# Aprendizagem por Reforço Profundo com Redes Convolucionais Aplicada à Navegação Autônoma de Robôs Reais Utilizando Treinamento em Cenário Virtual

1st Carlos Daniel de Sousa Bezerra
*Instituto de Informática (INF) - UFG*
*Universidade Federal de Goiás (UFG)*
Goiânia, Brazil
carlosdbez@discente.ufg.br

2nd Flávio Henrique Teles Vieira
*Escola de Engenharia Elétrica Mecânica e de Computação (EMC)*
*Universidade Federal de Goiás (UFG)*
Goiânia, Brazil
flavio_vieira@ufg.br

***Abstract*—Aplicações industriais de sistemas robóticos móveis autônomos tem exigido o desenvolvimento de algoritmos de navegação autônoma utilizando inteligência artificial. Este artigo propõe um método baseado em algoritmo de aprendizagem por reforço profundo e rede convolucional (DQN-CNN: Deep Q-Network with Convolutional Network) para propiciar navegação autônoma a robôs dotados de câmera RBG, onde a fase de treinamento deste algoritmo é realizada no ambiente de simulação *Coppelia VREP* com a placa de desenvolvimento *Jetson Nano* da *NVIDIA*. A metodologia proposta abrange desde o treinamento em ambiente de simulação até a integração com sistema real. Para prototipação, utiliza-se uma pista de navegação com o robô controlado por algoritmo de aprendizado por reforço DQN. Os resultados apontam para efetividade da proposta, onde foi possível verificar o robô real desempenhando com sucesso a missão planejada. Espera-se que este trabalho possa contribuir para o avanço da área de robótica móvel e aprendizado por reforço, além de fornecer uma metodologia útil para pesquisadores e desenvolvedores que trabalham com estas tecnologias.**

***Index Terms*—DQN, Mobile Robots, Autonomous Navigation, Sim-to-Real.**

## I. Introdução

Com a chegada da Quarta Revolução Industrial (Indústria 4.0), a automação de tarefas e a digitalização dos serviços estão cada vez mais presentes nas fábricas e processos industriais. De acordo com [1], a Indústria 4.0 é conhecida pela integração de tecnologias avançadas, como inteligência artificial, internet das coisas (IoT), robótica avançada, análise de dados em tempo real, entre outras.

A robótica móvel é um dos principais pilares da Indústria 4.0, permitindo a automação flexível e adaptável em ambientes industriais complexos. Os robôs móveis autônomos (*Autonomous Mobile Robot* - AMR) são capazes de se moverem livremente pelo chão de fábrica, colaborar com os trabalhadores humanos e interagir com outros equipamentos, tornando assim, a produção mais eficiente e econômica [1]. Além disso, a robótica móvel permite a coleta de dados em tempo real, melhorando as tomadas de decisões, otimizando, portanto, o planejamento da produção. Com a robótica móvel, a Indústria 4.0 é capaz de alcançar uma maior eficiência operacional, reduzir custos e aumentar a competitividade no mercado global.

Nesse contexto, a robótica móvel tem ganhado destaque como uma solução versátil e eficiente para a automação de tarefas em ambientes industriais. Atualmente existem diversas de aplicações de robótica móvel, dentre elas: os robôs de seguidores de linha, os jogadores de futebol robótico e os AMR presentes em linhas industriais modernas para execução de diversas tarefas personalizadas.

Por essas razões, é crescente a aplicação de protótipos de AMR que possibilitam diversas configurações de *Hardware* e *Software*. Em [2], são citados como exemplos os robôs *M-Blocks* , um projeto do *Massachusetts Institute of Technology* (MIT) [3]. Os *M-Blocks* são capazes de se auto organizar e criar novas estruturas a partir da união com outros robôs do mesmo tipo. O projeto apresenta um novo sistema de *hardware* usando pulsos de momento angular e dobradiças magnéticas na tentativa de simplificar desafios práticos envolvendo módulos que se movem e se unem com outros módulos. Atualmente a inteligência artificial tem sido aliada da Indústria 4.0 e vem sendo pesquisada para a sua implementação em situações reais.

A Inteligência Artificial (IA) tem sido amplamente utilizada em sistemas robóticos móveis, aprimorando sua capacidade de interação com o ambiente e tornando-os mais autônomos. A IA é capaz de aprender com o ambiente e adaptar o comportamento do robô em tempo real, permitindo uma navegação mais eficiente e segura. Além disso, a IA é capaz de analisar grandes quantidades de dados em tempo real, fornecendo informações valiosas para a tomada de decisões em tempo real. Um exemplo disso é o trabalho de [4] que usa a IA em sistemas de navegação robótica móvel para ajudar o robô a evitar obstáculos e planejar rotas mais eficientes. Uma das técnicas de IA amplamente utilizada em sistemas robóticos é o aprendizado por reforço.

De acordo com [5], o Aprendizado por Reforço (*Rein-*

*forcement Learning* - RL) é uma técnica de aprendizado de máquina e portanto um subcampo da IA. Problemas de RL são compostos basicamente por: um agente que realiza ações em um ambiente, uma função de recompensa obtida e os estados do agente. O ambiente pode ser real ou simulado e produz informações que descrevem os estados do sistema. Utilizando estas informações de estados, o agente realiza uma determinada ação resultando em uma recompensa ou uma penalização. O objetivo da técnica é maximizar as recompensas obtidas, de forma que o agente aprenda por experiências e execute boas ações.

No entanto apesar da existência de diversos simuladores, encontrar uma plataforma robótica móvel adequada para o uso e prototipação em múltiplas aplicações não é uma tarefa fácil, como afirmado por [2], principalmente no âmbito do aprendizado por reforço. O aprendizado por reforço exige o aprendizado por experiências, em outras palavras, a tentativa e erro. Esta experiência por tentativas podem acabar danificando o sistema robótico devido a colisões e outras tentativas frustradas no processo de aprendizagem.

Desta forma levanta-se a principal **hipótese** desta pesquisa: Se é possível desenvolver uma metodologia que integre o treinamento desenvolvido em simulador robótico com sistema real, um robô real, de forma que o aprendizado especificamente por reforço profundo com redes convolucionais adquirido na plataforma de simulação seja transferido de forma segura para o ambiente real onde se utiliza um robô dotado de câmera RGB.

O **objetivo principal** é propor um método baseado em algoritmo de aprendizagem por reforço profundo, que denominamos de DQN-CNN (Deep Q-Network with Convolutional Network) para propiciar navegação autônoma a robôs dotados de câmera RBG onde aplica-se transferência de aprendizado, entre simulação e ambiente real. Para tal, propomos uma função de recompensa para a rede DQN-CNN assim como uma configuração para a rede CNN. Através do simulador, é desenvolvido o sistema de navegação autônoma utilizando aprendizado por reforço para controlar a navegação de um robô móvel. Posteriormente o aprendizado adquirido é transferido para um protótipo de robô real. Analisa-se, portanto, a eficiência desta transferência, bem como a performance em ambiente real.

O presente trabalho está organizado da seguinte forma: Na seção 2 são apresentados alguns trabalhos relacionados a pesquisa atual. Na seção 3 é feito uma descrição do problema de transferência de aprendizado. Os resultados da pesquisa são apresentados na seção 4 e por fim, a seção 5 dispõe as conclusões.

## II. Trabalhos Relacionados

Diversos pesquisadores têm se dedicado ao desenvolvimento de protótipos de robôs móveis para atender às demandas da Indústria 4.0. Um exemplo é o trabalho de [6], que apresenta um estudo sobre o desenvolvimento de um protótipo de robô móvel de baixo custo. Os pesquisadores visam explorar soluções acessíveis para a automatização de processos industriais, levando em consideração a crescente demanda por sistemas robóticos eficientes e econômicos. O estudo abrange diferentes aspectos do projeto, começando pelo design mecânico do robô, considerando sua estrutura, mobilidade e capacidade de carga.

O trabalho de [7] apresenta uma abordagem para o controle de robôs móveis que utiliza um controlador de estados finitos assíncrono. O controlador foi projetado para ser capaz de lidar com a incerteza e imprecisão presentes em ambientes dinâmicos e complexos, como os encontrados na indústria. O artigo apresenta detalhadamente a implementação do controlador em um protótipo de robô móvel, além de resultados experimentais que demonstram a eficácia do método proposto. Em resumo, o trabalho propõe uma abordagem inovadora para o controle de robôs móveis em ambientes industriais. Um desafio apresentado no trabalho é a integração do controlador e dos sensores em ambiente real, mesmo desafio citado no presente artigo.

A pesquisa de [8] apresenta uma revisão sobre o estado da arte das técnicas de transferência de aprendizado em RL para robótica. O trabalho discute os principais desafios na aplicação de técnicas de aprendizado por reforço em robôs reais, que incluem limitações na coleta de dados, dificuldades na generalização de políticas aprendidas em simulação para a implementação real. O artigo apresenta algumas das principais áreas de pesquisa em aberto na transferência de aprendizado em robótica, que incluem a exploração de novas técnicas de simulação. O presente artigo visa atuar no campo de Sim-to-Real Transfer citado na pesquisa de [8], integrando o aprendizado obtido por simulação em um robô real.

## III. Aprendizado por Reforço Profundo com CNN

Nesta seção, apresentamos o algoritmo proposto de navegação autônoma que se baseia em aprendizado por reforço profundo com CNNs. O sistema de Aprendizado por Reforço (*Reinforcement Learning - RL*) tem como objetivo fornecer conhecimento a um agente por meio de interações em um ambiente. As qualidades das ações desse agente são medidas por meio da função de recompensa. A função valor de estado-ação $Q(s, a)$ quantifica as recompensas obtidas por esse agente em relação aos seus estados e ações, tanto no curto quanto no longo prazo. Dessa forma, o agente aprende políticas de ação $\pi$ que maximizam a função valor. A função $Q(s, a)$ é uma variação do algoritmo de aprendizado temporal proposto por Bellman e é dada por:

$$Q^{\pi}(s, a) = r + \gamma \ \max Q^{\pi}(s', a') \tag{1}$$

onde $r$ é a recompensa imediata obtida e $\gamma$ é um fator de desconto para recompensas futuras, $s$, $s'$, $a$ e $a'$ são os estados e ações presentes e futuros, respectivamente.

O algoritmo DQN (*Deep Q-Networks*), um tipo de técnica de Aprendizado por Reforço (RL-Reinforcement Learning), incorpora os parâmetros de uma rede neural em seu algoritmo, com pesos sinápticos ($\theta$) e viés. Normalmente, a rede utilizada é profunda (com mais de três camadas) e tem como objetivo

estimar a função $Q(s, a)$ ótima, exibida na Equação 1. O DQN não requer uma representação tabular para armazenar e mapear os estados e ações do agente, como o *Q-Learning* tradicional. Em um ambiente com muitos estados e ações, por exemplo, imagens capturadas por uma câmera, representações tabulares não são viáveis. O mapeamento estado-ação do DQN é fornecido pela própria rede neural. Além disso, ao contrário do aprendizado supervisionado, no DQN os dados são dinâmicos e atualizados a cada etapa da integração do algoritmo. Este trabalho utilizará as redes DQN como método de aprendizagem por reforço.

Essa missão dada ao robô móvel é se locomover em uma pista circular de maneira autônoma, se mantendo dentro das vias indicadas. Essa missão pode ser uma indicação da capacidade do AMR (Robô Móvel Autônomo) de realizar tarefas compatíveis com as necessidades da Indústria 4.0, como coleta de dados do processo e navegação autônoma, entre outras. Caso o robô ultrapasse a via indicada ele sofre uma punição e o episódio de treinamento é interrompido. A função de recompensa, que guia o processo de aprendizado por reforço, dado por:

$$R = \begin{cases} dist_{k-1} - dist_k & \text{se ativo} \\ -1, & \text{se colidir} \\ +1, & \text{se atingir o alvo} \end{cases} \quad (2)$$

Onde $dist_k$ e $dist_{k-1}$ são respectivamente as distâncias do robô até o alvo no instante $k$. Esse cálculo permite recompensar ações que reduzem cada vez mais a distância em relação ao alvo, objetivo a ser alcançado. Caso o robô entre em colisão com uma parede, ele recebe uma penalidade de -1 ponto. Os alvos variam em relação à sequência que o robô deve percorrer, por exemplo, cada curva completada com sucesso, o robô recebe uma recompensa.

A rede neural convolucional (CNN) desempenha um papel fundamental na implementação do algoritmo DQN. A CNN é uma arquitetura de rede neural projetada especificamente para processar dados, como imagens por exemplo.

A estrutura da CNN consiste em camadas convolucionais, que aplicam filtros nas imagens de entrada para detectar padrões e características específicas, seguidas por camadas totalmente conectadas, que combinam as informações extraídas e geram as saídas correspondentes às ações possíveis. Essa combinação de camadas convolucionais e totalmente conectadas possibilita que a rede aprenda a representação visual das imagens e tome decisões com base nessa representação. A utilização da CNN no DQN permite que o robô processe eficientemente as informações visuais e tome decisões adequadas em seu ambiente. A Tabela I resume as camadas da CNN utilizada.

A Figura 1 apresenta o diagrama de blocos da rede neural DQN proposta. O robô móvel utilizado possui três ações de saída: i) virar direita, ii) virar a esquerda e iii) seguir em frente. Os movimentos angulares acontecem graças a dinâmica diferêncial adotada pelo robô, como aumentar a velocidade de uma determinada roda em relação a outra.

TABLE I
ESTRUTURA DA REDE CNN.

| CNN Layer | Filters | Kernel | Stride |
|---|---|---|---|
| 1st Layer | 16 | (5, 5) | 5 |
| 2nd Layer | 32 | (3, 3) | 2 |
| 3rd Layer | 32 | (2, 2) | 2 |

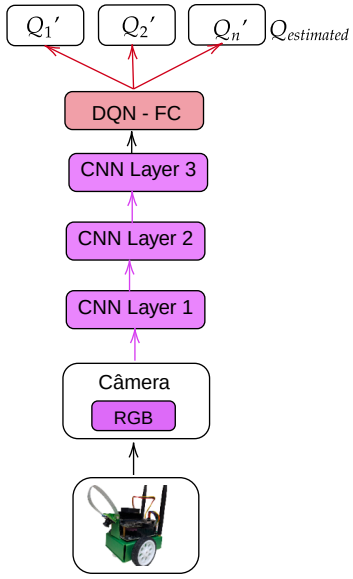


Fig. 1. Diagrama de Blocos da Rede Neural DQN Utilizada.

A rede neural possui várias camadas, cada uma desempenhando um papel específico no processo de aprendizado por reforço. Inicialmente, a rede recebe como entrada uma imagem com dimensões 64x64x3, representando a visualização do ambiente pelo robô. Essa imagem passa por três camadas convolucionais, onde cada camada aplica filtros para extrair características relevantes das imagens.

Cada camada convolucional é responsável por detectar padrões e características específicas nas imagens, como bordas, texturas e formas. À medida que a informação passa pelas camadas convolucionais, as características extraídas vão se tornando cada vez mais complexas e abstratas. Em seguida, a saída das camadas convolucionais é alimentada em duas camadas totalmente conectadas, que são responsáveis por combinar as características extraídas e gerar representações de alto nível da imagem.

Finalmente, a rede chega à camada de saída, que consiste em um conjunto de neurônios correspondentes às ações possíveis que o robô pode executar. Cada neurônio na camada de saída representa um valor Q estimado para uma determinada ação. Com base nesses valores Q, o algoritmo DQN é capaz de selecionar a ação com a maior recompensa esperada.

### *A. Rede DQN-CNN com Transferência de Aprendizado*

A transferência de aprendizado tem sido um tópico de grande interesse no campo de sistemas robóticos, especialmente quando combinado com técnicas de aprendizado por

reforço. Neste contexto, este artigo apresenta uma proposta inovadora de transferência de aprendizado para sistemas robóticos, com foco na abordagem Sim to Real. A ideia central é treinar um agente robótico em um ambiente simulado para completar um circuito de navegação de forma autônoma utilizando aprendizado por reforço. Posteriormente esse conhecimento é transferido para ambiente real. O diagrama de blocos da Figura 2 ilustra as etapas da proposta.

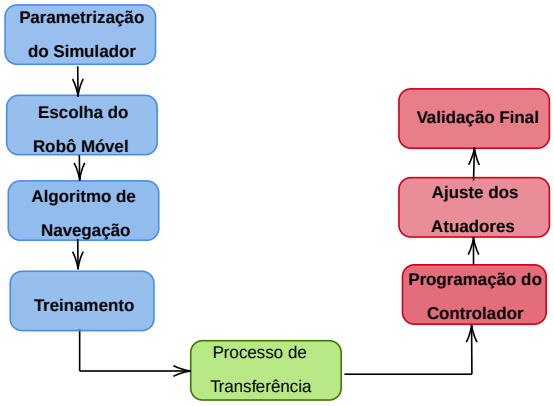


Fig. 2. Proposta de Transferência de Aprendizado.

Inicialmente é realizado a parametrização do simulador, isso é, a montagem do cenário de simulação etc. Logo em seguida é selecionado o robô mais adequado para desenvolver a tarefa em questão. Nesse caso foi selecionado o diferencial (duas rodas) pois é compatível com o robô real. Aplica-se então o algoritmo de navegação autônoma, utilizando aprendizado por reforço. Depois do completo aprendizado do agente (robô) inicia-se a transferência para o ambiente real.

O processo de transferência resulta em três etapas: i) Programação do Controlador, ii) Ajustes dos Atuadores e iii) Validação Final. Todas estas etapas são desenvolvidas no ambiente de simulação e precisam ser traduzidas em saídas e entradas do microprocessador do robô móvel. Durante o processo de transferência de aprendizado é comum observar que o modelo do robô móvel empregado no ambiente de simulação não é exatamente o mesmo utilizado em ambiente real, pois seus atuadores físicos se diferem. Por exemplo, o robô móvel utilizado no Simulador é o Pioneer 3-DX, já o utilizado em ambiente real é o Jetbot. Assim é necessário realizar alguns ajustes finos, como velocidade angular, tempo de amostragem, entre outros, para que o robô desempenhe um papel satisfatório.

As seções abaixo descrevem elementos primordiais para execução do diagrama de blocos da Figura 2. além do algoritmo baseado em aprendizado por reforço descrito nesta seção, como: o Ambiente de Simulação e o *Hardware* do Sistema Robótico.

## IV. Ambientes de Simulação e Teste

Este trabalho adota como ambiente de simulação o Coppelia V-REP (*Virtual Robot Experimentation Platform*). O Coppelia V-REP é um ambiente de simulação amplamente utilizado na pesquisa e desenvolvimento de robótica, permitindo a criação de cenários virtuais realistas para testes e experimentação. Com uma interface intuitiva e recursos avançados, o Coppelia V-REP oferece a capacidade de modelar ambientes complexos, controlar robôs virtuais e simular interações físicas precisas. A escolha desse sistema de simulação visa fornecer uma plataforma confiável e versátil para o desenvolvimento e validação de protótipos de robôs móveis, bem como para a transferência de aprendizado do ambiente virtual para o mundo real.

A Figura 3 ilustra o método utilizado nesse trabalho. O robô móvel é treinado a navegar de forma autônoma em uma pista montada no ambiente de simulação *Coppelia* e posteriormente seu conhecimento será enviado para ambiente real.

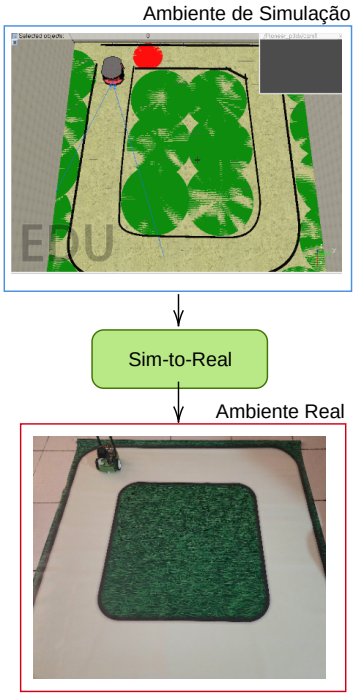


Fig. 3. Proposta de Transferência de Aprendizado.

Ao utilizar o Coppelia V-REP, é possível explorar diferentes cenários de simulação, ajustar parâmetros e observar o comportamento do agente robótico em um ambiente controlado e seguro.

A missão atribuída ao robô autônomo consiste em completar uma volta na pista desenvolvida no ambiente de simulação (vide Figura 3), sem sair dos limites da pista. Essa tarefa requer que o robô utilize seu algoritmo de controle por aprendizado por reforço DQN para tomar decisões de movimentação e navegação de forma autônoma. O objetivo é que o robô seja capaz de percorrer a pista seguindo uma trajetória correta, evitando colisões com obstáculos e mantendo-se dentro dos limites definidos. O simulação permite que o robô erre durante o processo de aprendizagem sem causar danos reais ao robô real.

### A. *Hardware - Sistema Robótico*

O sistema robótico empregado em ambiente real é o robô diferencial Jetbot, fabricado pela NVIDIA em parceria com a *WaveShare*. Este robô está ilustrado na Figura 4.

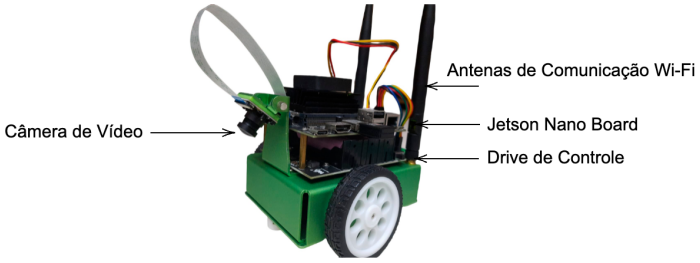


Fig. 4. Robô móvel com Placa de Desenvolvimento Jetson Nano.

O *Jetbot* é baseado na plataforma de controle e desenvolvimento NVIDIA *Jetson Nano*. Combinando o poder de processamento da *Jetson Nano* e uma estrutura de robô versátil, o *Jetbot* oferece uma solução completa para aplicações de robótica educacional e experimentação.

A placa *Jetson Nano* apresenta um processador ARM Cortex-A57 quad-core e uma GPU NVIDIA Maxwell. Essa combinação oferece um desempenho de processamento robusto e a capacidade de aceleração por GPU. O *Jetbot* também é equipado com uma câmera de vídeo de alta resolução, permitindo a visão computacional e a captura de imagens em tempo real. A câmera desempenha um papel fundamental em aplicações de reconhecimento de objetos, navegação autônoma e detecção de obstáculos. Vale ressaltar que a *Jetson Nano* permite programação em linguagem *Python*, o que facilita a implementação de algoritmos de inteligência artificial e integração com o simulador. Hoje existe um *framework* para movimentação e acionamento dos atuadores, disponível em https://github.com/NVIDIA-AI-IOT/jetbothttps://github.com/NVIDIA-AI-IOT/jetbot.

## V. Resultados e Discussões

A Figura 5 mostra a curva de aprendizado, ou seja os valores da função de recompensa versus os episódios, obtida a partir da simulação do sistema de navegação autônoma utilizando rede DQN (*Deep Q-Network*) em 500 episódios de treinamento.

A média móvel das recompensas obtidas (curva vermelha) mostra um aumento das recompensas ao longo do tempo, indicando que o agente de aprendizado por reforço está melhorando sua performance na tarefa de navegação proposta. À medida que o número de episódios aumenta, a curva de aprendizado mostra uma tendência ascendente, revelando o progresso contínuo do agente na busca por ações que maximizem suas recompensas. O robô móvel aprendeu a executar a missão proposta a ele.

A Tabela II contém os resultados dos testes de simulação realizados, nos quais foram analisadas duas métricas principais: a taxa de sucesso e a recompensa média. Cada teste consistiu em um conjunto de 5 episódios.

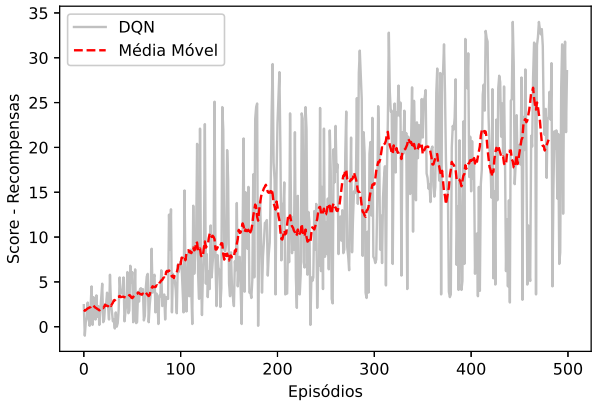


Fig. 5. Curva de Aprendizado do Algoritmo DQN - 500 Episódios.

A métrica de taxa de sucesso indica a proporção de episódios bem-sucedidos em relação ao total de episódios simulados. A métrica de recompensa média representa a média das recompensas obtidas ao longo dos episódios simulados.

TABLE II
Resultados dos Testes da Rede DQN-CNN

| **Rede DQN** | **Média de Recompensas Final** | **Taxa de Sucesso Final** |
|---|---|---|
| Teste 1 | 15,06 | 0,4 |
| Teste 2 | 17,00 | 0,4 |
| Teste 3 | 17,40 | 0,8 |
| Teste 4 | 19,26 | 0,8 |
| Teste 5 | 16,82 | 0,6 |
| **Média Final** | 17,51 | 0,6 |

Foi observado que o robô móvel aprendeu a completar a missão em ambiente de simulação com taxa de sucesso média de 60%. A média das recompensas obtidas foi de 17,51. Enfatiza-se que a máxima recompensa que o robô pode obter neste ambiente é de aproximadamente 20.

A Figura 6 ilustra a missão proposta para o Robô real, semelhante à desenvolvida no ambiente de simulação. É evidente que a câmera instalada no robô real capturou imagens e detalhes de cores não contidas no ambiente de simulação. Entretanto isso faz parte do processo de generalização Sim-to-Real.

Após a implementação da rede DQN no simulador e a validação dos resultados, o próximo passo foi a parametrização e programação do sistema na placa *Jetson Nano*. Foram realizados os ajustes necessários nos atuadores do robô *Jetbot* para que execução das ações aprendidas durante o treinamento em simulação. A placa *Jetson Nano*, conhecida por suas capacidades de processamento e baixo consumo de energia, proporcionou o ambiente ideal para a transferência do aprendizado por reforço para o mundo real. Vale ressaltar que a programação de suas GPIOs para executar a movimentação dos atuadores, como os motores do robô, foi por meio da linguagem *Python*.

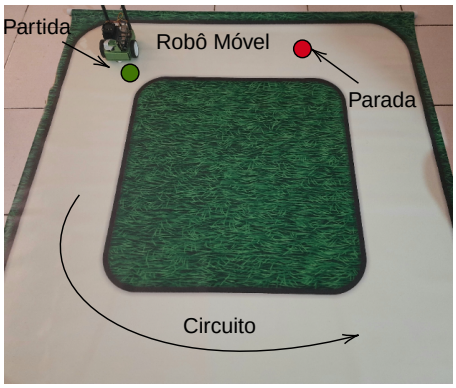


Fig. 6. Frames capturados de movimentação do Robô.

Após a parametrização da placa *Jetson Nano*, foi possível observar que o robô real executou com sucesso a missão proposta na pista, que antes era desempenhada em ambiente de simulação. A Figura 7 ilustra os *frames* capturados através de um vídeo da movimentação do robô móvel na pista referente ao ambiente real. É possível verificar a movimentação do robô *frame* a *frame* (do 1 ao 8). O vídeo de atuação do robô móvel está disponível em: https://bit.ly/3IKlRb5https://bit.ly/3IKlRb5.

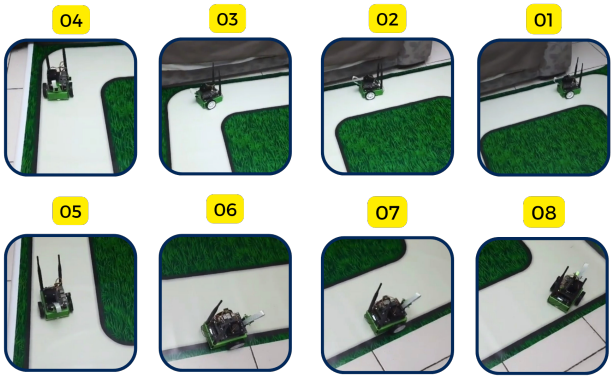


Fig. 7. Frames capturados de movimentação do Robô.

A implementação da missão proposta no robô real ocorreu de forma eficiente. Através da análise da figura, composta pelos frames de movimentação do robô, é possível verificar sua trajetória dentro da delimitação da pista. O robô foi capaz concluir a missão. Isso demonstra a eficácia do sistema de controle e da programação do robô, garantindo uma execução precisa da tarefa. A movimentação suave e precisa do robô dentro da pista reforça a capacidade do sistema em realizar tarefas compatíveis com as demandas da indústria 4.0.

## VI. Conclusões

Neste artigo, apresentamos uma metodologia para propiciar navegação autônoma a rôbos móveis através de um algoritmo baseado em aprendizado por reforço profundo utilizando rede CNN. Em conclusão, a proposta de transferir o aprendizado por reforço para um robô real se mostrou efetiva e bem-sucedida. Através da simulação e treinamento utilizando a rede DQN, o robô adquiriu habilidades de navegação autônoma e foi capaz de realizar a missão proposta de forma eficiente. A curva de aprendizado exibiu uma clara maximização das recompensas ao longo dos episódios de treinamento, indicando que o robô foi capaz de aprender e melhorar seu desempenho ao longo do tempo.

A implementação da missão no robô real demonstrou que o aprendizado obtido durante a simulação foi transferido com sucesso. O robô executou a tarefa com precisão, se mantendo dentro da pista delimitada e concluindo o circuito. A capacidade de realizar a tarefa proposta de forma eficiente em um ambiente real ressalta a viabilidade e eficácia do método de transferência de aprendizado por reforço.

Esses resultados evidenciam o potencial da abordagem de sim-to-real na robótica, onde o treinamento inicial ocorre em um ambiente simulado e é transferido para o mundo real. Essa abordagem permite uma economia significativa de tempo e recursos, além de possibilitar o desenvolvimento de sistemas robóticos adaptáveis e robustos. A transferência de aprendizado por reforço para um robô real abre caminho para a aplicação prática da robótica em ambientes industriais, contribuindo para o avanço da Indústria 4.0 e suas demandas de automação inteligente e flexível.

## References

[1] R. Andreja, "Industry 4.0 concept: Background and overview," *International Journal of Interactive Mobile Technologies (iJIM)*, vol. 11, 2017.

[2] M. Tkáčik, A. Březina, and S. Jadlovská, "Design of a prototype for a modular mobile robotic platform," *IFAC-PapersOnLine*, vol. 52, no. 27, pp. 192–197, 2019, 16th IFAC Conference on Programmable Devices and Embedded Systems PDES 2019. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2405896319327041

[3] J. W. Romanishin, K. Gilpin, S. Claici, and D. Rus, "3d m-blocks: Self-reconfiguring robots capable of locomotion via pivoting in three dimensions," in *2015 IEEE International Conference on Robotics and Automation (ICRA)*, 2015, pp. 1925–1932.

[4] C. Lu and S. Wang, "The general-purpose intelligent agent," *Engineering*, vol. 6, no. 3, pp. 221–226, 2020. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2095809919309075

[5] B. Sutton, R. S., "Reinforcement learning: An introduction," 1998, MIT Press, 7 1998, also translated into Japanese and Russian.

[6] C. Hajdu, J. Hollósi, R. Krecht, Ballagi, and C. R. Pozna, "Economical mobile robot design prototype and simulation for industry 4.0 applications," in *2020 IEEE 3rd International Conference and Workshop in Óbuda on Electrical and Power Engineering (CANDO-EPE)*, 2020, pp. 000 155–000 160.

[7] A. Bozzi, S. Graffione, R. Sacile, and E. Zero, "Design and implementation of an asynchronous finite state controller for wheeled mobile robots," *Actuators*, vol. 11, no. 11, 2022. [Online]. Available: https://www.mdpi.com/2076-0825/11/11/330

[8] W. Zhao, J. Peña Queralta, and T. Westerlund, "Sim-to-real transfer in deep reinforcement learning for robotics: a survey," 09 2020.