CEA Comité Español de Automática
XLVII JORNADAS DE AUTOMÁTICA CÓRDOBA 2026
UNIVERSIDADE DA CORUÑA

Jornadas de Automática

Control neuronal de un convertidor boost: modelado GRU y aprendizaje por refuerzo

Juan Enseñat[a,*], Pablo Serrano-Torres[a], Gerson Portilla[a], Carolina Albea Sánchez[a], Alexandre Seuret[a]

[a]*Dpto. de Ingeniería de Sistemas y Automática, Universidad de Sevilla, Camino de los Descubrimientos s/n, 41092 Sevilla, España.*

To cite this article: Enseñat, J., Serrano-Torres, P., Portilla, G., Albea, C., Seuret, A. 2026. Neural control of a boost converter: GRU modelling and reinforcement learning Jornadas de Automática, 47. https://doi.org/10.17979/ja-cea.2026.47.13709

Resumen

Este trabajo presenta una estrategia de control neuronal aplicada a un convertidor DC/DC tipo boost, combinando la identificación de la planta mediante una red recurrente GRU con el diseño de un controlador por aprendizaje por refuerzo. Se formula el problema de control considerando un lazo interno de corriente y un lazo externo de tensión. Una red GRU se entrena para reproducir la dinámica del convertidor a partir del ciclo de trabajo, la corriente del inductor, la tensión de salida y la corriente de carga, prediciendo incrementos en lugar de valores absolutos para mitigar la acumulación de error. Esta planta neuronal se emplea como entorno para entrenar un agente TD3, cuya acción es el ciclo de trabajo. El controlador resultante se integra en una estructura en cascada con un PI externo de tensión y se implementa en una plataforma experimental basada en módulos Imperix, validando su capacidad de seguimiento y rechazo de perturbaciones sobre un convertidor real.

Palabras clave: Aprendizaje por refuerzo, Identificación no lineal, Aprendizaje automático, Aprendizaje para control, Control basado en datos, Identificación para control, Validación de modelos

Neural control of a boost converter: GRU modelling and reinforcement learning

Abstract

This paper presents a neural control strategy applied to a DC/DC boost converter, combining plant identification using a recurrent GRU network with the design of a reinforcement learning controller. The control problem is formulated considering an inner current loop and an outer voltage loop. A GRU network is trained to reproduce the converter dynamics from the duty cycle, inductor current, output voltage and load current, predicting increments rather than absolute values to mitigate error accumulation. This neural plant is used as the training environment for a TD3 agent, whose action is the duty cycle. The resulting controller is integrated into a cascaded structure with an external PI voltage loop and implemented on an experimental platform based on Imperix modules, validating its tracking capability and disturbance rejection on a real converter.

Keywords: Reinforcement learning control, Nonlinear system identification, Machine Learning, Learning for control, Data-based control, Identification for control, Model Validation

1. Introducción

Los convertidores DC/DC son componentes fundamentales en numerosos sistemas de electrónica de potencia, presentes en aplicaciones de energías renovables, almacenamiento energético, accionamientos eléctricos y alimentación regulada. El convertidor boost permite elevar una tensión continua de entrada, aplicable a casos como adaptar la tensión generada por paneles fotovoltaicos a las necesidades de carga del sistema.

Tradicionalmente, el control de estos convertidores se ha abordado a partir de modelos matemáticos derivados de las leyes físicas del circuito, linealizados alrededor de un punto de operación para diseñar estrategias clásicas como reguladores PI (*Proportional–Integral*) o estructuras en cascada (Verma

*Autor para correspondencia: jensenat@us.es
Este trabajo fue financiado por las ayudas PID2023-150118OB-I00 y ATR2023-145067, financiadas por MICIU/AEI/10.13039/501100011033/.

Attribution-NonCommercial-ShareAlike 4.0 International (CC BY-NC-SA 4.0)

et al., 2023). También se encuentran técnicas más avanzadas, como el control predictivo (Niu et al., 2023) o el *Sliding Mode Control* (Guldemir, 2005), que presentan mayor robustez pero exigen un conocimiento suficiente de la dinámica del sistema. Sin embargo, en determinadas aplicaciones los parámetros del convertidor pueden no ser completamente conocidos o variar con las condiciones de operación, lo que motiva el diseño del controlador desde un enfoque basado en datos.

Dentro de este enfoque, las redes neuronales recurrentes resultan adecuadas para modelar sistemas dinámicos no lineales. Las redes LSTM (*Long Short-Term Memory*) Hochreiter and Schmidhuber (1997) han sido empleadas para modelar convertidores de potencia Za'ter (2022). Por su parte, las unidades GRU (*Gated Recurrent Unit*) ofrecen una estructura más compacta manteniendo la capacidad de modelar dependencias temporales (Cho et al., 2014), con propiedades de estabilidad demostradas Bonassi et al. (2021). Trabajos recientes señalan que ambas arquitecturas pueden integrarse en estrategias de control, aunque su coste computacional condiciona su aplicabilidad (Ławryńczuk and Zarzycki, 2025).

Por otra parte, el aprendizaje por refuerzo permite formular el diseño del controlador como un problema de interacción entre un agente y un entorno. Saha et al. (2024) aplican el algoritmo PPO (*Proximal Policy Optimization*) al control de un convertidor boost, mientras que Cheng et al. (2025) proponen una variante compensada del algoritmo TD3 (*Twin Delayed Deep Deterministic Policy Gradient*) para reducir las fluctuaciones del ciclo de trabajo. En particular, el algoritmo TD3 Fujimoto et al. (2018) ha demostrado ventajas frente a otros métodos actor-crítico en espacios de acción continuos, al incorporar dos redes críticas para mitigar la sobreestimación del valor y una actualización retardada del actor para estabilizar el entrenamiento. Su aplicación a convertidores DC/DC ha sido estudiada por Wang et al. (2024) para topologías MIMO y por Rajamallaiah et al. (2024) para la regulación de tensión en convertidores buck con cargas de potencia constante, confirmando su capacidad para generar señales de control suaves en escenarios con variaciones paramétricas.

El objetivo de este artículo es evaluar una metodología de identificación y control basada únicamente en datos del convertidor. La principal aportación radica en combinar un modelo de planta GRU entrenado en incrementos, que actúa como entorno de entrenamiento, con un controlador de corriente obtenido por aprendizaje por refuerzo (TD3) e integrado en una estructura en cascada, obteniendo la ley de control sin conocer explícitamente los parámetros internos del sistema y validando la metodología completa sobre una plataforma experimental real. El artículo se organiza como sigue: la Sección 2 formula el problema de control, las Secciones 3 y 4 describen la identificación GRU y el diseño del controlador por aprendizaje por refuerzo, y la Sección 5 presenta la validación experimental.

2. Descripción del sistema y formulación del problema

El sistema considerado es un convertidor DC/DC boost del cual se cuenta con un conjunto de datos de sus entradas y salidas características; estas son la tensión de entrada v_{in}, la corriente en la bobina i_L, la tensión de salida v_o, la corriente de salida i_o y el ciclo de trabajo d. Las unidades y los valores de los parámetros del sistema se recogen en la Tabla 1. Se considera que el convertidor se controla mediante el ciclo de trabajo d, es decir, la ley de control se implementa con un modulador tipo PWM (*Pulse Width Modulation*). Por tanto, la señal de control será la que gobierne los transistores que conmutan los modos de funcionamiento del convertidor, a través del ciclo de trabajo d.

Hipótesis 1. *Considera un conjunto de datos procedentes de un convertidor boost al cual no se tiene acceso. Considera que las señales $i_{L,k}$, $i_{o,k}$ y $v_{o,k}$ son accesibles y muestreadas cada instante de tiempo $k \in \{1, 2, ...\}$ y que la señal de control cumple la siguiente restricción:*

$$d_{min} \leq d_k \leq d_{max}. \tag{1}$$

El objetivo de este trabajo es minimizar el error de la corriente $i_{L,k}$ y de la tensión de salida $v_{o,k}$ respecto a una referencia variable dada $i_{L,k}^{ref}$ y $v_{o,k}^{ref}$, respectivamente.

El problema se formula como sigue:

Problema 1. *Considera un convertidor boost que satisface la Hipótesis 1. Se desea:*

1. *Identificar la planta del convertidor a partir de una muestra de datos de las variables $i_{L,k}$, $i_{o,k}$, $v_{o,k}$ y d_k de forma que se obtenga un comportamiento equivalente.*
2. *Establecer una señal de control d_k que minimice los errores*

$$e_{i,k} = i_{L,k} - i_{L,k}^{ref} \tag{2}$$

$$e_{v,k} = v_{o,k} - v_{o,k}^{ref}. \tag{3}$$

La metodología a seguir consiste en realizar una identificación de la planta mediante una red GRU, consiguiendo el objetivo del Problema 1.1. Posteriormente, se emplea dicha planta como entorno para entrenar una segunda red basada en aprendizaje por refuerzo que satisfaga los objetivos del Problema 1.2, obteniendo una ley de control $\pi_\theta : \mathbb{R}^{n_o} \to \mathbb{R}$, parametrizada por un vector de pesos $\theta \in \mathbb{R}^{n_\theta}$, que genera directamente el ciclo de trabajo $d_k = \pi_\theta(o_k)$.

3. Identificación de la planta mediante GRU

En esta sección se aborda la solución al Problema 1.1, identificando el convertidor boost mediante una red GRU. Se describe primero la formulación de la unidad GRU y su empleo como modelo de la planta, y a continuación el procedimiento de entrenamiento.

3.1. Formulación matemática de la unidad GRU e identificación neuronal de la planta

La unidad GRU, introducida en Cho et al. (2014), constituye una arquitectura recurrente con puertas cuyas propiedades de estabilidad han sido analizadas en Bonassi et al. (2021) y cuyo uso como modelo de procesos dinámicos en control ha sido estudiado en Ławryńczuk and Zarzycki (2025). En una GRU, la información temporal se almacena en un estado oculto $h_k \in \mathbb{R}^{n_h}$, que actúa como memoria interna de la red. Para cada instante k, la red recibe $x_k \in \mathbb{R}^{n_x}$ y actualiza dicho estado mediante la puerta de actualización $z_k \in \mathbb{R}^{n_h}$ y la puerta de reinicio $r_k \in \mathbb{R}^{n_h}$. La primera determina qué parte de la

información anterior debe conservarse, mientras que la segunda regula qué componentes del estado previo se emplean para construir la nueva representación temporal. Ambas puertas se calculan como:

$$z_k = \sigma(W_z x_k + U_z h_{k-1} + b_z), \tag{4}$$
$$r_k = \sigma(W_r x_k + U_r h_{k-1} + b_r), \tag{5}$$

donde $\sigma(\cdot)$ es la función sigmoide; $W_z, W_r \in \mathbb{R}^{n_h \times n_x}$, $U_z, U_r \in \mathbb{R}^{n_h \times n_h}$ y $b_z, b_r \in \mathbb{R}^{n_h}$ son los pesos y sesgos de cada puerta. A partir de estas variables se obtiene el estado candidato $\tilde{h}_k \in \mathbb{R}^{n_h}$, calculado mediante una combinación no lineal de la entrada actual y del estado oculto previo filtrado por r_k. El nuevo estado oculto se obtiene finalmente como una interpolación elemento a elemento entre h_{k-1} y $\tilde{h}_k$:

$$\tilde{h}_k = \tanh(W_h x_k + U_h(r_k \odot h_{k-1}) + b_h), \tag{6}$$
$$h_k = (1 - z_k) \odot h_{k-1} + z_k \odot \tilde{h}_k, \tag{7}$$

donde $\tanh(\cdot)$ es la tangente hiperbólica, $\odot$ el producto de Hadamard, y $W_h \in \mathbb{R}^{n_h \times n_x}$, $U_h \in \mathbb{R}^{n_h \times n_h}$ y $b_h \in \mathbb{R}^{n_h}$ son los parámetros del estado candidato. Una vez procesada la secuencia, el estado oculto de la última capa GRU se transforma en la salida mediante dos capas densas:

$$\hat{y}_k = W_2 \,\mathrm{ReLU}(W_1 h_k + b_1) + b_2, \tag{8}$$

con $\mathrm{ReLU}(\cdot) = \max(0, \cdot)$ y W_1, W_2, b_1 y b_2 los pesos y sesgos correspondientes.

A partir de esta formulación, la planta del convertidor se identifica mediante una red GRU entrenada con datos del propio convertidor. El objetivo no es sustituir de forma definitiva al modelo físico, sino disponer de un entorno dinámico representativo, diferenciable y rápido de evaluar para el posterior diseño del controlador por aprendizaje por refuerzo. El vector de entrada se define como

$$x_k = \begin{bmatrix} d_k & i_{L,k} & v_{o,k} & i_{o,k} \end{bmatrix}^T, \tag{9}$$

y, en lugar de predecir directamente los valores absolutos de corriente y tensión en el siguiente instante, la red estima sus incrementos:

$$y_k = \begin{bmatrix} \Delta i_{L,k} & \Delta v_{o,k} \end{bmatrix}^T, \tag{10}$$

donde $\Delta i_{L,k} = i_{L,k+1} - i_{L,k}$ y $\Delta v_{o,k} = v_{o,k+1} - v_{o,k}$. Esta formulación en incrementos reduce la acumulación de error durante la predicción autorregresiva, ya que la red aprende las variaciones locales de la dinámica y el estado predicho se obtiene sumando el incremento estimado al estado actual, en lugar de estimar directamente los valores absolutos.

La estructura empleada se representa en la Figura 1. El vector x_k se procesa a través de dos capas GRU apiladas, con una capa de *dropout* de probabilidad p entre ambas para reducir el sobreajuste, y una etapa de regresión final que produce Δi_L y Δv_o; estos incrementos se suman al estado actual y se emplean como entrada del siguiente paso.

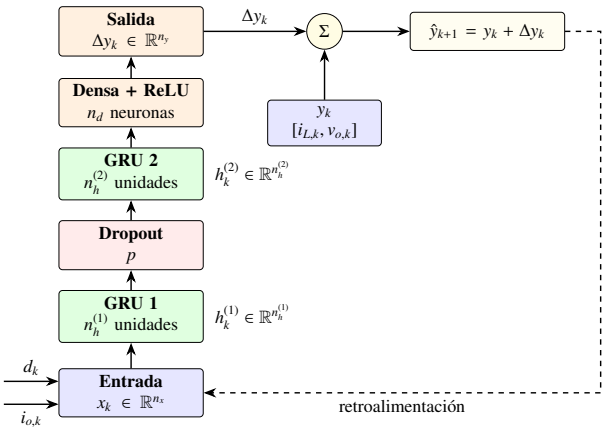

Figura 1: Estructura de la red GRU empleada para la identificación de la planta. Las señales d_k e $i_{o,k}$ se proporcionan externamente, mientras que $i_{L,k}$ y $v_{o,k}$ provienen de la retroalimentación autorregresiva.

3.2. *Entrenamiento*

Antes del entrenamiento, todas las variables se normalizan mediante una transformación z-score calculada únicamente sobre el conjunto de entrenamiento para evitar fuga de información desde validación. El conjunto de datos procede de trayectorias simuladas en bucle abierto sobre el modelo conmutado del convertidor, excitando el ciclo de trabajo con escalones aleatorios, rampas y componentes sinusoidales superpuestos, mientras que la carga varía de forma independiente mediante escalones y las condiciones iniciales se seleccionan aleatoriamente para cubrir el espacio de operación. La duración de cada trayectoria se denota por T_{traj} y el número de muestras útiles por trayectoria por N_T, de modo que el tamaño total del conjunto de datos viene determinado por el número de trayectorias y por N_T. La partición entrenamiento/validación se realiza por trayectorias. Tras construir pares entrada-salida consecutivos, las señales se organizan en ventanas deslizantes de longitud N_s, con un salto de ventana asociado al solape α, dando lugar a los subconjuntos de secuencias empleados para entrenamiento y validación. Los valores numéricos de estos parámetros, junto con el resto de hiperparámetros, se recogen en la Tabla 1.

El entrenamiento minimiza el error cuadrático medio entre los incrementos predichos $\hat{y}_k$ y los reales y_k mediante Adam, con reducción escalonada de la tasa de aprendizaje, regularización L2 y recorte de gradiente para estabilizar la optimización. Se emplea además parada anticipada cuando la pérdida de validación deja de mejorar durante un número prefijado de evaluaciones consecutivas, evitando el sobreajuste y seleccionando el modelo que ofrece mejor capacidad de generalización sobre trayectorias no utilizadas durante el ajuste de pesos.

4. Control mediante aprendizaje por refuerzo

En esta sección se aborda la solución al Problema 1.2, obteniendo una ley de control mediante aprendizaje por refuerzo. Se presenta primero el marco formal y el algoritmo TD3, y a continuación el diseño concreto del controlador.

4.1. *Marco del aprendizaje por refuerzo y algoritmo TD3*

El problema se formula como un proceso de decisión de Markov (MDP, *Markov Decision Process*), definido por

$(\mathcal{S}, \mathcal{A}, P, r, \gamma)$, donde $\mathcal{S} \subseteq \mathbb{R}^{n_o}$ es el espacio de estados, $\mathcal{A} \subseteq \mathbb{R}$ el espacio de acciones, P la transición del entorno, r la recompensa y $\gamma \in [0, 1)$ el factor de descuento; este marco formal se describe en Sutton and Barto (2018). En cada instante k, el agente observa $o_k \in \mathcal{S}$, aplica una acción $a_k \in \mathcal{A}$ mediante la ley π_θ, recibe r_k y el entorno evoluciona a o_{k+1}. En este trabajo, dicha evolución no se calcula durante el entrenamiento a partir del modelo conmutado, sino mediante la planta neuronal GRU de la Sección 3. El objetivo es maximizar el retorno acumulado descontado a lo largo de un episodio:

$$J(\theta) = \mathbb{E}_{\pi_\theta}\left[\sum_{k=0}^{K} \gamma^k \, r_k\right], \tag{11}$$

donde $K \in \mathbb{N}$ es el número de pasos del episodio.

El algoritmo TD3 Fujimoto et al. (2018) se selecciona por estar orientado a espacios de acción continuos, como el ciclo de trabajo del convertidor. Emplea un *actor*, que parametriza la ley determinista $\mu_\theta : \mathbb{R}^{n_o} \to \mathbb{R}$ y genera la acción de control d_k, y dos *críticos*, que estiman de forma independiente la función de valor acción $Q_\phi : \mathbb{R}^{n_o} \times \mathbb{R} \to \mathbb{R}$:

$$Q^{\pi_\theta}(o_k, a_k) = \mathbb{E}_{\pi_\theta}\left[\sum_{j=0}^{\infty} \gamma^j \, r_{k+j} \,\middle|\, o_k, a_k\right]. \tag{12}$$

A partir de las transiciones (o_k, a_k, r_k, o_{k+1}) almacenadas en un *buffer*, el valor objetivo se calcula como

$$y_k = r_k + \gamma \min_{i=1,2} Q_{\phi'_i}(o_{k+1},\, \mu_{\theta'}(o_{k+1}) + \epsilon)\,, \tag{13}$$

donde $Q_{\phi'_1}$, $Q_{\phi'_2}$ y $\mu_{\theta'}$ son las redes objetivo, y $\epsilon \sim \text{clip}(\mathcal{N}(0, \sigma), -c, c)$ es un ruido gaussiano recortado añadido a la acción objetivo. Esta perturbación suaviza la política evaluada por los críticos, mientras que el mínimo entre ambas estimaciones mitiga la sobreestimación del valor Fujimoto et al. (2018). Los críticos se actualizan minimizando el error cuadrático medio respecto a y_k, mientras que el actor se actualiza con menor frecuencia mediante

$$\nabla_\theta J(\theta) \approx \frac{1}{N} \sum_{k=1}^{N} \nabla_a Q_{\phi_1}(o_k, a)\Big|_{a=\mu_\theta(o_k)} \nabla_\theta \mu_\theta(o_k). \tag{14}$$

Finalmente, las redes objetivo se actualizan suavemente como $\phi'_i \leftarrow \tau\phi_i + (1-\tau)\phi'_i$ y $\theta' \leftarrow \tau\theta + (1-\tau)\theta'$, con $\tau \ll 1$, aportando estabilidad al aprendizaje Lillicrap et al. (2016).

4.2. Diseño del controlador

El controlador se plantea en cascada: un lazo interno rápido de corriente, obtenido mediante TD3, proporciona la dinámica necesaria para que un lazo externo más lento regule la tensión de salida. El entorno de entrenamiento es la planta neuronal GRU, lo que reduce el tiempo de entrenamiento y evita ejecutar la simulación conmutada durante la generación de episodios. De este modo, el agente aprende sobre un modelo representativo del convertidor y la política resultante puede integrarse posteriormente en la estructura experimental en cascada. El vector de observación utilizado por el agente se define como

$$o_k = \begin{bmatrix} e_{i,k} & e_{I,k} & i_{L,k} & v_{o,k} & i_{o,k} & d_{k-1} \end{bmatrix}^T, \tag{15}$$

donde $e_{i,k} = i_{L,\text{ref},k} - i_{L,k}$ es el error de corriente [A], $e_{I,k} = e_{I,k-1} + e_{i,k}\, T_s$ su integral discreta [A·s], $i_{L,k}$ la corriente del inductor [A], $v_{o,k}$ la tensión de salida [V], $i_{o,k}$ la corriente de carga [A] y d_{k-1} el ciclo de trabajo anterior [adim.]. La inclusión de d_{k-1} facilita penalizar variaciones bruscas del actuador, y cada componente se normaliza a un rango acotado. La acción es el ciclo de trabajo continuo saturado al rango operativo:

$$a_k = d_k \in [d_{\text{mín}},\, d_{\text{máx}}]. \tag{16}$$

Dado que el controlador obtenido por aprendizaje por refuerzo constituye el lazo interno rápido de corriente de la estructura en cascada, la regulación de tensión se delega en el lazo externo. Por coherencia con esta separación de dinámicas, la recompensa se diseña para minimizar el error de seguimiento de corriente, penalizar su integral y limitar variaciones bruscas del ciclo de trabajo:

$$r_k = r_{\text{base}} - \beta_e \, e_{i,k}^2 - \beta_I \left|\frac{e_{I,k}}{e_{I,\text{máx}}}\right| - \beta_{\Delta d}(\Delta d_k)^2 + r_{\text{bonus}}, \tag{17}$$

donde $r_{\text{base}} \in \mathbb{R}^+$ incentiva episodios largos, β_e, β_I y $\beta_{\Delta d} \in \mathbb{R}^+$ ponderan el error instantáneo, la integral normalizada por $e_{I,\text{máx}}$ y la variación $\Delta d_k = d_k - d_{k-1}$, y $r_{\text{bonus}} \in \mathbb{R}^+$ premia que $|e_{i,k}|$ permanezca dentro de una banda reducida. La penalización sobre Δd_k favorece acciones más suaves y reduce oscilaciones innecesarias del actuador. Además, el episodio termina anticipadamente si la corriente o la tensión superan límites de seguridad, asignando una penalización proporcional a los pasos restantes.

La Figura 2 esquematiza el entrenamiento. En cada paso, el actor calcula d_k a partir de o_k, la GRU predice Δi_L y Δv_o, se evalúa r_k y se construye o_{k+1}. Las transiciones se almacenan en el *buffer* de experiencias, desde el cual se muestrean mini-lotes para actualizar el actor y los dos críticos.

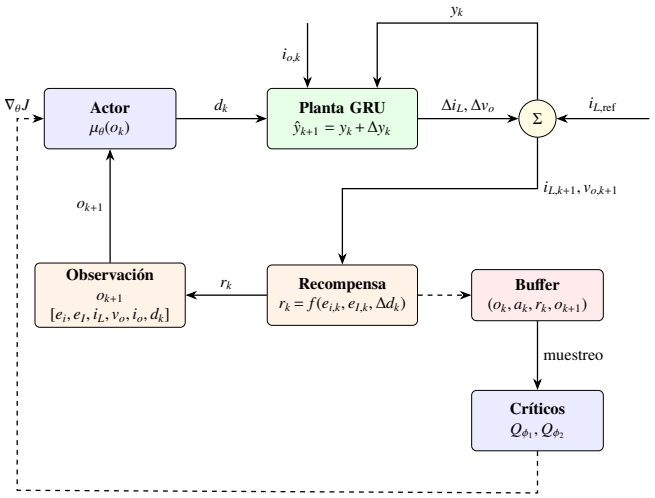

Figura 2: Esquema del entrenamiento del controlador por aprendizaje por refuerzo. El actor calcula el ciclo de trabajo, la planta GRU predice el siguiente estado, y las transiciones se almacenan para actualizar las redes mediante TD3.

El actor procesa la observación mediante dos capas ocultas de $n_a^{(1)}$ y $n_a^{(2)}$ neuronas con activación ReLU, y una salida tanh reescalada a $[d_{\text{mín}}, d_{\text{máx}}]$ para respetar las restricciones físicas del ciclo de trabajo. Cada crítico recibe la observación y la acción en dos ramas de n_c neuronas, fusionadas antes de una

capa oculta y una salida escalar Q. La ley de control resultante es

$$\hat{d}_k = W_3\,\mathrm{ReLU}(W_2\,\mathrm{ReLU}(W_1\bar{o}_k + b_1) + b_2) + b_3, \quad (18)$$

$$d_k = \tfrac{d_{\text{máx}}-d_{\text{mín}}}{2}\tanh(\hat{d}_k) + \tfrac{d_{\text{máx}}+d_{\text{mín}}}{2}, \quad (19)$$

donde $\bar{o}_k \in \mathbb{R}^{n_o}$ es la observación normalizada, $\hat{d}_k$ la salida preactivación de la red, y el escalado afín transforma la salida de $\tanh(\cdot)$ al rango admisible $[d_{\text{mín}}, d_{\text{máx}}]$. Al inicio de cada episodio se seleccionan aleatoriamente la referencia de corriente, la corriente de carga y las condiciones iniciales; durante el episodio se perturba aleatoriamente la carga para forzar una política robusta frente a variaciones de operación, y este finaliza al alcanzar el número máximo de pasos o al superar algún límite de seguridad.

5. Validaciones experimentales

5.1. Plataforma experimental

La validación experimental se realiza sobre un convertidor boost implementado en una plataforma basada en módulos Imperix. La Tabla 1 recoge los parámetros del convertidor, la configuración de la red GRU, los hiperparámetros de su entrenamiento, la configuración del agente TD3 y las constantes del controlador PI externo. Para la regulación de tensión de salida, el controlador de corriente obtenido por aprendizaje por refuerzo se integra como lazo interno en una estructura en cascada, donde el PI externo genera la referencia de corriente $i_{L,\text{ref}}$ a partir del error de tensión.

Tabla 1: Parámetros del convertidor, la red GRU, su entrenamiento, el agente TD3 y el PI.

Parámetro	Símbolo	Valor
Convertidor boost		
Tensión de entrada	V_{in}	50 V
Resistencia serie	R_L	0.3 Ω
Inductancia	L	2.2 mH
Capacidad de salida	C	10 μF
Frecuencia de conmutación	f_{sw}	20 kHz
Periodo de muestreo	T_s	50 μs
Rango de carga	R	30–150 Ω
Rango de ciclo de trabajo	d	0.1–0.9
Red GRU (planta neuronal)		
Variables de entrada	n_x	4
Unidades ocultas GRU 1	$n_h^{(1)}$	128
Probabilidad de dropout	p	0.1
Unidades ocultas GRU 2	$n_h^{(2)}$	64
Neuronas capa densa	n_d	64
Variables de salida	n_y	2
Entrenamiento red GRU		
Trayectorias del dataset	–	300
Duración por trayectoria	T_{traj}	0.5 s
Muestras útiles/tray.	N_T	10000
Muestras útiles totales	–	$3{,}0 \times 10^6$
Trayectorias entren./valid.	–	255/45
Partición entren./valid.	–	85/15 %
Pares entrada-salida/tray.	–	9999
Longitud de secuencia	N_s	64 (3.2 ms)
Salto de ventana	–	15 muestras
Solape de ventana	α	0.77
Secuencias entren./valid.	–	169065/29835
Épocas máximas	–	15
Tamaño de mini-lote	–	256
Tasa de aprendizaje inicial	–	10^{-3}
Factor reducción lr / 5 ép.	–	0.2
Regularización L2	–	10^{-4}
Recorte de gradiente	–	1
Paciencia (early stopping)	–	5
Agente TD3		
Observaciones	n_o	6
Neuronas ocultas actor	$n_a^{(1)}, n_a^{(2)}$	64, 64
Neuronas ocultas críticos	n_c	64
Controlador PI externo		
Constante proporcional	K_p	0.5
Constante integral	K_i	100

Las constantes K_p y K_i del PI externo se sintonizaron de forma empírica sobre el modelo conmutado Simulink/PLECS, imponiendo que el lazo de tensión fuese suficientemente más lento que el lazo interno de corriente para garantizar la separación de dinámicas propia de la estructura en cascada, y refinándose posteriormente en la plataforma experimental; no se empleó conocimiento explícito de los parámetros internos del convertidor en el diseño del lazo de corriente. Por otra parte, el control se ejecuta de forma sincronizada con el modulador PWM, muestreando en el punto medio del periodo de conmutación, de modo que se adquiere el valor medio de la corriente del inductor en lugar del rizado de alta frecuencia. Este muestreo sincronizado es habitual en el control digital de convertidores conmutados y evita el aliasing del rizado, por lo que la coincidencia entre el periodo de muestreo y el de conmutación no introduce problemas en la implementación.

5.2. Validación de la planta neuronal

La validación del modelo GRU se realiza comparando su salida autorregresiva con la del modelo de referencia ante señales de excitación no vistas durante el entrenamiento. Para ello, se generan 10 trayectorias de validación con escalones aleatorios de ciclo de trabajo d_k y perturbaciones de la carga conectada al convertidor, y se calculan las métricas de error recogidas en la Tabla 2.

Tabla 2: Métricas de validación de la planta GRU (10 trayectorias).

Variable	MAE	RMSE	R^2
Corriente i_L [A]	0,539 ± 0,074	0,806 ± 0,155	0,9759 ± 0,0232
Tensión v_{out} [V]	1,532 ± 0,457	2,348 ± 0,933	0,9977 ± 0,0008

5.3. Validación experimental en cascada

Tras el entrenamiento del agente TD3, la ley de control se valida en un modelo conmutado Simulink/PLECS, que incorpora efectos de conmutación no contemplados por el modelo promediado. Verificado su desempeño en simulación, los pesos del actor se exportan a la plataforma Imperix para su ejecución en tiempo real. Se realizan tres ensayos experimentales.

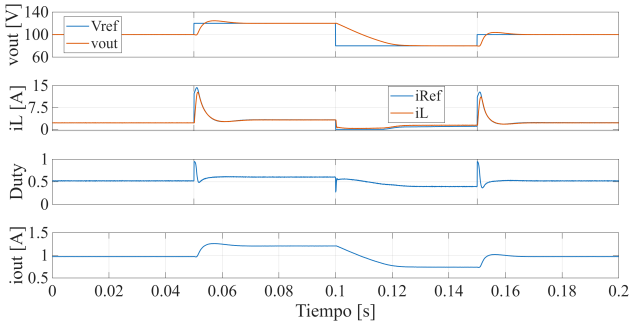

Figura 3: Ensayo 1: seguimiento de referencia de tensión variable con carga constante.

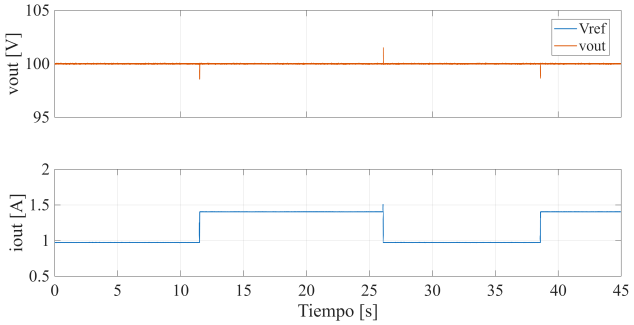

Figura 4: Ensayo 2: respuesta ante variaciones de la resistencia de carga con referencia de tensión constante.

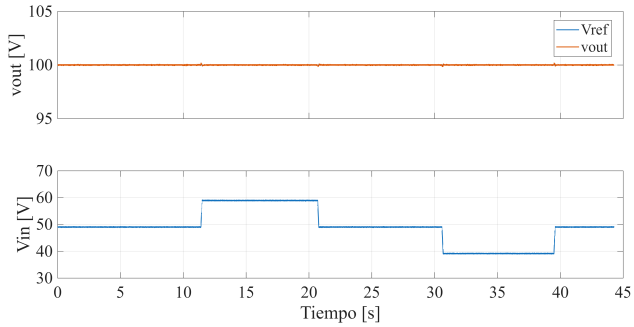

Figura 5: Ensayo 3: respuesta ante variaciones de la tensión de entrada con referencia de salida constante.

Tabla 3: Indicadores de desempeño en la validación experimental.

Ensayo	RMSE [V]	IAE [V]
Seguimiento de referencia	7,775	0,557
Variación de carga	0,032	0,721
Variación de tensión de entrada	0,022	0,721

La Figura 3 muestra el ensayo de seguimiento de referencia, donde se aplican cambios escalonados en la referencia de tensión. El controlador es capaz de seguir las transiciones entre los distintos niveles, aunque los cambios bruscos generan picos transitorios en la corriente del inductor y saturación temporal del ciclo de trabajo, lo que se refleja en el mayor valor de RMSE de los tres ensayos.Este error se asocia exclusivamente a los transitorios producidos por los cambios escalonados: durante estos, la corriente del inductor presenta picos y la acción de control alcanza temporalmente sus límites $[d_{\text{mín}}, d_{\text{máx}}]$, quedando saturada. Mientras el ciclo de trabajo permanece saturado, la acción está limitada por las restricciones físicas del actuador, lo que ralentiza el seguimiento hasta que el sistema vuelve al régimen permanente, donde el error se reduce de nuevo a valores pequeños. Las Figuras 4 y 5 corresponden a los ensayos de variación de carga ($\pm 10\ \Omega$) y de tensión de entrada (± 10 V respecto al valor nominal), manteniendo constante la referencia de tensión. En ambos casos la tensión de salida permanece prácticamente inalterada, con perturbaciones apenas perceptibles, lo que evidencia una buena capacidad de rechazo por parte de la estructura en cascada frente a perturbaciones acotadas. En conjunto, los indicadores de la Tabla 3 reflejan que el error en régimen permanente es reducido en los tres ensayos, siendo el transitorio de seguimiento de referencia el aspecto más exigente.

6. Conclusiones

Este trabajo ha presentado una evaluación de una metodología de control neuronal para un convertidor DC/DC boost. En primer lugar, la validación frente al modelo de referencia muestra que la red GRU reproduce adecuadamente la evolución de la corriente del inductor y de la tensión de salida, con valores elevados de R^2 en ambas variables, aunque los transitorios rápidos siguen siendo el aspecto más exigente del modelado.

Posteriormente, se ha entrenado un controlador de corriente mediante aprendizaje por refuerzo utilizando la planta neuronal como entorno. Los ensayos experimentales sobre la plataforma Imperix muestran que la estructura en cascada es capaz de seguir referencias variables de tensión y de rechazar perturbaciones tanto de carga como de tensión de entrada, con errores reducidos en régimen permanente. Estos resultados permiten validar la viabilidad de la metodología propuesta para el control de convertidores boost a partir exclusivamente de un modelo basado en datos.

Como líneas de trabajo futuro se plantea incorporar el error de tensión en la función de recompensa y explorar un esquema de un único agente que regule directamente la tensión de salida, evaluando si dicha formulación mejora el desempeño frente a la estructura en cascada actual. Asimismo, se contempla entrenar al agente empleando exclusivamente datos experimentales del convertidor y extender la metodología a otras topologías de convertidor DC/DC.

Referencias

Bonassi, F., Farina, M., Scattolini, R., 2021. On the stability properties of gated recurrent units neural networks. Systems & Control Letters 157, 105049.
DOI: 10.1016/j.sysconle.2021.105049

Cheng, H., Jung, S., Kim, Y.-B., 2025. A novel reinforcement learning controller for the dc-dc boost converter. Energy 321, 135479.
DOI: 10.1016/j.energy.2025.135479

Cho, K., van Merrienboer, B., Gulcehre, C., Bahdanau, D., Bougares, F., Schwenk, H., Bengio, Y., 2014. Learning phrase representations using rnn encoder-decoder for statistical machine translation. In: Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing. pp. 1724–1734.
DOI: 10.3115/v1/D14-1179

Fujimoto, S., Hoof, H. v., Meger, D., 2018. Addressing function approximation error in actor-critic methods. In: Proceedings of the 35th International Conference on Machine Learning. pp. 1587–1596.

Guldemir, H., 2005. Sliding mode control of dc-dc boost converter. Journal of Applied Sciences 5 (3), 588–592.

Hochreiter, S., Schmidhuber, J., 1997. Long short-term memory. Neural Computation 9 (8), 1735–1780.
DOI: 10.1162/neco.1997.9.8.1735

Ławryńczuk, M., Zarzycki, K., 2025. Lstm and gru type recurrent neural networks in model predictive control: A review. Neurocomputing 632, 129712.
DOI: 10.1016/j.neucom.2025.129712

Lillicrap, T. P., Hunt, J. J., Pritzel, A., Heess, N., Erez, T., Tassa, Y., Silver, D., Wierstra, D., 2016. Continuous control with deep reinforcement learning. arXiv preprint arXiv:1509.02971.

Niu, R., Zhang, H., Song, J., 2023. Model predictive control of dc–dc boost converter based on generalized proportional integral observer. Energies 16 (3), 1245.
DOI: 10.3390/en16031245

Rajamallaiah, A., Karri, S. P. K., Shankar, Y. R., 2024. Deep reinforcement learning based control strategy for voltage regulation of DC-DC buck converter feeding CPLs in DC microgrid. IEEE Access 12, 17419–17430.
DOI: 10.1109/ACCESS.2024.3358412

Saha, U., Jawad, A., Shahria, S., Rashid, A. H.-U., 2024. Proximal policy optimization-based reinforcement learning approach for dc-dc boost converter control: A comparative evaluation against traditional control techniques. Heliyon 10 (18), e37823.
DOI: 10.1016/j.heliyon.2024.e37823

Sutton, R. S., Barto, A. G., 2018. Reinforcement learning: An introduction.

Verma, P., Anwar, M. N., Ram, M. K., Iqbal, A., 2023. Internal model control scheme-based voltage and current mode control of dc–dc boost converter. IEEE Access 11, 110558–110569.
DOI: 10.1109/ACCESS.2023.3320272

Wang, Y., et al., 2024. TD3 algorithm based reinforcement learning control for multiple-input multiple-output DC–DC converters. IEEE Journal of Emerging and Selected Topics in Power Electronics.
DOI: 10.1109/JESTPE.2024.3564013

Za'ter, M. E., 2022. Modelling of a DC-DC buck converter using long-short-term-memory (LSTM). arXiv preprint arXiv:2211.03040.