Universidade Federal de Ouro Preto
Escola de Minas
Engenharia de Controle e Automação

Yasmin Adriane de Paula Campos

APLICAÇÃO DE APRENDIZADO POR REFORÇO EM UM SISTEMA DE AUTOMAÇÃO INDUSTRIAL

Ouro Preto, 2026

Yasmin Adriane de Paula Campos

APLICAÇÃO DE APRENDIZADO POR REFORÇO EM UM SISTEMA DE AUTOMAÇÃO INDUSTRIAL

Monografia apresentada ao Curso de Engenharia de Controle e Automação da Universidade Federal de Ouro Preto como requisito parcial para obtenção do título de Engenheira de Controle e Automação.

Orientador: Me. Lucas Andery Reis
Coorientador: Dr. José Agnaldo da R. Reis

Ouro Preto, 2026

MINISTÉRIO DA EDUCAÇÃO
UNIVERSIDADE FEDERAL DE OURO PRETO
REITORIA
ESCOLA DE MINAS
DEPARTAMENTO DE ENGENHARIA CONTROLE E AUTOMACAO

FOLHA DE APROVAÇÃO

Yasmin Adriane de Paula Campos

Aplicação de Aprendizado por Reforço em um Sistema de Automação Industrial

Monografia apresentada ao Curso de Engenharia de Controle e Automação da Universidade Federal de Ouro Preto como requisito parcial para obtenção do título de Engenheira de Controle e Automação

Aprovada em 02 de março de 2026

Membros da banca

Mestre - Lucas Andery Reis - Orientador (Vale SA)
Doutor - Agnaldo José da Rocha Reis - Coorientador (Universidade Federal de Ouro Preto)
Doutor - Danny Augusto VieiraTonidandel - Examinador (Universidade Federal de Ouro Preto)
Doutor - Thomás Vargas Barsante Pinto - Examinador (Instituto Tecnológico Vale)

Lucas Andery Reis e Agnaldo José da Rocha Reis, orientadores do trabalho, aprovaram a versão final e autorizaram seu depósito na Biblioteca Digital de Trabalhos de Conclusão de Curso da UFOP em 07/03/2026.

Documento assinado eletronicamente por **Agnaldo Jose da Rocha Reis**, **PROFESSOR DE MAGISTERIO SUPERIOR**, em 07/03/2026, às 23:00, conforme horário oficial de Brasília, com fundamento no art. 6º, § 1º, do Decreto nº 8.539, de 8 de outubro de 2015.

A autenticidade deste documento pode ser conferida no site http://sei.ufop.br/sei/controlador_externo.php?acao=documento_conferir&id_orgao_acesso_externo=0 , informando o código verificador **1070941** e o código CRC **C7F16E34**.

R. Diogo de Vasconcelos, 122, - Bairro Pilar Ouro Preto/MG, CEP 35402-163
Telefone: 3135591533 - www.ufop.br

Agradecimentos

Agradeço aos meus pais, Rosely e Adriano, por serem o alicerce de tudo. Este trabalho é o reflexo do esforço e da renúncia de vocês para que eu pudesse trilhar os meus próprios passos. Às minhas irmãs e almas gêmeas, Ingrid e Andreany, e ao meu irmão Hiago, agradeço pelo apoio incondicional e por sempre me lembrarem de quem eu sou.

Às minhas raízes, minhas avós Rosária e Raimunda, e à memória de meus avôs Alair e Felipe e de meu tio Ezequias, cujo legado de bondade e inspiração continuam vivos em mim. À toda minha família, meu carinho pelo apoio, amor e conselhos.

Ao Kayo, pela paciência, pelo cuidado diário e por estar ao meu lado em cada etapa. Sua presença tornou o caminho mais leve e me deu forças para seguir adiante.

Aos meus amigos da vida, e em especial à Samira e ao Matias, minha gratidão pelo incentivo e por acompanharem minha trajetória, tornando a distância irrelevante.

À República Fruto Proibido, minha eterna casinha, e às irmãs que me deu. O apoio de vocês foi fundamental para guiar meu crescimento e garantir que eu chegasse até aqui.

Aos amigos que fiz ao longo da graduação, pelo suporte mútuo e por compartilharem as angústias e vitórias. Em especial, à Júlia Elice e Júlia Reis, por caminharem ao meu lado do início ao fim, e pela linda amizade que trouxe leveza à conclusão deste ciclo.

Aos meus orientadores, Lucas e Agnaldo, pela condução brilhante e pela confiança ao me entregarem este tema.

À Vale, ao ITV e ao GEPSA, pelas experiências e pelo aprendizado prático.

À Universidade Federal de Ouro Preto, à Escola de Minas e à Fundação Gorceix, pela excelência do ensino público e por todas as oportunidades oferecidas.

Por fim, minha gratidão a todos que, direta ou indiretamente, contribuíram para que este sonho se tornasse realidade.

Resumo

O avanço da Indústria 4.0 tem ampliado o interesse pela aplicação de algoritmos avançados de controle e aprendizado de máquina em sistemas de automação industrial. Nesse contexto, métodos baseados em Aprendizado por Reforço (*Reinforcement Learning* - RL) destacam-se pelo potencial de adaptação a processos dinâmicos e incertos, operando sem a necessidade de modelos matemáticos explícitos do processo. Entretanto, a aplicação prática desses algoritmos no chão de fábrica ainda é limitada, restringida pela arquitetura de *hardware* dos Controladores Lógicos Programáveis (CLP) e pelas exigências de determinismo e alocação estática de memória. Objetiva-se desenvolver um *framework* para a transpilação automática de políticas de aprendizado por reforço, treinadas em ambiente Python, para a linguagem Texto Estruturado (ST), em conformidade com a norma IEC 61131-3. A metodologia consiste no treinamento *offline* de um agente *Q-learning* em ambiente simulado, seguido da transpilação da matriz de conhecimento e do código de controle a ser inserido no CLP. Como estudo de caso, considerou-se o controle de um sistema dinâmico de primeira ordem, utilizando um controlador Proporcional-Integral (PI) como *baseline* de desempenho. A política convertida é executada e validada em ambiente de automação industrial simulado utilizando o *software* de programação *EcoStruxure Control Expert*. Os resultados demonstraram viabilidade técnica da conversão, preservando a integridade da lógica de decisão do agente. Na análise comparativa de desempenho, o controlador baseado em RL superou um controlador PI clássico sintonizado analiticamente, apresentando uma redução de 16,8% na Integral do Erro Absoluto (IAE). A solução proposta oferece uma alternativa robusta escalável para a modernização de malhas de controle em setores críticos, como a mineração, viabilizando a adoção de estratégias adaptativas em plataformas de automação consolidadas.

Palavras-chaves: Aprendizado de Máquina; Aprendizado por Reforço; *Q-learning*; CLP; Texto Estruturado; Automação Industrial; Mineração 4.0.

Abstract

The advancement of Industry 4.0 has expanded interest in the application of advanced control algorithms and machine learning in industrial automation systems. In this context, methods based on Reinforcement Learning (RL) stand out for their potential to adapt to dynamic and uncertain processes, operating without the need for explicit mathematical models. However, the practical application of these algorithms on the shop floor is still limited, constrained by the hardware architecture of Programmable Logic Controllers (PLCs) and by the requirements for determinism and static memory allocation. The objective is to develop a framework for the automatic translation of reinforcement learning policies, trained in a Python environment, into Structured Text (ST) language, in compliance with the IEC 61131-3 standard. The methodology consists of the offline training of a Q-learning agent in a simulated environment, followed by the translation of the knowledge matrix and the control code for insertion into the PLC. As a case study, the control of a first-order dynamic system was considered, using a Proportional-Integral (PI) controller as a performance baseline. The converted policy is executed and validated in a simulated industrial automation environment using the EcoStruxure Control Expert programming software. The results demonstrated the technical feasibility of the conversion, preserving the integrity of the agent's decision logic. In the comparative performance analysis, the RL-based controller outperformed a classically tuned PI controller, achieving a 16.8% reduction in the Integral of Absolute Error (IAE). The proposed solution offers a robust and scalable alternative for the modernization of control loops in critical sectors, such as mining, enabling the adoption of adaptive strategies on established automation platforms.

Key-words: Machine Learning; Reinforcement Learning; Q-learning; PLC; Structured Text; Industrial Automation; Mining 4.0.

Lista de Figuras

Lista de abreviaturas e siglas

CLP	Controlador Lógico Programável (Programmable Logic Controllers)
IA	Inteligência Artificial (Artificial Intelligence)
IAE	Integral do Erro Absoluto (Integral of Absolute Error)
IDE	Ambiente de Desenvolvimento Integrado (Integrated Development Environment)
IEC	Comissão Eletrotécnica Internacional (International Electrotechnical Commission)
IoT	Internet das Coisas (Internet of Things)
MDP	Processo de Decisão de Markov (Markov Decision Process)
PI	Controlador Proporcional-Integral (Proportional-Integral Controller)
PID	Controlador Proporcional-Integral-Derivativo (Proportional-Integral-Derivative Controller)
PLS	Mínimos Quadrados Parciais (Partial Least Squares)
RL	Aprendizado por Reforço (Reinforcement Learning)
SARSA	Estado-Ação-Recompensa-Estado-Ação (State–Action–Reward–State–Action)
SCADA	Sistema de Supervisão e Aquisição de Dados (Supervisory Control And Data Acquisition)
ST	Texto Estruturado (Structured Text)
TD	Diferença Temporal (Temporal Difference)

Sumário

1 Introdução

1.1 Contexto

A convergência de tecnologias digitais, Internet das Coisas (IoT) e inteligência artificial tem redesenhado o panorama industrial nas últimas décadas, dando origem ao paradigma da Indústria 4.0. Esse movimento caracteriza-se pela integração de sistemas físicos e digitais, pela intensificação da coleta e análise de dados em tempo real e pela incorporação de métodos avançados de automação e tomada de decisão. No setor mineral, essa transformação assume papel ainda mais relevante, uma vez que operações distribuídas geograficamente, plantas de grande porte e elevada variabilidade operacional impõem desafios adicionais de segurança, eficiência energética e produtividade (FERREIRA; GONTIJO et al., 2022; DIAS, 2023).

A evolução rumo à *Mining* 4.0 parte de uma trajetória histórica de transformações industriais. Conforme discutido por Dias (2023), a mineração acompanhou as quatro revoluções industriais, desde a mecanização a vapor (1.0), passando pela eletrificação e produção em massa (2.0) e pela automação baseada em eletrônica e TI (3.0), até a atual integração de sistemas ciberfísicos, IoT, inteligência artificial e *big data* (4.0). Essa última fase, também chamada de *Smart Mining*, busca não apenas ganhos de eficiência e produtividade, mas também a redução de impactos ambientais e o aumento da segurança operacional, refletindo a crescente demanda por operações mineradoras mais sustentáveis e tecnologicamente integradas.

É nesse cenário que a digitalização dos processos de lavra e beneficiamento tem como objetivo ampliar a autonomia operacional, reduzir a dependência de intervenção humana e aumentar a previsibilidade do processo produtivo. Entretanto, a natureza dos sistemas minerais, marcados por não linearidades, conexão entre variáveis, atrasos de medição e mudanças frequentes na qualidade do minério, limita a efetividade de estratégias de controle baseadas exclusivamente em modelos fixos ou regras empíricas, conforme discutido em aplicações industriais reais (FERREIRA; GONTIJO et al., 2022).

Por isso, grandes empresas mineradoras como Vale, Rio Tinto e BHP já adotam tecnologias como frotas autônomas, drones para monitoramento, sensores IoT e centros de operações remotas, conforme documentado em estudos de caso (DIAS, 2023). No entanto, a implementação de soluções baseadas em inteligência artificial adaptativa, em especial o aprendizado por reforço, ainda é incipiente, especialmente quando se trata de execução direta em controladores industriais como os CLPs.

Diante dessas limitações, a indústria tem adotado soluções de controle avançado que

permitem maior adaptação às variações do processo. Um exemplo recente é a implementação de estratégias de controle *override* baseadas na corrente elétrica de peneiras vibratórias para otimizar a eficiência do peneiramento primário a umidade natural (ALBUQUERQUE et al., 2023). Nessa aplicação, uma correlação estatística entre a corrente do motor, características do minério (umidade, litologia) e a taxa de alimentação permitiu criar uma malha de controle que ajusta dinamicamente a velocidade dos alimentadores, reduzindo em 67% as paradas por sobrecarga e elevando a operação automática para 98,5% do tempo. Este caso ilustra como a automação adaptativa, ainda que baseada em regras pré-definidas e correlações estáticas, pode gerar ganhos operacionais significativos em cenários de alta variabilidade. Contudo, mesmo estratégias como essa possuem uma limitação intrínseca: não aprendem com a experiência nem se ajustam continuamente a novas condições não previstas durante o projeto, dependendo de intervenção humana para reajustes.

Nesse contexto de busca por modernização, sensores virtuais, ou *soft sensors*, têm se consolidado como ferramentas essenciais em plantas industriais modernas. Esses sensores permitem a estimação em tempo real de variáveis de processo que não dispõem de instrumentação direta ou cuja medição é lenta e onerosa, como granulometria, teor e densidade de polpa. Em operações de beneficiamento mineral, tais variáveis são frequentemente obtidas por análises laboratoriais com latência significativa, dificultando a adoção de estratégias de controle mais responsivas. O uso de *soft sensors* possibilita inferir essas grandezas a partir de sinais já disponíveis no sistema de automação, além de operar de forma redundante a instrumentos físicos, contribuindo para detecção de falhas e estratégias de *fallback* (FERREIRA; BRAGA; SEIXAS FILHO, 2010).

Paralelamente, o avanço de técnicas de modelagem e simulação tem impulsionado o uso de gêmeos digitais como representações virtuais de processos industriais. Esses modelos permitem testar estratégias de controle, validar algoritmos e antecipar comportamentos do sistema real em ambientes seguros e controlados. No contexto da mineração, os gêmeos digitais desempenham papel fundamental ao viabilizar o desenvolvimento e o treinamento de algoritmos avançados sem comprometer a segurança operacional, constituindo uma base natural para a aplicação de métodos de aprendizado por reforço.

É nessa lacuna, entre a automação baseada em regras estáticas e a necessidade de sistemas que aprendam continuamente, que o aprendizado por reforço se apresenta como uma evolução promissora. Diferentemente de soluções pré-programadas, o RL permite que um agente aprenda iterativamente a partir da interação com o ambiente, desenvolvendo políticas de controle que se refinam com o tempo e se adaptam a variações não modeladas inicialmente. A combinação de *soft sensors* com algoritmos de RL pode potencializar não apenas a estimação, mas também a otimização adaptativa das variáveis controladas, criando sistemas de controle cada vez mais autônomos e resilientes.

Este trabalho propõe um *framework* para a transpilação de um algoritmo de

modelo de aprendizado por reforço para permitir sua integração em controladores lógicos programáveis de uma planta de processo de mineração como estudo de caso. Com isso, é introduzida uma inteligência adaptativa ao chão de fábrica de um sistema de controle preenchendo a lacuna entre a pesquisa e a aplicação prática em ambientes industriais reais.

1.2 Motivação

A indústria mineral enfrenta desafios operacionais que impactam diretamente sua competitividade e sustentabilidade, incluindo elevados consumos energéticos em etapas como cominuição, forte variabilidade na qualidade do concentrado e a necessidade de operar em ambientes remotos e severos. Essas características tornam crítica a disponibilidade de informações de processo em tempo real e aumentam a complexidade das estratégias de controle (DIAS, 2023; FERREIRA; FERREIRA, 2025).

Métodos tradicionais de controle, como controladores PID (proporcional, integral e derivativo), apresentam desempenho satisfatório em condições bem definidas. Contudo, em processos minerais sujeitos a mudanças frequentes de regime, atrasos significativos de medição e não linearidades acentuadas, esses métodos podem demandar reajustes de parâmetros constantes e elevada intervenção humana. Como consequência, parte do conhecimento operacional permanece implícita, dependente da experiência dos operadores, o que limita a escalabilidade e a reprodutibilidade do controle.

Mesmo soluções avançadas como o controle *override*, embora eficazes em cenários específicos, não possuem capacidade de aprendizado contínuo. A integração de *soft sensors* com algoritmos adaptativos baseados em aprendizado por reforço surge, portanto, como uma evolução necessária. O RL permite que um agente aprenda políticas de decisão por meio da interação com o ambiente, maximizando uma função de recompensa cumulativa (SUTTON; BARTO et al., 1998). Entre os métodos *model-free* mais difundidos destacam-se o *Q-learning* e o *SARSA*. Enquanto o *Q-learning* estima a função ação-valor ótima de forma *off-policy*, o *SARSA* é um método *on-policy* que atualiza a política com base nas ações efetivamente executadas, o que pode resultar em comportamentos mais conservadores em cenários onde a exploração envolve riscos operacionais (NIAN; LIU; HUANG, 2020).

Apesar dos avanços teóricos acerca do aprendizado por reforço, sua aplicação direta no chão de fábrica esbarra, principalmente, na incompatibilidade estrutural entre os ambientes de desenvolvimento de IA e a arquitetura de *hardware* dos controladores industriais. Os CLPs operam sob normas rígidas de sintaxe e semântica, além de exigir alocação estática de memória, tipagem forte e tempos de varredura determinísticos. Traduzir e implementar matrizes de conhecimento dinâmicas manualmente nesses equipamentos é uma tarefa exaustiva, altamente suscetível a erros humanos de transcrição e praticamente inviável à medida que a dimensionalidade do sistema aumenta.

Diante da dificuldade de transpor o RL para a planta real, a motivação central deste trabalho é atuar como uma etapa base para preencher essa lacuna. O caráter inédito da pesquisa reside justamente na criação de um *framework* de conversão automatizado, atuando como um transpilador capaz de mapear a inteligência treinada em ambiente virtual diretamente para a linguagem ST. Ao superar as restrições de *hardware* e validar o controle de um sistema dinâmico de primeira ordem, é possível evidenciar a viabilidade da execução de agentes inteligentes nativamente no controlador industrial, garantindo segurança, escalabilidade e pavimentando o caminho para adoção de tecnologias mais complexas na Mineração 4.0.

1.3 Objetivos

1.3.1 Objetivo Geral

Desenvolver um *framework* para a implementação de agentes de aprendizado por reforço (*Q-learning*) em sistemas de automação industrial, com foco na execução nativa em controladores lógicos programáveis e viabilizar a portabilidade de políticas treinadas em ambiente simulado (Python) para a linguagem Texto Estruturado.

1.3.2 Objetivos Específicos

Para atingir o objetivo geral, foram definidos os seguintes objetivos específicos:

- Modelar e simular um sistema dinâmico de primeira ordem, presentes em processos de mineração, para atuar como planta de teste controlada, estabelecendo um *baseline* de desempenho com um controlador PI clássico;
- Implementar o algoritmo *Q-learning* com estratégia de exploração ϵ-*greedy* em ambiente computacional, ajustando hiperparâmetros para maximizar a recompensa baseada na minimização do erro de rastreamento;
- Desenvolver um conversor automático capaz de mapear as estruturas de dados dinâmicas do agente (*q-table* e parâmetros) para vetores estáticos e lógica de decisão em Texto Estruturado;
- Validar a equivalência funcional entre a execução da política no ambiente de simulação original e no ambiente de simulação industrial (EcoStruxure Control Expert), verificando a integridade da conversão e a estabilidade do controle.
- Avaliar o desempenho do agente RL implementado no CLP e o controlador PI com a métrica IAE.

1.4 Estrutura do Trabalho

Este trabalho está organizado em cinco capítulos, estruturados da seguinte forma: o Capítulo 2 fundamenta os conceitos de aprendizado por reforço, descreve as características dos sensores virtuais e discute as restrições de *hardware* dos CLPs conforme a norma IEC 61131-3. O Capítulo 3 detalha a metodologia proposta, descrevendo a modelagem do processo, a formulação matemática do agente e a arquitetura do conversor desenvolvido para a geração automática do código. O Capítulo 4 apresenta o resultado da transpilação do algorimto de aprendizado por reforço a um estudo de caso e uma análise comparativa de desempenho entre o controlador clássico e o agente inteligente. Por fim, o Capítulo 5 sintetiza as contribuições do trabalho e apresenta as perspectivas de continuidade.

2 Revisão Bibliográfica

Este capítulo estrutura a base teórica de estudos pertinentes à proposta deste trabalho. Na Seção 2.1, apresenta-se a fundamentação sobre aprendizado por reforço, concentrando-se nos aspectos teóricos e práticos dos algoritmos *model-free Q-Learning* e *SARSA*, bem como nas implicações do *trade-off* entre exploração e exploitação em ambientes industriais variáveis (NIAN; LIU; HUANG, 2020). A Seção 2.2 aborda os conceitos essenciais de sensores virtuais, enfatizando seu papel como estimadores em plantas de beneficiamento mineral e discutindo suas vantagens e limitações operacionais, com base em Ferreira, Braga e Seixas Filho (2010). Na Seção 2.3, descreve-se a pirâmide de automação e a função dos CLPs na mineração, salientando as restrições práticas, como memória limitada, determinismo de ciclo e conformidade com a norma IEC 61131-3, que condicionam a integração de algoritmos de IA em controladores industriais (CARVALHO VALE, 2014). A Seção 2.4 revisa os princípios do controle clássico, com ênfase no controlador PI. Esta revisão fundamenta a escolha do PI como referência de desempenho para comparação com o agente inteligente, apoiando-se em autores como Lourenço (1997). Por fim, a Seção 2.5 sumariza o estudo de Reis (2024), que fornece evidências empíricas sobre a aplicação de RL em usinas de minério de ferro e orienta a metodologia adotada neste trabalho.

2.1 Aprendizado por Reforço - *Q-learning* e *SARSA*

O aprendizado por reforço é uma área do aprendizado de máquina em que um agente aprende a tomar decisões sequenciais interagindo com um ambiente, com o objetivo de maximizar uma função de recompensa cumulativa. Em problemas de controle, essa formulação é particularmente adequada porque traduz objetivos de desempenhos (por exemplo, erro de estimativa, estabilidade, eficiência energética) em sinais de recompensa que guiam a adaptação do agente. Formalmente, muitos problemas tratados em RL são modelados como Processos de Decisão de Markov (MDP), definidos pela quintúpla (X, U, p, R, γ), onde X é o espaço de estados, U o espaço de ações, p a dinâmica de transição de estados, R a função de recompensa e ($\gamma \in [0, 1)$) o fator de desconto para retorno futuro (NIAN; LIU; HUANG, 2020).

O objetivo do agente é aprender uma política π que maximize o retorno esperado definido por:

$$G_t \;=\; \sum_{k=0}^{\infty} \gamma^k R_{t+k+1}, \tag{2.1}$$

para avaliar e comparar políticas, definem-se as funções valor $v^{\pi}(x)$(valor esperado de estar

no estado x seguindo π) e $q^{\pi}(x, a)$(valor de executar a ação a em x e seguir π depois). As equações de Bellman relacionam estas quantidades e constituem a base teórica para os algoritmos de RL (NIAN; LIU; HUANG, 2020).

Entre os métodos *model-free* (que não requerem um modelo explícito do processo) mais utilizados em problemas práticos de controle destacam-se o *Q-learning* e o *SARSA*. Ambos são métodos de diferença temporal (TD) que atualizam estimativas de $Q(x, a)$ a partir de experiências de iteração (estado, ação, recompensa, próximo estado), mas apresentam diferenças fundamentais em relação à forma como tratam a política de exploração.

2.1.1 *Q-learning (off-policy)*

O *Q-learning* é um algoritmo *off-policy* que estima a função ação-valor ótima $q^{*}(x, a)$ independentemente da política de exploração utilizada durante o treinamento. A atualização tabular clássica é:

$$Q_{t+1}(x_t, a_t) \;=\; Q_t(x_t, a_t) + \alpha\Big[\, R_{t+1} + \gamma \max_{a'} Q_t(x_{t+1}, a') - Q_t(x_t, a_t) \,\Big], \tag{2.2}$$

em que α é a taxa de aprendizado, R_{t+1} a recompensa observada e x_{t+1} o estado seguinte. Por usar o termo $\max_{a'} Q_t(x_{t+1}, a')$, o *Q-learning* tende a aprender uma estimativa da melhor ação possível no próximo estado, mesmo que a ação de fato executada durante a exploração seja diferente. Essa propriedade torna o *Q-learning* atraente quando se pretende obter políticas com desempenho próximo ao ótimo, especialmente se for possível pré-treinar em simulador e depois transferir a política para o sistema real (NIAN; LIU; HUANG, 2020).

2.1.2 *SARSA (on-policy)*

SARSA (*State-Action-Reward-State-Action*) é um algoritmo *on-policy* cujo nome deriva da sequência de variáveis que constituem a sua atualização. A regra de atualização é:

$$Q_{t+1}(x_t, a_t) \;=\; Q_t(x_t, a_t) + \alpha\Big[\, R_{t+1} + \gamma Q_t(x_{t+1}, a_{t+1}) - Q_t(x_t, a_t) \,\Big], \tag{2.3}$$

em que a_{t+1} é a ação efetivamente selecionada pela política de exploração no estado x_{t+1} (por exemplo, uma ação escolhida por ϵ-*greedy*). Como resultado, o *SARSA* aprende o valor da política comportamental (a política que foi usada para explorar), o que faz com que a política aprendida incorpore explicitamente o efeito da exploração. Na prática, isso tende a produzir comportamentos mais conservadores em cenários em que a exploração pode conduzir a estados perigosos, característica que é desejável em aplicações industriais sensíveis (NIAN; LIU; HUANG, 2020).

Os autores destacam que, apesar do potencial do RL para produzir políticas adaptativas e de alto desempenho, sua aplicação industrial exige atenção especial à formulação da recompensa, ao balanceamento exploração-exploitação, à validação em simuladores e à garantia de mecanismos de segurança antes da implantação. Esses pontos fundamentam as escolhas metodológicas adotadas neste trabalho e orientam as etapas experimentais descritas nos capítulos seguintes.

2.2 Sensores Virtuais

Soft sensors podem substituir ou complementar analisadores físicos de alto custo ao estimar variáveis de processo não mensuráveis diretamente, gerando estimativas em tempo real e reduzindo investimentos e custos de manutenção. Em operação conjunta com sensores convencionais, fornecem redundância, detecção de falhas e *fallback* automático quando instrumentos apresentam avarias (FERREIRA; BRAGA; SEIXAS FILHO, 2010).

A capacidade de adaptação *online* é uma característica relevante: métodos recursivos e algoritmos adaptativos permitem ajustar parâmetros do *soft sensor* conforme mudam as condições de processo, preservando a acurácia sem interrupção da planta. Contudo, a eficácia dessas abordagens depende da qualidade dos dados, de estratégias de validação e de mecanismos para detectar deriva do modelo.

Além de reduzir latências de análise laboratorial, o que beneficia malhas de controle sensíveis ao tempo, técnicas multivariadas como a Regressão de Mínimos Quadrados Parciais (PLS) oferecem coeficientes interpretáveis que ajudam a identificar variáveis-chave para a inferência. Modelos mais complexos, como redes neurais, podem melhorar desempenho preditivo, porém exigem maior poder computacional e perdem em interpretabilidade. Assim, *soft sensors* atuam também como ferramentas de suporte à decisão e otimização, desde que acompanhados de validação contínua e salvaguardas operacionais (FERREIRA; BRAGA; SEIXAS FILHO, 2010).

2.3 Pirâmide da Automação e o papel dos CLPs na mineração

A automação em plantas de beneficiamento mineral é usualmente representada por uma pirâmide de camadas (1) que organiza as funções e responsabilidades desde a medição de campo até os sistemas de otimização corporativa. Na base da pirâmide encontram-se os sistemas de instrumentação (sensores, transdutores e atuadores), responsáveis por garantir a qualidade e disponibilidade das medições; acima deles estão os controladores (camada dos CLPs), a seguir os sistemas de supervisão (SCADA) e, no topo, os sistemas de otimização e planejamento. Essa hierarquia evidencia que a efetividade de estratégias avançadas depende diretamente da integridade e do projeto das camadas inferiores, em especial da

instrumentação e do controle em tempo real (CARVALHO VALE, 2014).

Figura 1 – Pirâmide de automação industrial aplicada ao beneficiamento de minérios.
Fonte: Adaptado de Carvalho Vale (2014), com base em Queiroz (2011).

No contexto específico da mineração, diversas variáveis críticas (por exemplo: granulometria, densidade de polpa, teor de concentrado) costumam ser obtidas por análises laboratoriais ou equipamentos caros, com latência elevada. Por isso, a integração entre medição de campo confiável e níveis superiores de controle é essencial para reduzir a variabilidade, aumentar recuperação e manter padrões de segurança e meio ambiente. Um bom projeto de automação, pensado em todas as camadas da pirâmide, auxilia na maximização dos rendimentos e na redução de custos operacionais na planta.

Os CLPs ocupam a camada de execução em tempo real da pirâmide e desempenham funções centrais: leitura e condicionamento de sinais, execução de lógicas de intertravamento e segurança, comando de atuadores, geração de alarmes e comunicação com o supervisório. Em usinas de minério, esses sistemas garantem determinismo, disponibilidade e conformidade com normas de segurança funcional, servindo de interface entre os dispositivos de campo e os níveis superiores de automação. Carvalho Vale (2014) descreve detalhadamente essas funções e evidencia como os CLPs viabilizaram a automação massiva das plantas desde a adoção generalizada desses dispositivos.

Ao mesmo tempo, o estudo chama atenção para limitações práticas relevantes ao propor algoritmos avançados embarcados: restrições de memória e processamento, exigência de ciclos determinísticos, limitações das linguagens IEC 61131-3 e a necessidade de um projeto cuidadoso de E/S e rede de campo (Profibus, Foundation Fieldbus, Modbus, etc). Essas restrições implicam que modelos complexos (grandes redes neurais ou bibliotecas pesadas) dificilmente rodarão diretamente no CLP sem adaptações; por isso, soluções viáveis envolvem simplificação/quantização de modelos, pré-treino em simuladores e

implementação de rotinas em formatos compactos (como *q-tables* discretizadas ou funções lineares) para manter o determinismo e a segurança operacional.

Por fim, a autora reforça a importância do projeto, comissionamento e manutenção adequados para o sucesso de qualquer solução de automação: documentação clara, testes em bancada, procedimentos de validação, treinamento de operadores e esquemas de *fallback* são medidas essenciais para mitigar riscos ao introduzir novas camadas de inteligência na pirâmide. Esses pontos fundamentam diretamente as escolhas metodológicas do presente TCC, priorizando o pré-treino em simulador e a validação em ambiente virtual controlado antes de qualquer execução em *hardware*. Essa abordagem visa mitigar riscos operacionais e assegurar a estabilidade do controle, respeitando as diretrizes de segurança da pirâmide de automação.

2.4 Controle Clássico e Controladores PI

O controle clássico constitui a base histórica da automação industrial, sendo amplamente empregado em plantas reais devido à sua simplicidade de implementação, previsibilidade e robustez operacional. Entre as estratégias mais difundidas destacam-se os controladores proporcionais, integrais e derivativos, que continuam sendo a principal solução adotada em controladores industriais.

O controlador proporcional–integral é uma forma simplificada do PID, na qual o termo derivativo é omitido, sendo especialmente indicado para processos em que a ação derivativa pode amplificar ruídos de medição ou não trazer ganhos significativos de desempenho. A lei de controle do PI pode ser expressa por:

$$u(t) = K_p e(t) + K_i \int_0^t e(\tau)\, d\tau, \tag{2.4}$$

em que $e(t)$ representa o erro entre a referência e a variável controlada, K_p é o ganho proporcional e K_i o ganho integral. O termo proporcional atua diretamente sobre o erro instantâneo, enquanto o termo integral tem como principal função eliminar o erro em regime permanente.

O projeto e a sintonia de controladores PI são tradicionalmente realizados a partir de modelos matemáticos do processo, em sua maioria representados por funções de transferência no domínio da frequência. Ferramentas como o Lugar das Raízes permitem analisar graficamente a influência dos ganhos do controlador na posição dos pólos do sistema em malha fechada, fornecendo critérios de estabilidade e desempenho transitório (LOURENÇO, 1997). Essas técnicas assumem, de forma implícita, que o sistema pode ser adequadamente aproximado por um modelo linear e invariante no tempo.

Apesar de sua ampla aplicação, os controladores PI apresentam limitações quando empregados em processos industriais complexos, como aqueles encontrados na mineração. A presença de não linearidades, atrasos de medição, variações de regime operacional e incertezas paramétricas pode comprometer o desempenho de controladores com ganhos fixos, exigindo reajustes frequentes ou intervenção manual de operadores especializados.

Nesse contexto, embora o controle PI continue sendo uma solução eficiente e confiável para uma ampla gama de aplicações, sua dependência de modelos explícitos e de sintonia estática motiva a investigação de estratégias alternativas baseadas em adaptação e aprendizado, como os métodos de aprendizado por reforço. Sob essa perspectiva, a metodologia adotada estabelece o controlador PI como referência clássica de desempenho, permitindo uma comparação objetiva com a abordagem baseada em agentes inteligentes.

2.5 Trabalhos Relacionados: Aprendizado por Reforço no Ajuste de *Soft Sensor* de Nível de Silo para Sequenciamento de *Tripper Car* em Usinas de Minério de Ferro

Reis (2024) aplicou aprendizado por reforço para calibrar um *soft sensor* de nível em silos de uma usina de beneficiamento de minério de ferro, servindo de base para o sequenciamento do *tripper car*. O problema foi estruturado como um Processo de Decisão de Markov, definindo estados pelas faixas de taxa de alimentação do silo e ações pelas vazões de enchimento. Para explorar o espaço de políticas, utilizou-se *Q-learning off-policy* com estratégia ϵ-*greedy*, realizando o ajuste dos hiperparâmetros α (taxa de aprendizado) e γ (fator de desconto) via *Grid Search*.

O algoritmo foi validado em dados reais da usina de Timbopeba da Vale S.A e demonstrou:

- Redução do erro médio absoluto do *soft sensor* de 13,5% para 10,5%;
- Convergência estável da *q-table* para uma política consistente;
- Aumento progressivo da recompensa acumulada e estabilidade após 100 episódios;
- Correspondência clara entre faixas de alimentação e ações de enchimento na *q-table* final.

Este estudo demonstra a eficácia do *Q-learning* para atualização *online* de *soft sensors* em processos sujeitos a desgaste, manutenção e variabilidade operacional. O presente trabalho utiliza os fundamentos teóricos de modelagem de estados e recompensas descritos pelo autor, não para recriar o *soft sensor*, mas para resolver o gargalo de sua implementação em *hardware*.

3 Materiais e Métodos

Este capítulo descreve a metodologia adotada para o desenvolvimento e validação do *framework* proposto para execução de agentes de aprendizado por reforço em controladores lógicos programáveis. São apresentados o modelo do processo utilizado como estudo de caso, o projeto do controlador clássico de referência, a formulação do problema de controle sob a ótica do RL, bem como o fluxo de treinamento, exportação e execução das políticas aprendidas.

3.1 Visão Geral

A metodologia adotada neste trabalho é estruturada em cinco etapas principais:

1. Modelagem de um sistema dinâmico de primeira ordem, utilizado como processo de referência;
2. Projeto de um controlador proporcional-integral, empregado como base comparativa clássica;
3. Formulação do problema de controle utilizando aprendizado por reforço, incluindo definição de estados, ações e função de recompensa;
4. Treinamento do agente em ambiente simulado, seguido da exportação da política aprendida para posterior execução em CLP;
5. Testes e validação do agente exportado para o CLP com avaliação da métrica de desempenho IAE.

Essa abordagem permite separar o treinamento do agente, realizado em ambiente seguro e computacionalmente flexível, da execução em *hardware* industrial, respeitando as restrições típicas de sistemas embarcados.

3.2 Ambiente de desenvolvimento

As simulações computacionais e a ferramenta de conversão de código foram implementadas em linguagem Python. A biblioteca NumPy foi empregada para todas as operações de álgebra linear e armazenamento da *q-table*, garantindo eficiência no processamento vetorial durante o treinamento. Para a geração dos gráficos de resposta temporal e convergência, utilizou-se a biblioteca Matplotlib.

Adicionalmente, o ambiente de desenvolvimento fez uso de bibliotecas padrão do Python, como Importlib e manipuladores de I/O (entrada e saída), para carregar dinamicamente as configurações do agente e gerar os arquivos *.txt* contendo as declarações de variáveis e a lógica de controle em Texto Estruturado. Essa arquitetura permitiu prototipar rapidamente a solução e automatizar a exportação da política para o CLP.

3.3 Modelo do processo

Para validação da metodologia proposta, foi adotado um sistema dinâmico linear de primeira ordem. Esta classe de modelos é amplamente utilizada na engenharia de controle como aproximação robusta para processos industriais que apresentam comportamento monotônico e estável, tais como o nível em tanques de armazenamento, trocadores de calor e sistemas de vazão em tubulações.

No domínio de Laplace, o processo é representado pela função de transferência apresentada na Equação 3.1:

$$G(s) = \frac{Y(s)}{U(s)} = \frac{K}{\tau s + 1}, \tag{3.1}$$

em que $Y(s)$ é a saída do processo, $U(s)$ é a entrada de controle, K representa o ganho estático e τ a constante de tempo do sistema.

No domínio do tempo, o comportamento desse sistema é regido pela equação diferencial linear ordinária de primeira ordem:

$$\tau \frac{dy(t)}{dt} + y(t) = Ku(t). \tag{3.2}$$

Os parâmetros do modelo possuem interpretações físicas diretas que facilitam a análise de desempenho:

- **Ganho Estático (K):** determina a relação entre a variação da entrada e a variação da saída em regime permanente. Um ganho unitário ($K = 1$), por exemplo, implica que a saída tende a igualar a entrada após o regime transiente.
- **Constante de Tempo (τ):** representa a inércia do processo, indicando o tempo necessário para que a resposta atinja aproximadamente 63,2% de sua variação total após uma mudança em degrau na entrada.

A escolha deste modelo visa isolar os efeitos da ação de controle, permitindo uma análise controlada sem a introdução de dinâmicas oscilatórias ou instabilidades naturais de sistemas de ordem superior. Além disso, para a simulação computacional em ambiente

discreto (Python e CLP), o modelo contínuo é discretizado com um período de amostragem T_s adequado, garantindo que a dinâmica digital reproduza o comportamento físico esperado.

3.4 Implementação do controlador PI

Com o objetivo de estabelecer uma referência clássica de desempenho, foi projetado um controlador proporcional-integral para o processo descrito na Subseção 3.3.

A lei de controle adotada é expressa por:

$$u(t) = K_p e(t) + K_i \int_0^t e(\tau) d\tau, \tag{3.3}$$

em que $u(t)$ é o sinal de controle calculado, $e(t)$ representa o erro entre a referência e a saída do processo, enquanto K_p e K_i são, respectivamente, os ganhos proporcional e integral.

Optou-se pela utilização exclusiva da ação PI, omitindo-se o termo derivativo, uma vez que sistemas de primeira ordem não apresentam oscilações naturais. Nesse contexto, a ação derivativa poderia amplificar ruídos de medição sem oferecer ganhos significativos em termos de estabilidade ou velocidade de resposta.

A sintonia dos ganhos K_p e K_i foi realizada utilizando o método analítico de Cancelamento de Pólos. Essa técnica consiste em ajustar o tempo integral do controlador ($T_i = K_p/K_i$) para que seja igual à constante de tempo dominante do processo (τ).

Matematicamente, ao fazer $T_i = \tau$, o zero introduzido pelo controlador PI cancela o pólo estável da planta na função de transferência de malha aberta. Isso simplifica a dinâmica do sistema em malha fechada, transformando-o em um integrador puro com ganho ajustável, o que permite definir a velocidade de resposta desejada diretamente através do ganho proporcional, sem introduzir sobressinais (*overshoot*). Para o processo estudado, onde $\tau = 5.0$s, definiu-se $K_p = 5.0$ para assegurar uma resposta rápida, resultando no ganho integral:

$$K_i = \frac{K_p}{\tau} = \frac{5.0}{5.0} = 1.0. \tag{3.4}$$

Essa estratégia de sintonia foi selecionada por favorecer, teoricamente, uma resposta dinâmica isenta de oscilações e com erro acumulado reduzido. O objetivo é estabelecer um critério de referência rigoroso para a validação comparativa do desempenho do agente de aprendizado por reforço.

3.5 Formulação do problema de controle via Aprendizado por Reforço

Nesta etapa, o problema de controle é modelado como um MDP, onde um agente interage com a planta buscando maximizar uma função de recompensa cumulativa ao longo do tempo.

Como o treinamento foi realizado integralmente em ambiente simulado sem restrições de segurança associadas à exploração, como o risco de danos aos equipamentos reais durante ações aleatórias do agente. Diante disso, optou-se pelo uso do algoritmo *Q-learning* em detrimento ao *SARSA*, visto que sua característica *off-policy* leva o agente a encontrar o caminho analítico mais rápido e eficiente para o controle regulatório.

3.5.1 Definição do estado

O estado do ambiente é definido primariamente a partir do erro de controle $e(k)$, calculado como a diferença entre a referência $r(k)$ e a saída medida $y(k)$ no instante discreto k:

$$e(k) = r(k) - y(k). \tag{3.5}$$

Considerando que CLPs operam com memória finita e que métodos tabulares como o *Q-learning* exigem estados discretos, o espaço contínuo do erro precisa ser mapeado para um conjunto finito de valores. Embora a discretização por faixas fixas seja conceitualmente simples, sua implementação em Texto Estruturado exige cadeias extensas de lógica condicional, o que engessaria o código, prejudicaria o tempo de varredura e tornaria a escalabilidade impraticável caso novas variáveis de estado fossem adicionadas no futuro.

Para superar essa limitação, optou-se por um método de quantização vetorial baseado no vizinho mais próximo. Nesta abordagem, define-se um vetor com N estados protótipos (centróides) $S = \{s_1, s_2, \ldots, s_N\}$. A cada ciclo de controle, o estado corrente é identificado calculando-se a distância Euclidiana entre o valor medido e os protótipos, selecionando-se aquele que minimiza essa distância. Além de garantir um tempo de execução determinístico no controlador, o método permite uma transição suave entre as regiões de controle e facilita enormemente a extensão para problemas multidimensionais.

3.5.2 Definição das ações

As ações disponíveis ao agente foram modeladas como variações incrementais aplicadas ao sinal de controle anterior, respeitando os limites de saturação do atuador no intervalo $[0, 1]$. O conjunto de ações discretas foi definido como:

$$A = \{-0.05, -0.01, 0.0, +0.01, +0.05\}.$$

Essa abordagem confere ao sistema um comportamento integrativo intrínseco, auxiliando na eliminação do erro em regime permanente sem a necessidade de estados adicionais. Além disso, a limitação da magnitude dos incrementos evita variações bruscas no sinal de controle, o que garante uma exploração mais estável do ambiente e preservaria a integridade física de atuadores reais.

3.5.3 Função de recompensa

A função de recompensa é o mecanismo crítico que guia o aprendizado. Para este trabalho, projetou-se uma função limitada e normalizada, focada na minimização do erro absoluto:

$$r(e) = 1 - \min(|e|, 1). \tag{3.6}$$

Essa formulação apresenta características desejáveis para o treinamento:

- A recompensa é mantida no intervalo $[0, 1]$, evitando instabilidade numérica na atualização da *q-table*;
- É máxima ($R = 1$) apenas quando o erro é nulo;
- Erros muito grandes (maiores que a unidade normalizada) recebem recompensa nula, impedindo que "penalidades infinitas"desestabilizem os pesos da rede ou da tabela no início do treinamento.

Embora tenha sido validada neste estudo de caso para um sistema de primeira ordem, sua adaptação para processos com dinâmicas mais complexas pode exigir ajustes e outras opções de formulação, o que é deixado como trabalho futuro.

3.5.4 Algoritmo de Aprendizado por Reforço

Conforme fundamentado na revisão bibliográfica, o treinamento do agente baseou-se no algoritmo *Q-learning*, cuja regra de atualização temporal da função valor de ação foi apresentada na Equação 2.2. Na implementação computacional deste trabalho, a matriz de conhecimento foi otimizada iterativamente, ponderando a incorporação de novas informações pela taxa de aprendizado (α) e a importância das recompensas futuras pelo fator de desconto (γ).

O procedimento de treinamento do agente para o controle da planta é detalhado no Algoritmo 3.1. O algoritmo recebe como entrada os hiperparâmetros de aprendizado

($\alpha, \gamma, \epsilon_{decay}, \epsilon_{min}$) e as configurações de duração do treinamento, como o número de episódios (N_{ep}) e o limite de passos por episódio (N_{steps}). Ao final do processo, a rotina retorna a matriz Q otimizada, que será posteriormente exportada para o conversor.

Na fase de inicialização (linhas 1–2), a *q-table* é criada com valores nulos e a probabilidade de exploração ϵ é definida como 100%, garantindo que o agente explore o ambiente amplamente no início do processo. O laço principal (linha 3) itera sobre o número total de episódios definidos.

No início de cada episódio (linhas 4–6), as variáveis de processo são reiniciadas ($Nivel = 0$) e o estado inicial s é discretizado com base no erro de controle. O laço interno (linha 7) representa a interação passo a passo com o processo. A seleção da ação segue a estratégia ϵ-greedy: com probabilidade ϵ, seleciona-se uma ação aleatória (linha 10) para exploração; caso contrário, seleciona-se a ação de maior valor Q conhecida (linha 12) para exploração do conhecimento adquirido (exploitação).

A ação escolhida, correspondente a um incremento Δu, é integrada ao sinal de controle atual respeitando os limites de saturação (linha 13). Em seguida, a função `SimularProcesso` (linha 14) calcula o novo nível do tanque baseada na dinâmica de primeira ordem discretizada. O novo erro é calculado e convertido para o próximo estado s', e a recompensa r é computada com base na magnitude do erro (linha 17).

A atualização da tabela Q ocorre na linha 18, utilizando a equação de Bellman para ajustar o valor da ação executada com base na recompensa imediata e na melhor estimativa de valor futuro. Por fim, ao término de cada episódio, o parâmetro ϵ é decaído multiplicativamente (linha 22), reduzindo progressivamente a aleatoriedade das ações conforme o agente converge para uma política ótima.

Algorithm 3.1: Treinamento do agente para controle do processo

Entrada: Número de episódios N_{ep}, passos por episódio N_{steps}, taxa de aprendizado α, fator de desconto γ, decaimento ϵ_{decay}, epsilon mínimo ϵ_{min}

Saída : *Q-table* otimizada $Q(s, a)$

Inicializar $Q(s, a)$ com zeros para todos os estados e ações;
$\epsilon \leftarrow 1.0$, $\epsilon_{min} \leftarrow 0.01$;
for $ep \leftarrow 1$ **to** N_{ep} **do**
    $Nivel \leftarrow 0.0$; $u \leftarrow 0.0$;
    $Erro \leftarrow Setpoint - Nivel$;
    $s \leftarrow$ `ObterEstado(`$Erro$`)`;
    for $t \leftarrow 1$ **to** N_{steps} **do**
        Gerar número aleatório $rnd \in [0, 1]$;
        if $rnd < \epsilon$ **then**
            Escolher ação aleatória a no conjunto de ações;
        else
            Escolher ação $a \leftarrow \arg\max_{a'} Q(s, a')$;
        $u \leftarrow \text{clip}(u + \Delta u_a, 0, 1)$;
        $Nivel' \leftarrow$ `SimularProcesso(`$Nivel, u$`)`;
        $Erro' \leftarrow Setpoint - Nivel'$;
        $s' \leftarrow$ `ObterEstado(`$Erro'$`)`;
        $r \leftarrow$ `ObterRecompensa(`$Erro'$`)`;
        $Q(s, a) \leftarrow Q(s, a) + \alpha\,[r + \gamma \max_{a'} Q(s', a') - Q(s, a)]$;
        $s \leftarrow s'$;
        $Nivel \leftarrow Nivel'$;
    if $\epsilon > \epsilon_{min}$ **then**
        $\epsilon \leftarrow \epsilon \times \epsilon_{decay}$;
return $Q(s, a)$;

3.5.5 Treinamento em ambiente simulado

O treinamento do agente foi realizado em ambiente simulado utilizando a linguagem Python, abordagem que viabiliza a rápida prototipagem e o ajuste fino de hiperparâmetros (como α, γ e ϵ) sem os riscos inerentes à operação de equipamentos físicos. Nessa dinâmica, cada episódio de treinamento é executado até atingir um limite fixo de passos ou um critério de convergência pré-estabelecido, de modo que, ao final do aprendizado, a política ótima resultante é congelada para a etapa de exportação, assegurando que o agente transferido para o ambiente industrial apresente um comportamento estável e validado.

3.6 Conversão da política aprendida para Texto Estruturado

Uma vez concluído o treinamento do agente em ambiente Python, torna-se necessário converter a política aprendida para um formato compatível com CLPs industriais. Para viabilizar a execução e atender às limitações desses dispositivos, adaptações específicas e a tradução da lógica devem ser realizadas em conformidade com a norma IEC 61131-3.

3.6.1 Arquitetura e funcionamento do conversor

Após o treinamento, o conhecimento adquirido pelo agente é exportado para um arquivo *.py* que contém a tabela Q, os vetores de estados discretizados, o vetor de ações e os hiperparâmetros do algoritmo. Esse arquivo materializa a política ótima de controle e serve como entrada direta para o conversor. Atuando como um transpilador de domínio específico, a ferramenta processa o módulo inserido e gera, automaticamente, dois arquivos *.txt*:

- **Arquivo de Variáveis:** contém as declarações das variáveis de entrada, saída e internas, servindo como uma espécie de gabarito para que o usuário faça as declarações corretamente no editor de variáveis. O conversor calcula explicitamente as dimensões das matrizes com base no produto cartesiano dos estados e no conjunto de ações, garantindo a alocação estática de memória exigida pelo CLP.
- **Arquivo de Lógica:** contém o corpo do bloco funcional em Texto Estruturado, implementando a lógica de controle e aprendizado.

O Algoritmo 3.2 ilustra como o conversor traduz dinamicamente as dimensões do Python para a sintaxe rígida de declaração de vetores do CLP.

Algorithm 3.2: Geração dinâmica das declarações de variáveis em ST

Entrada : Dimensão do estado (`state_dim`), número de estados (`n_states`), número de ações (`n_actions`)

Saída : String contendo o código em Texto Estruturado

```
st += "VAR_INPUT\n"
st += f"    IN_Estado_Dim : ARRAY[0..{state_dim-1}] OF REAL;\n"
st += "END_VAR\n"
st += "VAR\n"
st += f"    Q_TABLE : ARRAY[0..{n_states-1}, 0..{n_actions-1}] OF REAL;\n"
```

Nesse trecho, observa-se que as dimensões da tabela Q e do vetor de estados são inseridas diretamente no código gerado, assegurando compatibilidade com a tipagem estática e com a alocação prévia de memória exigidas pelos CLPs industriais.

3.6.2 Geração do código Texto Estruturado

A geração do código ST vai além da simples transcrição de uma tabela de consulta. O conversor implementa algoritmos auxiliares essenciais para a autonomia do agente no CLP:

- **Discretização vetorial (L2):** implementa-se um cálculo de distância Euclidiana em tempo real para mapear as entradas analógicas contínuas no estado discreto mais próximo (centróide), permitindo generalização espacial;
- **Gerador de números aleatórios:** como CLPs industriais frequentemente carecem de funções nativas robustas para geração de números aleatórios (necessários para a estratégia de exploração e desempate de ações), o conversor embute no código ST um gerador congruente linear (LCG). Isso assegura que o comportamento estocástico do agente seja reprodutível e computacionalmente eficiente;
- **Atualização de Bellman:** a equação de atualização do *Q-Learning* é transcrita para operações matemáticas nativas, permitindo o refinamento incremental da política diretamente no chão de fábrica.

3.7 Implementação da planta simulada em ambiente de CLP

Para permitir a validação da política de aprendizado por reforço em um ambiente industrial sem a necessidade de uma bancada física, foi implementado diretamente no simulador do controlador Modicon M580, da fabricante Schneider Electric, um modelo dinâmico do processo em Texto Estruturado. Essa abordagem possibilita que a política de controle interaja com uma planta simulada sob as mesmas restrições temporais e estruturais de um controlador real.

O modelo adotado corresponde a um sistema linear de primeira ordem, equivalente ao processo definido na Seção 3.3. A implementação no CLP baseia-se na discretização explícita da equação diferencial contínua, resultando em uma equação em diferenças executada ciclicamente a cada varredura do controlador:

$$Nivel(k+1) = Nivel(k) + \frac{T_s}{\tau}\left(K \cdot u(k) - Nivel(k)\right), \tag{3.7}$$

em que T_s é o tempo de amostragem, τ a constante de tempo do processo, K o ganho estático e $u(k)$ a ação de controle no instante k.

A variável manipulada foi definida como uma válvula normalizada no intervalo [0,1], enquanto a variável de processo representa um nível também normalizado. Essa escolha facilita a comparação com a simulação em Python e reflete práticas comuns em

sistemas industriais. Além disso, a ação de controle foi implementada de forma incremental. A política aprendida não fornece diretamente o valor absoluto de $u(k)$, mas sim uma variação $\Delta u(k)$, a qual é integrada ao longo dos ciclos de execução do CLP:

$$Valvula\_Real := Valvula\_Real + Delta\_Output;$$

Essa estratégia confere ao sistema um comportamento integrativo intrínseco e reproduz fielmente a lógica adotada durante o treinamento do agente em Python. A saturação explícita da variável manipulada no intervalo [0,1] garante a compatibilidade com restrições físicas típicas de atuadores industriais.

3.8 Métrica de desempenho: Integral do Erro Absoluto (IAE)

Para a avaliação comparativa do desempenho dos controladores implementados, foi adotada a métrica Integral do Erro Absoluto. Essa métrica é amplamente utilizada na análise de sistemas de controle por fornecer uma medida quantitativa acumulada do erro entre a variável de processo e o valor de referência ao longo do tempo.

O IAE é definido pela expressão:

$$IAE = \int_0^T |e(t)|\, dt, \tag{3.8}$$

em que $e(t) = r(t) - y(t)$ representa o erro entre o *setpoint* $r(t)$ e a saída do sistema $y(t)$, e T corresponde ao horizonte de tempo analisado.

A escolha do IAE como indicador de desempenho justifica-se por sua capacidade de penalizar o erro de forma uniforme, uma vez que a utilização do valor absoluto considera igualmente os desvios positivos e negativos em relação ao *setpoint*. Além disso, a métrica incorpora a sensibilidade tanto ao regime transitório quanto ao estacionário, contabilizando no valor final da integral os efeitos de oscilações, *overshoot*, tempo de acomodação e erro residual. Essa característica confere uma interpretação intuitiva aos resultados, onde valores menores de IAE evidenciam um melhor desempenho global do controlador, traduzido em maior rapidez e precisão na resposta.

Realizou-se o cálculo do IAE a partir dos dados simulados do controlador PI e dos dados experimentais obtidos da execução do agente de RL no CLP, ambos aplicados à mesma planta de primeira ordem e sob condições equivalentes de referência. Para o cálculo numérico da integral, utilizou-se discretização temporal dos sinais e integração pelo método do trapézio.

Essa abordagem permite uma comparação objetiva entre a estratégia clássica de controle e a abordagem baseada em aprendizado por reforço, complementando a análise

visual das respostas temporais com uma métrica quantitativa consolidada na literatura de controle.

4 Resultados e Discussões

Este capítulo apresenta os resultados da transpilação do algoritmo de aprendizado por reforço a um estudo de caso de um processo de mineração. O algoritmo de aprendizado por reforço foi desenvolvido em Python para o treinamento do agente e posteriormente foi transpilado para a linguagem Texto Estruturado do CLP Modicon M580, da Schneider Electric, com a utilização do *software* EcoStruxure Control Expert, da Schneider Electric.

Para avaliação do procedimento de transpilação do algoritmo de aprendizado por reforço, realizou-se a comparação com um controlador PI. Ambos foram aplicados em um modelo de processo de primeira ordem, presentes em processos de mineração, para uma avaliação quantitativa utilizando a métrica IAE.

4.1 Modelo do processo em Malha Aberta

O processo considerado nos experimentos consiste em um sistema dinâmico de primeira ordem, amplamente utilizado em processos de mineração. O modelo matemático adotado, bem como seus parâmetros, foi apresentado no Capítulo 3.

Para fins de análise dos resultados, utilizou-se um ganho estático unitário e uma constante de tempo positiva, garantindo um comportamento estável e monotônico. O *setpoint* foi fixado em regime unitário, permitindo avaliar de forma clara o desempenho das estratégias de controle em termos de tempo de resposta, erro estacionário e capacidade de rejeição a desvios iniciais.

Antes da aplicação das estratégias de controle, analisou-se o comportamento do processo em malha aberta, conforme diagrama de blocos da Figura 2. A Figura 3 apresenta a resposta do sistema a um degrau unitário na entrada.

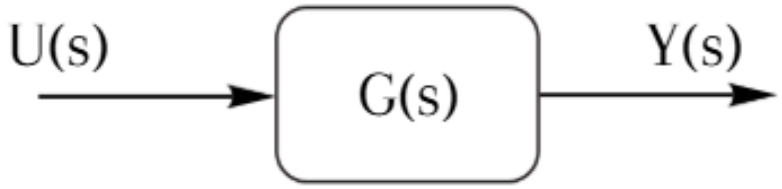

Figura 2 – Diagrama de blocos do sistema em malha aberta.

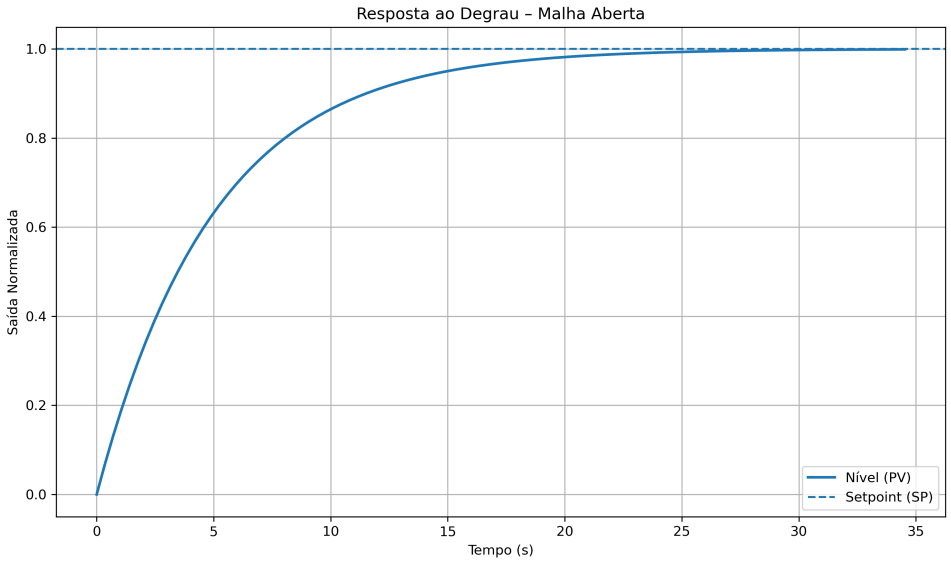

Figura 3 – Resposta ao degrau unitário do sistema em malha aberta.

Observa-se que o sistema apresenta a dinâmica típica de um processo de primeira ordem, caracterizada por uma resposta suave, sem oscilações e com erro estacionário significativo quando submetido a uma entrada constante. Esse comportamento evidencia a necessidade da aplicação de técnicas de controle em malha fechada, de modo a garantir o rastreamento adequado do valor de referência.

A análise em malha aberta fornece, portanto, uma base para a avaliação dos ganhos obtidos com a introdução dos controladores estudados nas seções seguintes.

4.2 Controle do processo com controlador PI

Aplicou-se ao processo um controlador proporcional-integral, sintonizado por meio do método analítico de cancelamento de pólos. Essa escolha visa estabelecer um padrão de desempenho clássico de alta eficiência, explorando a capacidade do PI de eliminar o erro em regime permanente e garantir estabilidade robusta para sistemas de primeira ordem.

A Figura 4 apresenta a resposta temporal do sistema para um degrau unitário no *setpoint*. Observa-se que a estratégia de sintonia adotada resultou em um desempenho superior: o controlador conduziu a saída ao valor de referência de forma rápida e estritamente monotônica, sem apresentar *overshoot* ou oscilações no regime transitório.

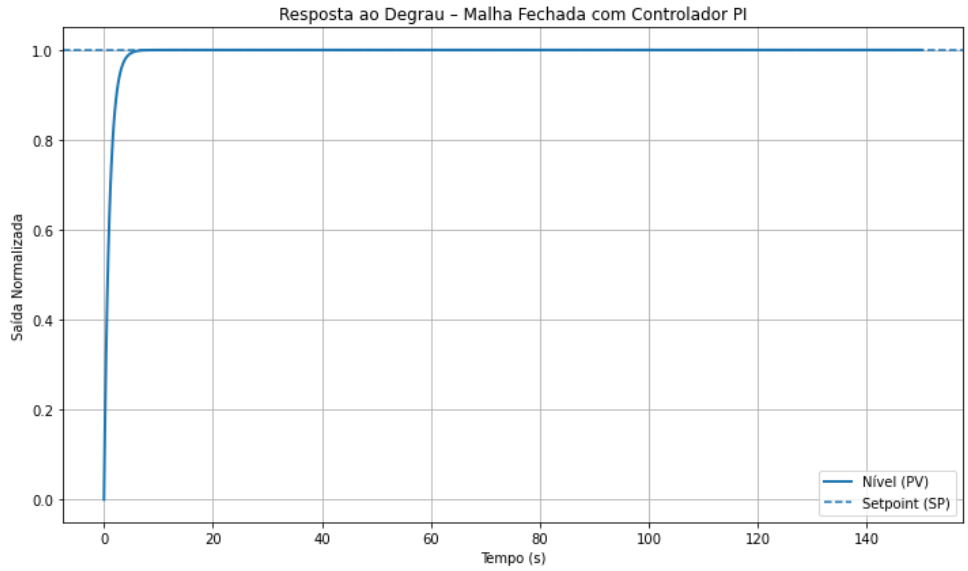

Figura 4 – Resposta ao degrau unitário do sistema em malha fechada com controlador PI.

O erro estacionário foi eliminado por completo e a resposta dinâmica apresentou uma área de erro mínima, restrita essencialmente à inércia natural de subida do processo. Esse comportamento indica que a integral do erro foi reduzida ao limite físico imposto pela constante de tempo da planta, confirmando a eficácia do método de cancelamento de pólos para o modelo nominal.

Apesar desse desempenho excelente no ponto de operação projetado, é importante notar que o controlador PI depende de ganhos fixos (K_p e K_i). Em um cenário industrial real, sujeito a desgastes, não-linearidades ou alterações nos parâmetros da planta (K ou τ), essa resposta ideal tenderia a degradar-se, exigindo reajuste manual. Essa limitação intrínseca dos controladores fixos motiva a investigação de estratégias adaptativas, como o aprendizado por reforço, capazes de ajustar sua política de controle dinamicamente frente a incertezas.

4.3 Controle do processo com aprendizado por reforço

Nesta seção, avalia-se a aplicação do algoritmo de aprendizado por reforço *Q-learning* no controle regulatório do processo de primeira ordem. Enquanto a fundamentação teórica e os detalhes de implementação foram discutidos nos Capítulos 2 e 3, respectivamente, o foco desta etapa reside na análise do desempenho da política obtida e em sua capacidade de estabilização do sistema.

O objetivo principal do agente é sintetizar, por meio da interação contínua com o ambiente, uma política de controle ótima capaz de minimizar o erro de rastreamento entre a variável de processo e a referência. Adicionalmente, o agente deve operar dentro das

restrições operacionais, respeitando os limites de saturação dos atuadores e a dinâmica temporal do sistema.

4.3.1 Formulação do problema

A definição do estado baseada no erro discretizado permitiu ao agente segmentar o espaço de operação em regiões distintas, aprendendo uma resposta específica para diferentes magnitudes de desvio. Com essa segmentação o controlador pode atuar de forma não-linear, aplicando correções agressivas para grandes erros e ajustes finos nas proximidades do *setpoint*.

Um aspecto importante desta formulação reside na escolha das ações como variações incrementais (Δu), abordagem que confere à política aprendida uma característica integral intrínseca: o agente acumula correções no sinal de controle a cada passo de tempo até que o estado de erro nulo seja alcançado e mantido. Isso explica a capacidade do sistema em eliminar o erro em regime permanente sem a necessidade de um termo integrador explícito, como ocorre no PI clássico.

4.3.2 Treinamento e convergência da política

O processo de treinamento no ambiente simulado desenvolvido em Python possibilitou a exploração sistemática do espaço de estados e ações do agente. A adoção da estratégia ϵ-*greedy* permitiu, nas fases iniciais, uma exploração ampla da dinâmica do processo, enquanto a redução gradual do parâmetro ϵ conduziu à convergência para uma política predominantemente determinística, voltada à maximização da recompensa acumulada.

Ao final do treinamento, a tabela Q resultante consolidou-se como uma estrutura de decisão estável, representando um mapeamento aprendido entre o erro do sistema e a ação incremental mais adequada. Diferentemente de uma lei de controle analítica com parâmetros fixos, essa política incorpora implicitamente informações sobre o ganho estático e a constante de tempo do processo, adquiridas por meio da interação contínua com o ambiente simulado.

A Figura 5 apresenta a resposta ao degrau unitário do sistema controlado pela política aprendida.

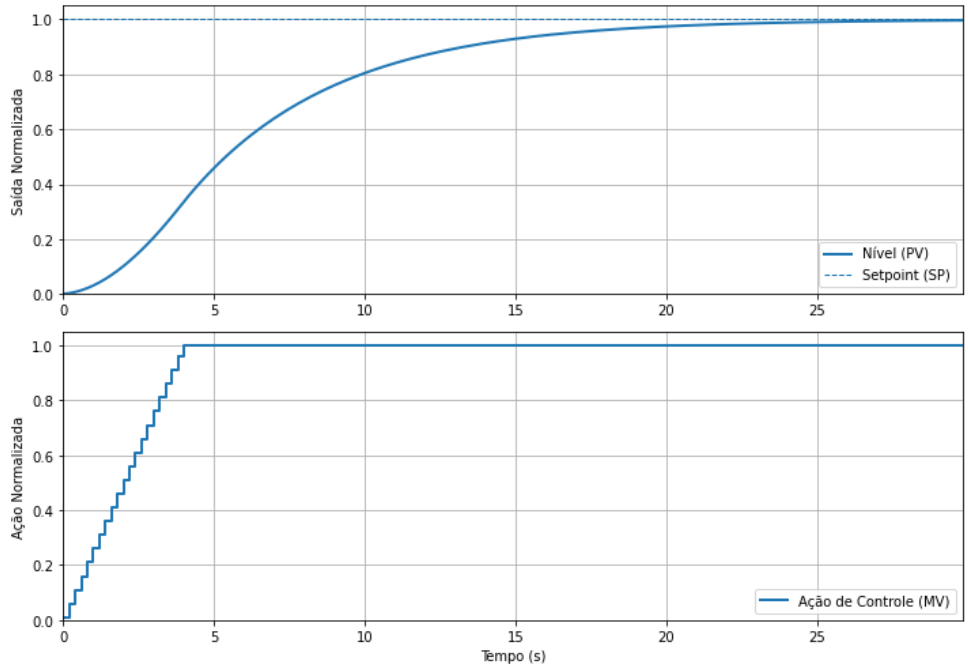

Figura 5 – Resposta do sistema controlado por aprendizado por reforço.

Observa-se que o agente promove uma elevação progressiva da variável de processo, seguida de uma acomodação suave em torno do valor de referência. A ausência de sobressinal e de oscilações significativas indica que a função de recompensa penalizou de forma eficaz variações abruptas no sinal de controle, levando o agente a reduzir gradualmente a intensidade da ação à medida que o erro se aproxima de zero. Esse comportamento resulta em uma resposta monotônica, estável e robusta.

4.4 Análise comparativa das estratégias de controle

Esta seção apresenta uma comparação qualitativa entre o desempenho do controlador PI clássico e o agente baseado em aprendizado por reforço, ambos aplicados ao modelo simulado em ambiente Python. O objetivo desta análise preliminar é validar a eficácia da política aprendida frente a um controlador linear sintonizado analiticamente, demonstrando a viabilidade da abordagem proposta antes de sua conversão para o ambiente industrial.

Diferenças estruturais e de projeto

O controlador PI, sintonizado pelo método de cancelamento de pólos, representa a solução analítica ideal para o modelo linear conhecido. Sua estrutura fixa garante uma resposta rápida e determinística, porém exige conhecimento prévio preciso dos parâmetros da planta (K e τ) para garantir a ausência de oscilações.

Em contrapartida, o controlador baseado em aprendizado por reforço atua sobre o paradigma *model-free*. Isso significa que o algoritmo não utiliza as equações diferenciais ou

os parâmetros físicos do processo para calcular a ação de controle. Sua política é construída empiricamente, baseada apenas na relação de causa e efeito observada durante a interação. Em resumo, o agente RL 'descobre' a dinâmica do processo sozinho.

Comportamento dinâmico

A comparação das respostas temporais evidencia perfis distintos de atuação:

- **Controlador PI:** caracteriza-se por uma atuação proporcional direta ao erro, podendo produzir variações rápidas no sinal de controle dentro dos limites de saturação. Essa característica tende a reduzir o tempo de subida e acelerar a correção do erro, especialmente quando o modelo da planta é bem conhecido. Em condições ideais de simulação, isso resulta em rastreamento preciso da referência com resposta rápida e comportamento previsível.
- **Controlador RL:** apresenta um comportamento mais conservador e suave. Devido à definição das ações como incrementos discretos (Δu), o agente atua de forma progressiva, "construindo"o sinal de controle passo a passo. Isso resulta em um tempo de subida ligeiramente superior, mas confere ao sistema uma robustez natural contra variações bruscas, funcionando como um filtro para a ação de controle.

A Figura 5 demonstra que, mesmo sem conhecer a dinâmica da planta, o agente foi capaz de eliminar o erro em regime permanente e estabilizar o sistema de forma monotônica, comportamento qualitativamente equivalente ao do controlador clássico otimizado.

Justificativa para implementação em CLP

A equivalência de desempenho observada nas simulações valida a tese de que agentes de aprendizado por reforço, mesmo com discretizações simplificadas de estado e ação, são capazes de controlar processos dinâmicos com eficácia comparável a controladores clássicos.

A principal vantagem reside na capacidade adaptativa. Diferentemente do PI, que possui ganhos estáticos, a estrutura do agente RL permite, em tese, a continuidade do aprendizado. Ao manter uma taxa de atualização ($\alpha > 0$) ativa, o controlador preserva a plasticidade necessária para ajustar sua política frente a alterações nos parâmetros da planta ou desgastes de atuadores, característica que será explorada na etapa de validação em ambiente industrial simulado.

4.5 Validação da política de aprendizado por reforço em ambiente de CLP

Com o objetivo de validar a viabilidade da conversão automática da política de controle aprendida para a linguagem Texto Estruturado, a *q-table* obtida ao final do

treinamento em Python foi importada e executada em um ambiente de simulação de CLP industrial, utilizando o *software* EcoStruxure Control Expert, da Schneider Electric. O ambiente permite a programação, configuração e simulação de controladores da família Modicon, garantindo compatibilidade com a norma IEC 61131-3.

A Figura 6 detalha a interface de desenvolvimento, evidenciando a integração do código gerado ao projeto do CLP. Na área central observa-se parte da lógica em Texto Estruturado, especificamente a rotina responsável pela tomada de decisão: o algoritmo identifica o estado atual do processo, consulta a matriz de conhecimento (previamente alocada na memória do controlador como um vetor multidimensional) e seleciona a ação de controle correspondente. Essa visualização confirma a eficácia da conversão automática, demonstrando que a política treinada foi traduzida para instruções nativas do controlador, respeitando a sintaxe e os tipos de dados exigidos pela plataforma.

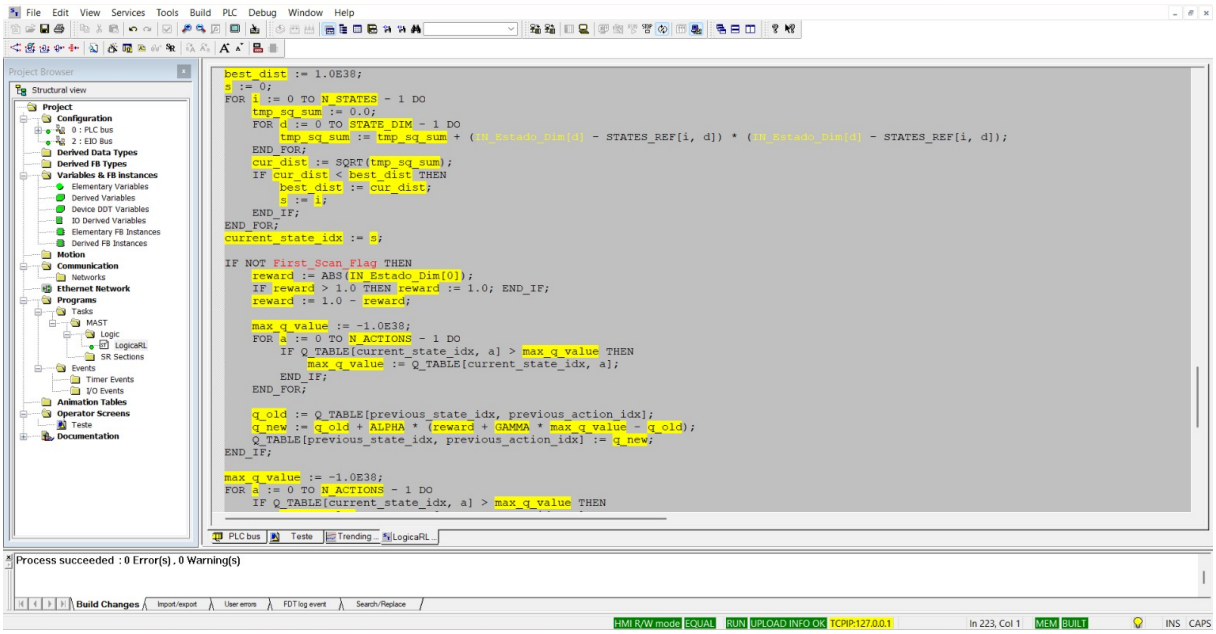

Figura 6 – Trecho da lógica de controle implementada em Texto Estruturado no ambiente de programação EcoStruxure Control Expert.

Para a análise do comportamento dinâmico, os dados das variáveis de processo (PV), *setpoint* (SP) e ação de controle foram coletados durante a execução e processados graficamente, conforme apresentado na Figura 7.

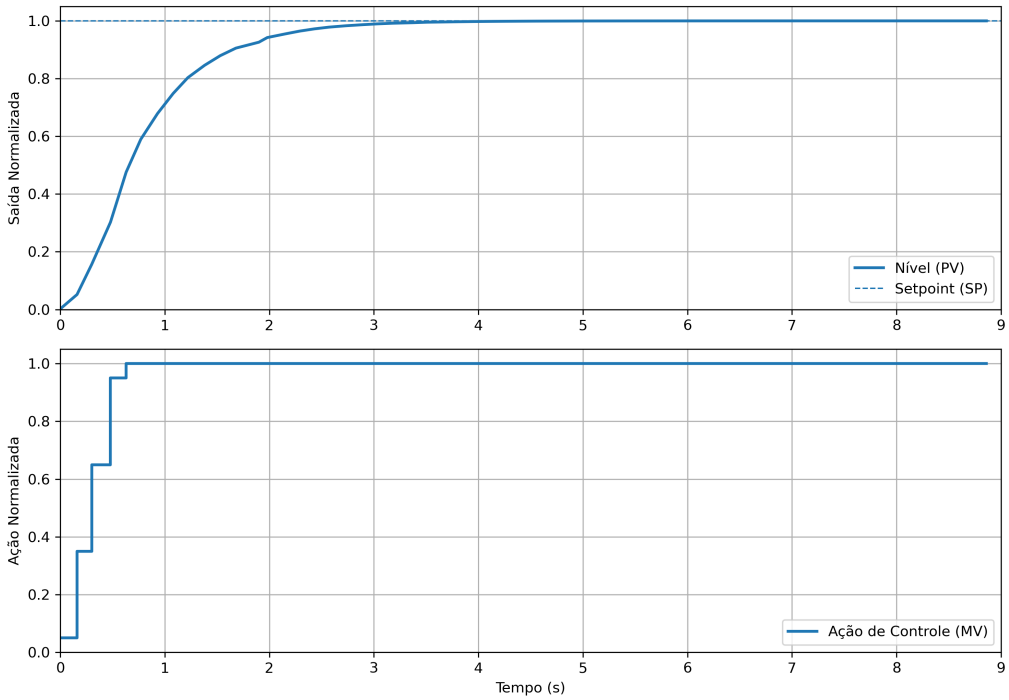

Figura 7 – Resposta do sistema controlado por aprendizado por reforço em ambiente de simulação do CLP no EcoStruxure Control Expert.

A análise visual da curva de resposta confirma a integridade da conversão. O comportamento dinâmico observado no CLP reproduz fielmente as características aprendidas durante o treinamento em Python: a resposta é estável, não apresenta oscilações bruscas e alcança o erro estacionário nulo. A transição suave da variável de processo em direção ao *setpoint* valida a capacidade do código ST gerado de replicar a política de decisão do agente e evidencia que a lógica de discretização de estados e a seleção de ações foram interpretadas e executadas corretamente.

4.5.1 Comparação dos resultados entre as respostas do controlador PI e do aprendizado por reforço

Com o objetivo de realizar uma avaliação quantitativa do desempenho dos controladores, foi adotada como métrica a Integral do Erro Absoluto, conforme metodologia adotada em estudos similares de otimização de controle (REIS et al., 2025). O IAE quantifica o erro acumulado ao longo de todo o período de simulação, penalizando tanto a magnitude dos desvios quanto a duração do regime transitório.

A Tabela 1 apresenta os valores obtidos para o controlador PI clássico e para o agente de aprendizado por reforço executado no ambiente de simulação do CLP.

Tabela 1 – Comparação dos índices de desempenho entre os controladores.

Estratégia de Controle	**IAE**
Controlador PI	1.0033
Agente RL	0.8343

Observa-se uma redução significativa do erro acumulado quando se utiliza o agente baseado em aprendizado por reforço, correspondendo a aproximadamente 63% de redução no IAE em relação ao controlador PI. Esse resultado indica que o agente aprendido foi capaz de conduzir o sistema ao *setpoint* de forma mais rápida e com menor erro transitório.

Esse comportamento pode ser atribuído à natureza adaptativa do aprendizado por reforço. Diferentemente do controlador PI, cuja ação é baseada em uma relação linear fixa entre erro e controle, o agente RL aprende uma política de controle não linear orientada pela maximização de recompensas. Na prática, isso se traduz na aplicação de ações de controle mais intensas no início da resposta, reduzindo rapidamente o erro, seguidas de ajustes mais suaves próximos ao regime permanente, evitando sobressinais e oscilações.

Além disso, a consistência entre os resultados obtidos em Python e os observados na simulação do CLP indica que o processo de conversão da política para Texto Estruturado preservou adequadamente a lógica de controle aprendida, reforçando a viabilidade da abordagem proposta para implementação prática em ambientes industriais.

4.5.2 Viabilidade e limitações

A validação em ambiente de simulação aderente à norma IEC 61131-3 demonstra que a abordagem é compatível com o rigor da automação industrial. Esse resultado evidencia a viabilidade prática de integrar o aprendizado por reforço aos controladores clássicos, configurando-se como um passo intermediário sólido entre a modelagem computacional e a futura implementação em *hardware* real.

O conversor automático desenvolvido desempenha papel central ao viabilizar a transferência de políticas aprendidas em ambientes computacionais flexíveis para plataformas industriais, principalmente no que diz respeito a redução de erros humanos inerentes à transcrição manual de tabelas extensas, dispensando reimplementações e assegurando a fidelidade da lógica de controle. Além disso, a flexibilidade do *framework* permite sua adaptação para diferentes processos e estratégias de discretização, favorecendo a escalabilidade e a padronização da solução, o que reduz o esforço de engenharia.

Do ponto de vista do algoritmo, a natureza *model-free* do RL minimiza a dependência de representações matemáticas precisas da planta, característica que é particularmente vantajosa em processos sujeitos a incertezas, não linearidades e variações operacionais.

Por outro lado, algumas limitações próprias à arquitetura dos controladores e à metodologia devem ser consideradas. A utilização de *q-tables* discretizadas pode implicar um aumento exponencial no consumo de memória à medida que cresce a dimensionalidade do espaço de estados e ações, o que pode esbarrar nas restrições físicas dos CLPs. Adicionalmente, a dinâmica de execução cíclica (*scan time*) impõe restrições ao aprendizado *online* em tempo real, exigindo cautela na definição das taxas de atualização para garantir a estabilidade do sistema em operação contínua.

Por fim, ressalta-se que a validação apresentada baseou-se em um modelo simulado de primeira ordem e não englobou ensaios em bancada física. Embora adequado como estudo de caso inicial para comprovar a eficácia da conversão, a aplicação em processos industriais reais, potencialmente mais complexos, multivariáveis e sujeitos a perturbações externas, constitui uma etapa futura essencial para consolidar a robustez da metodologia proposta.

5 Considerações Finais

Este trabalho teve como objetivo principal investigar a viabilidade da aplicação de técnicas de aprendizado por reforço em sistemas de automação industrial, com ênfase na execução de políticas de controle em controladores lógicos programáveis. Para isso, propôs-se um *framework* que integra o treinamento de um agente em ambiente simulado, utilizando a linguagem Python, à conversão automática da política aprendida para Texto Estruturado, linguagem padronizada pela norma IEC 61131-3.

Inicialmente, o comportamento dinâmico do processo foi analisado em malha aberta, permitindo a compreensão de suas características fundamentais e servindo como base para o projeto dos controladores. Em seguida, foram implementadas e avaliadas duas estratégias de controle: um controlador PI clássico, utilizado como referência, e um controlador baseado em aprendizado por reforço, treinado por meio do algoritmo *Q-learning* incremental. Os resultados quantitativos reforçam a eficácia da abordagem proposta. A análise baseada na Integral do Erro Absoluto (IAE) indicou que o agente de aprendizado por reforço obteve uma redução aproximada de 16,8% no erro acumulado em comparação ao controlador PI clássico, evidenciando que a política convertida não apenas reproduz a lógica aprendida, mas proporciona ganhos efetivos de desempenho ao explorar estratégias de controle não lineares.

Além da validação técnica por meio do controle realizado, ressalta-se o impacto prático e os benefícios diretos gerados pelo conversor desenvolvido. A automação do processo de transpilação da política de aprendizado do agente e de sua lógica de decisão para a linguagem ST elimina por completo o risco de falhas humanas inerentes à transcrição manual de dados. Mais do que uma tradução de sintaxe, o *framework* assegura a conformidade com as restrições impostas pela norma IEC 61131-3, gerenciando de forma automatizada a declaração de variáveis e a alocação estática da memória do controlador. Este conversor é uma infraestrutura inovadora, verificável e escalável, que facilita a introdução do aprendizado por reforço em controladores industriais e consolida uma base para que aplicações industriais mais complexas possam ser embarcadas diretamente no chão de fábrica.

Tendo em vista que o objetivo central desta pesquisa consistiu na construção e validação funcional dessa infraestrutura de conversão, a aplicação do método em um sistema de primeira ordem em ambiente simulado cumpriu plenamente seu papel como prova de conceito. Com essa base tecnológica consolidada e validada, abrem-se amplas perspectivas para a evolução contínua da ferramenta.

5.1 Trabalhos Futuros

Como continuidade, diversas extensões podem ser exploradas de modo a ampliar a aplicabilidade e a robustez do *framework* proposto. Uma possibilidade imediata consiste na adaptação da abordagem para outros algoritmos de aprendizado por reforço, como o método *SARSA*. Sua característica *on-policy* é particularmente relevante em ambientes industriais, nos quais a política executada durante o aprendizado deve respeitar restrições operacionais e de segurança, tornando essa investigação futura um caminho promissor.

Outra linha de pesquisa promissora consiste na aplicação do *framework* a processos dinâmicos mais complexos, como sistemas de ordem superior, plantas não lineares ou sistemas multivariáveis. Naturalmente, essa transição exigirá o estudo e a implementação de diferentes funções de recompensa, visto que neste trabalho essa métrica foi desenvolvida de forma específica para otimizar o controle do sistema de primeira ordem. A adoção de múltiplas funções, bem como a seleção automática ou adaptativa para novos cenários, demanda um estudo mais aprofundado, pois tais mudanças impactam diretamente a modelagem do espaço de estados, o conjunto de ações e o comportamento de convergência do agente.

Como aplicação direta no contexto da Mineração 4.0, propõe-se a integração embarcada de *soft sensors* adaptativos baseados em RL. Utilizando o conversor validado como infraestrutura de base, torna-se possível transpor soluções desenvolvidas em simuladores de alto nível para execução nativa nos controladores industriais. Essa frente de pesquisa permitiria que estimações de variáveis complexas em tempo real operem respeitando a alocação estática de memória. Além disso, a capacidade do agente de aprender continuamente com as interações do processo viabilizaria a recalibração autônoma do *soft sensor*, garantindo o ajuste a desgastes físicos e flutuações operacionais sem a necessidade de intervenção humana, unindo, assim, a adaptabilidade da inteligência artificial à robustez do controle industrial.

Por fim, uma etapa fundamental de validação experimental consiste na implementação do controlador baseado em RL em um CLP físico, com testes em bancada ou em ambiente industrial real. Essa etapa permitiria avaliar o desempenho do método frente a ruídos, atrasos, limitações de *hardware* e outras incertezas não modeladas, consolidando sua aplicabilidade prática.

Referências

ALBUQUERQUE, Kaike S; FONSECA, Alexandre G; DUARTE, Robson A; FINA, Rodrigo; PINTO, Thomás VB; EUZÉBIO, Thiago AM. Controle Override de corrente elétrica para o Aumento de Eficiência do Peneiramento no Processamento Mineral. In: 2. SIMPÓSIO Brasileiro de Automação Inteligente-SBAI. 2023. v. 1. DOI: 10.20906/SBAI-SBSE-2023/3944. Citado 1 vez na página 11.

CARVALHO VALE, Júlia Maria de. Estratégias de controle no processamento de minério de ferro. Universidade Federal de Minas Gerais, 2014. Acesso em: 09 ago. 2025. Disponível em: https://hdl.handle.net/1843/VRNS-9QBMC7. Citado 3 vezes nas páginas 15, 18.

DIAS, Amanda Ribeiro Lutterback. Mineração 4.0: a evolução e os benefícios da indústria 4.0 no setor de mineração. Universidade Federal do Rio de Janeiro, 2023. Disponível em: http://hdl.handle.net/11422/21146. Citado 4 vezes nas páginas 10, 12.

FERREIRA, Grasyelle Maria Mota; GONTIJO, Hugo M; MONTALVÃO, Marcelo; ALVES, Fernanda Andrade; SOUZA, Janine Cerqueira de Oliveira; MAGALHÃES, Marcus; FILGUEIRA, Adrielle Cintra; FERNANDES, Tatiane A. Automação do circuito de flotação de CDSII e seus desafios, 2022. Disponível em: https://proceedings.science/entmme-2022/trabalhos/automacao-do-circuito-de-flotacao-de-cdsii-e-seus-desafios?lang=pt-br. Citado 2 vezes na página 10.

FERREIRA, Jamille Souza; FERREIRA, Kelly Cristina. HPGR NA COMINUIÇÃO MINERAL: AVANÇOS TECNOLÓGICOS, DESAFIOS OPERACIONAIS E CRITÉRIOS DE SELEÇÃO. *REVISTA FOCO*, v. 18, n. 8, e9589–e9589, 2025. DOI: https://doi.org/10.54751/revistafoco.v18n8-142. Citado 1 vez na página 12.

FERREIRA, Tiago Paulinelli; BRAGA, Carmela Maria Polito; SEIXAS FILHO, Constantino. Sensores Virtuais–Soft Sensors. *Universidade Federal de Minas Gerais*, 2010. Disponível em: https://www.researchgate.net/profile/Carmela-Polito-Braga/publication/281305286_Sensores_Virtuais_-_Soft_Sensors/links/55e1239608aecb1a7cc61d24/Sensores-Virtuais-Soft-Sensors.pdf. Citado 4 vezes nas páginas 11, 15, 17.

LOURENÇO, João. Sintonia de controladores PID. *Escola superior de tecnologia*, 1997. Acesso em: 20 set. 2025. Disponível em: http://alvarestech.com/temp/smar/2019/PID-LugarDasRaizes-Matlab-Tutorial.pdf. Citado 2 vezes nas páginas 15, 19.

NIAN, Rui; LIU, Jinfeng; HUANG, Biao. A review on reinforcement learning: Introduction and applications in industrial process control. *Computers & Chemical Engineering*, Elsevier, v. 139, p. 106886, 2020. DOI: https://doi.org/10.1016/j.compchemeng.2020.106886. Citado 6 vezes nas páginas 12, 15, 16.

QUEIROZ, Júlio César Braz de. *Integração de Sistemas*. 2011. Revisão 2. Citado 0 vez na página 18.

REIS, L. A. *Aprendizado por Reforço no Ajuste de Soft Sensor de Nível de Silo para o Sequenciamento de Tripper Car em Usinas de Minério de Ferro*. 2024. Monografia, Disciplina de Aprendizado por Reforço, Universidade Federal de Ouro Preto. Citado 2 vezes nas páginas 15, 20.

REIS, Lucas A.; ASSIS, Jonas A.; LÚCIO, Leonardo D.; REIS, Júlia S. Enhancing Crushing Productivity Using the Smith Predictor-Based Control. *IFAC PapersOnLine*, Elsevier, v. 59, n. 32, p. 24–29, 2025. Conference Paper Archive. ISSN 2405-8963. DOI: 10.1016/j.ifacol.2025.12.391. Citado 1 vez na página 39.

SUTTON, Richard S; BARTO, Andrew G et al. *Reinforcement learning: An introduction*. MIT press Cambridge, 1998. v. 1. Disponível em: http://incompleteideas.net/book/the-book-1st.html. Citado 1 vez na página 12.