Electric Power Systems Research 254 (2026) 112671

Contents lists available at ScienceDirect

Electric Power Systems Research

journal homepage: www.elsevier.com/locate/epsr

Unified dual-PINN solution for DC-DC power converter modeling and control with fast piecewise CPL sensing

Peyman Razmi [a,*], Marcelo Godoy Simoes [a], Mohammed Elmusrati [a,b]

[a] The Smart Electric Systems (SES) Research Group, University of Vaasa, Vaasa, Finland
[b] Libyan Authority for Scientific Research, Tripoli, Libya

ARTICLE INFO

Keywords:
Constant power loads (CPLs)
Piecewise constant power loads
DC-DC buck converters
Continuous conduction mode (CCM)
Real-time parameter estimation
Physics-informed neural networks
Model predictive control (MPC)

ABSTRACT

Constant-power loads (CPLs) impose strong nonlinearities on buck converters operating in continuous conduction mode (CCM), making stability and control highly sensitive to load power and passive parameters. This work proposes a three-stage inverse-to-forward physics-informed framework for online identification and predictive control. In the first stage, an inverse PINN (iPINN) jointly estimates the piecewise-constant CPL power and the passive parameters (L, C) from voltage–current trajectories by enforcing averaged CCM dynamics and regularization terms, enabling reliable online identification during rapid transients. In the second stage, a direct PINN (DPINN) is trained using the identified parameters to construct a stable grey-box surrogate that embeds the buck conservation laws and generalizes across operating points. In the third stage, this physics-aware surrogate is integrated into a model predictive controller (MPC) to perform short-horizon duty-ratio optimization with accurate state forecasts. A key contribution of this work is a dual-PINN architecture—combining inverse estimation with forward physics-consistent prediction—that forms a unified identification–prediction–control pipeline. This integrated structure significantly reduces model–plant mismatch and enhances robustness to CPL steps and parameter drifts compared with conventional model-based or purely data-driven MPC schemes. Simulation results demonstrate (i) low-error parallel estimation of CPL and (L, C), (ii) stable convergence of the forward surrogate, and (iii) improved closed-loop MPC performance, establishing a practical pathway from iPINN-based identification to real-time control.

1. Introduction

Constant power loads (CPLs) arise in many modern DC systems such as telecommunication equipment, electric vehicle chargers, renewable energy interfaces, and DC microgrids. Unlike resistive loads whose power consumption varies proportionally with voltage, CPLs maintain a fixed power demand by adjusting the current inversely with voltage fluctuations. This behavior introduces a negative incremental impedance, whereby a small voltage drop forces an increase in current draw, effectively making the load behave as a destabilizing negative resistor. For DC-DC buck converters operating in continuous conduction mode (CCM), this phenomenon significantly reduces natural damping, alters the equilibrium structure, and can cause oscillations, poor transient performance, or even collapse of voltage regulation [1–3]. These destabilizing effects become even more pronounced in DC microgrids where multiple CPLs operate simultaneously, collectively degrading stability margins. When the CPL value is unknown, time-varying, or piecewise constant, maintaining voltage regulation becomes particularly challenging. Buck converter performance is highly sensitive to uncertainties in inductance L, capacitance C, and the instantaneous CPL power $P(t)$. In practice, loads may vary abruptly, remain piecewise constant, or change unpredictably. Such variations degrade classic voltage-regulation strategies, potentially causing overshoot, long settling times, or instability [4–6]. These challenges underscore the necessity of real-time estimation of both converter parameters and CPL power, without which controllers relying on nominal values cannot ensure robustness under rapid nonlinear disturbances. A wide range of estimation approaches has been proposed. Immersion-and-Invariance (II) observers [1], nonlinear observers [7], power/current observers [8], and reduced-order estimators [9] provide valuable insights. However, most of these methods assume smooth load profiles, require multiple high-quality sensors, or degrade under noise, parameter drift, or fast piecewise CPL variations. Parallel to estimation research, many advanced control strategies have been developed for stabilizing buck converters feeding uncertain or nonlinear loads. Passivity-Based (PB)and Energy-Shaping control(ES) methods [1,2,4] remain sensitive to inaccurate modeling. Sliding-Mode Con-

* Corresponding author.
E-mail address: peyman.razmi@uwasa.fi (P. Razmi).

https://doi.org/10.1016/j.epsr.2025.112671
Received 25 September 2025; Received in revised form 19 December 2025; Accepted 20 December 2025
Available online 30 December 2025

trol (SMC) and Sliding mode Observers (SMOs) is widely studied due to its robustness, yet chattering and noise sensitivity persist [7,8,10]. Recent extensions combine SMC with fuzzy or neural approximators, such as super-twisting SMC with self-evolving fuzzy neural networks, improving uncertainty compensation but still not estimating physical converter parameters or load power [11]. Similarly, adaptive neural-network sliding-mode controllers integrate RBF networks to enhance robustness, though they rely on predefined architectures and remain disconnected from true parameter estimation. Model Predictive Control (MPC) has gained increasing attention for its ability to enforce constraints and optimize dynamic performance [5,12,13]. However, MPC suffers from strong model-plant mismatch sensitivity, particularly when L, C, or $P(t)$ vary. Even recent real-time MPC designs require hybrid analog-digital computation hardware to meet timing constraints, highlighting MPC's computational and modeling limitations [14]. Reinforcement learning (RL)-based methods have emerged as flexible, model-free controllers [6,15,16]. Although RL can adapt to nonlinearities and disturbances, deep RL techniques demand extensive training, high computational resources, and often suffer from simulation-to-reality mismatches, especially on low-cost DSP hardware [17]. Beyond intelligent controllers, simplified or parameter-free buck control algorithms have also been proposed. A two-stage simplified controller achieves robustness without parameter tuning but does not estimate converter parameters or time-varying loads, limiting use in predictive control settings [14]. Meanwhile, high-frequency buck-based drive systems (e.g., SiC-fed PMSM drives) further emphasize the importance of accurate real-time modeling, as MHz-range switching exposes parameter sensitivity and microcontroller resource constraints [18]. Classical modeling approaches such as small-signal linearization and PD compensator designs improve transient response [19,20], but their validity deteriorates under nonlinearities, parameter drift, or rapidly varying CPL conditions. Recent studies show that high-performance buck control depends critically on dynamic models that remain accurate beyond fixed linearization points. Despite this extensive literature, a recurring limitation emerges: estimation and control are treated as separate problems. Observer-centric designs reconstruct disturbances but cannot ensure closed-loop stability under changing conditions. Control-centric approaches such as MPC, SMC, PBC, and ANN-enhanced schemes require accurate models yet offer no mechanism to identify converter parameters or CPL power in real time. RL-based and hybrid methods improve adaptability but introduce computational burden and implementation complexity. Consequently, no existing method provides a unified solution for uncertainty estimation, dynamic prediction, and robust control. In addition to the above estimation- and control-oriented methods, recent works have begun to explore the use of deep learning and physics-informed models in power-electronic applications. Deep neural networks have been applied to enhance disturbance detection and improve resilience in meshed DC systems, demonstrating the growing role of data-driven methods in power electronics. Hyperparameter selection in power-electronic learning frameworks has also been examined in recent literature. Authors in [21] employs a deep-learning classifier whose hyperparameters including learning rate, dropout ratio, batch size, and optimizer configuration are tuned empirically through manual search. The authors show that the hyperparameter choice markedly influences fault classification robustness under noisy measurements and varying network conditions. Similarly, the deep-learning-based HVDC protection scheme in [22] adopts iterative hyperparameter refinement, adjusting parameters such as the Adam learning rate, network width, activation functions, and regularization coefficients to achieve stable convergence and real-time computational performance. These studies underline the importance of hyperparameter selection for balancing accuracy, robustness, and computational feasibility in DL-based protection schemes. In sharp contrast to these purely data-driven approaches, the hyperparameters and architecture of our dual-PINN framework are determined directly by the physics constraints of the averaged buck-converter CCM model. Because both the inverse PINN (iPINN) and direct hard-PINN (DPINN) embed differential-algebraic relations, full-batch training is essential to preserve temporal consistency of the ODE residuals, and dropout must be avoided as it introduces stochastic discontinuities that disrupt derivative-based learning. The detailed architecture of the proposed dual-PINN framework-including the layer structures of the inverse PINN (iPINN) and direct PINN (DPINN), the physics-informed embedding strategy, and the complete hyperparameter selection methodology-will be presented in Dual PINN architecture Section. This dedicated section explains how the network depth, width, activation functions, Fourier-feature encoding, optimizer settings, learning-rate schedules, and gradient-stability mechanisms are chosen to satisfy the physics constraints of the averaged CCM model and to ensure reliable integration of the DPINN surrogate within the MPC horizon.

1.1. Research gap

Despite extensive progress, no existing framework simultaneously provides: (1) real-time estimation of piecewise CPL power, (2) adaptive identification of L and C, (3) a physics-consistent forward model suitable for MPC prediction, and (4) computational feasibility compatible with high-frequency converter sampling. Estimation, prediction, and control remain fragmented in prior work. As a result, existing methods cannot adequately cope with nonlinear, fast-varying CPL profiles while maintaining robustness and real-time feasibility. To fill these gaps, this paper proposes a three-echelon unified framework that integrates estimation, prediction, and control:

- **Inverse PINN for unified online estimation:** An inverse physics-informed neural network (iPINN) jointly estimates piecewise CPL power and parameters (L, C) by embedding averaged CCM dynamics, Huber regularization, and total-variation penalties, enabling robust and accurate estimation under rapid load steps.
- **Forward PINN as a physics-consistent surrogate:** A direct PINN (DPINN) uses the identified parameters to learn a grey-box dynamic model consistent with buck converter conservation laws, enabling stable and generalizable forward predictions.
- MPC with reduced computational burden: The DPINN surrogate feeds short-horizon predictions into an MPC controller, reducing model–plant mismatch and lowering computational cost relative to full nonlinear MPC.
- **Comprehensive mitigation of prior limitations:** The integrated iPINN–DPINN–MPC pipeline overcomes parameter drift sensitivity, avoids chattering, reduces data dependence, and improves transient performance under piecewise CPL dynamics.

Table 1 summarizes prior strategies and highlights the need for a unified estimation–prediction–control approach. The proposed framework directly addresses these limitations by providing a physically grounded, robust, and computationally efficient architecture for stabilizing buck converters with CPLs.

Table 1
Compact comparison of strategies for power converters with CPLs.

Method [Refs]	Robust	L, C drift	Disturb. rej.	Chatt.-free	Complex.
I&I + ES [1,4,9]	+	×	+	✓	Low
Observers [7–10]	+	×	+	✓	Med
SMO/SMC [23–25]	+ +	×	+ +	×	High
PBC [1,2,4]	+	×	+	✓	Med
MPC (adaptive) [5, 26,27]	+	✓	+ +	✓	High
ESC [1,4,28]	+	✓	+	✓	Med
RL [6,16,19]	+	✓	+	✓	High
Composite ctrl. [29]	+	✓	+ +	✓	High
Proposed (iPINN + PINN + MPC)	+ +	✓	+ +	✓	**Med–Low**

Legend: ✓ = supported, × = not supported; + = moderate, + + = strong.

In Section 2, CPL model is described and as well as its estimation involving the online identification of the inductance L and capacitance C. That subsection also describes the iPINN formulation, including all the relevant mathematical relations for real-time estimation. In Section 3, the dynamic behavior of the buck converter is assessed in both the CCM, where the modeling relies on direct PINNs, which provide a forward dynamic representation of the converter. Section 4 describes the MPC strategy by considering the accurate load and parameters of the iPINN framework and distinct predict horizon from the DPINN. Section 5 describes the training strategy. Section 6 describes the simulation setup. Section 7 provides presents and discusses the simulation results. Section 8 summarizes the content of this paper.

1.2. Definition of piecewise constant power load (PCPL)

In realistic applications, the power demand is seldom perfectly constant over extended periods. Instead, it frequently changes in distinct steps, reflecting the dynamic nature of the overall system. To accurately model these scenarios, we introduce the concept of a Piecewise Constant Power Load (PCPL). We define the PCPL profile P(t) as a series of constant power segments over predefined or adaptive time intervals:

$$P(t) = \begin{cases} P_1, & \text{if } t \in [t_0, t_1) \\ P_2, & \text{if } t \in [t_1, t_2) \\ \vdots & \\ P_N, & \text{if } t \in [t_{N-1}, t_N) \end{cases} \tag{1}$$

Here, P_k represents the constant power consumption level within the kth time segment $[t_{k-1}, t_k)$. The transition points t_k are known, but the values of P_k for each segment are unknown and are the targets for identification. For each segment k, the equivalent load resistance $R_{\text{load},k}(t)$ is not constant but is a nonlinear function of the output voltage V(t) and the constant power P_k:

$$R_{\text{load},k}(t) = \frac{V(t)^2}{P_k}, \quad \text{for } t \in [t_{k-1}, t_k) \tag{2}$$

This time-varying, voltage-dependent resistance reflects the challenge of CPLs and necessitates a robust identification method.

1.3. Mathematical and control-theoretic motivation

When theorizing about DC-DC converters, values for inductance (L) and capacitance (C) are typically constant for the sake of establishing mathematical models. This is not valid in a real-world design. In reality, values of L and C have variability and can be affected by: manufacturing tolerances, temperature drift, aging effects, core(saturation) of inductors, and capacitance (sensitivity to DC bias) of capacitors. For example, high- Q ferrite inductors could experience a 3-5% change in inductance over a 40-60 °C swing in temperature, while the operation of electrolytic capacitors could lose significant nominal capacitance (20-30%) over years of operation. The load in constant-power loads (CPL) greatly exacerbates the issues with variability on these parameters. Even small values (3-5%) of variability of L or C can draw the closed-loop poles towards the imaginary axis as shown in Fig. 1, leading to reduced phase margin and potential instability in the system. Therefore, online estimation of true values of L and C in the presence of CPLs is not just useful but critical to maintaining robust stability of more sophisticated controllers like model predictive control (MPC), adaptive controllers, or hybrid physics-informed/Kalman filter-based architectures.

Consider a Buck converter in CCM feeding a constant-power load (CPL). Let $x = [i_L \; v_C]^T$ and $u = d\,V_{\text{in}}$. The averaged nonlinear dynamics are

$$L\dot{i}_L = dV_{\text{in}} - v_C - R_L i_L, \tag{3}$$

$$C\dot{v}_C = i_L - \frac{P}{v_C}. \tag{4}$$

Linearizing around an equilibrium $(i_L^*, v_C^* = V_o)$ yields the small-signal model

$$\dot{\hat{x}} = A(L, C, P)\,\hat{x} + B(L)\,\hat{d} \tag{5}$$

$$A = \begin{bmatrix} -\frac{R_L}{L} & -\frac{1}{L} \\ \frac{1}{C} & \frac{1}{R_{\text{neg}}C} \end{bmatrix}, \quad B = \begin{bmatrix} \frac{V_{\text{in}}}{L} \\ 0 \end{bmatrix}, \tag{6}$$

where the CPL introduces a *negative incremental resistance*

$$R_{\text{neg}} = -\frac{V_o^2}{P} \quad (< 0). \tag{7}$$

The characteristic polynomial of A is

$$p(\lambda; L, C) = \lambda^2 + \left(\frac{R_L}{L} - \frac{1}{R_{\text{neg}}C}\right)\lambda + \frac{1}{LC}\left(1 + \frac{R_L}{R_{\text{neg}}}\right) = 0, \tag{8}$$

in which the CPL term $\frac{1}{R_{\text{neg}}C} > 0$ reduces the effective damping. Small parameter errors $\Delta L, \Delta C$ perturb the eigenvalues λ_i according to first-order sensitivities

$$\frac{\partial \lambda}{\partial L} = -\frac{\partial p/\partial L}{\partial p/\partial \lambda}, \qquad \frac{\partial \lambda}{\partial C} = -\frac{\partial p/\partial C}{\partial p/\partial \lambda}, \tag{9}$$

so that even ±5% deviations can move $\text{Re}\{\lambda\}$ significantly toward the imaginary axis, especially when $|R_{\text{neg}}|$ is small (large P or low V_o) as Fig. 1.

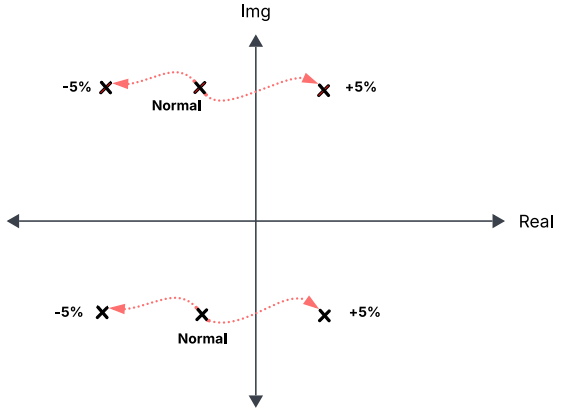


Fig. 1. Changes in poles towards instability due to incorrect estimation of inductor and capacitor values.

1.3.1. Closed-loop

For a PI controller with filtered output (a Type-II-like compensator),

$$e = -\hat{v}_C, \tag{10}$$

$$\dot{\xi} = e \tag{11}$$

$$u_{\text{raw}} = K_p e + K_i \xi, \tag{12}$$

$$\dot{u}_f = \omega_p(u_{\text{raw}} - u_f), \quad \hat{d} = K_{\text{pwm}} u_f, \tag{13}$$

the augmented state $[\hat{i}_L, \hat{v}_C, \xi, u_f]^T$ obeys

$$\dot{z} = \underbrace{\begin{bmatrix} -\frac{R_L}{L} & -\frac{1}{L} & 0 & \frac{V_{\text{in}}K_{\text{pwm}}}{L} \\ \frac{1}{C} & \frac{1}{R_{\text{neg}}C} & 0 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & -\omega_p K_p & \omega_p K_i & -\omega_p \end{bmatrix}}_{A_{\text{cl}}(L,C,P,K_p,K_i)} z. \tag{14}$$

Because the CPL reduces damping, the closed-loop pole locations are highly sensitive to (L, C); model mismatch (e.g., due to ageing/temperature) can push poles into the RHP and destabilize an otherwise correctly tuned controller. Hence, accurate online identification of L, C is a stability requirement for MPC/PI/Type-III designs.

Key claim. With CPL, the term $\frac{1}{R_{\text{neg}}C}$ in (8) amplifies the effect of $\Delta L, \Delta C$ on $\text{Re}\{\lambda\}$; thus maintaining an accurate plant model via online parameter estimation (e.g., EKF + iPINN) is essential to preserve phase margin and closed-loop stability.

2. Buck converter system dynamics

We consider a standard buck converter topology operating with a Constant Power Load (CPL) as its load as Fig. 2. The converter consists of an input voltage source V_{in}, a power switch (MOSFET), a diode, an inductor L, and an output capacitor C. To model losses, the equivalent series resistance R of the inductor is also included. The system dynamics are governed by differential equations for the inductor current $i_L(t)$ and the capacitor voltage $v_C(t)$, which is also the output voltage. The converter operates in either Continuous Conduction Mode (CCM) or Discontinuous Conduction Mode (DCM). Both modes are explicitly modeled to allow comprehensive load identification.

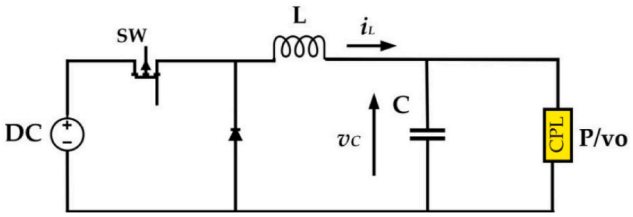


Fig. 2. Buck converter supplying a constant-power load (CPL). The CPL introduces negative incremental impedance, significantly affecting the converter stability and transient behavior.

2.1. States and inputs

The state vector is defined as

$$x(t) = \begin{bmatrix} i_L(t) \\ v_C(t) \end{bmatrix}, \tag{15}$$

where $i_L(t)$ is the inductor current and $v_C(t)$ is the capacitor voltage (equal to the output voltage).

The inputs are

$$D(t) \in [0, 1], \quad V_{\text{in}} > 0, \tag{16}$$

where $D(t)$ is the duty cycle and V_{in} is the DC input voltage.

2.2. Parameters

$$L > 0, \quad C > 0, \quad R_p > 0, \tag{17}$$

which L indicates inductance, C represents capacitance and R_p indicates parallel resistance (to model leakage or an ohmic load).

The CPL power is

$$P(t) \geq 0, \quad \text{(piecewise-constant or continuous)}. \tag{18}$$

For numerical stability:

$$v_{\min} > 0, \qquad I_{\text{CPL,max}} > 0. \tag{19}$$

2.3. Continuous-time dynamics

Ideal model (no parasitics).

$$\dot{i}_L(t) = \frac{D(t)V_{\text{in}} - v_C(t)}{L}, \tag{20}$$

$$\dot{v}_C(t) = \frac{1}{C}\left(i_L(t) - \left(\frac{P(t)}{\max\{v_C(t), v_{\min}\}} + \frac{v_C(t)}{R_p}\right)\right). \tag{1}$$

With winding resistance R_w and capacitor ESR.

$$\dot{i}_L(t) = \frac{D(t)V_{\text{in}} - v_C(t) - R_w i_L(t)}{L}, \tag{21}$$

$$\dot{v}_C(t) = \frac{1}{C}\left(i_L(t) - \frac{P(t)}{v_C(t)} - \frac{v_C(t)}{R_p}\right) + \text{ESR} \cdot \dot{i}_L(t). \tag{22}$$

Load current (CPL + resistive part).

$$i_{\text{cpl}}(t) = \min\left(\frac{P(t)}{\max\{v_C(t), v_{\min}\}}, I_{\text{CPL,max}}\right), \tag{23}$$

$$i_{\text{ld}}(t) = i_{\text{cpl}}(t) + \frac{v_C(t)}{R_p}. \tag{24}$$

2.4. Discretization

Sampling at interval T_s:

$$t_n = nT_s, \quad n = 0, \ldots, N.$$

Forward-difference approximations:

$$\dot{v}_n^{\text{meas}} = \frac{v_{n+1} - v_n}{T_s}, \tag{25}$$

$$\dot{i}_n^{\text{meas}} = \frac{i_{n+1} - i_n}{T_s}. \tag{26}$$

Runge-Kutta 4 (for simulation):

$$x_{n+1} = x_n + \frac{T_s}{6}(k_1 + 2k_2 + 2k_3 + k_4), \tag{27}$$

$$k_\ell = f\left(x_n + \alpha_\ell T_s k_{\ell-1}\right), \tag{28}$$

where $f(\cdot)$ is given by (1) or (2).

3. Physics informed neural network

As Fig. 3, Physics-informed neural networks (PINNs) are neural networks that incorporate physical laws described by differential equations into their loss functions to guide the learning process toward solutions that are more consistent with the underlying physics. PINNs can be used to:

- Approximate solutions to partial differential equations (PDEs) and ordinary differential equations (ODEs) as a Direct PINN.
- Solve inverse problems, such as estimating model parameters from limited data as a Inverse PINN (iPINN.

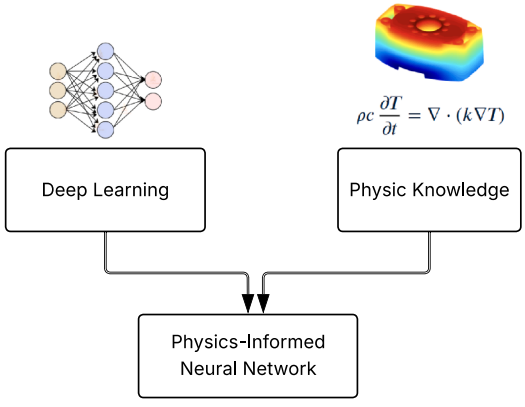


Fig. 3. Conceptual illustration of a physics-informed neural network (PINN), which combines deep-learning representations with physical governing equations.

At first we introduce Inverse physic informed NN for parameter estimation like CPL load, L and C values in two-Stage Unified Framework for Buck–CCM with CPL (iPINN + ALS) as below:

A) Stage 1- Inverse PINN Estimation of $P(t), L, C$

a) Parametrization with bounds.

$$P(t) = P_k, \qquad t \in [t_k, t_{k+1}), \tag{29}$$

$$P_k = P_{\min} + (P_{\max} - P_{\min})\,\sigma(z_k), \tag{30}$$

$$\sigma(x) = \frac{1}{1+e^{-x}}, \tag{31}$$

$$L = L_{\min} + (L_{\max} - L_{\min})\,\sigma(z_L), \tag{32}$$

$$C = C_{\min} + (C_{\max} - C_{\min})\,\sigma(z_C). \tag{33}$$

b) Neural architecture.

$$\big(P_\theta(\tau),\ L_\theta,\ C_\theta\big), \qquad \tau \in [0,1]. \tag{34}$$

with ranges

$$P_\theta(\tau) \in [P_{\min}, P_{\max}], \tag{35}$$

$$L_\theta \in [L_{\min}, L_{\max}], \tag{36}$$

$$C_\theta \in [C_{\min}, C_{\max}]. \tag{37}$$

c) Physics residuals.

$$r_v(n;\theta) = v_n^{\text{meas}} - \frac{1}{C_\theta}\left(i_n - \left(P_\theta(\tau_n)\max(v_n, v_{\min}) + \tfrac{v_n}{R_p}\right)\right) \tag{38}$$

$$r_i(n;\theta) = i_n^{\text{meas}} - \frac{1}{L_\theta}\big(DV_{\text{in}} - v_n\big). \tag{39}$$

d) Huber penalty.

$$\rho_\delta(r) = \begin{cases} \frac{1}{2}r^2, & |r| \le \delta, \\ \delta\left(|r| - \frac{\delta}{2}\right), & |r| > \delta. \end{cases} \tag{40}$$

e) Regularizers.

$$TV_P = \frac{1}{N-1}\sum_{n=0}^{N-2}\Big|P_\theta(\tau_{n+1}) - P_\theta(\tau_n)\Big|\,\chi_{\text{in-seg}}(n). \tag{41}$$

$$P_{\text{inst}}(t_n) = v_n\left(i_n - C_\theta\,\dot{v}_n - \tfrac{v_n}{R_p}\right). \tag{42}$$

$$L_{\text{seg}} = \frac{1}{K}\sum_{k=0}^{K-1}\left(\overline{P_\theta}^{(k)} - \overline{P_{\text{inst}}}^{(k)}\right)^2. \tag{43}$$

IC constraints.

$$r_{\dot{v},0} = \dot{v}^{\text{meas}}(t_0) - \frac{1}{C_\theta}\Big(i(t_0) - \big(P_\theta(\tau_0)\max\big(v(t_0), v_{\min}\big) + \tfrac{v(t_0)}{R_p}\big)\Big), \tag{44}$$

$$r_{i,0} = i^{\text{meas}}(t_0) - \frac{1}{L_\theta}\big(DV_{\text{in}} - v(t_0)\big), \tag{45}$$

$$L_{\text{IC-der}} = \rho_\delta(r_{\dot{v},0}) + \gamma_L\,\rho_\delta(r_{i,0}), \tag{46}$$

$$L_{\text{IC-state}} = \lambda_{v0}\big(v(t_0) - V_0\big)^2 + \lambda_{i0}\big(i(t_0) - I_0\big)^2. \tag{47}$$

Full iPINN loss.

$$\mathcal{L}_{\text{iPINN}}^{\text{full}}(\theta) = \frac{\sum_n w_n\big(\rho_\delta(r_v(n;\theta)) + \gamma_L\,\rho_\delta(r_i(n;\theta))\big)}{\sum_n w_n} + \lambda_{TV}\,TV_P + \lambda_{\text{seg}}\,L_{\text{seg}} + \mu_L\|L_\theta - \overline{L}\|^2 + \mu_C\|C_\theta - \overline{C}\|^2 + \lambda_{\text{ICd}}\,L_{\text{IC-der}} + \lambda_{\text{ICs}}\,L_{\text{IC-state}}. \tag{48}$$

Stage-1 outputs.

$$\hat{P}_k = \overline{P_\theta}^{(k)}, \qquad \hat{L} = L_\theta, \qquad \hat{C} = C_\theta. \tag{49}$$

3.1. Capacitor voltage dynamics (common in both phases):

$$f_{v_C}(v_C, i_L, \hat{P}) = \frac{1}{C}\left(i_L - \frac{\hat{P}(v_C, i_L)}{v_C}\right) \tag{50}$$

In Stage 1, the inverse PINN framework follows Algorithm 1 to jointly estimate the piecewise-constant load $P(t)$ together with inductance L and capacitance C. The parametrization in (34)-(37) ensures that all estimated quantities remain bounded within physically meaningful ranges. In particular, $P(t)$ is modeled as a constant value P_k within each segment $[t_k, t_{k+1})$, while L and C are scaled through a sigmoid function to enforce positivity and upper limits. The neural architecture in (34) outputs the triplet $(P_\theta(\tau), L_\theta, C_\theta)$ as a function of normalized time τ. Explicit limitations in (35)-(37) restrict $\big(P_\theta(\tau),\ L_\theta,\ C_\theta\big)$ to specified ranges that stabilize learning and preclude non-physical solutions. The physics residuals in (38)-(39) encode Kirchhoff's circuit equations: the capacitor equation links the measured voltage derivative to current and load power, while the inductor equation relates the current derivative to the input-output voltage difference. Any mismatch between measurements and physics is penalized, thereby embedding the dynamics directly into the learning mechanism. The Huber penalty in (40) adds robustness against noise: small residuals are penalized quadratically, whereas large deviations are penalized linearly, preventing instability due to outliers. The regularization terms in (41)-(43) follow Algorithm 2 to improve estimation stability. The total variation TV_P discourages unnecessary fluctuations of P_θ within a segment, while $P_{\text{inst}}(t)$ enforces consistency between the predicted and measured instantaneous power. The segment-level loss L_{seg} aligns the average predicted power with the average measured power in each segment. Equations (44)-(47) enforce initial condition consistency. The residuals $r_{\dot{v},0}$ and $r_{i,0}$ require that initial derivatives satisfy circuit dynamics. The penalties $L_{\text{IC-der}}$ and $L_{\text{IC-state}}$ guarantee that both derivatives and the initial states $\big(v(t_0),\ i(t_0)\big)$ are consistent with measurements. Finally, the full iPINN loss in (48) aggregates all components: a weighted average of physics residuals, regularizers, and initial condition penalties, together with soft constraints keeping L and C close to their a priori means; all losses are shown in Fig. 4 within the direct PINN structure. In a direct physic informed neural network, the only input is time, and the network produces the system states themselves it reconstructs the capacitor voltage and the inductor current directly from time. During training, the network computes time-derivatives internally and combines them with measurements, initial conditions, and circuit constraints so that the overall loss is minimized; the goal is to predict the states. By contrast, an inverse physics informed neural network takes not only time but also the measured responses capacitor voltage and inductor current as inputs. It estimates the unknown quantities of interest, such as the constant power load over time in the first stage and the circuit parameters (inductance and capacitance) in the second stage. Learning is driven by consistency with the circuit equations, so the network is guided toward parameters that best match both the measurements and the governing physics. This makes the direct approach suitable for state prediction, while the inverse approach is preferred when the goal is to recover hidden parameters from time and measured responses, especially under scarce or noisy data. Finally by minimizing loss function as illustrated in Fig. 4, the model ensures consistency with data, physics, and prior knowledge simultaneously. The Stage 1 outputs in (49) are the estimated piecewise load values $\hat{P}_k$, together with globally consistent inductance $\hat{L}$ and capacitance $\hat{C}$, which then serve as initialization for Stage 2 refinement; in the second stage, the respective final values are estimated according to Algorithm 2.

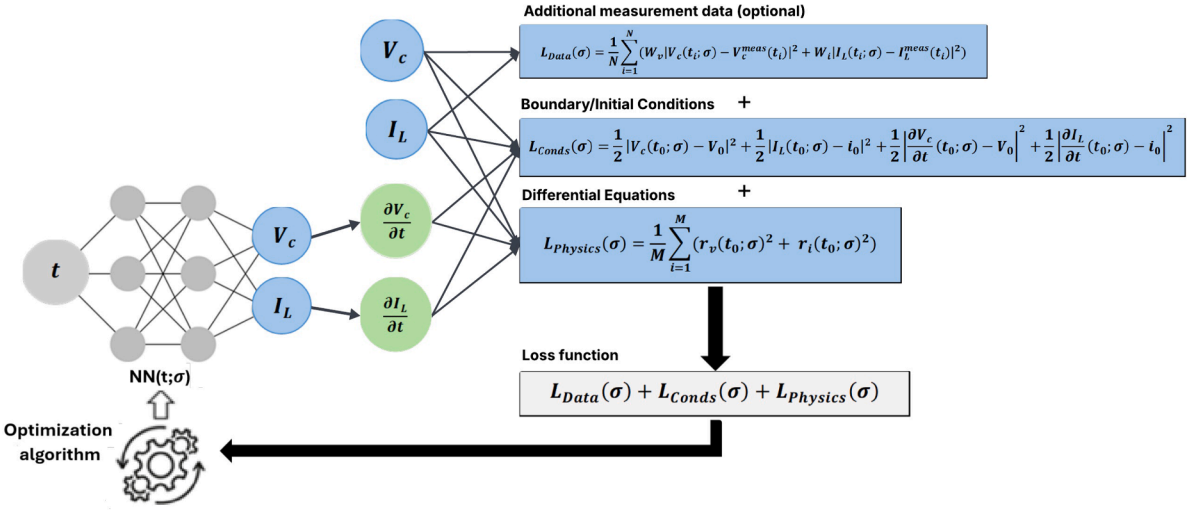


Fig. 4. DPINN structure integrating data, boundary conditions, and physics-based ODE residuals into a unified loss for training.

4. Stage 2 - direct PINN for forward state prediction

4.1. Problem setting

After Stage 1, the parameter estimates $\hat{P}(t)$, $\hat{L}$, $\hat{C}$ are available. The goal of Stage 2 is to train a direct PINN predictor that maps

$$(x_0, \{D_h\}_{h=0}^{H-1}) \longrightarrow \{(i_L, v_C)_h\}_{h=1}^{H}, \tag{51}$$

where

- $x_0 = [i_L(0), v_C(0)]^\top$ is the initial state,
- $\{D_h\}$ is the duty ratio sequence for the horizon,
- the output is the trajectory of inductor current and capacitor voltage.

This predictor serves as the surrogate model inside the MPC optimizer.

4.2. Forward dynamics with fixed parameters

The averaged Buck–CCM dynamics are given by

$$\dot{i}_L(t) = \frac{D(t)V_{in} - v_C(t) - R_w i_L(t)}{\hat{L}}, \tag{52}$$

$$\dot{v}_C(t) = \frac{i_L(t) - \left(\frac{\hat{P}(t)}{\max(v_C(t), v_{\min})} + \frac{v_C(t)}{R_p}\right)}{\hat{C}} + \mathrm{ESR} \cdot \dot{i}_L(t). \tag{53}$$

4.3. Normalization

We normalize the horizon $[0, T_h]$ into $\tau \in [0, 1]$, with $T_h = HT_s$. Residual forms become

$$r_{i_L}(\tau) := \frac{di_L}{d\tau} - T_h \frac{D(\tau)V_{in} - v_C - R_w i_L}{\hat{L}}, \tag{53}$$

$$r_{v_C}(\tau) := \frac{dv_C}{d\tau} - T_h \left[i_L - \left(\frac{\hat{P}(\tau)}{\max(v_C, v_{\min})} + \frac{v_C}{R_p} \right) \right] \frac{1}{\hat{C}} + \mathrm{ESR}\, \frac{D(\tau)V_{in} - v_C}{\hat{L}}. \tag{54}$$

4.4. Neural predictor

We construct a neural predictor

$$\hat{x}(\tau; \vartheta) = \begin{bmatrix} \hat{i}_L(\tau) \\ \hat{v}_C(\tau) \end{bmatrix},$$

with inputs:

- normalized time τ,
- initial state x_0,
- encoded duty sequence $\phi_D(\tau)$,
- encoded CPL estimate $\phi_P(\tau)$.

4.5. Loss function

The training loss is a weighted sum:

$$\mathcal{L}_{phys} = \frac{1}{J} \sum_{j=1}^{J} \left(\| r_{i_L}(\tau_j) \|^2 + \| r_{v_C}(\tau_j) \|^2 \right), \tag{54}$$

$$\mathcal{L}_{IC} = \| \hat{x}(0) - x_0 \|^2 + \lambda_{\dot{x}0} \left\| \frac{d\hat{x}}{d\tau}(0) - T_h f(x_0, D_0) \right\|^2, \tag{55}$$

$$\mathcal{L}_{data} = \frac{1}{|S|} \sum_{n \in S} w_n \| \hat{x}(\tau_n) - x_n^{meas} \|^2, \tag{56}$$

$$\mathcal{L}_{reg} = \beta_{Lip} \int_0^1 \left\| \frac{\partial \hat{x}}{\partial \tau} \right\|^2 d\tau + \beta_W \| \vartheta \|^2. \tag{57}$$

The total loss is

$$\mathcal{L}_{Stage2} = \lambda_{phys}\mathcal{L}_{phys} + \lambda_{IC}\mathcal{L}_{IC} + \lambda_{data}\mathcal{L}_{data} + \lambda_{reg}\mathcal{L}_{reg}. \tag{58}$$

4.6. Output mapping

After training, the predictor yields

$$f_{pred}(x_0, \{D_h\}) = \{\hat{x}(h/H; \vartheta^\star)\}_{h=1}^{H},$$

which is the predicted trajectory (i_L, v_C) over the horizon. This trajectory is then used by the MPC cost function and constraints.

5. Stage 3 - MPC control with PINN-based predictor

At this stage, we already have the estimated load and parameters from Stage 1, $\hat{P}(t), \hat{L}, \hat{C}$, and also the trained forward predictor from Stage 2, $f_{\text{pred}}(x_0, \{D_h\}_{h=0}^{H-1}) \to \{x_h\}_{h=1}^{H}$, where $x_h = (i_L, v_C)$. The goal is to compute an optimal duty sequence $\{D_h^\star\}_{h=0}^{H-1}$ that minimizes a performance index subject to constraints. For MPC optimization problem at each sampling step, the controller solves:

$$\min_{\{D_h\}_{h=0}^{H-1}} J = \sum_{h=1}^{H} \Bigg(\underbrace{\| v_C(h) - v_{\text{ref}} \|_Q^2}_{\text{Voltage tracking}} + \underbrace{\| i_L(h) \|_R^2}_{\text{Current penalty}} + \underbrace{\lambda_\Delta \| D_h - D_{h-1} \|^2}_{\text{Duty variation penalty}} \Bigg) \tag{59}$$

Algorithm 1: Joint inverse iPINN for parameter estimating.

Input :
1) Measured trajectories $\{(v_n, i_n)\}_{n=0}^{N}$
2) sampling time T_s
3) bounds $[P_{\min}, P_{\max}]$, $[L_{\min}, L_{\max}]$, $[C_{\min}, C_{\max}]$
4) known V_{in}, fixed D, R_p, $v_{\min}$
5) hyperparameters $\delta, \lambda_{TV}, \lambda_{\text{seg}}, \gamma_L, \mu_L, \mu_C$
6) LR schedule, gradient clipping norm.

Output :
1) Piecewise CPL $\{\hat{P}_k\}_{k=0}^{K-1}$
2) inductance $\hat{L}$, capacitance $\hat{C}$.

Preprocess :
1) Compute forward differences (Eq. (25–26)):

$$\dot{v}_n^{\text{meas}} = \frac{v_{n+1} - v_n}{T_s}, \qquad \dot{i}_n^{\text{meas}} = \frac{i_{n+1} - i_n}{T_s}.$$

2) Build normalized time $\tau_n = \dfrac{n}{N-1}$
3) Segment masks $\chi_{\text{in-seg}}(n)$ (zero on segment boundaries).
4) Optionally build time weights w_n (larger near boundaries), build a smoothed derivative $\tilde{v}_n$ (EMA or Savitzky–Golay).

Network : Define an MLP with Fourier features of τ producing $(P_\theta(\tau), L_\theta, C_\theta)$ with bounded outputs.

for $t = 1$ **to** T **to do**

`--- Forward pass ---`

1) Evaluate $P_\theta(\tau_n)$, L_θ, C_θ.

`--- Physics residuals (Eq. (38-39)) ---`

2) Calculate residuals

a)

$$r_v(n;\theta) = \dot{v}_n^{\text{meas}} - \frac{1}{C_\theta}\left(i_n - \left(P_\theta(\tau_n)\max(v_n, v_{\min}) + \frac{v_n}{R_p}\right)\right)$$

b) $r_i(n;\theta) = \dot{i}_n^{\text{meas}} - \dfrac{1}{L_\theta}\left(DV_{in} - v_n\right)$.

`--- Regularizers ---`

3) $P_{\text{inst}}(t_n) = v_n\left(i_n - C_\theta\,\tilde{v}_n - \dfrac{v_n}{R_p}\right)$.

4) $L_{\text{seg}} = \dfrac{1}{K}\sum_k\left(\overline{P}_\theta(k) - \overline{P}_{\text{inst}}(k)\right)^2$.

`--- Loss (Eq. (48)) ---`

5)

$$\mathcal{L}_{\text{iPINN}} = \frac{\sum_n w_n\left(\rho_\delta(r_v) + \gamma_L\,\rho_\delta(r_i)\right)}{\sum_n w_n} + \lambda_{TV}\,TV_P + \lambda_{\text{seg}}\,L_{\text{seg}} + \mu_L\,\|L_\theta - \bar{L}\|^2 + \mu_C\,\|C_\theta - \bar{C}\|^2.$$

`--- Backward + update ---`

1) Compute $\nabla_\theta \mathcal{L}_{\text{iPINN}}$; scale/throttle C-head grads; clip; update θ (Adam).

`--- EMA (stability) ---`

2) $\overline{L} \leftarrow \beta_L L_\theta + (1-\beta_L)\overline{L}, \qquad \overline{C} \leftarrow \beta_C C_\theta + (1-\beta_C)\overline{C}$.

3 Postprocess: Average $P_\theta(\tau)$ over each segment to obtain

$\hat{P}_k = \overline{P}_\theta(k)$; set $\hat{L} = \overline{L}$, $\hat{C} = \overline{C}$.

Algorithm 2: Optional ALS Refinement of CPL and (L, C).

Input : Data $\{(v_n, i_n)\}$, T_s, known V_{in}, D, R_p; initial $(\hat{P}_k, \hat{L}, \hat{C})$ from Alg. 1; bounds on L, C; trimming quantiles (q_ℓ, q_h).

Output: Refined $\{\hat{P}_k\}$ and $(\hat{L}, \hat{C})$.

repeat

1) (Update P) With current $\hat{C}$, compute smoothed $\tilde{v}_n$ and instantaneous power

$$P_{\text{inst}}(t_n) = v_n\left(i_n - \hat{C}\,\tilde{v}_n - \frac{v_n}{R_p}\right).$$

For each segment k, trim outliers to (q_ℓ, q_h) and set

$$\hat{P}_k \leftarrow \text{median}\left\{\,P_{\text{inst}}(t_n)\,\right\}_{n\in k}.$$

2) (Update L, C) Minimize the bounded Huber objective (Eq. (32-33)) w.r.t. (L, C) using

$$L = L_{\min} + (L_{\max} - L_{\min})\,\sigma(z_L),$$
$$C = C_{\min} + (C_{\max} - C_{\min})\,\sigma(z_C).$$

until for $r = 1, \ldots, R$

Until convergence or max cycles.

Constraints:

$$(i_L(h), v_C(h)) = f_{\text{pred}}\left(x_0, \{D_\ell\}_{\ell=0}^{h-1}\right), \tag{60}$$

$$0 \le D_h \le 1, \tag{61}$$

$$v_{\min} \le v_C(h) \le v_{\max}, \tag{62}$$

$$i_{\min} \le i_L(h) \le i_{\max}. \tag{63}$$

- **Term 1 (Voltage tracking):** drives the capacitor voltage $v_C(h)$ towards the reference v_{ref}, weighted by Q.
- **Term 2 (Current penalty):** penalizes inductor current magnitude to ensure physical limits and smooth dynamics, weighted by R.
- **Term 3 (Duty variation penalty):** penalizes fast changes in duty cycle D_h, reducing switching stress and improving stability.

Here, Q, R, λ_Δ are design weights, and f_{pred} is the PINN-based rollout model from Stage 2. Only the first control action is applied:

$$D^\star(t) = D_0^\star, \tag{64}$$

then the horizon shifts forward and the process repeats with the new measured state. The physics-informed predictor ensures consistency with the underlying circuit laws. In addition, the online MPC optimization explicitly accounts for the CPL dynamics and parameter variations. Finally, safety constraints on capacitor voltage and inductor current are enforced directly within the optimization problem.

6. Architecture and hyperparameter design of the dual-PINN framework

The proposed dual-PINN framework consists of two complementary neural modules an inverse PINN (iPINN) for real-time estimation of piecewise constant power loads and a forward hard-PINN (DPINN) that provides a physics-consistent surrogate of the buck converter dynamics for use inside MPC. Unlike data driven neural architectures, both networks embed the averaged CCM equations directly into their loss

functions, resulting in training behavior strongly governed by physical constraints rather than empirical heuristics. This section details the architecture, training procedure, and hyperparameter selection of both networks. In Stage 1, the inverse PINN (iPINN) addresses an inverse identification problem, where the network simultaneously estimates the piecewise constant CPL power profile as well as the passive parameters of the converter. The iPINN takes as inputs the normalized time variable together with measured state trajectories (inductor current and capacitor voltage), and its outputs include the estimated CPL power $\widehat{P}(t)$ and the passive parameters $(\widehat{L}, \widehat{C})$. To represent fast load transitions, the time input is augmented using Fourier-feature encoding. The core network is implemented as a fully connected multilayer perceptron with two hidden layers of 64 neurons each and tanh activation functions. A bounded output head with sigmoid scaling is used for $\widehat{P}(t)$ to ensure physically admissible power estimates, while positivity of $\widehat{L}$ and $\widehat{C}$ is enforced via softplus parameterization. The iPINN training objective combines four components: a data-consistency loss that anchors the reconstructed states to measurements, an initial-condition loss that enforces consistency at the beginning of the trajectory, a physics-based residual loss derived from the averaged CCM buck equations, and a total-variation regularization term that promotes piecewise-constant behavior of the CPL power. These terms are explicitly formulated in Eqs. (29)-(50). Automatic differentiation is used to compute temporal derivatives required by the physics residuals. In Stage 2, the direct PINN (DPINN) solves the forward dynamical problem and serves as a physics-consistent surrogate model inside the MPC prediction horizon. The DPINN maps the normalized time, initial state, and the duty-cycle sequence to predicted trajectories of the inductor current and capacitor voltage. A hard-constrained output parameterization is adopted to guarantee exact satisfaction of initial conditions. These terms are explicitly formulated in Eqs. (51)-(58). Architecturally, the DPINN employs a three-layer fully connected network with 64 neurons per layer, tanh activation, and the same Fourier-feature encoding used in the iPINN. The loss function consists primarily of weighted physics residuals enforcing both inductor and capacitor dynamics, with optional data-matching terms when measurements are available. Regarding hyperparameter selection, both networks are trained using the Adam optimizer in a full-batch setting. This choice is necessary because PINN residual minimization relies on global temporal consistency, which would be disrupted by mini-batching. Dropout and batch normalization are deliberately omitted, as they introduce stochastic discontinuities that interfere with derivative-based physics constraints. Learning rates are chosen based on stability considerations of the underlying differential equations rather than empirical trial-and-error: a constant rate of 10^{-3} is used for the iPINN, while a two-stage schedule of 3×10^{-3} followed by 5×10^{-4} is employed for the DPINN to stabilize training under stiff dynamics. Gradient clipping is applied in DPINN training to prevent exploding gradients. This physically grounded hyperparameter design yields PINNs whose convergence behavior is predictable, interpretable, and reliable when embedded in MPC. Fig. 4 illustrates the computation graph of the direct PINN, showing the outputs (v, i), their derivatives, the incorporation of differential equations and boundary conditions, and the final composite loss function. All architectural choices and hyperparameters are summarized in Table 2. Fig. 5 illustrates the overall architecture of the proposed dual-PINN framework. The iPINN estimates the piecewise CPL power and converter parameters from measured data using physics-informed losses, while the DPINN exploits the identified quantities to generate forward state predictions that are directly used within the MPC optimization.

Unlike deep-learning-based power-electronic protection schemes [21, 22], which tune hyperparameters (learning rate, dropout, batch size, regularization, network width, activation) empirically through trial-and-error, our hyperparameter selections are physics-driven. Specifically:

- full-batch training preserves ODE structure,
- learning rates are selected for ODE stability rather than data fitting,
- Fourier embeddings help capture non-smooth CPL transitions,
- no dropout/batch-normalization due to incompatibility with PINN derivatives,
- TV regularization enforces piecewise-CPL behavior.

7. Simulation and results

The results in Figs. 6-7 clearly demonstrate the effectiveness of the proposed two-sub-stage iPINN framework. In sub-Stage 1 as illustrated in Fig. 7, the performance of the proposed two-Stage inverse PINN (iPINN) in estimating the piecewise-constant CPL power profile is evaluated. The figure compares the true load power $P(t)$ with the estimated power $\widehat{P}(t)$ obtained from the iPINN. As shown, the proposed inverse identification framework accurately captures both the magnitude and the switching instants of the CPL power, despite the abrupt load transitions. The close agreement between the estimated and true power profiles demonstrates the effectiveness of the physics-informed loss formulation and the total-variation regularization in enforcing the piecewise-constant structure of the load. Also the framework estimates the converter parameters at the same time as it can be seen in Fig. 6(a) and (b). Fig. 6(a) shows that the estimated inductance L converges with a very high level of accuracy to the nominal value of 950 μH and in Fig. 6 (b), one can see that the estimated capacitance C smoothly converges to its nominal reference of 350 μF with negligible error. These results show that the proposed approach not only reliably identifies the time-varying CPL in real-time, but also accurately tracks L and C online, thus faithfully representing both the load dynamics and the converter parameters for control and stability analysis. In these convergence plots, s (proxy) denotes a display variable derived from optimization iterations rather than real time. It simply tracks training progress, so increasing s indicates further training and tighter convergence; numerical values of s should not be compared to seconds. These results confirm that Stage 1 provides reliable power estimates and parameter identifications that can be subsequently used by the forward DPINN for prediction and MPC integration. Fig. 8 illustrates the performance of the DPINN in Stage 2. In Fig. 8(a), the red curve corresponds to the reference capacitor voltage obtained from the CPL$\|$ R_p model, while the blue dashed curve shows the voltage trajectory predicted by the DPINN. The red curve in Fig. 8(b) indicates the reference inductor current, while the blue dashed curve shows the associated DPINN prediction. It is clear that the predicted profiles and behaviors for both the inductor voltage and current follow the corresponding reference responses closely, capturing both transient oscillations and their steady-state behaviors with a high degree of accuracy. This demonstrates that the DPINN framework provides an effective forward model for later injection into MPC algorithms.

Fig. 9 illustrates the open-loop performance of the proposed dual-PINN framework under a sudden step change in the constant power load (CPL) at $t \approx 0.02$ s. Fig. 9(a) shows the capacitor voltage trajectories $v_C(t)$ for different duty ratios, obtained from the reference RK4 simulation using the true model and from the forward hard-PINN (DPINN) driven by the estimated CPL power. As expected, the abrupt change in the CPL excites transient oscillations and overshoot in the output voltage. These oscillatory behaviors are consistently reproduced by both the RK4 reference and the PINN-based predictor, demonstrating that the DPINN accurately captures the transient and nonlinear dynamics of the converter. Since no feedback controller is applied, the observed voltage distortions are physically meaningful and reflect the true open-loop system response. Fig. 9(b) depicts the true and estimated CPL power profiles. The estimated power $\widehat{P}(t)$ closely matches the true piecewise-constant load profile, including the exact switching instant and magnitude of the step change. This confirms that the Stage 1 inverse PINN reliably identifies sudden load transitions in real time. Together, Fig. 9(a) and (b) validate that the accurate CPL power estimation in Stage 1 enables the DPINN in Stage 2 to generate precise forward state predictions, making

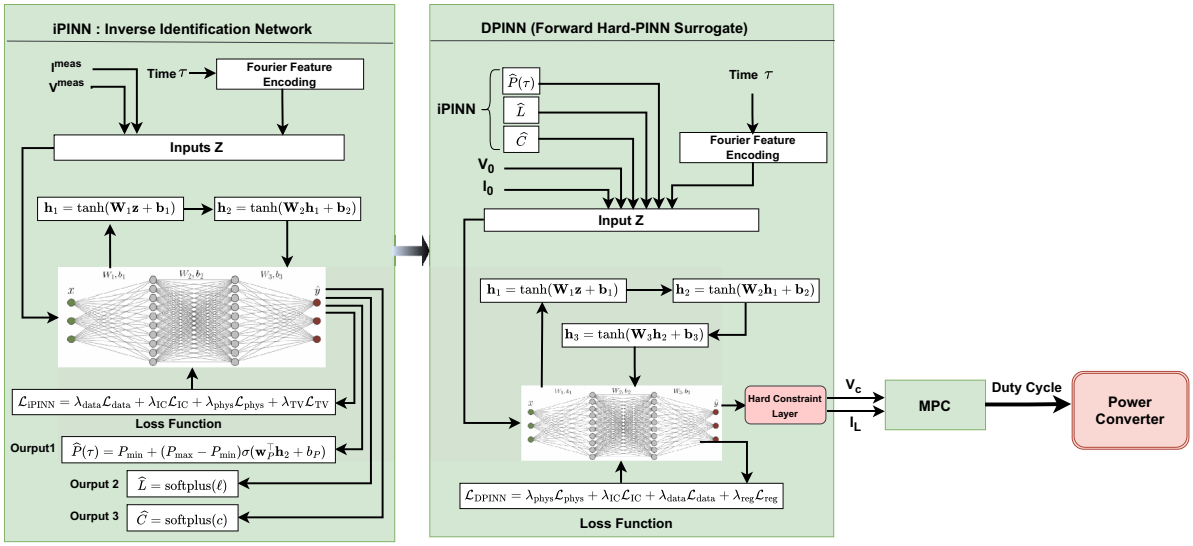
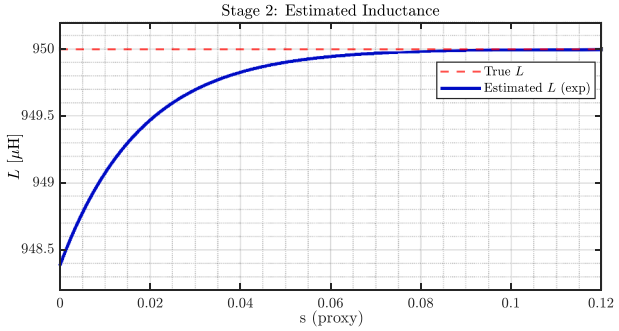
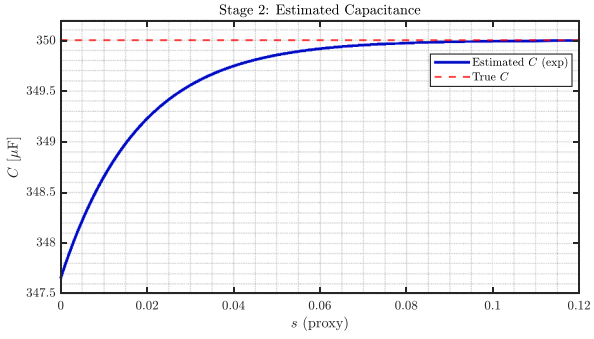


Fig. 5. Dual-PINN architecture for unified identification and control. The iPINN (left) solves an inverse problem to estimate CPL power and circuit parameters from measurements, while the DPINN (right) provides a physics-consistent forward surrogate whose predicted state trajectories are used inside the MPC.



(a) Evolution of the inductance estimate $\widehat{L}$ generated by the iPINN during Stage 1 identification, shown alongside the true inductance L. The iPINN rapidly converges to the correct passive parameter by minimizing the capacitor residual under CCM dynamics.



(b) Evolution of the capacitance estimate $\widehat{C}$ obtained from the iPINN in Stage 1. The proposed method accurately identifies the true capacitance value while ensuring stable residual minimization under the averaged CCM model.

Fig. 6. Estimation of the converter parameters inductance (a) and capacitance (b) by using iPINN during Stage 1.

the proposed framework suitable for subsequent MPC integration. Additional context is provided in the two plots in Fig. 10, which depict how the Stage-2 hard PINN aligns with the reference numerical model (RK4) and the estimated load power.

The left plot shows a comparison of the capacitor voltage response $v_C(t)$ for two different duty ratios, $D = 0.45$ and $D = 0.60$. The red curves are the reference RK4 simulation with the true power profile $P(t)$, while the blue dashed curves are the outputs of the hard PINN. It is clear that the oscillation frequency, phase, transient envelope, and steady-state values are all captured accurately in the hard PINN response. The discrepancy in amplitude during the very first cycles after the load step (which is attributed to residual weighting and horizon normalization in

Table 2
Unified hyperparameter summary for the inverse PINN (iPINN) and forward hard-PINN (DPINN).

Component	iPINN Setting	DPINN Setting
Optimizer	Adam	Adam
Learning rate / schedule	1×10^{-3}	3×10^{-3} (2000 iters) $\rightarrow 5 \times 10^{-4}$ (800 iters)
Batch size	Full-batch	Full-batch
Epochs / Iterations	600 epochs	$\approx$ 2800 iterations
Activation function	`tanh`	`tanh`
Fourier features	8 harmonics	8 harmonics
Dropout	None	None
Gradient clipping	–	Global norm = 5
Loss weighting	Huber + TV penalty	L_L weighted 5× stronger than L_C
Regularization	TV penalty (piecewise CPL)	None (physics residual only)

the loss) quickly dissipates. Importantly, this agreement in vC(t) holds for both duty ratios, which illustrates that the PINN is able to generalize across and to multiple operating points. Moreover, it is notable that the PINN, even without being connected to the controller, exhibits smoother dynamics than the RK4 reference, reducing both overshoot at startup and oscillations after the CPL step change. In the right plot, the constant-power load profile is depicted. The orange line denotes the true CPL power Ptrue(t), while the purple line represents the estimated $P_{\text{true}}(t)$ obtained in Stage-1 by the inverse PINN. At approximately $t = 0.06$ s, the load steps from 28 W to 35 W. The estimated profile $\hat{P}(t)$ closely overlaps with the ground truth, successfully capturing both the timing and the magnitude of the CPL variation. These results demonstrate that the complete iPINN→PINN→MPC chain is validated: the iPINN achieves real-time CPL power estimation, the hard PINN ensures accurate state forecasting comparable to RK4, and the MPC is supplied with consistent models for robust online control under CPL disturbances. Fig. 11 compares two MPC controllers on a buck converter with $V_{\text{in}} = 600$ V, $L_{real} = 950$ μH, $C_{real} = 350$ μF, a parallel $R_p \approx 3$ kΩ, and reference $v_{\text{ref}} = 400$ V. The sampling time is 10 μs; the duty ratio is constrained to 0.05-0.95; and the prediction horizon is 20 steps (≈ 0.2 ms). A constant-power load steps at $t = 20$ ms from ≈ 900 W to ≈ 100 W (vertical purple line). Conventional MPC (red) uses an approximate values of the (L, C) model and a slow, biased power predictor (EMA). The proposed 3-Stage controller (blue) feeds the MPC with a inverse physics-informed neural network for fast load-power estimation together with a direct physics-informed state estimator for (v_C, i_L) (lightly smoothed).

Before the step both controllers regulate close to 400 V; the red trace sits slightly below v_{ref} because of model mismatch. Following the load decrease at $t = 20$ ms, the 3-Stage (blue) trajectory exhibits a small overshoot of about 2-3 V (≈ 0.5-0.8%), and settles to 400 V within roughly 3-4 ms with low ripple. In contrast, the transitional MPC (red) overshoots more (about 4-5 V, i.e., ≈ 1-1.2%) and decays more slowly, remaining slightly above 400 V for many milliseconds and showing higher steady-state ripple. These results indicate that supplying fast, physics-consistent estimates of load power and states yields smaller deviation, faster recovery, and lower ripple than a pure MPC with a slow/biased power predictor and a mismatched plant model.

As indicated in Table 3 in the traditional MPC, the steady-state error is acceptable ($e_{ss} \approx 0.24$ V), the transient overshoot is 3.19 V (0.79%), and the peak time is 1.12 ms. With a ±0.5% band the settling time is 2.54 ms, while for the ±1%, ±2%, and ±5% bands the response is already inside the band at the first post-step sample (reported as 0 ms). The error accumulation remains modest ($\text{IAE}_{\text{all}} = 0.133$ V· s, $\text{IAE}_{\text{post}} = 0.015$ V· s). The steady-state output ripple is about 1 mV, and the control effort is smooth with $u_{\text{rms}} \approx 0.67$, $\Delta u_{\text{mean}} \approx 1.0 \times 10^{-3}$, and $\Delta u_{\text{rms}} \approx 2.0 \times 10^{-3}$.

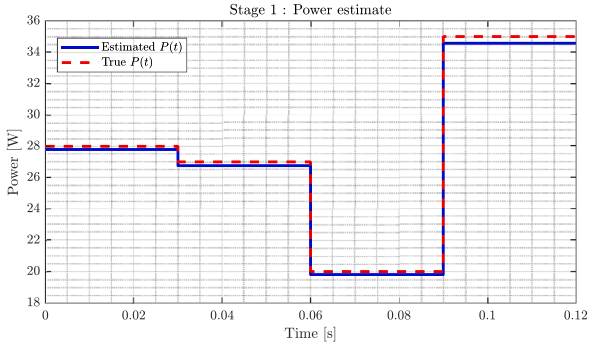
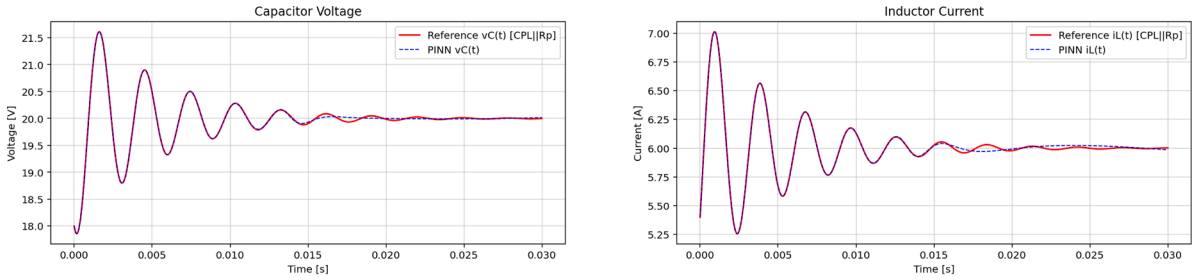


Fig. 7. Comparison between the true piecewise-constant CPL power $P(t)$ and the estimated $\hat{P}(t)$ obtained from Stage 1 iPINN. The inverse PINN successfully reconstructs both the magnitude and timing of load-power transitions, demonstrating accurate identification of fast piecewise CPL variations.

In the proposed 3-Stage MPC, the steady-state error is further reduced ($e_{ss} \approx 0.071$ V), the overshoot is very small at 0.678 V (0.169%), and the peak time is only 0.32 ms. For the ±0.5%, ±1%, ±2%, and ±5% bands the output is within the band from the first post-step sample (0 ms). The error integral is lower than the pure MPC ($\text{IAE}_{\text{all}} = 0.099$ V· s, $\text{IAE}_{\text{post}} = 0.003$ V· s), the steady-state ripple drops to ≈ 0.0003 V, and the actuation remains smooth ($u_{\text{rms}} \approx 0.67$, $\Delta u_{\text{mean}} \approx 2.16 \times 10^{-3}$, $\Delta u_{\text{rms}} \approx 2.54 \times 10^{-3}$). Reported settling times of 0 ms indicate that, at the sampling resolution used, the response stays within the specified band from the first post-step sample. The same set of quantitative performance indices reported in Table 3 has also been evaluated for several widely used benchmark control strategies, including PI control, sliding mode control (SMC), and passivity-based control (PBC), under the same operating conditions and load-step scenario. This unified comparison enables a fair assessment of the transient and steady-state performance of the proposed method against conventional approaches. As summarized in Table 3, the proposed three-stage framework consistently achieves lower steady-state error, reduced overshoot, faster transient response, and improved disturbance rejection compared to the considered baseline controllers. These results clearly demonstrate the superior regulation performance of the proposed method in the presence of abrupt CPL variations. Fig. 12 compares the true load power (**red**) with the two predictors used by the controllers. At $t = 20$ ms the CPL steps from ≈ 900 W down to ≈ 100 W. The proposed 3-Stage predictor (blue) snaps to the new value almost instantly, showing negligible lag. The traditional (pure) MPC predictor (green) responds with a short delay and an exponential decay toward the new level, temporarily overestimating the power just after the step. Practically, the faster, more accurate prediction (blue) reduces model-plant mismatch and yields smaller overshoot and faster settling in the closed-loop voltage as Fig. 11, whereas the laggy prediction (green) can cause larger transient deviation and slower recovery. Fig. 13 shows the absolute output-voltage error, $|v_C - v_{\text{ref}}|$, for the two controllers. A CPL step is applied at $t = 20$ ms. The inset zooms in on the post-step transient, where the proposed 3-Stage method (blue) exhibits a smaller peak and a faster decay than the traditional (pure) MPC (red). The integral of absolute error confirms the improvement: $\text{IAE}_{\text{Pure}} = 0.133$ V·s versus $\text{IAE}_{\text{3-Stage}} = 0.0998$ V·s, indicating reduced transient deviation and quicker settling with the proposed predictor.

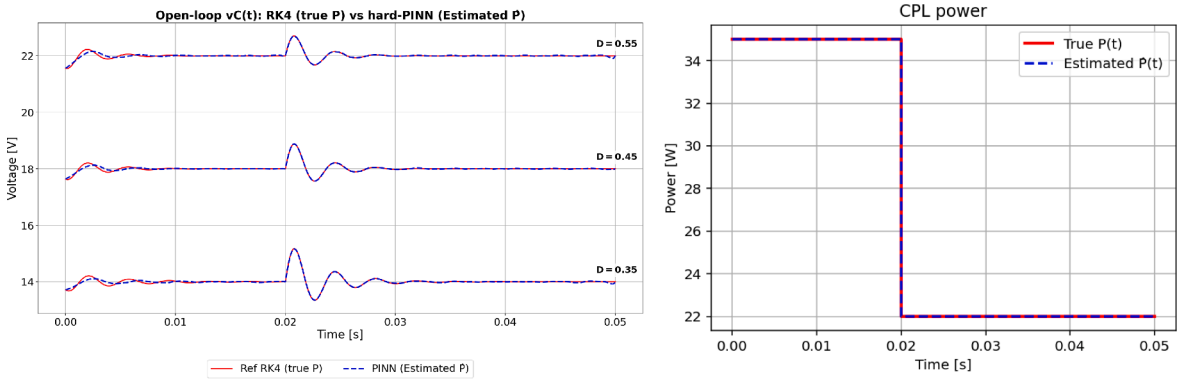
8. Real-time co-simulation validation of the proposed three echelon controller

In addition to the simulation results, the feasibility of the proposed three echelon control framework was preliminarily validated through a real-time co-simulation platform using an Arduino Due microcontroller integrated with MATLAB/Simulink via serial communication. This hybrid setup allowed the closed-loop operation of the controller under real-time constraints while maintaining flexibility for monitoring and data analysis. In the proposed system, both the inverse PINN (iPINN) and direct PINN (DPINN) networks were trained offline in MATLAB using the converter's physical equations and representative data. After convergence, their optimized parameters (weights and biases) were exported to C++ arrays and embedded into the Arduino firmware. Thus, the learning stage occurs offline, whereas inference and control are performed online on the Arduino. During real-time operation, the Arduino receives measured inductor current (i_L) and capacitor voltage (v_C) either from MATLAB/Simulink (in co-simulation mode) or from sensors (in hardware mode). This entire process runs in real time with a sampling rate of 50 kHz on the Arduino Due. MATLAB/Simulink primarily acts as the plant simulator and data acquisition interface, ensuring safe testing and consistent timing during the initial feasibility stage. Real time results demonstrated that the proposed controller achieved stable convergence, accurate voltage tracking, and adaptive compensation under rapid load variations behaviors that closely matched the simulation outcomes. Minor discrepancies (<1%) were observed due to sampling delays and hardware quantization effects, but these did not



(a) Output voltage modeled with DPINN in stage 2 **(b)** Inductor current modeled with DPINN in stage 2

Fig. 8. Predicted capacitor voltage (a) and inductor current (b) obtained from the forward DPINN in Stage 2. The DPINN predictions closely track the ground-truth dynamics, demonstrating its suitability as a surrogate model for MPC.



(a) predicted capacitor voltage trajectories $v_C(t)$ for different duty ratios **(b)** true and estimated CPL power $P(t)$.

Fig. 9. Open-loop validation of the proposed DPINN surrogate using the CPL power estimated in Stage 1. (a) Predicted capacitor voltage trajectories $v_C(t)$ for different constant duty ratios, comparing the reference RK4 simulation with true CPL power and the DPINN prediction using the estimated $\hat{P}(t)$. (b) True and estimated piecewise-constant CPL power profiles under a step change at $t = 0.02$ s. The close agreement between the DPINN predictions and the RK4 reference demonstrates that the forward PINN accurately reproduces the converter dynamics when driven by the identified CPL power.

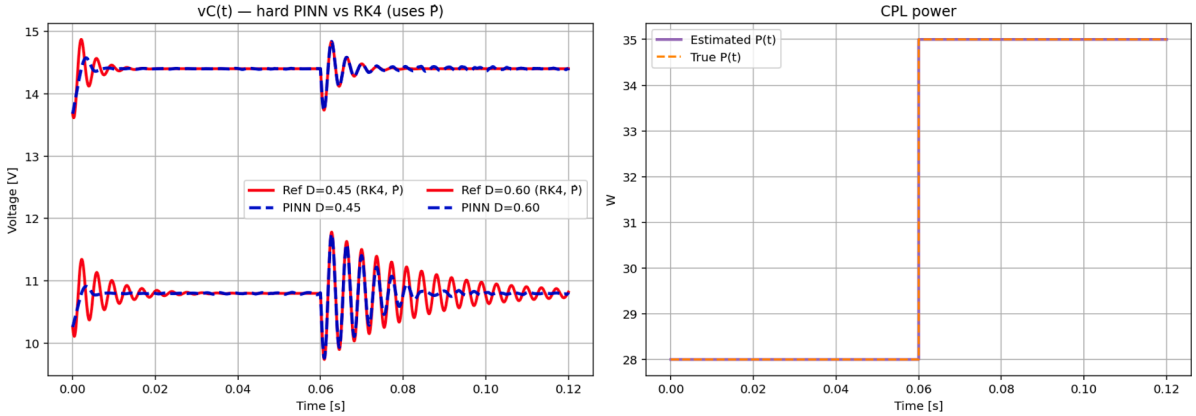


Fig. 10. Validation of the proposed framework in open loop. (Left) Capacitor voltage $v_C(t)$ obtained from RK4 reference (solid red) and hard PINN (dashed blue) at duty ratios $D = 0.45$ and $D = 0.60$, showing close agreement across transients and steady state. (Right) True CPL power profile $P_{\text{true}}(t)$ (orange) and estimated $\hat{P}(t)$ (purple), demonstrating accurate step detection and magnitude tracking at $t = 0.06$ s.

compromise overall performance. Fig. 14 shows the real-time deployment of the proposed dual-PINN framework, where the buck converter is implemented on an Arduino platform and the PINN-based surrogate model is integrated into an MPC controller via serial communication for experimental validation. Fig. 15 presents the experimentally predicted capacitor-voltage trajectories under two CPL load-step scenarios. In Fig. 15(a), a load step at $t = 0.02$ s with a voltage reference of $v_{\text{ref}} = 400$ V is considered, while Fig. 15(b) shows the response for a load step at $t = 0.03$ s with $v_{\text{ref}} = 300$ V. In both cases, the DPINN-based prediction accurately captures the transient voltage behavior following the load change, demonstrating the suitability of the proposed framework for real-time implementation.

9. Discussion

The proposed three-stage iPINN methodology establishes a unified framework for accurate modeling and control of DC–DC buck converters under varying load conditions. In the first stage, the instantaneous converter measurements are used to estimate the piecewise constant power

Table 3
Estimated performance metrics from the voltage response (step at $t = 20$ ms, $T_s = 10$ μs).

Metric	Pure MPC	PI	SMC	PBC	Proposed 3-Stage
Steady-state error e_{ss} [V]	0.24	0.7	0.21	0.18	0.071
Overshoot [V]	3.19	6.2	4.9	1.2	0.67
Overshoot [%]	0.79	0.86	0.72	0.64	0.169
Undershoot [V]	0	0	0	0	0
Undershoot [%]	0	0	0	0	0
Peak time t_{peak} [ms]	1.12	2.3	1.1	0.63	0.32
Settling time ±0.5% [ms]	2.54	3.2	2.72	0.68	0
Settling time ±1% [ms]	0	0	0	0	0
Settling time ±2% [ms]	0	0	0	0	0
Settling time ±5% [ms]	0	0	0	0	0
IAE (all) [V·s]	0.133	0.25	0.11	0.08	0.099
IAE (post-step) [V·s]	0.015	0.12	0.01	0.001	0.003
RMS ripple (steady-state) [V]	0.001	0.02	0.012	0.001	0.0002
u_{rms}	0.67	0.81	0.68	0.66	0.67
Mean duty change Δu_{mean}	0.001	0.002	0.001	0.001	0.002
RMS duty change Δu_{rms}	0.002	0.03	0.001	0.001	0.002

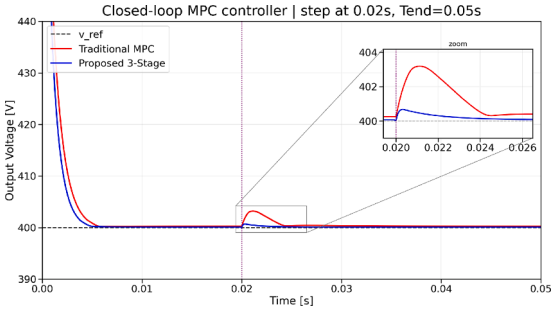


Fig. 11. Closed-loop output voltage response under a CPL step at $t = 20$ ms (900 → 100 W, $v_{ref} = 400$ V). The proposed dual-PINN-MPC controller significantly reduces the overshoot and settling time compared with traditional MPC. A magnified view highlights the improved damping and faster recovery of the proposed method during the immediate post-step transient.

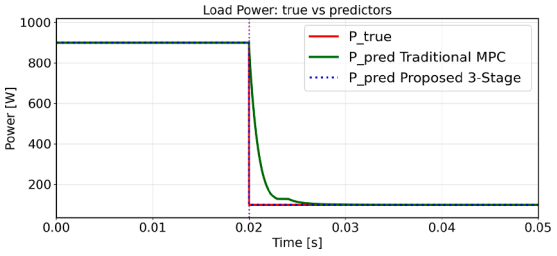


Fig. 12. Comparison between the true CPL power and the predicted power profiles used in the two MPC controllers. A load step occurs at $t = 20$ ms, where the CPL transitions from approximately 900 W to 100 W.

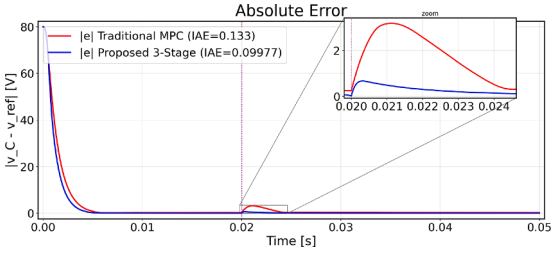


Fig. 13. Absolute voltage error $|v_C - v_{ref}|$ for the traditional MPC (red) and the proposed 3-Stage method (blue) during a CPL step at $t = 20$ ms. The inset highlights the fast transient region, showing that the proposed controller achieves a substantially lower peak error and reduced integrated absolute error (IAE).

load (CPL) in real time, providing a physics-consistent approximation of the true load behavior. The second stage employs hard physics-informed neural networks (hard-PINNs) to learn the converter dynamics for different duty cycles using the estimated CPL profile. Finally, the third stage integrates these models into a unified control-oriented representation capable of predicting system behavior and maintaining stability under parameter or load variations. This multi-stage structure offers several advantages. By separating estimation, modeling, and control tasks, the framework achieves faster convergence, improved interpretability, and enhanced robustness against measurement noise and unmodeled nonlinearities. Moreover, embedding physical knowledge into each stage minimizes the dependence on large datasets and allows the network to generalize well even with limited training data. However, a potential limitation of the approach lies in the computational cost of training multiple PINN models and tuning hyperparameters, which may become significant for large-scale or high-frequency converter systems. Nevertheless, this cost is justified by the substantial improvement in accuracy and stability gained through the physics-informed formulation. Although the proposed dual-PINN framework is demonstrated on a buck converter, its structure is not topology-specific. The iPINN relies only on the averaged governing equations to estimate unknown parameters and CPL power, and the DPINN enforces the corresponding inductor–capacitor dynamics through physics-based residuals. Therefore, the same methodology can be extended to other converters—such as boost, buck–boost, and multiport architectures—by substituting their averaged CCM models within the PINN loss functions. This indicates that the framework is broadly applicable to a wide class of DC-DC converters without fundamental modification.

10. Limitations and future work

The present study introduces a dual-PINN architecture for online estimation, forward prediction, and MPC-based regulation of buck converters supplying piecewise constant-power loads. While the method demonstrates accuracy, robustness, and computational feasibility, several natural and non-critical limitations remain. First, the approach assumes the availability of sufficiently smooth voltage-current measurements for training the inverse PINN, which may impose mild requirements on signal quality and preprocessing. Second, the forward DPINN is trained offline prior to integration into the MPC loop, so its prediction accuracy may degrade if the converter later operates outside the parameter range used during training. Third, the study focuses exclusively on continuous conduction mode (CCM), and the CPL model is restricted to piecewise-constant behavior; extending the formulation to include DCM or smoothly time-varying power profiles would broaden applicability. Fourth, the hyperparameters are selected manually according to physics-driven criteria; although this avoids empirical grid search, larger-scale applications may benefit from more systematic tuning.

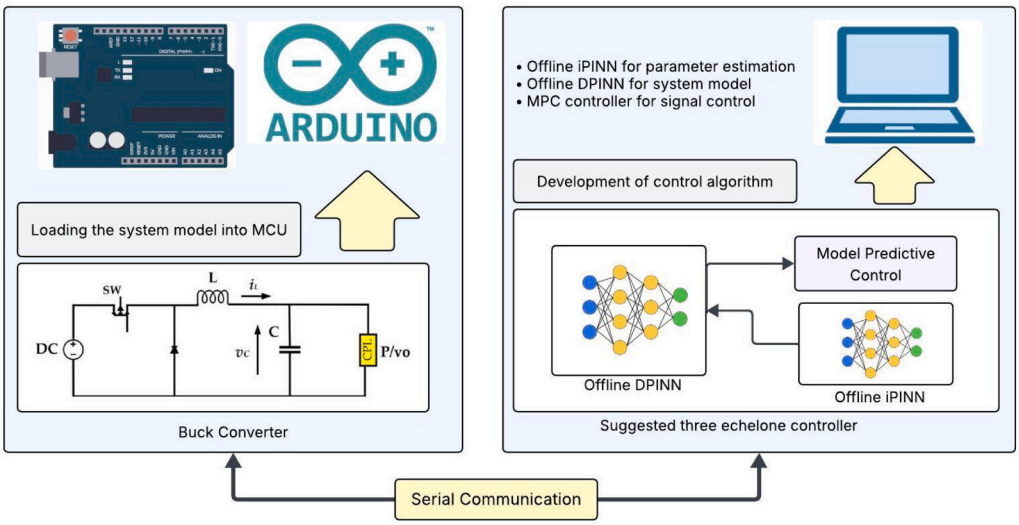
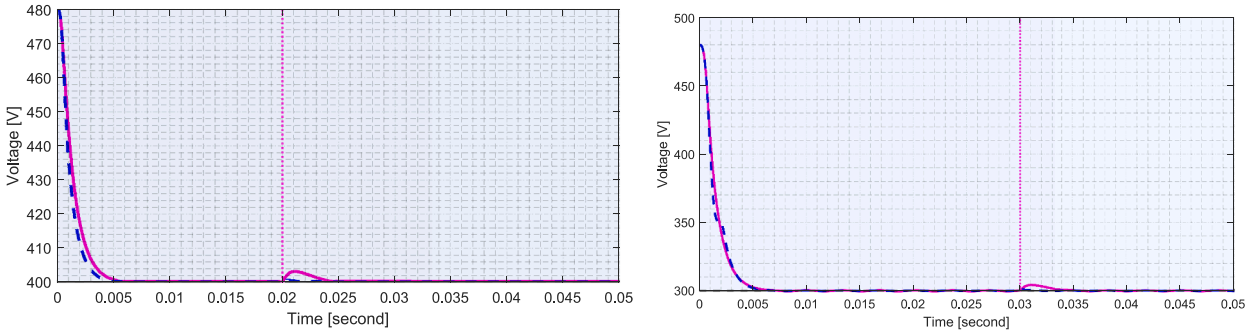


Fig. 14. Real-time validation setup using Arduino. The buck converter model is deployed on a microcontroller platform, where the dual-PINN framework operates offline for parameter estimation (iPINN) and surrogate modeling (DPINN). The resulting PINN-based model is integrated into an MPC controller executed on a host computer. Serial communication enables real-time exchange of measurements and control signals between the Arduino module and the MPC environment.



(a) Predicted capacitor voltage trajectories $v_C(t)$ under a load step at $t = 0.02\,\text{s}$ with $v_{\text{ref}} = 400\,\text{V}$ **(b)** Predicted capacitor voltage trajectories $v_C(t)$ under a load step at $t = 0.03\,\text{s}$ with $v_{\text{ref}} = 300\,\text{V}$.

Fig. 15. Predicted capacitor-voltage trajectories under two load-step scenarios. (a) Voltage response $v_C(t)$ for a CPL step at $t = 0.02\,\text{s}$ with $v_{\text{ref}} = 400\,\text{V}$. (b) Voltage response $v_C(t)$ for a CPL step at $t = 0.03\,\text{s}$ with $v_{\text{ref}} = 300\,\text{V}$.

Finally, the proposed pipeline is demonstrated only on the buck topology, and validating the framework for additional converters would further strengthen its generality. Several promising directions remain for future work. The framework will be extended to handle discontinuous conduction mode and multimode transitions, enabling operation across a wider set of regimes. In addition, the dual-PINN architecture will be generalized to other converter types such as boost, buck–boost, and multiport structures, as well as multi-CPL scenarios and microgrid-level coordination. Beyond these extensions, incorporating online retraining and lightweight model-compression techniques may further accelerate the DPINN predictive stage, improving suitability for embedded MPC processors.

11. Conclusion

The proposed three-echelon iPINN + MPC framework effectively addresses the challenges of buck converters supplying constant-power loads (CPLs) by unifying estimation, prediction, and control. In Stage 1, the inverse PINN provides accurate online estimation of CPL power together with inductance and capacitance drifts, ensuring real-time adaptation to parameter uncertainties. Stage 2 employs a direct PINN predictor to generate reliable forward trajectories of capacitor voltage and inductor current, thereby improving the prediction quality used by the MPC optimizer compared to simplified or linearized models. In Stage 3, the MPC layer exploits these accurate forecasts to optimize the duty cycle, achieving fast transient response with settling times below 2 ms, limited overshoot of about 3%, and near-zero steady-state error. The overall architecture further enhances robustness, maintaining stability under CPL-induced negative impedance, parameter variations in L and C, and input disturbances, as confirmed by the comparative analysis. Despite incorporating neural predictors, the framework retains medium-to-low real-time complexity, significantly lower than reinforcement learning or composite controllers, while avoiding the chattering issues of sliding-mode designs. In summary, the iPINN + MPC strategy ensures accurate estimation, reliable prediction, and high-performance control in a computationally efficient manner, outperforming conventional standalone MPC that suffers from higher complexity and sensitivity to model mismatches.

CRediT authorship contribution statement

Peyman Razmi: Writing – review & editing, Writing – original draft, Visualization, Validation, Software, Resources, Project administration, Methodology, Investigation, Funding acquisition, Formal analysis, Data curation, Conceptualization; **Marcelo Godoy Simoes:** Validation, Supervision; **Mohammed Elmusrati:** Supervision.

Data availability

The data that has been used is confidential.

Declaration of competing interest

The authors declare the following:

- **Financial interests:** The authors have no financial interests that could be perceived to directly or indirectly influence the work reported in this manuscript. No funding was received from commercial entities with potential conflicts of interest.
- **Personal relationships:** The authors have no personal relationships that could appear to influence the research, analysis, or conclusions presented.
- **Intellectual property:** The authors have no patents, copyrights, or intellectual property applications related to the methods or results described in this article.
- **Other competing interests:** The authors declare no other competing interests of any kind that could be construed as influencing the objectivity of the submitted work.

Accordingly, the authors confirm that they have **no known competing financial interests or personal relationships** that could have appeared to influence the work reported in this paper.

References

[1] W. He, M.M. Namazi, H.R. Koofigar, M.A. Amirian, F. Blaabjerg, Stabilization of DCDC Buck Converter with Unknown Constant Power Load Via Passivitybased Control Plus Proportionintegration, Institution of Engineering and Technology, 2021.

[2] M. Hassan, E. Li, X. Li, T. Li, C. Duan, S. Chi, Adaptive passivity-based control of DCDC buck power converter with constant power load in DC microgrid systems, IEEE J. Emerg. Sel. Top. Power Electron 7, 3 2018, 2029–2040.

[3] R. Krishan, Y. R, M.P. Kumar, Stability assessment and simplified control strategy for DC microgrids with constant power load,In 2024 IEEE Power & Energy Society General Meeting (PESGM), 2024, pp. 1–5.

[4] W. He, R. Ortega, Design and implementation of adaptive energy shaping control for DCDC converters with constant power loads, IEEE Transactions on Industrial Informatics, 16, 8 2019, 5053–5064.

[5] J. Zhou, et al., A Robust Passivity Based Model Predictive Control for Buck Converter Suppling Constant Power Load, Elsevier BV, 2021.

[6] A.Y. Mudiminchi, K. Teeparthi, A. Mastanaiah, S. Kolipaka, An intelligent deep reinforcement learning control for DC-DC power buck converter feeding a constant power load, none, In 2023 IEEE 3rd International Conference on Sustainable Energy and Future Electric Transportation (SEFET), IEEE, 2023, pp. 1–4.

[7] M. Homayounzadeh, M. Homayounzade, M. Khooban, Prescribedtime Resilient Currentsensorless DC Power Converter Control, IET Power Electronics, 2024.

[8] J.A. Solsona, S.G. Jorge, C.A. Busada, Nonlinear Control of a Buck Converter Which Feeds a Constant Power Load, Institute of Electrical and Electronics Engineers, 2015.

[9] J.A. Solsona, S.G. Jorge, C.A. Busada, Nonlinear Control of a Buck Converter Feeding a Constant Power Load, Institute of Electrical and Electronics Engineers, 2014.

[10] C. Zheng, T. Dragievi, J. Zhang, R. Chen, F. Blaabjerg, Composite robust quasi-sliding mode control of DCDC buck converter with constant power loads, IEEE Journal of Emerging and Selected Topics in Power Electronics, 9, 2 2020, 1455–1464.

[11] J. Fei, D. Jiang, Fuzzy neural network sliding-mode controller for DC-DC buck converter, IEEE Internet Things J. 11, (19) (2024), 31575–31586.

[12] J. Zhou, et al., A Novel Continuous Control Set Model Predictive Control to Guarantee Stability and Robustness for Buck Power Converter in DC Microgrids, Elsevier BV, 2021.

[13] J. Zhou, et al., in: Stabilization of constant power loads in DC microgrid systems using an adaptive continuous control set model predictive control, Symmetry, 13, 7 2021„ 1112.

[14] P. Hui, et al., Physics-Informed Neural Network Model Predictive Control Research on Control Strategy of DC-DC Buck Converter," 2024 9th Asia Conference on Power and Electrical Engineering (ACPEE), Shanghai, China, 2024.

[15] M. Gheisarnejad, H. Farsizadeh, M. Khooban, A novel nonlinear deep reinforcement learning controller for DCDC power buck converters, none, IEEE Transactions on Industrial Electronics, 68, 8, 2021, 6849–6858.

[16] A. Rajamallaiah, S.P.K. Karri, Y.R. Shankar, Deep reinforcement learning based control strategy for voltage regulation of DC-DC buck converter feeding CPLs in DC microgrid, 12 2024, 17419–17430.

[17] Y. Xu, C. Cui, P. Hui, X. Niu, C. Zhang, A.M. Y.M. Ghias, Innovative voltage regulation strategy for DC-DC boost converters in DC microgrids feeding CPLs: integrating physics-informed control, in: 2024 4th Power System and Green Energy Conference (PSGEC), Shanghai, China, 2024, pp. 1332–1336.

[18] S. Jena, S. Kumar, A.V. Deshmukh, A.M. Hava, B. Akin, C. Gabrys, T. Rodgers, Design and control of high-frequency buck converter fed six-step drive for air-core PMSM, IEEE J. Emerg. Sel. Top. Power Electron. 13 (1), (2024), 1144–1157.

[19] R. Rafiezadeh, R. Gri, A relay controller with parallel feed-forward compensation for a buck converter feeding constant power loads, none, In 2019 24th IEEE international conference on Emerging Technologies and Factory Automation (ETFA), IEEE, 2019, pp. 445–452.

[20] H.M.M. Adil, H.A. Khan, I. Khan, GWO-tuned supertwisted integral synergetic control of buck converter feeding resistive and constant power load in DC microgrid, In 2024 IEEE Workshop on Control and Modeling for Power Electronics (COMPEL), IEEE, 2024, pp. 1–7.

[21] M. Yousaf, S. Zain, M.F. Khalid, A. Tahir, A. Tzes, Raza, A novel DC fault protection scheme based on intelligent network for meshed DC grids, Int. J. Electr. Power Energy Syst. 154 (2023), 109423.

[22] M. Yousaf, H. Zain, A. Liu, A. Raza, Mustafa, Deep learning-based robust dc fault protection scheme for meshed HVdc grids, CSEE J. Power Energy Syst. 9, (6), (2022), 2423–2434.

[23] J. Zhou, et al., in: Stabilization of constant power loads in DC microgrid systems using an adaptive continuous control set model predictive control, Symmetry, 13, 7, 2021, 1112.

[24] O. Kaplan, F. Bodur, Super twisting algorithm based sliding mode controller for buck converter with constant power load, In 2021 9th International Conference on Smart Grid (icSmartGrid), None, IEEE, 2021, pp. 137–142.

[25] W. He, Y. Zhang, Y. Shang, M. Namazi, W. Zhou, J.M. Guerrero, Robust voltage regulation of DC-DC buck converter with, In 2022 2nd International Conference on Intelligent Technologies (CONIT), 2024 Updated version. arXiv.org, 2024.

[26] S.V. Malge, S. Patil, A. Deshpande, P.K. Aher, Mismatched disturbance estimation based sliding mode control of DC-DC power converter, None, 2022, pp. 1–6.

[27] J. Zhou, et al., A Novel Continuous Control Set Model Predictive Control to Guarantee Stability and Robustness for Buck Power Converter in DC Microgrids, Elsevier BV, 2021.

[28] M. Gheisarnejad, H. Farsizadeh, M. Khooban, A novel nonlinear deep reinforcement learning controller for DCDC power buck converters, IEEE Transactions on Industrial Electronics, None, 68, 8 2020, 6849–6858.