

# **CONTROLE DE POTÊNCIA COM REALIMENTAÇÃO DE ESTADOS E INTELIGÊNCIA ARTIFICIAL EM CONVERSOR BOOST, VISANDO A ESTABILIDADE DO SISTEMA DE POTÊNCIA EM UM SIMULADOR EM TEMPO REAL**

Eduardo Rochinski Arbighaus, Rodrigo Trentini Preus, Davi Bernardo Silva, Rodrigo José Piontkewicz  
Instituto Federal de Santa Catarina  
Câmpus Jaraguá do Sul – Rau – Curso de Bacharelado em Engenharia Elétrica  
e-mail: erarbigaus@gmail.com, rodrigo.trentini@ifsc.edu.br, davi.bernardo@ifsc.edu.br, rodrigo.piontkewicz@ifsc.edu.br  
Trabalho de Conclusão de Curso – 04/02/2025

**Resumo** – A integração de fontes de energia renovável, como sistemas fotovoltaicos (PV) e usinas eólicas, às redes elétricas apresenta desafios para a manutenção da estabilidade devido à ausência de inércia natural nesses sistemas. Este artigo aborda a necessidade crítica de controlar a injeção de potência dessas fontes renováveis para mitigar a instabilidade de frequência, exemplificada por um recente apagão no Brasil. Enquanto abordagens convencionais geralmente focam no controle da tensão do barramento CC, este estudo propõe uma nova estratégia de controle de potência utilizando um conversor boost em sistemas fotovoltaicos. A pesquisa combina métodos baseados em IA, especificamente Regressão Linear (Machine Learning) e Otimização por Enxame de Partículas (PSO), para projetar um controlador otimizado de retroalimentação em espaço de estados para conversores boost. Uma análise comparativa avalia o desempenho dessa abordagem em relação aos métodos convencionais de controle digital, considerando eficiência, níveis de corrente e tensão de entrada e saída, comportamento do sinal dos controladores, resposta dinâmica e tempo de estabilização. A implementação prática foi realizada em um simulador em tempo real, demonstrando a eficácia da estratégia proposta no gerenciamento da injeção de potência e na redução de oscilações. Este trabalho destaca o potencial das técnicas de controle baseadas em IA para melhorar o desempenho de conversores de potência, oferecendo insights valiosos sobre suas vantagens e limitações em sistemas de energia renovável.

**Palavras-chave** – Controle de Potência, Estabilidade do Sistema de Potência, Regressão Linear, Otimização por Enxame de Partícula

## **AI-DRIVEN STATE FEEDBACK POWER CONTROL DESIGN OF BOOST CONVERTERS AIMED AT POWER SYSTEM STABILITY IN A REAL-TIME SIMULATOR**

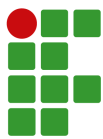
**Abstract** – The integration of renewable energy sources,

such as photovoltaic (PV) systems and wind power plants, into electrical grids presents challenges in maintaining stability due to the absence of natural inertia in these systems. This paper addresses the critical need to control the power injection of these renewable sources to mitigate frequency instability, exemplified by a recent blackout in Brazil. While conventional approaches typically focus on controlling the DC bus voltage, this study proposes a novel power control strategy using a boost converter in photovoltaic systems. The research combines AI-based methods, specifically Linear Regression (Machine Learning) and Particle Swarm Optimization (PSO), to design an optimized state-space feedback controller for boost converters. A comparative analysis evaluates the performance of this approach against conventional digital control methods, considering efficiency, input and output current and voltage levels, controller signal behavior, dynamic response, and stabilization time. Practical implementation was carried out on a real-time simulator, demonstrating the effectiveness of the proposed strategy in managing power injection and reducing oscillations. This work highlights the potential of AI-based control techniques to improve the performance of power converters, offering valuable insights into their advantages and limitations in renewable energy systems.

**Keywords** – Power Control, Power System Stability, Linear Regression, Particle Swarm Optimization.

## I. INTRODUÇÃO

O século XXI testemunhou uma mudança significativa em direção às fontes de energia renovável, impulsionada pela necessidade urgente de reduzir as emissões de gases de efeito estufa e mitigar as mudanças climáticas. Usinas fotovoltaicas (PV) e eólicas emergiram como contribuintes-chave para essa transformação [1], com países como Alemanha, Dinamarca e Irlanda alcançando taxas de penetração de energia renovável superiores a 20% [2, 3]. No entanto, integrar essas fontes intermitentes às redes elétricas apresenta desafios,



particularmente na manutenção da estabilidade de frequência devido à falta de inércia rotacional natural e amortecimento. Ao contrário dos geradores tradicionais, as fontes renováveis dependem de conversores de potência para se conectar à rede, o que dificulta o controle da injeção de potência e a estabilidade da rede [4, 5].

Controlar a potência injetada por esses conversores é crucial para a estabilidade da rede, embora não seja o foco em muitas aplicações práticas [5]. Como os conversores de fontes renováveis possuem inércia nula, a injeção de potência deve ser controlada com base na variação da frequência detectada. Eventos recentes destacam a importância desses desafios. Por exemplo, um apagão no Brasil em 2023 foi atribuído à insuficiência na previsão de potência reativa, o que desestabilizou a rede e levou a falhas em cascata [6].

Esses incidentes reforçam a necessidade de estratégias avançadas de controle que possam gerenciar a injeção de potência de maneira dinâmica e confiável. Métodos convencionais geralmente focam no controle da tensão do barramento CC em sistemas PV, garantindo o máximo de potência entregue à rede. No entanto, essa abordagem pode não ser ideal em cenários com alta penetração de energia renovável, onde o controle preciso da injeção de potência se torna essencial para a estabilidade da rede [7].

O conversor boost, um componente-chave em sistemas PV, eleva a tensão CC a níveis adequados para conexão à rede. Tradicionalmente, controladores PI são utilizados para regular a saída do conversor boost. No entanto, as características de fase não mínima e a sensibilidade a variações do ciclo de trabalho frequentemente tornam esses controladores inadequados em termos de resposta dinâmica e estabilidade. O controle por retroalimentação de estado completo oferece uma alternativa mais robusta, capaz de abordar essas limitações [8].

Um desafio significativo no projeto de controladores de retroalimentação de estado completo, como o Regulador Linear Quadrático (LQR), reside na determinação das matrizes  $Q$  e  $R$  que definem a função de custo. Matrizes mal escolhidas podem levar a um desempenho subótimo, especialmente em aplicações que exigem rastreamento preciso da referência de potência. Em [9] discutem-se as limitações do LQR em alcançar um rastreamento preciso de potência. Nesse contexto, determinar as matrizes  $Q$  e  $R$  adequadas torna-se crucial, pois sua escolha inadequada resulta em respostas dinâmicas insatisfatórias.

Uma abordagem clássica é o uso da fórmula de Ackermann, que permite a colocação dos polos de malha fechada em locais desejados. Essa técnica foi explorada em trabalhos anteriores [10], demonstrando sua viabilidade para o controle de potência. Contudo, o processo de seleção de polos de malha fechada apropriados permanece arbitrário e fortemente dependente da experiência do projetista, o que pode limitar sua aplicabilidade.

Para lidar com essa arbitrariedade, este artigo propõe a aplicação de métodos baseados em inteligência artificial (IA) para o projeto de controladores de retroalimentação de estado completo. Ao alavancar técnicas como Otimização por

Enxame de Partículas (PSO) e Aprendizado de Máquina (ML), buscamos automatizar e otimizar a seleção das matrizes  $Q$  e  $R$ , garantindo melhor desempenho dinâmico e robustez no controle de conversores boost para sistemas PV.

Nesse contexto, avanços recentes em ML têm mostrado potencial para enfrentar desafios complexos de controle. Por exemplo, em [11], estratégias de aprendizado por reforço foram exploradas para superar as ineficiências de métodos convencionais no controle de conversores boost, enquanto em [12] aplicam-se técnicas de regressão linear multivariada (MLR) para maximizar o rastreamento do ponto de potência máxima em sistemas PV, demonstrando resiliência a variações de parâmetros. Esses estudos destacam o potencial de abordagens baseadas em ML para superar métodos clássicos, oferecendo maior robustez e adaptabilidade em sistemas dinâmicos.

No entanto, ainda há uma escassez de pesquisas que implementem tais técnicas baseadas em IA no controle de sistemas de potência e eletrônica de potência, especialmente no controle primário de inversores fotovoltaicos. Dessa maneira, este artigo propõe uma nova estratégia de controle para um conversor boost em sistemas PV, combinando métodos clássicos e baseados em IA. O estudo inclui uma comparação abrangente de duas técnicas baseadas em IA para o projeto do controlador de retroalimentação de estado: PSO e Regressão Linear. As metodologias propostas são avaliadas em um ambiente de simulação em tempo real, com foco no desempenho dinâmico, eficiência computacional e robustez a variações paramétricas.

Este trabalho contribui para o avanço da integração de energia renovável ao fornecer soluções inovadoras para o gerenciamento de potência e estabilidade em sistemas PV. Ao apresentar técnicas impulsionadas por IA, o estudo aborda desafios críticos em sistemas de energia renovável, oferecendo uma estrutura robusta para aprimorar a estabilidade e a confiabilidade da rede diante do aumento da penetração de energia renovável.

Este artigo está organizado em cinco seções. A Seção I apresenta a introdução. Na Seção II, é feita uma revisão da literatura, abordando os principais temas relacionados ao trabalho. A Seção III descreve a metodologia adotada para o desenvolvimento do controle de potência, incluindo detalhes sobre o controle e a implementação dos métodos de inteligência artificial. Na Seção IV, são apresentados os resultados obtidos a partir das simulações em tempo real. Por fim, a Seção V apresenta as conclusões com base nos resultados obtidos, além de perspectivas para pesquisas futuras.

## II. FUNDAMENTAÇÃO TEÓRICA

### A. Conversor boost

O conversor boost é um tipo de conversor não isolado elevador de tensão, ou seja, a tensão de saída é maior do que a tensão de entrada. Sua topologia básica pode ser observada na figura 1. Este conversor possui dois principais modos de condução, o modo de condução contínuo (MCC) e o modo de condução descontínuo (MCD), onde a principal diferença

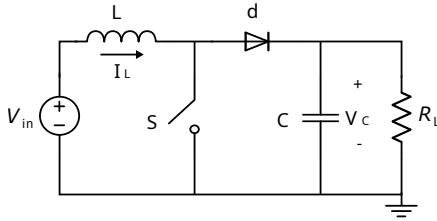


Fig. 1. Topologia típica de um conversor boost.

entre eles é que no MCD a corrente no indutor zera durante o período de comutação apresentando, assim, uma terceira etapa de operação, já que em MCC a corrente no indutor nunca chega a zero. O conversor desenvolvido neste trabalho atua em MCC.

A primeira etapa de operação em MCC ocorre no fechamento da chave. Nesta etapa o diodo está polarizado inversamente impedindo o fluxo de corrente através dele sendo que a corrente no circuito circula entre a fonte de alimentação e o indutor, que passa a acumular energia. Durante esta etapa a tensão na carga é fornecida pelo capacitor.

A segunda etapa de operação ocorre pela abertura da chave, invertendo novamente a polarização do diodo, que passa a conduzir a corrente armazenada no indutor somada à da fonte, alimentando a carga e carregando o capacitor.

O ganho de tensão na saída de um conversor boost se dá pela razão cíclica (*duty cycle*) que aciona a chave, conforme mostra a Eq. (1). É possível observar a relação direta entre a razão cíclica e o ganho estático do conversor pela Eq. (2). Normalmente o ganho máximo utilizado nesse tipo de conversor é de  $D = 0,8$  que representa um ganho estático de 5 vezes a tensão de entrada  $V_{in}$ . Acima disso o conversor passa a ter muitas perdas e se torna muito difícil de controlar [13].

$$d = 1 - \frac{V_{in}}{V_{out}} \quad (1)$$

$$M = \frac{1}{1-d} \quad (2)$$

Como exemplo de aplicação, um esquemático de um sistema fotovoltaico é representado na figura 2, com destaque para o conversor boost e o controle de potência de sua saída.

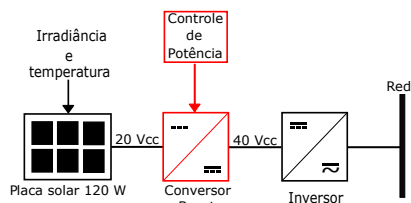


Fig. 2. Topologia típica de um sistema fotovoltaico.

## B. Controle digital

Os níveis de tensão e corrente nas células solares dependem fortemente dos valores de irradiação solar que variam conforme as mudanças ambientais. Quanto maior for a irradiação solar,

maior será a entrada de energia solar na célula fotovoltaica e, portanto, a magnitude da potência aumentará para o mesmo valor de tensão. Para atender as demandas de carga necessárias são utilizados mecanismos de controle [14]. Além disso outros fatores relevantes, como temperatura e acúmulo de resíduos sobre as células fotovoltaicas, podem interferir nos valores de entrada do circuito.

Um sistema de controle consiste em subsistemas e processos (ou plantas) construídos com o objetivo de se obter uma saída desejada com um desempenho desejado, dada uma entrada especificada, mesmo com uma perturbação [15]. Os controladores podem ser analógicos ou digitais, porém nas últimas décadas os controles analógicos vêm sendo substituídos por controles digitais, que consistem em um programa implementado em um computador ou microcontrolador que executa a ação de controle.

Para controlar um sistema físico usando controle digital, o controlador precisa receber medições do sistema, processá-los, e então enviar o sinal de controle para o atuador que executa a ação de controle. Uma vez que o sistema é analógico e o controle é digital é necessário processar o sinal com conversores analógico-digital (ADC) e conversores digital-analógico (DAC). Além disso, faz-se necessário utilizar um sensor para medir a variável controlada e comparar seu comportamento com um sinal de referência enviando um sinal de realimentação (feedback) para o controlador. A ação de controle é baseada em um sinal de erro definido como a diferença entre os valores de referência e os valores reais [16]. O esquema resumido desta configuração está representado pela figura 3.

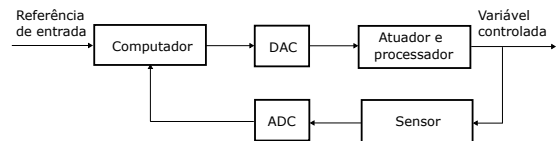
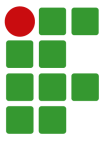


Fig. 3. Topologia típica de um sistema de controle digital.

A popularidade dos controles digitais se dá por uma série de vantagens que eles oferecem em relação aos modelos analógicos. Entre as principais vantagens é possível citar: maior precisão, menor erro de implementação levando em consideração as variações dos componentes analógicos, maior flexibilidade caso necessite ajustes e modificações, velocidade de processamento do sinal de controle superior e baixo custo de implementação até mesmo para aplicações de baixo custo [16].

## C. Controle por variáveis de estados

As representações de sistemas dinâmicos normalmente são do tipo entrada-saída sem haver, nesses casos, um controle dedicado as variáveis internas desses sistemas, representadas por  $x_1, x_2, \dots, x_n$ . Porém existem situações em que se faz necessário obter os valores de uma ou mais variáveis internas do sistema para garantir a eficiência do controle. Para esses casos existe uma abordagem na qual essas variáveis são explicadas por meio de estados [17].



O estado de um sistema dinâmico pode ser definido como um conjunto de variáveis chamadas variáveis de estado. Uma vez tendo conhecimento do estado de uma variável em um determinado instante de tempo aliado ao conhecimento da entrada do sistema é possível determinar os próximos estados dessa variável para qualquer tempo após esse evento, de modo a prever o comportamento futuro do sistema.

Caso o sistema seja não linear é necessário linearizá-lo em torno de uma condição nominal chamado de ponto de equilíbrio. Dessa maneira o sistema se torna, do ponto de vista matemático, invariável no tempo, assumindo que esteja em regime permanente. A linearização é geralmente baseada na expansão do modelo não-linear da série de Taylor. De uma maneira geral a representação matemática mais comum de um sistema linear invariante no tempo se dá pela equação (3).

$$\begin{aligned}\dot{x}(t) &= A(t)x(t) + B(t)u(t) \\ y(t) &= C(t)x(t) + D(t)u(t)\end{aligned}\quad (3)$$

Onde  $A$  é a matriz de estados,  $B$  é a matriz de entrada,  $C$  é a matriz de saída e  $D$  é a matriz de realimentação (*feedforward*).  $x$  é o vetor de estado e  $\dot{x}$  sua primeira derivada,  $u$  é o vetor de controle e  $y$  o vetor de saída.

#### D. Modelagem do conversor

As grandezas mais fundamentais para o projeto de um conversor boost são a tensão de saída ( $V_C$ ) e a corrente no indutor ( $I_L$ ). Uma vez que o conversor em questão possui dois modos de operação (MCC) é possível analisar seu comportamento em cada um dos modos separadamente através das Leis de Kirchhoff. Resolvendo o equacionamento para as variáveis de interesse encontra-se as equações (4) para o modo de condução e (5) para o modo de interrupção [18].

$$\underbrace{\begin{bmatrix} \dot{v}_C \\ \dot{i}_L \end{bmatrix}}_{\dot{x}} = \underbrace{\begin{bmatrix} -\frac{1}{R_L C} & 0 \\ 0 & 0 \end{bmatrix}}_{A_1} \underbrace{\begin{bmatrix} v_C \\ i_L \end{bmatrix}}_x + \underbrace{\begin{bmatrix} 0 \\ \frac{1}{L} \end{bmatrix}}_B V_i, \quad (4)$$

$$\underbrace{\begin{bmatrix} \dot{v}_C \\ \dot{i}_L \end{bmatrix}}_{\dot{x}} = \underbrace{\begin{bmatrix} -\frac{1}{R_L C} & \frac{1}{C} \\ -\frac{1}{L} & 0 \end{bmatrix}}_{A_2} \underbrace{\begin{bmatrix} v_C \\ i_L \end{bmatrix}}_x + \underbrace{\begin{bmatrix} 0 \\ \frac{1}{L} \end{bmatrix}}_B V_i, \quad (5)$$

A partir dessas equações é possível obter o modelo médio de operação do conversor, conforme a equação (6), levando em consideração o regime de trabalho da chave, representado pelo *duty cycle* ( $d$ ).

$$\dot{x} = [A_1 d + A_2(1-d)]x + [Bd + B(1-d)]V_i \quad (6)$$

É possível assumir a tensão de entrada ( $V_i$ ) e o *duty cycle* ( $d$ ) como constantes e representar os eventuais distúrbios do sinal de entrada como  $\Delta d$  e  $\Delta x$ , conforme as expressões:

$$\begin{aligned}d &= D + \Delta d \\ x &= X + \Delta x\end{aligned}\quad (7)$$

Assim, reescrevendo a equação (6), obtém-se a expressão (8):

$$\begin{aligned}\dot{X} + \Delta \dot{x} &= (A_1 D + A_2 - A_2 D)X + \dots \\ &+ (A_1 D + A_2 - A_2 D)\Delta x + \dots \\ &+ (A_1 - A_2)X\Delta d + \dots \\ &+ (A_1 - A_2)\Delta x\Delta d + B V_i.\end{aligned}\quad (8)$$

Desconsiderando os termos de valores muito pequenos e assumindo o ponto de regime permanente ( $\Delta x = \Delta d = \Delta \dot{x} = \dot{X} = 0$ ), obtém-se a equação linearizada (9) que representa o ponto de equilíbrio do conversor.

$$X = -(A_1 D + A_2 - A_2 D)^{-1} B V_i \quad (9)$$

Substituindo a equação (9) na equação (8) encontra-se o modelo médio do conversor boost, que representa o sistema dinâmico dado pela equação (10):

$$\begin{aligned}\begin{bmatrix} \Delta \dot{v}_C \\ \Delta \dot{i}_L \end{bmatrix} &= \underbrace{\begin{bmatrix} -\frac{1}{R_L C} & \frac{(1-D)}{C} \\ -\frac{(1-D)}{L} & 0 \end{bmatrix}}_{A_c} \begin{bmatrix} \Delta v_C \\ \Delta i_L \end{bmatrix} \dots \\ &+ \underbrace{\begin{bmatrix} -\frac{V_i}{(1-D)^2 R_L C} \\ \frac{V_i}{(1-D)L} \end{bmatrix}}_{B_c} \Delta d\end{aligned}\quad (10)$$

Uma vez que a proposta é controlar a potência de saída do conversor ( $p_0 = v_C i_{R_L}$ ), se torna necessário fazer uma breve análise do comportamento das correntes  $i_{R_L}$  e  $i_L$ , onde a razão da primeira pela segunda é igual ao *duty cycle* ( $1-D$ ).

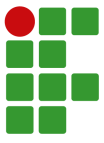
$$\frac{i_{R_L}}{i_L} = 1-d \therefore i_{R_L} = i_L - i_L d \quad (11)$$

Logo, a equação da potência de saída do conversor se dá por:

$$p_0 = v_C i_L - v_C i_L d \quad (12)$$

Linearizando a equação (12) pela série de Taylor, tem-se:

$$\Delta p_0 = \bar{i}_L(1-D)\Delta v_C + \bar{v}_C(1-D)\Delta i_L - \bar{v}_C \bar{i}_L \Delta d \quad (13)$$



Dessa maneira, a equação de saída do conversor é expressa por:

$$\Delta p_0 = \underbrace{[\bar{i}_L(1-D)\bar{v}_c(1-D)]}_{C_c} \left[ \begin{array}{c} \Delta v_c \\ \Delta i_L \end{array} \right] \underbrace{-\bar{v}_c\bar{i}_L}_{D_d} \Delta d \quad (14)$$

As equações (10) e (14) representam o modelo médio do conversor boost com a potência de saída como referência. Essas equações são fundamentais para se obter os valores usados no controlador.

### E. Regulador Linear Quadrático

O Regulador Linear Quadrático (LQR) é um método de controle usado em sistemas lineares que tem por objetivo garantir que o sistema opere próximo a referência com o mínimo de esforço aumentando, assim, a eficiência do controle.

Em um sistema linear com modelagem característica de estados, como na equação (3), a função de custo quadrática ( $J$ ) é minimizada conforme a equação (15).

$$J = \int_0^{\infty} (x^T Q x + u^T R u) dt \quad (15)$$

Onde  $Q$  é uma matriz simétrica positiva que penaliza o desvio da referência, e  $R$  é uma matriz simétrica que penaliza o esforço de controle. É uma prática comum projetar reguladores variando os valores da matriz  $Q$ , mantendo  $R$  fixo [19].

De acordo com [20] a minimização da função custo, mostrada pela equação (15), fornece ao estado leis de controle de *feedback*, conforme a equação (16):

$$u^* = -R^{-1} B^T P x = -K x \quad (16)$$

Onde  $P$  é a solução simétrica definida positiva da Equação de Riccati Algébrica Contínua (ARE), mostrada pela equação (17).

$$A^T P + P A - P B R^{-1} B^T P + Q = 0 \quad (17)$$

### F. Métodos baseados em IA para determinação de $Q$ e $R$

Diversas técnicas de otimização têm sido empregadas para enfrentar os desafios em sistemas de potência, eletrônica de potência e aplicações relacionadas. Em [21] citam a programação linear, abordagens baseadas em gradiente e métodos meta-heurísticos.

Embora os métodos clássicos e de programação linear sejam eficazes para resolver muitos problemas de otimização, eles apresentam limitações devido à necessidade de funções objetivo contínuas e diferenciáveis [22]. Essas restrições matemáticas, juntamente com as altas demandas computacionais desses algoritmos, podem torná-los impraticáveis para abordar desafios complexos de otimização, especialmente em áreas como eletrônica de potência, onde os espaços de solução podem ser irregulares e multidimensionais. Como resultado, esses métodos frequentemente são inadequados para lidar com as complexidades das tarefas de otimização modernas.

Reconhecidos por sua simplicidade de implementação e

confiabilidade, as abordagens de Aprendizado de Máquina, particularmente a Regressão Linear, e a Otimização por Enxame de Partículas (PSO) são altamente eficazes na resolução de problemas de otimização [23, 24]. Essas técnicas são adaptáveis e se destacam na gestão de espaços de busca irregulares e de alta dimensionalidade. Diferentemente dos métodos clássicos, elas evitam suposições sobre continuidade, diferenciabilidade ou convexidade da função objetivo [25].

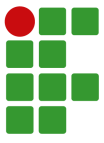
Os desafios de otimização relacionados a sistemas de potência são particularmente bem resolvidos pelo PSO, devido ao seu equilíbrio entre simplicidade e desempenho. Sua eficiência computacional, em comparação com métodos tradicionais e outros heurísticos, decorre de seu design intuitivo, que simplifica tanto a implementação quanto a parametrização [26]. Além disso, o PSO alcança rápida convergência e alta precisão sem depender de propriedades matemáticas específicas da função objetivo [22, 27].

O aprendizado de máquina tem se tornado cada vez mais valioso na otimização de sistemas de potência e eletrônica de potência. A Regressão Linear modela a relação entre variáveis de entrada e saídas dos sistemas de potência com simplicidade e eficiência computacional, tornando-a ideal para cenários que envolvem grandes conjuntos de dados e dinâmicas de sistemas complexos [28]. Ela evita a necessidade de suposições rigorosas, como diferenciabilidade, fornecendo uma abordagem robusta e adaptável que melhora a precisão preditiva, especialmente ao lidar com sistemas em larga escala e comportamentos não lineares em aplicações de eletrônica de potência [29]. Por exemplo, em [30] foi empregado regressão linear orientada por dados para linearizar modelos de fluxo de potência, demonstrando maior precisão de cálculo em comparação com métodos tradicionais.

Além disso, em [31] foi utilizado uma abordagem de aprendizado profundo em conjunto, incorporando regressão linear, para estimativa de estado de sistemas de potência em tempo real, alcançando desempenho superior em relação a métodos orientados por dados existentes. Essas aplicações destacam a eficácia da Regressão Linear em resolver tarefas complexas de otimização em sistemas de potência e eletrônica de potências. As próximas seções deste artigo abordam especificamente a implementação do PSO e do ML com Regressão Linear visando o projeto das matrizes  $Q$  e  $R$  para o controle de potência de um conversor boost baseado em sistemas fotovoltaicos.

### G. Otimização por enxame de partículas

*Particle swarm optimization* (PSO), ou otimização por enxame de partículas, é um método computacional de otimização por iterações com o objetivo de se aproximar ao máximo do ponto ótimo ideal de um sistema, podendo ser máximo ou mínimo. Essa técnica foi desenvolvida a partir de pesquisas sobre o comportamento em grupo de alguns tipos de animais como cardumes de peixe e bandos de pássaros. O PSO destaca-se principalmente pela eficiência computacional e



fácil implementação [32]. Trabalhos como [33] demonstram a capacidade de otimização dos parâmetros de controle como máximo sobrepasso e tempo de assentamento utilizando PSO para calcular os ganhos de um controle PID.

Cada partícula se move no espaço de busca com uma certa velocidade, que se ajusta a cada iteração. A velocidade de cada partícula é influenciada pela sua própria melhor posição encontrada até o momento e pela melhor posição coletiva encontrada por qualquer outra partícula no enxame. Para atualizar a velocidade da partícula utiliza-se a equação (18).

$$v_i(k+1) = w(k)v_i(k) + c_1r_1[p_i - x_i(k)] + c_2r_2[p_g - x_i(k)] \quad (18)$$

Onde  $v_i(k)$  e  $x_i(k)$  são, respectivamente, a velocidade e a posição da partícula  $i$  na iteração  $k$ .  $p_i$  e  $p_g$  representam a melhor solução candidata individual e global do enxame na iteração.  $c_1$  e  $c_2$  são os fatores de aprendizado e  $r_1$  e  $r_2$  são números aleatórios uniformes.  $w_k$  é o coeficiente de inércia dado pela equação (19):

$$w(k) = w_{max} - k \frac{w_{max} - w_{min}}{k_{max}} \quad (19)$$

Sendo  $w_{max}$  e  $w_{min}$  os termos superior e inferior do coeficiente de inércia e  $k_{max}$  é o número máximo de iterações.

Como medida para evitar o afastamento excessivo das partículas do domínio de exploração, a velocidade de cada partícula é limitada pela equação (20).

$$v_{max} = v_{factor} \left( \frac{x_{max} - x_{min}}{2} \right) \quad (20)$$

Onde  $v_{factor}$  é o fator de limitação da velocidade.  $x_{max}$  e  $x_{min}$  representam o intervalo do espaço de busca. Dessa maneira a velocidade da partícula ( $v_i$ ) é limitada entre  $-v_{max} \leq v_i \leq v_{max}$ .

Uma vez atualizadas as velocidades, pode-se determinar a posição de cada partícula a partir da equação (21):

$$x_i(k+1) = x_i(k) + v_i(k+1) \quad (21)$$

Para a interrupção do algoritmo, pode-se considerar a variação residual de uma matriz de otimização ou definir um número máximo de iterações. Neste trabalho, optou-se pelo segundo método devido à sua simplicidade de implementação. O algoritmo PSO é implementado seguindo os passos sequenciais abaixo [34, 35]:

1. Inicializar o enxame atribuindo valores aleatórios ou estrategicamente determinados às posições e velocidades das partículas. Neste contexto, as partículas representam os parâmetros nas matrizes  $Q$  e  $R$ .
2. Avaliar a aptidão (*fitness*) de cada partícula utilizando a função objetivo agregada, incorporando uma função de penalidade para lidar com possíveis violações de restrições.
3. Atualizar a melhor solução pessoal ( $p_{best}$ ) para cada

partícula. Se o valor de aptidão atual de uma partícula superar seu valor anterior de  $p_{best}$ , atribuir o novo valor a  $p_{best}$  e definir a posição atual da partícula,  $x_i$ , como  $p_i$ .

4. Determinar a melhor solução global identificando a partícula com o maior valor de aptidão. A posição dessa partícula torna-se  $p_g$ , e seu valor de aptidão é designado como  $g_{best}$ .
5. Atualizar a velocidade e a posição de cada partícula usando as Eq.(18) e Eq.(21), respectivamente.
6. Iterar os passos 2 até 5 até que o critério de parada seja atendido. Neste estudo, o processo termina ao atingir o número máximo de iterações.

#### H. Regressão linear

A regressão linear é um método estatístico fundamental utilizado para modelar a relação entre uma variável dependente e uma variável independente, ajustando uma equação linear aos dados observados [36]. Como o modelo emprega uma única variável preditora, ele é classificado como um modelo de regressão linear simples [37]. Na regressão linear simples, essa relação é expressa como:

$$y = \beta_1 x + \beta_0, \quad (22)$$

onde  $y$  representa a variável dependente, que está sendo prevista ou explicada. A variável independente  $x$  é o preditor que influencia  $y$ . O coeficiente  $\beta_1$  é o coeficiente angular, indicando como  $y$  varia para cada unidade de aumento em  $x$ . O coeficiente angular quantifica a intensidade e a direção da relação entre as duas variáveis [36]. Um coeficiente angular positivo indica que  $y$  aumenta conforme  $x$  aumenta, enquanto um coeficiente angular negativo indica que  $y$  diminui conforme  $x$  aumenta. O coeficiente angular é expresso como:

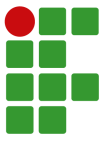
$$\beta_1 = \frac{\sum (x_i - \bar{x})(y_i - \bar{y})}{\sum (x_i - \bar{x})^2}. \quad (23)$$

O coeficiente  $\beta_0$  representa o intercepto, que descreve o valor de  $y$  quando  $x$  é igual a zero. Ele indica onde a linha de regressão cruza o eixo  $y$  e é expresso como:

$$\beta_0 = \bar{y} - \beta_1 \bar{x}. \quad (24)$$

O algoritmo de regressão linear simples é implementado seguindo os passos sequenciais [38]:

1. Coletar o conjunto de dados que contém pares de valores das variáveis independente ( $x$ ) e dependente ( $y$ ).
2. Assumir uma relação linear entre  $x$  e  $y$  usando a Eq. (22). O objetivo é encontrar os valores de intercepto e coeficiente angular que melhor se ajustem aos dados.
3. Determinar a taxa de variação de  $y$  para uma unidade de variação em  $x$ . Calcular o coeficiente angular usando a Eq. (23).
4. Definir o valor de  $y$  quando  $x$  é igual a zero. Calcular o intercepto usando a Eq. (24).



- Combinar  $\beta_0$  e  $\beta_1$  para criar a equação linear que melhor descreve a relação entre  $x$  e  $y$ .
- Avaliar quão bem o modelo se ajusta aos dados utilizando métricas como o coeficiente de determinação e a análise de resíduos.
- Usar a equação ajustada para prever o valor de  $y$  para novos valores de  $x$ .

### III. METODOLOGIA

Nesta seção estão elencadas as especificações do projeto bem como os componentes integrantes do conversor boost e as aplicações das técnicas de controle utilizadas.

#### A. Projeto do conversor boost

Para aplicar um controle de potência em um conversor boost é necessário conhecer alguns valores fundamentais como tensão de entrada, tensão de saída, corrente de entrada, corrente de saída, frequência de chaveamento e potência de saída. Para isso é preciso assumir alguns valores como parâmetros iniciais que são fundamentais para o projeto do conversor. Estes valores iniciais estão dispostos na tabela I e, após a demonstração dos cálculos, os valores dos componentes são mostrados na II. Para  $\Delta V_C$  foi assumido 1% de  $V_{out}$ .

**TABELA I**  
Parâmetros iniciais do projeto para o conversor boost

$V_{in}$	20 V
$V_{out}$	40 V
$P_{out}$	120 W
$F_s$	75 kHz
$\Delta V_C$	0,4 V
$\Delta I_L$	20%* $I_L$

Uma vez que este trabalho propõe a aplicação do conversor em uma planta fotovoltaica é fundamental entender que, para esta aplicação, a tensão de entrada pode variar de acordo com mudanças climáticas ou outras razões que diminuam a incidência luminosa sob a placa. Nestes casos a compensação pela diminuição da tensão de entrada se dá pela variação do *Duty cycle*.

Utilizando a equação (1), obtém-se o valor do *duty cycle* de acordo com os parâmetros iniciais de projeto, assim como o ganho estático do conversor, determinado pela equação (26).

$$d = 1 - \frac{20}{40} = 0,5 \quad (25)$$

$$M = \frac{1}{1 - 0,5} = 2 \quad (26)$$

Uma vez determinados os valores de potência e tensão de saída, é possível calcular a corrente que circula na carga do circuito conforme a equação (27) e, a partir dela, definir o valor da carga pela equação (28).

$$I_0 = \frac{120}{40} = 3A \quad (27)$$

$$R_l = \frac{40}{3} = 13,333 \Omega \quad (28)$$

Para calcular o capacitor em paralelo com a carga, utilizou-se a equação (29).

$$C_0 = \frac{I_o \cdot D}{\Delta V_C \cdot F_s} = 50\mu F \quad (29)$$

A corrente no indutor também é um parâmetro importante, pois trata-se da corrente de entrada do conversor. A corrente no indutor é deduzida pela equação (30):

$$I_L = \frac{P_{out}}{V_{in}} = 6A \quad (30)$$

É fundamental, também, para o projeto conhecer o valor da corrente de pico no indutor. Este valor é importante para definir a chave e o diodo a serem utilizados no circuito, conforme a equação (31).

$$I_{Lpico} = I_L + \frac{\Delta I_L}{2} = 6,6A \quad (31)$$

O indutor é calculado pela equação (32).

$$L = \frac{V_{in} \cdot D}{\Delta I_L \cdot F_s} = 111,111\mu H \quad (32)$$

Para facilitar a visualização, a Tabela II apresenta os componentes básicos calculados para o conversor boost.

**TABELA II**  
Valores calculados dos componentes do conversor boost

<i>Dutycycle</i>	0,5
$R_l$	13,333 $\Omega$
$C_1$	50 $\mu F$
$L$	111,111 $\mu H$

#### B. Regulador Quadrático Linear com Integrador

Considerando que a saída do sistema é definida como a potência  $P_{out}$ , com a tensão de saída  $V_{out}$  e a corrente do indutor  $I_L$  sendo mensuráveis, a avaliação da controlabilidade do sistema torna-se essencial para a implementação do controle por realimentação de estados. A medição dos estados do sistema simplifica a implementação do controle por realimentação completa de estados, reduzindo a complexidade geral.

Para garantir o rastreamento eficaz de referência e a rejeição de perturbações para sinais do tipo degrau, um integrador é incorporado no laço direto devido à ausência de um integrador natural na planta. A Figura 4 ilustra a arquitetura do sistema de controle, onde  $r, x, u, w$  e  $y$  representam, respectivamente, o sinal de referência, os estados medidos, o sinal de controle, a

perturbação e a saída.

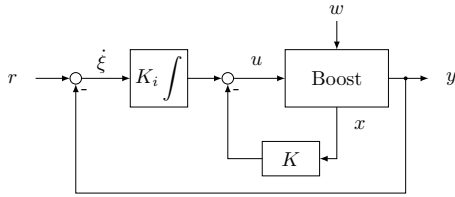


Fig. 4. Diagrama em blocos do controle por realimentação de estados.

O objetivo do projeto de controle é determinar adequadamente  $K$  e  $K_i$  de acordo com alguns requisitos de projeto. Este trabalho utiliza o Regulador Quadrático Linear com Integrador (LQI) como controlador, o qual realiza a otimização da função custo,

$$J = \int_0^{\infty} ((y-r)^T Q (y-r) + u^T R u) dt, \quad (33)$$

onde  $J$  é o custo a ser minimizado,  $y$  é a saída,  $r$  é a referência,  $u$  é o sinal de controle, e  $Q$  e  $R$  são as matrizes de ponderação. A inclusão do integrador introduz um novo estado,  $\xi$ , na dinâmica do sistema. A minimização de  $J$  com relação a  $u$  resulta na formulação do sinal de controle dada por:

$$u = -Kx + K_i \xi, \quad (34)$$

onde  $K$  denota a matriz de ganho de realimentação de estados e  $K_i$  representa o ganho do integrador.

Teoricamente, a seleção das matrizes de ponderação  $Q$  e  $R$  parece ser simples, já que  $Q$  é uma matriz diagonal  $3 \times 3$  e  $R$  é um escalar, resultando em apenas quatro parâmetros a serem determinados.

Na prática, entretanto, esse processo é desafiador devido às infinitas combinações de valores possíveis. Por exemplo, o uso de uma diagonal uniforme para  $Q$  frequentemente leva a resultados subótimos, independentemente do valor de  $R$ .

Isso destaca a significativa dependência da experiência do projetista, motivando a exploração de métodos baseados em IA, como PSO e aprendizado de máquina, conforme discutido nas seções subsequentes.

### C. Implementação do PSO

O principal objetivo deste trabalho é empregar o Particle Swarm Optimization (PSO) para o treinamento de modelos de Machine Learning (ML), aproveitando as capacidades de otimização do PSO para lidar com os desafios de ajuste de parâmetros em sistemas dinâmicos. Especificamente, o PSO é utilizado para otimizar os parâmetros de controle por realimentação no espaço de estados do conversor boost, que serve como modelo dinâmico. O modelo do conversor boost é definido por sua representação no espaço de estados, conforme apresentado nas Eqs. (10) e (14).

O algoritmo PSO é configurado com 30 iterações e um enxame de 65.536 partículas, escolhido para equilibrar a

exploração do espaço de busca e a viabilidade computacional. O coeficiente de inércia é fixado em  $w = 0.9$ , enquanto os coeficientes de aprendizado são definidos como  $c_1 = c_2 = 2$ , garantindo um processo de busca bem distribuído. Uma vez que  $w$  mantém-se fixo não há atualizações na equação 17. As posições iniciais das partículas são inicializadas usando uma função aleatória com distribuição normal para promover diversidade no enxame.

Para guiar a otimização, uma função de custo personalizada é projetada, considerando métricas de desempenho críticas do conversor boost. A função de custo do PSO é definida como:

$$J_{\text{PSO}} = \gamma_1 t_r + \gamma_2 t_s + \gamma_3 M_p + \gamma_4 m_p + \mu, \quad (35)$$

onde  $\gamma_i$  ( $i = 1, 2, 3, 4$ ) são fatores de ponderação que determinam a importância relativa de cada métrica de desempenho:  $t_r$  (tempo de subida),  $t_s$  (tempo de assentamento),  $M_p$  (sobressinal) e  $m_p$  (subsobressinal). O fator de penalidade  $\mu$  é introduzido para lidar com restrições no sinal de controle. Neste estudo, os fatores de ponderação são definidos como  $\gamma_1 = 0.2$ ,  $\gamma_2 = 0.4$ ,  $\gamma_3 = 0.3$  e  $\gamma_4 = 0.1$ .

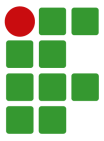
Garantir a positividade definida das matrizes  $Q$  e  $R$  é essencial para a estabilidade e robustez do sistema em malha fechada. Para isso, restrições são incorporadas no algoritmo PSO, assegurando que todos os coeficientes permaneçam positivos durante o processo de otimização.

Para cada par candidato de matrizes  $Q$  e  $R$  gerado pelo PSO, o sistema em malha fechada é simulado. A potência de saída  $p_o$  é avaliada em termos de  $t_r$ ,  $t_s$ ,  $M_p$  e  $m_p$ . Adicionalmente, o sinal de controle  $u$  é monitorado para garantir que permaneça dentro dos limites aceitáveis. Caso  $u$  exceda seu valor máximo permitido de um, o fator de penalidade  $\mu$  recebe um valor alto, aumentando significativamente  $J_{\text{PSO}}$  para desencorajar soluções inválidas. Caso contrário,  $\mu$  é definido como zero. O processo de otimização é resumido no pseudocódigo apresentado no Algoritmo 1.

### D. Implementação da regressão linear

O algoritmo de Machine Learning é utilizado para prever as matrizes ótimas  $Q$  e  $R$  do controlador de retroalimentação em espaço de estados, com base no conjunto de dados gerado pela otimização via PSO. O processo de ML envolve as seguintes etapas: carregamento dos preditores e variáveis-alvo, estruturação do modelo de ML, divisão do conjunto de dados em subconjuntos de treinamento e validação, e aplicação da técnica de Regressão Linear para ajustar um modelo preditivo aos dados.

Primeiramente, o conjunto de dados obtido das execuções do PSO é pré-processado para separar os preditores (características de entrada) das variáveis-alvo (saídas). Os preditores são derivados das métricas de desempenho do sistema com conversor boost, enquanto as variáveis-alvo correspondem aos coeficientes das matrizes  $Q$  e  $R$ . Isso garante que o algoritmo de ML aprenda a relação entre os requisitos de desempenho e os parâmetros de projeto do controlador.



**Algoritmo 1** OTIMIZAÇÃO POR ENXAME DE PARTÍCULAS

**Entrada:** número de iterações, número de partículas, dimensões das partículas,  $w$ ,  $c1$  e  $c2$

**Saída:** melhor posição global das matrizes Q e R

**begin**

Inicializar as partículas do enxame com posições aleatórias;  
Inicializar as velocidades das partículas e as melhores posições;  
Definir fatores de ponderação para a otimização;

**for** cada iteração **do**

**for** cada partícula do enxame **do**

Atualizar a posição da partícula;  
Calcular os ganhos do controlador;  
Simular o sistema em malha fechada;  
**if** o sinal de controle excede o limite máximo **then**  
    Aplicar penalidade alta;

**else**

    Aplicar penalidade zero;  
    Calcular as informações de resposta ao degrau;  
    Calcular a função de custo  $J$  com os fatores de ponderação;

**if** a posição atual é melhor **then**

    Atualizar a melhor posição da partícula;

Salvar a melhor posição global;

**for** cada partícula do enxame **do**

    Atualizar a velocidade da partícula;

Reter a melhor posição global;

O conjunto de dados é então dividido em subconjuntos de treinamento e validação. Tipicamente, 80% dos dados são alocados para treinamento e os 20% restantes para validação. Essa abordagem assegura que o modelo de ML seja treinado em uma grande parte dos dados, enquanto retém dados suficientes para avaliar seu desempenho em exemplos não vistos.

A validação cruzada é um componente crítico do processo de treinamento, aumentando a robustez e a capacidade de generalização do modelo de ML. Neste estudo, uma abordagem de validação cruzada em 5 partes é implementada, onde o conjunto de dados é dividido em cinco partes de tamanhos iguais. Em cada iteração, o modelo é treinado em quatro partes (conjunto de treinamento) e validado na parte restante (conjunto de teste). Esse processo é repetido cinco vezes, garantindo que cada parte sirva como conjunto de validação exatamente uma vez. Para cada iteração, métricas-chave de desempenho, como o Erro Médio Quadrático (RMSE) e o coeficiente de determinação ( $R^2$ ), são calculadas conforme as equações 36 e 37, e os valores médios fornecem uma medida abrangente da precisão preditiva do modelo.

$$RMSE = \sqrt{\frac{1}{n} \sum_{i=1}^n (y_i - p_i)^2} \quad (36)$$

Onde  $n$  é o número de amostras,  $y_i$  é o valor observado para a amostra e  $p_i$  é o valor previsto pelo modelo para a amostra  $i$ .

$$R^2 = 1 - \frac{SQ_{res}}{SQ_{tot}} \quad (37)$$

Onde  $SQ_{tot}$  é a soma total dos quadrados das diferenças entre a média e cada valor observado e  $SQ_{res}$  é a soma dos quadrados dos resíduos, que correspondem às diferenças entre os valores observados e os valores previstos pelo modelo.

O modelo de ML utiliza a técnica de Regressão Linear para estabelecer uma relação linear entre os preditores e as variáveis-alvo. Para cada variável-alvo (coeficientes de  $Q$  e  $R$ ), um modelo linear independente é ajustado, permitindo prever cada coeficiente com base nas características de entrada.

A estrutura do processo de ML é resumida no pseudocódigo apresentado no Algoritmo 2.

**Algoritmo 2** MACHINE LEARNING – MODELO DE REGRESSÃO LINEAR

**Entrada:** matrizes Q e R e  $k$

**Saída:** média do modelo de regressão linear ajustado

**begin**

Dividir os dados em preditores ( $J_{PSO}$ ) e alvos (coeficientes  $Q$  e  $R$ );

Criar partições  $k$ -folds para validação cruzada;

**for** cada partição **do**

    Extrair dados de treinamento com base nas partições;  
    Extrair dados de teste com base nas partições;  
    Calcular a média dos preditores e dos alvos;  
    Calcular o coeficiente angular;  
    Calcular o coeficiente linear;  
    Ajustar o modelo de regressão linear aos dados de treinamento;  
    Realizar previsões usando os dados de teste;  
    Armazenar previsões na ordem original;  
    Calcular o erro médio quadrático (RMSE);  
    Calcular o coeficiente de determinação ( $R^2$ );

Retornar a média do modelo de regressão linear;

Essa abordagem baseada em ML assegura um mecanismo eficiente e automatizado para aproximar as matrizes  $Q$  e  $R$  necessárias para o controle em espaço de estados. Ao utilizar o conjunto de dados gerado pela otimização via PSO, o modelo de ML pode prever rapidamente os parâmetros de controle ótimos para o conversor boost em condições variadas do sistema.

*E. Simulação do circuito no software Simulink*

Para verificar o funcionamento do circuito considerando as perdas foi utilizado o *software* Simulink integrado ao *software* Matlab simulando os componentes levando em consideração suas não idealidades e, assim, as perdas de energia e quedas de tensão no circuito, permitindo uma maior aproximação dos resultados simulados aos reais. A figura 5 mostra o circuito como diagrama de blocos no Simulink.

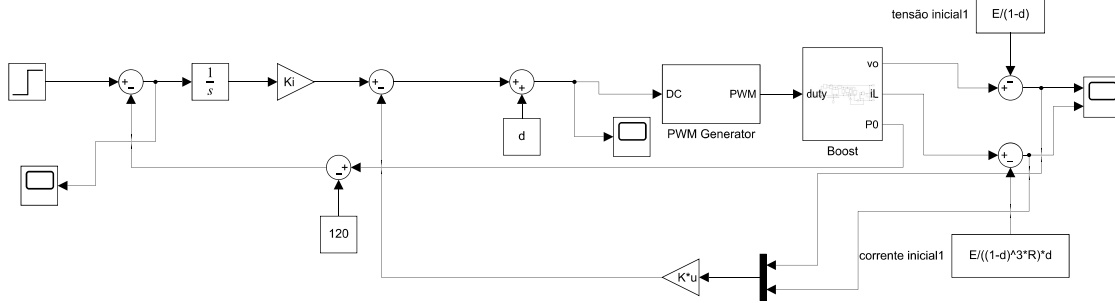
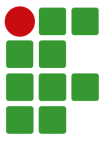


Fig. 5. Circuito de controle no *software* Simulink.

O bloco contendo o valor constante "120" determina a potência de interesse para a operação do sistema. O bloco  $d$  representa o *dutycycle* nominal (0,5), assim como os blocos *tensão inicial* e *corrente inicial* o fazem. Um degrau, condicionado para atuar em 100ms foi adicionado na entrada do sistema para verificar a rejeição a perturbação após a estabilização. No esquemático, é possível perceber que a potência de saída do conversor é realimentada antes do bloco integrador  $K_i$  e que a tensão de saída e a corrente no indutor estão sendo realimentados no sistema após o integrador.

A Figura 6 detalha o interior do bloco "Boost", apresentado no diagrama do circuito de controle na Figura 5. Esta representação permite visualizar os detalhes da estrutura interna e dos componentes do conversor dentro do simulador.

#### IV. RESULTADOS DE SIMULAÇÃO EM TEMPO REAL

Esta seção apresenta os resultados obtidos das simulações em tempo real. Os experimentos foram conduzidos utilizando a plataforma Typhoon HIL 404, com dinâmica eletromagnética operando em uma resolução de 200 ns, enquanto as medições e algoritmos de controle operam com um período de amostragem de 100  $\mu$ s. As simulações demonstram o desempenho do controle digital aplicando variações expressivas nos parâmetros fundamentais do conversor.

##### A. Desempenho dos algoritmos

O algoritmo PSO gerou um conjunto de dados com 4051 amostras contendo  $Q$ ,  $R$  e os respectivos valores de  $J_{PSO}$ . Este conjunto de dados foi utilizado para o treinamento subsequente de modelos de ML. Em média, o tempo de execução de cada execução foi de aproximadamente 9732 s (Intel Core i7-7500 CPU @ 2.7 GHz, 16 GB RAM), demonstrando a intensidade computacional do procedimento. Notavelmente, a função de custo  $J_{PSO}$  apresentou rápida convergência, normalmente estabilizando-se em 12 iterações, conforme ilustra a figura 7, destacando a eficiência do algoritmo PSO na exploração do espaço de soluções.

Notavelmente, o tempo de execução do algoritmo de ML é uma fração do tempo computacional do PSO, com média de aproximadamente 0,76 s por execução na mesma plataforma

de hardware. O algoritmo gerou 4.051 matrizes  $Q$  e  $R$  únicas, que foram analisadas estatisticamente para derivar valores médios de  $Q = \text{diag}(0,001; 0,056; 339,2)$  e  $R = 0,001$ . Os correspondentes desvios padrão são  $\sigma_{q11} = 0$ ,  $\sigma_{q22} = 0,003$ ,  $\sigma_{q33} = 15,17$  e  $\sigma_r = 0$ , resultando em variações relativas de 0%, 5,36%, 4,47% e 0%, respectivamente. Esses baixos desvios padrão, particularmente para  $q_{11}$  e  $r$ , indicam um alto nível de consistência nos resultados da otimização, reforçando a confiabilidade das previsões baseadas em ML.

A importância de um modelo de ML treinado reside em sua capacidade de acelerar drasticamente o processo de projeto do controlador. Enquanto o PSO oferece um mecanismo eficaz para otimizar  $Q$  e  $R$ , seu alto custo computacional — com média de 9732 s por execução — torna-o impraticável para aplicações em tempo real. Em contraste, o modelo de ML pode prever as matrizes  $Q$  e  $R$  quase instantaneamente, permitindo rápida adaptação às mudanças na dinâmica do sistema sem a necessidade de reexecutar o processo computacionalmente intensivo do PSO. Essa capacidade é particularmente crucial em ambientes dinâmicos e de rápida mudança, como sistemas de energia renovável, onde a responsividade em tempo real é essencial para manter a estabilidade e o desempenho.

##### B. Validação do Modelo

Primeiramente, validar o modelo do conversor boost descrito pelas Eqs. (10) e (14) é essencial para garantir sua precisão e confiabilidade. Em [10], os autores demonstraram a validação do modelo para uma variação de 0,5% no ciclo de trabalho  $d$ . Para explorar a robustez do modelo, este estudo estende a análise testando um aumento de 1% em  $d$ , seguido por uma subsequente redução de 2%. Essa abordagem visa avaliar a sensibilidade do sistema a variações maiores no ciclo de trabalho, fornecendo informações mais profundas sobre seu comportamento dinâmico. Os resultados dessa análise de sensibilidade são apresentados na Fig. 8.

É evidente que, diferentemente dos resultados reportados em [10], surge uma pequena discrepância no modelo quando o ciclo de trabalho  $d$  é aumentado em 1%. Essa discrepância torna-se mais pronunciada para uma redução de 2% em  $d$ , destacando a maior sensibilidade do conversor a variações na entrada.

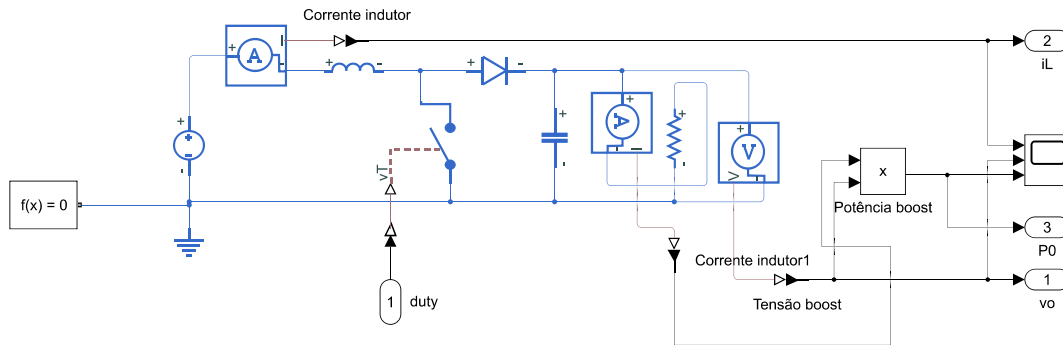


Fig. 6. Conversor boost no software Simulink.

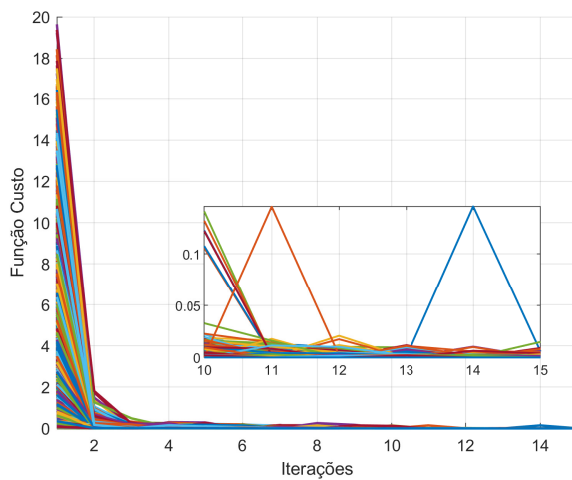


Fig. 7. Função custo pelo n° de iterações

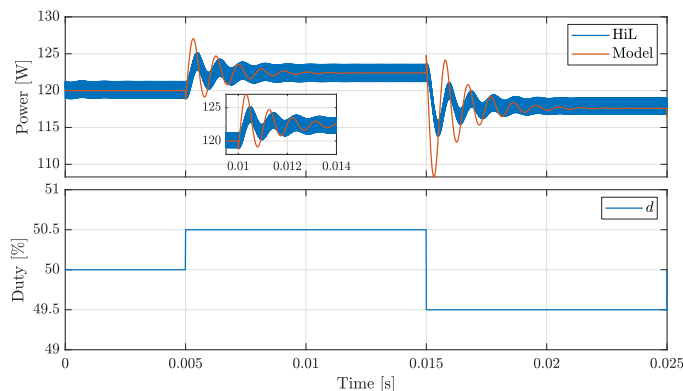


Fig. 8. Validação do modelo médio.

Esses achados enfatizam as não linearidades e as possíveis limitações do modelo quando submetido a perturbações maiores, ressaltando a necessidade de estratégias de controle avançadas para garantir desempenho dinâmico preciso e estabilidade do sistema sob condições operacionais variáveis.

### C. Validação do Controle

Como o modelo do conversor boost é formulado em tempo contínuo, enquanto o algoritmo de controle opera em tempo discreto, é necessário discretizar a dinâmica do conversor boost para projetar um controlador LQI em tempo discreto, o que é realizado utilizando a capacidade do Typhoon HIL 404 de computar cada passo em  $100 \mu\text{s}$ . As matrizes de ponderação  $Q$  e  $R$ , derivadas do algoritmo de ML, resultam em ganhos de controle  $K = [-0.01 \quad 0.028]$  e  $K_i = 1.98$ .

Três cenários de simulação são considerados para avaliar o desempenho do controlador:

- Caso 1: Variação em degrau de  $\pm 10\%$  na potência de referência  $P_{\text{ref}}$ .
- Caso 2: Variação em degrau de  $\pm 25\%$  na tensão de entrada  $V_i$ .
- Caso 3: Variação em degrau de  $\pm 50\%$  na resistência de carga  $R_L$ .

A Fig. 9 ilustra os resultados para o Caso 1, mostrando a potência de saída  $P_{\text{out}}$ , a corrente do indutor  $I_L$ , a corrente da carga  $i_{R_L}$ , a tensão no capacitor  $V_C$  e o ciclo de trabalho  $d$ . A referência de potência é rastreada com sucesso em aproximadamente 6 ms para uma mudança de referência em degrau de 50 Hz. Nesse cenário, a corrente do indutor  $I_L$  oscila entre 5 A e 7 A, enquanto a corrente da carga  $i_{R_L}$  varia entre 2,9 A e 3,1 A. Similarmente, a tensão no capacitor  $V_C$  varia de 38 V a 42 V. Essas variações correspondem aos ajustes no ciclo de trabalho  $d$ , que é o sinal de controle gerado pelo LQI, variando de aproximadamente 46% a 58%.

A Fig. 10 apresenta os resultados para o Caso 2, onde a tensão de entrada  $V_i$  é variada em degraus de 15 V a 25 V a cada 200 ms. Pode-se observar que o sistema de controle mantém efetivamente a potência de referência  $P_{\text{out}}$  em 120 W durante o estado estacionário. Durante os períodos transitórios, quando  $V_i$  cai para 15 V, a potência de saída  $P_{\text{out}}$  atinge um pico de 356 W, estabilizando de volta para 120 W em aproximadamente 5 ms.

A Fig. 11 apresenta os resultados para o Caso 3, onde a resistência de carga é variada em  $\pm 50\%$  a cada 50 ms. Apesar dessas variações significativas, o sistema de controle rastreia

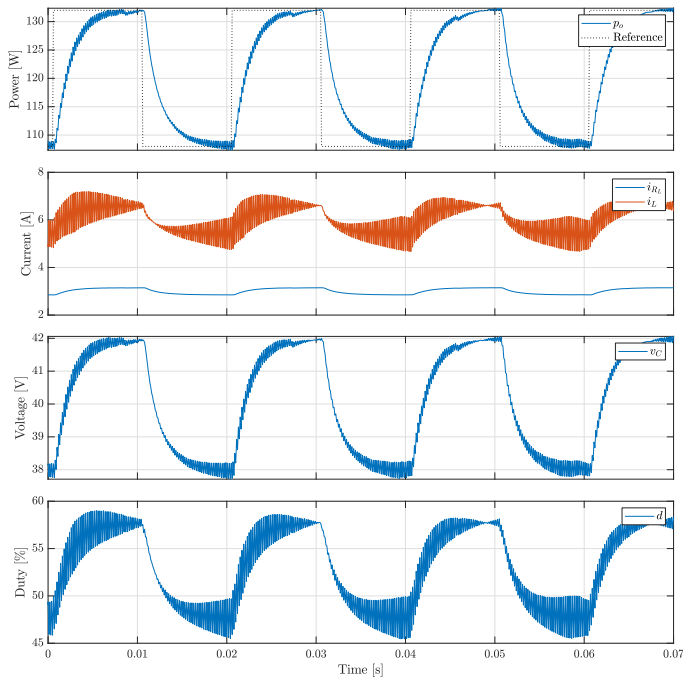


Fig. 9. Caso 1: Variação de  $\pm 10\%$  na potência de referência  $P_{ref}$ .

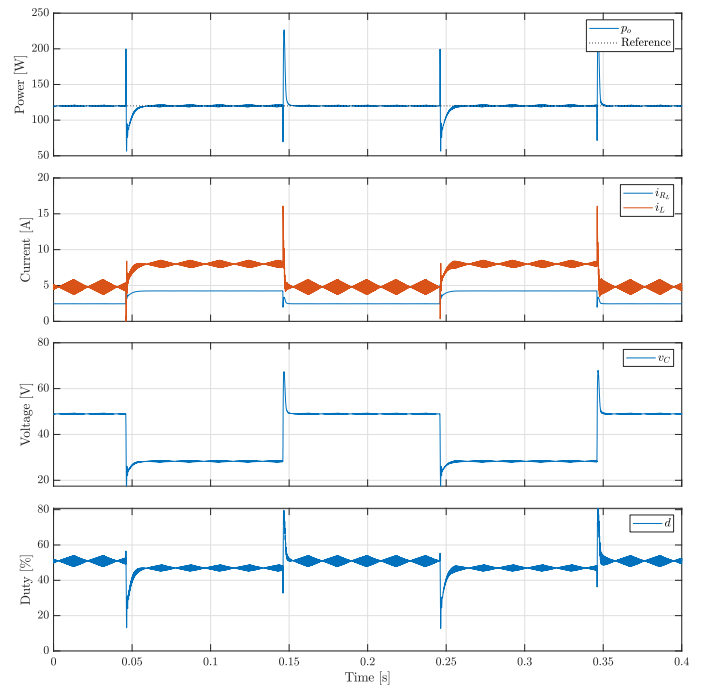


Fig. 11. Caso 3: Variação de  $\pm 50\%$  na resistência de carga  $R_L$ .

## V. CONCLUSÃO

Este artigo apresentou uma abordagem inovadora para o projeto de controladores em espaço de estados aplicados a conversores boost em sistemas de energia renovável, combinando técnicas de Otimização por Enxame de Partículas (PSO) e Aprendizado de Máquina (ML). A metodologia proposta aproveita a robustez e as capacidades de otimização global do PSO para determinar parâmetros ótimos para controladores Lineares Quadráticos Integrais (LQI), enquanto os modelos de ML aceleram a previsão desses parâmetros, possibilitando adaptabilidade em tempo real.

Os resultados demonstraram a capacidade do controlador LQI de rastrear a potência de referência de forma eficaz e mitigar oscilações sob condições operacionais variáveis. Em cenários com variações de tensão de entrada, o sistema regulou consistentemente a potência de saída e estabilizou parâmetros críticos, como a corrente do indutor, a tensão do capacitor e a corrente da carga, em um curto período transitório. Esses achados ressaltam a robustez do sistema de controle e sua capacidade de manter uma operação estável, mesmo diante de perturbações significativas na entrada.

De forma semelhante, em cenários com grandes variações na resistência da carga, o sistema de controle demonstrou sua capacidade de lidar com a alta sensibilidade do conversor a flutuações no duty cycle. A rápida compensação das perturbações e a estabilização das variáveis do sistema, incluindo a potência de saída e a tensão do capacitor, destacam a eficácia do controlador em ambientes dinâmicos.

Ao integrar a otimização baseada em PSO com o poder

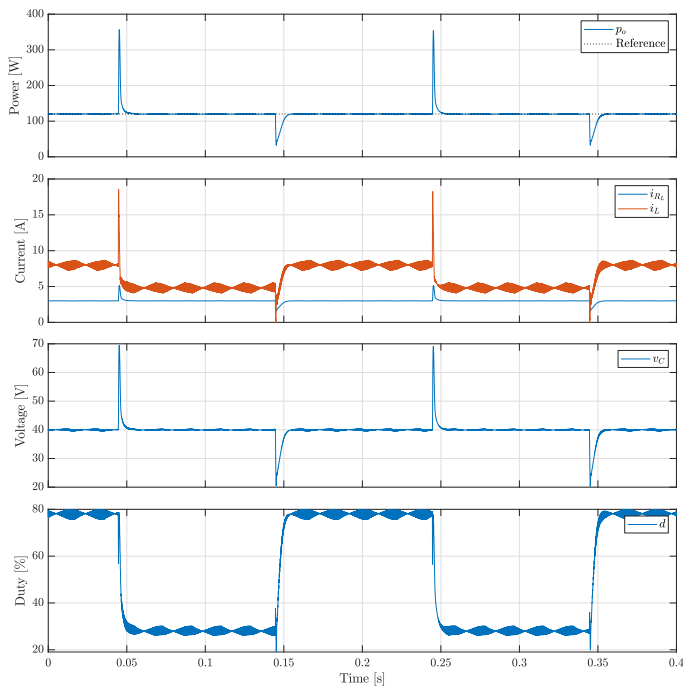
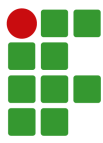


Fig. 10. Caso 2: Variação de  $\pm 25\%$  na tensão de entrada  $V_i$ .

com sucesso a potência de referência  $P_{out}$  durante o estado estacionário, demonstrando desempenho robusto.

O comportamento transitório dos diferentes casos reforça a eficácia do controlador LQI, ajustado pelas matrizes  $Q$  e  $R$  otimizadas, para garantir estabilidade e desempenho dinâmico sob diversas condições operacionais.

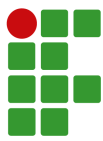


preditivo do ML, a abordagem proposta viabiliza o projeto de controladores eficientes, adaptativos e escaláveis, capazes de enfrentar as complexidades dos sistemas de energia renovável. Essa sinergia entre otimização e aprendizado não apenas melhora a robustez do sistema de controle, mas também fornece uma base para implementação em tempo real em aplicações que exigem alta adaptabilidade dinâmica.

Trabalhos futuros se concentrarão na expansão da metodologia para outras topologias de conversores e na exploração da integração de técnicas adicionais de inteligência artificial, visando aprimorar ainda mais o desempenho. Além disso, será realizada a validação experimental em protótipos de hardware para avaliar a viabilidade prática e a eficácia da abordagem proposta em cenários reais de energia renovável.

### REFERÊNCIAS

- [1] R. Bondarik, L. A. Pilatti, D. J. Horst, “Uma visão geral sobre o potencial de geração de energias renováveis no Brasil”, *Interciencia*, vol. 43, no. 10, pp. 680–688, 2018.
- [2] B. Kroposki, B. Johnson, Y. Zhang, V. Gevorgian, P. Denholm, B.-M. Hodge, B. Hannegan, “Achieving a 100% Renewable Grid: Operating Electric Power Systems with Extremely High Levels of Variable Renewable Energy”, *IEEE Power and Energy Magazine*, vol. 15, no. 2, pp. 61–73, 2017, doi:10.1109/MPE.2016.2637122.
- [3] R. S. de Araújo, F. L. N. de Sousa, P. S. Vanderley, S. O. da Silva Bentes, L. M. Gomes, F. C. L. Ferreira, “Fontes de energias renováveis: pesquisas, tendências e perspectivas sobre as práticas sustentáveis”, *Research, Society and Development*, vol. 11, no. 11, pp. e468111133893–e468111133893, 2022.
- [4] A. Fernández-Guillamón, E. Gómez-Lázaro, E. Muljadi, Ángel Molina-García, “Power systems with high renewable energy sources: A review of inertia and frequency control strategies over time”, *Renewable and Sustainable Energy Reviews*, vol. 115, p. 109369, 2019, doi:https://doi.org/10.1016/j.rser.2019.109369.
- [5] M. Dreidy, H. Mokhlis, S. Mekhilef, “Inertia response and frequency control techniques for renewable energy sources: A review”, *Renewable and Sustainable Energy Reviews*, vol. 69, pp. 144–155, 2017, doi:https://doi.org/10.1016/j.rser.2016.11.170.
- [6] ONS, “Análise da Perturbação do Dia 15/08/2023 às 08h30min – Relatório de Análise de Perturbação”, , 2023, URL: [www.ons.org.br](http://www.ons.org.br).
- [7] R. Trentini, R. Kutzner, J. J. Saldanha, A. Nied, T. J. M. Dezuco, M. S. M. Cavalca, “A Comprehensive Analysis of the Penetration of Detailed Type 4 Wind Turbine Generators in the Two-Area Benchmark System”, *Energies*, vol. 16, no. 13, p. 4970, 2023.
- [8] K. K. Patri, S. Samanta, “State feedback with integral control for boost converter & its microcontroller implementation”, in *2018 IEEMA Engineer Infinite Conference (eTechNXT)*, pp. 1–5, 2018, doi:10.1109/ETECHNXT.2018.8385374.
- [9] D. Masti, M. Zanon, A. Bemporad, “Tuning LQR Controllers: A Sensitivity-Based Approach”, *IEEE Control Systems Letters*, vol. 6, pp. 932–937, 2022, doi:10.1109/LCSYS.2021.3087556.
- [10] Y. R. Ferreira, R. Trentini, D. Greff, “State Feedback Power Control of Boost Converter Aimed at Power System Stability”, in *2024 10th International Conference on Control, Decision and Information Technologies (CoDIT)*, pp. 1804–1807, 2024, doi:10.1109/CoDIT62066.2024.10708398.
- [11] D. J. Pradeep, M. M. Noel, N. Arun, “Nonlinear control of a boost converter using a robust regression based reinforcement learning algorithm”, *Engineering Applications of Artificial Intelligence*, vol. 52, pp. 1–9, 2016.
- [12] P. V. Mahesh, S. Meyyappan, R. K. R. Alla, “A new multivariate linear regression MPPT algorithm for solar PV system with boost converter”, *ECTI Transactions on Electrical Engineering, Electronics, and Communications*, vol. 20, no. 2, pp. 269–281, 2022.
- [13] I. Barbi, *Eletrônica de Potência*, vol. único, 7ª ed., Edição do autor, 2013.
- [14] A. Attou, A. Massoum, M. Saidi, “Photovoltaic power control using MPPT and boost converter”, *Balkan journal of electrical & Computer Engineering*, vol. 2, no. 1, pp. 23–27, 2014.
- [15] N. S. Nise, “Engenharia de Sistemas de Controle. 5a Edição”, *Rio de Janeiro*, 2009.
- [16] M. S. Fadali, A. Visioli, *Digital control engineering: analysis and design*, Academic Press, 2012.
- [17] P. A. V. Ferreira, “NOTAS DE AULAS DE EA72 PRINCÍPIOS DE CONTROLE SERVOMECANISMOS”, Engenharia elétrica computação, Universidade Estadual de Campinas, 2006.
- [18] R. Y. Ferreira, “State Feedback Power Control of Boost Converter Aimed at Power System Stability”, in *CoDIT Conference*, Valetta, Malta, 2024.



- [19] S. Das, I. Pan, K. Halder, S. Das, A. Gupta, “LQR based improved discrete PID controller design via optimum selection of weighting matrices using fractional order integral performance index”, *Applied Mathematical Modelling*, vol. 37, no. 6, pp. 4253–4268, 2013.
- [20] S. H. Zak, *et al.*, *Systems and control*, vol. 198, Oxford university press New York, 2003.
- [21] O. N. Onsomu, E. Terciyanlı, B. Yeşilata, “Comprehensive review of energy management strategies: Considering battery energy storage system and renewable energy sources”, *Eng Rep*, vol. 6, no. 11, p. e12995, 2024, doi: <https://doi.org/10.1002/eng2.12995>, URL: <https://onlinelibrary.wiley.com/doi/abs/10.1002/eng2.12995>, <https://onlinelibrary.wiley.com/doi/pdf/10.1002/eng2.12995>.
- [22] H. Yang, Z. Gong, Y. Ma, L. Wang, B. Dong, “Optimal two-stage dispatch method of household PV-BESS integrated generation system under time-of-use electricity price”, *Int J Electr Power Energy Syst*, vol. 123, no. June, p. 106244, 2020, doi:10.1016/j.ijepes.2020.106244, URL: <https://doi.org/10.1016/j.ijepes.2020.106244>.
- [23] A. M. Nassef, M. A. Abdelkareem, H. M. Maghrabie, A. Baroutaji, “Review of Metaheuristic Optimization Algorithms for Power Systems Problems”, *Sustain (Switzerland)*, vol. 15, no. 12, pp. 1–27, 2023, doi: 10.3390/su15129434.
- [24] V. Tomar, M. Bansal, P. Singh, “Metaheuristic Algorithms for Optimization: A Brief Review”, *Eng Proc*, vol. 59, no. 1, 2023, doi:10.3390/engproc2023059238, URL: <https://www.mdpi.com/2673-4591/59/1/238>.
- [25] W. Yang, Z. Peng, W. Feng, M. I. Menhas, “A Novel Real-Coded Genetic Algorithm for Dynamic Economic Dispatch Integrating Plug-In Electric Vehicles”, *Front Energy Res*, vol. 9, 2021, doi: 10.3389/fenrg.2021.706782, URL: <https://www.frontiersin.org/journals/energy-research/articles/10.3389/fenrg.2021.706782>.
- [26] Y. del Valle, G. K. Venayagamoorthy, S. Mohagheghi, J.-C. Hernandez, R. G. Harley, “Particle Swarm Optimization: basic Concepts, Variants and Applications in Power Systems”, *IEEE Trans Evol Comput*, vol. 12, no. 2, pp. 171–195, 2008, doi:10.1109/TEVC.2007.896686.
- [27] M. Jahandideh-Tehrani, G. Jenkins, F. Helfer, “A comparison of particle swarm optimization and genetic algorithm for daily rainfall-runoff modelling: a case study for Southeast Queensland, Australia”, *Optim Eng*, vol. 22, no. 1, pp. 29–50, 2021, doi:10.1007/s11081-020-09538-3, URL: <https://doi.org/10.1007/s11081-020-09538-3>.
- [28] G. James, *An introduction to statistical learning*, Springer, 2013.
- [29] H. S. Krishnamoorthy, T. Narayanan Aayer, “Machine Learning based Modeling of Power Electronic Converters”, in *2019 IEEE Energy Conversion Congress and Exposition (ECCE)*, pp. 666–672, 2019, doi: 10.1109/ECCE.2019.8912608.
- [30] Y. Liu, N. Zhang, Y. Wang, J. Yang, C. Kang, “Data-driven power flow linearization: A regression approach”, *IEEE Transactions on Smart Grid*, vol. 10, no. 3, pp. 2569–2580, 2018.
- [31] N. Bhusal, R. M. Shukla, M. Gautam, M. Benidris, S. Sengupta, “Deep ensemble learning-based approach to real-time power system state estimation”, *International Journal of Electrical Power & Energy Systems*, vol. 129, p. 106806, 2021.
- [32] J. J. Saldanha, A. Nied, R. Kutzner, R. Trentini, “Active and Reactive Power Scheduling Optimization of Battery Energy Storage System to Support the Distribution System Operation”, in *2024 10th International Conference on Control, Decision and Information Technologies (CoDIT)*, pp. 260–265, IEEE, 2024.
- [33] M. I. Solihin, L. F. Tack, M. L. Kean, “Tuning of PID controller using particle swarm optimization (PSO)”, in *Proceeding of the international conference on advanced science, engineering and information technology*, vol. 1, pp. 458–461, 2011.
- [34] R. Poli, J. Kennedy, T. Blackwell, “Particle swarm optimization: An overview”, *Swarm intelligence*, vol. 1, pp. 33–57, 2007.
- [35] F. Marini, B. Walczak, “Particle swarm optimization (PSO). A tutorial”, *Chemometrics and Intelligent Laboratory Systems*, vol. 149, pp. 153–165, 2015.
- [36] S. J. Russell, P. Norvig, *Artificial intelligence: a modern approach*, Pearson, 2021.
- [37] D. C. Montgomery, E. A. Peck, G. G. Vining, *Introduction to linear regression analysis*, John Wiley & Sons, 2012.
- [38] A. Géron, *Hands-on machine learning with Scikit-Learn, Keras, and TensorFlow*, "O'Reilly Media, Inc.", 2022.