Aplicações de Aprendizado por Reforço em Manipuladores Robóticos: Uma Revisão Sistemática

Franklin Andrade de Brito
RAI - Robotics and Artificial Intelligence
Centro de Ciências Exatas e Tecnológicas
Universidade Federal do Recôncavo da Bahia
Cruz das Almas, Brasil
frank.engcet@gmail.com

André Luiz Carvalho Ottoni
RAI - Robotics and Artificial Intelligence
Centro de Ciências Exatas e Tecnológicas
Universidade Federal do Recôncavo da Bahia
Cruz das Almas, Brasil
andre.ottoni@ufrb.edu.br

Lara Toledo Cordeiro Ottoni
Universidade Federal da Bahia
Salvador, Brasil
lara.toledo@ufba.br

***Resumo* — O Aprendizado por Reforço (AR) tem diversas aplicações em tecnologia como um método expoente de resolução de problemas otimizando ações por meio de recompensas. Existe uma gama de estudos aplicados à robótica com o objetivo de aprimorar o estado da arte nesta área. Assim, este trabalho visa discutir as aplicações de AR em manipuladores robóticos através de uma revisão sistemática da literatura, onde foram analisados 38 trabalhos da área publicados entre 2013 e 2023. Desta forma, foram elaboradas 6 perguntas de pesquisa para o desenvolvimento do trabalho. Baseado nestas perguntas, foi possível destacar os resultados da resvisão sistemática. Entre as técnicas discutidas, *Q-Learning* (23,68%), *Deep Reinforcement Learning* (28,95%), *Actor-Critic* (26,32%) e *Policy Gradient* (21,05%) foram as principais. Entre os ambientes e equipamentos para experimentação física e simulada, os mais utilziados foram o simulador Matlab (18,42%) e o manipulador UR3 (7,89%). Também foram apresentados resultados sobre o ajuste de hiperparâmetros, sendo que apenas 10,53% dos trabalhos realizaram o ajuste. Além disso, foi realizada uma comparação com outros trabalhos de revisão sistemática do tema proposto. Por fim, foram discutidas as perguntas de pesquisa deste trabalho e apresentadas as principais indicações de trabalhos futuros para promover a continuidade do desenvolvimento de aplicações na área.**

***Palavras-Chave* — Aprendizado por reforço, Manipuladores Robóticos, Hiperparâmetros, Simuladores, Revisão de literatura.**

1. Introdução

O Aprendizado por reforço (AR) provê métodos para soluções de problemas os quais podem ser notados e estudados em numerosas áreas, inclusive possuindo diversas aplicações em manipuladores robóticos [39]. Por exemplo, o trabalho desenvolvido por [48] que apresenta um esquema de controle adaptativo baseado em aprendizado por reforço para manipuladores robóticos. O estudo de [29] realiza uma análise de desempenho acerca de AR para resolução do problema da mochila multidimensional. Por outro lado, o estudo de [8] apresenta uma abordagem baseada em AR para promover a segurança na operação do sistema de energia através de mudanças autônomas na topologia. Assim, o estudo e as aplicações de AR têm potencial para promover avanços significativos na indústria 4.0 e otimização de controle para robôs [15].

Algumas revisões sobre AR aplicado a manipuladores robóticos se encontram na literatura. Apesar disso, o presente trabalho se diferencia dos demais, devido a ser o único a apresentar uma abordagem acerca do ajuste de hiperparâmetros e, juntamente com [50], ser o único trabalho que aborda os manipuladores e simuladores utilizados nos trabalhos analisados. Entre os trabalhos de revisão sistemática da área presentes na literatura estão [11] e [23], os quais apresentam e examinam os problemas envolvidos na aplicação de *Deep Reinforcement Learning* em manipuladores robóticos e suas soluções. Ambos trabalhos também apresentam avanços recentes relacionados à implementação de *Deep Reinforcement Learning* para tarefas de manipulação robótica. Os autores de [28], além de apresentarem limitações de algoritimos atuais em suas aplicações em manipuladores, também discutem fundamentos teóricos essenciais acerca dos algoritimos de AR. Por fim, o trabalho [26] estuda artigos relevantes sobre manipulação baseada em *Deep Reinforcement Learning*.

Nesse contexto, o objetivo deste trabalho é apresentar uma revisão sistemática acerca das aplicações de aprendizado por reforço em manipuladores robóticos, apresentando as técnicas utilizadas, resultados, metodologias e indicações para trabalhos futuros. Ressalta-se que, para isso, foram considerados trabalhos publicados em periódicos com fatores de impacto acima de 3.

Este trabalho possui 4 seções. A seção 2 apresenta o método de pesquisa aplicada e suas especificidades. Posteriormente a seção 3 expõe os resultados da revisão sistemática acerca de pontos relevantes, além de um comparativo do presente trabalho com outras revisões sistemáticas da literatura. E por fim, a seção 4 apresenta a conclusão e indicação de trabalhos futuros.

2. Método de Pesquisa

O método de pesquisa proposto para o desenvolvimento da pesquisa, ilustrado na Figura 1, foi definida em etapas. Na primeira etapa, foi definida a base de dados, provinda do site Scopus e, logo após, os periódicos dos quais seriam obtidos os trabalhos a serem analisados. Os periódicos foram selecionados com base em seu respectivo fator de impacto (FI) e sua relevância no que se refere ao tema, sendo eles mencionados a seguir:

- *Robotics and Autonomous Systems*, FI = 8,060.
- *Applied Sciences (Switzerland)*, FI = 3,700.
- *Robotics*, FI = 4,899.
- *Neural Computing and Applications*, FI = 5,102.
- *IEEE Access*, FI = 3,476.
- *Engineering Applications of Artificial Intelligence*, FI = 7,802.
- *Neurocomputing*, FI = 5,779.
- *Sensors*, FI = 3,847.

As revistas de base supracitadas estão de acordo com a área de pesquisa em foco e possuem um relevante fator de impacto, tornando-as significativas no tocante às etapas sequentes. Posteriormente, foram definidas as combinações de 2 expressões de busca para que houvesse o retorno de trabalhos relacionados ao tema proposto durante a pesquisa. As expressões utilizadas foram: *"Reinforcement learning" "Robotic manipulator"*, *"Reinforcement learning" "Robot manipulator"*, *"Reinforcement learning" "Robotic arm"*, *"Reinforcement learning" "Robot arm"*, *"Q-learning" "Robotic manipulator"*, *"Q-learning" "Robot manipulator"*, *"Q-learning" "Robotic arm"*, *"Q-learning" "Robot arm"*, *"Sarsa" "Robotic manipulator"*, *"Sarsa" "Robot manipulator"*, *"Sarsa" "Robotic arm"* e *"Sarsa" "Robot arm"*.

O período de enfoque para a coleção dos trabalhos foi definido entre 2013 e 2023, sendo assim, trabalhos publicados nos últimos 10 anos. Após a análise e obtenção dos trabalhos que seguiram os requerimentos previamente definidos, também foram estabelecidos critérios de exclusão de artigos da revisão:

- O trabalho não utiliza ou estuda técnicas de AR a serem aplicadas em um manipulador robótico;
- Trabalho repetido em 2 ou mais seções de pesquisa.

Após a etapa de exclusão, Foram determinadas as 6 seguintes perguntas de pesquisa:

(1) Quais manipuladores são adotados nos artigos?
(2) Quais técnicas de AR são aplicadas?
(3) O trabalho utiliza simuladores? Quais?
(4) O trabalho realiza ajuste de hiperparâmetros de AR?
(5) Quais os principais resultados da aplicação de AR no manipulador?
(6) Quais as tendências para a área? Observando o que os artigos mais recentes indicam como trabalhos futuros.

3. Resultados da Revisão Sistemática

Esta seção é dividida em 5 subseções. A subseção A apresenta e discute acerca dos manipuladores robóticos e simuladores utilizados pelos trabalhos da revisão. A subseção B discute sobre os algoritmos de AR aplicados a manipuladores robóticos. Já a subseção C apresenta e discute acerca dos trabalhos que fizeram ou não o ajuste de hiperparâmetros. Na subseção D é feita uma comparação entre a presente revisão e demais trabalhos da literatura de revisão sistemáticas sobre do tema. E a subseção E responde e discute as perguntas de pesquisa propostas neste trabalho.

A. *Manipuladores Robóticos e Simuladores*

A partir dos dados exibidos na Tabela 1, conclui-se que 28,95% dos trabalhos estudados não utilizam manipuladores reais, mas simuladores para validar os experimentos e suas respectivas abordagens propostas. Estas abordagens foram: aprendizado de movimentos ponto a ponto [3], aprendizado com feedback interativo [27], algoritmos de interação contínua [24], manipulação de objetos [35], tarefa de abertura de porta [45], manipulação coordenada de multi-robôs [20], controle neural adaptativo [42], controle de manipuladores [18], planejamento de trajetória [47], inspeção robótica [14] e controle de posição [49].

Tabela I
Manipuladores robóticos utilizados nos trabalhos avaliados na revisão sistemática

Manipuladores	Trabalhos
NEXTAGE	[43]
Yumi	[9]
UR3	[6,22,4]
BCN3D moveo	[44]
IRB 1600	[1,2]
Mitsubishi RV-12SDL-12S	[21]
UR5	[31,22]
RM-X52	[32,33]
PANDA	[10,34]
NAO	[12]
KUKA	[12]
Sawyer	[52]
SCARA	[39]
Openmanipulator	[16]
robotiq 2f-85 gripper	[41]
Produzido em laboratório	[36,46,37]
Não informou	[48,17,19,38]
Não utilizou	[3,27,24,35,45,20,42,18,47,14,49]

Verifica-se também que a variedade de manipuladores robóticos utilizados é ampla, sendo o modelo UR3 da Universal Robots o mais utilizado entre estes em trabalhos com enfoque em: Tarefa *peg-in-hole* [6,4] e controle de braço duplo robótico [23]; seguido do PANDA [10,34], UR5 [31,22], RM-X52 [32,33] e IRB 1600 [1,2]. Além disso, 3 trabalhos fizeram o uso de manipuladores produzidos em laboratório, customizados ou com peças impressas em 3D, implementados em: Controle de articulações robóticas [36], planejamento de movimento [46] e mapeamento de controlador de braço robótico [37]. Já 10,53% dos artigos não informaram o modelo de manipulador implementado no trabalho.

É notório, a partir das análises apresentadas na Tabela 2, que 13,16% dos trabalhos da revisão [6,44,12,31,38] não utilizam

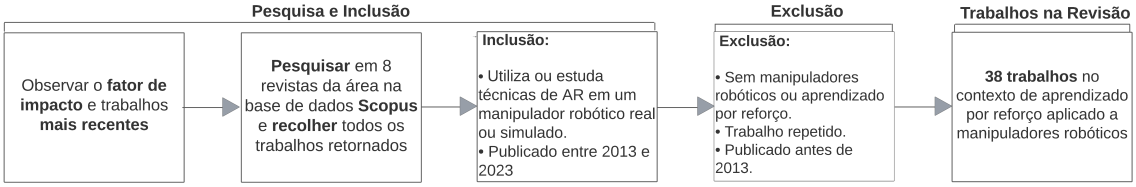

Figura 1. Fluxograma para seleção de artigos da revisão sistemática

simuladores, mas fazem puramente o uso de manipuladores robóticos físicos no processo desenvolvimento dos experimentos devidos para com as metodologias propostas de cada trabalho, sendo elas respectivamente aplicadas em: tarefa *peg-in-hole* [6], manipulação de roupas [44], estudo de efeito [12], método de compensação [31] e preensão de objetos [38].

Tabela II
SIMULADORES UTILIZADOS NOS TRABALHOS AVALIADOS NA REVISÃO SISTEMÁTICA

Simulador	Trabalhos
Matlab	[3,49,18,52,33,16,41]
ROS-Indigo	[14]
Pybullet	[34]
RoboDK	[1,2]
Gazebo	[32,46,16,22,4]
MuJoCo	[10,36,21,35,37,23]
CoppeliaSim	[18,27,45,47]
Tensor Flow/Keras	[43]
RobotStudio	[9]
Não utiliza	[6,44,12,31,38]
Não informado	[19,42,20,39,48,17]

Entre os simuladores mais utilizados estão o Matlab e o MuJoco, seguidos do Gazebo, CoppeliaSim e RoboDK. Os demais simuladores foram utilizados em 1 trabalho cada. Por outro lado, 15,79% dos trabalhos não informaram o simulador utilizado.

B. Técnicas de AR utilizadas nos trabalhos analisados

A Tabela 3 apresenta a divisão dos trabalhos analisados com base nos algoritmos de aprendizado por reforço e seus derivados aplicados ou estudados nos mesmos.

*1) **Actor-Critic**:* O método de AR *Actor-Critic (AC)* foi utilizado, por exemplo, em [42] visando implementar um algoritmo de controle adaptativo para uma classe de manipuladores robóticos. Através dos resultados obtidos, observou-se que, utilizando a abordagem proposta, obteve-se uma entrada de controle ótima e um bom desempenho de rastreamento dos manipuladores. Uma vertente do AC, o método *Continuous Actor Critic Learning Automaton (CACLA)*, foi utilizado por [12] no estudo de efeito da frequência de feedback em Aprendizado por Reforço Interativo. Neste trabalho foram feitos experimentos em manipuladores robóticos para a análise dos efeitos estudados e foi descoberto que a interpretação da melhor taxa de interação com um professor é influenciada tanto pela complexidade da tarefa, quanto pelo limite de performance; além disso, foi verificado que a taxa de interação ótima muda de acordo com o tempo e que uma trajetória ótima é determinada pela complexidade da tarefa.

Tabela III
TÉCNICAS DE APRENDIZADO POR REFORÇO ABORDADAS NOS TRABALHOS AVALIADOS NA REVISÃO SISTEMÁTICA

Simulador	Trabalhos
Q-Learning	[1,2,44,41,39,37,48,49,52]
Deep Reinforcement Learning	[52,17,47,16,21,38,35,27,37,23,43]
Actor-Critic	[12,36,10,32,17,33,4,20,42,46]
Policy Gradient	[23,38,45,21,47,16,22,10]
Policy Improvement With Path Integrals	[19,3]
Linguistic Lyapunov fuzzy Reinforcement Learning	[18]
Reinforcement Learning from Expert Demonstrations	[34]
Policy Learning by Weighting Exploration With the Returns	[6]
e-greedy Forward Tree Search	[14]
Hierarchical Reinforcement Learning	[9]
Interactive Robust	[45]
Input Compensation	[31]
Reference Compensation	[31]

O método *Soft-Actor-Critic (SAC)* foi estudado e aplicado em alguns dos trabalhos presentes na revisão [36,46,17] e também juntamente ao Algoritmo *Deep Reinforcement Learning (DRL)*. Um exemplo dessa combinação é apresentado no trabalho desenvolvido por [33], no qual se utiliza do algoritmo proposto para solucionar o problema de planejamento de caminho para manipuladores robóticos multi-braços com obstáculos em movimento periódico. Os resultados do trabalho demonstram que o algoritmo proposto gera caminhos mais curtos e mais suaves comparado a resultados já existentes na literatura quando há um obstáculo se movendo periodicamente. Outro exemplo do uso de ambos SAC e DRL foi estudado por [4], de modo a propor um método para resolver tarefas *peg-in-hole* com incerteza de posição do furo e finalmente obtendo uma alta taxa de sucesso no resultado de experimentos feitos sob condições variáveis, rigidez do ambiente, novas tarefas de inserção e incertezas de posicionamento.

Em [17] foi implementado um algoritmo baseado em SAC para planejamento de caminho juntamente com LSTM *(long short-term memory)* em manipuladores multi-braços para realizar a previsão da posição de obstáculos em movimento e foi demonstrado nos resultados que o planejamento abordado teve sucesso no cálculo do caminho ótimo sendo com obstáculos móveis ou fixos, obtendo 100% de taxa de sucesso na geração de caminhos.

Os autores de [20] aplicaram o algoritmo *Natural Actor-Critic (NAC)* no controle para manipulação coordenada de multi-robôs. Os resultados demonstraram que o controle proposto atinge o desempenho de rastreamento do objeto manipulado por multi-robôs enquanto a conformidade de cada manipulador robótico individual também é garantida.

O estudo de [10] fez uma comparação entre os resultados obtidos com dois algoritmos: *Actor Critic Using Kronecker-Factored Trust Region* (ACKTR) e *Proximal Policy Optimization* (PPO2), os quais foram aplicados no estudo do impacto da compensação da gravidade nas tarefas de alcance para manipuladores robóticos. E apesar de ter uma variação significativamente maior entre cada sessão de treinamento, PPO2 conseguiu treinar 5 vezes mais rápido do que o ACKTR.

2) ***Q-Learning****:* Em [2] foi aplicado *Q-Learning* em combinação com Redes Neurais para planejamento de caminho de braços robóticos baseado em visão computacional em um espaço tridimensional. Os resultados mostraram que, com a implementação do *Q-Learning*, foi possível determinar sequências de ações ideais e evitar obstáculos, superando limitações de pesquisas anteriores feitas pelos próprios autores. Além disso, em [1] foi demonstrado que o método híbrido baseado em *Q-Learning* e redes neurais proposto provê melhorias expressivas quanto à lentidão e a complexidade do problema de planejamento de caminhos.

Alguns trabalhos aplicaram *Q-Learning* em conjunto com lógica fuzzy, alcançando: estabilidade e segurança no controle de manipuladores robóticos de dois elos e 2 graus de liberdade [39]; uma boa acurácia e performance de rastreamento para cada junta de um manipulador minimizando os erros de rastreamento de estado estacionário, mesmo com a presença da incerteza paramétrica do sistema e distúrbios externos [49]; e o decrescimento do efeito de trepidação e melhora na precisão de rastreamento [48].

Fitted Q-Learning foi o método abordado por [44] para manipulação robótica de roupas. Foi demonstrado que o método proposto aplicado a uma plataforma robótica dupla de preço acessível pode aprender a dobrar uma peça têxtil incluindo a sensação tátil no processo de aprendizagem.

Outra importante aplicação de *Q-Learning*, em conjunto com DRL foi executada por [37] no mapeamento *Sim-Real* de um controlador de braço robótico baseado em imagem, onde foi feita uma comparação entre um sistema de transferência de aprendizado convencional DRL e o método com mapeamento proposto. Concluiu-se que o sistema proposto obteve uma taxa de sucesso de 100% na tarefa de preensão, sendo superior ao sistema convencional que obteve entre 15 e 57% de taxa de sucesso a depender da posição do objeto.

O estudo de [41] não propôs uma técnica de AR, porém fez comparações entre a técnica proposta (*artificial bee colony algorithm(ABC)*) e o algorítimo *Q-Learning* para a tarefa de paletização utilizando um braço robótico. Os resultados demonstraram que a abordagem baseada em AR é capaz de obter a mesma solução que a proposta baseada no algoritmo ABC. Apesar disso, na tentativa de minimizar o consumo de energia elétrica ou manter os recipientes igualmente cheios na tarefa proposta a abordagem em AR acaba não sendo trivial, além de não garantir que as restrições serão atendidas visto que não há um mecanismo de tratamento para as mesmas. Já o algoritmo ABC mostrou ser facilmente extensível em relação a restrições adicionais e objetivos.

3) ***Deep Reinforcement Learning (DRL)****:* O trabalho [27] apresenta resultados de 3 algoritmos: *Deep Reinforcement Learning (DRL)*, *Interactive Deep Reinforcement Learning (Agent-IDeepRL)* utilizando um agente previamente treinado como orientador interativo e Human-IDeepRL utilizando um Humano como orientador interativo do robô a ser treinado. Foi demonstrado que os algoritmos interativos superam o DRL convencional sem feedback, obtendo recompensas mais rápidas durante o aprendizado de tarefas domésticas realizadas por um robô manipulador. [47] também apresenta resultados de 3 outros algoritmos de DRL de última geração baseados em métodos de *Policy Gradient e Actor-Critic: Deep Deterministic Policy Gradient (DDPG), Asynchronous Advantage Actor-Critic (A3C)* e *Distributed Proximal Policy Optimization (DPPO)*. O artigo demonstrou que os métodos e funções de recompensas abordadas podem melhorar a taxa de convergência, a precisão e a robustez do planejamento de trajetória para manipuladores robóticos.

Em [43] foi feita a comparação dos algoritmos propostos (*Deep P-Network* (DPN) e *Dueling Deep P-Network* (DDPN)) com métodos de DRL anteriores presentes na literatura em uma tarefa de alcance de um manipulador simulado, além de tarefas exercidas por um robô de braço duplo real tendo resultados que demonstraram eficiência e estabilidade de aprendizagem superiores aos métodos anteriores.

O método de DRL baseado em sinergia proposto por [35] foi utilizado para manipular objetos com uma mão robótica antropomórfica. Através dessa metodologia, foi possível atingir uma taxa média de sucesso de 88,38% para as tarefas de manipulação de objetos, enquanto o DRL padrão sem espaço de sinergia atinge apenas 50,66%. Os resultados qualitativos mostram que a política de DRL baseada em sinergia proposta produz posicionamentos de dedos semelhantes aos humanos sobre a superfície de cada objeto utilizado nos experimentos.

Em [51] foi aplicado *Deep Q-Network* (DQN) para solução de cinemática inversa de um manipulador robótico de alto grau de liberdade, um algoritmo baseado em DRL que provou possuir uma arquitetura mais simples e ser uma solução com boa acurácia em um tempo de computação razoável, além de poder ser modificado para diferentes objetivos e necessidades. Apesar da incerteza ao calcular vários pontos iniciais da trajetória do manipulador relacionado aos pontos da trajetória nos quais o algoritmo possui uma maior perda e aptidão, o agente foi capaz de computar diferentes tipos de trajetórias e encontrar novas trajetórias a cada execução.

4) ***Policy Gradient****:* Outros trabalhos analisados também aplicam métodos de *Policy Gradient* baseados em DRL. Por exemplo, No trabalho [23] é feita a implementação do algoritmo *Dual-arm Deep Deterministic Policy Gradient* (DADDPG) para controle colaborativo de um braço robótico duplo. Os resultados evidenciam que o algoritmo permite

que o robô aprenda o movimento cooperativo de forma autônoma e que pode concluir tarefas colaborativas simples como ”alcançar”, ”pegar”e ”empurrar”. Por outro lado, os autores de [38] estudam a utilização de DDPG no controle de braços robóticos com o fim de agarrar objetos de forma autônoma e demonstra experimentalmente que o algoritmo utilizado pode atingir um objeto alvo com uma cinemática inversa do braço do robô, além de atingir uma precisão de 95,5% na tarefa de preensão. Com outro objetivo, o trabalho [21] implementa DDPG para controle de um braço robótico antropomórfico a fim de evitar obstáculos. Testes realizados comprovaram que o algoritmo é confiável e provê um bom prognóstico para a implementação do mesmo na indústria. Sob outro enfoque, o trabalho [16] aplica o método *Twin Delayed Deep Deterministic Policy Gradient* (TD3) no planejamento de movimento de robôs manipuladores proporcionando caminhos mais suaves e curtos em geral com melhor eficiência de operação.

O estudo de [45] aplicou e demonstrou que o método PPO aprimorado tem alta precisão e estabilidade ao concluir a operação de abertura de porta realizada por um braço robótico móvel.

Manipulation-Actor Proximal Policy Optimization (MAPPO) foi o método proposto por [22] para aquisição de habilidades de manipulação por um robô. O mesmo foi comparado com PPO e A3C e foi verificado que o algoritmo MAPPO melhora a taxa de aprendizado em 33,27% e 41,07% e reduz consideravelmente tentativas inúteis no processo de aprendizagem.

5) ***Outras técnicas de AR****:* O estudo feito em [31] propõe dois métodos de compensação baseados em AR para robôs manipuladores, sendo eles: *input compensation* e *Reference Compensation*. Ambos foram abordados e comparados com 3 métodos de aprendizado conhecidos: proportional-derivative (PD), *model predictive control (MPC)*, e *Iterative Learning Control (ILC)*. O método *Input Compensation* demonstrou vantagens na velocidade de resposta e menores erros em comparação ao segundo método proposto, apesar de quê, o mesmo é mais suscetível ao comportamento oscilatório que é normalmente induzida por ruídos ou incertezas no sistema e resulta em uma velocidade de aprendizagem mais lenta quando se mantém seguro o processo de aprendizagem. Por outro lado, o método *Reference Compensation* demonstrou vantagem na suavidade da resposta da resposta de rastreamento e na velocidade de convergência em comparação com o primeiro método, apesar de que o segundo método possui uma resposta menos agressiva. Já os resultados das comparações mostraram que os métodos propostos possuem uma melhor resposta do que o MPC e o ILC para uma referência descontínua e trajetórias suaves e mais simples, enquanto para uma tarefa mais complexa, o MPC e o ILC superam os métodos propostos.

O trabalho [14] aborda a técnica *e-greedy Forward Tree Search (FTS)* na realização de busca de política de planejamento online, somado à formulação de Processos de decisão de Markov (PDM) para a inspeção robotizada. No trabalho foi feita a comparação entre o método baseado em *Next-Best-View* (NBV) e o método proposto, concluindo-se que o último é capaz de superar o NBV no processo de inspeção. Além de que a utilização do algoritmo abordado juntamente ao PDM revelou ser capaz de lidar com o problema de planejamento de cobertura para inspeção robótica.

O método *Reinforcement Learning from Expert Demonstrations (RLED)*, o qual é uma combinação de AR com *Imitation Learning (IL)* foi aplicado em [34] e avaliado como uma técnica viável para acelerar o processo de AR com sucesso, apesar de apresentar limites quanto a segurança em relação à exploração tendenciosa.

A técnica nomeada como *Linguistic Lyapunov reinforcement learning control (LLRLC)* proposta em [18] demonstrou ter um desempenho de rastreamento superior, menor requisito de torque de controle e possuir um custo computacional mais baixo quando comparados a *Fuzzy Q-Learning (FQL)* e *Lyapunov theory based Markov game controller (LMGC)*.

O método proposto por [24] caracterizado por ser uma junção entre *Robust Reinforcement Learning (RRL)* e *Interactive Reinforcement Learning (IRL)* e nomeado como *Interactive Robust Reinforcement Learning (IRRL)* foi aplicado à tarefa de organização de objetos por um braço robótico simulado. O agente proposto conseguiu aprender a tarefa em um tempo de treinamento menor comparado a agentes autônomos. Os resultados mostraram que a abordagem implementada tem um bom funcionamento em ambientes contínuos e grandes sem necessitar de conhecimento prévio do modelo ou comportamento do estado.

Por outro lado, o trabalho [19] apresentou um estudo a cerca do evitamento de obstáculos utilizando a estrutura *Dynamic movement primitives (DMPs)* e implementando o algoritmo PI². Os resultados das simulações mostraram que a aprendizagem de potencial e forma na estrutura de AR proposta é válida e posteriormente foi verificado que a abordagem aplicada ao robô real de 7 graus de liberdade obteve os resultados esperados. Também utilizando o mesmo algoritmo, os autores de [3] abordaram o aprendizado de movimentos ponto a ponto avaliando PI² através de experimentos na simulação de um membro robótico com 2 graus de liberdade. O algoritmo aplicado em conjunto com a estrutura DMP proposta mostrou permitir soluções simples e rápidas para problemas complexos de geração de trajetórias.

O algoritmo *Hierarchical Reinforcement Learning*, estudado e implementado por [9] foi responsável e capaz de realizar, dentro do contexto de aprendizado de multitarefas, o aprendizado da decomposição de tarefas em 4 níveis de hierarquia e por explorar a própria decomposição com a finalidade de combinar a complexidade das sequências de ações primitivas a cada tarefa do braço robótico utilizado.

Por outro prisma, em [6] foi abordado o uso de *Policy Learning by Weighting Exploration with the Returns (PoWER)* aplicado à tarefa *Peg-in-Hole* utilizando um braço robótico. Ao ser implementado junto ao *Imitation Learning*, o algoritmo proposto (PoWER), utilizado no processo de AR, melhorou os caminhos das habilidades motoras iniciais do robô e os otimizaram a fim de reduzir o número de etapas de tempo de

execução.

C. Ajuste de Híperparâmetros

Os hiperparâmetros estão presentes em todo sistema de aprendizagem de máquina e o ajuste dos mesmos tem o potencial de aprimorar o desempenho do sistema desenvolvido [13,30]. No entanto, percebe-se na literatura que poucos trabalhos realizam o ajuste de hiperparâmetros e utilizam dos benefícios da otimização do mesmo para aperfeiçoar os algoritmos abordados. Observando a tabela 4, nota-se que somente 4 dos 38 trabalhos na revisão, ou seja, apenas 10,81% fazem o ajuste de hiperparâmetros.

Tabela IV
TRABALHOS QUE REALIZARAM OU NÃO AJUSTE DE HIPERPARÂMETROS

Realizou ajuste	Trabalhos
Sim	[36,10,17,21].
Não	[3,49,18,52,33,41,14,34,1,2,32,46,16], [18,27,45,47,9,43,6,44,12,31,38,19,20], [42,39,48].

Em [36] foi escolhida uma taxa de aprendizado inicial $\alpha = 0,001$ adotada à função α_d (t) = $\alpha\frac{t}{T}$, sendo t referente ao episódio atual e T o número total de episódios. Finalmente foi demonstrado, que as taxas de aprendizado decrescentes superaram as taxas de aprendizado constantes. [10] realizou 192 permutações de seleções de hiperparâmetros para o algoritimo ACKTR e 128 para o algorítimo PPO2, nas quais foram realizadas dez sessões de treinamento em cada uma delas. Para o ACKTR, foi utilizado $2,5x10^5$ interavalos de tempo (passos dados no ambiente), enquanto para o PPO2, $1x10^5$ intervalos de tempo. Concluiu-se que o sistema de compensação de gravidade proposto, é submetido a um menor arrependimento acumulativo do que o sistema não compensado entre a região do espaço de hiperparâmetros explorado. Sob outra perspectiva, em [17] foi demonstrado que o fator de desconto adaptativo proposto resultou em um bom desempenho para vários ambientes de forma consistente, mas, dependendo do ambiente, um fator de desconto um pouco menor pode gerar um desempenho melhor do que um desconto maior. Ademais, o trabalho [21] utilizou a estrutura de otimização de hiperparâmetro OPTUNA para realizar o ajuste. Os demais 89,19% dos trabalhos, não realizam o ajuste de hiperparâmetros, utilizam valores fixos próprios ou baseados em resultados de trabalhos anteriores.

D. Comparação com outros estudos

Ao decorrer desta seção são comparados trabalhos de revisão sistemática relacionada a aplicações de aprendizado por reforço em manipuladores robóticos da literatura com o presente trabalho. Na tabela 5, os trabalhos são identificados como I (este trabalho), II [11], III [23], IV [28], V [50] e VI [26].

Na Tabela 5, nota-se que todos os artigos de comparados com esta revisão apresentam os resultados da aplicação das técnicas de AR no manipulador robótico, mas nenhum deles apresentaram indicações de trabalhos que fizeram ajuste de hiperparâmetros e os resultados da aplicação do mesmo, apesar de que os trabalhos [11] e [28] brevemente mencionam apenas um trabalho nesse sentido. [26] enfatiza aplicações de algoritmos baseados em DRL para a tarefa de preensão de objetos realizada por manipuladores robóticos, porém menciona alguns trabalhos que utilizam outros algoritmos junto ao DRL. O mesmo também apresenta links dos repositórios dos projetos de trabalhos analisados. Por outro lado, é notável que apenas [50] também aborda e cita os manipuladores e simuladores utilizados nos experimentos, apesar do mesmo focar apenas em mãos robóticas. Além disso, apenas [23] não apresenta indicações de trabalhos futuros da área e não disserta acerca de uma diversidade de técnicas de AR, abordando apenas técnicas baseadas em DRL. Já [28] discute possíveis direções de pesquisa futuras baseadas nos resultados da própria revisão e os demais apresentam as principais indicações para trabalhos futuros na área, além de abordarem diversas técnicas de AR.

Tabela V
COMPARAÇÃO COM REVISÕES SISTEMÁTICAS PRESENTES NA LITERATURA [11,23,28,50,26]

Abordagens	I	II	III	IV	V	VI
Resultados da aplicação das técnicas de AR	✓	✓	✓	✓	✓	✓
Ajuste de Hiperparâmetros	✓					
Simuladores utilizados	✓				✓	
Manipuladores utilizados	✓				✓	
Indicações de trabalhos futuros	✓	✓		✓	✓	✓
Espectro amplo de técnicas analisadas	✓	✓		✓	✓	

Portanto, nota-se que o presente trabalho se diferencia dos demais ao apresentar uma abordagem sobre a realização do ajuste de hiperparâmetros bem como sobre os resultados alcançados a partir de tal realização. Também é notório que 2 dos 6 trabalhos de revisão comparados focam no estudo de apenas um tipo de técnica de AR, enquanto apenas 1 deles não apresenta indicações de trabalhos futuros. Fatores estes que caracterizam as principais diferenças entre os trabalhos de revisão sistemática presentes na literatura e este trabalho.

E. Discussões

Nesta subseção são discutidas as perguntas de pesquisa utilizadas para guiar a revisão sistemática.

(1) Quais manipuladores são adotados nos artigos?

O manipulador UR3 (7,89%) foi o mais utilizado. Juntos, IRB 1600, UR5, RM-X52 e PANDA somam 26,68%, sendo 6,67% o percentual de uso de cada um. Por outro lado, os modelos YuMi, BCN3D moveo, Mitsubishi RV-12SDL-12S, NAO, KUKA, Sawyer, Openmanipulator, SCARA e robotiq 2f-85 gripper são utilizados em 2,63% dos trabalhos cada. Além disso, 7,89% dos trabalhos utilizaram manipuladores produzidos em laboratório, 10,53% não informaram e 28,95% dos trabalhos não utilizaram manipuladores reais.

(2) Quais técnicas de AR são aplicadas?

As técnicas mais utilizadas foram *Q-Learning* (23,68%), *Deep Reinforcement Learning* (28,95%), *Actor-Critic* (26,32%) e *Policy Gradient* (21,05%). *Policy Improvement with Path Integrals* foi abordado em 5,26% dos trabalhos analisados, enquanto as demais técnicas estão presentes em 2,63% dos trabalhos cada.

(3) O trabalho utiliza simuladores? Quais?

Os mais utilizados foram: MATLAB (18,42%), MuJoco (15,79%), Gazebo (13,16%) e CoppeliaSim (10,53%). Ademais, o RoboDK é utilzado por 5,26% dos trabalhos. Equanto ROS-Indigo, Pybullet, RobotStudio e Tensor Flow/Keras somam 10,52%. Por fim, dos trabalhos 15,79% não informaram e 13,16% não utilizaram simuladores.

(4) O trabalho realiza ajuste de hiperparâmetros de AR?

Como apresentado no trabalho, apenas 10,53% dos trabalhos estudados nesta revisão realizam o ajuste de hiperparâmetros, sendo o fator de desconto e taxa de aprendizado os mais abordados.

(5) Quais os principais resultados da aplicação do AR no manipulador?

Se destacam os seguintes resultados:

- Planejamentos de movimento precisos e estáveis
- Planejamento de caminhos mais curtos e viáveis
- Melhoria na velocidade do aprendizado de tarefas
- Qualidade no controle de manipuladores robóticos

(6) Quais as tendências para a área? Observando o que os artigos mais recentes indicam como trabalhos futuros.

Entre as indicações apresentadas nos trabalhos estudados, vale ressaltar as seguintes direções de pesquisas futuras:

- Aplicar os algoritmos propostos em manipuladores robóticos para resolução de tarefas diferentes
- Combinar outros métodos presentes na literatura com os implementados nos trabalhos da presente revisão
- Aprimorar as abordagens propostas
- Ampliar a complexidade de tarefas e incertezas do ambiente para realizar o estudo utilizando os métodos propostos.

4. Conclusão

O objetivo deste trabalho foi realizar uma revisão sistemática acerca das aplicações de AR em manipuladores robóticos. Para isso, foram estudados 38 trabalhos publicados entre 2013 e 2023 disponíveis na base de dados Scopus. Realizando um comparativo entre revisões sistemáticas da área, as quais já se encontravam na literatura, e a presente revisão, foi possível notar a falta de abordagens referentes ao ajuste de hiperparâmetros, bem como da apresentação dos simuladores e manipuladores robóticos utilizados nos artigos analisados. Estes fatores diferenciam a maioria das demais revisões da presente revisão sistemática, a qual apresenta e discute acerca destes temas.

Através das análises feitas, foi avaliado que os algoritmos mais utilizados entre os trabalhos analisados foram *Q-Learning, Policy-based, Deep Reinforcement Learning, Actor-Critic* e demais algoritmos derivados. Somados, estão presentes 74,92% dos artigos. Alguns manipuladores e simuladores também obtiveram destaque, estando presentes em 2 ou mais trabalhos. Um exemplo foi o manipulador robótico UR3, presente em 7,89% dos trabalhos e o ambiente de simulação MATLAB, presente em 18,42% dos trabalhos. Os resultados demonstraram que, entre os trabalhos analisados na revisão, apenas 10,81% realizam o ajuste de hiperparâmetros, o que revela carência na literatura relativa à realização do ajuste para fins de aprimoramento dos algoritmos abordados.

Em trabalhos futuros, sugere-se a implementação de diferentes tarefas de manipulação explorando as abordagens e algoritmos já presentes na literatura, bem como aprimorar os algoritmos de AR já existentes aplicando-os em manipuladores robóticos. Outra possibilidade envolve acrescentar desafios, por exemplo, o aumento da complexidade das tarefas e incertezas externas no problema de controle e manipulação a ser solucionado utilizando algum entre os métodos propostos. Além disso, é relevante implementar os métodos de AR presentes nessa revisão em conjunto com demais métodos de aprendizagem. Por fim, é importante que hajam mais trabalhos na área a realizarem o ajuste de hiperparâmetros de AR, buscando otimizar os resultados já obtidos no estado-da-arte.

Agradecimentos

Agradecemos à UFRB.

Referências

[1] Abdi, Ali, Dibash Adhikari, and Ju Hong Park. 2021. "A Novel Hybrid Path Planning Method Based on q-Learning and Neural Network for Robot Arm." Applied Sciences 11 (15): 6770.

[2] Abdi, Ali, Mohammad Hassan Ranjbar, and Ju Hong Park. 2022. "Computer Vision-Based Path Planning for Robot Arms in Three-Dimensional Workspaces Using q-Learning and Neural Networks." Sensors 22 (5): 1697.

[3] Basa, Daniel, and Axel Schneider. 2015. "Learning Point-to-Point Movements on an Elastic Limb Using Dynamic Movement Primitives." Robotics and Autonomous Systems 66: 55–63.

[4] Beltran-Hernandez, Cristian C, Damien Petit, Ixchel G Ramirez-Alpizar, and Kensuke Harada. 2020. "Variable Compliance Control for Robotic Peg-in-Hole Assembly: A Deep-Reinforcement-Learning Approach." Applied Sciences 10 (19): 6923.

[5] Bhatnagar, Shalabh, Richard S. Sutton, Mohammad Ghavamzadeh, and Mark Lee. 2009. "Natural Actor–Critic Algorithms." Automatica 45 (11): 2471–82.

[6] Cho, Nam Jun, Sang Hyoung Lee, Jong Bok Kim, and Il Hong Suh. 2020. "Learning, Improving, and Generalizing Motor Skills for the Peg-in-Hole Tasks Based on Imitation Learning and Self-Learning." Applied Sciences 10 (8): 2719.

[7] Clifton, Jesse, and Eric Laber. 2020. "Q-Learning: Theory and Applications." Annual Review of Statistics and Its Application 7 (1): 279–301.

[8] Damjanović, Ivana, Ivica Pavić, Mate Puljiz, and Mario Brcic. 2022. "Deep Reinforcement Learning-Based Approach for Autonomous Power Flow Control Using Only Topology Changes." Energies 15 (19): 6920.

[9] Duminy, Nicolas, Sao Mai Nguyen, Junshuai Zhu, Dominique Duhaut, and Jerome Kerdreux. 2021. "Intrinsically Motivated Open-Ended Multi-Task Learning Using Transfer Learning to Discover Task Hierarchy." Applied Sciences 11 (3): 975.

[10] Fugal, Jonathan, Jihye Bae, and Hasan A Poonawala. 2021. "On the Impact of Gravity Compensation on Reinforcement Learning in Goal-Reaching Tasks for Robotic Manipulators." Robotics 10 (1): 46.

[11] Han, Dong, Beni Mulyana, Vladimir Stankovic, and Samuel Cheng. 2023. "A Survey on Deep Reinforcement Learning Algorithms for Robotic Manipulation." Sensors 23 (7): 3762.

[12] Harnack, Daniel, Julie Pivin-Bachler, and Nicolás Navarro-Guerrero. 2022. "Quantifying the Effect of Feedback Frequency in Interactive Reinforcement Learning for Robotic Tasks." Neural Computing and Applications, 1–13.

[13] Hutter, Frank, Lars Kotthoff, and Joaquin Vanschoren. 2019. Automated Machine Learning: Methods, Systems, Challenges. Springer Nature.

[14] Jing, Wei, Chun Fan Goh, Mabaran Rajaraman, Fei Gao, Sooho Park, Yong Liu, and Kenji Shimada. 2018. "A Computational Framework for Automatic Online Path Generation of Robotic Inspection Tasks via Coverage Planning and Reinforcement Learning." IEEE Access 6: 54854–64.

[15] Kegyes, Tamás, Zoltán Süle, and János Abonyi. 2021. "The Applicability of Reinforcement Learning Methods in the Development of Industry 4.0 Applications." Complexity 2021: 1–31.

[16] Kim, MyeongSeop, Dong-Ki Han, Jae-Han Park, and Jung-Su Kim. 2020. "Motion Planning of Robot Manipulators for a Smoother Path Using a Twin Delayed Deep Deterministic Policy Gradient with Hindsight Experience Replay." Applied Sciences 10 (2): 575.

[17] Kim, MyeongSeop, Jung-Su Kim, Myoung-Su Choi, and Jae-Han Park. 2022. "Adaptive Discount Factor for Deep Reinforcement Learning in Continuing Tasks with Uncertainty." Sensors 22 (19): 7266.

[18] Kumar, Abhishek, and Rajneesh Sharma. 2018. "Linguistic Lyapunov Reinforcement Learning Control for Robotic Manipulators." Neurocomputing 272: 84–95.

[19] Li, Ang, Zhenze Liu, Wenrui Wang, Mingchao Zhu, Yanhui Li, Qi Huo, and Ming Dai. 2021. "Reinforcement Learning with Dynamic Movement Primitives for Obstacle Avoidance." Applied Sciences 11 (23).

[20] Li, Yanan, Long Chen, Keng Peng Tee, and Qingquan Li. 2015. "Reinforcement Learning Control for Coordinated Manipulation of Multi-Robots." Neurocomputing 170: 168–75.

[21] Lindner, Tymoteusz, and Andrzej Milecki. 2022. "Reinforcement Learning-Based Algorithm to Avoid Obstacles by the Anthropomorphic Robotic Arm." Applied Sciences 12 (13): 6629.

[22] Liu, Dong, Zitu Wang, Binpeng Lu, Ming Cong, Honghua Yu, and Qiang Zou. 2020. "A Reinforcement Learning-Based Framework for Robot Manipulation Skill Acquisition." IEEE Access 8: 108429–37.

[23] Liu, Luyu, Qianyuan Liu, Yong Song, Bao Pang, Xianfeng Yuan, and Qingyang Xu. 2021. "A Collaborative Control Method of Dual-Arm Robots Based on Deep Reinforcement Learning." Applied Sciences 11 (4): 1816.

[24] Millan-Arias, Cristian C, Bruno JT Fernandes, Francisco Cruz, Richard Dazeley, and Sergio Fernandes. 2021. "A Robust Approach for Continuous Interactive Actor-Critic Algorithms." IEEE Access 9: 104242–60.

[25] Mnih, Volodymyr, Adria Puigdomenech Badia, Mehdi Mirza, Alex Graves, Timothy Lillicrap, Tim Harley, David Silver, and Koray Kavukcuoglu. 2016. "Asynchronous Methods for Deep Reinforcement Learning." In International Conference on Machine Learning, 1928–37. PMLR.

[26] Mohammed, Marwan Qaid, Kwek Lee Chung, and Chua Shing Chyi. 2020. "Review of Deep Reinforcement Learning-Based Object Grasping: Techniques, Open Challenges, and Recommendations." IEEE Access 8: 178450–81.

[27] Moreira, Ithan, Javier Rivas, Francisco Cruz, Richard Dazeley, Angel Ayala, and Bruno Fernandes. 2020. "Deep Reinforcement Learning with Interactive Feedback in a Human–Robot Environment." Applied Sciences 10 (16): 5574.

[28] Nguyen, Hai, and Hung La. 2019. "Review of Deep Reinforcement Learning for Robot Manipulation." In 2019 Third IEEE International Conference on Robotic Computing (IRC), 590–95.

[29] Ottoni, André LC, Erivelton G Nepomuceno, Marcos S de Oliveira, and Daniela CR de Oliveira. 2022. "Reinforcement Learning for the Traveling Salesman Problem with Refueling." Complex Intelligent Systems 8 (3): 2001–15.

[30] Ottoni, André Luiz Carvalho, Erivelton Geraldo Nepomuceno, and Marcos Santos de Oliveira. 2017. "Análise Da Definição de Parâmetros Do Aprendizado Por Reforço No Desempenho de Um Manipulador Robótico." ForScience 5 (3).

[31] Pane, Yudha P., Subramanya P. Nageshrao, Jens Kober, and Robert Babuška. 2019. "Reinforcement Learning Based Compensation Methods for Robot Manipulators." Engineering Applications of Artificial Intelligence 78: 236–47.

[32] Park, Kwan-Woo, MyeongSeop Kim, Jung-Su Kim, and Jae-Han Park. 2022. "Path Planning for Multi-Arm Manipulators Using Soft Actor-Critic Algorithm with Position Prediction of Moving Obstacles via LSTM." Applied Sciences 12 (19): 9837.

[33] Prianto, Evan, Jae-Han Park, Ji-Hun Bae, and Jung-Su Kim. 2021. "Deep Reinforcement Learning-Based Path Planning for Multi-Arm Manipulators with Periodically Moving Obstacles." Applied Sciences 11 (6): 2587.

[34] Ramirez, Jorge, and Wen Yu. 2023. "Reinforcement Learning from Expert Demonstrations with Application to Redundant Robot Control." Engineering Applications of Artificial Intelligence 119: 105753.

[35] Rivera, Patricio, Edwin Valarezo Añazco, and Tae-Seong Kim. 2021. "Object Manipulation with an Anthropomorphic Robotic Hand via Deep Reinforcement Learning with a Synergy Space of Natural Hand Poses." Sensors 21 (16): 5301.

[36] Robbins, AS, M Ho, and M Teodorescu. 2022. "Model-Free Dynamic Control of Robotic Joints with Integrated Elastic Ligaments." Robotics and Autonomous Systems 155: 104150.

[37] Sasaki, Minoru, Joseph Muguro, Fumiya Kitano, Waweru Njeri, and Kojiro Matsushita. 2022. "Sim–Real Mapping of an Image-Based Robot Arm Controller Using Deep Reinforcement Learning." Applied Sciences 12 (20): 10277.

[38] Sekkat, Hiba, Smail Tigani, Rachid Saadane, and Abdellah Chehri. 2021. "Vision-Based Robotic Arm Control Algorithm Using Deep Reinforcement Learning for Autonomous Objects Grasping." Applied Sciences 11 (17): 7917.

[39] Sharma, Rajneesh. 2016. "Lyapunov Theory Based Stable Markov Game Fuzzy Control for Non-Linear Systems." Engineering Applications of Artificial Intelligence 55: 119–27.

[40] Sutton, Richard S, and Andrew G Barto. 2018. Reinforcement Learning: An Introduction. MIT press.

[41] Szczepanski, Rafal, Krystian Erwinski, Mateusz Tejer, Artur Bereit, and Tomasz Tarczewski. 2022. "Optimal Scheduling for Palletizing Task Using Robotic Arm and Artificial Bee Colony Algorithm." Engineering Applications of Artificial Intelligence 113: 104976.

[42] Tang, Li, Yan-Jun Liu, and Shaocheng Tong. 2014. "Adaptive Neural Control Using Reinforcement Learning for a Class of Robot Manipulator." Neural Computing and Applications 25: 135–41.

[43] Tsurumine, Yoshihisa, Yunduan Cui, Eiji Uchibe, and Takamitsu Matsubara. 2019. "Deep Reinforcement Learning with Smooth Policy Update: Application to Robotic Cloth Manipulation." Robotics and Autonomous Systems 112: 72–83.

[44] Verleysen, Andreas, Thomas Holvoet, Remko Proesmans, Cedric Den Haese, and Francis Wyffels. 2020. "Simpler Learning of Robotic Manipulation of Clothing by Utilizing Diy Smart Textile Technology." Applied Sciences 10 (12): 4088.

[45] Wang, Yang, Liming Wang, and Yonghui Zhao. 2022. "Research on Door Opening Operation of Mobile Robotic Arm Based on Reinforcement Learning." Applied Sciences 12 (10): 5204.

[46] Wong, Ching-Chang, Shao-Yu Chien, Hsuan-Ming Feng, and Hisasuki Aoyama. 2021. "Motion Planning for Dual-Arm Robot Based on Soft Actor-Critic." IEEE Access 9: 26871–85.

[47] Xie, Jiexin, Zhenzhou Shao, Yue Li, Yong Guan, and Jindong Tan. 2019. "Deep Reinforcement Learning with Optimized Reward Functions for Robotic Trajectory Planning." IEEE Access 7: 105669–79.

[48] Xie, Zhicheng, Tao Sun, Trevor Hocksun Kwan, Zhongcheng Mu, and Xiaofeng Wu. 2020. "A New Reinforcement Learning Based Adaptive Sliding Mode Control Scheme for Free-Floating Space Robotic Manipulator." IEEE Access 8: 127048–64.

[49] Xie, Zhihang, and Qiquan Lin. 2023. "Reinforcement Learning-Based Adaptive Position Control Scheme for Uncertain Robotic Manipulators with Constrained Angular Position and Angular Velocity." Applied Sciences 13 (3): 1275.

[50] Yu, Chunmiao, and Peng Wang. 2022. "Dexterous Manipulation for Multi-Fingered Robotic Hands with Reinforcement Learning: A Review." Frontiers in Neurorobotics 16.

[51] Lee, Seung Hyun, Young Jae Lee, and Dong Hwan Kim. 2023. "Inverse Kinematic Solution Extention for a Robot to Cope with Joint Angle Constraints." International Journal of Control, Automation and Systems, 1–11.

[52] Malik, Aryslan, Yevgeniy Lischuk, Troy Henderson, and Richard Prazenica. 2022. "A Deep Reinforcement-Learning Approach for Inverse Kinematics Solution of a High Degree of Freedom Robotic Manipulator." Robotics 11 (2): 44.