Sürdürülebilir Mühendislik Uygulamaları veTeknolojik Gelişmeler Dergisi
Journal of Sustainable Engineering Applications and Technological Developments
2026, 9(1): 213-231

Real-Time Chaos on Embedded Hardware: A Critical Review of Implementation, Observation and Control

Gömülü Donanımda Gerçek Zamanlı Kaos: Gerçekleme, Gözlem ve Kontrol Üzerine Eleştirel Bir Derleme

Berk DEMİRSOY[1], Mert Süleyman DEMİRSOY[2*], Yusuf HAMİDA EL NASER[2]

[1] Zonguldak Bulent Ecevit University, Department of Electronics and Automation, Alaplı Campus, Zonguldak, Turkiye
[2] Sakarya University of Applied Sciences, Department of Mechatronics Engineering, Esentepe Campus, Sakarya, Turkiye

ABSTRACT

Generating chaotic flows on digital hardware is a broader research problem than simply moving a differential equation solver onto an embedded processor. When a state variable is observed on a screen, the visible trace is shaped jointly by the chosen numerical method, finite precision, the sampling rate, analog conversion and, if present, feedback. The aim of this review is to evaluate, within a common critical framework, experimental studies on implementing continuous-time chaotic models on digital signal processors (DSPs) and related embedded hardware, on observing them with an oscilloscope through a digital-to-analog converter (DAC), and on changing the control gain during operation. The initial literature pool was complemented by a documented search of the Directory of Open Access Journals (135 records, 118 screened, 79 eligible), and the implementation parameters of representative studies were extracted into a common table. Integer-order and fractional-order flows, microcontroller and field-programmable gate array (FPGA) examples, and different chaos control objectives are treated separately. The reviewed studies show that numerical attractors can be produced as analog signals and that, in some set-ups, stabilization or synchronization is achieved in hardware. However, the integration step, the physical update period and the number format are seldom reported together, validation is mostly visual, and only a few studies close the loop around a physical analog chaotic circuit. In addition, an attractor on the oscilloscope does not, on its own, prove that the state update is completed on time at every sample, or that a feedback gain changed by an operator remains stabilizing during the transition; this question should be framed with the existing theory of switched, slowly varying and gain-scheduled systems. Apparent methodological differences between studies cannot be treated as direct performance contradictions when the same model, integration step and validation criterion are not used. The main conclusion of the review is that hardware studies of chaos provide different levels of evidence about numerical dynamics, timing, analog observation and control behavior, and that these can only be compared once they are clearly separated.

Keywords: Chaotic differential equations, digital signal processor, real-time computation, digital-to-analog converter, chaos control

Submit: 30.09.2026 Accept: 16.10.2026
Doi: 10.51764/smutgd.2051390

[2]*Corresponding Author: Sakarya University of Applied Sciences, Department of Mechatronics Engineering, Esentepe Campus, Sakarya, Turkiye E-mail: mertdemirsoy@subu.edu.tr; ORCID: 0000-0002-7905-2254

1. INTRODUCTION

Lorenz's deterministic nonperiodic flow is one of the landmarks that demonstrated sensitivity to initial conditions in the context of a continuous-time differential equation (Lorenz, 1963). Rössler showed that complex behavior can appear in a simpler family of equations, while studies of Chua's circuit brought the mathematical description of the attractor and its electronic observation closer together (Rössler, 1976; Matsumoto, 1984; Chua et al., 1986). Sprott's simple flows and the Chen–Ueta system provide different nonlinear terms and computational loads that can be tested on hardware (Sprott, 1994; Chen & Ueta, 1999).

In the chaos control literature, the stabilization of an unstable periodic orbit by small perturbations by Ott, Grebogi and Yorke, Pyragas's delayed feedback and Pecora–Carroll synchronization are different answers to the same problem (Ott et al., 1990; Pecora & Carroll, 1990; Pyragas, 1992). When these control ideas are moved to a digital set-up, sampling, computational delay and analog input–output are added to them.

The models and solution techniques used by Wang et al. (2015) and by Shi et al. (2026), who implemented chaotic flows on digital signal processors (DSPs), differ from each other. Köse and Mühürcü (2017), who built a Lorenz oscillator on a PIC18F452, and Bulut et al. (2026), who obtained digital-to-analog converter (DAC) output from an STM32F429, make the path from the state to the analog voltage visible, even though the processor class is different. In contrast, Yang et al. (2020) add a stabilization experiment to attractor generation on a DSP, and Rivera-Blas et al. (2019) add active synchronization control to the solution of a chaotic model on a microcontroller. This difference is especially important when interpreting a gain that can be changed during operation: an attractor whose parameter can be changed, a controller working with a fixed gain and a feedback coefficient changed online by an operator are three different types of experiment.

The motivating use case of this review is a laboratory or teaching set-up in which a DSP integrates a chaotic flow in real time, displays two states on an oscilloscope through a DAC and allows an operator to change a feedback gain with an external potentiometer read by an analog-to-digital converter (ADC). The use case is narrow, but it combines numerical integration, real-time execution, analog observation and a time-varying control gain in a single experiment; it is therefore used here as a test case for the evaluation framework, not as the only relevant application.

This paper focuses on the conversion of a continuous-time ordinary differential equation (ODE) into a sampled digital solution, the experimental findings obtained on different hardware, the evidential value of DAC–oscilloscope measurements and the open questions of live gain changes in chaos control. Studies on discrete-time chaotic maps and pseudo-random numbers are cited only when they are needed to explain finite-precision or platform limits. The aim is neither to present new experimental results nor to rank all studies in a quantitative order of superiority; it is to critically synthesize the connected and diverging findings within the scope shown by the original studies.

Existing books and reviews already cover parts of this ground. Treatments of electronic and FPGA realization describe how chaotic oscillators are designed, described in hardware description languages and sized in fixed-point arithmetic (Tlelo-Cuautle et al., 2016), and a separate line of work examines the dynamical degradation of digital chaotic systems: short cycles and degraded statistics of finite-precision maps (Li et al., 2005), the multiple pseudo-orbits produced by discretization schemes and roundoff (Nepomuceno & Mendes, 2017), and the dependence of chaos preservation on the integrator and the embedded platform (Méndez-Ramírez et al., 2019). These sources address how to realize an oscillator or how arithmetic degrades it. The present review adds a cross-platform reading of what each hardware experiment actually evidences: numerical fidelity, real-time execution, analog observation and control behavior are separated into levels of evidence, the implementation parameters needed for comparison are extracted into a common table, and the question of a gain changed during operation is examined against the available control theory.

2. REVIEW METHOD AND EVALUATION FRAMEWORK

This study is a narrative and critical review. It does not follow a systematic-review protocol such as PRISMA, but its literature coverage is documented so that it can be reproduced. The literature was assembled in two steps. First, an initial set of studies on the implementation, analog observation and control of chaotic flows was identified through preliminary non-systematic searches, followed by a further search on chaos control and its application areas; this initial pool corresponds to the 59 references of the originally submitted version, which also include theoretical, numerical-method and manufacturer sources. Second, to check and extend the coverage of this pool, a structured search was carried out on 13 October 2026 in the Directory of Open Access Journals (DOAJ) through its public application programming interface, without a lower date limit (the retrieved records span 2011–2026). Four search strings were applied: S1, (chaotic OR chaos) AND (DSP OR "digital signal processor") in the title or abstract;

S2, (chaotic OR chaos) AND (microcontroller OR Arduino OR STM32) in the title or abstract; S3, (chaotic OR chaos) AND FPGA in the title or abstract, combined with (oscilloscope OR DAC OR "digital-to-analog") in the abstract; and S4, (chaos OR chaotic) AND (control OR stabilization OR synchronization) in the title, combined with (DSP OR microcontroller OR FPGA OR "hardware-in-the-loop" OR Arduino OR dSPACE) in the abstract. DOAJ was chosen because its queries and counts can be reproduced openly; because it indexes only open-access journals, studies in subscription journals were covered through the initial pool, backward citation tracking from the included studies and targeted searches on topics raised during the review (dynamical degradation, switched and gain-scheduled systems, and control of physical chaotic circuits).

Records were eligible when (i) the chaotic system was a continuous-time flow (integer-order, fractional-order or delay), (ii) the flow or its controller was executed on embedded digital hardware (DSP, microcontroller, FPGA or hardware-in-the-loop target), and (iii) a hardware result was reported, either as analog or oscilloscope output or as control or synchronization on hardware. Records were excluded when the chaotic system was a discrete map ($n = 17$), when chaos was not the implemented dynamics, as in chaotic optimizers, chaos-based sequences or unrelated topics ($n = 17$), when no embedded execution was reported ($n = 2$), or when the system type or the role of the hardware could not be identified from the abstract ($n = 3$). The search returned 135 records; after 17 duplicates were removed, 118 records were screened by title and abstract and 79 were eligible (Figure 1). Nine eligible records were already in the initial pool. Most of the 70 newly eligible records report the DSP, microcontroller or FPGA realization of a newly proposed attractor and therefore represent an evidence type already covered; four were added to the synthesis because they belong to categories that were under-represented in the initial pool, namely the coupling of digital hardware with a physical analog chaotic circuit and microcontroller- or FPGA-based synchronization (Yau et al., 2011; Tagne et al., 2022; Babajans et al., 2025; Karimov et al., 2026). Two further hardware studies were identified through the targeted searches (Méndez-Ramírez et al., 2019; Kent et al., 2024), and one further application study on chaotic speed control was added (Sarıkaya et al., 2025). The record-level screening decisions are listed in Supplementary Table S1. Screening and data extraction were not performed in duplicate.

In the critique of the studies, the processor and integrator reported directly by each paper, the observed analog output, the actual target of the control and the limit of the validation obtained were kept separate. For each representative hardware study, the model and integrator, the platform and clock, the model step h, the physical update period T_s, the number format, the analog interface, the validation measure and the control objective were extracted (Table 1). Studies were assessed on their full texts unless marked "Abs", which denotes that only the abstract and the bibliographic record were assessed; for these studies no inference is drawn about what was or was not done. Only studies for which every extracted item could be checked against the assessed text are included in Table 1; the other studies are discussed in the text. In the tables, "not reported" (NR) means that an item was not found in the assessed text; it is not taken as evidence that the corresponding measurement or experiment was not performed. For example, the statement "the oscilloscope portraits agree with MATLAB" is the result of a visual comparison; it does not include the worst-case execution time of the processing loop or Lyapunov stability for a varying gain. Because the structured search covered a single open-access database, screening was not duplicated and no protocol was pre-registered, the study is not presented as a systematic review or meta-analysis. The identified gaps are also interpretations limited to the reviewed studies.

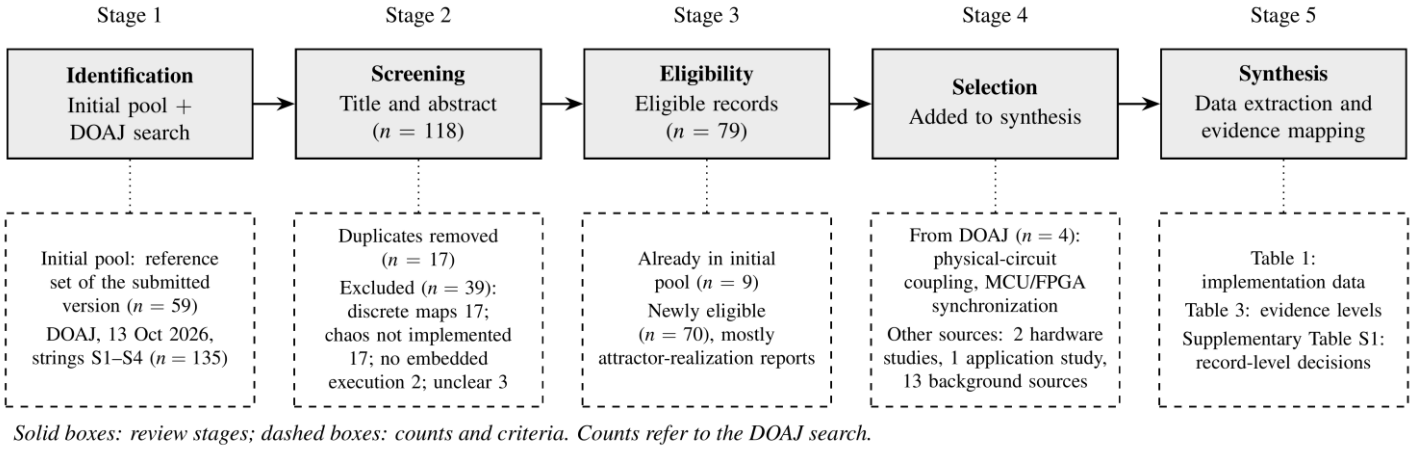


Figure 1. Assembly of the reviewed literature: initial pool, documented DOAJ search (13 October 2026), screening, eligibility and additions from targeted searches. Counts refer to the DOAJ search; record-level decisions are listed in Supplementary Table S1.

3. THEORETICAL BACKGROUND AND NUMERICAL SOLUTION

This section summarizes the numerical concepts on which the hardware comparisons in Sections 4–6 rely. Section 3.1 distinguishes the model step h from the physical update period T_s and relates the choice of integrator to its computational cost, and Section 3.2 discusses how finite precision, the validity horizon of chaotic trajectories and multistability limit what a digital attractor can demonstrate.

3.1 Integration Method, Step Size and Computational Load

On a processor, a chaotic flow takes the form $x[n+1] = \Phi_h(x[n], u[n])$, where Φ_h represents the integrator applied to $dx/d\tau = f(x,u)$ and h is the model step. The physical period T_s of each hardware trigger is a separate quantity. Therefore, when the same h is used with different values of T_s, the discrete index of the generated state sequence may remain the same, but the time scale drawn on the oscilloscope changes. The set-ups of Bulut et al. (2026), using the classical fourth-order Runge–Kutta method (RK4), and Shi et al. (2026), using Euler, are suitable examples of this distinction: since the two papers study different systems, their phase portraits alone do not allow a ranking of the methods in terms of per-sample accuracy or execution time. Figure 2 shows the distinction between these two time scales schematically.

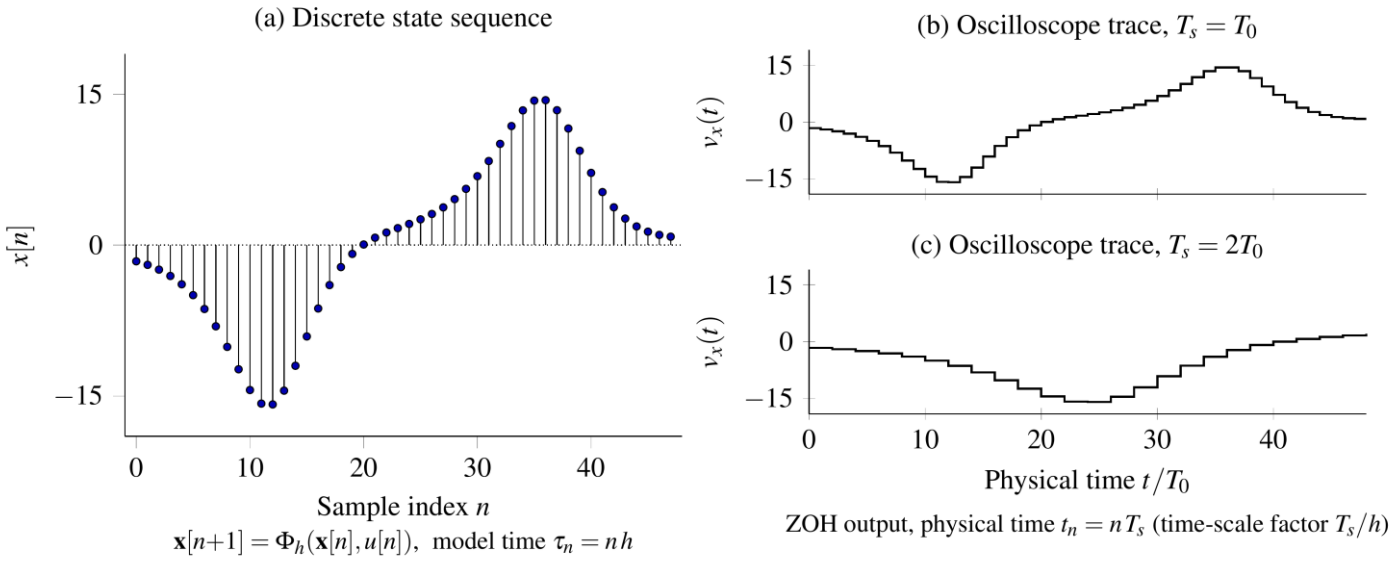


Figure 2. Representation of the same discrete state sequence in model time $\tau_n = nh$ and physical time $t_n = nT_s$: (a) $x[n]$ against the sample index; (b) zero-order-hold output for $T_s = T_0$ and (c) for $T_s = 2T_0$. The sequence was generated from the Lorenz system ($\sigma = 10$, $\rho = 28$, $\beta = 8/3$) using RK4 with $h = 0.03$ from the initial condition $(x, y, z) = (1,1,1)$; the first 2000 steps (60 model time units) were discarded as a transient, and the displayed 48 samples ($n = 2061 - 2108$ of the full sequence) begin 20 samples before the first upward zero crossing of x after the transient.

The fact that the Euler method needs one, Heun's method two and the classical RK4 method four right-hand-side evaluations creates a clear difference in computational cost. However, a simple operation count is incomplete if the truncation error at a fixed h, the h required to meet the same accuracy target and the model time processed per computation are not evaluated together. Practical comparisons of Runge–Kutta formulas and embedded solvers with error estimation show that the relationship between step size and accuracy changes with the method (Dormand & Prince, 1980; Shampine, 1986). The error control of the MATLAB ODE solvers also clarifies how hardware results should be compared with an offline reference (Shampine & Reichelt, 1997).

Sambas et al. (2019), who turned this methodological distinction directly into a hardware comparison, implemented forward Euler and RK4 in the same FPGA design, showed two different attractor views on the oscilloscope and reported resource usage; with a 50 MHz clock, a new iteration required 10 clock cycles for forward Euler and 34 for RK4. Tuna et al. (2019) implemented the Lü–Chen flow on an FPGA with Heun's method and a 32-bit IQ-Math fixed-point representation and presented a numerical error analysis. Damaj et al. (2024) compared different integration orders and floating-point widths for the same Lorenz family in terms of speed, area, power and precision.

3.2 Finite Precision and Validation Horizon

In chaotic solutions, small differences in initial values and computation can grow; this makes it inappropriate to match a hardware trajectory point by point with an offline solution for an unlimited time. The question of numerical orbits raised by Hammel et al. (1987), the study of shadowing by Grebogi et al. (1990) and the work of Sauer et al.

(1997) on the validity horizon form the mathematical basis of this problem. The state error from the same initial condition over a short horizon, and the coverage of the attractor and statistical indicators over a longer horizon, serve different testing purposes. Nepomuceno and Mendes (2017) further show that, for a fixed step and initial condition, different discretization schemes and roundoff yield distinct pseudo-orbits, and they relate the time at which these separate to the largest Lyapunov exponent and the bit resolution.

Goldberg's analysis of floating-point arithmetic explains why bit width cannot be only a technical footnote in a DSP–FPGA comparison (Goldberg, 1991). Yeniçeri et al. (2017) directly examine the dynamical degradation caused by finite precision in a fully digital FPGA implementation of a time-delay chaotic system. Although the results for this specific system cannot be transferred quantitatively to all Lorenz implementations, they show that the short-term appearance of an attractor on a screen is not strong evidence of long-term dynamical behavior. Méndez-Ramírez et al. (2019) analyze, for five three-dimensional flows implemented with Euler, Heun and RK4 on 8-, 16- and 32-bit microcontrollers and an FPGA, the maximum step size and the parameter intervals for which chaos is preserved after discretization.

The validation problem becomes even larger when sensitivity to initial conditions is confused with multistability. When different attractors are obtained from different initial regions with the same parameters, a digital result that looks good for a single initial condition does not exclude another attractor. The memristive system with hidden attractors that Guo et al. (2023) also studied on a DSP and the active memristor model in which Song et al. (2020) investigated transient chaos together with coexisting attractors are more complex examples of this problem than the Lorenz attractor. In these studies, the multistability of the model and the finite number representation of the processor are two separate mechanisms.

The scientific interpretation of an oscilloscope trace depends on the relationship between at least three levels of data: the state computed in the processor, the number written to the DAC and the physical voltage. The effect that changes the appearance of the attractor may be integration and rounding at the first level, scaling and saturation at the second level, or the converter and the measuring instrument at the third level.

4. CHAOTIC FLOW IMPLEMENTATIONS ON EMBEDDED HARDWARE

Here, embedded implementations are classified by hardware type and by the experimental evidence they provide. Generating a chaotic attractor, observing it as an analog signal and stabilizing it with feedback on the same device are different results; Table 1 lists the parameters extracted from representative studies, and Figure 3 shows how completely these parameters are reported.

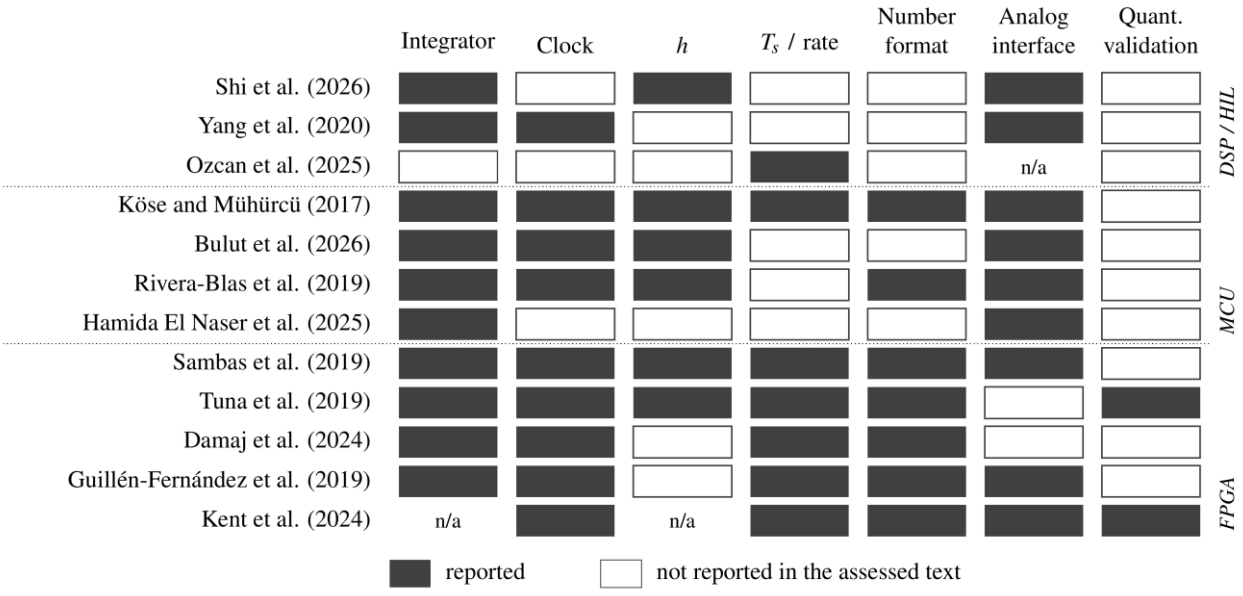


Figure 3. Reporting of the implementation parameters extracted in Table 1. Quantitative validation denotes an error measure against a reference solution rather than a visual comparison of portraits.

4.1 Chaotic Flows and Control on DSPs

In the study by Wang et al. (2015), a fractional-order simplified Lorenz system is examined on a DSP on the basis of Adomian decomposition. The hyperchaotic Lorenz study by He et al. (2015), published in the same year, also converts fractional dynamics into a discrete computational form and implements it on a DSP. Both studies investigate the relationship between the computed dynamics and physical hardware; however, the memory-

sensitive structure of a fractional system cannot be compared directly with the computational budget of an RK4 loop for a three-state integer-order flow.

Liu et al. (2022) relate the phase behavior of a fractional-order model with a memcapacitor and a meminductor to DSP portraits. When read together with the simplified Lorenz approach of Wang et al. (2015), the common point is that the numerical convergence of the fractional derivative is central to the success of the implementation; a different attractor geometry, however, is not a measure of the cost that the fractional order alone places on the processor.

Shi et al. (2026) discretize a three-dimensional system with the Euler method ($h = 0.001$) on a TMS320F28335 and send the DAC output to an oscilloscope; they report qualitative agreement of the portraits with MATLAB and circuit simulations. Yang et al. (2020), who use the same DSP family, combine RK4, a 12-bit, four-channel DAC7724 on a TMS320F28335 running at 150 MHz and a stabilizing controller; they display the states computed on the DSP as analog signals and visualize the convergence to an equilibrium under a fixed gain. The main difference between the two studies is not only the integrator: Shi et al. observe families of attractors by changing a parameter or offset, whereas Yang et al. test the convergence of the state to zero under a given control law.

The study by Yang et al. builds an especially important bridge for the present paper: in the same publication, it links a theoretical stability condition to a DSP and oscilloscope experiment. However, the numerical comparison in the paper is based on $K = -2$ and $K = -16$, while the DSP demonstration is based on $K = -6$; the result does not consist of the same protocol repeated at different gains or of a stability test for a $K[n]$ that changes randomly between samples (Yang et al., 2020).

The HIL study by Ozcan et al. (2025), which compares sliding-mode control (SMC) and input–state linearization (ISL) with a proportional–integral–derivative (PID) controller for the Lorenz system, moves the sampled control computation to C2000 hardware at the order of 1 kHz. This experiment provides stronger evidence of timed execution than an offline simulation; however, it does not identify the virtual plant with a physical Lorenz circuit: the Lorenz model runs on a host computer and exchanges data with the board through a real-time serial link at $T_s = 1$ ms, and the controller is switched on at 20 s after an open-loop phase. The fact that one part of the paper mentions a LAUNCHXL-F28069M while the description of the HIL set-up mentions an F28379D board makes board-level reproduction and execution-time comparison more difficult.

Not every parameter that changes the chaotic behavior in DSP implementations can be called a “control gain”. Dai et al. (2019) observe an attractor enriched by a fractal transformation as an analog signal using a TMS320F28335 and a DAC8552; Guo et al. (2023) examine the hidden attractors of a six-dimensional memristive model and implement it on a DSP with RK4. Song et al. (2020) address the multistability and digital implementation of a system with an active memristor. The contribution of Yang et al. (2020) lies exactly at this boundary: beyond generating the attractor, the stabilization target and the control input are clearly defined.

4.2 Microcontroller Implementations

Köse and Mühürcü (2017) transfer the continuous-time Lorenz equations to a PIC18F452 through transformations in the s- and z-planes and show single-state traces and XY portraits on the oscilloscope; they also measure the cost of one update of the discrete oscillator, 350 µs on the 10 MIPS processor, against a 4 ms update period. Bulut et al. (2026) solve the Scaled Zhongtang flow with RK4 ($\Delta h = 0.001$) on an STM32F429 running at 180 MHz, obtain analog output from the internal 12-bit DAC and compare the oscilloscope results with MATLAB visually, without a quantitative error measure. The common contribution of these two papers is that a physical display can be obtained on low-cost processors; because the mathematical transformation and integration paths differ, they are not a processor comparison of the same solution.

Rivera-Blas et al. (2019) synchronize a master–slave pair of the Petrzela system with active control on a 32-bit ARM-based Arduino Due; after the numerical solution of the equations and the control error are computed, they show the DAC outputs on the oscilloscope. The paper’s explicit discussion of the analog offset and of scaling the state variables to the voltage range is more valuable than a simple “attractor observed” finding: the 12-bit DACs of the Arduino Due cover only 0.5–2.7 V, so the states are mapped to the DAC codes with an offset that cannot be removed in software. The equations are integrated with RK4 at $\Delta t = 0.01$ in IEEE 754 single precision, and the controller is switched on at $t = 4$ s, after which the synchronization errors decay within about 0.8–1 s. While the target in the DSP stabilization of Yang et al. is the zero equilibrium point, the error here is the convergence of the states of two chaotic systems (Rivera-Blas et al., 2019; Yang et al., 2020).

Hamida El Naser et al. (2025) convert the x state of the Sprott-A flow, obtained with RK4, into a pulse-width modulation (PWM) duty cycle for a DC-motor-driven robotic end-effector on an STM32F407 and output the states through the DAC for visual comparison with simulation. The chaotic signal is used as an open-loop speed command

rather than as a stabilizing feedback, so the study is cited here only as an example of a generated chaotic signal used as an actuator command. Chaotic speed commands are used in the same open-loop sense by Sarıkaya et al. (2025), who set the speed of a programmable logic controller (PLC)-controlled biogas mixer from Hadley, Halvorsen, Lorenz and Sprott-A signals and report a higher daily methane yield than with fixed-speed mixing.

Table 1. Data extracted from representative hardware studies. NR: not reported in the assessed text; n/a: not applicable; FT: full text assessed; ADM: Adomian decomposition method; HIL: hardware-in-the-loop; NG-RC: next-generation reservoir computing.

Study	Model; integrator	Platform; clock	h	T_s or update rate	Number format	Analog interface	Validation measure	Control objective	Basis
Shi et al. (2026)	3D flow with offset/frequency control; Euler	TMS320F28335; NR	0.001	NR	NR	DAC module on the board (model NR), oscilloscope	Visual consistency with MATLAB (ode45) and Multisim	None (parameter/offset adjustment)	FT
Yang et al. (2020)	3D flow with nonhyperbolic equilibrium; RK4	TMS320F28335; 150 MHz	NR (symbol ΔT only)	NR (symbol ΔT only)	NR	DAC7724, 12-bit, 4 ch.; ADC NR	Qualitative agreement	Stabilization to equilibrium, fixed $K = -6$	FT
Ozcan et al. (2025)	Lorenz plant model on host PC (solver NR); trapezoidal integration in discrete PID	TI C2000 LaunchPad (F28379D in Sect. 3; LAUNCHXL-F28069M in Sect. 1); NR	NR	$T_s = 0.001$ s (1 kHz)	NR	None (real-time serial link)	ISE, IAE, ITSE and ITAE between controllers	Stabilization (SMC, ISL, PID); control started at 20 s	FT
Köse and Mühürcü (2017)	Lorenz; s- to z-domain (forward difference)	PIC18F452; 10 MHz crystal (10 MIPS)	$T = 0.005$ s (Sect. 2)	$T_s = 4$ ms; measured cost 350 µs per update	Floating point (4-byte) with int16/int8 variables	DAC7611 ×3 (Fig. 9 label), oscilloscope (1D, XY)	Simulink simulations; visual	None	FT
Bulut et al. (2026)	Scaled Zhongtang; RK4	STM32F429; 180 MHz	0.001	NR	NR	Internal 12-bit DAC, 2 ch.	Visual agreement with MATLAB	None	FT
Rivera-Blas et al. (2019)	Petrzela master–slave; RK4	Arduino Due (SAM3X8E, Cortex-M3); up to 84 MHz	0.01	NR	IEEE 754 single precision	DAC0/DAC1, 12-bit, 0.5–2.7 V with offset; TDS 1001B oscilloscope	Visual comparison with simulation	Active synchronization, fixed gains (2, 5, 5); control on at $t = 4$ s	FT
Hamida El Naser et al. (2025)	Sprott-A; RK4	STM32F407 (Cortex-M4); NR	NR	NR	NR	DAC (x, y, z); PWM to L298N driver	Visual comparison	None (open-loop chaotic speed command)	FT
Sambas et al. (2019)	Two circles of equilibria; Euler, RK4	Cyclone IV EP4CGX150DF31C7; 50 MHz	0.001 (0.01 in two captions)	200 ns (Euler), 680 ns (RK4) per iteration	32-bit fixed point (7.25)	16-bit DAC, 1 MS/s	Oscilloscope portraits	None	FT
Tuna et al. (2019)	Lü–Chen; Heun	Virtex-6 xc6vlx75t-3ff484; up to 464.688 MHz (synthesis)	0.01	One output every 47 clock cycles (min. period 2.152 ns)	32-bit IQ-Math fixed point (16Q16)	None (outputs recorded to file)	MSE and RMSE vs. MATLAB Heun model (first 500 values)	None	FT
Damaj et al. (2024)	Lorenz; Euler, RK4, RK6	Stratix IV; F_{max} up to 1329.79 MHz (synthesis)	NR for the cores	Throughput up to 21.17 Gb/s	8-, 16-, 32-, 64-bit floating point	NR (digital outputs)	Simulation; comparison with related work	None	FT
Guillén-Fernández et al. (2019)	SNLF series, Sprott G and L; forward Euler, trapezoidal, RK4	Cyclone IV GX EP4CGX150DF31C7; 14–116 MHz (synthesis)	NR (chosen from eigenvalues)	Iteration latency 51–251 ns (oscillators), 158–387 ns (synchronization)	Fixed point 7.21	Oscilloscope XY (converter NR)	Synchronization error in simulation; image correlation	Pecora–Carroll, Hamiltonian forms and OPCL synchronization	FT
Kent et al. (2024)	Physical double-scroll circuit; NG-RC controller	Max 10 10M50DAF484C6G FPGA	n/a	5 µs control period; latency ≥ 2.5 µs	18-bit fixed point	12-bit ADC; 16-bit DAC values; 1 MS/s	Control error	Stabilization and trajectory tracking of a physical circuit	FT

Table 1 itself illustrates the reporting gap discussed in this review (Figure 3). Shi et al. (2026) give $h = 0.001$ but neither the processor clock, the update period nor the DAC model; Yang et al. (2020) specify the clock and the converter but denote the sampling period only symbolically; Bulut et al. (2026) and Rivera-Blas et al. (2019) give h and the clock but not the update period of the DAC output; and Hamida El Naser et al. (2025) report neither h nor the update period. A measured execution time is reported only by Köse and Mühürcü (2017), who give a cost of 350 µs per update of the discrete Lorenz oscillator on a 10 MIPS PIC18F452 with a 4 ms update period. Otherwise, iteration latency, throughput or control period comes from FPGA design values (Sambas et al., 2019; Tuna et al., 2019; Guillén-Fernández et al., 2019; Damaj et al., 2024; Kent et al., 2024) or from the 1 kHz sampling rate of the hardware-in-the-loop set-up of Ozcan et al. (2025), in which the Lorenz model runs on a host computer and exchanges data with the board over a serial link. Where DSP and microcontroller studies compare hardware output with simulation, the comparison is visual or qualitative; among the studies in Table 1, only Tuna et al. (2019)

compare the hardware sequence with a reference solution through an error measure (MSE and RMSE against a MATLAB Heun model over the first 500 samples). Because the physical update rate is available for too few studies, a comparison of update rate against integrator order and platform cannot yet be constructed from the published data.

4.3 FPGA Implementations and Numerical Arithmetic

The real-time FPGA oscillator of Koyuncu et al. (2014), the Heun/IQ-Math implementation of Tuna et al. (2019), which uses $h = 0.01$, produces one output every 47 clock cycles at up to 464.688 MHz on a Virtex-6 and checks the recorded outputs against MATLAB with MSE and RMSE, and the processing of fractional-order oscillators with the Grünwald–Letnikov approach by Pano-Azucena et al. (2019) show the variety of methods in the FPGA literature. Rodríguez-Muñoz et al. (2024) present experimental results on observing three-dimensional flows from an FPGA as analog attractors. The processing pipeline of an FPGA makes it easier to place the arithmetic into regular clock cycles; however, the register-transfer-level (RTL) clock frequency does not by itself determine the moment at which each sample settles at the analog output or the delay in a multi-channel output. Representative FPGA studies are included in Table 1.

The Euler/RK4 comparison of Sambas et al. (2019) and the evaluation of different floating-point widths and integration orders in terms of area, speed and power by Damaj et al. (2024) limit the claim that "a higher order is always best". Guillén-Fernández et al. (2019) compare three synchronization approaches on an FPGA for different chaotic oscillators and report that the Pecora–Carroll scheme uses fewer logic elements but needs more cycles per iteration (36–40 cycles against 17–21 for Hamiltonian forms and OPCL, in 7.21 fixed-point arithmetic on a Cyclone IV GX). Here, the choice of the control rule is also part of the hardware load. The adaptive synchronization demonstrated by Rameshbabu and Suresh (2020) on a 32-bit fixed-point FPGA shows that a fixed control parameter is not the only option; nevertheless, an adaptively computed gain and a gain changed with a potentiometer are still different system definitions.

When an attractor on an FPGA is not produced with the same model and the same h as an attractor on a DSP, it is difficult to infer a general platform advantage from processing speed. The switched chaotic system of Azzaz et al. (2013), Lorenz-based FPGA implementations (Merah et al., 2013) and the non-embedded example of a Jetson AGX Orin with a custom DAC (Emin & Yaz, 2024) show the range of implementation architectures; however, when the application goal is image security, oscillator generation or analog observation, the meaning of the reported speed measures also changes. The concept of a deadline, which is the basic limit of real-time operation, covers not only the processing capacity of the core but also the interrupt and peripheral load (Liu & Layland, 1973).

The comparison between FPGAs and DSPs takes on another dimension when control is added. The parallel structure of an FPGA may be suitable for computing several state derivatives or synchronization controllers at the same time (Guillén-Fernández et al., 2019; Damaj et al., 2024). On a DSP, in contrast, reading the input from the analog-to-digital converter (ADC) into the control computation at a certain sample boundary and writing the result to the DAC are defined by the execution order of the program (Yang et al., 2020). A number that directly compares the internal clock of an FPGA with the interrupt period of a DSP hides these dependencies and the moment at which the analog output is ready; unless end-to-end latency is measured on the same model, the result remains only an architecture-specific potential.

5. ANALOG OBSERVATION AND CONTROL ARCHITECTURES

The oscilloscope trace of a state computed on a DSP is formed by the conversion between the numerical solution and the analog interface. When an external gain input is added to the control loop, it must also be specified when the variable is read, at which sample it is applied and whether the control acts on a real physical plant or on the model inside the processor. Figure 4 shows these distinctions together with the physical measurement path.

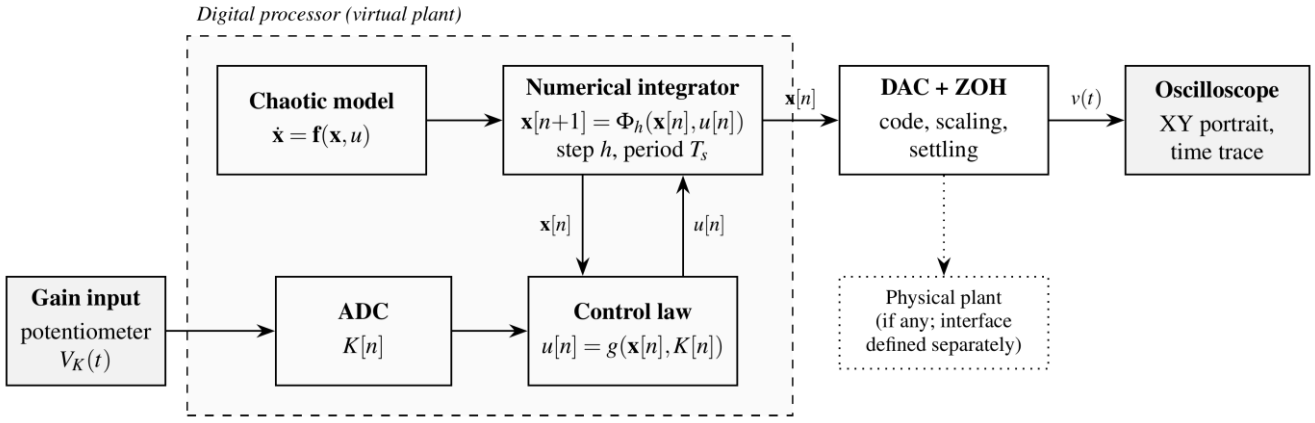


Figure 4. Model solution in a virtual plant, online gain input, DAC and oscilloscope observation. It is not a circuit diagram of a specific hardware board.

5.1 Oscilloscope Observation Through a DAC

The value $x[n]$ stored in the memory of a DSP or microcontroller unit (MCU) is not a voltage. The code applied to the DAC input is mapped to a limited analog range through a chosen offset and scale; for example, $V_x[n] = V_0 + g_x x[n]$ represents only this conversion. The external DACs of Köse and Mühürcü (2017), the internal STM32 DAC of Bulut et al. (2026), the DAC module used with the DSP by Shi et al. (2026) and the DAC7724 of Yang et al. (2020) are different concrete examples of this path. Manufacturer data describe the two-channel voltage output of the DAC8552 and the internal DAC capability of the TMS320F28379D; the DAC8552, for example, is specified as a 16-bit dual-channel converter with a settling time of 10 µs to ±0.003% of full scale (Texas Instruments, 2006, 2024); these are not deadline or chaotic accuracy data measured in the reviewed experiments.

For XY plotting on an oscilloscope, the values $x[n]$ and $y[n]$ on the two channels must correspond to the same sample index. The skew created by consecutive DAC writes, the zero-order hold, analog settling, the oscilloscope bandwidth limit and the persistence setting can affect the appearance of the geometry. The separate discussion of the offset by Rivera-Blas et al. (2019) while monitoring the control error shows concretely that the analog output scale is a research variable that must be resolved before observation. Figure 5 shows this conversion chain and the update timing of the two channels.

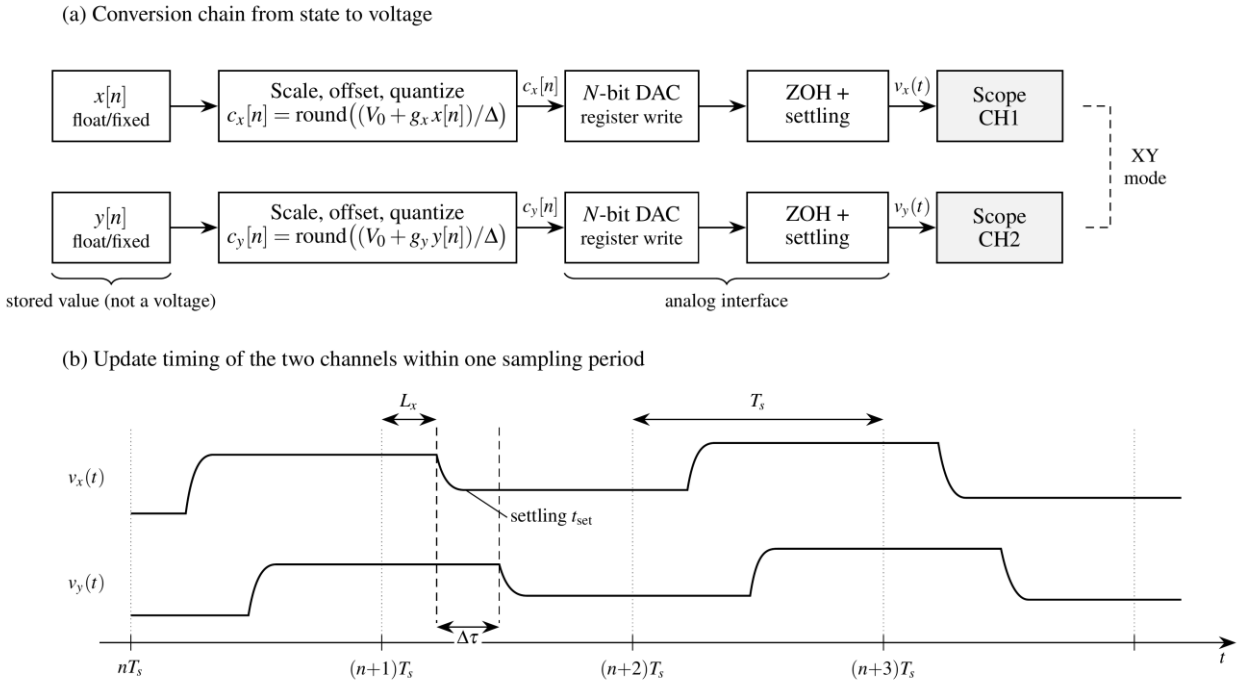


Figure 5. Oscilloscope observation through a DAC: (a) conversion chain from the state variable to the voltage; (b) sample-to-output latency L_x, inter-channel skew $\Delta\tau$ and settling time t_{set} in consecutive DAC writes. The timing is schematic.

For this reason, the criterion “it looks like the MATLAB results” does not have the same strength in different

experiments. The comparison of oscilloscope data with a numerical portrait by Bulut et al. (2026), the qualitative agreement with MATLAB and circuit simulations reported by Shi et al. (2026) and the real outputs recorded during stabilization by Yang et al. (2020) extend one another: the first two make analog reproduction visible, and the third makes the transition after control visible. The separate comparison of error and delay in the synchronization experiment of Guillén-Fernández et al. (2019) goes beyond phase geometry.

5.2 Control Objectives and Hardware Evidence

Ott–Grebogi–Yorke (OGY) control aims to lock onto a selected unstable periodic orbit in the attractor through small parameter perturbations (Ott et al., 1990). Pyragas's delayed feedback is based on the difference between the current state and a delayed state (Pyragas, 1992). In the drive–response approach of Pecora and Carroll (1990), the synchronization error is what matters. Theoretical and experimental reviews of chaos control emphasize that these objectives are different from each other (Boccaletti et al., 2000; Fradkov & Evans, 2005); synchronization studies in electronic circuits have shown that the objective can be observed in physical hardware (Carroll & Pecora, 1991; Cuomo & Oppenheim, 1993; Kocarev & Parlitz, 1995). A "control gain" is not just a symbol: which state it is applied to, whether it includes a delay and whether the target is an equilibrium, a periodic orbit or synchronization all change the meaning of the gain.

The experiment of Ditto et al. (1990), in which unstable periodic orbits were selected through small parameter perturbations, shows that the OGY principle can be transferred to a physical set-up. The test of delayed feedback on a boost DC–DC converter by Natsheh et al. (2009) belongs to a separate class of applications, in which the controlled object is not a mathematical flow created inside a processor but a power electronic circuit with its own switching behavior.

Yesil et al. (2025) build a memristive jerk oscillator as a physical analog circuit, design nonlinear feedback, classical sliding mode and integral sliding mode controllers, and report microcontroller-based control of the circuit with outputs kept within ±6 V. Only the abstract and the bibliographic record of this study were assessed (Abs in Table 2); the absence of timing, converter and channel-recording details in the abstract is therefore not taken as evidence that they were not reported in the full article. On the information assessed, the study is the closest microcontroller example of closing the loop around a physical chaotic oscillator, and its full experimental protocol would be needed for a one-to-one comparison with a DSP loop.

Lian et al. (2002) test adaptive chaos synchronization through a scalar drive signal in DSP-based experiments. This study is important as one of the early examples showing that gains can be adjusted online in a digital controller; however, a gain produced from the states by an adaptation law does not have the same dynamics as a fixed coefficient changed manually by an operator with an external potentiometer. Yang et al. (2020) demonstrate DSP stabilization under a specific fixed K, and Rivera-Blas et al. (2019) demonstrate microcontroller synchronization under fixed design conditions. Together, these studies invalidate the view that "chaos cannot be controlled on digital hardware"; however, they do not provide a universal stability result for a $K(t)$ that is changed continuously and irregularly by a user.

Two numerical studies by the authors' group, sliding mode suppression of the Newton–Leipnik system (Demirsoy & Hamida El Naser, 2026) and grey-wolf-optimized pole placement for the Moore–Spiegel oscillator (Sarıkaya et al., 2026), illustrate robustness and gain-selection questions but report no DSP timing, DAC voltage or online gain change; they are therefore not used as hardware evidence in this review.

5.3 Varying Gain With Virtual and Physical Plants

If a DSP computes both the flow $x[n]$ and the control $u[n]$ internally, the object under study is a digital virtual plant; the DAC only carries the computed result outside. If measurements are taken with an ADC from a physical Chua circuit or another analog oscillator and the control force is applied back through a DAC, the second set-up is a real physical plant. Both set-ups are meaningful, but in a physical plant, sensor noise, the output stage, and measurement and actuation delays enter the closed loop directly.

Studies that close the loop around a physical analog chaotic circuit remain few, and in the searched literature they are mostly FPGA-based. Yau et al. (2011) implemented a digital PID controller on an FPGA, with gains optimized offline by evolutionary programming, to synchronize Sprott circuits built from operational amplifiers, resistors and capacitors. Kent et al. (2024) measured the voltages of a physical double-scroll circuit with 12-bit ADCs on a Max 10 FPGA and applied a next-generation reservoir computing controller with a 5 μs update period, both to stabilize the unstable steady state and to track a random trajectory. Babajans et al. (2025) synchronized a discrete-component Vilnius oscillator with an FPGA counterpart integrated by the Euler–Cromer method in fixed-point arithmetic, in both the analog–discrete and the discrete–analog directions, and Karimov et al. (2026) synchronized a physical Chua circuit with its digital counterpart on a MyRIO-1900 FPGA. Among microcontroller studies, the

sliding mode control of a memristive jerk circuit by Yesil et al. (2025) is the closest example, whereas Tagne et al. (2022) replace the analog jerk circuit itself by a PIC16F877A implementation with 8-bit R-2R DAC outputs and study the synchronization of the digital oscillators. None of these studies reports a gain changed by an operator during operation.

Nešić and Teel (2004) study the stability of nonlinear sampled-data systems based on approximate discrete-time models; the input delay approaches of Fridman and co-workers show how sampling and the moment of actuation can affect stability (Fridman et al., 2004; Fridman, 2010). A property proven under a fixed K is transferred to a nonlinear, time-varying discrete system when $K[n]$ changes between samples. In addition, whether the integrator uses the same $u[n]$ in all RK4 intermediate stages or reads the gain again at each stage defines two different discrete maps.

The stability of time-varying gains is not, however, an open question in general. If the operator selects among a finite set of gains, the closed loop is a switched system: stability under arbitrary switching requires a common Lyapunov function, whereas stability under constrained switching can be guaranteed by dwell-time or average dwell-time conditions that bound how often the gain may change (Hespanha & Morse, 1999; Liberzon & Morse, 1999; Liberzon, 2003). If the gain varies continuously but slowly, results for slowly varying systems guarantee stability when every frozen-gain closed loop is stable and the rate of variation is sufficiently small (Desoer, 1969; Khalil, 2002). If the gain is scheduled on a measured variable, the gain-scheduling and linear parameter-varying literature provides analysis and design tools, again with conditions on the rate of the scheduling variable (Rugh & Shamma, 2000). For a potentiometer-driven gain, these results translate into testable requirements: a bounded set of admissible gains, a minimum dwell time or a maximum rate of change imposed by filtering or rate limiting, and a frozen-gain stability margin over the whole set. What is less documented is not this theory but its experimental use with chaotic flows on embedded hardware, where ADC quantization, filter delay and the sample at which a new gain takes effect determine whether the theoretical assumptions are met.

The scientific content of the potentiometer use case introduced in Section 1 depends more on the role of the gain profile in the experiment than on the existence of the knob. When consecutive ADC readings fluctuate because of quantization and noise, the applied $K[n]$ may change even if the knob is not turned. Filtering reduces this effect but adds phase delay; stronger filtering and a faster control response cannot be guaranteed at the same time. The proposal to record the applied $K[n]$ and the potentiometer position separately in a new experiment is an inference of this review aimed at making published fixed-gain and adaptive-gain studies directly comparable. Figure 6 shows these distinctions schematically on a common time axis.

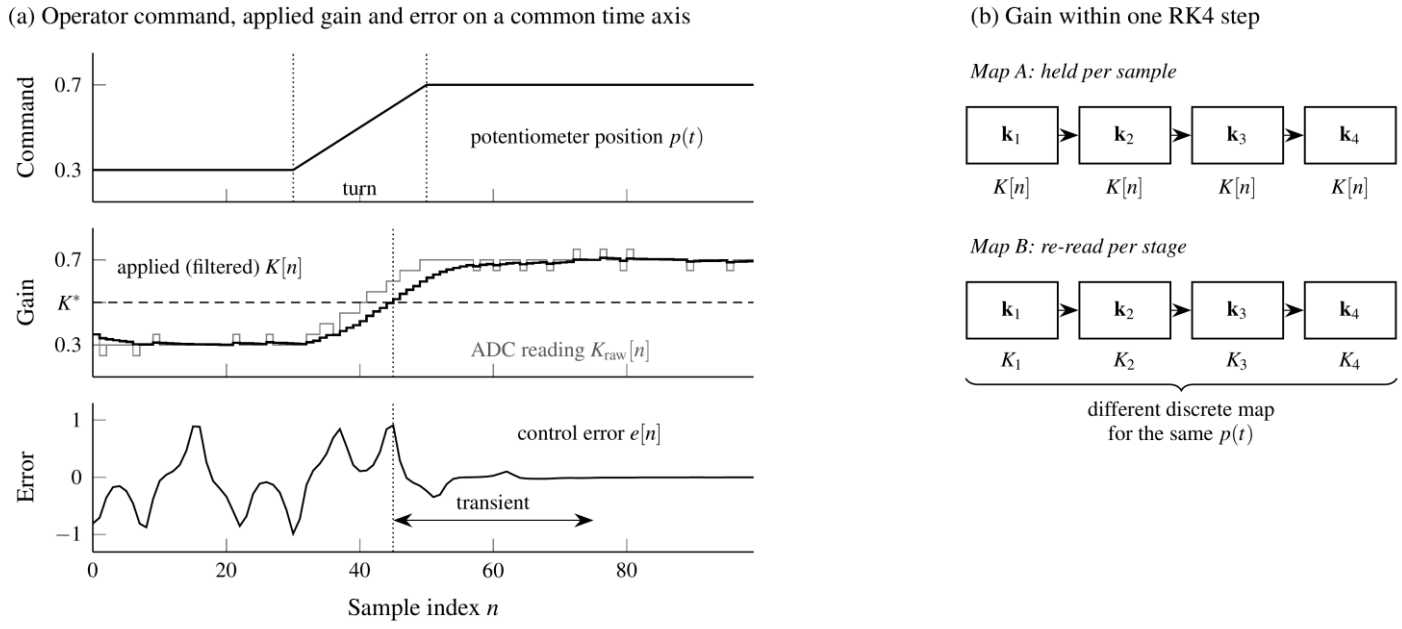


Figure 6. Gain change during operation: (a) potentiometer command $p(t)$, ADC reading $K_{\mathrm{raw}}[n]$, filtered and applied $K[n]$, and control error $e[n]$; (b) holding the gain per sample within an RK4 step (Map A) or re-reading it at each intermediate stage (Map B). The curves are illustrative and do not contain measured experimental data.

Table 2. Correspondence between control objectives, the controlled object and the interpretation of a varying gain across studies. Abs: assessed from the abstract only.

Source	Control objective and gain	Strength	Remaining distinction
Ott et al. (1990); Pyragas (1992)	Unstable orbit; parameter or delay	Theoretical basis of the control objective	Not a DSP/DAC execution measurement.
Lian et al. (2002)	Adaptive synchronization on a DSP	Gain computed online	Not a free $K[n]$ command from the user.
Yang et al. (2020)	Suppression to an equilibrium on a DSP, fixed $K = -6$	Theory and analog stabilization	Stability during a K transition is not examined.
Rivera-Blas et al. (2019)	Active master–slave synchronization on an MCU	Control error and DAC output	Differs from a loop with a physical plant.
Rameshbabu and Suresh (2020)	Adaptive synchronization on an FPGA	Fixed-point control implementation	The gain source is an adaptation law.
Yau et al. (2011), Abs	FPGA PID synchronization of op-amp Sprott circuits; gains optimized offline by evolutionary programming	Physical analog circuits in the loop	Fixed gains; the abstract gives no timing or converter details.
Kent et al. (2024)	NG-RC control of a physical double-scroll circuit on an FPGA	Physical plant measured through ADCs; 5 µs update period	Controller trained offline; no operator gain change.
Babajans et al. (2025)	Pecora–Carroll synchronization between a Vilnius oscillator on a PCB and an FPGA counterpart	Analog–discrete coupling in both directions	Synchronization, not stabilization; no gain change.
Karimov et al. (2026), Abs	Time-reversible synchronization of a physical Chua circuit with an FPGA (MyRIO-1900) counterpart	Physical circuit; NRMSE threshold of 0.01	Sensing application; no gain change.
Yesil et al. (2025), Abs	Nonlinear, sliding mode and integral sliding mode control of a memristive jerk circuit by a microcontroller	Physical analog circuit	The abstract gives no timing or converter details.

6. EVIDENCE FRAMEWORK AND METHODOLOGICAL LIMITATIONS

Evaluating the hardware image of a continuous-time chaotic flow requires attention to several different limits at the same time, from integration accuracy to the observation of the analog output. In the reviewed studies, each of these limits is documented to a different extent; the following subsections are therefore organized around the interpretability of the published results. Figure 7 summarizes these levels of evidence and the points at which each of them is measured. The caveats raised in the preceding sections are stated here once, as a framework, and are referred to rather than repeated elsewhere. Level 1 concerns numerical fidelity and is bounded by the validity horizon (Section 3.2); Level 2 concerns execution and analog output, and a visible attractor neither guarantees that deadlines are met nor bounds the skew between channels (Sections 5.1 and 6.1); Level 3 concerns the control transition, and a result obtained at a fixed gain does not establish the behavior under a time-varying $K[n]$ (Section 5.3). Table 3 maps representative studies onto these levels.

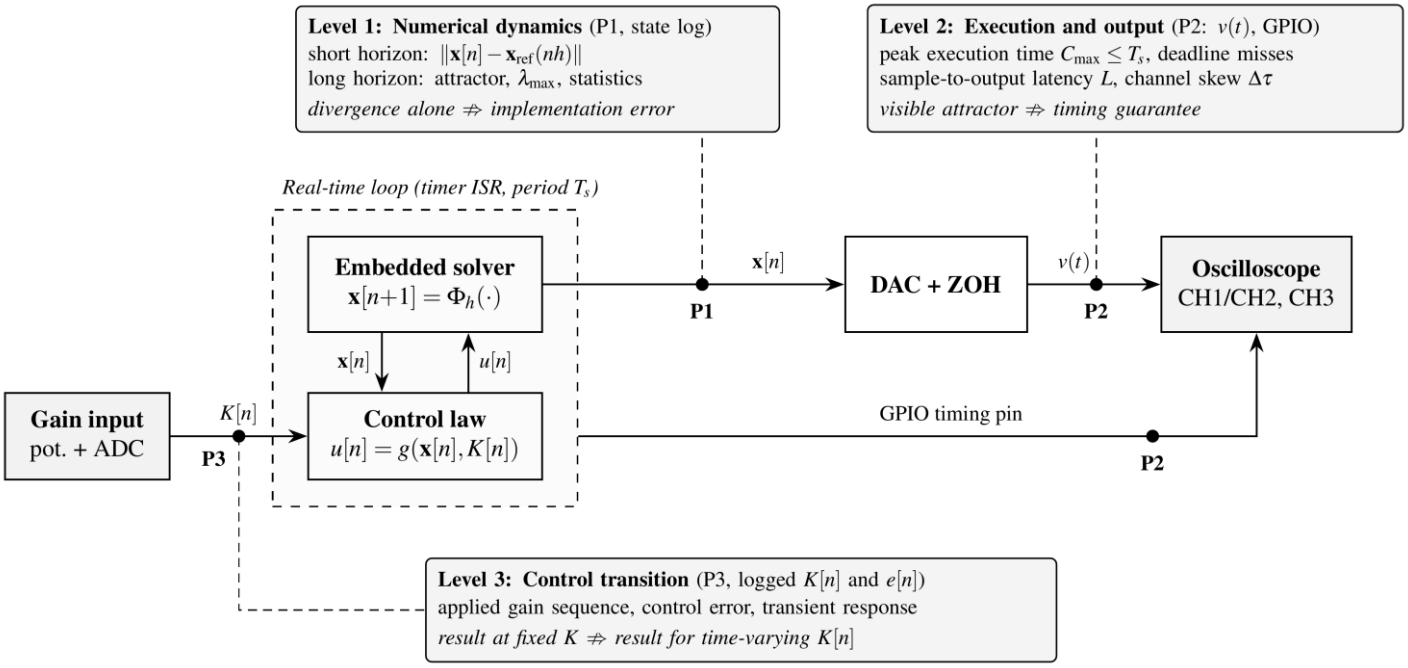


Figure 7. Separate levels of evidence for numerical dynamics (Level 1), real-time execution and analog output (Level 2) and control transition (Level 3), with measurement points P1–P3. The symbol ⇏ indicates that evidence at the given level does not by itself establish the stated result. The scheme does not contain measured experimental data.

Table 3. Mapping of representative studies onto the evidence levels of Figure 7. ● : quantitative evidence reported (measured or derived from the design); ◐: qualitative or visual evidence, or control under a fixed gain only; –: not reported in the assessed text; n/a: not applicable to the objective of the study. No study reports a logged gain sequence during an operator-driven gain change; the closest documented transitions are step activations of fixed-gain controllers.

Study	Level 1: numerical validation	Level 2: timing	Level 2: analog observation	Level 3: control
Shi et al. (2026)	◐	–	◐	n/a
Köse and Mühürcü (2017)	◐	● (350 µs per update, measured)	◐	n/a
Bulut et al. (2026)	◐	–	◐	n/a
Hamida El Naser et al. (2025)	◐	–	◐	n/a
Yang et al. (2020)	◐	–	◐	◐ (fixed K)
Rivera-Blas et al. (2019)	◐	–	◐	◐ (fixed gains; switched on at t = 4 s)
Ozcan et al. (2025)	–	◐ (1 kHz rate)	n/a (serial link)	◐ (fixed gains; switched on at 20 s)
Sambas et al. (2019)	◐	● (cycles per iteration)	◐	n/a
Tuna et al. (2019)	● (MSE, RMSE vs. MATLAB)	● (47 cycles per output)	–	n/a
Damaj et al. (2024)	◐	● ($F_{\max}$, throughput)	–	n/a
Guillén-Fernández et al. (2019)	◐	● (cycles, iteration latency)	◐	◐ (fixed-gain synchronization)
Kent et al. (2024)	n/a	● (5 µs period, latency)	● (ADC-measured circuit)	◐ (trained controller)

6.1 Execution Time and Real-Time Operation

The clock speed of a DSP or FPGA does not by itself show that the state update is completed before the physical sampling instant. ADC reading, filtering, control computation, integration, DAC transfer and interrupt overhead should be measured together. The peak execution time, the number of deadline violations and the latency from sample to analog output should be reported separately from the average loop time. Figure 8 defines these

quantities on a single time axis.

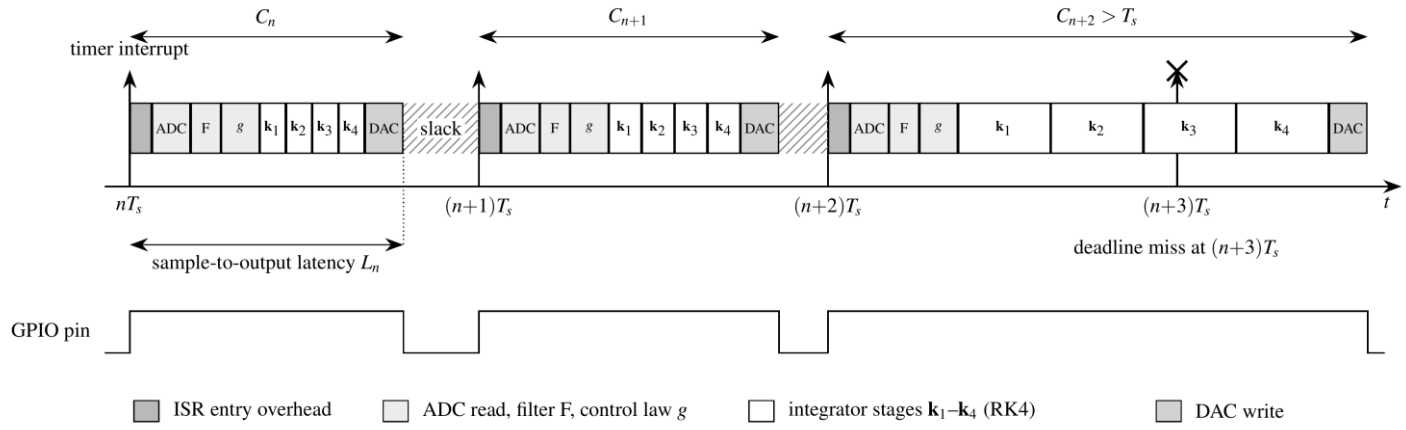


Figure 8. Execution time C_n, sample-to-output latency L_n, slack and a deadline violation in sampling cycles triggered by a timer interrupt. The general-purpose input/output (GPIO) trace is an example of measuring execution time with an oscilloscope. The durations are schematic.

7. COMPARATIVE EVALUATION AND DISCUSSION

The apparent tension in the literature that is easiest to misinterpret is the choice between Euler and RK4. Shi et al. (2026) have shown the feasibility of Euler on a DSP and its analog portrait; Bulut et al. (2026) have shown that RK4 can be implemented on an MCU and compared with an oscilloscope. Sambas et al. (2019), who evaluate the two methods on the same model, and Damaj et al. (2024), who examine a wider FPGA resource–precision space, give concrete context to the trade-off between high order and high computational load. The results of Shi and Bulut are not opposed to each other; since the model, step size, arithmetic and validation measure are not fixed, they do not establish the superiority of either method. A contradiction appears only when answers to different research questions are reduced to a single “fastest and most accurate” ranking.

The second tension lies between “showing control in hardware” and “control being robust to a varying gain”. The fixed-gain stabilization of Yang et al. (2020) and the synchronization of Rivera-Blas et al. (2019) prove that digital control and DAC observation can be carried out together. Therefore, the research contribution does not come simply from adding an adjustable knob, but from the relationship, established experimentally and if possible through the dwell-time, rate-of-variation or frozen-gain conditions reviewed in Section 5.3, between the $K[n]$ sequence produced by this knob and the control objective.

The limitations of the review are also part of its findings. Most of the selected studies use different attractors, different number representations and different types of validation; this heterogeneity does not produce an effect size that could be combined under a single number. Although the supplementary search is documented, it covered one open-access database, screening was not duplicated and several studies were assessed from abstracts only; it is therefore not claimed that all DSP experiments in the field are covered.

8. FUTURE RESEARCH DIRECTIONS

In future studies, the unit of comparison could be formed jointly by the model parameters, h, T_s, the numerical type and the DAC channel, instead of a single processor frequency. If the numerical state, the applied DAC code and the analog record are collected simultaneously under predefined gain steps, it becomes easier to test whether a regime change is related to the control, the integration or the measurement chain.

A priority comparison could involve generating a single integer-order chaotic flow with the same parameters on a DSP and on a similar hardware class with Euler, Heun and RK4, and reporting h, T_s, the arithmetic format and the voltage scale together. Such a protocol would better separate the comparisons in the literature in which the method and the platform change at the same time (Sambas et al., 2019; Damaj et al., 2024; Bulut et al., 2026; Shi et al., 2026).

Concrete measures can be suggested for the two horizons. For the short horizon, the state error between the hardware sequence, logged at P1 in Figure 7 or reconstructed from the DAC codes, and a high-accuracy offline reference started from the same initial condition can be reported as a normalized root-mean-square error, together with the time at which the error first exceeds a stated threshold, for example 10% of the attractor diameter; expressing this time in Lyapunov times, $\lambda_{\max} t$, makes results comparable across models (Sauer et al., 1997; Nepomuceno & Mendes, 2017). For the long horizon, where point-wise agreement is no longer expected, invariant

properties are appropriate: the largest Lyapunov exponent estimated from the recorded series (Wolf et al., 1985; Rosenstein et al., 1993), the 0–1 test for chaos (Gottwald & Melbourne, 2004), and the amplitude histogram or power spectrum of each state compared with those of the reference. The recording duration should be chosen so that the short-horizon window covers the expected divergence time, while the long-horizon record contains enough orbits for the invariant estimates to converge; a practical check is to report the estimates for two record lengths and show that they agree within a stated tolerance. The sampling rate of the recording should, in addition, resolve the fastest component of the DAC output without aliasing.

On the control side, the initial and target errors should first be defined with a fixed K, and then predefined gain steps and bounded ramp profiles should be applied. If a finding obtained with a virtual plant is to be transferred to a physical chaotic circuit, it should be tested again with the same error measure, including sensor noise and the DAC actuation delay (Lian et al., 2002; Rivera-Blas et al., 2019; Yang et al., 2020).

In application research, the chaotic source should first be matched with the goal: speed fluctuation for a PWM motor command (Hamida El Naser et al., 2025), dissolution and conductivity for a chaotic stirrer (Demirsoy et al., 2025), methane yield and combustion duration for chaotic speed-controlled mixing in a biogas digester (Sarıkaya et al., 2025), and the conducted electromagnetic interference (EMI) spectrum for chaotic modulation in a power converter (Wang et al., 2016), where chaos is exploited rather than suppressed as in the delayed-feedback control of a boost converter (Natsheh et al., 2009). For these outputs, the attractor geometry may be a necessary computational check, but it is not application performance on its own.

9. CONCLUSION

This review examined how continuous-time chaotic flows are implemented on DSPs, microcontrollers and FPGAs, observed through DACs and controlled in hardware, and it leads to three findings. First, the evaluation criteria brought together by the review are separable: numerical fidelity (short-horizon error and long-horizon invariants), real-time execution (peak execution time, deadline misses and sample-to-output latency), analog observation (scaling, quantization and inter-channel skew) and control behavior (error under a fixed gain and during a gain transition). An attractor on the oscilloscope is evidence of physical generation but addresses only part of the second level (Figure 7, Table 3).

Second, the published implementations rarely report the parameters needed to compare them. In the extracted studies, the integration step, the physical update period and the number format are seldom given together, validation against an offline reference is mostly visual, and timing information, where available, mostly consists of design values such as iteration latency, throughput or control period rather than measured worst-case execution times (Table 1, Figure 3). Apparent contradictions, such as Euler versus RK4 or FPGA versus DSP speed, therefore reflect different models, steps and validation criteria rather than conflicting results (Sambas et al., 2019; Damaj et al., 2024; Bulut et al., 2026; Shi et al., 2026).

Third, the research need that follows is experimental rather than theoretical. Fixed-gain stabilization, adaptive synchronization and closed-loop control of physical chaotic circuits have been demonstrated (Lian et al., 2002; Rivera-Blas et al., 2019; Yang et al., 2020; Kent et al., 2024), and switched-system, slowly varying and gain-scheduling theory provides conditions for gains that change in time (Hespanha & Morse, 1999; Rugh & Shamma, 2000; Liberzon, 2003). What is missing is an experiment that logs, on one clock axis, the processor state, the DAC code, the measured voltage, the applied gain and the control error while an operator changes the gain, and that checks the measured gain profile against these conditions. Reporting h, T_s, the number format, the converter path and quantitative validation measures in future studies would allow the pieces reviewed here to be compared directly.

Acknowledgment

None.

Conflict of Interest Statement

The authors declare that there is no conflict of interest.

Author Contribution Statement

Berk Demirsoy: Conceptualization, literature search and screening, data curation, visualization, writing – original draft. Mert Süleyman Demirsoy: Methodology, critical evaluation of the literature, results and discussion, writing – review and editing. Yusuf Hamida El Naser: Critical evaluation of the literature, results and discussion, supervision.

All authors have read and approved the final version of the manuscript.

REFERENCES

Azzaz, M. S., Tanougast, C., Sadoudi, S., Fellah, R., & Dandache, A. (2013). A new auto-switched chaotic system and its FPGA implementation. *Communications in Nonlinear Science and Numerical Simulation, 18*(7), 1792–1804. https://doi.org/10.1016/j.cnsns.2012.11.025

Babajans, R., Cirjulina, D., & Kolosovs, D. (2025). Field-programmable gate array-based chaos oscillator implementation for analog–discrete and discrete–analog chaotic synchronization applications. *Entropy, 27*(4), 334. https://doi.org/10.3390/e27040334

Boccaletti, S., Grebogi, C., Lai, Y.-C., Mancini, H., & Maza, D. (2000). The control of chaos: theory and applications. *Physics Reports, 329*(3), 103–197. https://doi.org/10.1016/s0370-1573(99)00096-4

Bulut, S., Pehlivan, I., & Arıcıoğlu, B. (2026). Cost-effective hardware realization of chaotic systems via high-performance STM32 DAC interface. *Chaos and Fractals, 3*(1), 16–20. https://doi.org/10.69882/adba.chf.2026013

Carroll, T. L., & Pecora, L. M. (1991). Synchronizing chaotic circuits. *IEEE Transactions on Circuits and Systems, 38*(4), 453–456. https://doi.org/10.1109/31.75404

Chen, G., & Ueta, T. (1999). Yet another chaotic attractor. *International Journal of Bifurcation and Chaos, 9*(7), 1465–1466. https://doi.org/10.1142/s0218127499001024

Chua, L., Komuro, M., & Matsumoto, T. (1986). The double scroll family. *IEEE Transactions on Circuits and Systems, 33*(11), 1072–1118. https://doi.org/10.1109/tcs.1986.1085869

Cuomo, K. M., & Oppenheim, A. V. (1993). Circuit implementation of synchronized chaos with applications to communications. *Physical Review Letters, 71*(1), 65–68. https://doi.org/10.1103/physrevlett.71.65

Dai, S., Sun, K., He, S., & Ai, W. (2019). Complex chaotic attractor via fractal transformation. *Entropy, 21*(11), 1115. https://doi.org/10.3390/e21111115

Damaj, I., Zaher, A., & Lawand, W. (2024). Optimizing FPGA implementation of high-precision chaotic systems for improved performance. *PLOS ONE, 19*(4), e0299021. https://doi.org/10.1371/journal.pone.0299021

Demirsoy, M. S., & Hamida El Naser, Y. (2026). Dynamical analysis and robust chaos suppression of the Newton-Leipnik system via sliding mode control. *Dicle Üniversitesi Mühendislik Fakültesi Mühendislik Dergisi, 17*(2). https://doi.org/10.24012/dumf.1964154

Demirsoy, M. S., Sarıkaya, M. S., Kurnaz, S., Hamida El Naser, Y., Hazar, M. A., Coşkun, S., Kutlu, M. Ç., & Hazar, G. (2025). Enhancing stirring performance through chaotic position control in a magnetic stirrer. *Journal of the Brazilian Society of Mechanical Sciences and Engineering, 47*, Article 580. https://doi.org/10.1007/s40430-025-05840-2

Desoer, C. A. (1969). Slowly varying system ẋ = A(t)x. *IEEE Transactions on Automatic Control, 14*(6), 780–781. https://doi.org/10.1109/TAC.1969.1099336

Ditto, W. L., Rauseo, S. N., & Spano, M. L. (1990). Experimental control of chaos. *Physical Review Letters, 65*(26), 3211–3214. https://doi.org/10.1103/PhysRevLett.65.3211

Dormand, J. R., & Prince, P. J. (1980). A family of embedded Runge-Kutta formulae. *Journal of Computational and Applied Mathematics, 6*(1), 19–26. https://doi.org/10.1016/0771-050x(80)90013-3

Emin, B., & Yaz, M. (2024). Digital implementation of chaotic systems using Nvidia Jetson AGX Orin and custom DAC converter. *Chaos and Fractals, 1*(1), 38–41. https://doi.org/10.69882/adba.chf.2024075

Fradkov, A. L., & Evans, R. J. (2005). Control of chaos: Methods and applications in engineering. *Annual Reviews in Control, 29*(1), 33–56. https://doi.org/10.1016/j.arcontrol.2005.01.001

Fridman, E. (2010). A refined input delay approach to sampled-data control. *Automatica, 46*(2), 421–427. https://doi.org/10.1016/j.automatica.2009.11.017

Fridman, E., Seuret, A., & Richard, J. P. (2004). Robust sampled-data stabilization of linear systems: an input delay approach. *Automatica, 40*(8), 1441–1446. https://doi.org/10.1016/j.automatica.2004.03.003

Goldberg, D. (1991). What every computer scientist should know about floating-point arithmetic. *ACM Computing Surveys, 23*(1), 5–48. https://doi.org/10.1145/103162.103163

Gottwald, G. A., & Melbourne, I. (2004). A new test for chaos in deterministic systems. *Proceedings of the Royal Society of London. Series A: Mathematical, Physical and Engineering Sciences, 460*(2042), 603–611. https://doi.org/10.1098/rspa.2003.1183

Grebogi, C., Hammel, S. M., Yorke, J. A., & Sauer, T. (1990). Shadowing of physical trajectories in chaotic dynamics: Containment and refinement. *Physical Review Letters, 65*(13), 1527–1530. https://doi.org/10.1103/physrevlett.65.1527

Guillén-Fernández, O., Meléndez-Cano, A., Tlelo-Cuautle, E., Núñez-Pérez, J. C., & Rangel-Magdaleno, J. de J. (2019). On the synchronization techniques of chaotic oscillators and their FPGA-based implementation for secure image transmission. *PLOS ONE, 14*(2), e0209618. https://doi.org/10.1371/journal.pone.0209618

Guo, Z., Wen, J., & Mou, J. (2023). Dynamic analysis and DSP implementation of memristor chaotic systems with multiple forms of hidden attractors. *Mathematics, 11*(1), 24. https://doi.org/10.3390/math11010024

Hamida El Naser, Y., Demirsoy, B., Erin, K., & Demirsoy, M. S. (2025). Chaotic speed control of a DC motor using the Sprott-A system for robotic end-effector applications. *Black Sea Journal of Engineering and Science, 8*(5), 1406–1414. https://doi.org/10.34248/bsengineering.1703012

Hammel, S. M., Yorke, J. A., & Grebogi, C. (1987). Do numerical orbits of chaotic dynamical processes represent true orbits? *Journal of Complexity, 3*(2), 136–145. https://doi.org/10.1016/0885-064x(87)90024-0

He, S., Sun, K., & Wang, H. (2015). Complexity analysis and DSP implementation of the fractional-order Lorenz hyperchaotic system. *Entropy, 17*(12), 8299–8311. https://doi.org/10.3390/e17127882

Hespanha, J. P., & Morse, A. S. (1999). Stability of switched systems with average dwell-time. In *Proceedings of the 38th IEEE Conference on Decision and Control* (Vol. 3, pp. 2655–2660). IEEE. https://doi.org/10.1109/CDC.1999.831330

Karimov, A., Shirnin, K., Babkin, I., Burundukov, P., Rybin, V., & Butusov, D. (2026). Time-reversible synchronization of Chua circuits for edge intelligent sensors. *Mathematics, 14*(8), 1359. https://doi.org/10.3390/math14081359

Kent, R. M., Barbosa, W. A. S., & Gauthier, D. J. (2024). Controlling chaos using edge computing hardware. *Nature Communications, 15*, 3886. https://doi.org/10.1038/s41467-024-48133-3

Khalil, H. K. (2002). *Nonlinear systems* (3rd ed.). Prentice Hall.

Kocarev, L., & Parlitz, U. (1995). General approach for chaotic synchronization with applications to communication. *Physical Review Letters, 74*(25), 5028–5031. https://doi.org/10.1103/physrevlett.74.5028

Köse, E., & Mühürcü, A. (2017). Realization of a digital chaotic oscillator by using a low cost microcontroller. *Engineering Review, 37*(3), 341–348. https://hrcak.srce.hr/184823

Koyuncu, İ., Özcerit, A. T., & Pehlivan, İ. (2014). Implementation of FPGA-based real time novel chaotic oscillator. *Nonlinear Dynamics, 77*(1-2), 49–59. https://doi.org/10.1007/s11071-014-1272-x

Li, S., Chen, G., & Mou, X. (2005). On the dynamical degradation of digital piecewise linear chaotic maps. *International Journal of Bifurcation and Chaos, 15*(10), 3119–3151. https://doi.org/10.1142/S0218127405014052

Lian, K.-Y., Liu, P., Chiang, T.-S., & Chiu, C.-S. (2002). Adaptive synchronization design for chaotic systems via a scalar driving signal. *IEEE Transactions on Circuits and Systems I: Fundamental Theory and Applications, 49*(1), 17–27. https://doi.org/10.1109/81.974871

Liberzon, D. (2003). *Switching in systems and control*. Birkhäuser. https://doi.org/10.1007/978-1-4612-0017-8

Liberzon, D., & Morse, A. S. (1999). Basic problems in stability and design of switched systems. *IEEE Control Systems Magazine, 19*(5), 59–70. https://doi.org/10.1109/37.793443

Liu, C. L., & Layland, J. W. (1973). Scheduling algorithms for multiprogramming in a hard-real-time environment. *Journal of the ACM, 20*(1), 46–61. https://doi.org/10.1145/321738.321743

Liu, X., Mou, J., Wang, J., Banerjee, S., & Li, P. (2022). Dynamical analysis of a novel fractional-order chaotic system based on memcapacitor and meminductor. *Fractal and Fractional, 6*(11), 671. https://doi.org/10.3390/fractalfract6110671

Lorenz, E. N. (1963). Deterministic nonperiodic flow. *Journal of the Atmospheric Sciences, 20*(2), 130–141. https://doi.org/10.1175/1520-0469(1963)020<0130:dnf>2.0.co;2

Matsumoto, T. (1984). A chaotic attractor from Chua's circuit. *IEEE Transactions on Circuits and Systems, 31*(12), 1055–1058. https://doi.org/10.1109/tcs.1984.1085459

Méndez-Ramírez, R., Arellano-Delgado, A., Murillo-Escobar, M., & Cruz-Hernández, C. (2019). Degradation analysis of chaotic systems and their digital implementation in embedded systems. *Complexity, 2019*, 9863982. https://doi.org/10.1155/2019/9863982

Merah, L., Ali-Pacha, A., Said, N. H., & Mamat, M. (2013). Design and FPGA implementation of Lorenz chaotic system for information security issues. *Applied Mathematical Sciences, 7(5), 237–246.* https://doi.org/10.12988/ams.2013.13022

Natsheh, A. N., Kettleborough, J. G., & Janson, N. B. (2009). Experimental study of controlling chaos in a DC–DC boost converter. *Chaos, Solitons & Fractals, 40*(5), 2500–2508. https://doi.org/10.1016/j.chaos.2007.10.048

Nepomuceno, E. G., & Mendes, E. M. A. M. (2017). On the analysis of pseudo-orbits of continuous chaotic nonlinear systems simulated using discretization schemes in a digital computer. *Chaos, Solitons & Fractals, 95*, 21–32. https://doi.org/10.1016/j.chaos.2016.12.002

Nešić, D., & Teel, A. R. (2004). A framework for stabilization of nonlinear sampled-data systems based on their approximate discrete-time models. *IEEE Transactions on Automatic Control, 49*(7), 1103–1122. https://doi.org/10.1109/tac.2004.831175

Ott, E., Grebogi, C., & Yorke, J. A. (1990). Controlling chaos. *Physical Review Letters, 64*(11), 1196–1199. https://doi.org/10.1103/physrevlett.64.1196

Ozcan, A. B., Bayezit, I., & Tekyurt, M. (2025). Stabilization of chaotic systems using hardware-in-the-loop validation for sliding mode control and input state linearization. *The Journal of Engineering, 2025*(1), e70127. https://doi.org/10.1049/tje2.70127

Pano-Azucena, A. D., Ovilla-Martinez, B., Tlelo-Cuautle, E., Muñoz-Pacheco, J. M., & de la Fraga, L. G. (2019). FPGA-based implementation of different families of fractional-order chaotic oscillators applying Grünwald–Letnikov method. *Communications in Nonlinear Science and Numerical Simulation, 72*, 516–527. https://doi.org/10.1016/j.cnsns.2019.01.014

Pecora, L. M., & Carroll, T. L. (1990). Synchronization in chaotic systems. *Physical Review Letters, 64*(8), 821–824. https://doi.org/10.1103/physrevlett.64.821

Pyragas, K. (1992). Continuous control of chaos by self-controlling feedback. *Physics Letters A, 170*(6), 421–428. https://doi.org/10.1016/0375-9601(92)90745-8

Rameshbabu, R., & Suresh, G. R. (2020). Design of adaptive feedback control for new 3D chaotic system and its digital implementation on FPGA. *Indian Journal of Science and Technology, 13*(20), 1977–1990. https://doi.org/10.17485/IJST/v13i20.498

Rivera-Blas, R., Rodríguez Paredes, S. A., Flores-Herrera, L. A., & Adrián Romero, I. (2019). Design and implementation of a microcontroller based active controller for the synchronization of the Petrzela chaotic system. *Computation, 7*(3), 40. https://doi.org/10.3390/computation7030040

Rodríguez-Muñoz, J. D., Tavizón-Aldama, J. D., & Tlelo-Cuautle, E. (2024). Experimental Observation of Chaotic Attractors from the FPGA implementation of 3D Chaotic Systems. *IETE Journal of Education, 65*(1), 6–14. https://doi.org/10.1080/09747338.2023.2267515

Rosenstein, M. T., Collins, J. J., & De Luca, C. J. (1993). A practical method for calculating largest Lyapunov exponents from small data sets. *Physica D: Nonlinear Phenomena, 65*(1–2), 117–134. https://doi.org/10.1016/0167-2789(93)90009-P

Rössler, O. E. (1976). An equation for continuous chaos. *Physics Letters A, 57*(5), 397–398. https://doi.org/10.1016/0375-9601(76)90101-8

Rugh, W. J., & Shamma, J. S. (2000). Research on gain scheduling. *Automatica, 36*(10), 1401–1425. https://doi.org/10.1016/S0005-1098(00)00058-3

Sambas, A., Vaidyanathan, S., Tlelo-Cuautle, E., Zhang, S., Guillén-Fernández, O., Sukono, Hidayat, Y., & Gundara, G. (2019). A novel chaotic system with two circles of equilibrium points: Multistability, electronic circuit and FPGA realization. *Electronics, 8*(11), 1211. https://doi.org/10.3390/electronics8111211

Sarıkaya, M. S., Demirsoy, M. S., & Hamida El Naser, Y. (2026). Robust local stabilization of the Moore–Spiegel chaotic oscillator via Grey Wolf Optimizer-based pole placement. *Erciyes Üniversitesi Fen Bilimleri Enstitüsü*

Fen Bilimleri Dergisi, 42(3). https://doi.org/10.65520/erciyesfen.1974950

Sarıkaya, M. S., Demirsoy, M. S., & Kutlu, M. Ç. (2025). Design of a chaotic speed-controlled mixing device and efficiency analysis in biogas. *Black Sea Journal of Engineering and Science, 8*(3), 672–679. https://doi.org/10.34248/bsengineering.1639478

Sauer, T., Grebogi, C., & Yorke, J. A. (1997). How long do numerical chaotic solutions remain valid? *Physical Review Letters, 79*(1), 59–62. https://doi.org/10.1103/physrevlett.79.59

Shampine, L. F. (1986). Some practical Runge-Kutta formulas. *Mathematics of Computation, 46*(173), 135–150. https://doi.org/10.1090/s0025-5718-1986-0815836-3

Shampine, L. F., & Reichelt, M. W. (1997). The MATLAB ODE Suite. *SIAM Journal on Scientific Computing, 18*(1), 1–22. https://doi.org/10.1137/s1064827594276424

Shi, S., Xiong, J., Liu, L., & Du, C. (2026). Analysis and application of a 3D chaotic system with flexible offset and frequency control. *Entropy, 28*(3), 260. https://doi.org/10.3390/e28030260

Song, Q., Chang, H., & Li, Y. (2020). Complex dynamics of a novel chaotic system based on an active memristor. *Electronics, 9*(3), 410. https://doi.org/10.3390/electronics9030410

Sprott, J. C. (1994). Some simple chaotic flows. *Physical Review E, 50*(2), R647–R650. https://doi.org/10.1103/physreve.50.r647

Tagne, S., Bodo, B., Ayissi Eyebe, G. F. V., & Eyebe Fouda, J. S. A. (2022). PIC micro-controller based synchronization of two fractional order jerk systems. *Scientific Reports, 12*, 14281. https://doi.org/10.1038/s41598-022-17029-x

Texas Instruments. (2006). *DAC8552: 16-bit, dual-channel, ultralow glitch, voltage output digital-to-analog converter* (Data sheet, Rev. A). https://www.ti.com/lit/gpn/dac8552

Texas Instruments. (2024). *TMS320F2837xD dual-core real-time microcontrollers* (Data sheet SPRS880P). https://www.ti.com/lit/pdf/sprs880

Tlelo-Cuautle, E., Rangel-Magdaleno, J., & de la Fraga, L. G. (2016). *Engineering applications of FPGAs: Chaotic systems, artificial neural networks, random number generators, and secure communication systems*. Springer. https://doi.org/10.1007/978-3-319-34115-6

Tuna, M., Alçın, M., Koyuncu, İ., Fidan, C. B., & Pehlivan, İ. (2019). High speed FPGA-based chaotic oscillator design. *Microprocessors and Microsystems, 66*, 72–80. https://doi.org/10.1016/j.micpro.2019.02.012

Wang, H., Sun, K., & He, S. (2015). Characteristic analysis and DSP realization of fractional-order simplified Lorenz system based on Adomian decomposition method. *International Journal of Bifurcation and Chaos, 25*(6), 1550085. https://doi.org/10.1142/S0218127415500856

Wang, Z., Liu, B., Zhang, Y., Cheng, M., Chu, K., & Xu, L. (2016). The chaotic-based control of three-port isolated bidirectional DC/DC converters for electric and hybrid vehicles. *Energies, 9*(2), 83. https://doi.org/10.3390/en9020083

Wolf, A., Swift, J. B., Swinney, H. L., & Vastano, J. A. (1985). Determining Lyapunov exponents from a time series. *Physica D: Nonlinear Phenomena, 16*(3), 285–317. https://doi.org/10.1016/0167-2789(85)90011-9

Yang, X.-B., He, Y.-G., Li, C.-L., & Liu, C.-Q. (2020). Analysis, stabilization, and DSP-based implementation of a chaotic system with nonhyperbolic equilibrium. *Complexity, 2020*, Article 6786832. https://doi.org/10.1155/2020/6786832

Yau, H.-T., Pu, Y.-C., & Li, S. C. (2011). An FPGA-based PID controller design for chaos synchronization by evolutionary programming. *Discrete Dynamics in Nature and Society, 2011*, 516031. https://doi.org/10.1155/2011/516031

Yeniçeri, R., Vardar, A., & Yalçın, M. E. (2017). Full digital implementation of a chaotic time-delay sampled-data system. In *2017 IEEE International Symposium on Circuits and Systems (ISCAS)* (pp. 1–4). IEEE. https://doi.org/10.1109/iscas.2017.8050862

Yesil, A., Gokyildirim, A., Babacan, Y., & Calgan, H. (2025). A novel memristive chaotic jerk circuit and its microcontroller-based sliding mode control. *Journal of Vibration and Control, 31*(17–18), 3845–3864. https://doi.org/10.1177/10775463241278038