



Experimental Modeling of a Buck Converter Using Genetic Algorithms in Python

Autores: **Juan Jose Berger**; Fernando Botterón; Roberto Esteban Carballo; Juan Gabriel Rojas; Gabriel Yonatan Aguirre.



1. Introducción.
 1. Relación de Compromiso al Modelar Sistemas.
 2. Algoritmos para Modelar Sistemas.
2. Metodología.
 1. Sistemas en Tiempo Discreto.
 2. Algoritmo Genético.
 3. CEP CC-CC reductor-Convertidor Buck.
3. Resultados Experimentales.
 1. Detalles del convertidor y uC.
 2. Implementación del GA.
 3. Resultados y Análisis.
4. Conclusión y trabajo futuro.
5. Referencias.

1.1 Introducción: Relación de Compromiso.



Fig. 1: Representación de un sistema o proceso de una entrada y una salida [1].

- Procedimiento para modelar:
1. Determinar Entrada-Salida.
 2. Simplificaciones.
 3. Ecuaciones diferenciales.
 4. Función de Transferencia.

Modelo Ideal

$$G_p(s) = \frac{V_s(s)}{d(s)} = \frac{V_e}{LCs^2 + \frac{L}{R}s + 1} \quad (5)$$

Modelo No Ideal

$$G_{pNI}(s) = \frac{12/5}{\left(\frac{1}{R} + \frac{Cs}{Cr_Cs + 1}\right)(r_L + Ls) + 1} \quad (8)$$

Modelar sistemas implica una relación de compromiso entre simplicidad y precisión del modelo [1],

1.1 Introducción: Algoritmos Para Modelar Sistemas.



XIV Jornadas de Investigación, Desarrollo Tecnológico,
Extensión, Vinculación y Muestra de la Producción

Modelado Automático de Sistemas Mediante Algoritmos Computacionales de Optimización.

Juan J. Berger ^{a,b,*}, Fernando E. Botterón ^{a,b}, Roberto E. Carballo ^{a,b}

^a Facultad de Ingeniería, Universidad Nacional de Misiones (UNaM), Oberá, Misiones, Argentina.

^b GIDE, FI-UNaM, Oberá, Misiones, Argentina.

e-mails: juanjoseberger1999@gmail.com, botteron@fio.unam.edu.ar, robertocarballo@fio.unam.edu.ar

Inteligencia Computacional

Base Matemática: Descenso del Gradiente

Tabu Search (TS) [2]

Genetic Algorithms (GA) [3]

Least Mean Squares (LMS) [5]

Neural Network (TS) [4]

“Modeling and identification of power electronic converters”. 2022 [13]

“Modelizado en caja negra de convertidores electrónicos de potencia para aplicaciones en microrredes de corriente alterna” 2019.[14]

2.1 Metodología: Sistemas en Tiempo Discreto.

$$G_P(z) = \frac{Y(z)}{U(z)} = \frac{a_{N_a} z^{N_a} + a_{N_a-1} z^{N_a-1} + \dots + a_1 z^1 + a_0}{z^{N_b} + b_{N_b-1} z^{N_b-1} + \dots + b_1 z^1 + b_0} \quad (1)$$

$$y(kT) = a_N u(kT) + \dots + a_0 u((k-N)T) - b_N y((k-1)T) - \dots - b_0 y((k-N)T) \quad (2)$$



$$e(kT) = y(kT) - \hat{y}(kT) \quad (3)$$

$$E = \frac{\sum_{k=0}^{N_m-1} [e(kT)]^2}{N_m - 1} \quad (4)$$

Fig. 1: Representación de un sistema o proceso de una entrada y una salida [1].

2.2 Metodología: Algoritmo Genético.

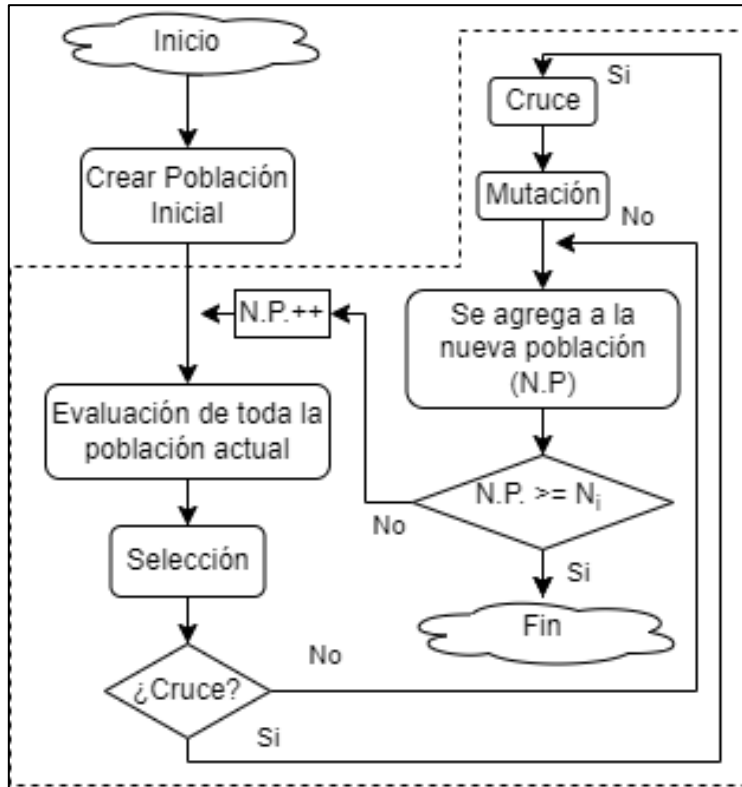


Fig. 1: Algoritmo de identificación.

Población: Conjunto de N_i individuos

Individuo: Arreglo de genes. $[a_1, b_1, b_2]$

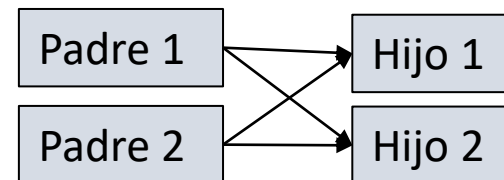
Población inicial: Generación aleatoria de N_i individuos cuyos genes están dentro de un rango $\pm \delta$ de su valor esperado

Evaluación: Se computa (4) para cada individuo

Selección: 3 Individuos aleatorios. El mejor será un padre.

Selección: 3 Individuos aleatorios. El mejor será un padre.

Cruce (γ)



Mutación (α)

Hijo

Genes

$\pm 1 \%$

La evolución se repite N_e veces.

2.3 Metodología: CEP CC-CC reductor-Convertidor Buck.

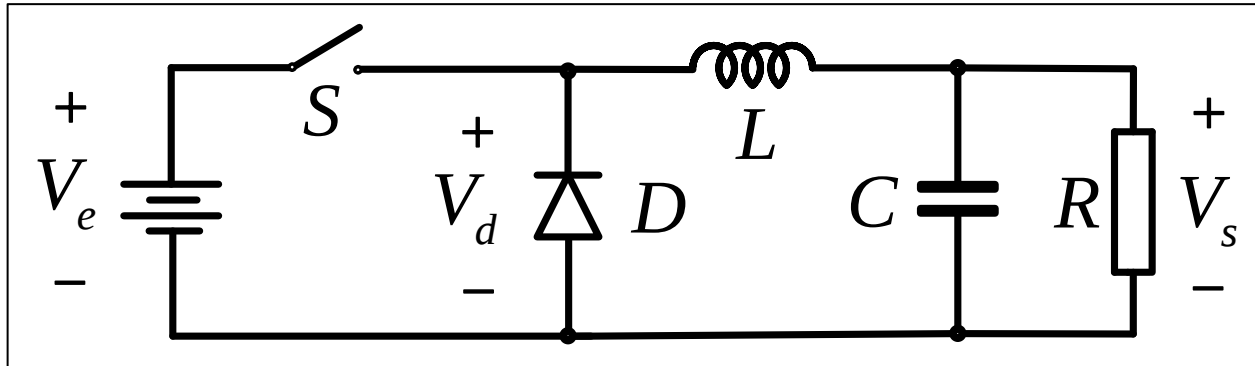


Fig. 2: Esquema simplificado del Convertidor Buck [17].

Modelo Ideal

$$G_p(s) = \frac{V_s(s)}{d(s)} = \frac{V_e}{LCs^2 + \frac{L}{R}s + 1} \quad (5)$$

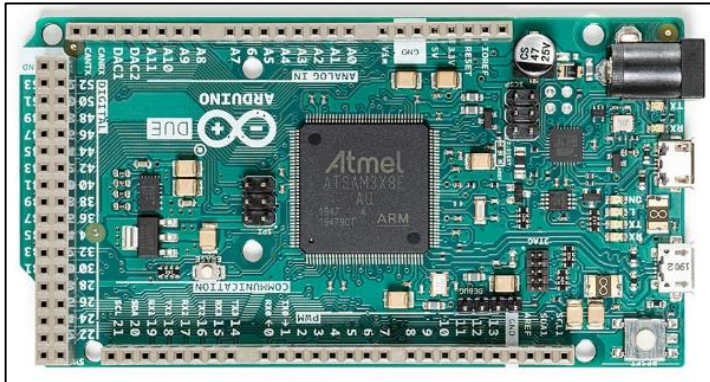
$$s = \frac{2}{T} \frac{1 - z^{-1}}{1 + z^{-1}} \quad (6)$$

$$G_p(z) = \frac{a_0 z^2 + a_1 z^1 + a_2}{z^2 + b_1 z^1 + b_2} \quad (7)$$

Modelo No Ideal

$$G_{pNI}(s) = \frac{12/5}{\left(\frac{1}{R} + \frac{Cs}{Cr_C s + 1} \right) (r_L + Ls) + 1} \quad (8)$$

3.1 Resultados Experimentales: Detalles del convertidor y uC.



uC: SAM3X8E; Placa de desarrollo: Arduino Due. Imagen adquirida de:
<https://store.arduino.cc/products/arduino-due>

- Reducción de 12 V a 5 V y una potencia 25 W [17].
- Transistor MOSFET IRF9540N que conmuta a una F_s de 40 kHz.
- Diodo MUR860.
- Inductor de 150 μ H,
- Capacitor 961 μ F.
- Carga de 2,2 Ω .

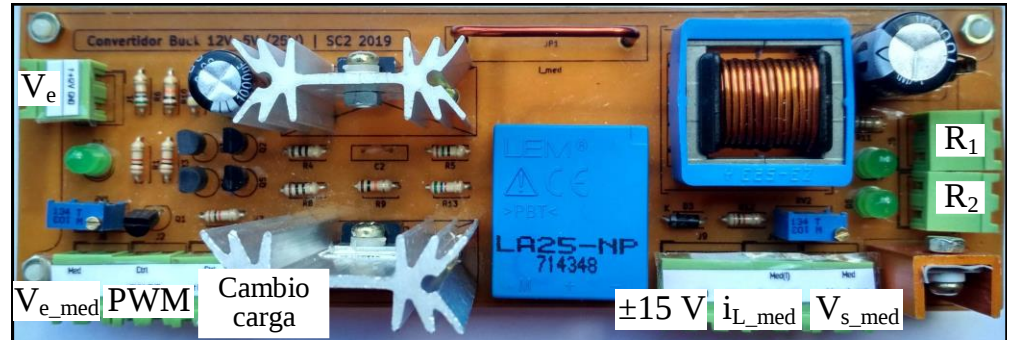


Fig. 4: Módulo didáctico de Convertidor Buck [17].

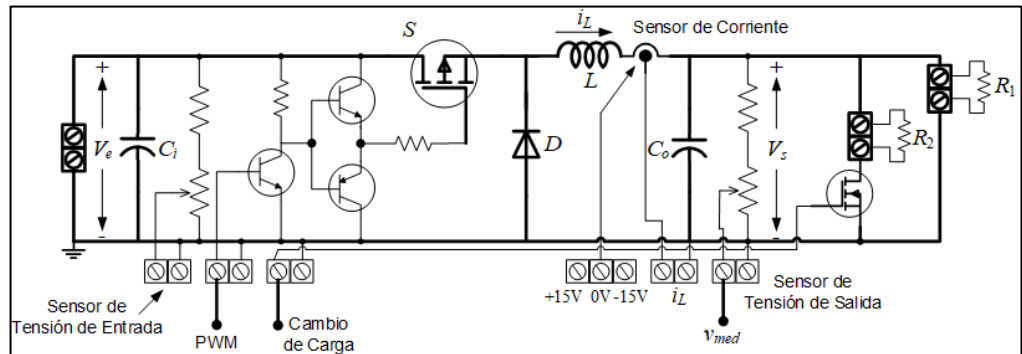


Fig. 3: Esquema de Convertidor Buck [17].

3.1 Resultados Experimentales: Implementación del GA.

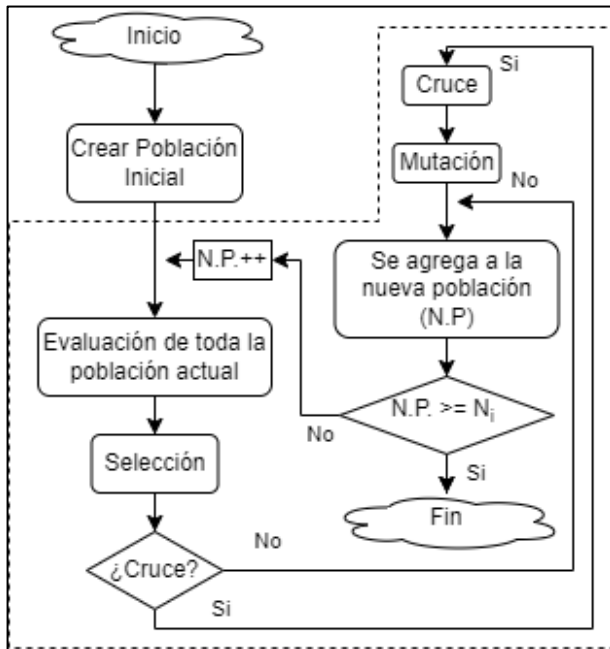


Fig. 1: Algoritmo de identificación.

Librerías:

- DEAP [18]
- Numpy [19]
- Pandas [20]
- Matplotlib [21].

Notebook de Google Colab [22]

Instalación e importación de librerías

Descarga de señales en formato csv.

Carga de información mínima. L, C, Vin y R

Implementación del GA mediante DEAP.

Modelado del sistema.

TABLA I: Hiperparámetros y otros parámetros.

Parámetro	Símbolo	Valor
Cantidad de Generaciones	N_e	100
Cantidad de Individuos	N_i	5000
Rango de variación	δ	100 %
Probabilidad de reproducción	γ	70%
Probabilidad de mutación	α	20%
Cantidad de muestras	N_m	100
Frecuencia de muestreo	F_m	4 kHz
Periodo de muestreo	T	25 μs

3.1 Resultados Experimentales: Resultados y Análisis.

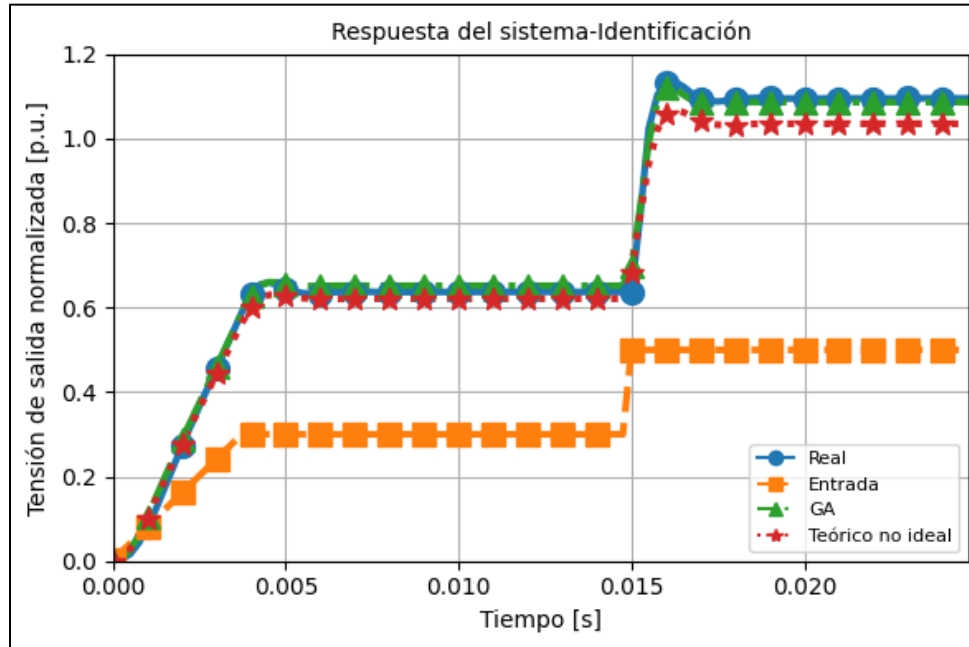


Fig. 5: Respuesta real del convertidor e identificación paramétrica.

Modelo GA

$$G_{p_GA}(z) = \frac{0,232z^2 + 0,464z^1 + 0,232}{z^2 - 0,932z^1 + 0,36} \quad (10)$$

$$E = 210,6 \times 10^{-6} \text{ p.u.}$$

GA logra una reducción del 87,1 % del ECM obtenido con el modelo no ideal

TABLA II: Resultados de la identificación.

Coeficiente	Valor teórico		Identificación AG
	Tustin Modelo Ideal	Valor	
$a_2 = a_0$	$\frac{V_e}{\frac{4LC}{T^2} + \frac{2L}{T} + 1}$	0,223	0,232
a_1	$2a_2$	0,446	0,464
b_1	$\frac{2 - \frac{8LC}{T^2}}{\frac{4LC}{T^2} + \frac{2L}{RT} + 1}$	-1,476	-0,932
b_2	$\frac{\frac{4LC}{T^2} - \frac{2L}{RT} + 1}{\frac{4LC}{T^2} + \frac{2L}{RT} + 1}$	0,898	0,36

Modelo No Ideal

$$G_{pNI}(z) = \frac{0.1305z^2 + 0.1305z + 3.654e-05}{z^2 - 1.124z + 0.4264} \quad (9)$$

$$E = 1633,12 \times 10^{-6} \text{ p.u.}$$

3.1 Resultados Experimentales: Resultados y Análisis.

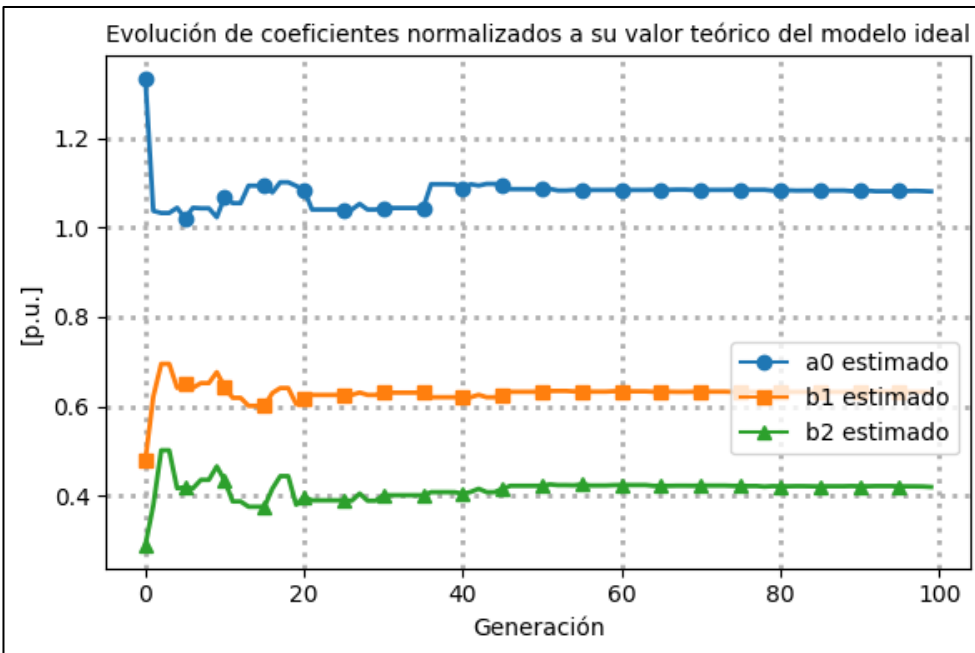


Fig. 6: Evolución de los coeficientes.

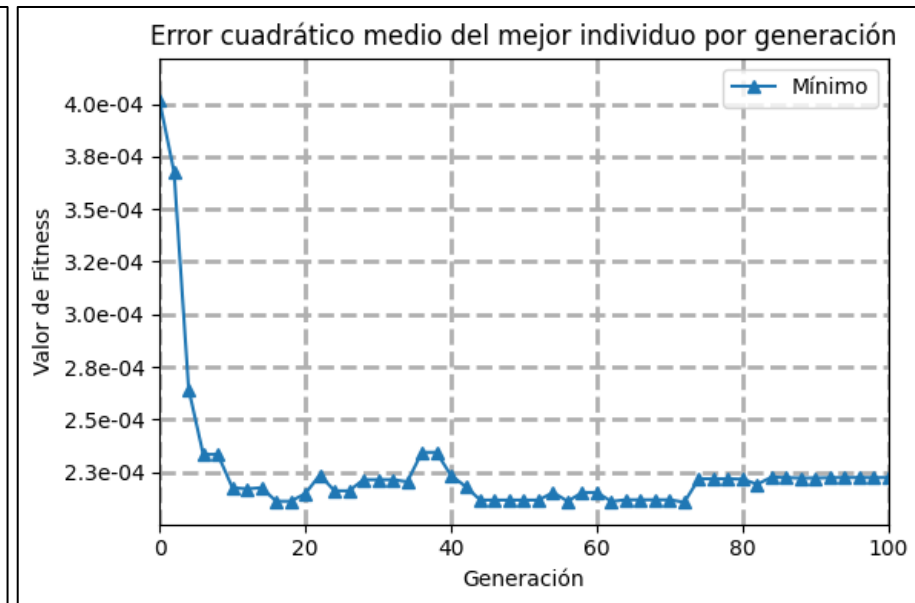


Fig. 7: Desempeño del mejor individuo en cada generación.

Tiempo de ejecución del GA

$$T_e = 115,37 \text{ s}$$

3.1 Resultados Experimentales: Resultados y Análisis.

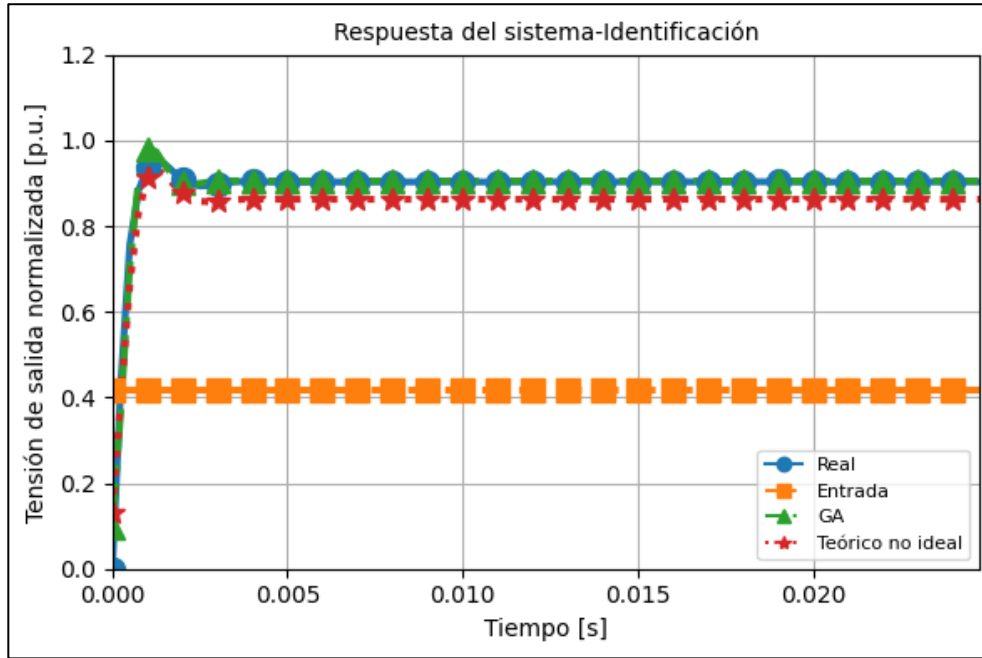


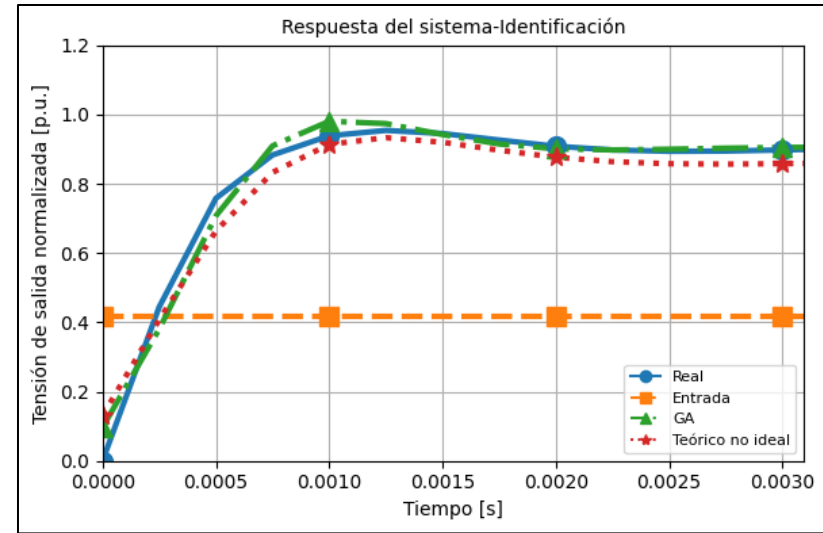
Fig. 8: Respuesta del sistema-Validación.

Modelo GA

$$G_{p_GA}(z) = \frac{0,232z^2 + 0,464z^1 + 0,232}{z^2 - 0,932z^1 + 0,36} \quad (10)$$

$$E = 195 \times 10^{-6} \text{ p.u.}$$

GA logra una reducción del 89,37 % del ECM obtenido con el modelo no ideal



Modelo No Ideal

$$G_{pNI}(z) = \frac{0.1305z^2 + 0.1305z + 3.654e-05}{z^2 - 1.124z + 0.4264} \quad (9)$$

$$E = 1838,39 \times 10^{-6} \text{ p.u.}$$

Conclusiones y Trabajo Futuro

Mediante este trabajo se ha:

- Obtenido y validado el modelo del tipo caja gris experimental en tiempo discreto de un CEP CC-CC Buck con un GA implementado en Python.
- A su vez, esto valida el diseño del GA posibilitando su uso para modelar otros CEP de forma experimental.
- Se ha innovado al aplicar y demostrar que los GA pueden modelar CEP reales a partir de señales experimentales de los mismos.
- Como trabajo futuro se propone la implementación del GA dentro de un DSP para evaluar la identificación **en línea** y, en paralelo, implementar técnicas de control al convertidor que aprovechen la sencillez y precisión del modelo.

- [1] K. Ogata, *Ingeniería de control moderna*. Pearson Educación, 2003.
- [2] F. Glover, “Tabu Search—Part I,” *ORSA Journal on Computing*, vol. 1, no. 3, pp. 190–206, 1989, doi: 10.1287/ijoc.1.3.190.
- [3] J. H. Holland, *Adaptation in Natural and Artificial Systems: An Introductory Analysis with Applications to Biology, Control, and Artificial Intelligence*. The MIT Press, 1992. doi: 10.7551/mitpress/1090.001.0001.
- [4] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, “Learning representations by back-propagating errors,” *Nature*, vol. 323, no. 6088, pp. 533–536, 1986, doi: 10.1038/323533a0.
- [5] B. Widrow and M. E. Hoff, “Adaptive switching circuits,” in *Neurocomputing, Volume 1: Foundations of Research*, J. A. Anderson and E. Rosenfeld, Eds., The MIT Press, 1988, p. 0. doi: 10.7551/mitpress/4943.003.0012.
- [6] Juan Jose Berger, Fernando Botterón, and Roberto Esteban Carballo, “Modelado Automático de Sistemas Mediante Algoritmos Computacionales de Optimización,” *Jornadas de Investigación y Desarrollo Tecnológico, Extensión, Vinculación y Muestra de la Producción - JIDeTEV*, 2024, [Online]. Available: <https://autoresjidetev.fio.unam.edu.ar/index.php/jidetev/article/view/121>
- [7] K. Kristinsson and G. A. Dumont, “Genetic algorithms in system identification,” in *Proceedings IEEE International Symposium on Intelligent Control 1988*, 1988, pp. 597–602. doi: 10.1109/isic.1988.65498.
- [8] Z. Zhang and F. Naghdy, “Application of genetic algorithms to system identification,” in *Proceedings of 1995 IEEE International Conference on Evolutionary Computation*, 1995, pp. 777–782 vol.2. doi: 10.1109/icec.1995.487484.
- [9] D. S. Pereira and J. O. P. Pinto, “Genetic algorithm based system identification and PID tuning for optimum adaptive control,” in *Proceedings, 2005 IEEE/ASME International Conference on Advanced Intelligent Mechatronics.*, 2005, pp. 801–806. doi: 10.1109/aim.2005.1511081.
- [10] M. Wakizono, T. Hatanaka, and K. Uosaki, “Time Varying System Identification with Immune Based Evolutionary Computation,” in *2006 SICE-ICASE International Joint Conference*, 2006, pp. 5608–5613. doi: 10.1109/sice.2006.315098.
- [11] J. Nowaková, M. Pokorný, P. Kömer, A. Abraham, and V. Snášel, “System Identification Using Genetic Algorithms,” Cham: Springer International Publishing, 2014, pp. 413–418.

- [12] M. Algreer, M. Armstrong, and D. Giaouris, “Active Online System Identification of Switch Mode DC–DC Power Converter Based on Efficient Recursive DCD-IIR Adaptive Filter,” *IEEE Trans Power Electron*, vol. 27, no. 11, pp. 4425–4435, 2012, doi: 10.1109/tpe.2012.2190754.
- [13] G. Rojas Dueñas, “Modeling and identification of power electronic converters,” 2022, doi: 10.5821/dissertation-2117-363405.
- [14] G. F. Guarderas Burbano, “Modelizado en caja negra de convertidores electrónicos de potencia para aplicaciones en microrredes de corriente alterna,” 2019.
- [15] V. Balakrishnan, “System identification: theory for the user (second edition): Lennart Ljung; Prentice-Hall, Englewood Cliffs, NJ, 1999, ISBN 0-13-656695-2,” *Automatica*, vol. 38, no. 2, pp. 375–378, 2002, doi: [https://doi.org/10.1016/S0005-1098\(01\)00214-X](https://doi.org/10.1016/S0005-1098(01)00214-X).
- [16] P. S. Foundation, “Python Language Reference Manual Version 3.12,” 2001, [Online]. Available: <https://www.python.org/>
- [17] Martín A. Berger, Brian Flech, Cesar E. Rebak, Rubén O. Núñez, Gabriel Y. Aguirre, and Fernando Botterón, “Diseño de módulo didáctico de un convertidor reductor (Buck) para la enseñanza de los sistemas de control y la electrónica de potencia,” *JIDeTEV*, 2023, [Online]. Available: https://drive.google.com/file/d/1i38myen4o-9s9eLIfpnf6j-h-qIQipk7/view?usp=drive_link
- [18] Félix-Antoine Fortin, Francois-Michel De Rainville, Marc-André Gardner, Marc Parizeau, and Christian Gagné, “DEAP: Evolutionary Algorithms Made Easy,” *Journal of Machine Learning Research*, vol. 13, pp. 2171–2175, 2012, [Online]. Available: <http://jmlr.org/papers/v13/fortin12a.html>
- [19] C. R. Harris *et al.*, “Array programming with NumPy,” *Nature*, vol. 585, no. 7825, pp. 357–362, 2020, doi: 10.1038/s41586-020-2649-2.
- [20] W. McKinney, “Data Structures for Statistical Computing in Python,” *Proceedings of the 9th Python in Science Conference*, 2010.
- [21] J. D. Hunter, “Matplotlib: A 2D Graphics Environment,” *Comput Sci Eng*, vol. 9, no. 3, pp. 90–95, 2007, doi: 10.1109/mcse.2007.55.
- [22] B. J. José, “Modelado_Buck_AG_Berger_Juan,” 2025, Accessed: May 26, 2025. [Online]. Available: https://colab.research.google.com/drive/1hbK3Ad2_-4GuI0ahWM3PaOw5IXJ_X436#scrollTo=Yfit0sVZXWG9
- [23] D. Gutiérrez, A. Tapia Córdoba, and Á. Rodríguez del Nozal, *Algoritmos Genéticos con Python: Un enfoque práctico para resolver problemas de ingeniería*. 2020.



I M A M



Gracias por la atención
