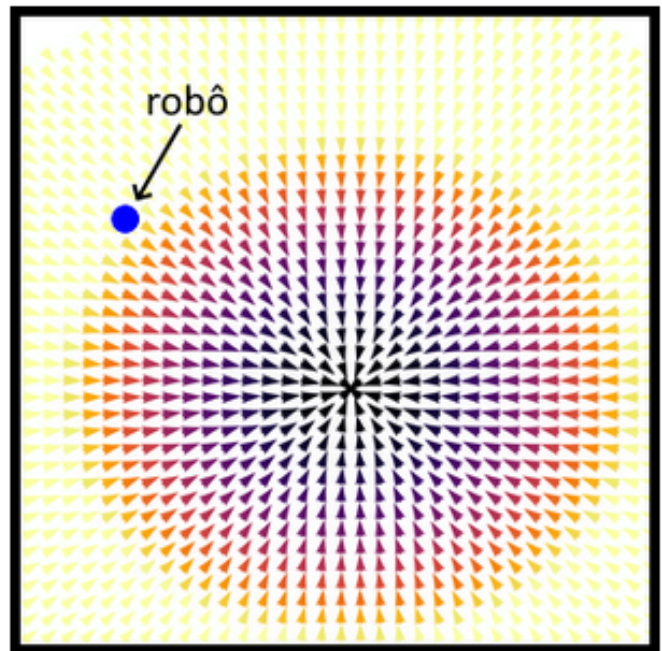


Fig. 5: Campo potencial de ponto

IV. TRABALHOS FUTUROS

Esse ano, nosso foco para a modalidade simulada foi usá-la de ferramenta de aprendizado para novos membros até que eles se habituem com conceitos básicos de robótica para

assim comecem a desenvolver estratégias para o ambiente real. Para participações futuras, pretendemos trazer inovações que distinguiam as atividades do ambiente robótico físico (chamado de Real Life internamente) do ambiente simulado.

Dentre essas atividades, estarão uma tunagem mais fina nos parâmetros de controle e de tomada de decisão, específicos para o ambiente do FIRASim. Um estudo inicial está sendo feito para usar o Rsoccer, desenvolvido pela equipe Robocin[2].

Por fim, como a equipe ainda não dispõe de um campo físico para a competição cinco contra cinco. Muito entendimento de como esse número a mais de robôs impacta nas estratégias ainda vai ser estudado dentro desse ambiente simulado, implementando idéias executadas por outras equipes observadas nessa competição.

REFERÊNCIAS

- [1] NeonFC code available in: <https://github.com/Project-Neon/NeonFC>
- [2] Felipe B. Martins, Mateus G. Machado, Hansenclever F. Bassani, Pedro H. M. Braga, Edna S. Barros. (2021). rSoccer: A Framework for Studying Reinforcement Learning in Small and Very Small Size Robot Soccer.