

# Validação de arquitetura de controle avançado em ambiente de simulação dinâmica integrada via OPC UA

Julián Gutiérrez<sup>a,\*</sup>, Thiago Vaz da Costa<sup>b</sup>, Heleno Bispo<sup>c</sup>

<sup>a</sup> Departamento de Ingeniería Eléctrica y Computadoras, ICIC-CONICET, Universidad Nacional del Sur, Bahía Blanca-BA, Argentina

<sup>b</sup> Faculdade de Engenharia Química, Universidade Federal de Uberlândia, Uberlândia-MG, Brasil

<sup>c</sup> Departamento de Engenharia Química, Universidade Federal de Campina Grande, Campina Grande-PB, Brasil

\* [julian.gutierrez@eq.ufcg.edu.br](mailto:julian.gutierrez@eq.ufcg.edu.br)

## RESUMO

Este trabalho apresenta uma arquitetura de simulação e controle baseada no protocolo de comunicação industrial OPC UA, com foco na implementação de um Controle Preditivo Baseado em Modelo (MPC) multivariável  $2 \times 2$  para controle de nível e pressão em um vaso de separação bifásica. A arquitetura integra três componentes que se comunicam exclusivamente via OPC UA: o servidor nativo do simulador AVEVA Dynamic Simulation (DYN-SIM), responsável pela modelagem dinâmica do processo; um servidor OPC UA desenvolvido em Python que emula um Controlador Lógico Programável (CLP) com múltiplos algoritmos de controle, incluindo o MPC; e uma interface gráfica centralizada (*GUI hub*) que atua como cliente OPC UA, provendo monitoramento, interação manual e sincronização bidirecional de variáveis entre os servidores. O controlador MPC MIMO foi projetado segundo a formulação incremental de Wang, com modelos de planta identificados via dados de malha aberta coletados do DYN-SIM. Os resultados demonstram seguimento de setpoint satisfatório para as variáveis controladas validando a viabilidade da arquitetura proposta como plataforma de simulação em malha fechada aderente aos princípios O-PAS.

**Palavras-chave:** OPC UA; Controle Preditivo (MPC); Simulação em Malha Fechada; AVEVA Dynamic Simulation; Automação Industrial.

## 1 Introdução

A crescente adoção de padrões de interoperabilidade na automação industrial tem impulsionado o desenvolvimento de arquiteturas de controle abertas e modulares. Nesse contexto, o protocolo OPC UA (*Open Platform Communications Unified Architecture*) consolidou-se como referência para comunicação segura, orientada a serviços e independente de plataforma entre dispositivos e sistemas de supervisão (Mahnke et al., 2009). Paralelamente, o padrão O-PAS (*Open Process Automation Standard*) promove a desagregação de sistemas de controle distribuído (DCS) tradicionais em componentes interoperáveis baseados justamente em OPC UA como camada de comunicação (Open Process Automation Forum, 2022).

Do ponto de vista do controle avançado, o Controle Preditivo Baseado em Modelo (MPC) é amplamente reconhecido como a estratégia de controle multivariável de maior penetração na indústria de processos (Qin & Badgwell, 2003). Sua capacidade de tratar explicitamente restrições de processo, acoplamentos entre variáveis e horizontes de predição configuráveis torna-o especialmente adequado para unidades de separação, reatores e colunas de destilação, onde as interações entre variáveis são inerentes.

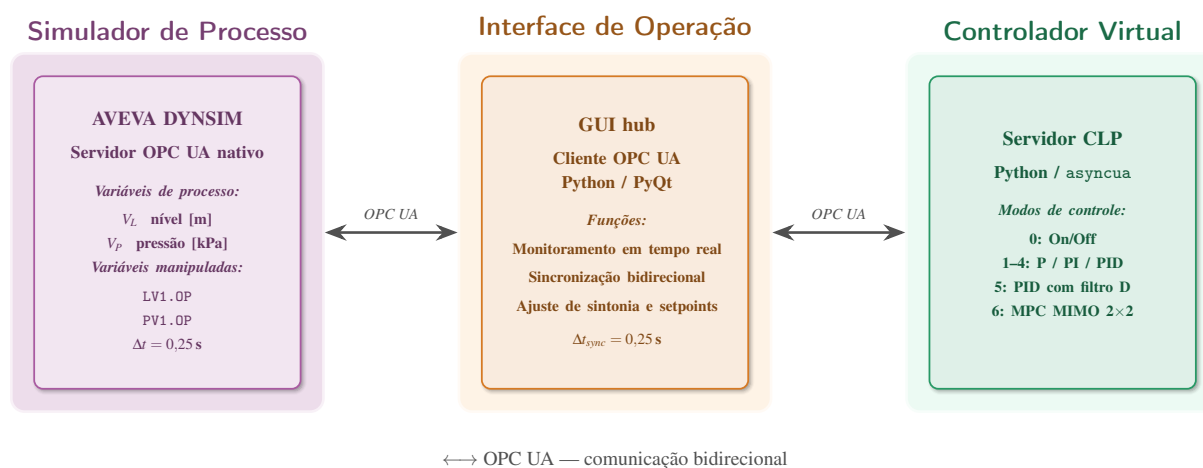
Apesar da maturidade individual de ambas as tecnologias, poucos trabalhos documentam arquiteturas que integrem MPC multivariável diretamente sobre OPC UA em plataformas de simulação em malha fechada com simuladores de alta fidelidade. A maioria das implementações de MPC industrial opera sobre protocolos proprietários (Modbus, PROFIBUS, redes DCS nativas), limitando a portabilidade e a capacidade de validação cruzada com simuladores de processo. Trabalhos recentes demonstram o uso de simuladores de alta fidelidade como plataforma de validação para MPC, tanto em ferramentas baseadas em gPROMS (Wu & Nauta, 2022) quanto em formulações multi-modelo para processos com saídas integradoras (Perez et al., 2014). Este trabalho complementa essa linha ao propor, implementar e validar uma arquitetura de malha fechada MPC-OPC UA completa sobre o simulador AVEVA DYN-SIM, com ênfase na interoperabilidade nativa via protocolo aberto.

## 2 Metodologia

### 2.1 Arquitetura geral do sistema

A arquitetura proposta é composta por três elementos que se comunicam exclusivamente via OPC UA, conforme ilustrado na figura 1. O primeiro elemento é o servidor OPC UA do DYNSIM, que expõe o espaço de endereçamento da simulação dinâmica do vaso de separação — incluindo variáveis de processo, sinais de controle e parâmetros do modelo. O segundo elemento é o servidor OPC UA desenvolvido em Python com a biblioteca *asyncua*, que implementa a lógica de controle de um CLP industrial em software. O terceiro elemento é a interface gráfica (*GUI hub*), também desenvolvida em Python com PyQt, que atua como cliente OPC UA simultâneo a ambos os servidores, provendo monitoramento em tempo real, comandos manuais e sincronização bidirecional de variáveis via uma camada de mapeamento configurável pelo operador.

**Figura 1:** Arquitetura de comunicação OPC UA proposta.



O servidor CLP disponibiliza seis modos de controle: On/Off, P, PI, PID, PID com filtro derivativo e MPC. A seleção do modo, os parâmetros de sintonia e os setpoints são todos acessíveis como nós do espaço de endereçamento OPC UA, podendo ser modificados em tempo real pela GUI ou por qualquer cliente OPC UA externo — o que garante conformidade com os princípios de sistema aberto do O-PAS.

### 2.2 Planta simulada

O processo simulado no DYNSIM é um vaso de separação bifásica operando a pressão elevada. As variáveis de processo controladas são o nível do líquido ( $V_L$ , em metros) e a pressão do vaso ( $V_p$ , em kPa). As variáveis manipuladas correspondentes são as aberturas da válvula de nível (LV1.OP) e da válvula de controle de pressão (PV1.OP), ambas no intervalo [0, 1] (fração de abertura). O vaso configura um processo MIMO 2×2.

A temperatura é mantida em ponto de operação fixo em 290 K. Experimentos preliminares indicaram forte acoplamento entre os canais de pressão e temperatura: variações na válvula de aquecimento perturbam significativamente a pressão do vaso, e os controladores competem entre si de forma que o sistema 3×3 apresenta instabilidade prática. Esse fenômeno é inerente à termodinâmica de processos flash, onde pressão e temperatura são variáveis de estado acopladas pela equação de equilíbrio de fases.

A dinâmica do processo apresenta duas características relevantes para a síntese do controlador: (i) o nível comporta-se como processo integrador, pois a variação de nível é proporcional ao desvio da válvula em relação ao ponto de equilíbrio; e (ii) a pressão responde como processo de primeira ordem com ganho negativo (abertura crescente da válvula alivia a pressão).

### 2.3 Identificação do modelo de planta

Os modelos de planta foram identificados a partir de dados de malha aberta coletados diretamente do DYNSIM, via experimentos de resposta ao degrau em três configurações distintas:

- **Experimento nível:** excitação de LV1.OP → identificação de  $G_{11}$  e  $G_{21}$ .

- **Experimento pressão:** excitação de PV1.OP → identificação de  $G_{22}$  e  $G_{12}$ .

Os parâmetros foram estimados por mínimos quadrados não-lineares sobre a resposta analítica do modelo correspondente (integrador ou FOPDT).

Para o canal de nível ( $G_{11}$ ), o modelo integrador discreto resultante é:

$$y_L(k+1) = y_L(k) + K_i(u_1(k) - u_{eq}) \Delta t, \quad (1)$$

com  $K_i = -0,0733$  m/(s·fração),  $u_{eq} = 0,269$  e  $\Delta t = 0,25$  s. O Erro Quadrático Médio Raiz Normalizado (NRMSE) de validação foi inferior a 2,7%.

Para o canal de pressão ( $G_{22}$ ), o modelo FOPDT resultante é:

$$G_{22}(s) = \frac{K_{22}}{\tau_{22}s + 1}, \quad K_{22} = -5677 \text{ kPa/fração}, \quad \tau_{22} = 7,56 \text{ s}, \quad (2)$$

com NRMSE entre 1,3% e 5,8%.

O acoplamento  $G_{21}$  (efeito de LV1.OP sobre a pressão,  $K_{21} = -308$  kPa/fração) foi incluído no modelo do controlador por satisfazer o critério  $|K_{21}/K_{22}| = 5,4\% > 5\%$ . Os demais acoplamentos foram considerados desprezáveis.

A matriz de transferência adotada no controlador é:

$$\mathbf{G}(s) = \begin{bmatrix} G_{11}(s) & 0 \\ G_{21}(s) & G_{22}(s) \end{bmatrix}. \quad (3)$$

#### 2.4 Formulação do controlador MPC

O controlador MPC foi implementado segundo a formulação incremental de Wang (Wang, 2009), com modelo aumentado em espaço de estados:

$$\mathbf{x}_a(k+1) = \mathbf{A}_a \mathbf{x}_a(k) + \mathbf{B}_a \Delta \mathbf{u}(k), \quad \mathbf{y}(k) = \mathbf{C}_a \mathbf{x}_a(k), \quad (4)$$

onde  $\mathbf{x}_a = [\Delta \mathbf{x}^\top, \mathbf{y}^\top]^\top \in \mathbb{R}^{n_x+n_y}$  e  $\Delta \mathbf{u}(k) = \mathbf{u}(k) - \mathbf{u}(k-1)$ . As matrizes aumentadas são:

$$\mathbf{A}_a = \begin{bmatrix} \mathbf{A} & \mathbf{0} \\ \mathbf{C}\mathbf{A} & \mathbf{I} \end{bmatrix}, \quad \mathbf{B}_a = \begin{bmatrix} \mathbf{B} \\ \mathbf{C}\mathbf{B} \end{bmatrix}, \quad \mathbf{C}_a = [\mathbf{0} \quad \mathbf{I}]. \quad (5)$$

Para o sistema  $2 \times 2$  com  $n_x = n_y = 2$ , o estado aumentado tem dimensão  $n_{x_a} = 4$ . A matriz  $\mathbf{A}$  é diagonal com  $a_{11} = 1$  (polo integrador) e  $a_{22} = e^{-\Delta t/\tau_{22}}$ . A matriz  $\mathbf{B}$  inclui o acoplamento identificado:

$$\mathbf{B} = \begin{bmatrix} B_{11} & 0 \\ B_{21} & B_{22} \end{bmatrix}. \quad (6)$$

As variáveis de processo foram normalizadas para  $[0, 1]$  com ranges:  $[0, 3]$  m (nível) e  $[0, 10.000]$  kPa (pressão).

O problema de otimização a cada instante  $k$  e:

$$\min_{\Delta \mathbf{u}} \sum_{i=1}^{H_p} \|\mathbf{y}(k+i|k) - \mathbf{r}(k+i)\|_{\mathbf{Q}_y}^2 + \sum_{j=0}^{H_c-1} \|\Delta \mathbf{u}(k+j)\|_{\mathbf{Q}_u}^2, \quad (7)$$

sujeito a  $\Delta u_{\min,i} \leq \Delta u_i(k+j) \leq \Delta u_{\max,i}$ , resolvido como QP via *cvxpy* com solver OSQP. Os parâmetros de sintonia estão resumidos na tabela 1.

O horizonte de predição  $H_p = 200$  passos (50 s) cobre  $\approx 6,6 \tau_{22}$ , assegurando que a resposta transitória da pressão seja completamente capturada. O horizonte de controle  $H_c = 2$  é um compromisso deliberado:  $H_c = 1$  não fornecia graus de liberdade suficientes para compensar o acoplamento  $G_{21}$ , enquanto  $H_c \geq 3$  induzia oscilações no canal de nível, que apresenta polo duplo em  $z = 1$  no modelo aumentado. O peso de seguimento da pressão  $q_{y,2} = 25$  é deliberadamente menor que  $q_{y,1} = 100$ : a pressão possui dinâmica mais rápida e tolerância maior a desvios

**Tabela 1:** Parâmetros de sintonia do controlador MPC MIMO 2×2.

Parâmetro	Símbolo	Valor
Horizonte de predição	$H_p$	200 passos
Horizonte de controle	$H_c$	2 passos
Peso seguimento — nível	$q_{y,1}$	100
Peso seguimento — pressão	$q_{y,2}$	25
Peso esforço — nível	$q_{u,1}$	3,0
Peso esforço — pressão	$q_{u,2}$	5,0
$ \Delta u_{\max} $ — nível	$\Delta u_{\max,1}$	0,15 (fração)
$ \Delta u_{\max} $ — pressão	$\Delta u_{\max,2}$	0,1 (fração)
Período de amostragem	$\Delta t$	0,25 s

transitórios, enquanto o nível — canal integrador — requer penalização elevada para evitar deriva acumulada. Os pesos de esforço  $q_{u,1} = 3,0$  e  $q_{u,2} = 5,0$  refletem a maior sensibilidade da válvula de pressão: incrementos excessivos em PV1.OP perturbam o nível via  $G_{21}$ , justificando penalização de esforço mais conservadora nesse canal. Os limites  $\Delta u_{\max,1} = 0,15$  e  $\Delta u_{\max,2} = 0,10$  permitem movimentos de válvula maiores, acelerando a resposta a degraus de setpoint de maior amplitude.

### 2.5 Implementação em Python e integração OPC UA

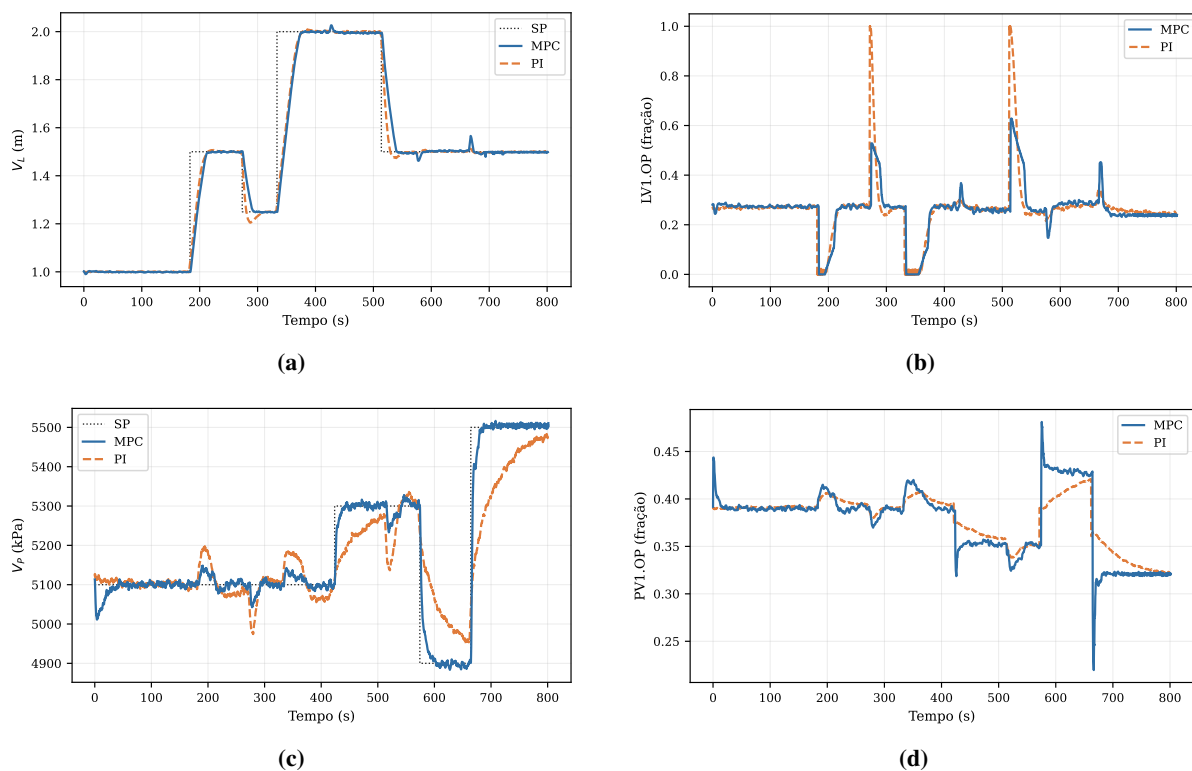
O módulo de controle do servidor CLP é estruturado em três camadas: (i) a classe MPC, que encapsula a formulação quadrática e o solver; (ii) a classe MPCController, que gerencia o vetor de estado aumentado  $\mathbf{x}_a$  e a interface com o loop de controle; e (iii) o loop assíncrono do servidor OPC UA, que lê variáveis de processo da GUI hub (provenientes do DYNMIM), normaliza os valores, invoca o controlador e escreve as ações de controle de volta ao espaço de endereçamento. Todo o fluxo é assíncrono com `asyncio`, compatível com o modelo de servidor `asyncua`. O período de execução do loop de controle é sincronizado com o passo de simulação do DYNMIM em  $\Delta t = 0,25$  s.

## 3 Resultados e Discussão

### 3.1 Desempenho do controlador MPC em malha fechada

O controlador MPC MIMO 2×2 foi avaliado em malha fechada com a simulação do DYNMIM, aplicando uma sequência de degraus de setpoint replicada de forma idêntica nos ensaios com controle PI independente. A sequência cobre quatro degraus de nível (1,0 → 1,5 → 1,25 → 2,0 → 1,5 m) e três degraus de pressão (5100 → 5300 → 4900 → 5500 kPa) ao longo de aproximadamente 800 s, intercalados para exercitar tanto o seguimento de setpoint em cada canal quanto a rejeição do acoplamento  $G_{21}$  entre eles. A figura 2 apresenta as respostas das duas variáveis de processo e das respectivas ações de controle para ambos os controladores.

**Figura 2:** Comparação MPC MIMO  $2 \times 2$  (linha contínua azul) vs. PI independente (linha tracejada laranja) em malha fechada. Esquerda: variáveis de processo e setpoints (linha pontilhada preta). Direita: ações de controle correspondentes.



No canal de nível (figura 2a), ambos os controladores apresentam seguimento de setpoint sem overshoot perceptível e tempos de acomodação comparáveis para os degraus de pequena amplitude. Para o degrau de maior amplitude ( $1,25 \rightarrow 2,0$  m,  $t \approx 273$  s), o MPC apresenta acomodação ligeiramente mais lenta que o PI devido à restrição  $\Delta u_{\max,1} = 0,15$ , que limita a velocidade de movimento da válvula — compromisso deliberado para evitar saturação. A ação de controle do MPC (figura 2b) é visivelmente mais suave que a do PI, característica relevante para a vida útil de atuadores em plantas reais.

No canal de pressão (figura 2c), a diferença entre os controladores é mais expressiva. O PI apresenta picos transitórios cada vez que ocorre um degrau de nível, evidenciando a perturbação introduzida por LV1.OP via  $G_{21}$  — distúrbio que o controlador descentralizado só consegue rejeitar de forma reativa. O MPC, por incorporar  $G_{21}$  explicitamente na matriz  $\mathbf{B}$ , antecipa o efeito do movimento da válvula de nível sobre a pressão e ajusta PV1.OP preventivamente, resultando em desvios transitórios significativamente menores. Esse comportamento é consistente com a teoria de controle multivariável: quando  $|K_{21}/K_{22}|$  não é negligenciável, o controlador descentralizado é subótimo por construção, e estratégias coordenadas como o MPC fornecem ganho de desempenho proporcional ao acoplamento (Qin & Badgwell, 2003).

### 3.2 Discussão comparativa

Os resultados evidenciam ganho assimétrico do MPC sobre o PI: comparável no seguimento de setpoint canal a canal, mas significativamente superior na rejeição do acoplamento  $G_{21}$ . Esse padrão é coerente com a literatura — em plantas onde o desacoplamento é fraco, controladores PI bem sintonizados são competitivos com MPC em seguimento, mas a vantagem do MPC cresce com o grau de acoplamento e com a presença de restrições explícitas (Qin & Badgwell, 2003). A formulação incremental adotada (Wang, 2009) garante ação integral natural sem necessidade de termo integral explícito, simplificando a sintonia e evitando *wind-up*.

Trabalhos anteriores que integram MPC com simuladores de processo seguem majoritariamente uma de

duas abordagens: acoplamento direto via API proprietária da plataforma de simulação, como em Wu e Nauta (2022) sobre gPROMS, ou geração offline de modelos lineares a partir de simulação para implementação em ambiente externo, como em Perez et al. (2014) com EMSO. Ambas as abordagens produzem soluções funcionais, mas vinculam o controlador à plataforma de simulação ou à ferramenta de modelagem. A arquitetura proposta neste trabalho adota uma terceira via: simulador e controlador como componentes independentes que se comunicam exclusivamente via OPC UA. Essa escolha tem três consequências práticas. Primeiro, o controlador é portátil — a mesma lógica de controle pode ser executada contra o simulador, contra um CLP físico ou contra a planta real sem modificação. Segundo, os parâmetros de sintonia ( $H_p$ ,  $H_c$ ,  $q_v$ ,  $q_u$ ) ficam expostos como nós navegáveis do espaço de endereçamento, modificáveis em tempo real por qualquer cliente OPC UA conforme — propriedade diretamente exigida pelo O-PAS para componentes de controle avançado (Open Process Automation Forum, 2022). Terceiro, a arquitetura é *vendor-agnostic*: o substituto do DYNOSIM por outro simulador com servidor OPC UA (Aspen HYSYS, gPROMS, ProMax) ou a substituição do servidor Python por um CLP físico com OPC UA não requerem reescrita do código.

A latência de comunicação medida ( $< 1$  ms em rede local, menos de 0,5% do período de amostragem de 250 ms) confirma a viabilidade do protocolo para o período adotado (Mahnke et al., 2009) e é compatível com tempos de ciclo de CLPs industriais modernos, indicando que a transição da arquitetura para validação Hardware-in-the-Loop com CLP físico não exigiria reformulação da camada de comunicação.

#### 4 Conclusões

Este trabalho validou uma arquitetura de simulação em malha fechada MPC-OPC UA para controle MIMO  $2 \times 2$  de nível e pressão. As principais contribuições são: (i) a integração nativa entre o simulador DYNOSIM e o controlador MPC via OPC UA, sem protocolos proprietários; (ii) a identificação de modelos pelo simulador, caracterizando o nível como integrador e quantificando o acoplamento  $G_{21}$ ; e (iii) a análise da sintonia na estabilidade, focando na influência de  $H_c$  sobre o canal integrador com polo duplo em  $z = 1$ .

Fixar a temperatura devido à forte interação termodinâmica com a pressão limita o escopo atual. Como trabalhos futuros, propõe-se: (a) a validação com CLP físico para avaliar o impacto da latência de rede; (b) o uso de desacoplamento explícito ou restrições para reincorporar o controle de temperatura; e (c) a extensão da comparação MPC-PI sob perturbações de carga.

A arquitetura proposta, inteiramente baseada em componentes de software aberto e no protocolo OPC UA, é diretamente compatível com os princípios O-PAS e representa um passo concreto em direção a sistemas de controle avançado abertos e portáteis para a indústria de processos.

**Agradecimentos:** J. Gutiérrez agradece ao Consejo Nacional de Investigaciones Científicas y Técnicas (CONICET, Argentina). H. Bispo agradece ao Departamento de Engenharia Química da Universidade Federal de Campina Grande (UFCG) e T. Vaz da Costa agradece à Faculdade de Engenharia Química da Universidade Federal de Uberlândia (UFU).

#### Referências

- Mahnke, W., Leitner, S.-H., & Damm, M. (2009). *OPC Unified Architecture*. Springer.
- Open Process Automation Forum. (2022). *Open Process Automation Standard (O-PAS), Version 3.0* (rel. técn. N. C22A). The Open Group. <https://www.opengroup.org/opus>
- Perez, J. M., Odloak, D., & Lima, E. L. (2014). Multi-model MPC with output feedback. *Brazilian Journal of Chemical Engineering*, 31(1), 131–144.
- Qin, S. J., & Badgwell, T. A. (2003). A survey of industrial model predictive control technology. *Control Engineering Practice*, 11(7), 733–764. [https://doi.org/10.1016/S0967-0661\(02\)00186-7](https://doi.org/10.1016/S0967-0661(02)00186-7)
- Wang, L. (2009). *Model Predictive Control System Design and Implementation Using MATLAB*. Springer. <https://doi.org/10.1007/978-1-84882-331-0>
- Wu, L., & Nauta, M. (2022). A rapid-prototype MPC tool based on gPROMS platform [arXiv:2209.00092v2 [eess.SY]]. *arXiv preprint arXiv:2209.00092*.