Modelo para Planejamento de Caminho Livre de Colisão para Manejo de Guindaste utilizando DDPG

Rafaela I. Machado * **Matheus M. Santos** * **Paulo L. J. Drews-Jr** *
Silvia S. C. Botelho *

* *Universidade Federal do Rio Grande - FURG, Rio Grande RS,*
rafaelaiovanovichi,matheusmachado, paulodrews, silviacb}@furg.br

Abstract:
To aid the crane operators this work proposes the integration of Reinforcement Learning (RL), using Deep Determinism Police Gradient (DDPG) based on the concepts used in autonomous cars, to generate collision-free paths. The proposed work uses a digital twin, which is a virtual replica of the physical system to perform agent training, the reward function used in this study allow the network to learn from its mistakes and optimize its behavior to reach the target position efficiently. Punishments and rewards are given to the agent in the form of positive or negative scores, and when the laser does not identify any object, the agent will not receive any score. The success of the network's performance on the Gazebo simulator environment demonstrates its potential to solve complex problems in the real world. With further development and fine-tuning, this approach could have practical applications in industries such as manufacturing and logistics. Additionally, the use of virtual environments to test and validate the network's performance can lead to significant cost savings and reduced risk compared to physical testing.

Resumo: Para auxiliar os operadores de guindaste, este trabalho propõe a integração de Aprendizado por Reforço (RL), utilizando o *Deep Determinism Police Gradient* (DDPG) com base nos conceitos utilizados em carros autônomos, para gerar caminhos livres de colisão. O trabalho proposto utiliza um *digital twin* (gêmeo digital), que é uma réplica virtual do sistema físico, para realizar o treinamento do agente. A função de recompensa utilizada neste estudo permite que a rede aprenda com seus erros e otimize seu comportamento para alcançar a posição alvo de forma eficiente. Punições e recompensas são atribuídas ao agente na forma de pontuações positivas ou negativas, e quando o laser não identifica nenhum objeto, o agente não recebe nenhuma pontuação. O sucesso do desempenho da rede no ambiente simulador Gazebo demonstra seu potencial para resolver problemas complexos no mundo real. Com um desenvolvimento e ajuste mais aprofundados, essa abordagem poderia ter aplicações práticas em setores como manufatura e logística. Além disso, o uso de ambientes virtuais para testar e validar o desempenho da rede pode levar a economias significativas de custos e redução de riscos em comparação com testes físicos.

Keywords: Reinforcement Learning; Crane; Path; Deep Deterministic Policy Gradient; Reward.
Palavras-chaves: Aprendizado por reforço; Guindaste; Caminho; Deep Deterministic Policy Gradient; Recompensa;

1. INTRODUÇÃO

A operação de guindastes *offshore* requer a aplicação de padrões rigorosos, tanto para garantir a segurança quanto a sua eficiência, uma vez que operam em condições adversas em alto mar, onde a falta de controle sobre a carga transportada pode acarretar na perda da carga útil movimentada Neupert et al. (2008). Além disso, esses guindastes *offshore* de alto desempenho possuem uma operação complexa, que depende significativamente das habilidades do operador para coordenar o seu movimento Park et al. (2021).

Com isso, buscam-se meios de através da tecnologia que possibilitem o desenvolvimento de simuladores e técnicas para auxiliar e capacitar o guindasteiro, bem como facilitar sua visualização quanto ao ambiente de carga e descarga e também o caminho desempenhado pelos guindastes em sua movimentação Juang et al. (2013).

Um dos principais riscos na operação de guindastes é a limitada visibilidade do operador, e isso se torna um grande motivo de preocupação Milazzo et al. (2015) e Cheng and Teizer (2014). Câmeras vem senso acopladas ao equipamento expandindo o campo de visão disponível para o guindasteiro. No entanto, se torna uma tarefa difícil acompanhar a movimentação do ambiente através de um único monitor sem nenhuma informação, como por exemplo, a posição do trabalhador em relação ao guindaste ou à carga. A baixa visibilidade ou ponto cego faz com

que o operador tenha dificuldade em identificar quaisquer pessoas ou objetos na zona de trabalho Sutjaritvorakul et al. (2020).

Esses dispositivos dependem fortemente das atitudes, reflexos e experiência do operador, e é por isso que a indústria está buscando inovações na segurança da manipulação de guindastes. Os guindastes realizam a maior parte das operações portuárias e desempenham um papel significativo na indústria da construção Henrique Miranda Galhardo (2018).

Neste trabalho propõe a integração do *Reinforcement Learning* (RL), baseado nos conceitos usados em carros autônomos, para gerar caminhos livres de colisão Kiran et al. (2021). RL é um tipo de aprendizado de máquina focado em obter uma recompensa de uma determinada ação. Ele usa sistemas de *feedback* para encorajar o comportamento desejado e recompensar ações corretas ou punir ações erradas Sutton (2018).

Uma vantagem do RL sobre outros métodos é que ele não requer dados prévios para treinamento. O agente aprende com o ambiente em que está inserido, visando alcançar o comportamento ideal de um modelo dentro de um contexto específico para maximizar seu desempenho. É particularmente útil quando a única forma de coletar informações sobre o ambiente é interagindo com ele Kaelbling et al. (1996).

Algoritmos RL podem ser usados para gerar caminhos comportamentais para um sistema físico, assumindo um módulo de percepção separado para extrair coordenadas de objetos e limites de velocidade. Ao usar um gêmeo digital, que é uma réplica virtual do sistema físico, esses caminhos comportamentais podem ser gerados e testados sem nenhum risco para o sistema físico. Uma vez gerados e testados os caminhos comportamentais no ambiente digital, uma política treinada em RL pode ser utilizada para planejar um caminho com base nessas informações no ambiente físico. Essa abordagem permite testes mais eficientes e seguros de algoritmos de RL em sistemas físicos, pois o gêmeo digital pode ser usado para simular uma ampla gama de cenários e avaliar o desempenho de diferentes políticas antes de aplicá-las ao sistema físico Souza (2009).

O uso do aprendizado por reforço tem sido amplamente pesquisado para o planejamento de caminho em diversas áreas. No trabalho de Guo et al. Guo et al. (2020), aprendizado por reforço profundo é usado para propor um modelo para planejamento autônomo de caminho para navios não tripulados. Em Wang et al. (2020) e Zhao et al. (2021a), diferentes algoritmos de RL são empregados para o planejamento da caminho de robôs móveis.

Em comparação com os métodos tradicionais de planejamento de caminhos, os métodos baseados em aprendizado por reforço profundo não precisam construir todo o modelo de ambiente e podem realizar autoaprendizagem para mapeamento de ações, que possui alta flexibilidade. Por meio da interação contínua entre o robô móvel e o ambiente, o aprendizado por reforço profundo usa a estratégia de ação correspondente para determinar a próxima ação do robô móvel de acordo com o estado do robô e combina a função de recompensa para otimizar continuamente a estratégia de ação. Como um dos algoritmos típicos de aprendizado por reforço profundo aplicado ao planejamento de caminho, o *Deep Deterministic Policy Gradient* (DDPG) pode treinar o modelo em um ambiente de simulação auto construído e ser aplicado diretamente ao ambiente real com forte capacidade de generalização Gong et al. (2022).

No modelo proposto, o agente será treinado em um cenário composto por quatro obstáculos e um alvo adicionado ao ambiente com alturas e posições aleatórias dentro do raio de cobertura do guindaste. A cada episódio de treinamento, os obstáculos e o alvo têm suas posições alteradas. Nesse sistema, o algoritmo toma como entrada as leituras do laser, a posição atual das juntas do guindaste e sua distância até o alvo desejado. A saída do sistema é a nova posição do guindaste.

A estrutura do trabalho será dividida da seguinte forma: nesta primeira seção foi apresentada uma introdução sobre a problemática proposta; Na segunda seção serão discutidos alguns fundamentos relacionados ao entendimento da temática; Na terceira seção serão apresentadas as configurações para realização do experimento; Na quarta seção será apresentada a metodologia da pesquisa; Na quinta seção serão descritos os testes e os resultados; E na quinta seção será apresentada a conclusão do trabalho.

2. REFERENCIAL TEÓRICO

Neste trabalho, utilizamos o *Deep Deterministic Policy Gradient* (DDPG), que é uma técnica de reforço de aprendizagem *off-policy* utilizada para lidar com espaços de ação contínua baseada no algoritmo ator-crítico. O DDPG combina as abordagens de *Deterministic Policy Gradient* (DPG) e *Deep Q-Network* (DQN), onde o ator é uma rede de política que toma o estado como entrada e produz a ação contínua em vez de uma distribuição de probabilidade sobre as ações. O crítico é uma rede de valor Q que recebe estado e ação como entrada e gera o valor Q Lillicrap et al. (2015).

Conforme mostrado na Fig. 1, o algoritmo faz uso de uma rede neural para a rede de atores e outra para a rede crítica.

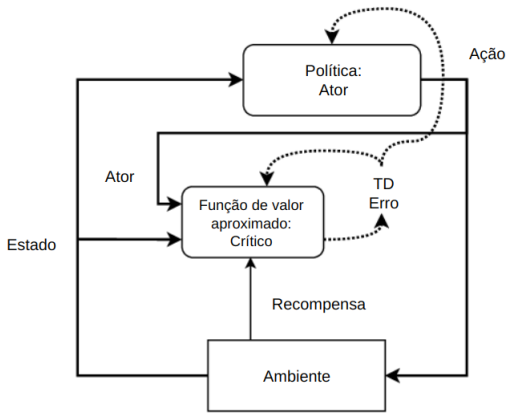

Figura 1. Arquitetura DDPG (traduzido de Garrido et al. (2019))

Na rede de atores, a entrada é o estado atual (observação do ambiente) que gera, como saída, uma ação para o

espaço de ação contínua. Na rede crítica, a entrada é a observação do ambiente e a ação correspondente, que gera como saída o valor Q estimado desse estado atual como a avaliação da ação e ajusta constantemente sua função de valor e a ação gerada pelo ator . Jesus et al. (2019) A rede de atores atual melhora continuamente a estratégia de ação de acordo com o valor-Q. O ator-alvo e as redes críticas são usados principalmente para o processo de atualização subsequente. Gong et al. (2022) O agente recebe diretamente os dados do sensor para gerar uma política ótima que planeje seu caminho seguro para o alvo Othman and Shilov (2021).

2.1 Trabalhos Relacionados

Em sua maioria, os algoritmos de planejamento de caminhos tradicionais tem baixa adaptabilidade e são adequados apenas para situações com modelos ambientais. Com o RL, os agentes podem otimizar estratégias interagindo com ambientes desconhecidos por meio de métodos de tentativa e erro e acumular recompensas (o valor das recompensas pode ser positivo ou negativo) para aprender o padrão ideal de comportamento. Além disso, o RL tem boa generalidade e é adequado para ambientes desconhecidos Zhao et al. (2021b).

O planejamento de caminhos é uma tarefa computacionalmente desafiadora de grande importância no campo da robótica. O principal objetivo do planejamento de caminho é encontrar a melhor rota entre a posição atual do robô e o alvo. Para a maioria dos robôs, o planejamento de caminho ideal é o caminho mais curto entre dois locais, no entanto, existem desafios que exigem o planejamento de um caminho que atenda a certos requisitos para lidar com condições incertas, como obstáculos em movimento e dinâmica do ambiente.

Existem algoritmos como o A* que é um algoritmo de busca informada que depende extremamente do cálculo do custo de movimento de uma determinada posição até o destino Hart et al. (1968). Alguns percursos que levam ao destino não são previamente conhecidos, por isso, o uso do algoritmo A* não encontrará o caminho mais curto, mesmo sendo um caminho viável Hu et al. (2021).

Geralmente existem outras aplicações do mundo real em que dados de mapeamento completos são inatingíveis. Um exemplo é a autocondução: é quase impossível mapear todas as vias e, mesmo assim, qualquer alteração nas vias, podem fazer com que o veículo se desloque para o local errado. Os algoritmos de aprendizado por reforço podem fazer um agente aprender a descobrir as ações mais valiosas para atingir o objetivo em um ambiente incerto, e podem produzir múltiplos caminhos Kaelbling et al. (1996).

O planejamento de caminho é a tecnologia chave para robôs móveis autônomos. Tendo em vista a escassez de caminhos encontrados pelo algoritmo tradicional *best first search* (BFS) e *rapidly-exploring random trees* (RRT) que não são curtos e suaves o suficiente para a navegação do robô. O *Q-Learning*, um algoritmo clássico de aprendizado por reforço, é usado para encontrar um caminho global para robôs. Caminhos mais curtos e suaves podem ser obtidos por este algoritmo em comparação com o algoritmo BFS e RRT Gao et al. (2019).

Com o aumento do custo da mão de obra e a preservação da saúde e bem estar humano, é tendência que os robôs substituam os humanos no meio industrial, sendo assim amplamente utilizados para a realização de tarefas em ambientes industriais hostis. O DQN é um método eficiente de aprendizado por reforço, e tem sido utilizado para planejamento de caminho de robôs móveis em ambientes desconhecidos, auxiliando em um importante problema de robôs móveis que é planejar seu caminho em um ambiente desconhecido. No entanto, é importante notar que DQN geralmente tem uma velocidade de convergência baixa Lin et al. (2022).

No trabalho Ruan et al. (2019) de Xiaogang et al., uma arquitetura Dueling Double Deep Q-Network (D3QN) é apresentada para navegação de robôs, usando uma câmera para percepção do ambiente considerando apenas imagens como entrada para navegar pelo ambiente. Os resultados do experimento mostram que o robô móvel pode atingir os alvos desejados sem colidir com nenhum obstáculo. No entanto, algoritmos como o DQN utilizam ações discretas, o que pode ser um problema em um ambiente com grande número de obstáculos. Em Miranda et al. (2022), propõe-se a utilização de Soft Actor Critic (SAC) para treinar a rede, buscando políticas de navegação que possam executar tarefas em ambientes não estruturados capazes de evitar mínimos locais.

3. CONFIGURAÇÃO EXPERIMENTAL

3.1 Robot Operating System (ROS)

O ROS é um software de código aberto que fornece comunicação de processo entre software e hardware. Possui um conjunto de bibliotecas e ferramentas de código aberto para desenvolvimento de robôs e uma infraestrutura de comunicação para troca de mensagens que podem ser facilmente enviadas para diversas ferramentas de visualização e teleoperação Quigley et al. (2009).

3.2 Unified Robotic Description Format (URDF)

O Unified Robotic Description Format (URDF) é um formato de arquivo XML usado no ROS para descrever todos os elementos de um robô, possibilitando a integração com softwares de simulação como o Gazebo Koenig and Howard (2004).

3.3 Gazebo

O Gazebo é um simulador que permite testar algoritmos, projetar robôs, realizar testes de regressão e treinar o sistema de IA usando cenários realistas. Também inclui ferramentas para criar simulações de robôs com precisão e eficiência em ambientes internos e externos complexos, levando em consideração as propriedades físicas e dinâmicas dos objetos simulados Koenig and Howard (2004).

4. METODOLOGIA

Nesta seção, fornecemos uma visão geral do processo envolvido na implementação do DDPG para planejamento de caminho. Conforme ilustrado na Fig. 2, o sistema consiste

em três elementos principais: simulação, controle e DDPG. O componente de simulação fornece as entradas visuais do sistema, enquanto o módulo de controle é responsável por indicar o novo valor de recompensa e reiniciar a simulação conforme necessário. Por fim, o DDPG, é responsável por aprender e melhorar o desempenho do planejamento de caminho do sistema.

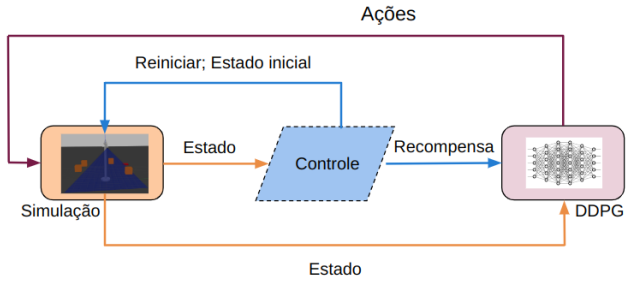

Figura 2. Visão geral do sistema

4.1 Simulação

O componente de simulação do sistema é responsável por fornecer a representação visual do guindaste, incluindo suas juntas e controladores, bem como os objetos presentes na cena e as leituras do sensor laser.

Na Figura 3, podemos ver o guindaste na cor cinza, composto por duas juntas: a junta 1, que é uma junta rotacional correspondente ao guindaste, e a junta 2, que é uma junta prismática representando o cabo na extremidade do guindaste. Além das juntas, é possível identificar a posição do laser na ponta do guindaste, cujo feixe (mostrado em azul) aponta para baixo e ajuda a detectar objetos no cenário.

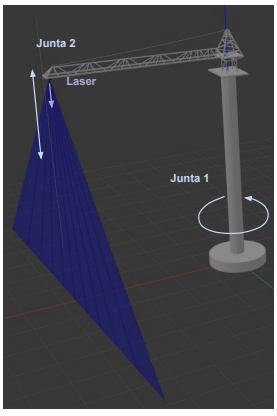

Figura 3. visão geral do guindaste

Na simulação, são obtidas as entradas do sistema, que incluem as leituras do laser, a posição atual das juntas do guindaste (junta 1 e junta 2) e sua distância ao alvo desejado. O módulo de simulação se comunica com os módulos de controle e DDPG enviando os dados de entrada do sistema, permitindo que eles tomem decisões e ações com base no estado atual do sistema.

4.2 Controle

O módulo de controle do sistema de guindaste desempenha um papel crítico no processo de treinamento. Ele é responsável por reiniciar a simulação, criando aleatoriamente obstáculos e alvos ao alcance do braço do guindaste e calculando a recompensa para o DDPG usando os parâmetros de entrada do modelo. Ao variar a posição dos obstáculos e alvo, independentemente do resultado, o processo de treinamento permanece dinâmico, evitando que o sistema repita os mesmos cenários.

O sistema de recompensa usado pelo módulo de controle é baseado em quatro parâmetros como segue:

- Recompensa positiva (+2): é dada quando a distância atual ao alvo é menor que a anterior;
- Recompensa negativa (-1): é dada quando a distância atual ao alvo é maior que a anterior;
- Recompensa positiva (+100): é dada quando o agente atinge o alvo sem bater em nenhum obstáculo;
- Recompensa negativa (-10): é dada quando o agente atinge um obstáculo.

O objetivo do agente é encontrar a posição alvo com uma distância mínima e sem colidir com nenhum obstáculo, ao atingir este objetivo, o agente recebe uma recompensa positiva de 100 pontos (+100) quando a distância entre o agente e a posição alvo for inferior a 1.5 metros. Ao receber a recompensa máxima, o alvo muda de posição, e o agente tenta reencontrar o alvo no mesmo episódio de treinamento. No entanto, se o agente colidir com algum obstáculo ou permanecer na mesma posição por mais de vinte etapas, o episódio de treinamento será encerrado com penalidade negativa de menos dez (-10).

Durante a fase de treinamento, o módulo de controle desempenha um papel crucial na orientação do sistema de aprendizado de máquina para a tomada de decisão ideal. No entanto, uma vez concluído o treinamento, o módulo de controle não é mais necessário para a aplicação final. Em vez disso, o sistema utilizará o conhecimento adquirido durante a fase de treinamento para tomar decisões informadas, sem exigir nenhuma orientação ou entrada adicional do módulo de controle.

4.3 Rede DDPG

A estrutura da rede DDPG utilizada neste estudo foi projetada com 14 entradas, conforme mostra a Figura 4. Essas entradas consistem em 11 medições de laser, o estado atual da junta 1 e junta 2 e a distância atual até o alvo.

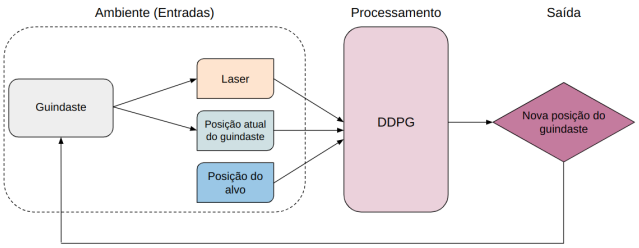

Figura 4. estrutura DDPG

Quanto à saída, a rede DDPG fornece o novo estado das juntas 1 e 2, que são executadas pelo guindaste para se aproximar do alvo evitando quaisquer obstáculos no cenário. Ao atualizar continuamente as posições das juntas com base nos parâmetros de entrada, o algoritmo DDPG

pode aprender e melhorar o desempenho do planejamento de caminho do sistema, tornando-o mais eficiente e preciso durante o treinamento.

5. RESULTADOS

O gráfico de recompensa obtido por meio do treinamento usando o código de Jesus et al. (2019), com as modificações necessárias para a operação, uma vez que o sistema original usava um robô diferencial, é mostrado na Fig. 5.

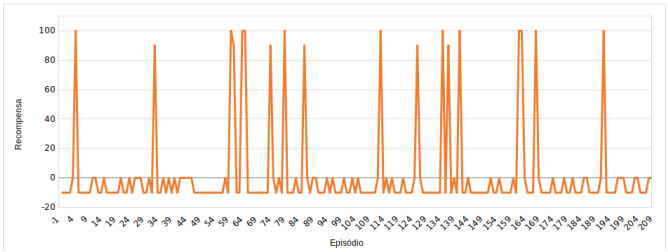

Figura 5. Recompensas usando o código adaptado

É possível observar que, no início, o agente encontra-se em processo de aprendizagem e apresenta um comportamento inadequado ao ambiente, representado pelas recompensas negativas recebidas nos primeiros instantes do treinamento. Entretanto, à medida que os episódios de treinamento progridem, o agente é capaz de alcançar o alvo em mais de uma ocasião, representado pelos picos no gráfico. Esse avanço evidencia uma melhoria no desempenho do agente ao longo do processo de treinamento.

Condições adicionais foram incluídas no sistema de recompensa para incentivar o agente a se mover em direção ao alvo de forma mais eficiente. Se a distância atual entre o guindaste e o alvo for menor do que a distância anterior, o agente recebe uma recompensa positiva de 2 pontos (+2). Por outro lado, se a distância entre o guindaste e o alvo for maior do que a distância previamente alcançada, o agente recebe uma penalização de 1 ponto (-1). Esses ajustes fornecem ao agente um *feedback* mais preciso sobre seu desempenho, incentivando-o a se mover em direção ao alvo de forma mais eficaz e evitar se afastar dele.

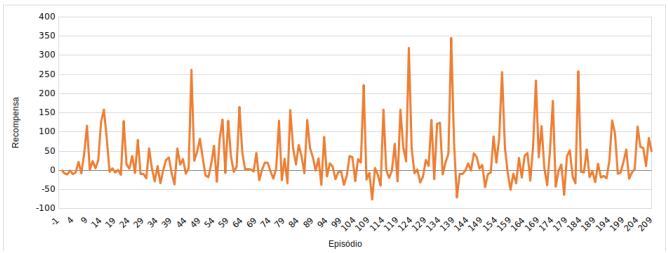

Figura 6. Quatro obstáculos depois de adicionado as novas regras de recompensa

A Fig. 7 e a Fig. 6 apresentam os resultados do treinamento com o sistema de recompensa modificado, que inclui novas regras de recompensa para incentivar o agente a se mover em direção ao alvo. O número de episódios bem-sucedidos representados pelos picos nos gráficos 6 e 7 demonstra quando o agente encontrou o alvo sem colidir com quaisquer obstáculos. Os experimentos demonstram que as modificações feitas no sistema de recompensa impactaram

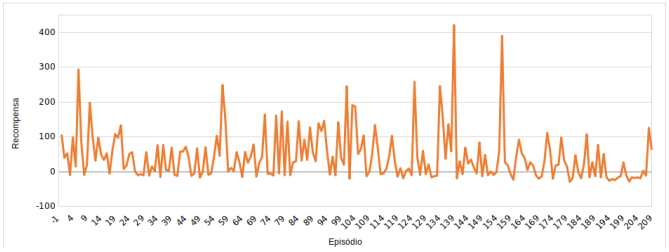

Figura 7. Dois obstáculos depois de adicionado as novas regras de recompensa

positivamente o processo de aprendizado, como pode ser visto no aumento dos picos de recompensa nas Fig. 7 e 6, e aumentaram a eficácia da tomada de decisão do agente, o que resultou em uma taxa de sucesso maior em encontrar o alvo enquanto evita colisões.

Além disso, é possível concluir que o agente obtém resultados superiores em ambientes com menos obstáculos, como demonstrado no cenário apresentado na Figura 7. Todos os testes realizados neste estudo foram executados com o mesmo número de episódios de treinamento. No entanto, para cenários mais complexos, como aqueles com quatro obstáculos, um número maior de episódios de treinamento é necessário para alcançar os mesmos resultados obtidos em ambientes mais simples.

6. CONCLUSÃO

Muitos algoritmos convencionais de planejamento de caminhos exibem adaptabilidade limitada e são viáveis apenas em situações em que modelos ambientais estão disponíveis. Por outro lado, os agentes de aprendizado por reforço (RL) podem otimizar suas estratégias interagindo iterativamente com ambientes desconhecidos, contando com métodos de tentativa e erro e acumulando recompensas para aprender padrões ideais de comportamento. Além disso, o RL possui uma generalidade impressionante e é particularmente adequado para navegar em territórios desconhecidos.

A implementação da rede neural DDPG para navegação de guindastes, utilizando controle contínuo em ambientes virtuais, é uma solução promissora para aplicações do mundo real. Para reduzir o número de acidentes e facilitar a operação do equipamento, este trabalho apresenta um sistema de planejamento de rotas sem colisões para movimentação de guindastes, desenvolvido por meio de aprendizado por reforço.

A função de recompensa proposta permitiu que a rede aprendesse com seus erros e otimizasse seu comportamento para atingir a posição alvo com eficiência. O sucesso do desempenho da rede nos ambientes de simulação Gazebo demonstra seu potencial para resolver problemas complexos no mundo real. Com mais desenvolvimento e ajuste fino, essa abordagem pode ter aplicações práticas em setores como manufatura e logística. Além disso, o uso de ambientes virtuais para testar e validar o desempenho da rede pode levar a economias de custos significativas e riscos reduzidos em comparação com os testes físicos. A aplicação de técnicas de aprendizado por reforço profundo em sistemas de navegação e controle é uma área interessante

de pesquisa que tem o potencial de revolucionar vários setores.

As próximas etapas do trabalho visam realizar treinos mais robustos com diferentes números de obstáculos no cenário. Além disso, planejamos realizar testes e comparações com outros algoritmos de aprendizado por reforço, como *Soft Actor-Critic* e *Proximal Policy Optimization*.

REFERÊNCIAS

Cheng, T. and Teizer, J. (2014). Modeling tower crane operator visibility to minimize the risk of limited situational awareness. *Journal of Computing in Civil Engineering*, 28(3), 04014004.

Gao, P., Liu, Z., Wu, Z., and Wang, D. (2019). A global path planning algorithm for robots using reinforcement learning. In *2019 IEEE International Conference on Robotics and Biomimetics (ROBIO)*, 1693–1698. IEEE.

Garrido, L.A., Nishtala, R., and Carpenter, P. (2019). Continuous-action reinforcement learning for memory allocation in virtualized servers. In *High Performance Computing: ISC High Performance 2019 International Workshops, Frankfurt, Germany, June 16-20, 2019, Revised Selected Papers 34*, 13–24. Springer.

Gong, H., Wang, P., Ni, C., and Cheng, N. (2022). Efficient path planning for mobile robot based on deep deterministic policy gradient. *Sensors*, 22(9), 3579.

Guo, S., Zhang, X., Zheng, Y., and Du, Y. (2020). An autonomous path planning model for unmanned ships based on deep reinforcement learning. *Sensors*, 20(2), 426.

Hart, P.E., Nilsson, N.J., and Raphael, B. (1968). A formal basis for the heuristic determination of minimum cost paths. *IEEE transactions on Systems Science and Cybernetics*, 4(2), 100–107.

Henrique Miranda Galhardo, B. (2018). Guindaste offshore: Modelagem e simulação baseadas em helicoides. Dissertação (Mestrado) - Instituto Militar de Engenharia.

Hu, Y., Yang, L., and Lou, Y. (2021). Path planning with q-learning. In *Journal of Physics: Conference Series*, volume 1948, 012038. IOP Publishing.

Jesus, J.C., Bottega, J.A., Cuadros, M.A., and Gamarra, D.F. (2019). Deep deterministic policy gradient for navigation of mobile robots in simulated environments. In *2019 19th International Conference on Advanced Robotics (ICAR)*, 362–367. IEEE.

Juang, J.R., Hung, W.H., and Kang, S.C. (2013). Simcrane 3d+: A crane simulator with kinesthetic and stereoscopic vision. *Advanced Engineering Informatics*, 27(4), 506–518.

Kaelbling, L.P., Littman, M.L., and Moore, A.W. (1996). Reinforcement learning: A survey. *Journal of artificial intelligence research*, 4, 237–285.

Kiran, B.R., Sobh, I., Talpaert, V., Mannion, P., Al Sallab, A.A., Yogamani, S., and Pérez, P. (2021). Deep reinforcement learning for autonomous driving: A survey. *IEEE Transactions on Intelligent Transportation Systems*, 23(6), 4909–4926.

Koenig, N. and Howard, A. (2004). Design and use paradigms for gazebo, an open-source multi-robot simulator. In *2004 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)(IEEE Cat. No. 04CH37566)*, volume 3, 2149–2154. IEEE.

Lillicrap, T.P., Hunt, J.J., Pritzel, A., Heess, N., Erez, T., Tassa, Y., Silver, D., and Wierstra, D. (2015). Continuous control with deep reinforcement learning. *arXiv preprint arXiv:1509.02971*.

Lin, S., Liu, A., Wang, J., and Kong, X. (2022). A review of path-planning approaches for multiple mobile robots. *Machines*, 10(9), 773.

Milazzo, M.F., Ancione, G., Spasojevic Brkic, V., et al. (2015). Safety in crane operations: An overview on crane-related accidents. In *Proceedings of the 6th International Symposium on Industrial Engineering, SIE*, 36–39.

Miranda, V.R., Neto, A.A., Freitas, G.M., and Mozelli, L.A. (2022). On the generalization of deep reinforcement learning methods in the problem of local navigation. *arXiv preprint arXiv:2209.14271*.

Neupert, J., Mahl, T., Haessig, B., Sawodny, O., and Schneider, K. (2008). A heave compensation approach for offshore cranes. In *2008 American Control Conference*, 538–543. IEEE.

Othman, W. and Shilov, N. (2021). Deep reinforcement learning for path planning by cooperative robots: Existing approaches and challenges. In *2021 28th Conference of Open Innovations Association (FRUCT)*, 349–357. IEEE.

Park, H.C., Chakir, S., Kim, Y.B., and Lee, D.H. (2021). A robust payload control system design for offshore cranes: Experimental study. *Electronics*, 10(4).

Quigley, M., Conley, K., Gerkey, B., Faust, J., Foote, T., Leibs, J., Wheeler, R., Ng, A.Y., et al. (2009). Ros: an open-source robot operating system. In *ICRA workshop on open source software*, volume 3, 5. Kobe, Japan.

Ruan, X., Ren, D., Zhu, X., and Huang, J. (2019). Mobile robot navigation based on deep reinforcement learning. In *2019 Chinese control and decision conference (CCDC)*, 6174–6178. IEEE.

Souza, E.J.C.d. (2009). *Controle anti-oscilatório de tempo mínimo para guindaste usando a programação linear*. Ph.D. thesis, Universidade de São Paulo.

Sutjaritvorakul, T., Vierling, A., and Berns, K. (2020). Data-driven worker detection from load-view crane camera. In *ISARC. Proceedings of the International Symposium on Automation and Robotics in Construction*, volume 37, 864–871. IAARC Publications.

Sutton, Richard S.and Barto, A.G. (2018). *REINFORCEMENT LEARNING: AN INTRODUCTION*. MIT press.

Wang, Y., Fang, Y., Lou, P., Yan, J., and Liu, N. (2020). Deep reinforcement learning based path planning for mobile robot in unknown environment. In *Journal of Physics: Conference Series*, volume 1576, 012009. IOP Publishing.

Zhao, Y., Wang, X., Wang, R., Yang, Y., and Lv, F. (2021a). Path planning for mobile robots based on tpr-ddpg. In *2021 International Joint Conference on Neural Networks (IJCNN)*, 1–8. IEEE.

Zhao, Y., Wang, X., Wang, R., Yang, Y., and Lv, F. (2021b). Path planning for mobile robots based on tpr-ddpg. In *2021 International Joint Conference on Neural Networks (IJCNN)*, 1–8. IEEE.