

Evolutionary Control of Industrial Processes Using Digital Twin

Renato Bruno Silva¹, Daniela Resende Silva Orbolato¹

¹Campus Uberaba Parque Tecnológico
Instituto Federal de Educação, Ciência e Tecnologia do Triângulo Mineiro (IFTM)
Uberaba, MG – Brasil

renato.b.s@hotmail.com, danielaorbolato@iftm.edu.br

Abstract. *The increasing variability in modern industrial processes necessitates control systems that are more flexible, capable of adapting without the need for traditional manual reprogramming. This study aims to demonstrate the feasibility of enhancing industrial control through an adaptive layer that responds dynamically to new operational scenarios, eliminating the requirement for re-configuration. To this end, the application of the NEAT algorithm is investigated for training an artificial intelligence model capable of autonomously learning and adjusting to changes within the production environment, modeled here through a digital twin. The results indicate that such an approach can achieve robust performance across various process configurations, paving the way for more resilient and autonomous automation systems.*

Resumo. *A crescente variabilidade nos processos industriais modernos exige sistemas de controle mais flexíveis, capazes de se adaptar sem reprogramações manuais tradicionalmente empregadas nesses casos. Este trabalho objetiva mostrar que é possível complementar o controle industrial com uma camada adaptativa que responda dinamicamente a novos cenários, sem necessidade de reconfiguração. Para isso, investiga a aplicação do algoritmo NEAT para treinar um modelo de inteligência artificial capaz de aprender e ajustar-se autonomamente a mudanças no ambiente produtivo, aqui modelado por meio de um gêmeo digital. Os resultados obtidos indicam que tal abordagem pode alcançar desempenho robusto frente a diferentes configurações de processo, abrindo caminho para sistemas de automação mais resilientes e autônomos.*

1. Introdução

A Indústria 4.0 representa um novo paradigma na manufatura, caracterizado pela digitalização, conectividade e uso intensivo de dados e inteligência artificial (IA) para tomada de decisão em tempo real. Nesse contexto, a flexibilidade e a resiliência dos sistemas de automação tornam-se requisitos essenciais para lidar com a crescente variabilidade dos processos produtivos – seja por alterações no mix de produtos, nas demandas de mercado, ou por mudanças físicas no layout fabril [Kagermann et al. 2013, Lee et al. 2015].

Tradicionalmente, o controle de processos industriais depende de Controladores Lógicos Programáveis (CLPs) configurados de forma estática, exigindo reprogramações manuais sempre que há modificações relevantes no processo. Essa abordagem não apenas

eleva o custo operacional, como também aumenta a dependência de especialistas e compromete a agilidade da produção. A limitação dessa rigidez funcional abre espaço para o uso de IA como camada adaptativa capaz de aprender políticas de controle diretamente da interação com o ambiente, levando à redução da necessidade de reconfiguração manual [Khan et al. 2024, Grus 2016].

A aplicação de algoritmos evolutivos em tarefas industriais é uma fronteira promissora da Inteligência Artificial, mas poucos estudos abordam sua eficácia em cenários práticos e visuais, como a separação autônoma de peças. Este trabalho propõe uma abordagem inovadora ao unir o algoritmo NEAT com a construção de um ambiente de simulação próprio, desenhado especificamente para treinar e validar redes neurais em tempo real, sem a necessidade inicial de *hardware* físico. A criação de tal ambiente virtual não apenas acelerou o processo de evolução da IA, mas também permitiu visualizar, em detalhes, como a rede neural se adaptava iterativamente a um sistema de transporte e classificação de objetos, algo pouco explorado na literatura mundial. A transição do ambiente virtual para simulações de controle de movimento representa um passo concreto na direção da adoção de métodos evolutivos em contextos industriais reais.

Este trabalho está estruturado da seguinte forma: a Seção 2 apresenta os fundamentos teóricos essenciais para o entendimento do estudo; a Seção 3 discute os trabalhos relacionados e a lacuna de trabalhos com a arquitetura abordada neste artigo; a Seção 4 descreve a aplicação prática e a metodologia que embasaram este estudo; a Seção 5 apresenta os resultados obtidos, acompanhados de uma discussão sobre os mesmos e as limitações identificadas; e, por fim, na Seção 6 são apresentadas as conclusões e as possibilidades de trabalhos futuros.

2. Fundamentação Teórica

A seção de fundamentação teórica contextualiza o uso da Inteligência Artificial (IA) evolutiva para automação industrial flexível. Aborda-se a Indústria 4.0, que demanda sistemas adaptativos além dos Controladores Lógicos Programáveis (CLPs) tradicionais, incapazes de lidar eficientemente com a alta variabilidade e personalização dos processos modernos [Kagermann et al. 2013, Lee et al. 2015].

Nesse cenário, os *Digital Twins*(DT), em português Gêmeos Digitais, surgem como ferramentas cruciais, oferecendo réplicas virtuais de alta fidelidade que permitem simular, otimizar e, fundamentalmente, treinar algoritmos de IA em um ambiente seguro, sem riscos operacionais [Grieves 2014, Qi and Tao 2018, Liu and Liu 2023].

O conceito de aprendizado adaptativo é apresentado como a capacidade da IA de ajustar suas estratégias autonomamente face a mudanças no ambiente (*concept drift*), superando as limitações de modelos supervisionados estáticos. Técnicas como neuroevolução e Aprendizado por Reforço aprendem com o *feedback* do ambiente, permitindo a reconfiguração contínua das políticas de controle [Haykin 1999, Sutton and Barto 2018, Fausett 1994].

Entre as abordagens de IA, a neuroevolução – em especial o algoritmo *Neuro-Evolution of Augmenting Topologies* (NEAT) – destaca-se para cenários que exigem grande adaptabilidade. O NEAT não apenas ajusta os pesos da rede como também evolui a própria topologia, característica decisiva em problemas de controle adaptativo nos quais

a arquitetura ótima não é conhecida a priori [Stanley and Miikkulainen 2002, Hoel 2023]. Seu funcionamento baseia-se em quatro mecanismos que se sustentam mutuamente, conforme figura 1: (i) *Starting minimally*, no qual a população nasce com redes extremamente simples; (ii) *Growth*, que adiciona nós e conexões conforme o progresso evolutivo; (iii) *Historical Marking (innovation numbers)*, que rotula cada gene novo para alinhar genomas de tamanhos distintos; e (iv) *Speciation*, que agrupa indivíduos geneticamente próximos e protege inovações radicais. Esses componentes formam um “triângulo” de dependências: começar mínimo torna o crescimento controlável; o crescimento cria genes que recebem marcações históricas; as marcações permitem calcular distâncias genéticas que definem as espécies; e a própria especiação, por isolar nichos, viabiliza trajetórias independentes de crescimento. Remover qualquer vértice desse triângulo costuma degradar o desempenho evolutivo.

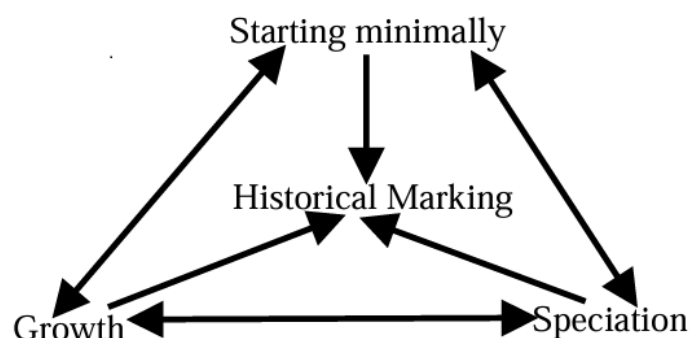


Figura 1. Dependências entre os componentes do NEAT

Fonte: [Stanley and Miikkulainen 2002]

Assim, diferentemente de métodos que partem de uma estrutura fixa, o NEAT inicia com arquiteturas mínimas (sem camadas ocultas) e, graças ao ciclo *Starting minimally* → *Growth* → *Historical Marking* → *Speciation*, insere nós e conexões apenas quando o ambiente “demonstra” necessidade de maior complexidade. Esse esquema de complexificação incremental evita um espaço de busca explosivo e preserva diversidade genética, garantindo que redes promissoras, mas ainda imaturas, não sejam descartadas prematuramente [Stanley and Miikkulainen 2002]. Por isso o NEAT é particularmente valioso em sistemas industriais e de processo, onde a topologia de controle ótima raramente é conhecida no início da operação e deve emergir – de forma segura e eficiente – ao longo da própria produção.

Vislumbra-se que a integração sinérgica entre Gêmeos Digitais e abordagens de IA adaptativa, como o NEAT, oferece um caminho promissor para o desenvolvimento de controladores industriais que não apenas operam, mas aprendem e evoluem com o processo, viabilizando a automação flexível e resiliente exigida pela Indústria 4.0.

3. Trabalhos Relacionados

Esta seção apresenta estudos que dão suporte à proposição central deste artigo — o emprego do algoritmo neuroevolutivo NEAT aliado a Gêmeos Digitais para controle adaptativo autônomo em processos industriais simulados. Até o momento, não foram identificados trabalhos que integrem diretamente essas duas abordagens; por isso, não há referência específica a pesquisas anteriores que tratem exatamente da solução aqui proposta. Ainda

assim, a literatura analisada evidencia a convergência crescente entre simulação avançada, inteligência artificial e controle adaptativo no contexto da Indústria 4.0, reforçando a pertinência e a originalidade desta investigação.

A viabilidade dos Gêmeos Digitais como plataformas eficazes para simulação, interação e treinamento de controladores de IA é validada por estudos como os de Ortiz [Ortiz et al. 2025] e Qiu [Qiu et al. 2025], que aplicam DT em contextos educacionais e de controle adaptativo (usando Redes Neurais Artificiais), respectivamente. O conceito evolutivo para Gêmeos Digitais Adaptativos (ADT), como proposto por Walmsley. [Walmsley et al. 2024], que não apenas simulam, mas aprendem e se adaptam, alinhando-se à visão de controle autônomo. Desafios arquiteturais e de integração de dados em DT complexos são abordados por Su. [Su et al. 2025], utilizando grafos de conhecimento.

A integração de IA com DT é uma área ativa, como evidenciado pela revisão de Huang [Huang et al. 2021] e por abordagens alternativas como o Aprendizado por Reforço (RL) aplicado por Ali [Ali et al. 2025] em manufatura aditiva robótica via DT. Esses trabalhos confirmam a relevância da combinação DT+IA para controle inteligente.

Apesar da crescente atividade na integração de IA com Gêmeos Digitais, conforme evidenciado pelos trabalhos mencionados, a maior parte da literatura recente tende a focar em abordagens baseadas em Aprendizado por Reforço ou *Deep Learning* para controle. Contudo, a aplicação e avaliação de algoritmos neuroevolutivos, como o NEAT, em cenários industriais simulados que envolvem aspectos práticos e visuais, como a manipulação e classificação de objetos em tempo real, ainda são menos prevalentes. Este trabalho busca preencher essa lacuna, investigando a eficácia do NEAT para desenvolver agentes de controle adaptativo em um ambiente de simulação visual, oferecendo uma perspectiva metodológica alternativa e complementar às abordagens mais comuns.

Finalmente, o trabalho seminal de Stanley e Miikkulainen [Stanley and Miikkulainen 2002] sobre o NEAT fornece a base algorítmica específica utilizada neste estudo, justificando sua escolha pela capacidade de evoluir tanto a estrutura quanto os parâmetros de redes neurais para tarefas de controle complexas e adaptativas. Em conjunto, essas referências estabelecem a relevância e a viabilidade da abordagem adotada, situando a contribuição deste artigo na interseção da simulação avançada com o controle inteligente evolutivo para automação industrial flexível.

4. Materiais e Métodos

Este estudo foi conduzido em um ambiente de simulação industrial utilizando ferramentas que viabilizam a experimentação segura de técnicas de controle inteligente.

O processo industrial foi modelado no sistema *Factory IO*, que, segundo sua documentação oficial, "é uma simulação de fábrica 3D para o aprendizado de tecnologias de automação. Projetado para ser fácil de usar, permite a construção rápida de uma fábrica virtual usando uma seleção de peças industriais comuns". O *Factory IO* também inclui diversas cenas inspiradas em aplicações industriais típicas, variando em níveis de dificuldade de iniciante a avançado"[REAL GAMES 2024].

O cenário simulado, que pode ser observado na Figura 2, consiste em uma esteira transportadora com sensores de presença e uma estação de carregamento automatizada, representando um fluxo típico de movimentação e processamento de peças.

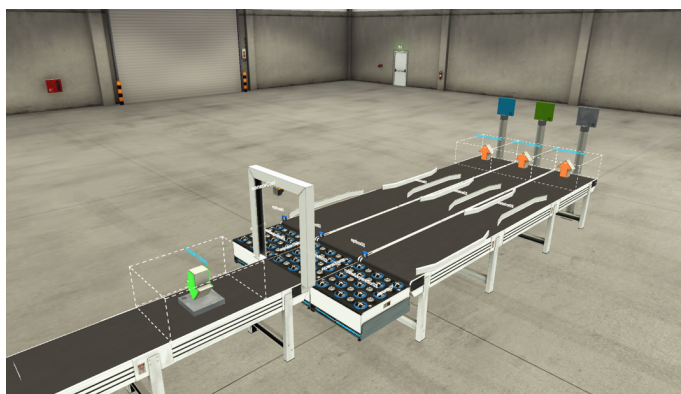


Figura 2. Cenário simulado com Factory IO

A interação entre a IA e o ambiente simulado foi viabilizada por meio da exportação dos sensores e atuadores do *Factory IO* através do protocolo *Modbus TCP/IP*, possibilitando comunicação em tempo real com *scripts em Python*. Tais *scripts*, juntamente com todos demais códigos escritos para este projeto, podem ser encontrados no GitHub do autor, assim garante-se que o trabalho pode ser reproduzido por qualquer pesquisador seguindo as instruções contidas no arquivo *readme.md*.

Para estabelecer a comunicação entre o agente de controle em *Python* e o processo simulado, foi utilizada a biblioteca *pyModbusTCP*. Essa biblioteca permite ler e escrever registradores *Modbus* de forma direta e eficiente, simulando a troca de dados entre um CLP e os dispositivos da planta. Foram mapeados os seguintes elementos:

- Entradas digitais: sensores de presença e estado do processo;
- Entradas analógicas: sensores de cor;
- Saídas digitais: motores, atuadores pneumáticos e sinalizações de processo.

O modelo foi implementado com a biblioteca *neat-python* [CodeReclaimers 2024], que fornece suporte ao algoritmo NEAT. A configuração do experimento foi definida no arquivo, especificando parâmetros como função de ativação, tamanho da população, número de entradas e saídas, e limiar de *fitness*.

O treinamento do agente evolutivo requer múltiplas gerações e populações, o que torna inviável sua execução diretamente no ambiente 3D do *Factory IO* devido ao elevado tempo de simulação e à limitação de complexidade dos cenários. Para contornar esse problema, o processo alvo de controle foi abstraído e reimplementado diretamente em *Python*, permitindo a realização de *backtests* automatizados mais rápidos, sem a dependência do simulador gráfico. Essa estratégia possibilitou um ciclo de treino mais fluido e escalável. Após o treinamento, a política evoluída pode ser exportada e validada no ambiente do *Factory IO*, permitindo verificar sua aplicabilidade prática em uma simulação industrial realista.

O treinamento da IA foi realizado com base em uma tarefa de classificação sensorial simulada. Cada indivíduo da população NEAT representa uma rede neural cujo objetivo é identificar corretamente a esteira de saída correspondente à "cor" de uma peça, simulada como um valor discreto gerado aleatoriamente (1, 4 ou 7), equivalentes às caixas azul, verde e cinza no *Factory IO*.

O valor de entrada é previamente normalizado para o intervalo $[-1, 1]$, prática

recomendada em redes neurais por melhorar a estabilidade numérica e acelerar a convergência durante o treinamento [Goodfellow et al. 2016]. Em vez de utilizar múltiplos neurônios na saída para representar cada possível classe (ou esteira), optou-se por uma abordagem mais compacta, com apenas uma saída contínua, posteriormente decodificada para uma ação discreta. Essa estratégia reduz a complexidade da rede e é adequada para tarefas onde os rótulos têm correspondência direta com valores numéricos [Neuromatch Academy 2024]. Após o processamento pela rede, a saída é convertida em um índice de esteira por meio de uma função no código. Se a ação tomada corresponder corretamente ao valor do sensor, o *fitness* do agente é incrementado em +1; caso contrário, uma penalização de -2 é aplicada. A decisão é então validada por sensores de presença localizados nas esteiras de saída, conforme representado no diagrama da Figura 3.

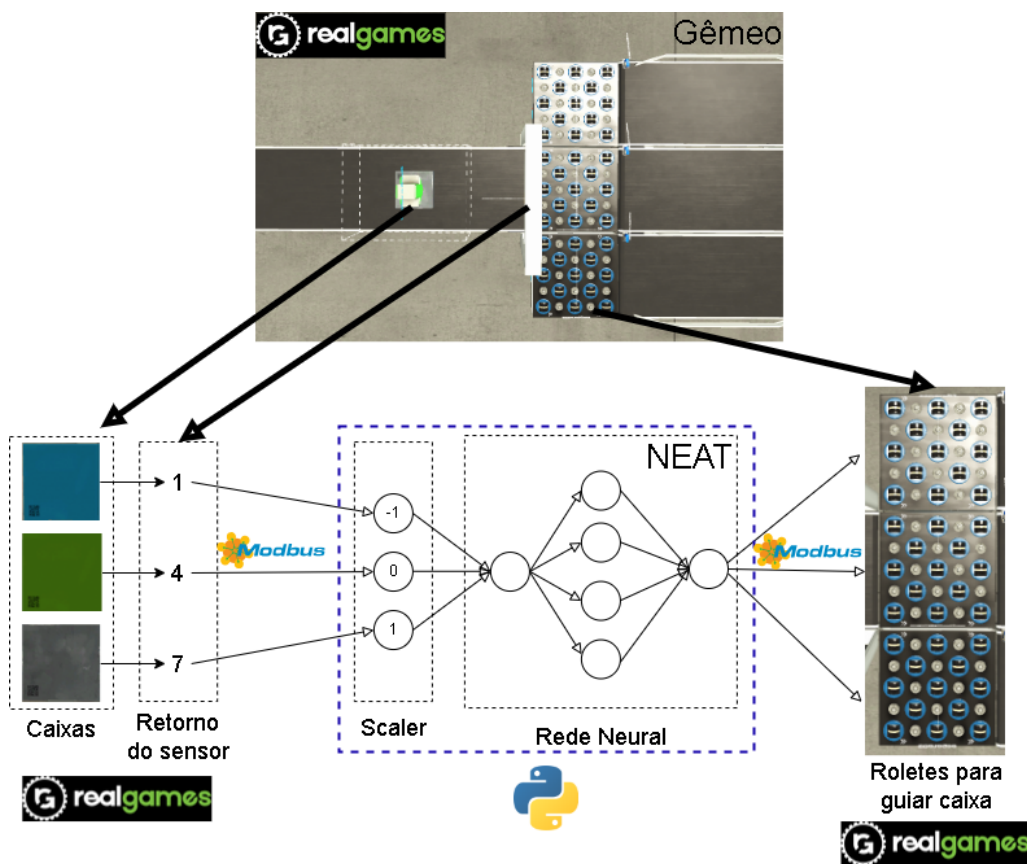


Figura 3. Fluxograma do processo de treinamento e avaliação

A simulação continua até que o indivíduo atinja o limiar de desempenho definido ou um número máximo de passos seja alcançado. O processo evolutivo se repete por várias gerações, selecionando os indivíduos mais aptos e aplicando operadores genéticos (cruzamento e mutação) para gerar novas populações.

Ao final de cada geração, os resultados são registrados em um arquivo CSV, e os genomas com maior *fitness* são salvos para visualização e reuso. Após o treinamento, o modelo poderá ser exportado e testado diretamente no *Factory IO*, validando sua capacidade de generalização frente a entradas reais.

5. Resultados

A avaliação da abordagem proposta foi realizada com base no desempenho do agente treinado com NEAT em uma tarefa de classificação sensorial simulada. A cada geração, os indivíduos foram testados em ciclos repetidos nos quais precisavam identificar corretamente a esteira de saída correspondente ao valor do sensor (relacionado à cor da caixa). O desempenho foi medido pelo valor de *fitness* acumulado, calculado com +1 ponto por acerto e -2 por erro. Esse valor foi registrado geração após geração, possibilitando a análise da curva de aprendizado do sistema. O histórico dos melhores indivíduos foi salvo para posterior validação e análise da estabilidade da política aprendida.

A Figura 4 reúne, em um único painel, a tabela com o número de gerações até a convergência e o gráfico da pontuação máxima alcançada em cada geração para diferentes tamanhos de população. Nota-se que populações muito pequenas (2 e 4 indivíduos) exibem grande variabilidade: o traçado laranja oscila por quase trezentas épocas até alcançar a solução, enquanto a linha azul-clara estabiliza-se após pouco mais de cem. Já com 10, 20 e 100 indivíduos (linhas verde, marrom e azul-escura, respectivamente) o aprendizado acelera dramaticamente, exigindo apenas 12, 3 e 1 geração para convergir, como quantificado na própria tabela inserida acima do eixo das gerações.

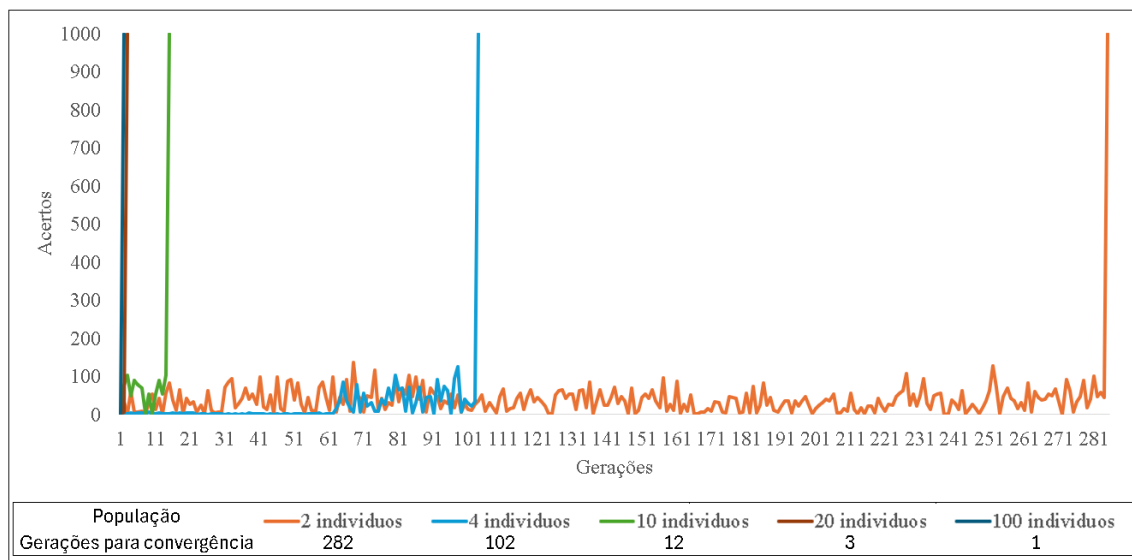


Figura 4. Comparativo entre gerações e número de indivíduos para treinamento

Esse resultado ilustra um comportamento clássico dos algoritmos evolutivos: quanto maior a diversidade populacional, maior a cobertura do espaço de busca e mais rápido o encontro de políticas eficazes. O salto súbito no fim da curva laranja (população = 2) evidencia que, mesmo com exploração limitada, o NEAT pode descobrir uma topologia vitoriosa — mas somente após centenas de mutações estocásticas. A arquitetura vencedora obtida nessa execução observada na Figura 5 permanece minimalista: um único neurônio de entrada ligado a um nó oculto com ativação tanh, que por sua vez conecta-se diretamente ao neurônio de saída, sem camadas adicionais nem conexões recorrentes — prova de que o algoritmo converge para uma solução simples, porém suficiente para a tarefa.

Esse resultado demonstra a capacidade do NEAT de adaptar a complexidade da rede de forma incremental e apenas quando necessário. Mesmo com uma arquitetura reduzida, a rede foi capaz de generalizar o comportamento de separação sensorial com alto desempenho.

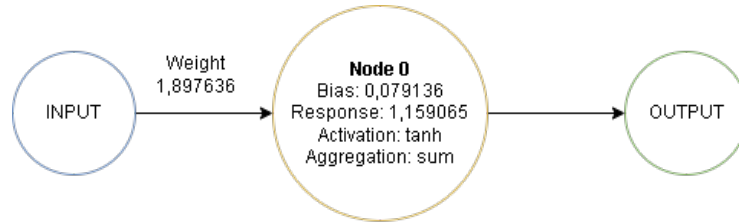


Figura 5. Fluxograma da rede no experimento com 3 caminhos na esteira

Para validar o funcionamento da rede, é possível realizar os cálculos, dada a simplicidade da vencedora.

- Para a caixa azul, o sensor retorna o valor 1, que normalizado se transforma em -1.
- Para calcular a entrada do nó, basta:

$$S = [weight * input] + bias$$

$$S = [1.897636 * (-1)] + (-0.079136) \therefore S = -1.8185$$

- Aplicação da função de ativação:

$$ativacao = \tanh(S)$$

$$ativacao = \tanh(-1.8185) \therefore ativacao = -0.948688652$$

- Aplicando a resposta (ganho):

$$Output = ativacao * response$$

$$Output = -0.948688652 * 1.159065 \therefore -1.099591812$$

- Decodificando: Ação de decodificação é a seguinte:

```
def decode_outputs(output, esteiras):
    return round((output + 1) / 2 * (esteiras - 1))
```

$$decodificado = round((-1,099591812 + 1) / 2 * (3 - 1)) \therefore decodificado = 0$$

Assim a primeira caixa é separada para a esteira 1 decodificada em [0], conforme previsto. (A variável esteiras, é a quantidade de esteiras usadas no treinamento).

Na Tabela 1, os cálculos para as outras caixas podem ser observados.

Tabela 1. Cálculos para caixas azul, verde e metálica

Cor caixa	Código	Normalizado	Entrada Neurônio	Função de ativação	Output	Decodificado	Esteira
Azul	1	-1	-1,8185	-0,948688652	-1,099591812	-0,099591812	0
Verde	4	0	0,079136	0,078971216	0,091532773	1,091532773	1
Metálica	7	1	1,976772	0,962349222	1,115425301	2,115425301	2

Após o término do treinamento, o modelo evoluído foi exportado e integrado ao ambiente tridimensional do *Factory IO*. O objetivo dessa etapa foi validar, em condições mais realistas, o desempenho do agente em um cenário de separação de peças com controle visual, sensores e atuadores simulados.

A rede treinada foi carregada a partir do arquivo.pkl e passou a processar, em tempo real, os dados dos sensores (via protocolo Modbus TCP). Cada peça detectada por um sensor frontal é representada por uma cor (azul, verde ou metálica), convertida em um valor normalizado no intervalo $[-1, 1]$, e processada pela rede neural. O modelo responde com uma saída contínua, que é decodificada para uma das três esteiras de destino — cada uma correspondente a uma classe de peça. Durante os testes, o modelo demonstrou comportamento consistente e generalização adequada, direcionando corretamente as peças mesmo com variações na sequência de entrada. Essa etapa confirmou a viabilidade de uso da IA evoluída não apenas em simulação lógica, mas em uma simulação industrial visual mais próxima do ambiente real. A Figura 6 apresenta visão do ambiente no *Factory IO* durante a execução do modelo, ilustrando visualmente a efetividade do sistema de separação inteligente.

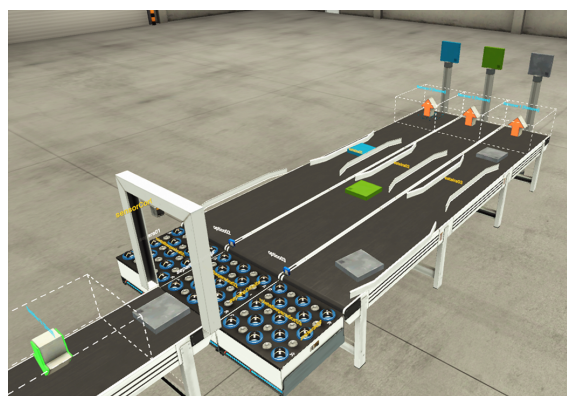


Figura 6. Modelo executando a separação de caixas no Factory IO

Embora controladores convencionais baseados em CLPs sejam amplamente utilizados na automação industrial, sua aplicação em processos que envolvem a classificação discreta de múltiplas entradas em tempo real pode exigir regras complexas e reprogramações frequentes. Neste trabalho, comparou-se a eficiência prática da abordagem baseada em NEAT com uma implementação tradicional em CLP, estruturada por meio de regras fixas e temporizações, avaliando a flexibilidade e adaptabilidade de cada solução frente a mudanças no processo.

Observa-se que a programação inicial da rede neural é mais demorada do que a lógica de controle padrão em CLPs. No entanto, após o treinamento, o sistema baseado em IA mostra-se altamente adaptativo, eliminando a necessidade de reprogramações manuais sempre que o processo sofre alterações, como mudanças no número ou tipo de caixas.

Para comprovar essa adaptabilidade, um novo experimento foi realizado com a esteira configurada para lidar com 9 tipos distintos de caixas, como ilustra a Figura 7.

A rede foi retreinada para essa nova condição com grande eficiência, sem a necessidade de alterações significativas na lógica de controle, apenas ajustando a quantidade de saídas esperadas. O processo de treinamento foi rápido, e a rede evoluiu naturalmente para uma topologia mais robusta capaz de classificar corretamente todos os casos, como observado nas Figuras 8 e 9.

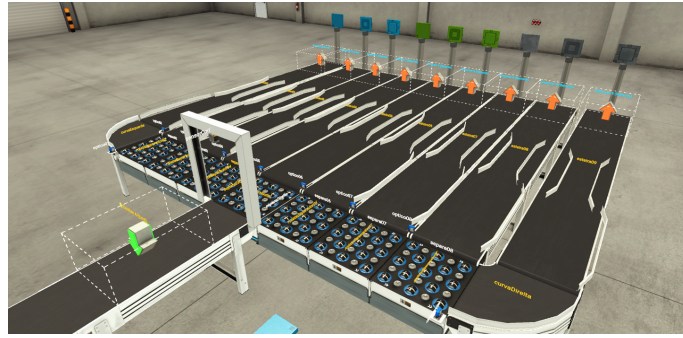


Figura 7. Planta de separação com 9 possibilidades de caminho



Figura 8. Acertos por gerações

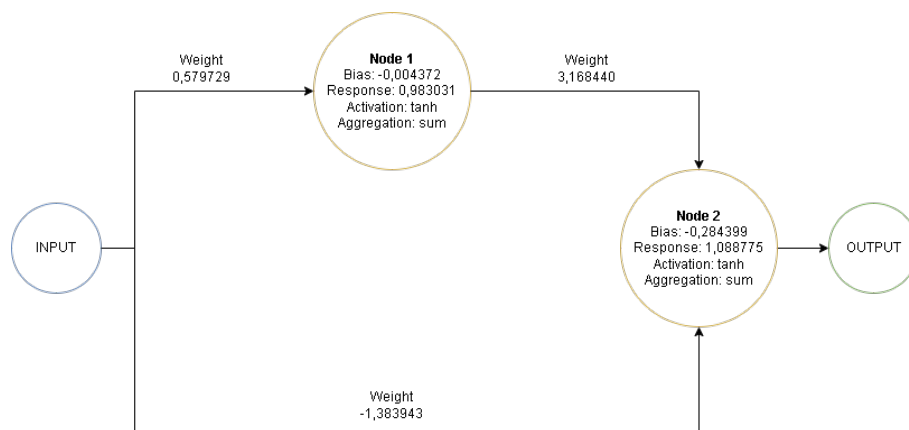


Figura 9. Fluxograma da rede no experimento com 9 caminhos na esteira

6. Conclusões e trabalhos futuros

Este trabalho demonstrou com sucesso a viabilidade e a eficácia do algoritmo neuroevolutivo NEAT para o desenvolvimento de agentes de controle adaptativo em ambientes de gêmeos digitais. Os resultados empíricos indicam que o NEAT pode ajustar dinami-

camente não apenas os parâmetros, mas a própria topologia da rede neural para atender às demandas de um processo industrial simulado, validando seu potencial para alcançar a flexibilidade e a resiliência exigidas pela Indústria 4.0. A abordagem se alinha com a visão emergente de *Adaptive Digital Twins* (ADT), Gêmeos Digitais Adaptativos, que aprendem e evoluem com o processo.

A principal contribuição deste estudo foi preencher uma lacuna na literatura, aplicando um algoritmo neuroevolutivo (NEAT) em um cenário de controle industrial prático e visual, algo menos explorado que as abordagens convencionais de Aprendizado por Reforço. A capacidade do sistema de generalizar o aprendizado, desde uma simulação puramente lógica até um ambiente 3D interativo com o *Factory IO*, e de se adaptar a um aumento na complexidade da tarefa (de 3 para 9 tipos de caixas), reforça a robustez da solução proposta.

Apesar dos resultados promissores, o estudo possui limitações que devem ser consideradas. Primeiramente, a validação ocorreu exclusivamente em ambiente simulado. Embora o *Factory IO* ofereça uma representação realista, ele pode não capturar todas as nuances de um ambiente de produção real, como ruídos de sensores, falhas mecânicas intermitentes ou a latência de redes industriais. Em segundo lugar, o estudo focou em uma tarefa de classificação bem definida. A complexidade do controle em cenários industriais mais dinâmicos, que envolvem múltiplos sensores e atuadores operando em paralelo ou a otimização de variáveis de processo contínuas, não foi investigada. Por fim, a interpretabilidade das redes neurais evoluídas, embora minimalistas, ainda representa um desafio, o que pode ser um obstáculo para a adoção em indústrias que exigem alta confiabilidade e diagnósticos claros.

Como trabalhos futuros, sugere-se realizar a implementação da abordagem aqui proposta em uma planta física em um ambiente real de produção para com o objetivo de validar tal abordagem. Outra sugestão seria também expandir a análise comparativa para incluir outras técnicas de controle adaptativo, como o Aprendizado por Reforço, a fim de avaliar os *trade-offs* em desempenho e custo computacional. Adicionalmente, investigar a robustez do NEAT frente ao *concept drift* (mudanças inesperadas no processo) e explorar métodos para aumentar a interpretabilidade das redes evoluídas são direções de pesquisa essenciais para viabilizar a aplicação industrial da solução.

Referências

- Ali, M., Liu, S., and Yang, Q. (2025). Digital twin-enabled real-time control in robotic additive manufacturing via soft actor-critic reinforcement learning. *arXiv preprint arXiv:2501.18016*.
- CodeReclaimers (2024). Neat-python documentation. <https://neat-python.readthedocs.io/>. Accessed: 2025-05-26.
- Fausett, L. (1994). *Fundamentals of neural networks*. Prentice Hall.
- Goodfellow, I., Bengio, Y., and Courville, A. (2016). *Deep learning*. MIT Press.
- Grieves, M. (2014). Digital twin: manufacturing excellence through virtual factory replication. White paper. Available online: <https://www.3ds.com/fileadmin/PRODUCTS-SERVICES/DELMIA/PDF/Whitepaper/DELMIA-APRISO-Digital-Twin-Whitepaper.pdf>.

- Grus, J. (2016). *Data science do zero*. Alta Books.
- Haykin, S. (1999). *Neural networks: principles and practice*. Pearson, 2 edition.
- Hoel, A. (2023). Balancing pole with neat in python.
- Huang, Z., Shen, Y., Li, J., Fey, M., and Brecher, C. (2021). A survey on ai-driven digital twins in industry 4.0: Smart manufacturing and advanced robotics. *Sensors*, 21(19):6340.
- Kagermann, H., Wahlster, W., and Helbig, J. (2013). Recommendations for implementing the strategic initiative industrie 4.0. Technical report, National Academy of Science and Engineering, Frankfurt.
- Khan, S. et al. (2024). Industrial ai: a 2024 review. *IEEE Access*, 12:12345–12356.
- Lee, J., Bagheri, B., and Kao, H. A. (2015). A cyber-physical systems architecture for industry 4.0-based manufacturing systems. *Manufacturing Letters*, 3:18–23.
- Liu, J. and Liu, B. (2023). Digital twin driven rl for production lines. *Applied Sciences*, 13(3):456.
- Neuromatch Academy (2024). Neural decoding with deep learning. https://compneuro.neuromatch.io/tutorials/W1D5_DeepLearning. Accessed: 2025-05-26.
- Ortiz, J. S., Quishpe, E. K., Sailema, G. X., and Guamán, N. S. (2025). Digital twin-based active learning for industrial process control and supervision in industry 4.0. *Sensors*, 25(7):2076.
- Qi, Q. and Tao, F. (2018). Digital twin and big data towards smart manufacturing and industry 4.0: 360 degree comparison. *IEEE Access*, 6:3585–3593.
- Qiu, H., Al-Nussairi, A. K. J., Chevinli, Z. S., Singh, N. S. S., Chyad, M. H., Yu, J., and Maesoumi, M. (2025). Integrating digital twins with neural networks for adaptive control of automotive suspension systems. *Scientific Reports*, 15(1):11078.
- REAL GAMES (2024). Factory i/o documentation. <https://docs.factoryio.com/>. Accessed on: 2025-04-25.
- Stanley, K. O. and Miikkulainen, R. (2002). Evolving neural networks through augmenting topologies. *Evolutionary Computation*, 10(2):99–127.
- Su, C., Tang, X., Jiang, Q., Han, Y., Wang, T., and Jiang, D. (2025). Digital twin system for manufacturing processes based on a multi-layer knowledge graph model. *Scientific Reports*, 15(1):12835.
- Sutton, R. S. and Barto, A. G. (2018). *Reinforcement learning: an introduction*. The MIT Press, 2 edition.
- Walmsley, T. G., Patros, P., Yu, W., Young, B. R., Burroughs, S., Apperley, M., Carson, J. K., Udugama, I. A., Aeowjaroenlap, H., Atkins, M. J., and Walmsley, M. R. W. (2024). Adaptive digital twins for energy-intensive industries and their local communities. *Digital Chemical Engineering*, 10:100139.