Pequi Mecânico - IEEE VSS Soccer Team - LARC/CBR 2022

Bryan L. M. Oliveira, Bruno B. S. Martins, Danielle S. L. Alfredo, Dimitri S. Veloso,
Felipe F. G. Neiva, João P. A. F. Matos, Luana G. B. Martins,
Pedro C. da C. Arantes, Pedro P. C. Vieira, Thiago H. Oliveira,
Werikcyano L. Guimarães

***Resumo*— Este trabalho descreve o projeto Very Small Size Soccer (VSSS) desenvolvido pela equipe do Núcleo de Robótica Pequi Mecânico da Universidade Federal de Goiás (UFG) para competir na categoria IEEE Very Small Size Soccer da Competição Latino Americana e Brasileira de Robótica 2022 (LARC/CBR 2022). Este trabalho tratará de explicar todas as partes do hardware, controle, visão computacional e estratégia.**

I. INTRODUÇÃO

A categoria IEEE Very Small Size Soccer trata-se de uma competição de robótica entre duas equipes de 3 robôs de dimensões até 7,5x7,5x7,5cm. Os robôs são controlados por um computador, chamado de técnico. Todas as jogadas dos jogadores são controladas autonomicamente, ou seja, sem a intervenção humana. Para saber as posições dos robôs, e assim enviar informações de posicionamento através de um rádio de comunicação, uma câmera é acoplada acima do campo. Essa câmera tem a finalidade de fornecer as informações de posicionamento tanto dos robôs aliados quanto dos robôs adversários bem como a posição da bola. O reconhecimento por visão computacional é feito através de tags coloridas colocadas acima de cada robô. Cada robô deve possui pelo menos a cor do time, que pode ser azul ou amarela, as chamadas tags principais. A maioria dos times utilizam de duas a três cores de modo a identificar separadamente cada robô. Essa segunda tag é a chamada tag de jogador ou simplesmente tag secundária. Desse modo, por meio dessas tags secundárias, o sistema computacional consegue reconhecer os jogador e calcular estratégias específicas para cada robô. A bola possui a cor laranja e nenhuma tag segundaria pode ser dessa cor, sendo exclusiva para a bola.

Um sistema de estratégia recebe a posição e dos jogadores e da boa e calcula as respectivas velocidades e posições-alvo, enviando-as através de um rádio transmissor.

A categoria IEEE Very Small Size Soccer é disputada no Brasil desde o ano de 2003. De lá pra cá, a categoria vem crescendo e ganhando mais adeptos todos os anos [1].

Um exemplo de projeto desenvolvido para a categoria IEEE Very Small Size Soccer é mostrado na Figura 1. Esses robôs foram desenvolvidos pelo Grupo de Robótica Pequi Mecânico (PMEC) da Universidade Federal de Goiás (UFG) para a CBR 2016 [2]. Eram utilizados microcontroladores da Atmel Atmega32U4 para o gerenciamento de recebimento de dados via rádio e controle de velocidade [2]. Durante o decorrer do projeto, viu-se a dificuldade de processamento que tal microcontrolador possui em se tratando de velocidade de processamento. Além disso a capacidade de armazenamento do algoritmo quase excede a capacidade disponível do microprocessador.

Todos os autores são do Núcleo de Robótica Pequi Mecânico, Universidade Federal de Goiás. pequimecanicoufg@gmail.com

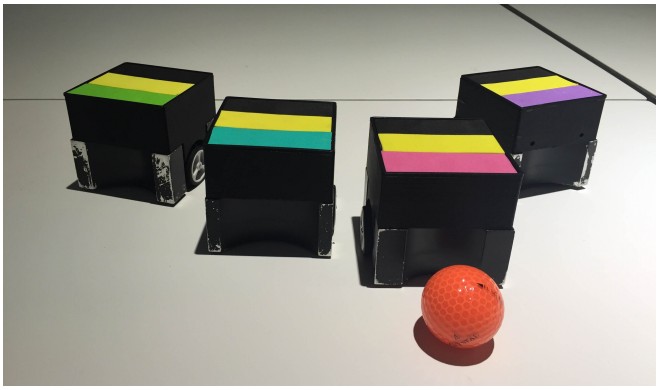


Fig. 1: Vista superior da estrutura dos robôs.

Nesse sentido, todo projeto foi reformulado, adicionando novos sensores, motores, rodas, estrutura e principalmente um microcontrolador melhor. Entre as melhorias no software, destaca-se o uso do Filtro de Kalman Estendido para fusão de sensores e o Univector Field (UVF) para navegação. Assim, o objetivo desse artigo é mostrar o trabalho de aprimoramento das partes de hardware e software desenvolvidos pelo Núcleo de Robótica Pequi Mecânico.

O artigo está assim dividido: Na seção 2 será discutido os componentes utilizados no projeto do robô físico bem como as ferramentas utilizadas para o projeto da estrutura. Na seção 3 será discutido como o sistema de visão computacional funciona bem como as melhorias feitas entre os anos de 2017 e 2022. Na seção 4 é apresentado a fusão de sensores. Adiante, na seção 5 é apresentado o sistema de controle, seguido pela estratégia dos jogadores na seção 6. Por fim, é apresentado as conclusões pertinentes ao trabalho e sobre futuras melhorias para os próximos anos.

II. DESCRIÇÃO DA CONSTRUÇÃO DOS ROBÔS

Nesta seção será discutido sobre as principais partes que compões o hardware dos robôs do Núcleo de Robótica Pequi Mecânico. Uma breve descrição dos componentes será dada bem como o porque dessas escolhas.

A. *Estrutura*

A estrutura do robô foi projetada para suportar os impactos que são inevitáveis quando se procura um mesmo objeto em comum, a bola. A estrutura do robô foi confeccionada em impressora 3D com material ABS premium principalmente devido a facilidade de se produzir qualquer formato de peça que se queira. A estrutura também é responsável por manter todas as partes do robô fixas, tais como placa, motores e rodas, baterias e case das tags.

O projeto 3D do robô foi feito previamente e é mostrado na Figura 2. A estrutura foi projetada com o objetivo de ter as menores dimensões possíveis e a melhor interação com o usuário. Para facilitar tal interação, os encaixes da tampa de bateria e da case de tags são feitos através de imãs e optamos por utilizar parafusos em lugares essênciais e indispensáveis para facilitar e reduzir o tempo de montagem do robô.

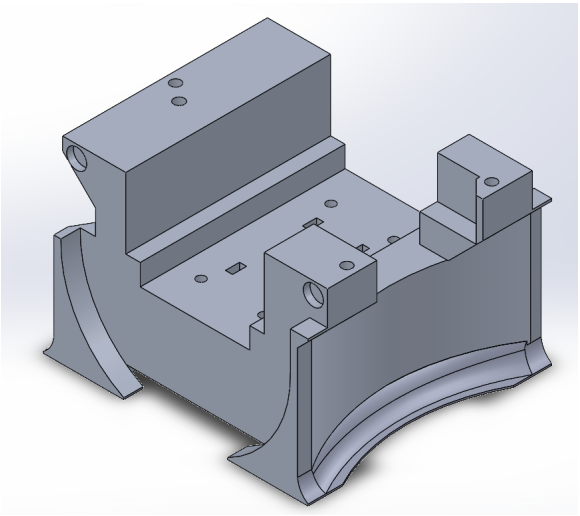


Fig. 2: Vista isométrica da estrutura dos robôs.

Outra característica básica é a altura da base da estrutura até o piso do campo. Pensando nesse quesito foi projetado uma nova estrutura de modo a manter a estrutura o mais rente possível do chão para uma melhor estabilidade. Conseguiu-se uma altura de 2mm do ponto mais baixo da estrutura até o chão. Além disso, outra característica importante é a geometria da frente do robô pois a condução da bola está diretamente ligado a ela. Optamos pela forma hiperbólica para que em posse da bola a mesma fique no meio do robô.

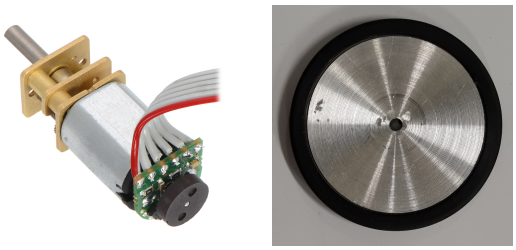
É importante, também, ressaltar que a estrutura foi projetada com a finalidade de manter o centro de gravidade do robô o mais próximo do solo possível resultando em maior estabilidade e facilidade para que este possa jogar sem uma frente definida, podendo inverter seu sentido rapidamente sem que o momento resultante de tal ação "capote"o robô.

B. *Rodas, Motores e Encoders*

As rodas e motores escolhidos para serem usados no projeto foram de tal forma que o robô pudesse ter um bom torque associado com uma boa velocidade. Motores de corrente contínua (CC) são bons, mas não possuem um torque adequadamente bom para o tamanho do motor que se deseja. Assim escolheu-se motores com caixa de redução (os chamados gearmotors). Esses motores possuem uma velocidade relativamente alta, cerca de 400rpm. O papel das engrenagens acopladas no eixo do motor é diminuir a velocidade de rotação do eixo acoplado às rodas e compensação ganhar mais torque. A redução adequada para o peso e para a estrutura do robô desenvolvido é de 75.81:1, o que significa que a cada 75.81 voltas do eixo do motor a roda gira 1 volta completa.

O consumo médio dos motores chega em torno de 1A de pico sendo alimentado por uma tensão de aproximadamente 6V. Como se trata de um motor bem pequeno o consumo médio é relativamente pequeno.

As rodas foram usinadas em alumínio e possuem um diâmetro de 60mm. Para uma melhor aderência usa se pneus de borracha evitando assim o deslizamento em ocasiões onde se exige um grande torque. A Figura 3 mostra o kit roda-motor utilizado no projeto.

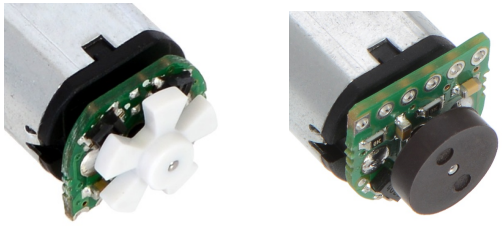


(a) Gearmotor 75:1 com encoder magnético.

(b) Roda de 60mm de diâmetro.

Fig. 3: Conjunto roda-motor utilizado no robô

Uma outra melhoria feita foi a troca do tipo de encoder das rodas. Passou-se de um encoder óptico para um encoder magnético. A Figura 4 mostra o detalhe dos dois tipos de encoders. O encoder óptico possui uma resolução de 20 pulsos por volta e o encoder magnético de efeito hall possui 12 pulsos por volta de resolução. Apesar de possui uma menor resolução, o encoder magnético não deixa a desejar.



(a) Encoder óptico.

(b) Encoder magnético.

Fig. 4: Tipos de encoders.

Apesar de não ter melhorado a resolução do sinal, melhorou-se a forma de onda do sinal que antes era parecida com uma onda senoidal e agora é uma onda quadrada perfeita. A Figura 5 mostra a forma de onda de ambos encoders.

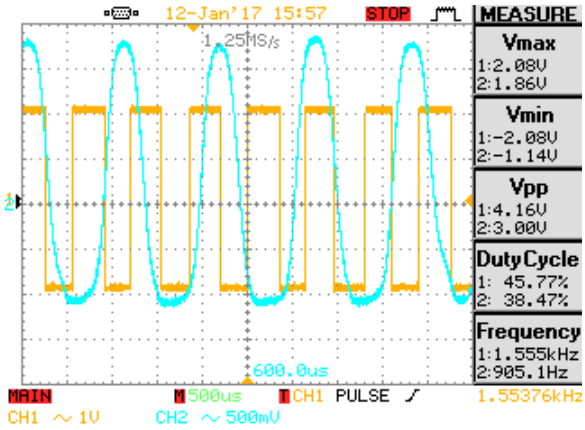


Fig. 5: Ondas fornecidas pelos encoders magnéticos e ópticos.

C. Sensor Inercial

Foi utilizado o Sensor Inercial MinIMU-9 v5, que inclui um acelerômetro, giroscópio e magnetômetro de 3 eixos. O microcontrolador se comunica com o sensor através do protocolo I2C. Os dados obtidos por dele foram utilizados em um Filtro de Kalman Extendido para estimar a posição, orientação e velocidade do robô.

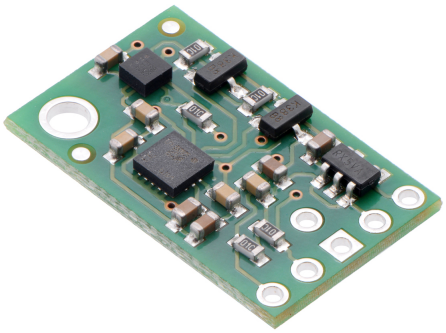


Fig. 6: MinIMU-9 v5

D. Comunicação

A comunicação entre robô e computador é feita utilizando módulos Xbee Série 2. Esses módulos possuem vários canais de comunicação que podem ser criptografados. A comunicação entre módulo e robô é feita por comunicação serial.

O envio de informação é bem simples e é atualizada várias vezes por segundo. A cada informação enviada um pacote contendo as informações de cada robô é criado e enviada aos respectivos robôs de modo que somente o robô correto recebe a informação. Como o pacote de dados é criptografado, somente os módulos do respectivo time conseguem decifrar os pacotes e vice-versa. Esse tipo de comunicação evita interferência no recebimento da mensagem o que garante que a mensagem recebida será a mensagem enviada.

O emissor de pacotes ligado ao computador também é um módulo Xbee que funciona como módulo mestre, e os outros três robôs, cada um com uma informação diferente, funcionam como escravos. Assim o fluxo de dados é sempre entre comutador e robôs.

E. Microcontrolador

Foi utilizado a plataforma Mbed que possui o microcontrolador LPC1768. Foi escolhido essa plataforma devido a sua espessura reduzida, seu número de portas ser suficientes para controlar todos os sensores e atuadores. Pode-se ter uma visão geral da plataforma MBED na figura 7.

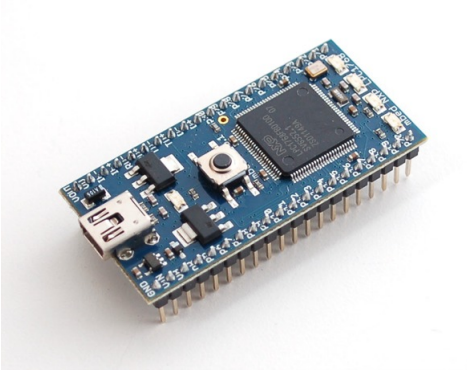


Fig. 7: Plataforma Mbed LPC1768.

A plataforma Mbed possui um núcleo Cortex-M3 ARM de alta performance, roda a 96MHz e possui 40 pinos do tipo 2.54mm. Suas dimensões são 54x26mm. Possui muitos tipos de comunição, sendo os mais comuns I2C, SPI e UART.

F. Fonte de Energia

A fonte de energia mais viável atualmente são as baterias de Polímero de Lítio (LIPO). Como o robô possui uma estrutura reduzida, a bateria não pode ser muito grande e deve caber no interior da estrutura do robô. Uma bateria ideal para o tamanho do robô é a de duas células do tipo LIPO. Cada uma das células da bateria possui cerca de 3.7 Volts, totalizando 7.4 Volts. Essa fonte de energia é utilizada tanto pelos motores quanto pela parte controladora do robô.

Estudos estão sendo feitos a respeito de utilizar baterias mais leves com um ciclo de recarga menor. Baterias do tipo Li-Íon são conhecidas por possuírem tamanhos menores, serem mais leves e armazenarem uma quantidade de energia maior comparada as baterias de LIPO.

G. Robô

O protótipo impresso em 3D é mostrado na Figura 8. O robô já está com bateria, rodas, motores, encoders, sensores e placa eletrônica montada.

III. SISTEMA DE VISÃO

A visão computacional é responsável por coletar os dados sobre o jogo em cada instante de tempo e passá-las de forma mais simplificada possível para o algoritmo responsável pela estratégia. Este processo é dividido nas seguintes etapas: primeiro, a câmera, colocada perpendicularmente a aproximadamente 2 metros de distância do campo, faz uma série de

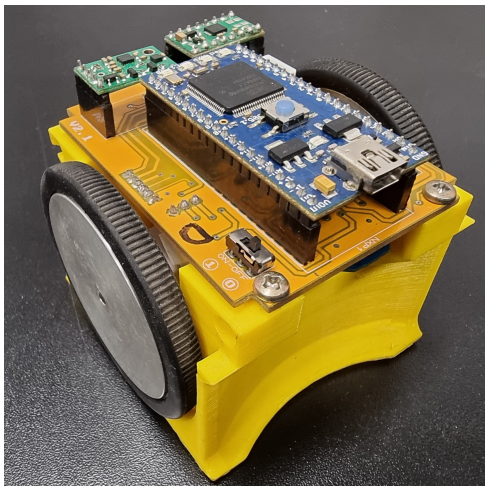
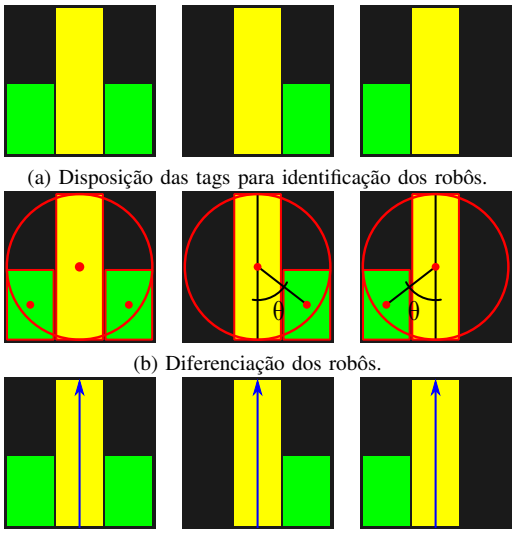


Fig. 8: Robô montado com motores, encoders, rodas, bateria, placa e sensores



(a) Disposição das tags para identificação dos robôs.

(b) Diferenciação dos robôs.

(c) Orientação dos robôs.

Fig. 9: Novo sistema de tags dos robôs.

captura de frames e os envia para o computador. Um frame é uma imagem individual enviada pela câmera. A partir daí, é utilizado um algoritmo baseado na biblioteca OpenCV [3] para o processamento dessas imagens. A imagem é tratada por de filtros que retiram partes da imagem que não são necessárias, deixando apenas o que é realmente importante: as tags de cores, que identificam os robôs, e a bola. Após o processamento da imagem para a detecção das tags de cores e da bola, um algoritmo de rastreamento é utilizado para encontrar, através de thresholds de cor, a posição absoluta de cada robô e da bola no campo.

O frame original é convertido do sistema de cores RGB para CIE L*A*B*. Esse espaço de cor foi escolhido por ser o que mais se aproxima com a visão humana. Como a determinação dos thresholds de cada cor é feito manualmente por um operador humano, a calibração das cores se torna um processo mais natural, já que as cores vistas pelo operador se aproximam das cores interpretadas pelo computador. Nos anos anteriores, era utilizado o espaço de cores HSV. Porém, com CIE L*A*B*, a etapa de calibração se tornou mais fácil e prática. Além disso, há bem menos dificuldades para identificar a cor laranja que representa a bola. Após feito o threshold, a imagem passa pelos filtros de suavização de imagem, erosão e dilatação, a fim de eliminar ruídos e corrigir formas e tamanhos.

Assim, é possível identificar a posição dos robôs e da bola, que são guardadas em memória e enviadas ao algoritmo de estratégia, que determina a ação de cada robô naquele instante de tempo. Como a nossa câmera permite uma taxa máxima de 30 frames por segundo, o algoritmo de visão é processado a cada 33ms.

O sistema de tags de cores, utilizado para a detecção dos robôs no campo, foi reformulado para a competição deste ano. A posição do robô é identificada pelo centro da tag principal, pois ambos são concêntricos. Cada robô possui até duas tags secundárias, de mesma cor, que estão dispostas ao lado da tag principal. Como pode ser visto na figura 9b, a diferenciação dos robôs é feita pela contagem de tags secundárias que estão dentro da área do robô. Caso o robô tenha apenas uma tag, o que os diferenciará será o ângulo em que a tag secundária se encontra em relação a tag principal. Caso o robô tenha duas tags secundárias, não é preciso analisar nenhum ângulo adicional, pois há apenas um robô com duas tags secundárias. Já a orientação é determinada pela direção do segmento de reta que divide a tag principal ao meio no sentido longitudinal. O sentido dessa reta é sempre do lado onde as tags secundárias estão para o lado oposto do robô, onde não há tags secundárias.

Com o sistema de tags atual, só há a necessidade de ter apenas uma cor secundária para identificação e orientação dos robôs. Menos cores secundárias implica em menos custo computacional e menos tempo para que o operador humano faça a calibração das cores. Além disso, há uma diminuição significativa da interpretação incorreta de cores pelo algoritmo de visão, já que com esse sistema, é preciso diferenciar apenas 4 cores: laranja (bola), verde (tag secundária), azul e amarelo (tag principal do nosso time e do time oponente). Quanto mais cores se tem para diferenciar, maior é a chance de que sejam confundidas, devido a proximidade que elas possam ter no espectro de cores.

IV. Fusão de Sensores

Para obter um bom desempenho no controle dos robôs, é necessária uma boa estimativa da posição, orientação e velocidades linear e angular deles. Para medir estas variáveis são utilizados um IMU, um encoder em cada roda e o sistema de visão computacional. Cada um desses sensores tem vantagens e desvantagens: O giroscópio é rápido e preciso, mas sofre de drift devido a integração da velocidade angular. O encoder permite calcular as velocidades linear e angular do robô, mas pode medir dados errados caso ele derrape. O sistema de visão computacional é capaz de medir posição e orientação diretamente, mas tem baixa taxa de

atualização e não tem alta precisão.

Pra reduzir o efeito das falhas e ruídos de cada sensor, foi implementado o Filtro de Kalman Extendido, que utiliza as medições de todos os sensores em conjunto para obter uma boa estimativa das variáveis. O ruído de cada sensor foi medido e modelado nas matrizes do EKF. O gráfico da figura 10 mostra a estimativa da orientação obtida pelo EKF e o valor medido pelo magnetômetro.

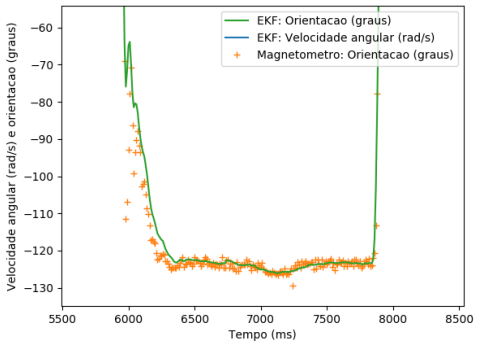


Fig. 10: Estimativa de orientação

No gráfico pode se perceber que a estimativa do EKF possui menos ruído que o valor medido pelo magnetômetro. Ao testar o controle do robô com e sem o EKF, foi notado um aumento significativo da estabilidade do sistema.

V. SISTEMA DE CONTROLE

O controle do robô é separado em duas etapas: A de baixo nível, que controla a velocidade das rodas do robô utilizando um PID, e a etapa de alto nível, que executa os comandos enviadas pelo computador. O robô é capaz de receber vários tipos de comandos, e aplica um controlador para cada um deles. Os tipos de controle são:

- Posição
- Orientação
- Direção (orientação e velocidade)
- Pose (posição e orientação)
- Velocidade de cada roda

Um exemplo de comando é "P1.2;0.5;0.8", que faz o robô ir para a posição 1.2 em x, 0.5 em y, com 0.8 m/s de velocidade. Como os robôs sabem a sua própria localização no campo devido ao EKF e sensores internos, eles são capazes de alcançar seus objetivos (posição, orientação, pose) de forma autônoma, sem que o computador tenha que controlar as velocidades das rodas continuamente. Isso reduz os delays de comunicação e permite que o controle execute a uma frequência maior, já que ela não fica restrito ao fps da câmera. O modelo utilizado no controle é o diferencial, que relaciona a velocidade linear e angular do robô com a velocidade de cada roda. O modelo é representado na figura 11.

O controle de alto nível utiliza este modelo para calcular a velocidade que deve ser aplicada em cada roda para fazer o robô alcançar o objetivo. Para alcançar uma pose, o robô segue um univector field, que define o caminho entre a pose atual até a pose desejada.

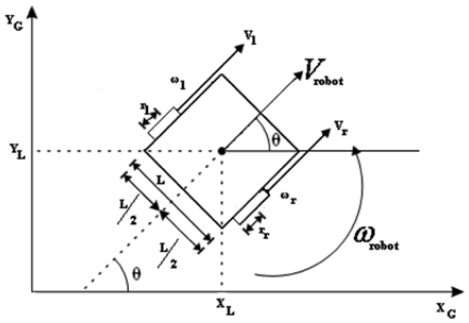


Fig. 11: Modelo diferencial aplicado ao robô.

VI. ESTRATÉGIA DE JOGO

A estratégia, assim como em um jogo real, deve avaliar a situação do jogo e tomar decisões que possam otimizar o comportamento dos jogadores. O papel do módulo de estratégia é analisar os dados coletados pela visão computacional e decidir quais serão as ações a serem tomadas por cada robô no próximo instante de tempo. As estratégias normalmente utilizadas no futebol de robôs seguem uma estrutura fixa, compostas normalmente de um sistema supervisor que recebe informações do algoritmo de visão computacional, decide qual ação os robôs devem fazer e, por fim, repassa essa ação para um sistema de controle [4] [5]. Esse sistema é confiável, mas apresenta uma performance não muito fluida, transformando as ações dos jogadores em jogadas pré-moldadas pelos criadores da estratégia. A proposta do time é criar uma estratégia simplificada, de modo a consumir menos recursos computacionais e ser de dinâmica com o ambiente de jogo.

Com o objetivo de transformar o módulo de estratégia em algo mais fluido, recorremos a técnicas de aprendizado de máquina, mais especificamente, aprendizado por reforço. O aprendizado por reforço busca mapear ações a estados maximizando uma função de recompensa [8]. No caso do futebol de robôs, utilizamos os dados coletados pelo módulo de visão, posições e orientações, para modelar uma representação da situação atual da partida. Esta representação seria o nosso *estado* a ser processado por um modelo de rede neural artificial. Em seguida, o modelo teria como saída, ações equivalentes à velocidades para cada motor de cada um dos robôs do time.

A. Modelagem do Problema

Para modelar o ambiente de futebol de robôs no formato de um problema de decisão de Markov, nós precisaríamos modelar um *estado* que não dependesse de estados passados, satisfazendo assim a propriedade de Markov [8]. Para tal, calculamos distâncias entre os diversos elementos no campo, como robôs, e a bola, bem como ângulos entre estes mesmos elementos. Uma representação estática de tal forma ainda carece de informações temporais, como velocidade e direção dos movimentos. Para inserir tais características ao estado a ser processado, nós empilhamos oito observações para enviar ao modelo. Desta forma, ele processaria leituras passadas

para inferir velocidade e direção. Esta abordagem não fere a propriedade de Markov pois não requer *memória* por parte do agente de tomada de decisão.

O modelo de rede neural artificial é composto de três camadas de neurônios com arquitetura *feed forward*, cujas saídas de uma camada são entradas da próxima. As camadas possuem 300, 200, e 100 neurônios respectivamente utilizando funções de ativação tangente hiperbólica. A camada de saída, por sua vez, possui dois valores para cada motor a ser controlado. Cada uma delas representa um alfa α e um beta β a serem utilizados na definição de uma função de densidade de probabilidade Beta [9]. Desta função é amostrada um único valor que representa a velocidade a ser aplicada no motor, após a transformação na Equação 1, onde x é a amostra da função Beta.

$$vel = (2x - 1) * 46.66 \tag{1}$$

Como função de recompensa, utilizamos diversos fatores descritivos da performance do time. Dentre estes fatores incluem-se a velocidade da bola em direção ao gol adversário, a distância do jogador mais próximo à bola, angulação entre a bola, o jogador em questão, e o gol adversário, e o ângulo entre a bola, o jogador, e o gol do time. Estes fatores representam a possibilidade de um time marcar um gol, bem como seu controle de bola e posicionamento no campo, ofensivo e defensivo. A recompensa em forma de combinação linear está descrita na Equação 2. O modelo foi treinado em um ambiente simulado dentro do simulador MuJoCo Physics Engine, 360 milhões de passos de simulação a uma frequência de 10 leituras por segundo.

$$r = vel * 0.7 + dist * 0.1 + angl\_off * 0.1 + angl\_def * 0.1 \tag{2}$$

B. Algoritmo de Reforço

O algoritmo de aprendizado por reforço utilizado foi o *Proximal Policy Optimization* [7]. Este algoritmo coleta experiências de diversos jogos ocorrendo de forma paralela e as utiliza para gerar um treinamento estável dos pesos da rede neural. O modelo final foi capaz de vencer a estratégia do Pequi Mecânico de 2019 com 84% de taxa de vitória. Nós também testamos o modelo contra um adversario com ações aleatórias como linha de base. Os resultados numéricos estão presentes na Tabela I.

Metrics	Random	Pequi
Win Rate	1.0	0.838
Draw Rate	0.0	0.09
Loss Rate	0.0	0.062
Score	8.179	2.884

TABLE I: Resultados de taxas de vitória, derrota, e empate, contra uma estratégia aleatória e a estratégia do Pequi Mecânico de 2019.

Além das altas taxas de vitória, o modelo demonstrou diversos comportamentos subjetivos durante o processo de aprendizado. Dentre eles incluem curvas complexas para pegar a bola em movimento, posicionamento espontâneo dos robôs, e um comportamento de martelar bloqueios de times adversários. Estes comportamentos são descritos e mostrados em um vídeo no canal do Pequi Mecânico: `https://www.youtube.com/watch?v=a6OA7QX8log&t`.

VII. CONCLUSÃO

Os resultados da aplicação dos métodos e técnicas apresentadas aqui estão sendo aprimorados mês após mês. O projeto do hardware vem sendo desenvolvido durante cinco anos e sofreu muitas mudanças de lá pra cá. Houve muitas melhorias no desempenho do controle com a alteração do microcontrolador e o uso do Filtro de Kalman Estendido.

O sistema de estratégia está sempre passando por melhorias e novos métodos estão sendo implementados. O algoritmo de reforço mostra resultados promissores.

Neste sentido, visou-se mostrar aqui as principais partes e divisões do projeto Very Small Size Soccer Team do Núcleo de Robótica Pequi Mecânico. Como o projeto ainda está em andamento, pequenas mudanças podem acontecer até a data da competição.

REFERÊNCIAS

[1] P. CBR ROBÓTICA (2018). Instituto Nacional de Estatística (2003). IEEE Very Small Size Soccer. Acesso em: 05 de julho de 2018. Disponível em: http://www.cbrobotica.org/?page_id=81&lang=pt.

[2] B. B. S. MARTINS, ET ALL, "Tdp equipe Pequi Mecânico VSSS - Futebol de robôs 2016 - Categoria Very Small" Competição Brasileira de Robótica, Recife, 2016. Acesso em: 08/07/2017. Disponível em: http://www.cbrobotica. org/mostravirtual/interna.php?id=14184.

[3] P. OPENCV. Acesso em: 05 de julho de 2018. Disponível em: http://opencv.org/.

[4] S. F. R. ALVES, ET ALL. Carrossel Caipira, o time de futebol de robôs da UNESP.

[5] A. ABRAHAM, ET ALL. Robot Soccer - Strategy Description and Game Analysis. Proceedings 24th European conference on Modelling and Simulation. Malaysia 2010.

[6] S. F. R. ALVES, J. M. ROSÁRIO, H. FERASOLI FILHO, L. K. A. RINCÓN, R. A. T. YAMASAKI. Conceptual Bases of Robot Navigation Modeling, Control and Applications. Unicamp e USP, 2010.

[7] SCHULMAN, J.; WOLSKI, F.; DHARIWAL, P.; RADFORD, A.; KLIMOV, O.Proximalpolicy optimization algorithms.eprint arXiv:1707.06347, 2017.

[8] SUTTON, R.; BARTO, A.Reinforcement Learning: An Introduction. AdaptiveComputation and Machine Learning series. MIT Press, 2nd edition, 2018.

[9] C HOU , P.-W.; M ATURANA , D.; S CHERER , S. Improving stochastic policy gradi- ents in continuous control with deep reinforcement learning using the beta dis- tribution. In: Precup, D.; Teh, Y. W., editors, Proceedings of the 34th International Conference on Machine Learning, volume 70 de Proceedings of Machine Learning Research, p. 834–843, International Convention Centre, Sydney, Australia, 06–11 Aug 2017. PMLR.