

## **ESTRATÉGIAS AVANÇADAS DE CONTROLE DE LINHA: UMA ABORDAGEM COM APRENDIZADO POR REFORÇO PROFUNDO**

**Rafael Gomes Paes de Lira**

Instituição: UFPE - Universidade Federal de Pernambuco  
Endereço: Av. Prof. Moraes Rego, 1235 - Cidade Universitária, Recife - PE, 50670-901  
Email: rafael.gplira@ufpe.br

**Dênis Leite**

Instituição: UPE - Universidade de Pernambuco  
Endereço: Av. Gov. Agamenon Magalhães - Santo Amaro, Recife - PE, 50100-010  
Email: denisleite@poli.br

**Lucas Matheus Nascimento**

Instituição: UFPE - Universidade Federal de Pernambuco  
Endereço: Av. Prof. Moraes Rego, 1235 - Cidade Universitária, Recife - PE, 50670-901  
Email: lucas.matheusn@ufpe.br

**Leandro Henrique Gomes da Silva**

Instituição: UFPE - Universidade Federal de Pernambuco  
Endereço: Av. Prof. Moraes Rego, 1235 - Cidade Universitária, Recife - PE, 50670-901  
Email: leandro.hgsilva@ufpe.br

**William da Silva Macedo Formágio**

Instituição: UPE - Universidade de Pernambuco  
Endereço: Av. Gov. Agamenon Magalhães - Santo Amaro, Recife - PE, 50100-010  
Email: formagio.william@gmail.com

**Márcio José das Chagas Moura**

Instituição: UFPE - Universidade Federal de Pernambuco  
Endereço: Av. Prof. Moraes Rego, 1235 - Cidade Universitária, Recife - PE, 50670-901  
Email: marcio.cmoura@ufpe.br

### **RESUMO**

Este estudo investiga a aplicação do Aprendizado por Reforço Profundo (DRL) na otimização do controle de produção em uma empresa multinacional, explorando a interseção entre inovação e tecnologia na Indústria 4.0. Ao substituir a lógica atual da linha de produção por um modelo de Aprendizado por Reforço (RL), foram observadas melhorias significativas, especialmente em cenários atípicos. O modelo DQL demonstrou uma redução na produção menor do que a lógica atual em simulações que envolviam o funcionamento com máquinas críticas desligadas. Esses resultados evidenciam a capacidade do DRL de lidar com desafios complexos e imprevisíveis, destacando-se como uma ferramenta valiosa para melhorar a eficiência e competitividade na indústria. Recomenda-se explorar mais algoritmos de aprendizado por reforço e configurações de redes neurais para avanços adicionais neste campo.

**PALAVRAS CHAVE.** Aprendizado por Reforço Profundo, Controle de Produção, Indústria 4.0, Eficiência Operacional, Tecnologia.

**Tópicos do artigo:** Inteligência Computacional e Simulação

## ABSTRACT

This study investigates the application of Deep Reinforcement Learning (DRL) in optimizing production control in a large multinational company, exploring the intersection of innovation and technology in Industry 4.0. By replacing the current logic of the production line with a Reinforcement Learning (RL) model, significant improvements were observed, especially in atypical scenarios. The DQL model demonstrated a reduction in production lower than the current logic in simulations involving critical machines being turned off. These results highlight the ability of DRL to handle complex and unpredictable challenges, emerging as a valuable tool for enhancing efficiency and competitiveness in the industry. Further exploration of reinforcement learning algorithms and neural network configurations is recommended for additional advancements in this field.

**KEYWORDS.** Deep Reinforcement Learning, Production Control, Industry 4.0, Operational Efficiency, Technology.

**Paper topics:** Computational Intelligence and Simulation

## 1. Introdução

Neste estudo, explora-se a intersecção crucial entre inovação, tecnologia e melhoria da produção em uma empresa multinacional de grande porte. O desafio enfrentado por esta empresa reside em sua constante busca pela melhoria da eficiência da produção, mesmo já operando em um patamar de excelência em suas fábricas. Nosso foco está em uma das unidades localizadas na América do Sul, onde examinamos as oportunidades e desafios relacionados à implementação de tecnologias da Indústria 4.0 para aprimorar processos, aumentar eficiência, reduzir custos e fomentar a inovação.

Atualmente, a sua linha de produção é automatizada e a modulação se baseia em regras empíricas, o que acaba resultando em múltiplas paradas das máquinas em cenários atípicos, prejudicando a produtividade e aumentando os custos de manutenção. Com a introdução de tecnologias avançadas, como o aprendizado por reforço profundo (Deep Reinforcement Learning – DRL), abre-se um novo horizonte para aprimorar a modulação da linha, permitindo uma resposta mais adaptável e eficiente às demandas variáveis da produção.

A abordagem do estudo utiliza o DRL para aprimorar o controle da linha de produção, especificamente pela máquina que atualmente é o gargalo da linha. O DRL é integrado ao modelo de simulação já utilizado pela empresa, criando uma representação virtual da planta. Isso permite que o modelo de DRL tome decisões sobre a velocidade da máquina na linha de produção, com o objetivo de minimizar paradas não planejadas e aumentar a produção global. Esta integração permite que o modelo treine o suficiente até alcançar um bom desempenho, sem a necessidade de fazer qualquer alteração na linha de produção ou diminuir a produção temporariamente.

O uso do Aprendizado por Reforço (Reinforcement Learning - RL) tem se expandido cada vez mais na indústria, principalmente na otimização de processos de produção. Uma aplicação promissora é o emprego do DRL para aprimorar a gestão de ordens de produção em ambientes complexos, como os job-shops, considerando restrições temporais. Estudos indicam que o algoritmo Deep Q-learning (DQL) possui um alto potencial para otimizar o despacho de ordens, superando benchmarks estabelecidos pela indústria [Altenmüller et al. 2020].

Para lidar com os desafios de complexidade e incerteza nos sistemas de manufatura, um modelo de Deep Q-Networks (DQN) de duas hierarquias pode ser eficiente quando comparado com modelos clássicos, apesar de não considerar certos fatores que podem afetar seu desempenho [Luo et al. 2021]. O potencial do DRL também se estende à tomada de decisão para veículos guiados automaticamente em indústrias automotivas, demonstrando sua aplicabilidade em diversas áreas da manufatura [Feldkamp et al. 2020].

Novas abordagens inovadoras estão sendo exploradas para aprimorar a flexibilidade e eficiência das linhas de montagem na indústria. A integração de algoritmos de aprendizado por reforço e modelos de gêmeos digitais está possibilitando a automação do planejamento e monitoramento das linhas de produção em tempo real, obtendo resultados promissores em experimentos práticos [Li et al. 2022].

No contexto do agendamento de tarefas em linhas de produção, o uso do DRL com um algoritmo DQL tem sido explorado para garantir o cumprimento de prazos e maximizar a produção [Marchesano et al. 2022]. Outra linha de pesquisa se concentra na substituição preventiva de máquinas em linhas seriadas, onde o Q-learning é aplicado para desenvolver políticas ótimas com base no envelhecimento das máquinas e em dados históricos [Huang et al. 2019]. Além disso, políticas de controle em tempo real têm sido propostas para sistemas que envolvem duas máquinas e um buffer, com o objetivo de reduzir os custos de produção e o trabalho em processo (WIP) [Zheng et al. 2017].

Esses estudos contribuem significativamente para a aplicação do RL na indústria, promovendo automação inteligente e eficiência. Ao integrar tecnologias avançadas com simulação industrial, eles demonstram um potencial notável para impulsionar a inovação e resolver desafios complexos na manufatura, alinhados aos princípios da Indústria 4.0.

Assim, este estudo se destaca como uma solução pioneira que combina inovação tecnológica com processos industriais tradicionais, mostrando um caminho promissor para aprimorar a eficiência e competitividade em um ambiente de produção desafiador.

## 2. Metodologia

As regras de controle de linha são cruciais para melhorar o fluxo produtivo, controlando a velocidade das máquinas e reduzindo paradas não planejadas. O sistema atual é composto por uma série de máquinas interconectadas por esteiras, que são baseadas em regras empíricas e velocidades fixas. Este sistema enfrenta dificuldades em adaptar-se a diferentes cenários atípicos. O desafio é desenvolver um modelo de controle mais flexível e adaptativo, para isso o presente estudo desenvolveu técnicas de inteligência artificial, como o DRL, para melhorar a eficiência e a eficácia da linha de produção em tempo real.

### 2.1 Controle de linha de Produção

O controle da linha se concentra em otimizar o fluxo de produção, especialmente ajustando a velocidade das máquinas. O objetivo é minimizar as paradas não programadas das máquinas, conhecidas como auto-restarts, as quais ocorrem devido à ineficiência no controle da linha. Isso será alcançado ao melhorar a configuração de velocidade e ao evitar a subutilização de recursos, visando maximizar a eficiência da linha de produção.

As máquinas desempenham diferentes etapas no processo de fabricação, enquanto as esteiras são fundamentais no transporte, no buffer e no controle de linha de produção, utilizando sensores para detectar a presença ou ausência de produtos em áreas específicas.

As paradas não planejadas das máquinas, representam um custo adicional para a empresa devido à interrupção do processo de produção e ao tempo necessário, após o religamento, para que algumas máquinas atinjam a temperatura ideal, bem como a velocidade que cresce de forma gradativa antes do funcionamento.

O controle atual da linha de produção é baseado em um conjunto limitado de regras empíricas, o que torna difícil lidar com todos os cenários possíveis de operação. Por exemplo, quando uma máquina à frente, na linha, apresenta uma falha, isso resulta no acúmulo de produtos na esteira. Como consequência, um sensor é acionado e a máquina atrás entra em auto-restart até que a máquina da frente volte a funcionar e libere espaço na esteira. Além disso, as máquinas operam em velocidades fixas (mínima, follow ou máxima), o que não é suficiente para lidar com os diferentes cenários do dia a dia de produção, levando à subutilização de recursos e à ocorrência de auto-restarts.

O desafio é desenvolver um modelo de controle mais flexível e adaptativo que possa ajustar dinamicamente a velocidade das máquinas com base nas condições em tempo real da linha de produção. Nesse contexto, o uso de técnicas de inteligência artificial, como o DRL, oferece uma oportunidade para desenvolver um sistema de controle inteligente que aprenda com a experiência e melhore continuamente o desempenho da linha.

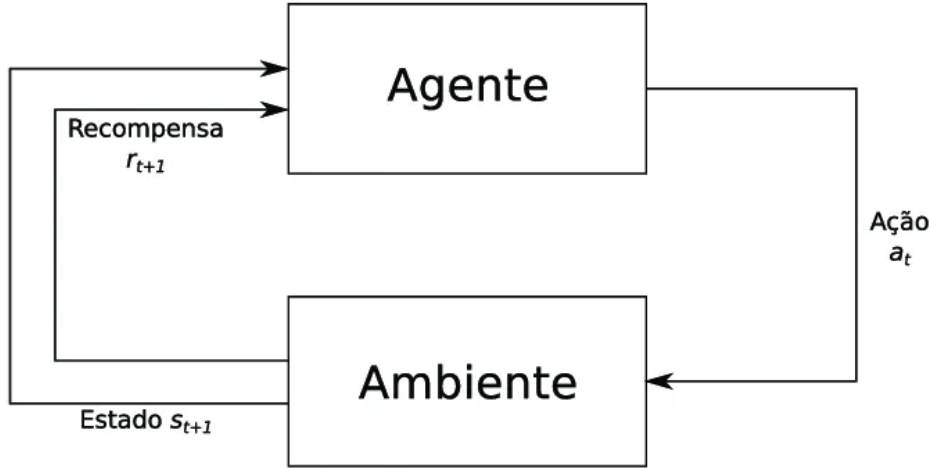
### 2.2 Modelo de Reinforcement Learning

Diante desse contexto, surge a seguinte questão: considerando a dificuldade de prever todos os cenários diários e de modelar correções para cada um deles, por que não desenvolver um modelo de aprendizado de máquina capaz de entender esses cenários, lidar com situações atípicas e formular sua própria lógica de controle de linha? Esta reflexão leva a uma segunda pergunta: qual modelo seria mais adequado para enfrentar esse tipo de desafio?

Segundo Agostinelli et al. [2018], o objetivo do RL é calcular uma estratégia de comportamento, ou política, que visa maximizar um critério de satisfação representado pela soma das recompensas a longo prazo, por meio de interações de tentativa e erro com um ambiente específico. O problema do RL envolve um agente, chamado de tomador de decisões, que opera em um ambiente composto por estados. O agente pode executar ações específicas, dependendo do estado atual. Após selecionar uma ação, o agente recebe uma recompensa escalar e, em seguida, se

encontra em um novo estado, determinado pelo estado atual e pela ação escolhida, conforme mostrado na Figura 1.

Figura 1 – Aprendizado por Reforço



Fonte: <https://medium.com/ranbookm/aprendizado-por-refor%C3%A7o-40d8800bd507>, acessado em 18/05/2024

O método de Q-learning é amplamente reconhecido como uma abordagem eficaz para aprendizado por reforço off-policy, ou livre de modelo [Watkins e Dayan 1992]. Esse algoritmo aprende gradualmente a função de política ótima interagindo com o ambiente, sem pressupor conhecimento prévio das recompensas ou probabilidades de transição. Em essência, o Q-learning busca aprender uma função  $Q$  que associa o estado atual  $s$  a uma ação  $a$  e ao valor de utilidade  $Q(s,a)$ . Este valor prediz a recompensa total futura descontada esperada ao tomar a ação atual  $a$  e seguir a política ótima  $\pi^*$  em todas as ações subsequentes. Segundo Watkins e Dayan [1992], a política ótima  $\pi^*$  é determinada pela função  $Q$ , sendo:

$$\pi^*(s) = \operatorname{argmax}_a Q(s, a) \quad (1)$$

O objetivo do Q-learning é aprender diretamente a função  $Q$ , sem a necessidade de conhecimento explícito das probabilidades de transição ou dos valores esperados das recompensas. Quando o ambiente está no estado  $s$ , o agente escolhe uma ação  $a$  para maximizar  $Q(s,a)$ , levando a uma transição para o estado  $s'$  no ambiente. A recompensa imediata da ação  $a$  a partir do estado  $s$ , seguindo a política ótima, é representada por  $Q(s,a)$ , acrescida da utilidade máxima possível do próximo estado  $s'$ , descontada pelo fator  $\gamma$ . Temos então, a seguinte equação:

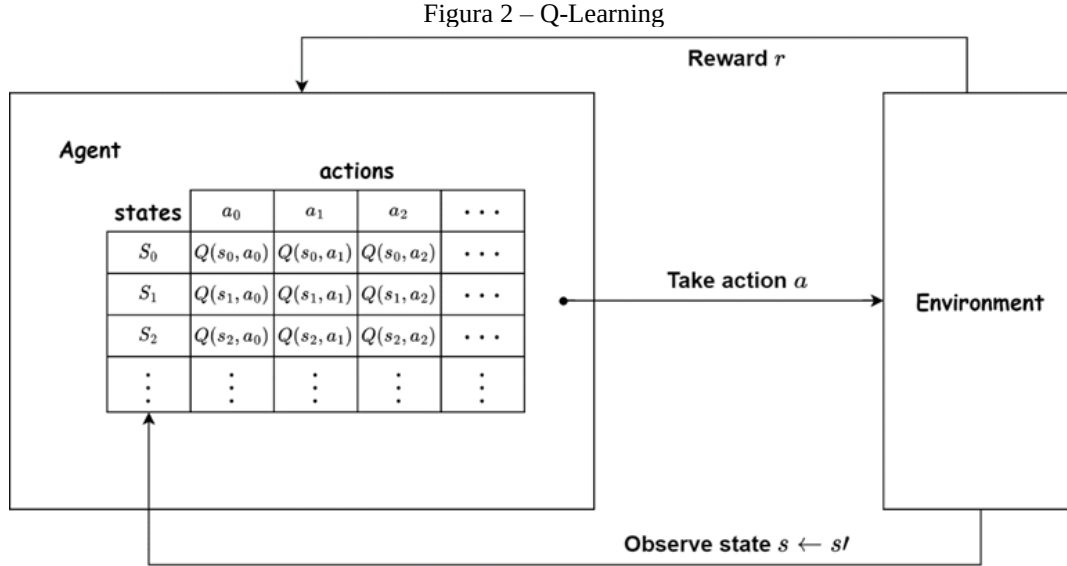
$$Q(s, a) = R(s, a) + \gamma * \sum_{s'} P_{s,s'}(a) * \max_b Q(s', b) \quad (2)$$

Os métodos de diferença temporal (TD), conforme descritos por Sutton e Barto [2018], são utilizados para construir incrementalmente a função  $Q$  por meio de estimativas. A atualização TD ajusta a estimativa  $Q(s,a)$  para reduzir a desigualdade na predição das recompensas. A equação de atualização de diferença temporal, uma forma específica deste ajuste, é:

$$Q_{n+1}(s, a) = Q_n(s, a) + \eta * (R(s, a) + \gamma * \max_b Q_n(s', b) - Q_n(s, a)) \quad (3)$$

Aqui,  $\eta$  representa uma taxa de aprendizado que pode ser uma pequena constante ou um valor que tende a zero conforme o tempo avança. Analisando a Equação (3), observa-se que o Q-learning realiza uma atualização de um passo, utilizando a função  $Q$  estimada no estado do próximo passo de tempo. Estudos realizados por Watkins e Dayan [1992] demonstraram que esse

procedimento converge sob condições específicas, especialmente quando o estado  $s'$  depende apenas do estado  $s$  e da ação tomada no estado  $s$ , e quando  $Q(s,a)$  pode ser representado em forma de tabela, ou seja, quando  $s$  e  $a$  pertencem a conjuntos finitos, conforme ilustrado na Figura 2.



Fonte: <https://wikidocs.net/174536>, acessado em 18/05/2024

Uma característica interessante do Q-learning é a capacidade de atualizar a função  $Q$  de forma assíncrona, permitindo um alto grau de paralelismo em sua construção. Isso possibilita a utilização de diferentes processadores para cada estimativa  $(s,a)$ , sem a necessidade de coordenação entre eles.

Em comparação com abordagens baseadas em modelos, que buscam aprender as regras subjacentes e as dinâmicas do ambiente, as abordagens livres de modelo, como o Q-learning, determinam a melhor ação para cada estado diretamente. Isso é particularmente útil em sistemas complexos e dinâmicos, como os sistemas modulares de produção, onde o espaço de estados é vasto e a construção de um modelo interno pode ser imprecisa e computacionalmente inviável [Gosavi 2015]. O Q-learning, ao aprender os valores  $Q$  para cada par de estado e ação, facilita a derivação de uma política ótima sem a necessidade de modelagem explícita do ambiente.

O agente recebe uma recompensa de feedback  $r_{t+1}$  a cada etapa de tempo  $t+1$  até alcançar o estado terminal  $T$ . No entanto, essa recompensa imediata  $r_{t+1}$  não representa o ganho a longo prazo. Em vez disso, utilizamos um valor de retorno generalizado  $R_t$  no momento  $t$ , onde  $\gamma$  é um fator de desconto que varia de 0 a 1. Quando  $\gamma$  se aproxima de 1, o agente se torna mais focado no longo prazo, enquanto que, quando  $\gamma$  se aproxima de 0, o agente se concentra mais no curto prazo [Sutton e Barto 2018], como pode ser visualizado na equação (4).

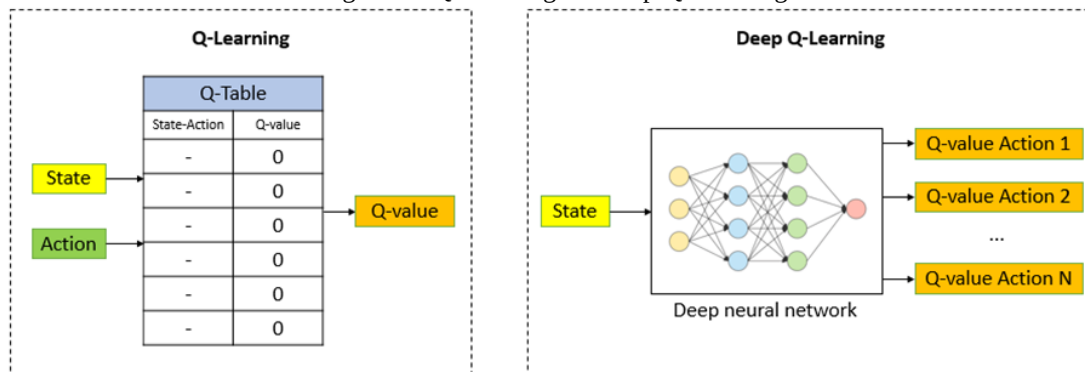
$$R_t = r_{t+1} + \gamma * r_{t+2} + \gamma^2 * r_{t+3} + \dots + \gamma^{T-t-1} * r_t = \sum_{i=0}^{T-t-1} \gamma^i * r_{t+i+1} \quad (4)$$

Uma extensão apropriada do Q-Learning tradicional é o Deep Q-Learning (DQL), que utiliza uma rede neural convolucional profunda para mapear a relação entre estados e ações, como exemplificado na Figura 3. Essa abordagem permite aplicar o princípio do Q-Learning mesmo em espaços de estado e ação substancialmente grandes [Mnih et al. 2015] [Shrestha e Mahmood 2019].

Segundo Alabadi e Albayrak [2020], controlar a escalabilidade em termos de número de estados e ações torna-se desafiador ao utilizar o Q-learning tradicional. Por isso, pesquisadores têm adotado o DQL, que emprega redes neurais para aproximar a função de valor  $Q$ . Devido à

complexidade do problema abordado, a utilização do Q-learning convencional seria praticamente inviável. À medida que a complexidade do problema aumenta, a Q-table cresce exponencialmente, resultando em tempos de execução impraticáveis.

Figura 3 - Q-Learning vs. Deep Q-Learning



Fonte: <https://www.baeldung.com/cs/q-learning-vs-deep-q-learning-vs-deep-q-network>, acessado em 17/05/2024

Por essa razão, será utilizado o DQL, que utiliza uma rede neural para calcular o valor Q para a próxima ação possível. Enquanto em problemas menores a Q-table pode ser vantajosa, devido à rapidez na construção de uma tabela pequena em comparação ao treinamento de uma rede neural, à medida que o problema se torna mais complexo, essa vantagem se inverte. O DQL se mostra mais adequado para lidar com problemas de maior complexidade, oferecendo uma abordagem mais eficaz e escalável.

### 2.3 Seleção de Parâmetros e Implementação do Modelo

O estado representa a situação atual do ambiente e do agente, o que implica a necessidade de determinar quais parâmetros serão considerados no treinamento do modelo, já que cada parâmetro reflete um aspecto do ambiente. Para este trabalho, foram utilizados dados obtidos por meio de uma integração entre o Plant Simulation e o Python, através da biblioteca Plantsim. O Plant Simulation, um software da Siemens, é amplamente utilizado para simulação e modelagem de processos produtivos, permitindo a análise detalhada de fluxos de material, utilização de recursos e otimização de sistemas complexos de manufatura. Sua capacidade de criar modelos dinâmicos e personalizáveis facilita a experimentação com diferentes cenários de produção e o ajuste fino de parâmetros.

A biblioteca Plantsim oferece ferramentas que permitem a leitura e modificação dos dados do Plant Simulation diretamente a partir de scripts Python, utilizando a interface COM (Component Object Model). Esta conexão permite a integração com algoritmos avançados, como o Deep Reinforcement Learning (DRL), que é aplicado para otimizar a tomada de decisão em sistemas produtivos. Através da comunicação pela porta COM, o Plantsim permite que o agente de DRL interaja com o ambiente simulado no Plant Simulation, ajustando parâmetros e estratégias de produção em tempo real, enquanto aprende com os resultados obtidos em cada iteração. Além da execução de funções, como iniciar e pausar a simulação, através de comandos no Python.

Embora existam outras abordagens possíveis para este problema, como o uso de imagens do Plant Simulation durante a execução da simulação, a decisão foi focar apenas nos dados coletados. Os parâmetros utilizados podem variar ao longo dos testes, mas serão detalhadamente especificados em cada experimento, incluindo a quantidade e quais parâmetros foram empregados.

A seleção dos parâmetros é um aspecto crucial nos testes, podendo impactar significativamente o desempenho do modelo. É importante ressaltar que, para quem está desenvolvendo o código, pode parecer óbvio incluir todos os dados disponíveis do Plant Simulation, pois está familiarizado com o ambiente. No entanto, é essencial adotar uma perspectiva

externa e imaginar-se em um ambiente completamente desconhecido, considerando o tipo de informação que cada parâmetro pode fornecer, como se estivéssemos em um ambiente completamente escuro.

Embora se acredite que mais informações melhorem o aprendizado do modelo, isso nem sempre ocorre. A inclusão de muitos parâmetros, como a velocidade das máquinas, seu estado de funcionamento e dados dos sensores, pode tornar o processo de associação entre o estado atual e as recompensas mais complexo. Em algumas situações, muitos parâmetros tornam o sistema menos eficiente na aprendizagem e tomada de decisões.

As ações disponíveis ao modelo são: 0, 1 e 2. O valor 0 indica nenhuma alteração na velocidade, 1 aumenta a velocidade em uma unidade, e 2 diminui. No entanto, essas ações devem respeitar os limites de velocidade da máquina. Se o limite máximo for atingido, não é possível aumentar a velocidade, e o mesmo vale para o limite mínimo e a redução.

No que se refere à função de recompensa, inicialmente será projetada para punir o modelo com um valor reduzido sempre que ocorrerem auto-restarts em alguma máquina, não se restringindo apenas à máquina que representa o gargalo da linha. A concepção inicial desta função de recompensa visa incentivar o modelo a evitar auto-restarts e a buscar constantemente o aumento da produção. Contudo, é importante ressaltar que nem sempre é tão simples assim, pois o modelo pode encontrar brechas não previstas, desafiando as expectativas e apresentando resultados surpreendentes.

O Algoritmo 1 a seguir descreve o processo de iteração utilizado no modelo de DRL em um ambiente simulado pelo Plant Simulation. A cada época, o modelo coleta todos os parâmetros relevantes do Plant Simulation. Durante a iteração, verifica-se se uma máquina entrou em auto-restart e ajusta a recompensa de acordo com seu estado. Este processo é repetido para apenas as máquinas próximas, pois não faz tanto sentido fornecer informações de todas as máquinas.

Entre o acontecimento de um evento com uma máquina muito à frente na linha e seu impacto na máquina analisada, há um tempo de várias épocas de treinamento, o que acabava mais confundindo do que ajudando o modelo. Em seguida, cria-se uma lista com todos os parâmetros coletados e a recompensa acumulada, que são fornecidos ao modelo. O modelo retorna uma ação, que pode ser não fazer nada, aumentar a velocidade em uma unidade, ou diminuir a velocidade. A recompensa é então zerada, concluindo a iteração de uma época. Este ciclo se repete até que todas as épocas estabelecidas sejam completadas.

---

**Algoritmo 1:** Algoritmo do Deep Reinforcement Learning

---

1. Para cada iteração dentro de um valor de épocas estabelecido:
  2.     Coleta todos os parâmetros do Plant Simulation
  3.     Para cada máquina em  $[X, \dots, 13 \text{ máquinas próximas}]$ :
  4.         Se máquina  $X$  entra em auto-restart:
  5.             recompensa = recompensa - 0.01
  6.         Caso contrário:
  7.             recompensa = recompensa + 0.02
  8.     Cria uma lista com todos os parâmetros
  9.     Fornece ao modelo a lista de parâmetros (estado atual) e a recompensa ( $r_{t+1}$ )
  10.     Modelo retorna o valor 0, 1 ou 2
  11.     Se o modelo retornou 0:
  12.         Não fazer nada
  13.     Se o modelo retornou 1:
  14.         Aumentar a velocidade em 1 unidade
  15.     Se o modelo retornou 2:
  16.         Diminuir a velocidade em 1 unidade
  17.     Zerar a recompensa
  18.     Finalizar 1 época
-

Portanto, o DRL, emerge como uma solução ideal para esse cenário. A complexidade do problema exige uma abordagem que possa aprender com as interações com o ambiente e, a partir delas, desenvolver uma estratégia melhor de controle de linha de produção. No contexto do DRL, o agente de aprendizado interage com o ambiente, recebendo recompensas ou penalidades com base em suas ações. Esse processo de retroalimentação contínua permite que o sistema aprenda e ajuste seu comportamento de forma dinâmica, sem a necessidade de um conjunto prévio de dados rotulados. Assim, o DRL apresenta-se como uma abordagem promissora para criar um sistema de controle de linha mais adaptável e eficiente, capaz de lidar com os desafios complexos e imprevisíveis encontrados no ambiente de produção industrial.

### 3. Algoritmo de treinamento e Experimentação

O algoritmo selecionado é o DQL, cuja rede neural é composta por 5 camadas. A primeira camada recebe os parâmetros como entrada, seguida por camadas intermediárias de 256, 128, 64, e 32 neurônios, respectivamente. A camada de saída retorna um vetor com 3 posições, conforme descrito anteriormente. O hiperparâmetro epsilon, que controla a taxa de exploração do modelo, foi ajustado para 0,95 para proporcionar um foco maior no longo prazo. Isso é importante porque, nas primeiras épocas, o ambiente simulado estará preenchendo as esteiras de produtos e as máquinas estarão iniciando seu funcionamento.

A linha de produção simulada contém 29 máquinas ao todo, organizadas em diferentes estágios. O processo inicia-se com uma única máquina, seguida por 11 máquinas semelhantes operando em paralelo. Após essa fase, há mais uma máquina, depois duas em paralelo, seguida por um grupo de 9 máquinas paralelas. Posteriormente, há mais uma máquina, seguida por outra, duas em paralelo, e uma última máquina, resultando em 9 estágios distintos ao longo da linha de produção. Essa configuração complexa permite simular cenários variados de gargalos e falhas em diferentes pontos da linha, crucial para a avaliação do desempenho do DRL frente à lógica atual de controle de produção.

Para os testes, utiliza-se o Plant Simulation, onde cada máquina possui um método acionado a cada 5 segundos para gerenciar auto-restarts e configurações de velocidade. Inicialmente, estabelece-se um baseline com 1000 simulações coletando dados de produção e auto-restarts das máquinas.

Foram realizados diversos testes considerando também cenários atípicos, variando recompensas e parâmetros do modelo. Inicialmente, foram utilizados 142 parâmetros, mas o melhor desempenho foi obtido com apenas 40 parâmetros. Observou-se que o modelo superou a lógica de controle de linha atual, em termos de produção, em 0,79%.

Apesar de já ter superado a referência inicial devido à experiência prévia na linha de produção, reconhecia-se que a lógica atual da linha de produção funcionava bem em situações normais, enquanto o RL apresentaria melhor desempenho em condições atípicas. Com isso, o modelo foi testado nessas condições.

Adiante da máquina controlada pelo DRL, há duas máquinas operando em paralelo que representam outro ponto crítico da linha de produção. Foi simulado um período de produção de dois dias utilizando apenas uma dessas máquinas. Nesse cenário, a lógica atual apresentou uma redução na produção de 37,10%, enquanto o DRL mostrou uma redução de apenas 35,55%, superando a lógica atual em 1,44%.

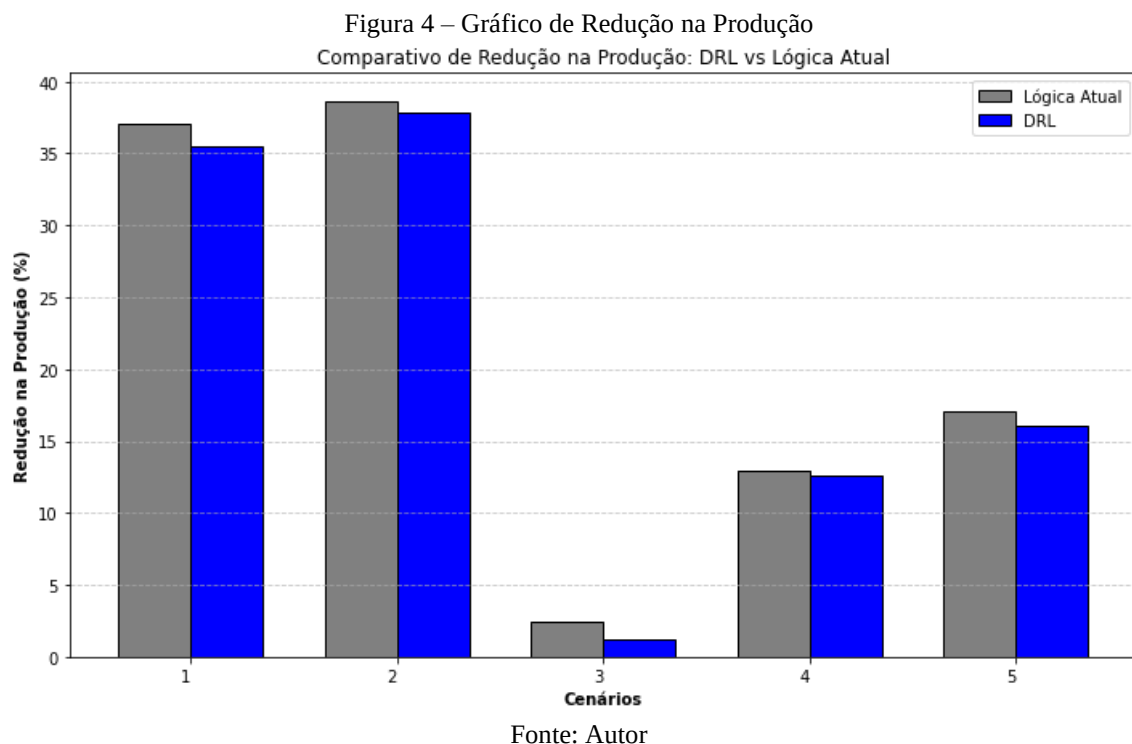
Foi realizado um teste adicional com a mesma estrutura, forçando a outra máquina a operar com uma velocidade superior e considerando uma redução de 15% na disponibilidade. Nesse cenário, a lógica atual teve uma perda de 38,66% na produção e o DRL teve uma perda de 37,80%, resultando em um ganho de 0,86% em relação à lógica atual.

Esses dois resultados forneceram um insight valioso: quando se tem duas máquinas operando em paralelo e uma delas falha, aumentar a velocidade da máquina remanescente para compensar a perda de produção pode não ser vantajoso, dependendo da redução na disponibilidade da máquina resultante desse aumento de velocidade.

O gargalo da linha de produção está na seção que possui 11 máquinas operando em paralelo. O primeiro cenário testado consistiu em 2 dias de produção com 1 dessas máquinas desligada. A lógica atual teve uma redução de 2,47% na produção e o DRL teve uma redução de 1,18%, resultando em um ganho de 1,29% em relação à lógica atual. O segundo cenário testado consistiu em 2 dias de produção com 3 dessas máquinas desligadas. A lógica atual apresentou uma redução na produção de 12,88%, enquanto o modelo de DRL teve uma redução de 12,64%, superando a lógica atual em 0,24%.

Por fim, foi realizado um teste considerando uma máquina ao final da linha que, dependendo do tempo parada, faz com que toda a linha entre em auto-restart até que essa máquina volte a operar. Considerando um cenário em que essa máquina para por 4 horas, a lógica atual teve uma perda de 17,02%, enquanto o DRL teve uma perda de 16,08%, resultando em um ganho de 0,94% em relação à lógica atual.

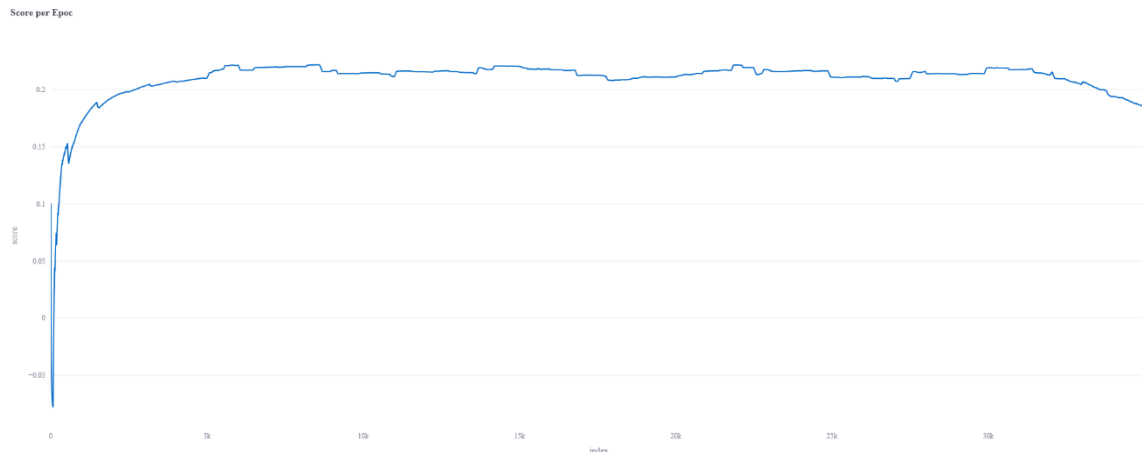
A Figura 4 apresenta um comparativo entre os testes do DRL e a lógica atual. Observa-se que o DRL demonstrou uma adaptação superior em todos os cenários, resultando em uma menor redução na produção e alcançando um ganho de 1,44% em relação à lógica atual.



Os ganhos obtidos com o DRL variam entre 1% e 2%, o que pode parecer pequeno à primeira vista. No entanto, para o cenário de uma multinacional, esse aumento representa a meta anual de crescimento de produção da empresa, traduzindo-se em ganhos financeiros na casa dos milhões. Isso evidencia a importância de melhorias aparentemente modestas em um ambiente produtivo de grande escala.

A Figura 5 representa o gráfico de Score por época que demonstra um rápido aumento inicial no desempenho do modelo de DQL, seguido por uma estabilização, indicando convergência do aprendizado. Após atingir um ponto de estabilidade, o modelo manteve um desempenho consistentemente alto, com pequenas variações. No entanto, observa-se um leve declínio nas últimas épocas, possivelmente sugerindo overfitting ou variações naturais nos dados. Em suma, o modelo apresentou um processo de treinamento bem-sucedido com rápida aprendizagem inicial e robustez no desempenho sustentado.

Figura 5 – Gráfico de Score por Época



Fonte: Autor

Portanto, a implementação do modelo de DQL demonstrou não apenas superar a lógica atual em condições normais, mas também proporcionar uma melhoria significativa em situações atípicas. Estes resultados confirmam a eficácia do modelo de RL, destacando seu potencial para melhorar a produção e minimizar as interrupções na linha de produção, consolidando-se como uma ferramenta valiosa para a gestão de sistemas produtivos complexos.

#### 4. Análise dos Resultados e Considerações Finais

Nesta pesquisa, explorou-se a aplicação inovadora do Aprendizado por Reforço Profundo (DRL) para aprimorar o controle de produção em uma grande empresa multinacional. O objetivo era integrar conceitos avançados de aprendizado de máquina com simulação de processos industriais para melhorar a eficiência e competitividade em um ambiente desafiador de produção.

Uma análise inicial dos resultados destacou a redução na quantidade de parâmetros como um aspecto significativo. Essa redução exigiu uma abordagem mais abstrata, focando nas ações do modelo e seu impacto nos parâmetros relevantes, como a velocidade das máquinas. Ao concentrar-se nos parâmetros próximos às máquinas principais, foi possível melhorar o treinamento do modelo.

O estudo evidenciou que, embora a lógica atual da linha de produção seja eficaz em cenários normais, o modelo de Aprendizado por Reforço (RL) se destaca em situações atípicas. Isso foi confirmado pelos testes realizados em pontos críticos da linha de produção. Ao simular o funcionamento com apenas uma das máquinas críticas em operação, o DQL mostrou uma redução na produção significativamente menor em comparação com a lógica atual, destacando sua superioridade em condições desafiadoras.

Da mesma forma, quando três máquinas críticas foram desligadas, o modelo de RL superou a lógica atual, evidenciando sua capacidade de lidar com situações adversas. Os resultados também foram complementados pela análise do comportamento da velocidade da máquina controlada pelo DQL ao longo do treinamento do modelo, ilustrando sua adaptação e eficácia no processo de modulação da linha de produção.

Apesar dos desafios e limitações, este estudo representa um avanço significativo na integração de tecnologias avançadas de inteligência artificial na indústria de produção. Os insights obtidos oferecem uma base sólida para pesquisas futuras, contribuindo para o avanço do conhecimento e busca contínua por eficiência e inovação na indústria.

Considerando os resultados, é possível destacar diversos benefícios práticos que a empresa pode obter com a aplicação do DRL na melhoria do controle de produção. Desde uma maior eficiência operacional e redução de custos até uma melhor adaptação a desafios complexos, o DRL oferece oportunidades tangíveis para melhorar a competitividade e excelência operacional da empresa.

Para futuras pesquisas, recomenda-se explorar outros algoritmos de aprendizado por reforço profundo e testar diferentes configurações de redes neurais e estratégias para a função de recompensa. Essas abordagens podem enriquecer ainda mais o campo de estudo e levar a avanços significativos na aplicação de inteligência artificial em ambientes industriais.

### Agradecimentos e financiamento

Os autores agradecem ao Fundo INOVAR-PE e ao Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq). O presente trabalho foi realizado com apoio da Coordenação de Aperfeiçoamento de Pessoal de Nível Superior – Brasil (CAPES) – Código de Financiamento 001. Os modelos de simulação deste trabalho foram feitos com a Siemens Tecnomatrix Plant Simulation Research License.

### Referências

- Agostinelli, F. et al. (2018). From Reinforcement Learning to Deep Reinforcement Learning: An Overview. In: *Computer Science*, p. 298-328, Irvine. Irvine.
- Altenmüller, T. et al. (2020) Reinforcement learning for an intelligent and autonomous production control of complex job shops under time constraints. *Production Management*, 14: 319-328.
- Feldkamp, N., Bergmann, S. e Strassburger, S. (2020). Simulation-Based Deep Reinforcement Learning For Modular Production Systems. In: *Winter Simulation Conference (WSC)*. Ilmenau. Technische Universität Ilmenau.
- Gosavi, A. (2015). Simulation-Based Optimization. Parametric Optimization - Techniques and Reinforcement Learning. Springer, New York.
- Huang, J., Chang, Q. e Chakraborty, N. (2019). Machine Preventive Replacement Policy for Serial Production Lines Based on Reinforcement Learning. In *15th International Conference on Automation Science and Engineering*. IEEE.
- Li, J. et al. (2022). A flexible manufacturing assembly system with deep reinforcement learning. *Control Engineering Practice*, 118: 104-957.
- Luo, S., Zhang, L., Fan, Y. (2021). Dynamic multi-objective scheduling for flexible job shop by deep reinforcement learning. *Computers & Industrial Engineering*, 159: 107-489.
- Marchesano, M. G. et al. (2022). Dynamic scheduling of a due date constrained flow shop with Deep Reinforcement Learning. *IFAC PapersOnLine*, 55(10): 2932-2937.
- Mnih, V. K. et al. (2015). Human-Level Control Through Deep Reinforcement Learning. *Nature*, 518(7540): 529–533.
- Alabadi, M. e Albayrak, Z. (2020). Q-Learning for Securing Cyber-Physical Systems : A survey. International Congress on Human-Computer Interaction, Optimization and Robotic Applications,
- Shrestha, A. e Mahmood, A. (2019). Review of Deep Learning Algorithms and Architectures. *IEEE Access*, 7: 53040–53065.
- Sutton, R. e Barto, A. (2018). Reinforcement Learning – An Introduction. 2ª Edição. The MIT Press, Cambridge, Massachusetts.
- Watkins, C. J. C. H. e Dayan, P. (1992). Q-learning. In *Machine Learning*, p. 279-292, Boston, Kluwer Academic Publisher.
- Zheng, W., Lei, T., Chang, Q. (2017). Comparison Study of Two Reinforcement Learning Based Real-Time Control Policies For Two-Machine-One-Buffer Production System. In *13th Conference on Automation Science and Engineering*, IEEE.