Programa de Pós-Graduação em Instrumentação, Controle e Automação de Processos de Mineração (PROFICAM)
Escola de Minas, Universidade Federal de Ouro Preto (UFOP)
Associação Instituto Tecnológico Vale (ITV)

Dissertação

CONTROLE POR APRENDIZAGEM POR REFORÇO APLICADO AOS PROCESSOS: CSTR E ESPESSADOR

Santino Martins Bitarães

Ouro Preto
Minas Gerais, Brasil
2022

Santino Martins Bitarães

CONTROLE POR APRENDIZAGEM POR REFORÇO APLICADO AOS PROCESSOS: CSTR E ESPESSADOR

Dissertação apresentada ao Programa de Pós-Graduação em Instrumentação, Controle e Automação de Processos de Mineração da Universidade Federal de Ouro Preto e do Instituto Tecnológico Vale, como parte dos requisitos para obtenção do título de Mestre em Engenharia de Controle e Automação.

Orientador: Prof. Thiago Antonio Melo Euzebio, D.Sc.
Coorientador: Prof. Moisés Tavares da Silva, D.Sc.

Ouro Preto
2022

SISBIN - SISTEMA DE BIBLIOTECAS E INFORMAÇÃO

B624c Bitaraes, Santino Martins.
Controle por aprendizagem por reforço aplicado aos processos [manuscrito]: CSTR e espessador. / Santino Martins Bitaraes. - 2022.
50 f.: il.: color., gráf., tab..

Orientador: Dr. Thiago Euzebio.
Coorientador: Dr. Moisés Silva.
Dissertação (Mestrado Profissional). Universidade Federal de Ouro Preto. Programa de Mestrado Profissional em Instrumentação, Controle e Automação de Processos de Mineração. Programa de Pós-Graduação em Instrumentação, Controle e Automação de Processos de Mineração.
Área de Concentração: Engenharia de Controle e Automação de Processos Minerais.

1. Minas e Mineração. 2. Inteligência Artificial. 3. Controle automático - Controle Avançado. 4. Aprendizagem - Aprendizagem por reforço. 5. Reator de Tanque com Agitação Contínua (CSTR). 6. Minas e recursos minerais - Espessador. I. Euzebio, Thiago. II. Silva, Moisés. III. Universidade Federal de Ouro Preto. IV. Título.

CDU 681.5:622.2

Bibliotecário(a) Responsável: Maristela Sanches Lima Mesquita - CRB-1716

MINISTÉRIO DA EDUCAÇÃO
UNIVERSIDADE FEDERAL DE OURO PRETO
REITORIA
ESCOLA DE MINAS
PROGR. POS GRAD. PROF. INST. CONT. E AUT.
PROCESSOS DE MIN.

FOLHA DE APROVAÇÃO

Santino Martins Bitarães

Controle por aprendizagem por reforço aplicado aos processos: CSTR e Espessador

Dissertação apresentada ao Programa de Pós-Graduação em Instrumentação, Controle e Automação de Processos de Mineração (PROFICAM), Convênio Universidade Federal de Ouro Preto/Associação Instituto Tecnológico Vale - UFOP/ITV, como requisito parcial para obtenção do título de Mestre em Engenharia de Controle e Automação na área de concentração em Instrumentação, Controle e Automação de Processos de Mineração.

Aprovada em 06 de setembro de 2022

Membros da banca
Doutor - Thiago Antonio Melo Euzébio - Orientador - Helmholtz-Zentrum Dresden Rossendorf
Doutor - Moisés Tavares da Silva - Instituto Tecnológico Vale
Doutor - Márcio Feliciano Braga - Universidade Federal de Ouro Preto
Doutor - Luciano Perdigão Cota - Instituto Tecnológico Vale
Doutor - José Mário Araújo - Instituto Federal da Bahia

Thiago Antonio Melo Euzébio, orientador do trabalho, aprovou a versão final e autorizou seu depósito no Repositório Institucional da UFOP em 07/10/2022

Documento assinado eletronicamente por **Bruno Nazário Coelho**, **COORDENADOR(A) DE CURSO DE PÓS-GRADUAÇÃO EM INST. CONTROLE AUTOMAÇÃO DE PROCESSOS DE MINERAÇÃO**, em 19/10/2022, às 14:16, 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 **0412181** e o código CRC **E36988E4**.

Referência: Caso responda este documento, indicar expressamente o Processo nº 23109.014179/2022-93 SEI nº 0412181

R. Diogo de Vasconcelos, 122, - Bairro Pilar Ouro Preto/MG, CEP 35400-000
Telefone: (31)3552-7352 - www.ufop.br

Dedico esse trabalho a Jesus Cristo, ao meu pai Santino (in memoriam), a minha mãe Benvinda (in memoriam), a minha esposa Priscila e a minha filha Martina.

Agradecimentos

Agradeço a Deus pela vida, pela saúde e sabedoria que me foi dada pela graça, para produzir esse trabalho. Agradeço ao meu pai, Santino (*in memoriam*), a minha mãe Benvinda (*in memoriam*) que se preocuparam em me garantir um suporte financeiro para que eu conseguisse chegar até aqui. No tempo deles, sempre me incentivaram e acreditaram nos meus sonhos, fizeram dos meus sonhos os sonhos deles. Sou grato à minha esposa Priscila, que tem estado ao meu lado desde a graduação, sempre me dando suporte, conselhos e apoio. Agradeço à minha filha Martina pelos sorrisos e olhares que conseguem transformar qualquer dia exaustivo num domingo ensolarado. Ao meu orientador pela oportunidade que deu desde a minha graduação de trabalhar em projetos de pesquisa e desenvolvimento. Agradeço-lhe também por ser um líder que sempre acreditou e incentivou-me a alçar voos mais altos, um deles, esse trabalho. Também sou grato pelo apoio durante situações difíceis que passei nessa trajetória. Agradeço também ao meu coorientador que com paciência e sabedoria deu todo suporte para o desenvolvimento desse trabalho. Agradeço ao Wellington Teixeira pelo apoio dado na utilização do simulador do espessador. Sou grato aos meus colegas de turma pelos aprendizados compartilhados durante o tempo de convivência.

O presente trabalho foi realizado com apoio da Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - Brasil (CAPES) - Código de Financiamento 001; do Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPQ), números 402759/2018-4 e 444425/2018-7; do Instituto Tecnológico Vale (ITV) e da Universidade Federal de Ouro Preto (UFOP).

"Quem, pois, conheceu a mente do Senhor? Ou quem foi o seu conselheiro? Ou quem primeiro deu a ele para que lhe venha a ser restituído? Porque dele, e por meio dele, e para ele são todas as coisas. A ele, pois, a glória eternamente. Amém!" Romanos 11:34-36 ARA

Resumo

Resumo da Dissertação apresentada ao Programa de Pós Graduação em Instrumentação, Controle e Automação de Processos de Mineração como parte dos requisitos necessários para a obtenção do grau de Mestre em Ciências (M.Sc.)

CONTROLE POR APRENDIZAGEM POR REFORÇO APLICADO AOS PROCESSOS: CSTR E ESPESSADOR

Santino Martins Bitarães

Setembro/2022

Orientadores: Thiago Antonio Melo Euzebio
Moisés Tavares da Silva

O controle por aprendizagem por reforço busca melhorar seu desempenho pelo aprendizado obtido ao interagir com o processo. As ações de controle deste tipo de controlador são norteadas unicamente por uma função de recompensa. O algoritmo *Augmented Random Search* (ARS) é uma método de aprendizagem por reforço baseado em busca aleatória simples com melhorias no processamento das recompensas e dos estados. As características apresentadas pela aprendizagem por reforço permitirá sua utilização em processos complexos e não lineares, como o tanque com agitação contínua (CSTR) e o espessador. Esses dois processos são complexos e apresentam comportamentos diferentes nos pontos de operação. Para o problema do CSTR, os estados são as referências do processo (referência atual e uma mudança de referência), as ações são os parâmetros do controlador PI e a recompensa foi definida em função do erro entre a referência e variável do processo (temperatura do reator). No caso do espessador os estados são o erro e a concentração do *underflow*, a ação é o ajuste direto da vazão de *underflow* e a função de recompensa foi definida em função do erro e da variação da ação de controle. Para o simulador do CSTR foi utilizado o *python* e para o espessador, utilizamos o *Matlab*. A sintonia proposta pelo ARS para o problema do CSTR apresenta uma melhoria de 8,3% (IAE), considerando o mesmo ponto de operação, em comparação com o *benchmark*. Já o o algoritmo ARS foi 19% (IAE) melhor na tarefa de controlar diretamente o espessador.

Palavras-chave: Mineração, Inteligência Artificial, Controle Avançado, Aprendizagem por reforço, Reator tanque com agitação contínua e Espessador.

Macrotema: Usina; **Linha de Pesquisa:** Análise e Projeto de Sistemas de Controle Avançado; **Tema:** Redução de Variabilidade e Melhoria de Controle.

Abstract

Abstract of Dissertation presented to the Graduate Program on Instrumentation, Control and Automation of Mining Process as a partial fulfillment of the requirements for the degree of Master of Science (M.Sc.)

REINFORCEMENT LEARNING CONTROL APPLIED TO NON-LINEAR/CONTINUOUS SYSTEMS: CSTR AND THICKENER

Santino Martins Bitarães

September/2022

Advisors: Thiago Antonio Melo Euzebio
Moisés Tavares da Silva

Control by reinforcement learning seeks to improve its performance by learning by interacting with the process. The control actions of this type of controller are guided solely by a reward function. The Augmented Random Search (ARS) algorithm is a reinforcement learning method based on a simple random search with improvements in processing rewards and states. The characteristics of reinforcement learning will allow its use in complex and non-linear processes, such as the continuous agitation tank (CSTR) and the thickener. These two processes are complex and exhibit different behaviors at operating points. For the CSTR problem, the states are the process references (current setpoint and a change of setpoint), the actions are the parameters of the PI controller, and the reward is defined as a function of the error between the reference and the process variable (temperature of the reactor). In the case of the thickener, the states are the error and the underflow concentration, the action is the direct adjustment of the underflow flow, and the reward function is defined as a function of the error and the variation of the control action. For the CSTR simulator, we used python, and for the thickener, we used Matlab. The tuning proposed by the ARS for the CSTR problem presents an improvement of 8.3% (IAE), considering the same operating point, compared to the benchmark. The ARS algorithm was 19% (IAE) better in directly controlling the thickener.

Keywords: Mining, Artificial Intelligence, Advanced Control, Reinforcement Learning, Continuous stirring tank reactor and Thickener .

Macrotheme: Plant; **Research Line:** Advanced Control Systems Analysis and Design; **Theme:** Variability Reduction and Control Improvement;

Lista de Figuras

Lista de Tabelas

Lista de Siglas e Abreviaturas

ARS *Augmented Random Search*

CARLA *Continuous Action Reiforcement Learning Automata*

CLP Controlador Lógico Programável

CSTR *Continuous Stirred-Tank Reactor*

DRL *Deep Reinforcement Learning*

ER Repetição de Experiência

ES Estratégia de Evolução

IAE *Integral Absolute Error*

MIMO *Multiple Input Multiple Output*

MPC *Model Predictive Control*

PI *Proporcional Integral*

PID *Proporcional Integral Derivativo*

SARSA *State Action Reward State Action*

SII Soma dos incrementos absolutos da entrada

SISO *Single Input Single Output*

Sumário

1. Introdução

Nesse capítulo é feita uma introdução dos conceitos que serão desenvolvidos no decorrer dos demais capítulos. Na seção 1.1 será apresentado uma contextualização do tema proposto, na seção 1.2 será apresentado a motivação desse trabalho na seção 1.3 são apresentados os objetivos propostos e por fim na seção 1.4 algumas perguntas são expostas.

1.1. Contexto

A aprendizagem por reforço é uma área que vem apresentando grande relevância no meio científico. Esta técnica é uma área de pesquisa ativa dentro da inteligência artificial. A sua origem é dada na pesquisa operacional e na ciência da computação para resolver problemas de tomada de decisão sequencial. Trata-se de uma técnica que mapeia ações para situações de modo a obter uma maior recompensa do ambiente. Inicialmente, o algoritmo não possui instruções de quais ações tomar dadas as situações. Dessa forma, a técnica de aprendizagem por reforço deverá descobrir quais são as melhores ações que geram maior recompensa, após experimentá-las (SPIELBERG *et al.*, 2019; SUTTON e BARTO, 2018).

De acordo com Wiering e van Otterlo (2012), a aprendizagem por reforço é uma classe geral de algoritmos de aprendizado de máquina que visa fazer com que o algoritmo tome ações em um ambiente onde o único retorno consiste em um sinal de recompensa escalar. O objetivo do agente é realizar ações que maximizem o sinal de recompensa ao longo prazo.

A característica de aprendizado contínuo da aprendizagem por reforço é aplicável nas estratégias de controle, pois na maioria dos casos deseja-se uma estratégia que se adapte a condições ainda não previstas pelo projeto de controle. Sendo assim diversos trabalhos na literatura aplicaram algoritmos de aprendizagem por reforço para sintonia de controladores Proporcional-Integral-Derivativo (PID). Em Howell e Best (2000), é automatizado o ajuste de um motor Zetec da *Ford Motors*. O algoritmo, denominado *Continuous Action Reinforcement Learning Automata* (CARLA), foi usado para ajustar controladores PIDs após os parâmetros iniciais serem definidos usando métodos de sintonia clássicos, como *Ziegler-Nichols* proposto em Ziegler *et al.* (1942). Os resultados mostraram uma redução de 60% na função custo após o ajuste da aprendizagem por reforço. Por outro lado, um algoritmo de aprendizagem por reforço chamado de Ator-critico foi usado por Wang *et al.* (2007) para ajustar os parâmetros PID de maneira adaptativa em um sistema complexo e não linear. Quando comparado com um controlador PID convencional, os resultados da simulação mostram que o controlador proposto é eficiente para sistemas não lineares complexos, adaptável e robusto.

O trabalho de Koszaka *et al.* (2006) utiliza uma abordagem de aprendizado por reforço baseado em *Q-learning* para ajuste adaptativo de um controlador PID. Os processos controlados foram simulados como funções de transferência no Matlab. Foi demonstrado que, para uma função de recompensa simples, o algoritmo proposto forneceu soluções satisfatórias e manteve

o sobressinal e erro de estado estacionário sob restrições estabelecidas.

Em Brujeni *et al.* (2010) foi usado o algoritmo SARSA para ajustar dinamicamente um controlador PI usado para controlar o processo de um reator tanque com agitação contínua (CSTR). O agente foi primeiro pré-treinado em um modelo estimado para o processo e, em seguida, foi implementado para ajustar continuamente o aquecedor do tanque *on-line*. O agente teve como objetivo rejeitar distúrbios e rastrear o ponto de operação. Ao final, os autores compararam o desempenho do controlador sintonizado pelo algoritmo SARSA com os métodos de ajuste de controle do modelo interno. Verificou-se que o algoritmo SARSA foi o método de ajuste superior devido à sua natureza adaptativa contínua. Em contraste, em Reddy *et al.* (2020) foi proposto um estimador de modelo polinomial recursivo do controlador PI. A proposta foi projetar o controlador para o reator químico simulado com objetivo de manter o pH do produto e nível de reagente. Os resultados da simulação mostram um bom desempenho do controlador PI adaptativo.

O desafio de controlar o processo CSTR é abordado por diversos trabalhos na literatura. Em Huang *et al.* (1982) é proposto um sistema de controle adaptativo para o controle do processo CSTR. O sistema é projetado para se adaptar às mudanças de parâmetros devido à variação das condições de operação e também às mudanças de concentração não mensuráveis na entrada. A adaptação dos parâmetros foi feita por sensibilidades no estado estacionário. Através dos experimentos realizados em um reator tanque agitado parcialmente simulado, o autor apresentou resultados do controle adaptativo proposto para o processo CSTR bastante satisfatórios. Na literatura são encontrados diversos trabalhos que utilizam várias estratégias de controle para controlar o CSR. O trabalho de Kumar *et al.* (2020) apresenta dois controladores: controle PID e controle preditivo baseado em modelo (MPC). O controle efetivo da temperatura foi obtido pelo uso do modelo de controle preditivo em termos de estabilidade, eliminando o efeito de perturbação e rastreamento de referência.

Outro processo que é complexo e não linear é o espessador. Estes por sua vez possuem grande relevância na mineração, pois seu uso está relacionado com operações de desaguamento. Esse processo possui o desafio de operar de forma continua e estabilizada. O trabalho de Magalhães e Euzébio (2018) propõe um controlador *fuzzy* supervisório, que monitora o comportamento do processo e então atua na malha de controle da vazão de *underflow*. A tarefa de controle foi manter a vazão de *underflow* e a concentração de sólidos de *underflow* dentro de uma faixa especificada. O processo de espessamento contínuo e o controlador foram simulados em *software* de simulação dinâmica, que inclui dados de uma planta real localizada em uma usina da Vale S.A. em Itabira, Brasil. Os resultados simulados mostraram que o controlador pode controlar com sucesso a concentração de sólidos de *underflow* e a vazão dentro de suas faixas alvo, e que sem o controlador supervisório o processo ficaria instável.

Em Lopes Jr. *et al.* (2022) é proposta uma estratégia avançada de controle regulatório composta por controle em cascata e *override* para um sistema com espessador. O objetivo principal foi aumentar o período de produção, mesmo sob falha do filtro e alterações nas carac-

terísticas da polpa de entrada. Essa estratégia de controle foi avaliada usando um modelo digital de uma planta de processamento de minério de ferro de grande porte. Dois cenários são investigados: a falha simultânea de dois filtros e distúrbios no fluxo e densidade do espessador. Os resultados da simulação mostram que a estratégia de controle proposta pode estender o período de operação do espessador sob falhas nos filtros de disco e rejeitar perturbações significativas.

O trabalho de Pereira *et al.* (2020a) apresenta os resultados de um controle MPC aplicado ao espessador simulado. Os resultados obtidos pelo controlador MPC foi comparado com os resultados obtidos por um controlador PI. As variáveis controladas foram a densidade na descarga e a altura da interface, enquanto que as variáveis manipuladas foram a vazão na descarga e a dosagem de floculante. O modelo não linear do espessador foi validado com dados reais de um espessador de minério de ferro encontrado em uma usina de beneficiamento mineral da Vale S.A. Utilizou-se o índice da Integral do Erro Absoluto (IAE) para avaliar o desempenho dos controladores, sendo possível observar um melhor desempenho para o controlador MPC.

Neste contexto, o objetivo deste trabalho é aplicar o algoritmo de aprendizagem por reforço ARS proposto por Mania *et al.* (2018b) para sintonizarmos um controlador PI do processo (CSTR) e controlar um espessador contínuo. Nós usamos este algoritmo em uma camada de controle avançado para definição dos parâmetros do controlador PI de um processo CSTR em diferentes pontos de operação. Além disso, usamos o algoritmo ARS para controlar diretamente a vazão de saída do *underflow* de um espessador contínuo. A principal motivação para uso do algoritmo ARS deve-se a sua simples implementação, possibilitando seu uso em CLPs.

1.2. Motivação

A aprendizagem por reforço é uma técnica adaptável, pois é capaz de aprender a solucionar problemas de forma autônoma partindo sem conhecimento do processo. A aplicação dessa técnica aplicada ao controle de processos contínuos e complexos como CSTR e espessador é motivada por serem exemplos de processos não lineares e com variações nas características do material processado. Em especial, no espessador, a vazão de entrada e características do material na alimentação variam significativamente ao longo do tempo. Enquanto, no CSTR as reações químicas trazem não linearidades ao processo. Assim, surge a necessidade de implementação de controladores adaptáveis, uma vez que controladores com parâmetros fixos operam com baixa eficiência a depender do ponto de operação. Em ambos os processos surge a necessidade de implementar novas estratégias que consigam se adaptar a variações pertinentes nessas operações.

1.3. Objetivos

1.3.1. Objetivo Geral

Este trabalho tem por objetivo aplicarmos um algoritmo de aprendizagem por reforço para controlar processos contínuos e não lineares, em especial, o CSTR e espessador contínuo.

1.3.2. Objetivos Específicos

1. Desenvolvermos um controlador baseado em aprendizagem por reforço capaz de aprender dinamicamente a melhor sintonia de um controlador PI para o processo CSTR.

2. Desenvolvermos um controlador baseado em aprendizagem por reforço capaz de aprender dinamicamente a controlar diretamente o processo espessador.

3. Aplicarmos controladores baseados em aprendizagem por reforço em simuladores já desenvolvidos.

4. Aumentarmos a produtividade e manter a estabilidade dos processos CSTR e do espessador de forma a evitar paradas nos processos sobrejacentes e subjacentes.

1.4. Perguntas

1. A técnica de aprendizagem por reforço é capaz de aprender a sintonizar um controlador PI do processo CSTR?

2. A técnica de aprendizagem por reforço é capaz de aprender a controlar diretamente um espessador?

Os próximo capítulos desse trabalho foram divididos da seguinte forma: O capítulo 2 apresenta uma fundamentação teórica dos conceitos básicos aplicados aos processos CSTR e espessador. No capítulo 3 são apresentados os resultados e descritos os índices de desempenho das estratégias aplicados aos processos em estudo. Finalmente no capítulo 4 apresentamos as conclusões do trabalho e ainda sugerimo possíveis atividades futuras.

2. Fundamentação Teórica

Nesse capítulo será apresentada um contexto teórico relacionado com o objeto de pesquisa desse trabalho. Na seção 2.1 fizemos um descritivo do processo CSTR e do simulador utilizado nesse trabalho. Já na seção 2.2 o espessador e seu simulador é apresentado com um breve descritivo e formulações. Na seção 2.3 apresentamos fundamentos básicos da aprendizagem por reforço e finalmente na seção 2.4 descrevemos com detalhes o algoritmo ARS, o qual é aplicado na estratégia de controle desse trabalho.

2.1. Descrição do processo CSTR

O processo sob estudo consiste em um CSTR. O trabalho de Antonelli e Astolfi (2003) descreve o processo CSTR em detalhes. Além disso, os autores afirmam que esse tipo de reator é amplamente utilizado na indústria química. Um reator é usado para converter um produto químico **A** em um produto químico útil **B** por um fluxo de resíduos antes de processos subsequentes. A Figura 2.1 apresenta um esquema do CSTR.

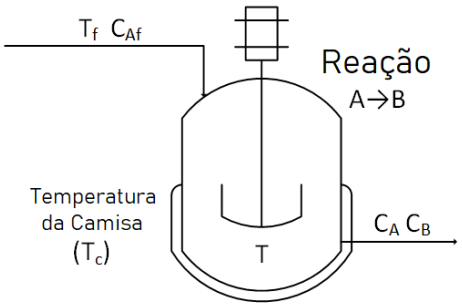

Figura 2.1: Representação simplificada do CSTR.
Fonte: Adaptado de Antonelli e Astolfi (2003).

O processo considerado é modelado dinamicamente como um CSTR com um mecanismo cinético simplificado que descreve a conversão do reagente **A** em produto **B** com uma reação irreversível e exotérmica. Como o analisador do produto **B** não é rápido o suficiente para controle em tempo real, é desejável manter a temperatura em um ponto de ajuste constante que maximize o consumo de **A** (temperatura mais alta possível). Assim, faz-se necessário ajustar a temperatura da camisa (T_c) para manter a temperatura desejada do reator e minimizar a concentração (C_A) do reagente **A** e maximizar a concentração (C_B) de **B**. A temperatura do reator nunca deve exceder 400 K. A temperatura da camisa de resfriamento pode ser ajustada entre 250 K e 350 K. As equações 2.1 e 2.1 do modelo do processo CSTR em estudo são apresentados a seguir e os parâmetros que foram adotados nesse trabalho são exibidos na Tabela 2.1.

$$\dot{C_A} = \frac{q}{V}(C_{AF} - C_A) - k_0 C_A e^{-(E/RT)} \tag{2.1}$$

$$\dot{T} = \frac{q}{V}(T_f - T) - \frac{\Delta H K_0}{\rho C_p} C_A e^{-(E/RT)} + \frac{\rho_c C_{pc}}{\rho C_p V} q_c (1 - e^{-\left(\frac{h_A}{\rho_c C_{pc} q_c}\right)})(T_{cf} - T). \tag{2.2}$$

Tabela 2.1: Parâmetros do CSTR.

Parâmetro	Notação	Valor
Vazão processo	q	100 m^3/s
Concentração da alimentação	C_{AF}	0,877 mol/m^3
Temperatura da alimentação	T_f	350 K
Temperatura da entrada refrigerante	T_{cf}	350 K
Volume do reator	V	100 m^3
Coef. transferência térmica	h_A	7×10^5 cal/ min/ K
Taxa de reação constante	k_0	$7{,}2 \times 10^{10}$ min^{-1}
Energia de ativação	E/R	8750 K
Reação de calor	ΔH	-5×10^4 J/ mol
Densidade dos líquidos	ρ, ρ_c	1×10^3 kg/m^3
Calor específico	C_p, C_{pc}	0,239 J/kg/K
Limite inferior da Temp. da Camisa	$T_{c_{min}}$	250 K
Limite superior da Temp. da Camisa	$T_{c_{max}}$	350 K

Fonte: Adaptado de Hedengren (2021).

2.2. Espessadores

A água está presente em diversas etapas dos processos de mineração. As principais vantagens do uso da água são relacionadas ao transporte e separação dos materiais. Entretanto, em termos ambientais, há uma preocupação do setor em formas mais eficientes de reutilizar a água nos processos de beneficiamento de minério. Nesse cenário, as operações de desaguamento com espessador ou filtro ganham relevância nos circuitos de beneficiamento (LUZ e LINS, 2018).

2.2.1. Descrição do Equipamento

Os espessadores podem ser descritos como tanques cilindro-cônicos alimentados pelo centro por um material que sedimenta e é retirado pelo fundo. Um mecanismo de raspagem é responsável por movimentar o material da região de compactação para o ponto de descarga da lama. A Figura 2.2 mostra em detalhes o diagrama funcional de um espessador contínuo.

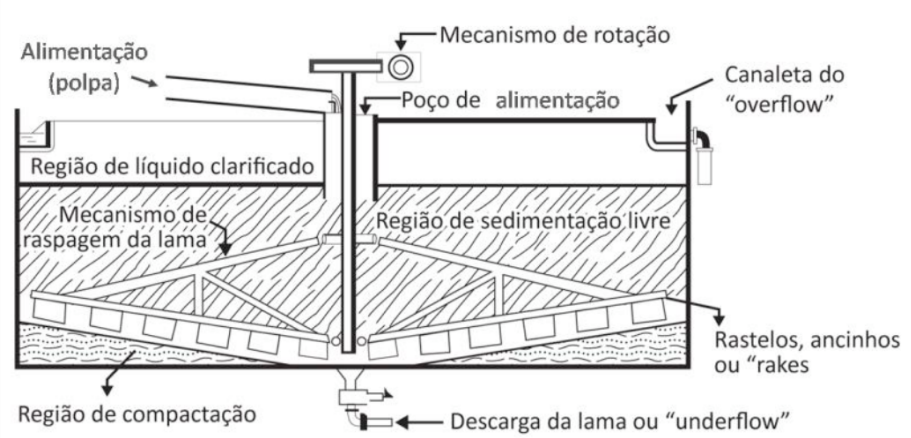

Figura 2.2: Representação de um espessador contínuo.
Fonte: Luz e Lins (2018).

A polpa que alimenta o espessador é constituída de partículas sólidas de rejeito e água. O espessador é alimentado com essa polpa pelo poço de alimentação. A partir desse momento, na região de sedimentação livre, a ação gravitacional sobre as partículas age de forma a carreá-las para a região de compactação. Enquanto isso, a canaleta do *overflow* receberá da região de líquido clarificado uma água pronta para ser reutilizada nos processos. O mecanismo de rotação movimenta os rastelos na região de compactação, assim direcionando o material concentrado para o ponto de descarga da lama, conhecido também como *underflow*. Em alguns casos, nesse ponto, os espessadores possuem uma bomba para succionar a lama.

De acordo com Chaves (2004), a sedimentação das partículas ocorre em três etapas distintas. Na etapa de clarificação a polpa se encontra altamente diluída e as partículas se sedimentam praticamente sem iteração com as outras. Após um tempo, as partículas vão colidir e formar flóculos, com isso elas passam a sedimentar com uma velocidade cada vez maior. Esse regime é denominado de sedimentação por fase, nela a concentração de partículas começa a aumentar. A próxima etapa é a compactação do material no fundo do espessador. As partículas estarão tão unidas e o peso das suprajacentes fará a compactação do material.

Os principais agentes que afetam o funcionamento do espessador são as forças da gravidade, o empuxo do líquido deslocado e as forças de atrito que se manifestam entre líquido e partícula. Segundo Chaves (2004), os fatores que influenciam essas forças são:

- As propriedades da partícula: dimensões, forma, densidade e rugosidade superficial;
- A densidade e viscosidade da polpa;
- As características do processo: percentual de sólidos, o estado de dispersão das partículas e o pH;
- Dimensões do equipamento.

2.2.2. Simulador

O trabalho de Pereira (2021) apresenta um modelo para espessadores cilindro-cônicos convencionais que considera a adição de floculantes com o objetivo de realizar o controle do nível da interface de sedimentos. O simulador foi desenvolvido no *software MATLAB* e é baseado em um método numérico para a simulação do modelo em ambiente matemático. O diagrama de blocos do simulador é apresentado na Figura 2.3.

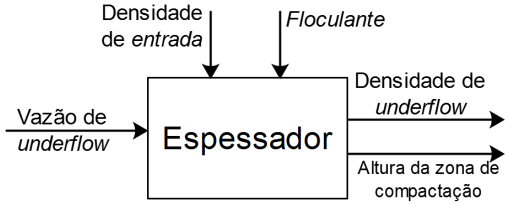

Figura 2.3: Principais variáveis do espessador simulado.
Fonte: Adaptado de Pereira (2021).

Em resumo, o conjunto de equações que representam o modelo matemático do espessador simulado são apresentadas a seguir.

$$\frac{\partial \phi}{\partial t} + \frac{\partial}{\partial z}\overline{F}(\phi, w/\phi, z, t) = \frac{\partial}{\partial z}\left(\gamma(z)\frac{w}{\phi}\frac{\partial D(\phi)}{\partial z}\right) + \frac{Q_f(t)\phi_f(t)}{A(z)}\delta(z) \tag{2.3}$$

$$\frac{\partial w}{\partial t} + \frac{\partial}{\partial z}\left(\frac{w}{\phi}\overline{F}(\phi, k, z, t)\right) = \frac{\partial}{\partial z}\left(\gamma(z)\frac{w^2}{\phi^2}\frac{\partial D(\phi)}{\partial z}\right) + \frac{Q_f(t)w_f(t)}{A(z)}\delta(z). \tag{2.4}$$

Definindo-se $Q(t)$ como:

$$Q(t) = \begin{cases} -Q_e(t), & \text{para} \quad z < 0, \\ Q_u(t), & \text{para} \quad z > 0, \end{cases}$$

pode-se redefinir $\overline{F}(\phi, k, z, t)$ como:

$$\overline{F}(\phi, z, t) = \begin{cases} Q(t)\phi/A(z), & \text{para} \quad z < -H \\ Q(t)\phi/A(z) + k\upsilon_{hs}(\phi)\phi, & \text{para} \quad -H < z < B \\ Q(t)\phi/A(z), & \text{para} \quad z > B \end{cases}$$

Ainda, multiplicando-se (2.3) e (2.4) por $A(z)$, tem-se

$$\frac{\partial (A(z)\phi)}{\partial t} + \frac{\partial}{\partial z}\left(A(z)\overline{F}(\phi, w/\phi, z, t)\right) = \frac{\partial}{\partial z}\left(A(z)\gamma(z)\frac{w}{\phi}\frac{\partial D(\phi)}{\partial z}\right) + Q_f(t)\phi_f(t)\delta(z) \tag{2.5}$$

$$\frac{\partial (A(z)w)}{\partial t} + \frac{\partial}{\partial z}\left(\frac{w}{\phi}A(z)\overline{F}(\phi, k, z, t)\right) = \frac{\partial}{\partial z}\left(A(z)\gamma(z)\frac{w^2}{\phi^2}\frac{\partial D(\phi)}{\partial z}\right) + Q_f(t)w_f(t)\delta(z). \tag{2.6}$$

A área da seção transversal é variável e depende da profundidade, denotada por $A(z)$.

$$A(z) = \begin{cases} \frac{\pi}{4}(D_{max}^2 - D_{fw}^2), & \text{para} \quad -H < z \leq 0 \\ \frac{\pi}{4}D_{max}^2, & \text{para} \quad 0 < z \leq H_c \\ \frac{\pi}{4}D(z)^2, & \text{para} \quad H_c < z \leq B, \end{cases} \tag{2.7}$$

As variáveis empregadas nas equações 2.3, 2.4, 2.5, 2.6 e 2.7 são descritas abaixo.

- ϕ: Concentração volumétrica de sólidos.
- $\overline{F}$: Função de fluxo convectivo.
- z Profundidade.
- w Quantidade conservativa que descreve a concentração de partículas sólidas que carregam a propriedade k consigo.
- $\gamma(z)$: é um parâmetro que indica se z é um nível dentro ($\gamma = 1$) ou fora ($\gamma = 0$) do espessador.
- D: é o diâmetro.
- $\delta(z)$: Função delta de *Dirac* de z.
- k: Propriedade referente à adição de floculante.
- w_f: Quantidade conservativa que descreve a concentração de partículas sólidas que carregam a propriedade k consigo na alimentação.
- $A(z)$: Área da seção transversal.
- $Q_e(t)$: vazão volumétrica do *overflow*.
- $Q_u(t)$: vazão volumétrica do *underflow*.
- υ_{hs}: Função de velocidade de sedimentação retardada.
- D_{fw}: Diâmetro do *feedwell*
- $Q(t)$: vazão volumétrica.

Os parâmetros empregados nesse trabalho são apresentados na Tabela 2.2.

Tabela 2.2: Principais parâmetros do modelo do espessador.

Parâmetro	Notação	Valor
Vazão volumétrica na alimentação	Q_f	$0{,}135 m^3/s$
Concentração volumétrica de sólidos na alimentação	ϕ_f	1,64
Profundidade do *overflow* a partir da alimentação	D_{max}	$2m$
Profundidade da região de área variável a partir da alimentação	H_c	$1{,}2m$
Profundidade do *underflow* a partir da alimentação	B	$5m$

Fonte: Adaptado de Pereira (2021).

O simulador proposto por Pereira (2021) propõe um novo modelo e apresenta benefícios inéditos que podem ser muito úteis na simulação de espessadores. Este ainda considerar a adição de floculante, algo pouco encontrado na literatura. O simulador proposto apresenta a vantagem de representar com fidelidade considerável a geometria de espessadores reais. Como desvantagem o simulador apresenta alto tempo de execução se utilizamos parâmetros que aumentam a acurácia e qualidade dos resultados.

2.3. Aprendizagem por Reforço Aplicado em Controle

A aprendizagem por reforço apresenta a capacidade de aprender tarefas sem nenhum conhecimento e ainda continuar aprendendo quando acontece situações antes não apresentadas pelo ambiente. Essas características são convidativas para aplicação nos problemas de controle de processo. Alguns trabalhos são relatados na literatura.

O trabalho de Wei *et al.* (2017) desenvolve uma abordagem baseada em dados que aproveita a técnica de aprendizado de reforço profundo para aprender de forma inteligente a estratégia eficaz para operar os sistemas de climatização de edifícios. O algoritmo foi aplicado primeiramente a um simulador e posteriormente em uma situação real. Os autores comparam o desempenho do seu algoritmo com um *Q-learning* convencional e um algoritmo de aprendizagem por reforço heurístico de 5 níveis. Os resultados demonstraram que o algoritmo proposto é mais eficaz na redução de custos de energia em comparação com a abordagem tradicional baseada em regras, enquanto manteve a temperatura ambiente dentro da faixa desejada.

Hallén *et al.* (2019) apresenta um controle de um circuito de moagem em dois estágios usando aprendizagem por reforço. Uma solução para integração de modelos de simulação industrial ao *framework* de aprendizagem por reforço *OpenAI Gym*. Um controlador (PI) é usado para comparar com o desempenho do controlador proposto. A comparação mostra que é possível controlar o circuito de moagem usando o aprendizado por reforço. Para alguns casos operacionais, o algoritmo é capaz de controlar a planta de forma mais eficiente em comparação com o controle existente.

Uma abordagem para controle com aprendizagem por reforço com repetição de experiência (ER) é relatada por Adam *et al.* (2012). Essa estratégia permite que o algoritmo aprenda

rapidamente com uma quantidade limitada de dados, apresentando-os repetidamente a um algoritmo de aprendizagem por reforço subjacente. A estratégia é avaliada em experimentos de controle em tempo real que envolvem o problema do pêndulo oscilante e um controle baseado na visão de um robô goleiro. Os mesmos experimentos são feitos com aprendizagem por reforço tradicional e os resultados apresentados são incentivadores para aplicabilidade da estratégia proposta na prática.

Spielberg *et al.* (2017) estende o aprendizado por reforço e o aprendizado profundo para problemas de controle de processos. Neste trabalho, foi mostrado que se as funções de recompensa forem formuladas adequadamente podem ser usadas para o controle de processos industriais. O formato do controlador segue a configuração típica da aprendizagem por reforço, em que o agente (controlador) interage com o ambiente (processo) por meio de ações de controle enquanto recebe recompensas. As redes neurais profundas também são utilizadas para generalizar e aprender as políticas de controle. A abordagem dos autores é avaliada em sistemas de entrada única e saída única (SISO) e sistemas de múltiplas entradas e múltiplas saídas (MIMO) em diferentes cenários.

A aprendizagem por reforço é aplicada para alterar os parâmetros de sintonia de um controlador PI no trabalho de Shipman e Coetzee (2019), considerando apenas a variável de processo, ponto de ajuste, variável manipulada e ganhos anteriores do controlador. O algoritmo usado é um agente *actor-critic* com redes neurais. Um sistema de primeira ordem foi simulado e usado para aplicar o método proposto. Os autores comparam o desempenho do seu algoritmo com constantes obtidas observando o modelo matemático. A inteligência artificial apresentou um valor de *integral absolute error* (IAE) de 4,37 enquanto o controlador clássico 3,96.

Outro algoritmo *actor-critic* baseado no gradiente de política determinística é apresentado por Lillicrap *et al.* (2015). Neste trabalho, foi usado um mesmo algoritmo de aprendizagem, arquitetura de rede neural e hiperparâmetros para solucionar de forma robusta mais de 20 tarefas simuladas. Essas tarefas incluem: o problema clássico do pêndulo invertido, locomoção por pernas e condução de veículo. O algoritmo proposto é capaz de encontrar soluções com desempenho competitivo com algoritmos que possuem acesso pleno ao modelo. Uma limitação, que acontece na maioria das abordagens de aprendizagem por reforço, foi observada pelos autores, o algoritmo necessitou de um grande número de episódios de treinamento para encontrar os melhores desempenhos.

O trabalho de Li *et al.* (2018) apresenta um controlador baseado em aprendizagem por reforço aplicado a um modelo experimental de helicóptero. Os autores construíram um módulo de função de recompensa para descrever o ambiente específico do sistema, levando as pertubações em consideração. O algoritmo usa uma estrutura de *actor-critic* com duas redes neurais para implementar o controlador. Os resultados mostraram que o controlador proposto pode estabilizar o sistema sob a incerteza do modelo e atenuar a perturbação.

Um problema de controle operacional ótimo para o processo de flotação industrial é estudado no trabalho de Jiang *et al.* (2018). O autor apresenta um novo método orientado

por dados e sem modelo para solução em tempo real do problema de controle. O algoritmo proposto é uma técnica conhecida como Aprendizagem Intercalada, que combina as melhores características do *Policy Iteration* e *Value Iteraction*. Foram usadas três redes neurais: uma para generalizar o modelo, uma para o *critic* e outra para o *actor*. O desempenho do algoritmo de aprendizagem, em que o *critic* e o *actor* são atualizados, alternadamente, uma vez por iteração, e apresentou um valor de MSE 3,6 vezes menor que o controlador PI.

O trabalho de Spielberg *et al.* (2019) explora desenvolvimentos recentes em aprendizagem por reforço e aprendizagem profunda para desenvolver um novo controlador adaptativo livre de modelo para processos gerais de tempo discreto. O controlador, denominado de *Deep Reinforcement Learning (DRL) Controller* é baseado em um gradiente de política determinística e implementado usando uma arquitetura *actor-critic* e usa duas redes neurais profundas para generalizar o *actor* e o *critic*. A eficácia do controlador DRL é demonstrada em quatro exemplos de simulação de problemas de rastreamento de *setpoint*. Os exemplos são: uma máquina de fabricação de papel de entrada única e saída única; uma coluna de destilação de alta pureza com múltiplas entradas e múltiplas saídas; e um sistema de aquecimento, ventilação e ar condicionado não linear. O controlador proposto apresentou-se como uma alternativa viável aos controladores clássicos, embora com desafios teóricos e práticos a serem superados. Os autores ressaltam que com o aumento do poder computacional e do volume de dados o controlador DRL tem potencial de se tornar uma ferramenta importante no controle de processo.

Finalmente, um algoritmo simples para aprendizagem por reforço livre de modelo é apresentado por Mania *et al.* (2018a), denominado *Augmented Random Search* (ARS). Em resumo, o algoritmo é uma busca aleatória simples com três melhorias, sendo elas: na etapa de atualização utiliza uma escala com o desvio padrão das recompensas; normalização em tempo real dos estados e utilização somente das direções que tiveram as melhores performances para atualizar os parâmetros. O algoritmo apresentou um grande potencial para lidar com problemas de controle contínuo. O método ARS foi 15 vezes mais eficiente computacionalmente do que a estratégia de evolução (ES) (SALIMANS *et al.*, 2017b), com essa eficiência foi possível explorar um amplo espaço de políticas mais adequadamente sobre muitas sementes aleatórias e diferentes escolhas de hiperparâmetros.

2.3.1. Fundamentos de Aprendizagem por Reforço

A aprendizagem por reforço é uma técnica que esta contida na inteligência artificial. A inteligência artificial por si é qualquer tecnologia que permite a um sistema demonstrar inteligência humana. Ainda dentro da inteligência artificial temos outras áreas, como a aprendizagem supervisionada e a não supervisionada. A aprendizagem por supervisionada aprende a partir de entradas e saídas conhecidas, enquanto a aprendizagem não supervisionada o algoritmo aprende a encontrar semelhanças e diferenças entre dados e então classificar-los.

A estrutura básica da técnica de aprendizagem por reforço é apresentada na Figura 2.4.

Esta técnica é baseada no aprendizado adquirido ao decorrer da iteração entre um agente e um ambiente em uma sequência de passos de tempo discretos, ($t = 1, 2, 3, \ldots$). A cada passo de tempo t, o agente recebe do ambiente um estado $s_t \in S$, onde S é um conjunto de possíveis estados, com base nesse estado seleciona uma ação $a_t \in A$, onde A é um conjunto de ações possíveis e aplica no ambiente. O ambiente retorna uma recompensa numérica r_t, que indica o quão bom foi a ação tomada, e um novo estado s_{t+1}. A cada passo de tempo, o agente faz um mapeamento dos estados de acordo com as probabilidades de selecionar cada possível ação a_t. O agente aprende como melhor interagir com o ambiente pela maximização das recompensas que recebe (SUTTON e BARTO, 2018).

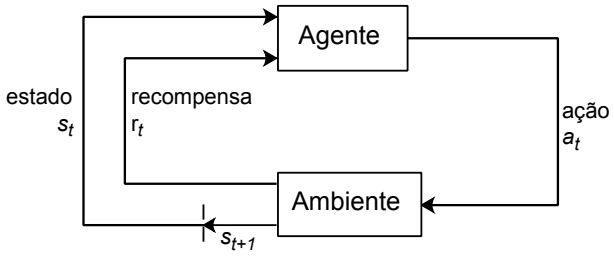

Figura 2.4: Estrutura básica da técnica de aprendizagem por reforço.
Fonte: Adaptado de Sutton e Barto (2018).

De acordo com Sutton e Barto (2018) os principais elementos da aprendizagem por reforço são: o agente, a ação, o estado, a recompensa e o ambiente. Esses elementos são descritos a seguir.

- **Agente**: O agente consiste no algoritmo que toma ações e aprende. O agente escolhe uma ação a_t em cada estado s_t, e as percepções que o agente obtém do ambiente são o estado do ambiente após cada ação e o sinal de recompensa escalar r_t em cada passo.
- **Ambiente**: O ambiente é tudo o que o agente não pode controlar. Em outras palavras, o ambiente é o processo em que o agente interage.
- **Ações**: As ações são o conjunto de formas que o agente pode interagir com ambiente. Esse conjunto pode ser composto de valores numéricos ou comandos direcionais, por exemplo.
- **Estados**: Os estados são as possíveis percepções que o agente pode ter do ambiente. Por exemplo, quando o ambiente for um tabuleiro de xadrez, os estados serão as possíveis posições que as peças podem ocupar.
- **Recompensa**: A recompensa é um valor escalar que indica numericamente quão bom foi para o agente tomar uma ação em cada passo de tempo. Analogamente, em um sistema biológico podemos pensar nas recompensas como experiência de prazer e dor. Em geral, os sinais de recompensa podem estar em função do estado, do ambiente e das ações realizadas.

- **Política**: Uma política, denotada por π, é uma regra pela qual o agente seleciona ações em função dos estados, assim definindo o comportamento do agente. Sendo o núcleo de um agente de aprendizagem por reforço no sentido de que por si só é suficiente para determinar o comportamento. A política pode ser determinística ou estocástica. Na determinística, é feito um mapeamento de cada estado para uma ação, enquanto, na estocástica, é feito um mapeamento de cada estado para uma probabilidade de selecionar cada ação possível. A política pode ser uma tabela de pesquisa, uma função simples, um método de busca e até mesmo uma rede neural.

A aprendizagem por reforço possui dois conjuntos de métodos de solução: *i*) O conjunto dos métodos de solução tabular; *ii*) o conjunto dos métodos de solução aproximada. Os métodos de solução tabular consideram um espaço finito de estados e ações na construção da aprendizagem por reforço e todas as combinações de ações, estados e a política precisam ser armazenadas, na forma de tabela, para estar sempre disponível para consulta durante as iterações. Enquanto os métodos de solução aproximada são capazes de considerar um espaço infinito de estados e ações na maioria dos casos. O conhecimento (política) é armazenado na forma parametrizada, seja por uma tabela parametrizada, função parametrizada ou uma rede neural profunda.

2.4. O algoritmo *Augmented Random Search* (ARS)

O algoritmo *Augmented Random Search (ARS)* propõe uma estratégia de aprendizagem por reforço livre de modelo mais simples de implementação. O método é proposto por Mania *et al.* (2018b) e aborda um novo método de aprendizagem por reforço baseado em busca aleatória simples com melhorias no processamento das recompensas e dos estados, baseadas em heurísticas aplicadas com sucesso na aprendizagem por reforço profunda. A proposta desse método surgiu devido aos outros métodos necessitarem de muitos dados para atingir um bom desempenho e apresentarem implementação complexa.

Para obter o método livre de modelo, duas abordagens foram combinadas para se criar o método ARS, sendo elas: o método proposto por Salimans *et al.* (2017a), no qual a política é otimizada sem uso de derivadas, e o método proposto por Rajeswaran *et al.* (2018), o qual mostra que as políticas lineares podem ser treinadas por meio de gradiente de política simples e obter desempenho competitivo com políticas de rede neural complexas. O pseudocódigo do algoritmo ARS é apresentado no Algoritmo 1.

Algoritmo 1 : ARS

Hiperparâmetros:

n ▷ *número de entradas*

p ▷ *número de saídas*

α ▷ *taxa de aprendizagem*

υ ▷ *ruído de exploração*

passos ▷ *número de iterações*

N ▷ *número de direções por direção de ajuste*

b ▷ *número de direções com os melhores desempenhos*

Inicializa:

$M_0 \leftarrow 0 \in R^{p \times n}$ ▷ *matriz de pesos da rede neural perceptron*

$\mu_0 \leftarrow 0 \in R^n$ ▷ *média das entradas*

$j \leftarrow 0$

Para $j \leftarrow 0$ **até** *passos* **Faça**

    Crie as matrizes $\delta_1, \delta_2, \ldots, \delta_N$ em $R^{p \times n}$ preenchidas com valores uniformes entre 0 e 1

    Crie dois vetores $r(\pi_{j,k,-})$ e $r(\pi_{j,k,+})$ em R^N vazios

    Para $k \leftarrow 0$ **até** N **Faça**

        $r(\pi_{j,k,+}) \leftarrow$ recompensa de $\pi_{j,k,+}(x) = (M_j + \upsilon\delta_k)\ \text{diag}\ (\sum_j)^{-\frac{1}{2}}(x - \mu_j)$

    Fim Para

    Para $k \leftarrow 0$ **até** N **Faça**

        $r(\pi_{j,k,-}) \leftarrow$ recompensa de $\pi_{j,k,-}(x) = (M_j - \upsilon\delta_k)\ \text{diag}\ (\sum_j)^{-\frac{1}{2}}(x - \mu_j)$

    Fim Para

    Ordena as direções δ_k pela máxima recompensa em $\{r(\pi_{j,k,+}), r(\pi_{j,k,-})\}$, sendo então $\pi_{j,(k),+}$ e $\pi_{j,(k),-}$ as políticas respectivas

    Atualiza

    $M_{j+1} = M_j + \frac{\alpha}{b\sigma^R} \sum_{k=1}^{b} [r(\pi_{j,k,+}) - r(\pi_{j,k,-})]\delta_k$, onde σ^R é o desvio padrão das recompensas usadas na etapa de atualização

    Configura μ_{j+1}, $\sum_{j+1}$ para ser a média e covariância dos estados $2NH(j+1)$ encontra dos desde o início do treinamento

    $j \leftarrow j + 1$

Fim Para

A seguir descrevemos em detalhes cada uma das etapas do algoritmo ARS para um exemplo hipotético. Alguns hiperparâmetros são iniciados levando em consideração o projeto de controle, como a quantidade de passos, o número de entradas (n - quantidade de estados) e as saídas (p - quantidade de ações). Enquanto, outros hiperparâmetros são definidos por tentativa e erro ao fim das execuções completas, são eles: a taxa de aprendizagem (α), o ruído de exploração (υ), a quantidade de direções de ajuste (N) e a quantidade de direções consideradas para ajuste da política (as que apresentarem b melhores desempenhos). Os hiperparâmetros conside-

rados nesse exemplo são os mencionados na Tabela 2.3. De forma simplificada o diagrama do ARS é representado pela Figura 2.5.

Tabela 2.3: Hiperparâmetros considerados no exemplo.

Hiperparâmetro	Notação	Valor
Número de entradas	n	3
Número de saídas	p	2
Taxa de aprendizagem	α	0,3
Ruído de exploração	υ	0,2
Número de iterações	*passos*	3000
Número de direções por direção de ajuste	N	4
Número de direções com os melhores desempenhos	b	2

Fonte: O autor.

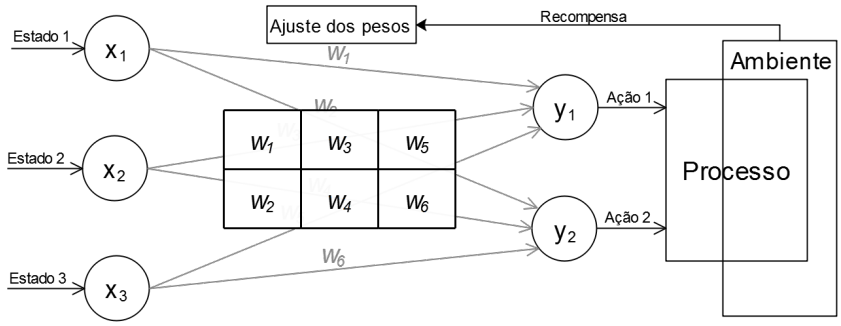

Figura 2.5: Diagrama simplificado do algoritmo ARS.
Fonte: O autor.

No primeiro passo j, inicializamos uma matriz com pesos de uma rede neural *perceptron*, sendo suas dimensões com (p) linhas e (n) colunas e seus valores preenchidos com o valor 0 e um vetor μ_0 para salvar a média das entradas no decorrer das iterações:

$$M_0 \leftarrow \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}, \quad \mu_0 \leftarrow \begin{bmatrix} 0 & 0 & 0 \end{bmatrix}$$

Após as inicializações, o ARS irá iteragir com o processo pelo número de passos já definidos. A seguir será descrito as iterações para um passo:

- Inicializa as $N = 4$ matrizes de ruído δ_1, δ_2, δ_3 e δ_4 com valores sorteados uniformemente entre 0 e 1

$$\delta_1 \leftarrow \begin{bmatrix} 0{,}12 & 0{,}36 & 0{,}98 \\ 0{,}02 & 0{,}23 & 0{,}57 \end{bmatrix}, \quad \delta_2 \leftarrow \begin{bmatrix} 0{,}01 & 0{,}04 & 0{,}25 \\ 0{,}15 & 0{,}36 & 0{,}42 \end{bmatrix}$$

$$\delta_3 \leftarrow \begin{bmatrix} 0{,}57 & 0{,}41 & 0{,}85 \\ 0{,}24 & 0{,}01 & 0{,}51 \end{bmatrix}, \quad \delta_4 \leftarrow \begin{bmatrix} 0{,}76 & 0{,}49 & 0{,}28 \\ 0{,}12 & 0{,}27 & 0{,}37 \end{bmatrix}$$

- Criam-se dois vetores $r(\pi_{j,k,-})$ e $r(\pi_{j,k,+})$ com $N = 4$ colunas preenchidas com 0 para armazenar as recompensas retornadas por cada direção de ajuste experimentado pela matriz de pesos M

$$r(\pi_{j,k,-}) \leftarrow \begin{bmatrix} 0 & 0 & 0 & 0 \end{bmatrix}, \quad r(\pi_{j,k,+}) \leftarrow \begin{bmatrix} 0 & 0 & 0 & 0 \end{bmatrix}$$

- Nessa etapa, o ARS irá explorar (por T iterações) o processo no sentido de ajuste positivo $(+)$ e negativo $(-)$ da matriz M utilizando-se das inicializações até o momento e armazenar as recompensas obtidas por cada ajuste nos vetores $r(\pi_{j,k,-})$ e $r(\pi_{j,k,+})$. Portanto, o ARS irá executar o processo/ambiente por 8 vezes, sendo 4 episódios no sentido de ajuste positivo, no qual a matriz M_j é somada ao produto de υ com a matriz δ_k e 4 episódios no sentido negativo, onde é feita a subtração da matriz M_j com o produto de υ com a matriz δ_k. Em cada execução, também conhecida como episódio, explora-se todo processo até a iteração T, com as devidas mudanças de referência propostas. Um episódio com a direção de ajuste positivo será descrito abaixo, já que o ajuste no sentido negativo é semelhante.

  - No episódio k, a matriz M_j inicialmente é somada ao produto do hiperparâmetro υ com uma matriz δ_k da seguinte forma:

$$(M_j + \upsilon\delta_k) \leftarrow \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} + 0{,}2 \times \begin{bmatrix} 0{,}12 & 0{,}36 & 0{,}98 \\ 0{,}02 & 0{,}23 & 0{,}57 \end{bmatrix}$$

$$(M_j + \upsilon\delta_k) \leftarrow \begin{bmatrix} 0{,}024 & 0{,}072 & 0{,}196 \\ 0{,}004 & 0{,}046 & 0{,}114 \end{bmatrix}$$

  - Assim essa matriz é usada na *perceptron* do ARS durante todo esse episódio. A Figura 2.6 ilustra o ARS com a *perceptron* utilizada nesse episódio.

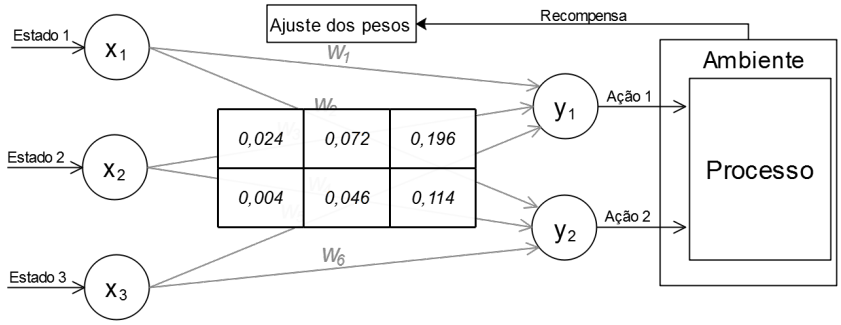

Figura 2.6: Diagrama simplificado do ARS com os pesos calculados.
Fonte: O autor.

  - As entradas da *perceptron* $x = (x_1, x_2, x_3)$ são padronizadas (diag $(\sum_j)^{-\frac{1}{2}}(x - \mu_j)$) da seguinte forma:

Algoritmo 2

1: $m \leftarrow m+1$ ▷ *contador para auxiliar no cálculo da média*

2: $lastMean \leftarrow mean$ ▷ *recupera a última média que foi calculada*

3: $mean \leftarrow mean + \frac{x-mean}{m}$ ▷ *calcula a média efetivamente*

4: $meanDiff \leftarrow meanDiff + (x-lastMean) \times (x-mean)$ ▷ *diferença entre a média atual e a media antiga*

5: $var \leftarrow \frac{meanDiff}{m}$ ▷ *calcula a variância*

6: $std \leftarrow \sqrt{var}$ ▷ *calcula o desvio padrão*

7: $x \leftarrow \frac{x-obsMean}{std}$ ▷ *padroniza as entradas* x

– Nessa etapa, tem-se a matriz de pesos M_j e as entradas x, e para cada tempo t, será calculada uma saída y da *perceptron* e aplicado ao processo

$$y_1 \leftarrow x_1 \times w_1 + x_2 \times w_3 + x_3 \times w_5$$

$$y_2 \leftarrow x_1 \times w_2 + x_2 \times w_4 + x_3 \times w_6$$

– Com a ação calculada, ela então será aplicada ao processo/ambiente e o mesmo retornará para o ARS uma recompensa referente a essa ação tomada nesse estado. O somatório da recompensa de todo episódio é então armazenado na posição k do vetor $r(\pi_{j,k,+})$.

- Agora, após a execução dos 8 episódios (4 no sentido de ajuste positivo e 4 no sentido de ajuste negativo), tem-se os vetores $r(\pi_{j,k,+})$ e $r(\pi_{j,k,-})$ preenchidos com a recompensa obtida em cada episódio. Neste momento, calcula-se o desvio padrão da diferença entre esses vetores:

$$\sigma^R \leftarrow std(r(\pi_{j,k,+}) - r(\pi_{j,k,-}))$$

- Neste passo criamos um vetor br de mesmo tamanho de $r(\pi_{j,k,+})$ com as maiores recompensas obtidas nos dois vetores $r(\pi_{j,k,+})$ e $r(\pi_{j,k,-})$. Por exemplo:

$$r(\pi_{j,k,+}) = \begin{bmatrix} 100 & 10 & 1 & 26 \end{bmatrix}, \quad r(\pi_{j,k,-}) = \begin{bmatrix} 25 & 20 & 2 & 20 \end{bmatrix}$$

  O vetor gerado será $br \leftarrow \begin{bmatrix} 100 & 20 & 2 & 26 \end{bmatrix}$

- O vetor br é ordenado de forma decrescente e apenas os b melhores valores e as respectivas matrizes δ_k utilizamos no próximo passo que será o ajuste dos pesos da matriz M_j.

– Nessa etapa fazemos o somatório da diferença de cada par de vetor r multiplicado pela respectiva matriz δ_k.

$$\sum_{k=1}^{b} [r(\pi_{j,k,+}) - r(\pi_{j,k,-})]\delta_k$$

– Agora a nova matriz M_{j+1}, que será usada no próximo passo $(j+1)$ será calculada.

$$M_{j+1} \leftarrow M_j + \frac{\alpha}{b\sigma^R} \sum_{k=1}^{b} [r(\pi_{j,k,+}) - r(\pi_{j,k,-})]\delta_k$$

3. Desenvolvimento e Aplicações de Controladores usando o ARS

Nesse capítulo apresentamos a estrutura da estratégia de controle propostas para o CSTR e o Espessador, bem como os resultados obtidos. Na seção 3.1 são apresentadas o diagrama e um descritivo da interação do ARS com o processo CSTR, após esse contexto, os resultados dessa interação são apresentados na seção 3.2. Já na seção 3.3 apresentamos o diagrama e um descritivo da interação do ARS com o espessador. Os resultados desta interação são descritos e ilustrados na seção 3.4.

3.1. Estrutura de controle proposta para o processo CSTR

O CSTR é um sistema não linear e multivariável. Neste estudo, consideramos um controlador PI que apenas atua na temperatura da camisa (T_c) para ajustar a temperatura do reator (T). A outra variável manipulada (T_f) mantemos fixa. O controle do processo é realizado através de um controlador PI, dado por:

$$u(t) = K_p e(t) + K_i \int_0^t e(\tau) d\tau.$$

Os parâmetros do controlador PI são definidos pelo algoritmo ARS para cada referência, pois, em cada ponto de operação, o sistema se comporta de forma diferente. O diagrama de blocos do sistema de controle é apresentado na Figura 3.1.

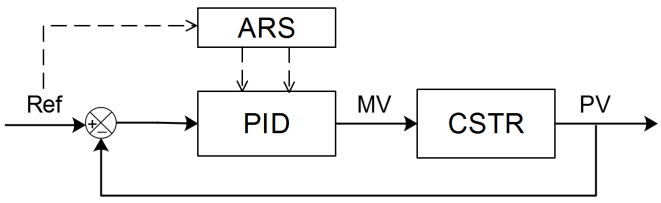

Figura 3.1: Diagrama de blocos do sistema de controle para o processo CSTR.
Fonte: O autor.

Um episódio de iteração do algoritmo ARS com o processo CSTR foi definido como a execução do simulador por 91 passos, considerando um tempo suficiente para o processo entrar em regime permanente. Nessa execução, em dado momento, o ambiente propõe uma nova referência como estado e assim o algoritmo ARS propõe parâmetros de sintonia do controlador PI. No início de cada episódio, o processo é redefinido para as suas condições iniciais. A cada passo do episódio, o algoritmo observa a referência atual e a próxima referência como o estado do ambiente e propõe para esse estado uma ação (parâmetros do controlador PI). Os parâmetros do processo simulado foram apresentados na Tabela 2.1.

Cada ação é um conjunto de parâmetros de sintonia do controlador PI contendo os parâmetros K_p e K_i , $\in R_+$. O ambiente então executa a mudança de referência com os parâmetros por um tempo t e então retorna ao algoritmo uma recompensa e os próximos estados. Uma vez que o objetivo de controle é melhorar o rastreamento da referência, define-se a recompensa como o inverso do somatório do módulo do erro entre a referência e variável controlada (temperatura do reator). Logo, a recompensa é dada por:

$$Reward = \frac{1}{\sum |erro_t|}.$$

A escolha da função de recompensa impacta diretamente nas especificações da resposta do sistema de controle, pois esta indica o quão bom foi a ação de controle obtida a partir do algoritmo. Os estados foram escolhidos como os *setpoints* pois a dinâmica do processo muda de acordo com a mudança do ponto de operação. Um resumo com os hiperparâmetros considerados são apresentados na Tabela 3.1. Os hiperparâmetros: taxa de aprendizagem, ruído de exploração, direções, melhores direções e iterações foram escolhidos por tentativa e erro. Já os passos por episódio foram determinados considerando uma janela de tempo de 1,75min para o controlador PI atuar com a sintonia proposta.

Tabela 3.1: Resumo dos hiperparâmetros do algoritmo ARS para sintonia do controlador PI do processo CSTR.

Hiperparâmetros	Valor
Estados	$[SP_n, SP_{n+1}]$
Ações	$[K_{pn}, K_{in}]$
Recompensa	$\frac{1}{\sum \lvert erro_t \rvert}$
Taxa de aprendizagem	0,3
Ruído de exploração	0,2
N Direções	16
b Melhores Direções	8
Iterações	3000
Passos por episódio	4

Fonte: O autor.

O algoritmo em *Python* desenvolvido por Hedengren (2021) para o controle do CSTR foi utilizado como simulador. O algoritmo ARS foi conectado ao simulador com objetivo de sintonizar o controlador PI. A cada iteração do algoritmo ARS, o processo é simulado por por 80 passos com a nova referência e a respectiva sintonia proposta. O algoritmo completo foi executado na plataforma online *Google Colaboratory*.

3.2. ARS aplicado a sintonia de um controlador PI no processo CSTR

Nesta seção apresentamos os resultados da sintonia dos controladores PI para o processo CSTR usando o algoritmo ARS. O algoritmo completo foi executado por 3 mil iterações e em cada iteração o episódio foi executado por 32 (2×16) direções de ajuste, totalizando 96 mil execuções completas do processo. O gráfico com o histórico de aprendizagem é apresentado na Figura 3.2.

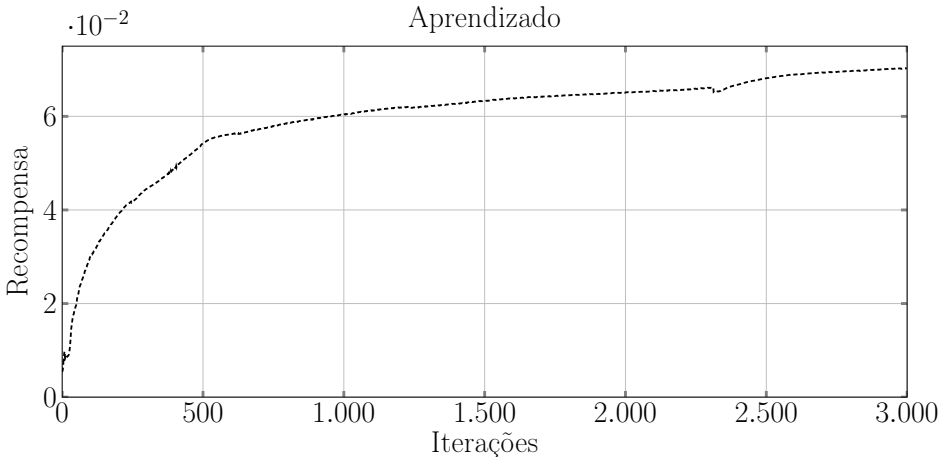

Figura 3.2: Aprendizado do ARS para sintonia do controlador PI do processo CSTR.
Fonte: O autor.

Conforme exibido na Figura 3.2, o algoritmo convergiu após 2500 iterações e teve o tempo de execução de 63 minutos. Observe que os parâmetros do controlador PI são calculados *offline*. Então, a sintonia que fornece a melhor recompensa a partir do algoritmo ARS é usada para avaliar o controlador PI projetado.

As métricas utilizadas para avaliação dos resultados são o IAE (integral do valor absoluto do erro) e o SII (Soma dos incrementos absolutos da entrada), respectivamente, dados por:

$$IAE = \sum_{t=1}^{T} |e(t) - e(t-1)|.$$

$$SII = \sum_{t=1}^{T} |u(t) - u(t-1)|.$$

3.2.1. Comparativo entre pontos de operações iguais

A sintonia do controlador PI usada como *benchmark* foi proposta por Hedengren (2021) no ponto de operação entre 300 e 320 K. O autor utilizou o método de sintonia de Controle por

Modelo Interno (IMC) para definir os parâmetros do controlador PI. Os parâmetros de sintonia proposto por Hedengren (2021) e os parâmetros definidos pelo algoritmo ARS para o mesmo ponto de operação são apresentados na Tabela 3.2.

Tabela 3.2: Parâmetros de sintonia do controlador para o processo CSTR.

	Referência	K_p	K_i
Hedengren	$300-320$	4,62	40,44
ARS	$300-320$	50,39	7,37

Fonte: O autor.

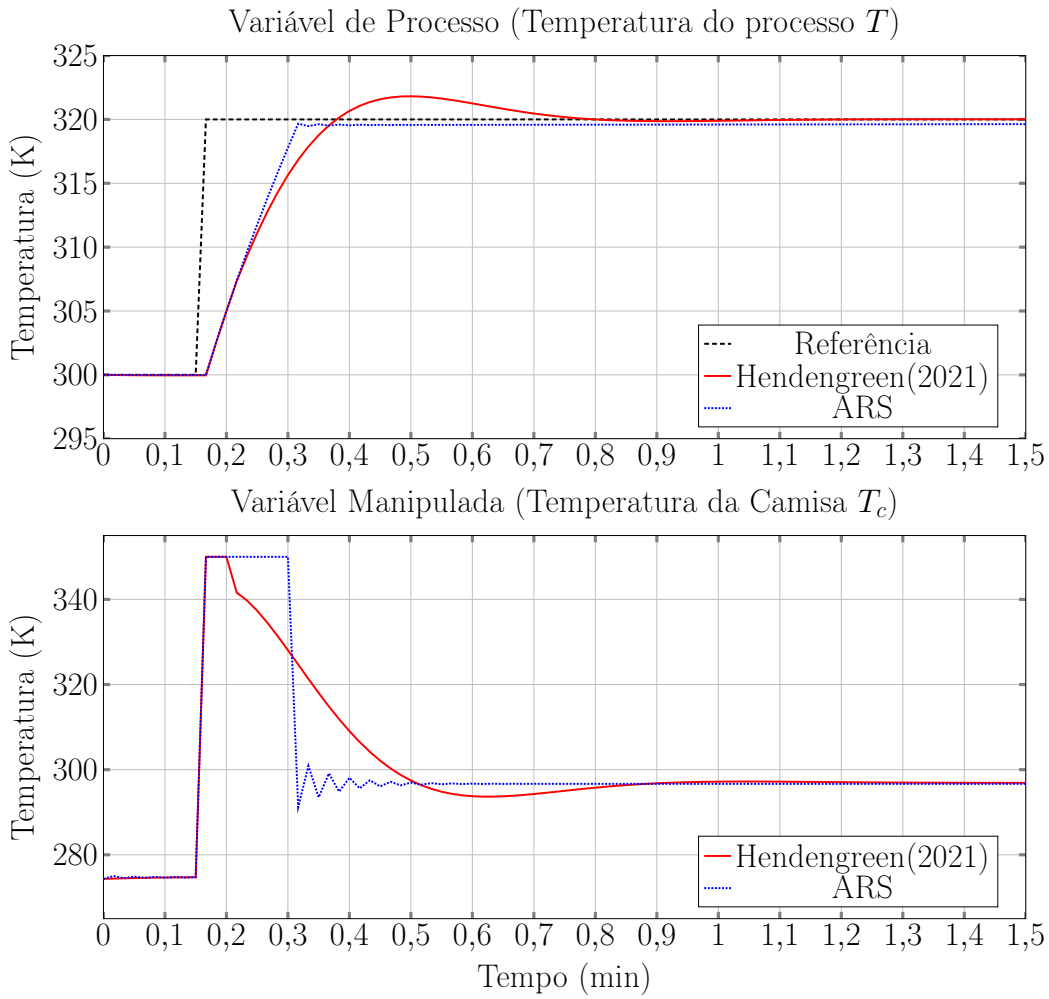

Figura 3.3: PV e MV do processo CSTR.

Fonte: O autor.

A Figura 3.3 apresenta a saída do processo (T) e ação de controle (T_c) obtidas com as sintonias listadas na Tabela 3.2. O processo CSTR quando controlado por um PI sintonizado por Hedengren (2021) apresentou um sobressinal de 9% ($t = 0{,}5min$), tempo de subida (t_s) de $0{,}17min$, tempo de acomodação (t_{ac}) de $0{,}47min$ e não apresentou erro significante em regime permanente (e_∞), enquanto a sintonia do controlador PI obtida a partir do algoritmo ARS não

apresentou sobressinal, tempo de subida de 0,12min, tempo de acomodação de 0,15min e erro em regime permanente de 0,4K. A variável manipulada T_c apresentou saturação em ambas as sintonias. Porém, a sintonia de Hedengren (2021) apresentou 2,4 segundos de saturação enquanto a sintonia proposta pelo algoritmo ARS apresentou 8,4 segundos de saturação. O tempo maior de saturação da sintonia proposta pela ARS está relacionada com a escolha da função de recompensa. Uma vez que o objetivo de controle escolhido foi rastrear a variação de referência o mais rápido.

A MV apresentou maior variação devido ao elevado valor do K_p definido pelo ARS, o que reflete numa ação de controle com maior variação. Um comparativo entre os índices são apresentados na Tabela 3.3.

Tabela 3.3: Índices de Desempenho.

Método	IAE	SII	%OS	$t_s(min)$	$t_{ac(5\%)}(min)$	e_∞
Hedengren	136,50	631,9	9	0,17	0,47	0,06
ARS	125,11	570,3	–	0,12	0,15	0,4

Fonte: O autor.

De acordo com a Tabela 3.3, os índices de desempenho obtidos considerando o intervalo de simulação proposto com o algoritmo ARS foram 8,3% (IAE) e 9,75% (SII) melhores que os obtidos pela sintonia de Hedengren (2021). O algoritmo ARS apresentou melhores tempos de acomodação e subida.

3.2.2. Avaliação do ARS em diferentes pontos de operação

Agora, avaliamos o desempenho da sintonia do controlador PI obtida a partir do algoritmo ARS em diferentes pontos de operação. O seguinte conjunto de referências foi usado:

$$Ref = [300,\ 320,\ 330,\ 340,\ 350].$$

Tabela 3.4: Sintonia proposta pelo ARS.

Referência	K_p	K_i
$300-320$	50,39	7,37
$320-330$	52,40	30,21
$330-340$	21,47	27,06
$340-350$	36,79	56,52

Fonte: O autor.

O conjunto de ações, ou seja, as sintonias do controlador PI para cada referência são apresentadas na Tabela 3.4. Os valores das variáveis de processo e manipulada obtidos com as sintonias são apresentados na Figura 3.4.

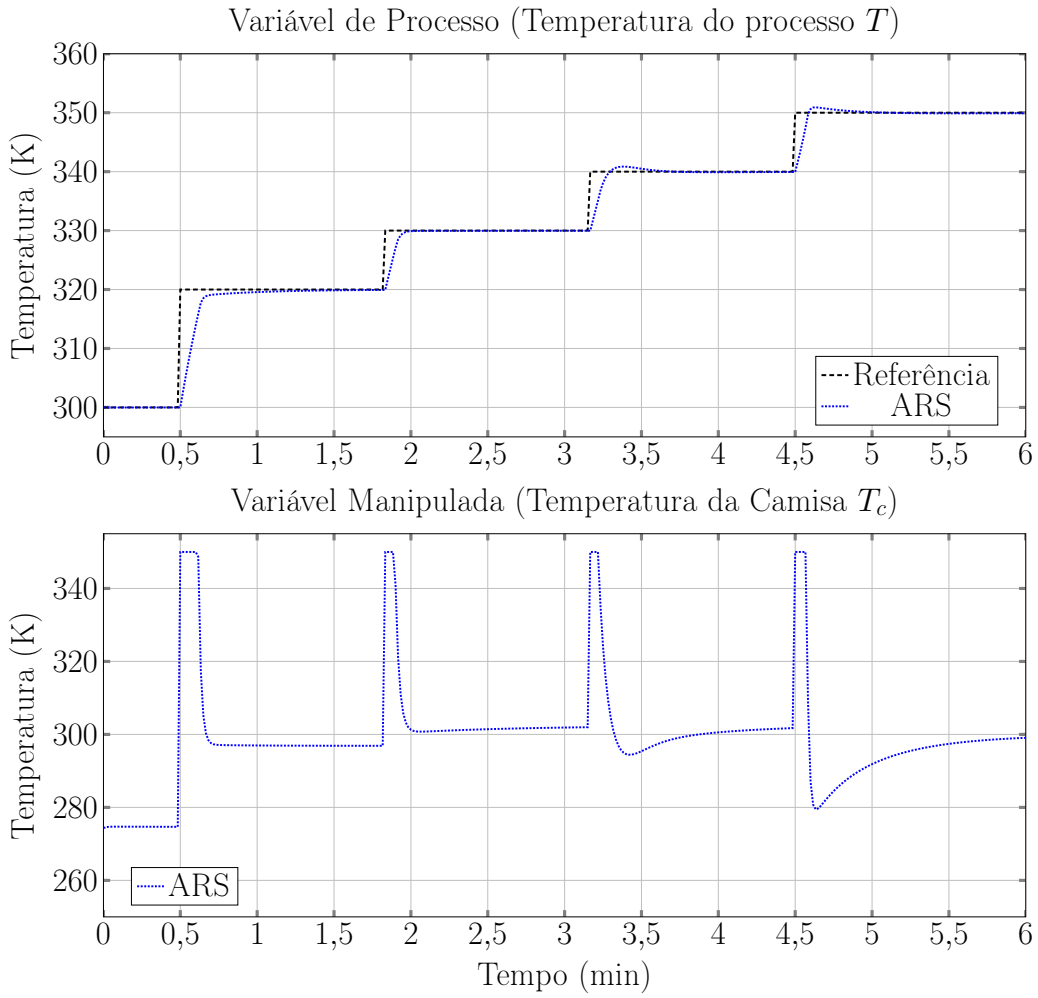

Figura 3.4: Variáveis de entrada e saída do processo CSTR em vários pontos de operação.
Fonte: O autor.

As sintonias propostas pelo algoritmo ARS apresentaram baixos sobressinais, apenas nas variações das referências de 330 para 340 e 340 para 350. A variável manipulada (T_c) apresentou saturação em todos pontos de operação, porém apresentou uma dinâmica suave em todos os pontos de operação, principalmente na última referência aplicada. Destacamos a troca de referência de 320 para 330 como a que apresentou os melhores índices de desempenho. Esta não apresentou sobressinal e um erro em regime permanente quase nulo. ainda apresentou a menor variabilidade em relação as outras. Os demais índices de desempenho são apresentados na Tabela 3.5.

Tabela 3.5: Índices de Desempenho.

Referência	IAE	SII	%OS	$t_s(min)$	$t_{ac(5\%)}(min)$	e_∞
$300-320$	121,2	128,4	–	0,14	0,18	0,06
$320-330$	38,1	103,6	–	0,08	0,12	0,02
$330-340$	50,6	110,9	8,4	0,08	0,34	0,05
$340-350$	49,0	138,4	9,1	0,06	0,26	0,09

Fonte: O autor.

3.3. Estrutura de controle proposta para o espessador

O espessador simulado é um processo não linear e multivariável. Assim, neste trabalho, propomos um diagrama de controle SISO em que o ARS manipule a vazão de *underflow* (MV) e controle a densidade do *underflow* (PV). O diagrama de controle é apresentado na Figura 3.5.

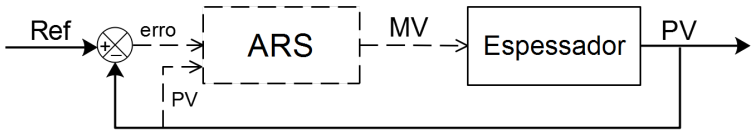

Figura 3.5: Diagrama de blocos do sistema de controle do espessador.
Fonte: O autor.

Nesse caso, o algoritmo ARS é o agente que observa os estados (erro e a densidade do *underflow* (ϕ_u)) e baseado na sua política atua diretamente na vazão de *underflow* (Q_u). Um episódio de iteração do ARS com o processo foi definido como uma alteração da referência do processo, de 0,345 para 0,36 durante um tempo de 18 horas. No início de cada episódio, o processo é redefinido para as suas condições iniciais. A cada passo do episódio, o ARS observa os estados e propõe para esse estado uma ação (valor de vazão do *underflow*). Então, o ambiente processa essa ação e retorna uma recompensa na forma de um valor numérico e o próximo conjunto de estado, assim o ciclo se reinicia conforme é apresentado na Figura 2.4. Uma vez que o objetivo de controle é melhorar o rastreamento da referência, define-se a função de recompensa em função do erro entre a referência e variável controlada e da variação da variável manipulada. Dessa forma, a função de recompensa é dada por:

$$Reward = -|erro_t| - 0{,}1 \times |Q_u(t) - Q_u(t-1)|$$

Observe que a escolha da função de recompensa interfere nas especificações da resposta do sistema de controle, pois indica o quão bom foi a ação de controle obtida a partir do algoritmo. O valor máximo de recompensa será quando o erro for nulo e não houver variação no sinal de controle, sendo portanto, esse valor igual a zero. Isso significará que o ARS está definindo a melhor ação de controle para o respectivo estado. Após inúmeras execuções completas,

os melhores hiperparâmetros foram escolhidos por tentativa e erro e são apresentados na Tabela 3.6. Os parâmetros do processo simulado foram apresentados na Tabela 2.2.

Tabela 3.6: Resumo dos hiperparâmetros do algoritmo ARS.

Hiperparâmetros	Valor
Estados	$[\phi_{u_n}, erro_n]$
Ações	$[Q_{u_n}]$
Recompensa	$-\lvert erro_t\rvert - 0{,}1 \times \lvert Q_u(t) - Q_u(t-1)\rvert$
Taxa de aprendizagem	0,1
Ruído de exploração	0,05
N Direções	10
b Melhores Direções	6
Iterações	400
Passos por episódio	1893

Fonte: O autor.

O algoritmo em *MATLAB*, desenvolvido por Pereira *et al.* (2020b) e Pereira *et al.* (2020c) para simulação do espessador, foi adaptado para permitir a implementação da estratégia de controle baseada em ARS.

3.4. ARS aplicado ao controle de um espessador

Nessa seção apresentamos os resultados da estratégia de controle proposta para o espessador. A vazão de entrada foi mantida em $487{,}8 m^3/h$ e a dosagem de floculante em $8{,}1 g/ton$. Neste estudo de caso, para um episódio, executamos o simulador por 1893 passos, nesse intervalo foi feita a troca de referência de 0,643 para 0,650 interagindo a todo instante com o algoritmo ARS. A execução completa foi de 5400 episódios pelo algoritmo ARS e armazenada a recompensa obtida em cada episódio. O gráfico com o histórico de aprendizagem é apresentado na Figura 3.6. A melhor recompensa foi obtida na iteração 3291 com o valor de $-1{,}3735$ e o tempo de execução de aproximadamente 24 horas.

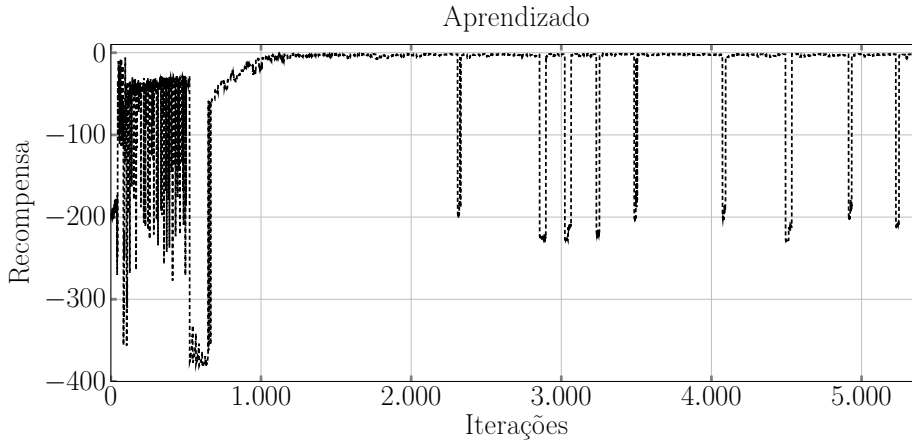

Figura 3.6: Aprendizado do ARS aplicado ao controle de um espessador.
Fonte: O autor.

Um controlador PI foi projetado com objetivo de comparar com os resultados do ARS. A função de transferência em malha aberta do modelo foi identificada utilizado o método da curva de reação e é apresentada na Equação 3.1. O método de sintonia do PI foi o IMC proposto por Rivera *et al.* (1986) com $\lambda = \frac{2}{3} \cdot \tau = 566{,}02$, e os parâmetros encontrados foram $K_p = -1{,}063$ e $T_i = 848{,}03$. A dinâmica do processo para o caso do controlador ARS e para o controlador PI são mostrados na Figura 3.7.

$$F(s) = \frac{1{,}4}{849{,}03s + 1} \tag{3.1}$$

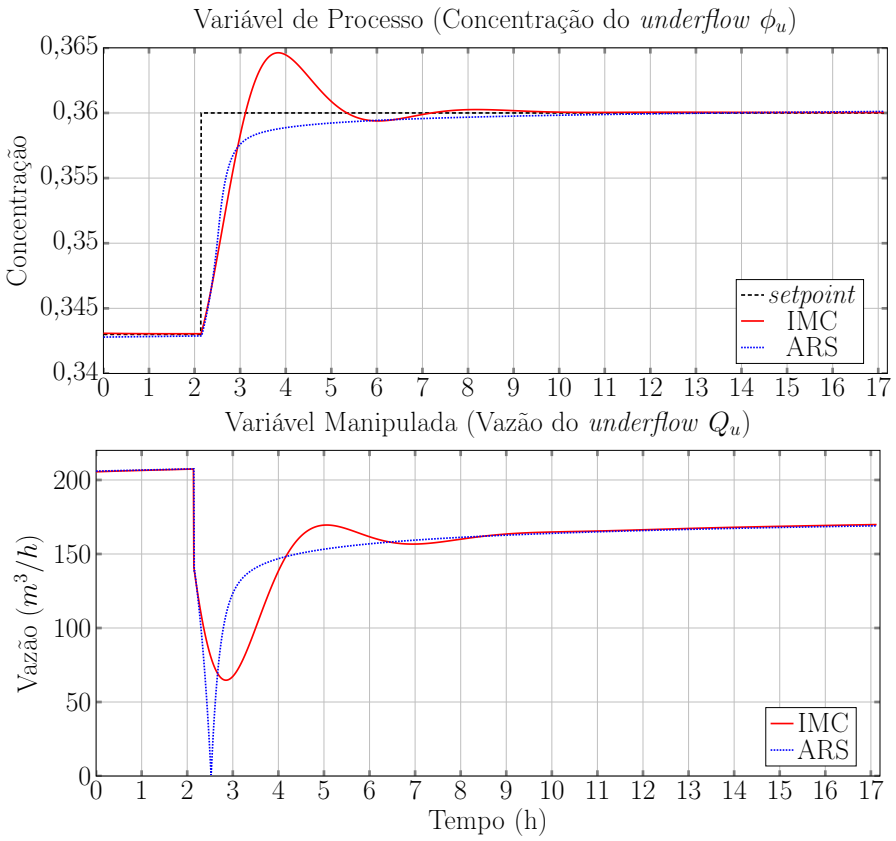

Figura 3.7: Variáveis de entrada e saída do Espessador.
Fonte: O autor.

O controlador ARS apresentou um tempo de subida de 0,88h e tempo de acomodação de 2,58h sem sobressinal, enquanto o controlador PI apresentou um tempo de subida de 0,74h, tempo de acomodação de 2,86h e 27,1% de sobressinal. Ambas as estratégias não apresentaram erro em regime permanente. O SII foi maior para o ARS, isso se dá pela função de recompensa valorizar mais um menor erro do que a variabilidade da MV. A altura da interface é um parâmetro importante e foi observado que não apresentou mudanças bruscas. Os índices de desempenho são apresentados na Tabela 3.7.

Tabela 3.7: Índices de Desempenho.

Método	IAE	SII	%OS	$t_s(h)$	$t_{ac(5\%)}(h)$
IMC	1,75	275,43	27,1	0,74	2,86
ARS	1,42	376,55	–	0,88	2,58

Fonte: O autor.

De acordo com a Tabela 3.7, o algoritmo ARS foi 19% (IAE) melhor que o obtido pelo controlador PI sintonizado por IMC e o controlador PI foi 27% (SII) melhor que o obtido pelo ARS. O algoritmo ARS apresentou melhor tempo de acomodação ($t_{ac(5\%)}$) e o controlador PI melhor tempo de subida (t_s).

4. Conclusões

Neste trabalho foram propostas duas estratégias de controle baseadas em aprendizagem por reforço. Na primeira estratégia propomos sintonizar um controlador PI de um processo CSTR simulado considerando as referências como estados, já a segunda estratégia diz respeito ao controle direto da vazão de um espessador simulado, considerando como estados a densidade e a vazão de *underflow*. A partir dos resultados, de simulação observamos que o algoritmo ARS aprendeu a sintonizar o PI para o rastreamento da referência proposto. Além disso, a MV apresentou menor variabilidade que a sintonia usada como *benchmark*. O ARS também se demonstrou eficaz no controle direto do espessador, com um tempo de acomodação inferior em $0{,}3h$ sem apresentar sobressinal. O uso do algoritmo ARS apresenta como principal vantagem a capacidade de aprender de forma independente do usuário a tarefa de sintonizar um controlador, bem como controlar diretamente um processo. As suas desvantagens são as inúmeras iterações, o que implica na necessidade de alto poder de processamento, um tempo considerável para o pleno aprendizado e a dificuldade na definição dos seus hiperparâmetros.

Ao início deste trabalho, algumas questões foram levantadas. Nesta seção as respostas são apresentadas a seguir:

1. A técnica de aprendizagem por reforço (ARS) é capaz de aprender a sintonizar um controlador PI do processo CSTR?
   R: Sim, a técnica foi capaz de aprender uma sintonia que resultou em melhores índices que uma técnica de sintonia por IMC, conforme foi apresentado na seção 3.2. Além disso, foi capaz de sintonizar o controlador PI para outros pontos de operação em um mesmo episódio conforme é apresentado na seção 3.2.2.

2. A técnica de aprendizagem por reforço (ARS) é capaz de aprender a controlar diretamente um espessador?
   R: Sim, conforme foi apresentado na seção 3.4, a técnica apresentou melhores índices de desempenho que um controlador PI sintonizado por IMC.

4.1. Trabalhos Futuros

Para os trabalhos futuros deseja-se:

1. Avaliar os controladores propostos obtidos usando ARS em uma condição de operação similar a encontrada no processo real, pretendemos aplicar perturbações na vazão de entrada e na adição de floculante.

2. Analisar diferentes formulações da função de recompensa considerando variáveis do processo diretamente relacionadas com o desempenho do controlador.

3. Avaliar os controladores propostos obtidos usando ARS com a saída do processo sujeita a ruído de medição.

4. Utilizar algoritmos de busca, por exemplo, algoritmo genético, para definir os melhores hiperparâmetros do algoritmo ARS.

5. Considerar a variável de controle na formulação da recompensa do caso do processo CSTR.

6. Implementar outras técnicas de aprendizagem por reforço, como o *Q-learning*.

4.2. Trabalhos produzidos

Os seguintes trabalhos foram gerados durante o mestrado.

Aceito no Congresso Brasileiro de Automática (2022):

BITARÃES, S., MOISES, S., EUZÉBIO, T. "Sintonia de Controladores PI baseada em *Augmented Random Search*: Estudo de Caso do Processo CSTR". 10 2022.

Submetido na revista *Minerals Engineering*:

MOISES, S., YAMASHITA, A., BITARÃES, S., EUZÉBIO, T., "Integrated crushing circuit control algorithm based on finite state machine".

Publicado no Simpósio Brasileiro de Automação Inteligente (2021):

PACÍFICO, W., MOISES, S., BITARÃES, S., PINTO, T., EUZÉBIO, T. "Sistema de Controle usando Máquina de Estado Finita para uma Planta de Beneficiamento de Minério". 10 2021.

Referências Bibliográficas

ADAM, S., BUSONIU, L., BABUSKA, R. "Experience Replay for Real-Time Reinforcement Learning Control", *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, v. 42, n. 2, pp. 201–212, 2012. doi: 10.1109/TSMCC.2011.2106494.

ANTONELLI, R., ASTOLFI, A. "Continuous stirred tank reactors: easy to stabilise?" *Automatica*, v. 39, n. 10, pp. 1817–1827, 2003.

BRUJENI, L. A., LEE, J. M., SHAH, S. L. *Dynamic tuning of PI-controllers based on model-free reinforcement learning methods*. s, IEEE, 2010.

CHAVES, A. P. *Teoria e Prática do Tratamento de Minérios*. São Paulo, Signus Editora, 2004.

HALLÉN, M., ÅSTRAND, M., SIKSTRÖM, J., et al.. "Reinforcement Learning for Grinding Circuit Control in Mineral Processing". pp. 488–495, Zaragoza, Spain, 2019. doi: 10.1109/ETFA.2019.8869212.

HEDENGREN, J. D. "Temperature Control of a Stirred Reactor". Disponível em: `https://apmonitor.com/pdc/index.php/Main/StirredReactor`. Acesso em: 19 de abril 2022, 2021.

HOWELL, M., BEST, M. "On-line PID tuning for engine idle-speed control using continuous action reinforcement learning automata", *Control Engineering Practice*, v. 8, n. 2, pp. 147–154, 2000.

HUANG, H.-P., CHAO, Y.-C., CHENG, C.-L. "Identification and Adaptive Control for a CSTR Process". Em: *1982 American Control Conference*, pp. 551–556, 1982. doi: 10.23919/ACC.1982.4787911.

JIANG, Y., FAN, J., CHAI, T., et al.. "Data-Driven Flotation Industrial Process Operational Optimal Control Based on Reinforcement Learning", *IEEE Transactions on Industrial Informatics*, pp. 1974–1989, 2018. doi: 10.1109/TII.2017.2761852.

KOSZAKA, L., RUDEK, R., POZNIAK-KOSZALKA, I. "An idea of using reinforcement learning in adaptive control systems". Em: *International Conference on Networking,*

International Conference on Systems and International Conference on Mobile Communications and Learning Technologies (ICNICONSMCL'06), pp. 190–190. IEEE, 2006.

KUMAR, U., SHARMA, V., RAHI, O. P., et al.. "MPC-Based Temperature Control of CSTR Process and Its Comparison with PID". Em: Sengodan, T., Murugappan, M., Misra, S. (Eds.), *Advances in Electrical and Computer Technologies*, pp. 1109–1115, Singapore, 2020. Springer Singapore.

LI, Z., XUE, S., LIN, W., et al.. "Training a robust reinforcement learning controller for the uncertain system based on policy gradient method", *Neurocomputing*, pp. 313–321, 2018. doi: 10.1016/j.neucom.2018.08.007.

LILLICRAP, T. P., HUNT, J. J., PRITZEL, A., et al.. "Continuous control with deep reinforcement learning", *arXiv preprint arXiv:1509.02971*, 2015.

LOPES JR., Ê., DA SILVA, M. T., EUZÉBIO, T. A. "Avoiding Buffer Tank Overflow in an Iron Ore Dewatering System with Integrated Control System", *Sustainability*, v. 14, n. 15, pp. 9347, 2022.

LUZ, A. B. D., LINS, F. A. F. "Introdução ao tratamento de minérios". CETEM/MCTIC, 2018.

MAGALHÃES, S., EUZÉBIO, T. "Supervisory Fuzzy Controller for Thickener Underflow Solids Concentration on a Simulated Platform". 06 2018.

MANIA, H., GUY, A., RECHT, B. "Simple random search provides a competitive approach to reinforcement learning", *ArXiv*, v. abs/1803.07055, 2018a.

MANIA, H., GUY, A., RECHT, B. "Simple random search provides a competitive approach to reinforcement learning". 2018b.

PEREIRA, A., MARTINS, W., MOREIRA, V., et al.. "Aplicação de Controle PI e DMC Multivariável em Espessadores de Minério de Ferro". 12 2020a. doi: 10.48011/asba.v2i1.1739.

PEREIRA, A. M. F., MARTINS, W. T., MOREIRA, V. S., et al.. "Aplicação de Controle PI e DMC Multivariável em Espessadores de Minério de Ferro", *Congresso Brasileiro de Automática 2020*, 2020b. doi: 10.48011/asba.v2i1.1739.

PEREIRA, A. M. F. *Simulação Dinâmica e Controle de Nível da Interface e Concentração de Sólidos na Descarga de Espessadores Sujeitos à Adição de Floculante*. Tese de Mestrado, Universidade Federal de Ouro Preto, Ouro Preto, MG, 12 2021.

PEREIRA, A. M., MARTINS, W. T., MOREIRA, V. S., et al.. "Aplicação de Controle PI e DMC Multivariável em Espessadores de Minério de Ferro". Em: *Congresso Brasileiro de Automática-CBA*, v. 2, 2020c.

RAJESWARAN, A., LOWREY, K., TODOROV, E., et al.. "Towards Generalization and Simplicity in Continuous Control". 2018.

REDDY, B. R., RAM, K. T., PRANAVANAND, S. "Adaptive PI Controller Design and Deployment for Chemical Reactor". Em: *2020 International Conference on Smart Electronics and Communication (ICOSEC)*, pp. 1210–1215, 2020. doi: 10.1109/ICOSEC49089.2020.9215456.

RIVERA, D. E., MORARI, M., SKOGESTAD, S. "Internal model control: PID controller design", *Industrial & Engineering Chemistry Process Design and Development*, v. 25, n. 1, pp. 252–265, 1986. doi: 10.1021/i200032a041.

SALIMANS, T., HO, J., CHEN, X., et al.. "Evolution Strategies as a Scalable Alternative to Reinforcement Learning". 2017a.

SALIMANS, T., HO, J., CHEN, X., et al.. "Evolution strategies as a scalable alternative to reinforcement learning", *arXiv preprint arXiv:1703.03864*, 2017b.

SHIPMAN, W. J., COETZEE, L. C. "Reinforcement Learning and Deep Neural Networks for PI Controller Tuning", *IFAC-PapersOnLine*, pp. 111–116, 2019. doi: 10.1016/j.ifacol.2019.09.173.

SPIELBERG, S. P. K., GOPALUNI, R. B., LOEWEN, P. D. "Deep reinforcement learning approaches for process control", *6th International Symposium on Advanced Control of Industrial Processes (AdCONIP)*, pp. 201–206, 2017. doi: 10.1109/ADCONIP.2017.7983780.

SPIELBERG, S., TULSYAN, A., LAWRENCE, N. P., et al.. "Toward self-driving processes: A deep reinforcement learning approach to control", *AIChE Journal*, v. 65, n. 10, Jun 2019.

SUTTON, R. S., BARTO, A. G. *Reinforcement learning: An introduction*. London, England, MIT press, 2018.

WANG, X.-S., HU CHENG, Y., SUN, W. "A Proposal of Adaptive PID Controller Based on Reinforcement Learning", *Journal of China University of Mining and Technology*, v. 17, n. 1, pp. 40–44, 2007.

WEI, T., WANG, Y., ZHU, Q. "Deep Reinforcement Learning for Building HVAC Control", *Association for Computing Machinery*, 2017. doi: 10.1145/3061639.3062224.

WIERING, M., VAN OTTERLO, M. *Reinforcement Learning: State-of-the-Art*. Berlin, Heidelberg, Springer, 2012.

ZIEGLER, J. G., NICHOLS, N. B., OTHERS. "Optimum settings for automatic controllers", *trans. ASME*, v. 64, n. 11, 1942.