RIGA TECHNICAL UNIVERSITY

Ruslans Babajans

DESIGN AND PERFORMANCE EVALUATION OF BIDIRECTIONAL ANALOG AND DISCRETE CHAOTIC OSCILLATOR SYNCHRONIZATION

Doctoral Thesis

RTU Press
Riga 2026

RIGA TECHNICAL UNIVERSITY

Faculty of Computer Science, Information Technology and Energy

Institute of Photonics, Electronics and Telecommunications

Ruslans Babajans

Doctoral Student of the Study Programme "Electronics"

DESIGN AND PERFORMANCE EVALUATION OF BIDIRECTIONAL ANALOG AND DISCRETE CHAOTIC OSCILLATOR SYNCHRONIZATION

Doctoral Thesis

Scientific supervisors:

Associate Professor Ph. D.
DENISS KOLOSOVS

Professor Dr. sc. ing.
DMITRIJS PIKUĻINS

RTU Press

RIGA 2026

Babajans, R. Design and Performance Evaluation of Bidirectional Analog and Discrete Chaotic Oscillator Synchronization. Doctoral Thesis. Riga: RTU Press, 2026. – 103 p.

Published in accordance with the decision of the RTU Promotion Council "RTU P-08" of 15 May, 2026, Minutes No. 51.

The research presented in this Doctoral Thesis was partially supported by the EU ERDF-funded project "RTU Doctoral Grants for Supporting Scientific Excellence in Smart Specialization Areas" (No. 1.1.1.8/1/24/I/007) within the framework of a doctoral grant (ID 8024).

Cover image generated using *Gemini*.

DOCTORAL THESIS PROPOSED TO RIGA TECHNICAL UNIVERSITY FOR PROMOTION TO THE SCIENTIFIC DEGREE OF DOCTOR OF SCIENCE

To be granted the scientific degree of Doctor of Science (Ph. D.), the present Doctoral Thesis has been submitted for the defence at the open meeting of RTU Promotion Council on 4 September 2026, at 11:00, at the Faculty of Electronics and Telecommunications of Riga Technical University, 12 Āzenes Street, Room 201.

OFFICIAL REVIEWERS

Associate Professor, Dr. sc. ing. Aleksandrs Ipatovs
Riga Technical University

Professor, Dr. Christos Volos
Aristotle University of Thessaloniki, Greece

Professor, Dr. Stavros Stavrinides
Democritus University of Thrace, Greece

DECLARATION OF ACADEMIC INTEGRITY

I hereby declare that the Doctoral Thesis submitted for review to Riga Technical University for promotion to the scientific degree of Doctor of Science (Ph. D.) is my own. I confirm that this Doctoral Thesis has not been submitted to any other university for promotion to a scientific degree.

Ruslans Babajans _____

Date: _____

The Doctoral Thesis has been written in English. It consists of Introduction, five chapters, Conclusions, 72 figures, and 11 tables; the total number of pages is 103, not including appendices. The Bibliography contains 51 titles.

ABSTRACT

The Doctoral Thesis “Design and Performance Evaluation of Bidirectional Analog and Discrete Chaotic Oscillator Synchronization” summarizes the research conducted for the degree of Doctor of Science (Ph. D.). This work is devoted to the analysis, modeling, and experimental implementation of analog and discrete chaotic systems, with a particular focus on synchronization and hardware realization. Chaotic oscillators, known for their sensitivity to initial conditions and complex nonlinear dynamics, offer significant potential for applications in secure communications and hardware-based security.

The research begins with the development and analysis of mathematical models of representative chaotic oscillators, including analog systems such as RC and Colpitts oscillators. Their dynamical behavior is examined through numerical simulations and validated using experimental hardware implementations. To enable efficient digital realization, discrete-time models of these systems are derived, with particular attention given to preserving key chaotic properties under finite-precision constraints.

A major contribution of this work is the design and implementation of chaotic systems on field-programmable gate array (FPGA) platforms. Fixed-point arithmetic and resource-efficient architectures are employed while maintaining the integrity of chaotic dynamics. The thesis further explores synchronization between analog and discrete chaotic systems in multiple configurations, primarily analog-discrete, discrete-analog. Both simulation and experimental results confirm that high-quality synchronization can be achieved, even in the presence of practical non-idealities.

The robustness of the proposed synchronization is evaluated by analyzing the effects of noise, parameter mismatch, and quantization errors. The findings indicate that, although such factors influence system performance, stable synchronization can still be maintained within acceptable limits. Additionally, the feasibility of chaos-based authentication is demonstrated using parameter estimation techniques, highlighting the potential of chaotic systems for physical-layer security applications.

Overall, the work establishes a unified framework for the design, implementation, and application of analog and discrete chaotic systems. The results contribute to the advancement of secure communication technologies and provide a foundation for future research in hardware-efficient and robust chaos-based systems.

ANOTĀCIJA

Promocijas darbā “Divvirzienu analogo un diskrēto haotisko oscilatoru sinhronizācijas realizācija un veiktspējas novērtēšana” apkopo pētījumus, kas veikti zinātņu doktora (*Ph. D.*) grāda iegūšanai. Šis darbs ir veltīts analogo un diskrēto haotisko sistēmu analīzei, modelēšanai un eksperimentālai realizācijai, īpašu uzmanību pievēršot sinhronizācijai un fiziskai realizācijai. Haotiskie oscilatori, kam piemīt jutība pret sākuma nosacījumiem un sarežģīta nelineāra dinamika, piedāvā ievērojamu potenciālu pielietojumiem drošās sakaru sistēmās un fiziska līmeņa drošībā.

Pētījums sākas ar haotisko oscilatoru matemātisko modeļu izstrādi un analīzi, tostarp analogo sistēmu, piemēram, RC un Colpitts oscilatoru, izpēti. To dinamiskā uzvedība tiek analizēta ar skaitliskām simulācijām un pārbaudīta, izmantojot eksperimentālas fiziskas realizācijas. Lai nodrošinātu efektīvu digitālo ieviešanu, tiek iegūti šo sistēmu diskrētā laika modeļi, īpašu uzmanību pievēršot galveno haotisko īpašību saglabāšanai ierobežotas precizitātes apstākļos.

Šī darba būtisks ieguldījums ir haotisko sistēmu izstrāde un realizācija uz lauka programmējamo vārtu masīvu (FPGA) platformām. Tiek izmantota fiksētā punkta aritmētika un resursefektīvas arhitektūras, vienlaikus saglabājot haotiskās dinamikas integritāti. Tālāk promocijas darbā tiek pētīta sinhronizācija starp analogajām un diskrētajām haotiskajām sistēmām dažādās konfigurācijās, primāri analogs–diskrēts, diskrēts–analogs. Gan simulāciju, gan eksperimentālie rezultāti apstiprina, ka augstas kvalitātes sinhronizāciju var panākt pat praktisku neideālitāšu klātbūtnē.

Ierosinātās sinhronizācijas izturība tiek novērtēta, analizējot trokšņu, parametru neatbilstības un kvantēšanas kļūdu ietekmi. Rezultāti liecina, ka, lai gan šie faktori ietekmē sistēmas veiktspēju, stabilu sinhronizāciju joprojām var uzturēt pieņemamās robežās. Turklāt, izmantojot parametru novērtēšanas metodes, tiek demonstrēta haosa balstītas autentifikācijas iespējamība, izceļot haotisko sistēmu potenciālu fiziskā slāņa drošības lietojumos.

Kopumā darbs izveido vienotu ietvaru analogo–digitālo haotisko sistēmu izstrādei, realizācijai un pielietošanai. Iegūtie rezultāti veicina drošu sakaru tehnoloģiju attīstību un nodrošina pamatu turpmākiem pētījumiem par resursefektīvām un robustām haosa balstītām sistēmām.

CONTENTS

LIST OF FIGURES

LIST OF TABLES

ACRONYMS

ADC analog-to-digital converter

ADP3450 Digilent's Analog Discovery Pro

ALM adaptive logic module

B.L.AWGN band-limited additive white Gaussian noise

BRAM block random access memory

CSK chaos shift keying

DAC digital-to-analog converter

DSP digital signal processing

FF flip-flop

FPGA field-programmable gate array

GW gateway

HiL hardware-in-the-loop

HSMC high-speed mezzanine card

IoT Internet-of-Things

LMS least mean squares

LUT lookup table

MMSE minimum mean squared error

PCB printed circuit board

PLA physical-layer authentication

PSD power spectral density

PSK phase shift keying

ROM read-only memory

SN sensor node

SNR signal-to-noise ratio

VHDL very high-speed integrated circuit (VHSIC) hardware description language

WSN wireless sensor network

LIST OF SYMBOLS

Operations

$\frac{\mathrm{d}x}{\mathrm{d}t}$	the derivative of x with respect to t
$\mathbb{E}$	mathematical expectation
$\mathscr{J}$	cost function
H	Heaviside step function
r	Pearson correlation coefficient

Variables

α	normalized parameter of the differential equation
β	normalized parameter of the differential equation
β_F	transistor's forward current gain
$\Delta\theta$	integration step
ϵ	normalized parameter of the differential equation
μ	step size
ω	characteristic frequency
ρ	oscillator's contour characteristic impedance
τ	oscillator's time constant
θ	normalized time
a	normalized parameter of the differential equation
$a_{estimated}$	estimated parameter a value
a_{true}	true parameter a value
b	normalized parameter of the differential equation
C	capacitor
c	normalized parameter of the differential equation
C_{eq}	equivalent capacitance
e	electron charge
f_0	fundamental frequency
f_{clk}	system clock frequency
i_0	current of the resistor R_0
i_B	transistor's base current
i_C	transistor's collector current
i_D	diode current
i_{L_1}	inductor L_1 current
i_S	saturation current of the diode
k	gain of the operational amplifier
k_B	Boltzmann's constant
L	inductor

n	sample
OA	operational amplifier
P	average power of the signal
R	resistor
s_n	acquired signal of the pipeline
t	time
V	supply voltage
v^*	characteristic voltage
v_0	forward voltage drop of the diode
v_{C1}	capacitor C_1 voltage
v_{C2}	capacitor C_2 voltage
v_{C3}	capacitor C_3 voltage
v_D	voltage across the diode
v_{R1}	resistance R_1 voltage
v_{RL}	resistance R_L voltage
V_{TH}	transistor's base-emitter threshold voltage
V_T	the thermal voltage of the diode
x	normalized state variable
y	normalized state variable
$y_{delayed}$	delayed y signal
z	normalized state variable

INTRODUCTION

Rationale

The rapid evolution of the Internet-of-Things (IoT) has transformed modern technological ecosystems into highly interconnected and data-driven environments. IoT systems integrate heterogeneous devices, sensors, and communication infrastructures to enable continuous monitoring, automation, and intelligent decision-making across domains such as healthcare, smart cities, and industrial automation [1]. These systems generate and exchange vast volumes of data, often in real time, creating significant opportunities for efficiency, scalability, and innovation.

However, this rapid expansion also introduces fundamental challenges. IoT architectures are inherently resource-constrained, highly distributed, and typically operate in open and potentially hostile communication environments. As a result, they are particularly vulnerable to security threats. Ensuring data confidentiality, integrity, and authenticity is therefore a critical requirement, especially in applications involving sensitive information such as medical or industrial data [2]–[4]. Conventional cryptographic approaches, while effective, may not always be suitable for IoT scenarios due to their computational complexity, energy consumption, and scalability limitations.

In this context, chaos theory offers a promising addition to secure communication and information processing. Chaotic systems are governed by deterministic nonlinear dynamics, yet exhibit properties such as sensitivity to initial conditions, broadband spectra, and pseudo-random behavior. These characteristics make them inherently suitable for generating complex, noise-like signals that are difficult to predict or reconstruct without precise system knowledge [5]. Consequently, chaos-based techniques have gained increasing attention as lightweight and hardware-efficient solutions for security in constrained environments.

The application of chaos theory has expanded significantly across engineering domains, particularly in areas requiring secure data transmission and processing. In IoT systems, chaos has been successfully applied to:

- cryptography and data encryption, where chaotic maps and oscillators generate highly unpredictable sequences for encryption of multimedia and sensor data [6]–[8];
- secure communication systems, including chaos-based modulation schemes that improve robustness against noise and interception [9]–[11];
- random number generation, where chaotic dynamics serve as entropy sources for cryptographic primitives [12]–[14].

These applications exploit key properties of chaotic signals, such as ergodicity, unpredictability, and wideband spectrum, which closely align with the requirements of secure and efficient IoT operation. Moreover, chaos-based methods often exhibit low computational complexity and high

parallelism, making them particularly suitable for hardware implementation in resource-limited environments.

From an implementation perspective, chaotic behavior can arise in a wide range of electronic systems, including analog oscillators described by nonlinear differential equations [15], discrete maps [16], switching power converters [17]–[19], phase-locked loops [20], and memristive systems [21], [22]. Each of these realizations introduces specific design trade-offs, particularly regarding the distinction between analog and digital implementations of chaos.

Analog chaotic systems, typically realized using electronic circuits, offer advantages such as large bandwidth, continuous-time operation, and low power consumption. However, they are sensitive to noise, component tolerances, and parameter drift, which can affect reproducibility and long-term stability. In contrast, discrete chaotic systems provide precise control over system parameters and initial conditions, as well as robustness to environmental variations and ease of integration with digital platforms such as microcontrollers and FPGAs. Nevertheless, digital implementations are inherently limited by finite precision and quantization effects, which may degrade chaotic properties and reduce entropy if not properly addressed.

Recent studies, including [23], [24], highlight a growing trend toward hybrid approaches that combine analog chaotic signal generation with digital processing and control. Such architectures aim to leverage the high entropy and bandwidth of analog chaos while benefiting from the flexibility, scalability, and reproducibility of digital systems. For instance, analog chaotic signals can be digitized using comparators or analog-to-digital converters and subsequently processed in digital hardware for encryption, synchronization, or communication tasks.

This duality underscores the importance of studying analog and discrete chaotic systems within a unified framework, particularly in the context of synchronization. Chaotic synchronization plays a central role in many applications, including secure communication and signal reconstruction. Existing studies predominantly focus on synchronization between either purely analog systems or purely discrete models [25]–[28]. While significant progress has been made in both domains, comparatively less attention has been given to synchronization between heterogeneous systems, such as analog and discrete chaotic oscillators.

Furthermore, prior works have addressed practical implementations of chaotic systems and synchronization on FPGAs and other digital platforms [6], [29]–[34]. However, the integration of analog and discrete chaotic systems into a unified hybrid synchronization framework remains an open and relevant research problem.

The Objective Statement and the Tasks

The **objective** of this Doctoral Thesis is to develop and evaluate a framework for synchronization between heterogeneous chaotic systems, specifically analog and discrete chaotic oscillators.

The following **tasks** are stated to achieve the previously defined objective:

- to develop mathematical models and hardware-efficient digital designs for the Vilnius, RC, and Colpitts chaotic oscillators, optimized for FPGA implementation;
- to investigate Pecora–Carroll synchronization between analog implementations and their discrete counterparts through numerical simulations, HiL setups, and full hardware integration;
- to assess the noise immunity and the time required for synchronization and de-synchronization across these hybrid systems;
- to design and implement a chaos-based authentication system utilizing a parameter estimator for the Vilnius oscillator, demonstrating the practical application of chaotic synchronization in secure communications.

The Subject and the Object of the Research

This Doctoral Thesis investigates synchronization between analog and discrete chaotic oscillators. The study follows a structured progression, beginning with the derivation of discrete-time models for the selected chaotic oscillators, then leading to full hardware deployment on an FPGA, followed by the subsequent performance evaluation of hybrid analog–discrete and discrete–analog chaotic synchronization.

The **object** of this research is the chaotic synchronization of analog and discrete oscillators, evaluated across multiple implementation levels. The object encompasses:

- discrete numerical models of the chaotic oscillators;
- drive–response synchronization configurations involving heterogeneous (analog and discrete) oscillators;
- estimation of system parameters of the drive oscillator through synchronization-based methods.

The **subject** of this research is the development of unified hybrid synchronization frameworks that enable reliable interaction between analog and discrete systems for secure authentication applications. This includes the study of:

- noise immunity of hybrid synchronization in non-ideal channels;
- convergence speed and stability of the hybrid synchronization process;
- the design and validation of a framework facilitating bidirectional interaction between analog circuits and FPGA-based models;

- the precision and error analysis of parameter estimation within the proposed physical-layer authentication scheme.

Research Methodology

This research follows a combined analytical, numerical, and experimental methodology to investigate synchronization between analog and discrete chaotic oscillators. The methodological framework is structured into several sequential stages, ensuring both theoretical rigor and practical validation.

First, mathematical models of the selected chaotic oscillators (Vilnius, RC, and Colpitts) are developed. The governing nonlinear differential equations are derived using circuit laws and normalized to enable efficient numerical processing. These models serve as the foundation for both simulation and digital implementation.

Second, numerical simulation and analysis are conducted using circuit simulation tools and MATLAB-based post-processing. Analog oscillator models are implemented in LTspice, and their dynamical behavior is analyzed through time-domain signals, phase-space attractors, and PSD. To enable digital realization, the analog oscillator models are discretized using numerical integration methods, specifically the forward Euler method, with careful consideration of stability and accuracy.

Third, digital design and implementation of the discrete chaotic oscillators are carried out on FPGA platforms. Fixed-point arithmetic is employed to ensure resource-efficient hardware realization while preserving essential chaotic properties. Dedicated architectures, such as derivative calculation pipelines and lookup table (LUT)-based nonlinear function implementations, are developed to optimize performance.

Fourth, synchronization analysis is performed using the Pecora–Carroll drive–response framework. Several synchronization configurations are investigated, primarily analog-discrete and discrete-analog. These studies are conducted across three levels:

- pure numerical simulations;
- HiL setups;
- full hardware integration involving physical circuits and FPGA implementations.

Fifth, experimental validation is conducted using fabricated PCB prototypes of the analog oscillators and FPGA-based digital systems. Measurement equipment is used to acquire real-time signals, which are then analyzed and compared with simulation results to verify consistency and accuracy.

Sixth, performance evaluation is carried out to assess synchronization quality under non-ideal conditions. Key metrics include:

- Pearson correlation coefficient for synchronization accuracy;

- noise immunity under additive disturbances;
- synchronization and de-synchronization time;
- sensitivity to parameter mismatch and quantization effects.

Finally, application-oriented validation is performed by developing a chaos-based authentication scheme. A parameter estimation approach based on adaptive filtering (least mean squares algorithm) is implemented and evaluated, demonstrating the feasibility of using chaotic synchronization in physical-layer authentication.

Overall, the methodology integrates theoretical modeling, simulation-based analysis, hardware design, and experimental validation into a unified framework, enabling a comprehensive investigation of hybrid analog-discrete chaotic systems.

Scientific Novelty and Main Results

The scientific novelty of this Thesis lies in integrating analog and discrete chaotic systems into a unified hybrid synchronization framework. This topic has received limited attention in prior research. The main results of the work include:

- for the first time, the experimental validation of stable synchronization between physical analog circuits and their discrete counterparts implemented on an FPGA;
- a comprehensive study of synchronization properties in hybrid configurations, demonstrating that a high correlation (above 0.8) can be maintained even in non-ideal environments;
- the development of a digital parameter estimator for the Vilnius chaotic oscillator, which is a hardware-efficient solution for physical-layer authentication.

In addition to the scientific contributions, this Thesis provides several important practical and methodological results.

- The development of ”Derivative Calculation Pipelines” and fixed-point models that preserve chaotic attractors despite the limitations of digital quantization.
- Establishing a methodology for comparing hybrid synchronization performance against purely discrete benchmarks using the Pearson correlation coefficient.

The Theses to be Defended

1. Chaotic synchronization between analog and discrete implementations of nonlinear oscillators is achievable in both analog-discrete and discrete-analog configurations under unidirectional Pecora-Carroll coupling, provided that time-domain alignment, the analog-digital interface satisfies sufficient sampling rate, quantization resolution, and bandwidth constraints to preserve the underlying system dynamics.

2. In the Vilnius and RC chaotic oscillators, hybrid synchronization converges more slowly than it diverges; moreover, state variables with stronger coupling to the synchronization signal attain higher synchronization quality and faster convergence.

3. Estimation of a drive chaotic oscillator's parameter that appears in the difference equation as a weighting coefficient is achievable by applying the adaptive least mean squares filter to the response oscillator corresponding parameter evaluated from state variables.

The Approbation and Practical Significance

The results presented in this Doctoral Thesis were obtained within the framework of applied research and experimental development activities carried out by the author during doctoral studies at Riga Technical University. The research evolved in close connection with ongoing academic, industrial, and applied research projects, ensuring both scientific relevance and practical applicability of the studied hybrid synchronization.

The scientific results of this Thesis have been published in peer-reviewed international journals and conference proceedings. The most significant publications directly reflecting the core contributions of this Thesis are listed below. Publications in which the author is the first author are highlighted in **bold**.

[35] R. Babajans et al. "Study on Analog and Discrete Chaos Oscillators Synchronization." en. In: *16th Chaotic Modeling and Simulation International Conference*. Ed. by C. H. Skiadas and Y. Dimotikalis. Cham: Springer Nature Switzerland, 2024, pp. 33–46. ISBN: 978-3-031-60907-7. DOI: 10.1007/978-3-031-60907-7_4

[36] R. Babajans et al. "Experimental Study on Analog and Discrete Chaos Oscillators Synchronization." In: *2023 Workshop on Microwave Theory and Technology in Wireless Communications (MTTW)*. Oct. 2023, pp. 24–28. DOI: 10.1109/MTTW59774.2023.10319995

[37] R. Babajans et al. "Synchronization of Analog-Discrete Chaotic Systems for Wireless Sensor Network Design." en. In: *Applied Sciences* 14.2 (Jan. 2024). Number: 2, p. 915. ISSN: 2076-3417. DOI: 10.3390/app14020915

[38] R. Babajans, D. Cirjulina, and D. Kolosovs. "Discrete Sequential Implementation of Chaos Oscillators for FPGA Integration." In: *2024 IEEE Workshop on Microwave Theory and*

Technology in Wireless Communications (MTTW). Riga, Latvia: IEEE, Oct. 2024, pp. 33–36. ISBN: 979-8-3315-3317-5. DOI: 10.1109/MTTW64344.2024.10742168

[39] R. Babajans, D. Cirjulina, and D. Kolosovs. "Field-Programmable Gate Array-Based Chaos Oscillator Implementation for Analog–Discrete and Discrete–Analog Chaotic Synchronization Applications." en. In: *Entropy* 27.4 (Apr. 2025). Number: 4, p. 334. ISSN: 1099-4300. DOI: 10.3390/e27040334

[40] R. Babajans et al. "Noise Performance Study of FPGA-based and Analog Chaos Oscillator Synchronization." en. In: *2025 IEEE 12th Workshop on Advances in Information, Electronic and Electrical Engineering (AIEEE)*. Vilnius, Lithuania: IEEE, 2025, pp. 1–5. DOI: 10.1109/AIEEE66149.2025.11050878

[41] D. Cirjulina et al. "Nonlinear Dynamics and Hybrid Synchronization of DC Biased Colpitts Chaotic Oscillators." en. In: *Electronics* 14.20 (Jan. 2025), p. 4005. ISSN: 2079-9292. DOI: 10.3390/electronics14204005

[42] R. Babajans et al. "Chaos-Based Dynamical Parameter Estimation for Physical Layer Authentication in Wireless IoT Networks." en. In: *Electronics* 15.4 (Jan. 2026), p. 748. ISSN: 2079-9292. DOI: 10.3390/electronics15040748

The research results of this Thesis were also presented at international scientific conferences, workshops, and symposia, providing peer feedback and dissemination within the research community. The author presented the results of this Doctoral Thesis at the following international scientific conferences:

- R. Babajans "Synchronization of Analog and Discrete Chaos Oscillators," Days of Applied Nonlinearity and Complexity (DANOC 2024), Greece, Thessaloniki, 12–14 January 2024.
- R. Babajans "FPGA-based Chaos Oscillator Parameter Estimator for Data Transfer," Days of Applied Nonlinearity and Complexity (DANOC 2026), Aristotle University of Thessaloniki, Thessaloniki, Greece, 23–25 January 2026.
- R. Babajans "Discrete Sequential Implementation of Chaos Oscillators for FPGA Integration," 2024 Workshop on Microwave Theory and Technology in Wireless Communications (MTTW 2024), IEEE Latvia Section COM/MTT/AP Joint Society Chapter, IEEE Latvia Section, Riga Technical University, Riga 2–4 October 2024.
- R. Babajans "Experimental Study on Analog and Discrete Chaos Oscillators Synchronization," 2023 Workshop on Microwave Theory and Technology in Wireless Communications (MTTW 2023), IEEE Latvia Section COM/MTT/AP Joint Society Chapter, IEEE Latvia Section, Riga Technical University, Riga, 4–6 October 2023.
- R. Babajans "Quadrature Chaos Phase Shift Keying Communication System Based on Vilnius Chaos Oscillator," 2022 Workshop on Microwave Theory and Technology in Wireless Communications (MTTW 2022), IEEE Latvia Section COM/MTT/AP Joint Society Chapter, IEEE Latvia Section, Riga Technical University, Riga, 5–7 October 2022.

- R. Babajans “Noise Performance Study of FPGA-based and Analog Chaos Oscillator Synchronization,” 2025 IEEE 12th Workshop on Advances in Information, Electronic and Electrical Engineering (AIEEE 2025), Lithuania, Vilnius, 15–17 May 2025.
- R. Babajans “Noise Performance Study of FPGA-based and Analog Chaos Oscillator Synchronization,” RTU 66th International Scientific Conference, Riga Technical University, Riga, Latvia, 27 November 2025.
- R. Babajans “FPGA-based Chaos Oscillator Implementation for Analog-Discrete and Discrete-Analog Chaotic Synchronization Applications,” RTU 66th Student Scientific and Technical Conference, Riga Technical University, Riga, Latvia, 25 April 2025.
- R. Babajans “Synchronization of Analog and Discrete Chaos Oscillators,” RTU 65th International Scientific Conference, Riga Technical University, Riga, Latvia, 11 October 2024.
- R. Babajans “Study on Analog and Discrete Chaos Oscillators Synchronization,” The 16th CHAOS 2023 International Conference, Technical University of Crete, Crete, Greece, 13–17 June 2023.

The research was conducted in parallel with participation in several national and institutional research projects, which supported the experimental validation, prototype development, and applied investigations addressed in this Thesis.

1. “Implementation of consolidation and management changes at RTU, LiepU, Rēzekne Academy of Technology and Latvian Maritime Academy and Liepāja Maritime College for progress towards excellence in higher education, science and innovation.“ Grant No RTU-PA-2024/1-0064 under the EU RRF project No. 5.2.1.1.i.0/2/24/I/CFLA/003 (01.01.2024 – 31.01.2026).
2. SAM 8.2.2. “Strengthening of Ph. D. students and academic personnel of Riga Technical University and BA School of Business and Finance in the strategic fields of specialization” of the Specific Objective 8.2.2 “To Strengthen Academic Staff of Higher Education Institutions in Strategic Specialization Areas” of the Operational Program “Growth and Employment” and the RTU doctoral grant program. Project 03000-3.1.2.1-e/177 | 8.2.2.0/20/I/008. (01.12.2022–30.11.2023).
3. LZP fundamental and applied research project “Advanced wireless power transmission techniques” 03000-3.1.2.1-e/4 | lzp-2021/1-0170 (03.01.2022–30.12.2024).
4. “Exploring the Next Generation IoT Network”, 03000-3.1.2.1-e/14 edoc | ZI-2023/3 | 4691, Internal funding of RTU (02.01.2023 – 31.12.2023).
5. “Exploring cyber-secure IoT data-driven agricultural management”, 04000-1.3-e/24 | ZI-2024/7 | 4834, Internal funding of RTU (02.01.2024 – 31.12.2024).
6. “RTU Doctoral Grants for Supporting Scientific Excellence in Smart Specialization Areas” 0B000-3.5.3-e/3 | 1.1.1.8/1/24/I/007 | 5021, within the framework of a doctoral grant ID 8024 (15.09.2025 – 15.09.2026).

During the Doctoral studies, the author has been involved in teaching the following study courses

at Riga Technical University.

- REA711 – Fundamentals of Digital Electronic Systems Design using HDL (4.5 ECTS).
- DE0092 – Simulation of Functional and Logical Circuits (4 ECTS).
- DE0735 – Signal Processing Systems (Study work) (3 ECTS).
- REA712 – Fundamentals of Digital Electronic Systems Design using HDL (study project) (3 ECTS).
- RTR822 – Signal Theory (4.5 ECTS).
- DE0101 – Signal Processing Systems (4 ECTS).
- DE0103 – Electronic Systems for Data Transmission (4 ECTS).
- DE0501 – Signal Theory (study project) (3 ECTS).

The development of this Doctoral Thesis was further supported by international academic and research mobility, enabling collaboration with external research institutions and access to advanced experimental infrastructure. During the doctoral studies, the author remained employed at Riga Technical University while completing the following mobility periods:

- KTH Royal Institute of Technology, Sweden (01.05.2023–31.05.2023);
- Politecnico di Torino, Italy (18.11.2024–22.11.2024);
- Aristotle University of Thessaloniki, Greece (30.11.2025–05.12.2025).

The scientific and academic contributions of the author during doctoral studies was recognized by the following award:

- Best Paper Award, 2023 Workshop on Microwave Theory and Technology in Wireless Communications, 06.10.2023.

Overall, the results presented in this Doctoral Thesis were validated through peer-reviewed publications, international conference presentations, experimental prototypes, and applied research projects. The combination of simulation-based analysis and experimental measurements confirms the scientific relevance and practical applicability of the hybrid analog-discrete, discrete-analog synchronization.

The Structure of the Thesis

The first chapter provides an overview of chaotic oscillators, including the Vilnius, RC, and Colpitts chaotic oscillators. Mathematical models are developed, and their dynamical properties are analyzed through numerical simulations and experimental validation.

The second chapter focuses on discrete-time modeling of chaotic oscillators and their implementation in digital hardware. Particular attention is given to FPGA-based realization, fixed-point arithmetic, and the preservation of chaotic behavior in digital form.

The third chapter investigates synchronization between analog and digital chaotic systems. Various configurations are examined, including analog-discrete, discrete-analog, with results obtained

from simulations and experimental platforms.

The fourth chapter presents a comprehensive performance analysis of the proposed systems. Key aspects such as synchronization accuracy, robustness to noise, parameter mismatch, and transient behavior are evaluated.

The fifth chapter explores the application of chaotic systems in physical-layer authentication. A stochastic gradient-based parameter estimation technique is used to demonstrate the feasibility of chaos-based physical-layer security mechanisms.

Together, these sections establish a unified framework for the analysis and implementation of chaotic systems, bridging analog and digital domains and highlighting their applicability in synchronization and physical-layer security.

1 CHAOTIC OSCILLATORS

The current chapter introduces three analog chaotic oscillators: the Vilnius, the RC and the Colpitts chaotic oscillator. For each oscillator, the circuit structure, LTspice simulation model, and experimental hardware prototype are presented and analyzed. The comparison between simulated and measured results provides insight into the practical realization of chaotic dynamics in electronic circuits. The developed LTspice models and fabricated prototypes are extensively used in the studies presented in this Thesis, with LTspice employed to model the analog oscillators and the prototypes for hardware verification.

Chaotic oscillators are nonlinear dynamical systems capable of generating complex, aperiodic signals that exhibit deterministic chaos. Such systems are characterized by strong sensitivity to initial conditions, meaning that small variations in the initial state of the system can lead to significantly different trajectories over time. As a result, chaotic signals often appear noise-like while still being governed by deterministic equations. These signals possess a wide spectrum and low cross-correlation properties.

In analog electronic circuits, chaotic behavior can be realized by constructing nonlinear dynamical systems with sufficient system order and nonlinearity. A common approach involves circuits containing at least three energy-storage (reactive) elements, such as capacitors and inductors, together with a nonlinear component and an active element that sustains oscillations. The interaction between the reactive elements defines the system dynamics, while the nonlinear component enables the transition from periodic oscillations to chaotic regimes.

Over time, several chaotic oscillators have been proposed and investigated, including circuits derived from well-known nonlinear dynamical systems such as the Rössler [43] and Lorenz [44] systems, as well as electronic oscillator-based implementations such as the Colpitts chaotic oscillator [45] and various minimal chaotic circuits including those proposed by Sprott [46]. These oscillators provide different approaches to achieving chaotic dynamics while varying in circuit complexity, component requirements, and operating frequency.

This Thesis focuses on three chaotic oscillators: Vilnius [47], RC [48] and Colpitts [45]. Although many chaotic oscillators have been introduced over the years, these particular oscillators were selected for multiple reasons. First, all of the listed oscillators can exhibit chaotic behavior when powered with low-voltage (≤ 5 V). This property, combined with the low component number, makes these oscillators suitable for embedded applications. Second, each of the listed oscillators can scale its dynamics to higher frequencies by carefully adjusting the reactive component values. This property makes chaotic oscillators more versatile for deployment in applications like data transmission. Third, the Vilnius, RC and Colpitts chaotic oscillators have key design differences. The Vilnius and RC oscillators use operational amplifiers as active elements and diodes to introduce nonlinearity, whereas the Colpitts oscillator uses a bipolar junction transistor as both an active and a nonlinear element. The Vilnius and Colpitts are based on an LC resonance tank, but RC chaotic

oscillator does not use an inductor, instead the circuit is based on an RC network. This selection of chaotic oscillators demonstrates that the methodologies developed later in the Thesis are applicable to different types of chaotic oscillators.

The chapter is organized as follows: Sections 1.1–1.3 present the Vilnius, RC, and Colpitts chaotic oscillators, respectively, including their mathematical models, LTspice simulations, and hardware implementations, followed by verification of chaotic dynamics and spectral analysis of both simulated and experimental signals.

1.1 Vilnius Chaotic Oscillator

The first oscillator selected for this Thesis is the Vilnius chaotic oscillator [47]. One of the main advantages of this oscillator is its relatively simple circuit structure and clear separation between the linear resonant network and the nonlinear element. The LC resonant network provides stable oscillatory dynamics and well-defined frequency characteristics, which simplifies tuning and analysis. Additionally, the circuit can operate at relatively low supply voltages and requires only a small number of active components, contributing to low power consumption and reliable experimental implementation.

Mathematical model

The circuit diagram of the Vilnius chaotic oscillator is presented in Fig. 1.1. The oscillator contains the fundamental building blocks required for chaotic behavior in electronic systems. Specifically, it includes three energy-storing (reactive) components — capacitors C_1 and C_2, and inductor L_1 — which provide the minimum third-order dynamics necessary for chaos. Nonlinearity is introduced by the diode D_1, whose nonlinear current–voltage characteristic enables non-periodic behavior. An operational amplifier OA, configured in a non-inverting topology, serves as the active component. It compensates for energy losses in the passive network and sustains oscillations in the circuit.

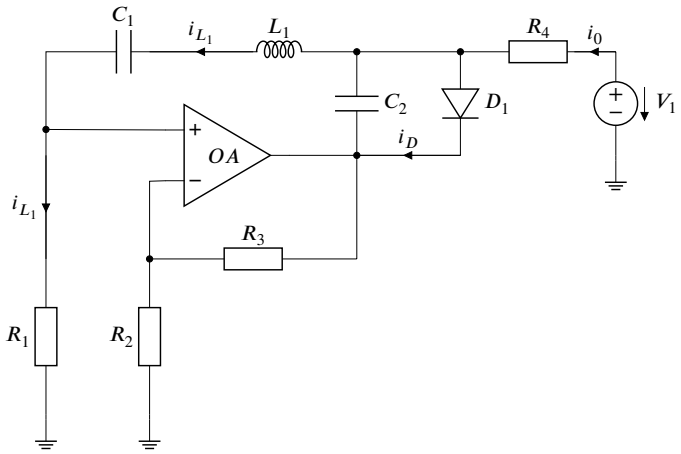


Fig. 1.1. Vilnius chaotic oscillator circuit diagram.

The oscillator dynamics is described by three state variables: the voltage across the capacitor C_1 (v_{C_1}), the current through the inductor L_1 (i_{L_1}), and the voltage across the capacitor C_2 (v_{C_2}). The set of nonlinear differential equations derived from Kirchhoff's laws of the Vilnius chaotic oscillator is:

$$\begin{cases} C_1 \dfrac{\mathrm{d}v_{C_1}}{\mathrm{d}t} = i_{L_1} \\ L_1 \dfrac{\mathrm{d}i_{L_1}}{\mathrm{d}t} = (k-1) \cdot R_1 \cdot i_{L_1} - v_{C_1} - v_{C_2} \\ C_2 \dfrac{\mathrm{d}v_{C_2}}{\mathrm{d}t} = i_0 + i_{L_1} - i_D \end{cases}, \tag{1.1}$$

where i_D is the current through diode D_1; i_0 is the current through resistor R_4; and k is the gain of the non-inverting operational amplifier circuit:

$$k = 1 + \frac{R_3}{R_2}. \tag{1.2}$$

The nonlinear behavior of the oscillator arises from the current-voltage characteristic of the diode, expressed as

$$i_D = i_S \cdot \left(\mathrm{e}^{\frac{v_{C_2}}{V_T}} - 1 \right), \tag{1.3}$$

where i_S is the diode saturation current (approximately 2×10^{-14} A for a 1N4148 diode), and V_T is the thermal voltage (approximately 26 mV at room temperature of 298 K).

Because oscillations are driven by the contour L_1, C_1, the fundamental frequency of the oscillator is defined by Thomson's formula:

$$f_0 = \frac{1}{2\pi \cdot \sqrt{L_1 \cdot C_1}}, \tag{1.4}$$

The original study [47] presented the Vilnius chaotic oscillator with the parameter values: $C_1 = 100$ nF, $C_2 = 15$ nF, $L_1 = 100$ mH, $R_1 = 1$ kΩ, $R_2 = 10$ kΩ, $R_3 = 6$ kΩ, $R_4 = 20$ kΩ, $V_1 = 5$ V and $k = 1.6$. This set of parameters produces chaotic dynamics and sets the fundamental frequency of the oscillations $f_0 = 1.6$ kHz. However, a 100 mH inductor presents several significant drawbacks in practical implementation. Such inductance value is achieved by long wire windings, introducing parasitic resistance of 100 to 300 Ω, thus reducing the quality factor. On top of that, the physical dimensions of such an inductor make it bulky and less suitable for small form-factor embedded applications. It is important that in order to scale the inductor and maintain chaotic dynamics, the values of C_1 and C_2 must be scaled accordingly. Thus, the following values were selected for the oscillator: $C_1 = 1$ nF, $C_2 = 150$ pF, $L_1 = 1$ mH. This results in a fundamental frequency $f_0 = 160$ kHz.

Simulation Model

The LTspice analog electronic circuit simulator was used to model the behavior of the Vilnius chaotic oscillator with the selected parameters. The model is presented in Fig. 1.2. The model is set to run 100 ms with a maximum step of 10 ns. This configuration allows for a sufficient signal length and resolution for further processing in MATLAB. The model is configured to save only the signals from the state variables v_{C_1}, v_{C_2} and i_{L_1}. The model also has the possibility of setting the initial conditions of the state variables.

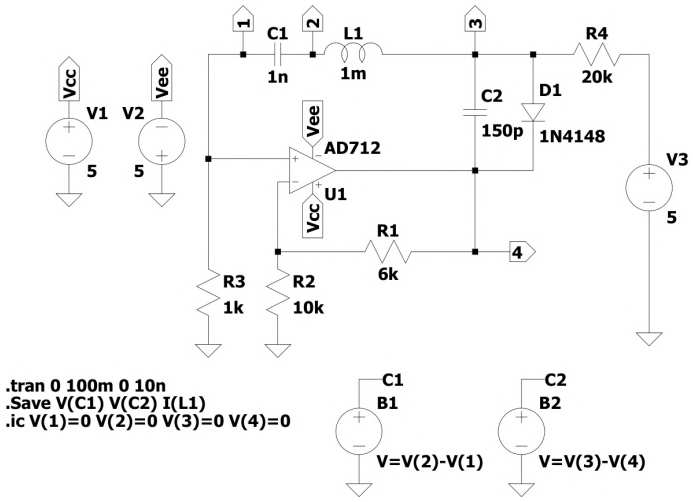
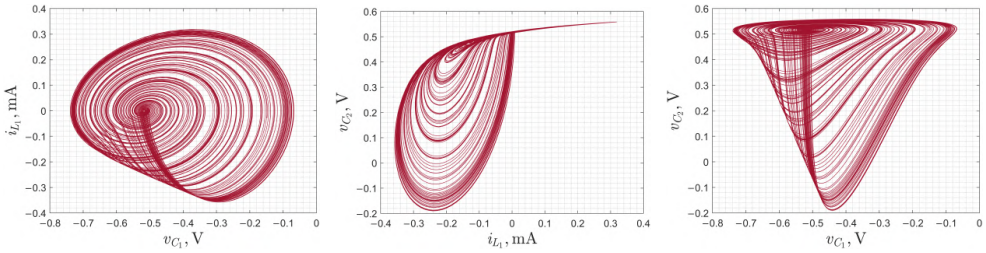


Fig. 1.2. Vilnius chaotic oscillator model in the LTspice analog electronic circuit simulator.

The time-domain simulation results obtained from LTspice were exported and further processed

in MATLAB. Due to the adaptive time-stepping algorithm used by the LTspice numerical solver, the simulation output is acquired with non-uniform time steps. Therefore, the exported data were resampled in MATLAB to obtain a uniformly spaced time series.

Fig. 1.3 demonstrates the attractor projections of the simulated Vilnius chaotic oscillator's LTspice model, confirming the presence of chaotic behavior. Unlike periodic oscillations, which produce simple closed trajectories in phase space, the observed attractor projections exhibit complex, non-repeating structures characteristic of deterministic chaos.



(a) (b) (c)

Fig. 1.3. Two-dimensional attractor projections v_{C_1} and i_{L_1} (a), i_{L_1} and v_{C_2} (b), v_{C_1} and v_{C_2} (c) of the Vilnius chaotic oscillator.

The uniformly resampled time series obtained in MATLAB enables the computation of PSD using Fourier-based methods, which require evenly spaced samples. The resulting spectra of the chaotic signals are presented in Fig. 1.4. For better comparison, the acquired spectra are normalized to their maxima and expressed in dB. The spectra provide additional confirmation of chaotic operation through their broadband characteristics, with a spectral maximum around the 160 kHz fundamental frequency.

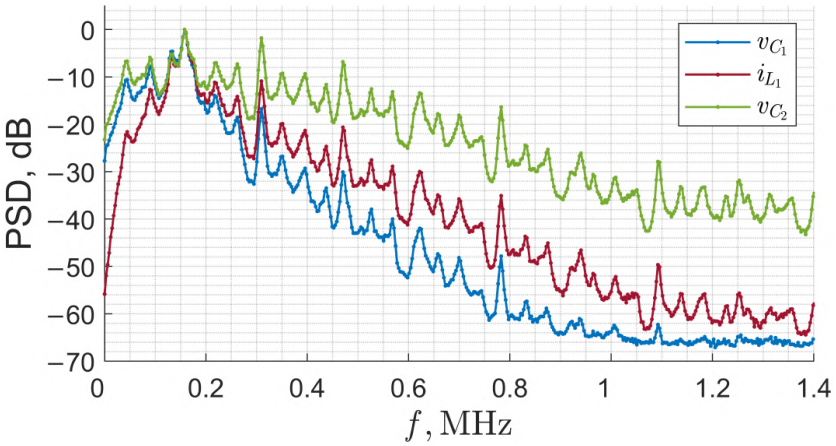


Fig. 1.4. PSD of v_{C_1}, i_{L_1} and v_{C_2} for the Vilnius chaotic oscillator.

Prototype

A hardware prototype of the Vilnius chaotic oscillator was developed to experimentally validate the Vilnius oscillator's chaotic behavior. The prototype PCB was designed using KiCad PCB design software and fabricated on a two-layer FR4. The fabricated PCB with the designed layout is presented in Fig. 1.5.

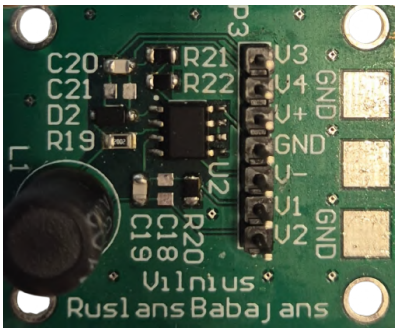
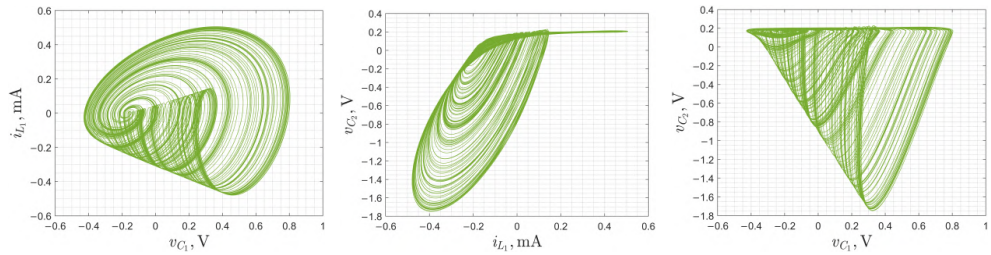


Fig. 1.5. Fabricated Vilnius chaotic oscillator PCB prototype.

The signals of the Vilnius chaotic oscillator were recorded using the Digilent's Analog Discovery Pro (ADP3450). It is important to note that the i_{L_1} is acquired by measuring the voltage of the R_1 resistor, and acquiring the current using the Ohm's law. This is possible due to the general assumption that there is no input current in the operational amplifier OA, which is highlighted in Fig. 1.1.

The recorded signals are then exported to MATLAB for further processing. The ADP3450 provides uniformly sampled time series; however, the data was resampled to match the resolution of the signals acquired in LTspice. Fig. 1.6 shows attractor projections of the experimental signals acquired with the same oscillator parameters. It is apparent that the circuit executes a chaotic behavior similar to the LTspice model, highlighted by the similarity of attractors in Fig. 1.6 and 1.3.



(a) (b) (c)

Fig. 1.6. Two-dimensional attractor projections v_{C_1} and i_{L_1} (a), i_{L_1} and v_{C_2} (b), v_{C_1} and v_{C_2} (c) of the Vilnius chaotic oscillator PCB.

Resampling the experimental signals, results in the same resolution of the signal spectrum, as demonstrated in Fig. 1.7. The broadband spectrum has the maximum around the fundamental frequency 160 kHz, like in the LTspice model. Overall, the hardware prototype demonstrates a similar

chaotic behavior, with only amplitude and offset differences from the LTspice model. These differences are attributed to the simplified LTspice model of the operational amplifier, the parasitic parameters of the physical elements, etc.

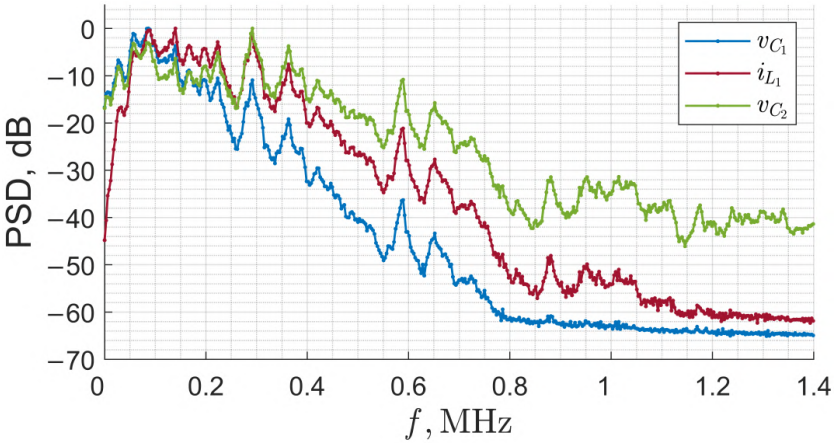


Fig. 1.7. PSD of v_{C_1}, i_{L_1} and v_{C_2} for the Vilnius chaotic oscillator prototype.

By constructing an LTspice model of the Vilnius chaotic oscillator and fabricating its hardware prototype, it was observed that both implementations exhibit qualitatively similar dynamical behavior. The agreement between simulated and experimental results is further supported by the analysis of the acquired PSD of the chaotic signals, which reveals well-defined spectral boundaries. These spectral characteristics are essential for subsequent stages of the Thesis and are therefore taken into account.

For the studies presented in this Thesis, the developed LTspice model is primarily used to model the behavior of the analog oscillator. This modeling stage serves as a necessary step prior to experimental validation, enabling efficient evaluation before using the fabricated hardware prototype for verification.

1.2 RC Chaotic Oscillator

The second chaotic oscillator for this Thesis is the "RC chaotic oscillator" [48]. Since the circuit relies exclusively on resistors, capacitors, operational amplifiers, and a diode, the circuit can be implemented using compact and inexpensive components. The absence of inductors significantly reduces the circuit size and eliminates issues related to magnetic coupling, parasitic inductance, and component variability. Furthermore, the oscillator operates reliably with low supply voltages, making it suitable for low-power analog systems. These properties also make the RC oscillator convenient for integrated circuit implementations.

Mathematical model

The circuit diagram of the RC chaotic oscillator is presented in Fig. 1.8. The oscillator is based on a Wien bridge configuration and incorporates the essential components required for chaotic dynamics in electronic systems. The circuit contains three reactive elements — capacitors C_1, C_2, and C_3. Capacitors C_1 and C_2 form the Wien bridge frequency-selective network, while the capacitor C_3 is introduced in the feedback loop of the first operational amplifier. Together, these three energy-storage elements provide the minimum third-order dynamics necessary for chaos generation. The nonlinearity is introduced by the diode D, which is placed in the feedback loop of the second operational amplifier. Through the action of the negative impedance converter, activated at higher voltages, the diode produces a piecewise-linear characteristic that governs the complex amplitude dynamics of the oscillator. Two operational amplifiers, OA_1 and OA_2, serve as the active components of the circuit. The first amplifier maintains the oscillations of the Wien bridge, while the second forms the negative impedance converter that compensates for losses and shapes the nonlinear response. These active elements supply energy to the system and ensure sustained oscillations.

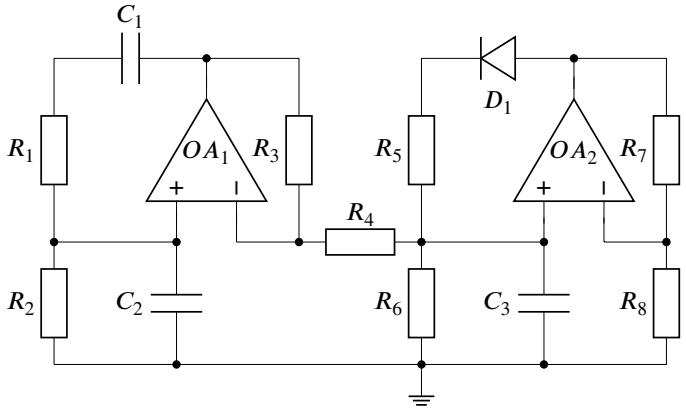


Fig. 1.8. RC chaotic oscillator circuit diagram.

The mathematical model of the RC chaotic oscillator is derived from Kirchhoff's current law applied to each capacitor node:

$$\begin{cases} C\frac{\mathrm{d}v_{C_1}}{\mathrm{d}t} = -\frac{v_{C_1}}{R} + \frac{k_1-1}{R}v_{C_2} - \frac{k_1-1}{R}v_{C_3} \\ C\frac{\mathrm{d}v_{C_2}}{\mathrm{d}t} = -\frac{v_{C_1}}{R} + \frac{k_1-2}{R}v_{C_2} - \frac{k_1-1}{R}v_{C_3} \\ C\frac{\mathrm{d}v_{C_3}}{\mathrm{d}t} = \frac{k_1}{R}v_{C_2} - \frac{k_1+\alpha}{R}v_{C_3} + \frac{\beta}{R}(v_{C_3} - v^*)H(v_{C_3} - v^*) \end{cases} . \tag{1.5}$$

Unlike passive RC networks, the circuit includes operational amplifiers that introduce additional feedback paths, resulting in coupling between all state variables. These active elements effectively implemeleading to coupling amongages, which are reflected in the coefficients of the governing

equations. As a result, the system cannot be described as a simple RC ladder. Instead, it represents an active RC network where the parameters k_1, α and β correspond to resistor-defined gains within the feedback structure. The nonlinear term arises from the use of a diode. The characteristic voltage v^* is expressed with the forward voltage drop of the diode v_0 (typically 0.7 V).

$$\begin{gathered} k_1 = \frac{R_3}{R_4} + 1, \qquad k_2 = \frac{R_7}{R_8} + 1, \qquad \alpha = \frac{R_1}{R_6}, \qquad \beta = \frac{R_1}{R_8}, \\ v^* = \frac{v_0}{k_2 - 1}, \quad C_1 = C_2 = C_3 = C, \quad R_1 = R_2 = \left(R_3 + R_4\right) = R. \end{gathered} \tag{1.6}$$

The nonlinearity in this circuit is introduced by the Heaviside step function:

$$H = \begin{cases} 0, & \text{if } v_{C_3} - v^* < 0 \\ 1, & \text{if } v_{C_3} - v^* \geq 0 \end{cases}. \tag{1.7}$$

For the RC chaotic oscillator, the fundamental frequency is determined by the Wien bridge network composed of R_1,R_2,C_1, and C_2. Unlike LC contour-based oscillators, this frequency is not associated with energy exchange between magnetic and electric fields, but rather with the phase-shift condition of the RC network. The fundamental oscillation frequency of the oscillator is:

$$f_0 = \frac{1}{2\pi \cdot C \cdot R}. \tag{1.8}$$

The original study [48] introduced the circuit with the following parameters: $C_1 = C_2 = C_3 =$ 1.3 nF, $R_1 = R_2 = 11$ kΩ, $R_3 = 9.1$ kΩ, $R_4 = 2$ kΩ, $R_5 = R_7 = 2.7$ kΩ, $R_6 = 1.1$ kΩ, $R_8 = 780$ Ω, the diode is 1N4148. The given parameters result in the fundamental frequency $f_0 = 11$ kHz.

Simulation Model

Like in the case of Vilnius chaotic oscillator, the LTspice analog electronic circuit simulator was used to model the behavior of the RC chaotic oscillator. The model is presented in Fig. 1.9. The model is set to run 1.43 s with a maximum step of 1 μs. The model is configured to save only the signals from the state variables v_{C_1}, v_{C_2} and v_{C_3}.

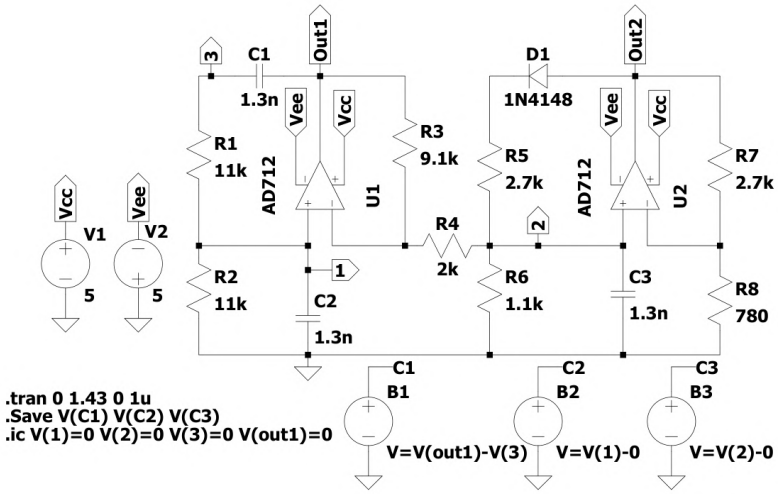
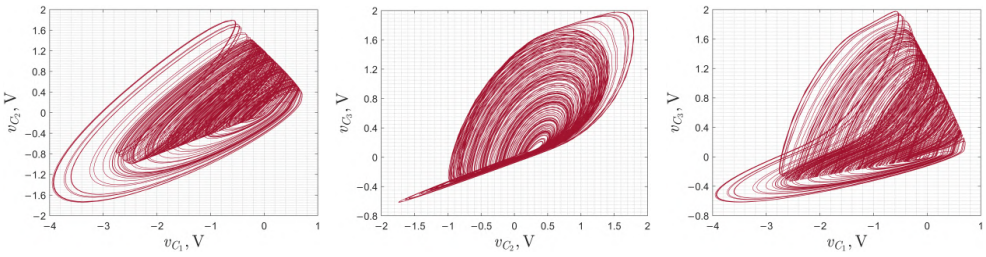


Fig. 1.9. RC chaotic oscillator model in the LTspice analog electronic circuit simulator.

The time-domain simulation results are then exported to MATLAB, where the data is resampled to a uniform time step, and the Fourier-based spectrum is acquired. Fig. 1.10 shows the attractor projections of the RC chaotic oscillator's LTspice model. The figure demonstrates complex, non-periodic trajectories characteristic of chaotic dynamics.



(a) (b) (c)

Fig. 1.10. Two-dimensional attractor projections v_{C_1} and v_{C_2} (a), v_{C_2} and v_{C_3} (b), v_{C_1} and v_{C_3} (c) of the RC chaotic oscillator.

The PSD of the RC chaotic oscillator's LTspice model is shown in Fig. 1.11. Although the fundamental frequency of the oscillator is 11 kHz, this value does not correspond to a dominant spectral component in the chaotic regime. The aperiodic behavior of the oscillator produces a broadband spectrum, therefore the f_0 represents a characteristic time scale of the system rather than a distinct oscillation frequency. The energy of the chaotic signal is distributed over a wide frequency range, and the resulting spectrum exhibits multiple local maxima instead of a single dominant peak.

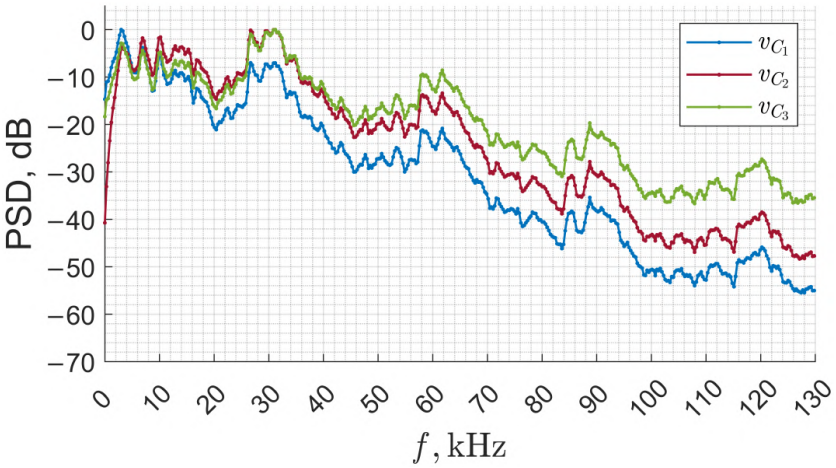


Fig. 1.11. PSD of v_{C_1}, v_{C_2} and v_{C_3} for the RC chaotic oscillator.

Prototype

A hardware prototype of the RC chaotic oscillator was developed to experimentally validate the chaotic behavior predicted by the LTspice model. The prototype PCB was designed using the KiCad and fabricated on a two-layer FR4 board. The fabricated PCB with the designed layout is presented in Fig. 1.12. The fabricated prototype implements R_8 using a trim resistor originally intended for exploring the chaotic regimes of the oscillator.

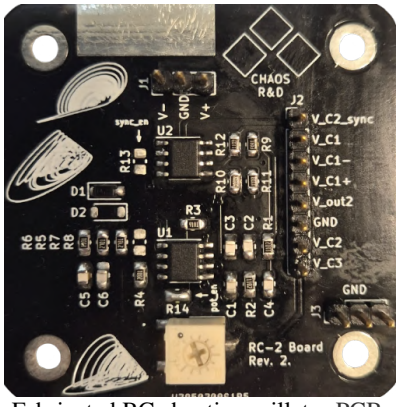
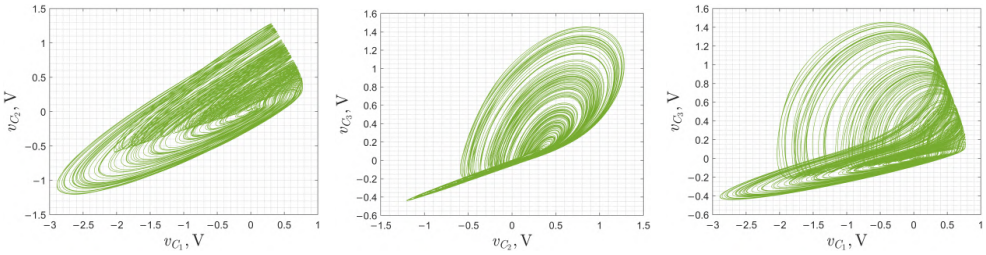


Fig. 1.12. Fabricated RC chaotic oscillator PCB prototype.

The signals of the RC chaotic oscillator were recorded using the ADP3450. The output voltages of the oscillator were acquired directly from the circuit nodes indicated in Fig. 1.8. The recorded signals were exported to MATLAB for further processing, where the data were resampled to match the resolution of the signal acquired from the LTspice simulation. Fig. 1.13 presents the attractor projections of the experimental signals acquired under the same oscillator parameters. The reconstructed attractors demonstrate good qualitative agreement with the LTspice results presented in Fig. 1.10.



(a) (b) (c)

Fig. 1.13. Two-dimensional attractor projections v_{C_1} and v_{C_2} (a), v_{C_2} and v_{C_3} (b), v_{C_1} and v_{C_3} (c) of the RC chaotic oscillator PCB.

The acquired PSD of the chaotic signals are presented in Fig. 1.14. The spectra demonstrate a broadband distribution of the spectral components with no distinct maximum at 14 kHz. Overall, the behavior of this oscillator also closely matches the LTspice model. The discrepancies can be attributed to the simplified device models used in LTspice, as well as to parasitic parameters and tolerances of the physical components in the experimental circuit.

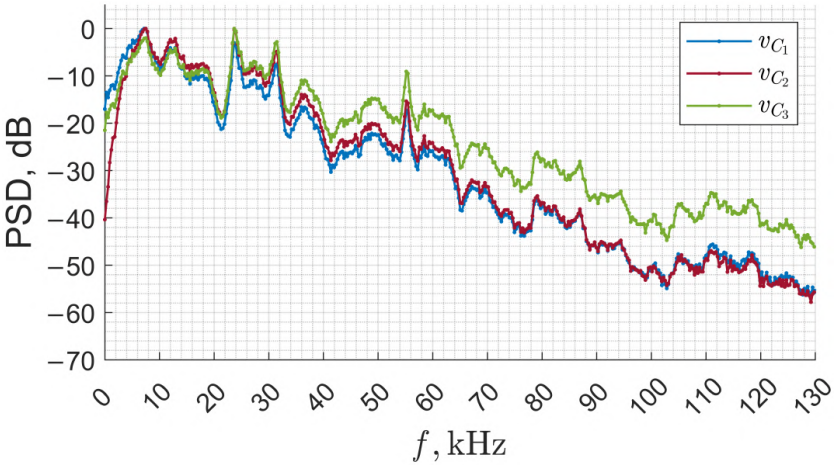


Fig. 1.14. PSD of v_{C_1}, v_{C_2} and v_{C_3} for the RC chaotic oscillator prototype.

The developed LTspice model and the fabricated prototype of the RC chaotic oscillator demonstrate good agreement, as evidenced by the similarity of the attractor projections and the corresponding PSD of the signals. An important observation is that, in this oscillator, the spectrum does not exhibit a dominant component centered at the fundamental frequency. Instead, the spectral energy is distributed over a wide frequency range, which is a characteristic feature of the chaotic dynamics observed in both the model and the hardware implementation.

For the studies presented in this Thesis, the developed LTspice model is employed to model the behavior of the analog oscillator. This modeling stage is a necessary step prior to experimental validation.

1.3 Colpitts Chaotic Oscillator

The third oscillator used in this Thesis is the Colpitts chaotic oscillator [45]. A key advantage of this oscillator is its high-frequency capability and efficient energy transfer within the LC tank circuit. Compared with op-amp-based chaotic oscillators, transistor-based Colpitts oscillator can achieve higher operating frequencies and improved energy efficiency. The circuit also requires a relatively small number of components and can operate from low supply voltages, making it suitable for compact hardware implementations and potential RF or communication-related applications.

Mathematical model

The circuit diagram of the Colpitts chaotic oscillator is shown in Fig. 1.15. The circuit incorporates the essential components required for the generation of chaotic dynamics in an analog circuit. It contains three reactive elements — capacitors C_1 and C_2, and an inductor L_1 — which form a third-order resonant network and provide the minimum number of energy storage elements necessary for chaos. Nonlinearity is introduced by the active device — a bipolar junction transistor, whose nonlinear current–voltage and current–gain characteristics govern the complex interaction between the reactive components. The transistor also simultaneously serves as the active element of the circuit, providing energy to compensate for losses in the resonant tank and sustaining oscillations.

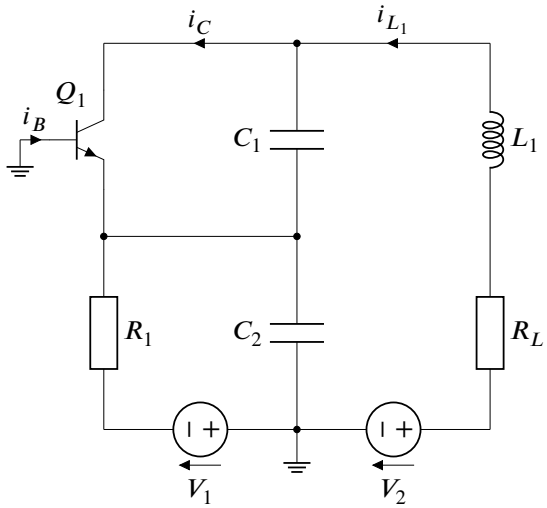


Fig. 1.15. Colpitts chaotic oscillator circuit diagram.

Applying Kirchhoff's laws to this circuit yields the following set of nonlinear differential equations:

$$\begin{cases} C_1 \dfrac{\mathrm{d}v_{C_1}}{\mathrm{d}t} = i_{L_1} - i_C \\ C_2 \dfrac{\mathrm{d}v_{C_2}}{\mathrm{d}t} = \dfrac{V_1 - v_{C_2}}{R_1} - i_{L_1} - i_B \\ L_1 \dfrac{\mathrm{d}i_{L_1}}{\mathrm{d}t} = V_2 - v_{C_1} - v_{C_2} - i_{L_1} \cdot R_L \end{cases} . \tag{1.9}$$

In these equations, i_B and i_C are nonlinear functions describing the transistor's operation. The transistor Q_1 alternates between the forward-active region and cutoff. A piecewise-linear approximation for the transistor currents is given by:

$$i_B = \begin{cases} 0, & \text{if } -v_{C_2} \leq V_{TH} \\ -\dfrac{v_{C_2} + V_{TH}}{R_{ON}}, & \text{if } -v_{C_2} > V_{TH} \end{cases} , \tag{1.10}$$

$$i_C = \beta_F \cdot i_B, \tag{1.11}$$

where V_{TH} is the base-emitter threshold voltage, R_{ON} is the small-signal on-resistance of the base-emitter junction, and β_F is the transistor's forward current gain.

Similarly, the Colpitts oscillator derives its fundamental frequency from the LC tank composed of the inductor L_1 and the series combination of capacitors C_1 and C_2:

$$f_0 = \frac{1}{2\pi \cdot \sqrt{L_1 \cdot C_{eq}}}. \tag{1.12}$$

The equivalent capacitance is:

$$C_{eq} = \frac{C_1 \cdot C_2}{C_1 + C_2}. \tag{1.13}$$

The original study [48] introduced the circuit with the following parameters: $V_1 = 5\,\mathrm{V}$, $V_2 = 5\,\mathrm{V}$, $R_L = 35\,\Omega$, $L_1 = 98.5\,\mu\mathrm{H}$, $C_1 = 54\,\mathrm{nF}$, $C_2 = 54\,\mathrm{nF}$, and $R_1 = 400\,\Omega$, using a transistor such as the 2N2222. However, for component availability reasons, the 100 µH inductor was selected, as well as R_L set to $40\,\Omega$. With the given parameters, the circuit oscillates with a fundamental frequency of approximately $f_0 = 96.8$ kHz.

Simulation Model

The LTspice model of the Colpitts chaotic oscillator with the selected parameters is presented in Fig. 1.16. The simulation was configured to run 100 ms with a maximum time step 10 ns. Because the dynamics of the circuit evolves into chaotic mode after 5 ms in the simulation, this time was set as the time to start saving the simulation results.

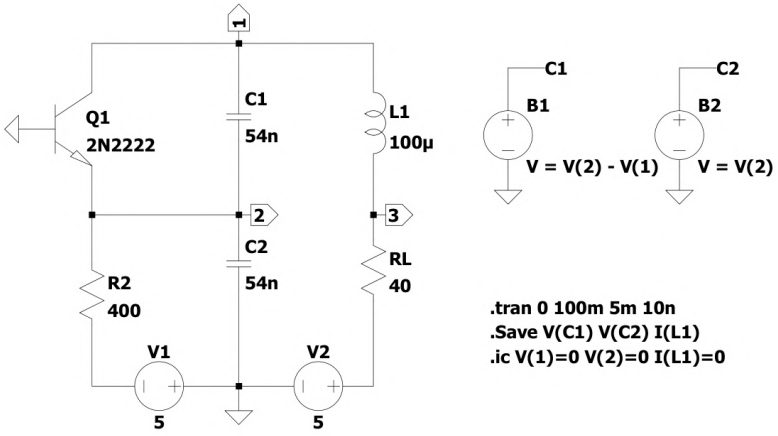
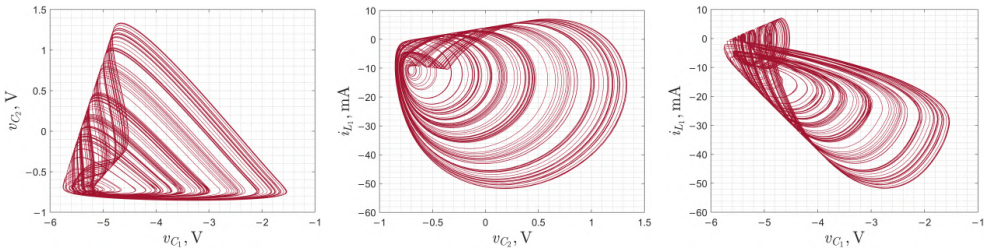


Fig. 1.16. Colpitts chaotic oscillator model in the LTspice analog electronic circuit simulator.

Like with the other oscillators, the signals were exported to MATLAB and resampled to a uniform step. Fig. 1.17 shows the attractor projections of the Colpitts chaotic oscillator acquired from the LTspice simulation. The figure demonstrates chaotic dynamic of the oscillator similar to one reported in the [45].



(a) (b) (c)

Fig. 1.17. Two-dimensional attractor projections v_{C_1} and v_{C_2} (a), v_{C_2} and i_{L_1} (b), v_{C_1} and i_{L_1} (c) of the Colpitts chaotic oscillator.

The PSD of the Colpitts chaotic oscillator signals from LTspice are demonstrated in Fig. 1.18. It is visible that the spectrum is broadband, with multiple distinct peaks, but the primary energy concentration is close to the 96.8 kHz fundamental frequency.

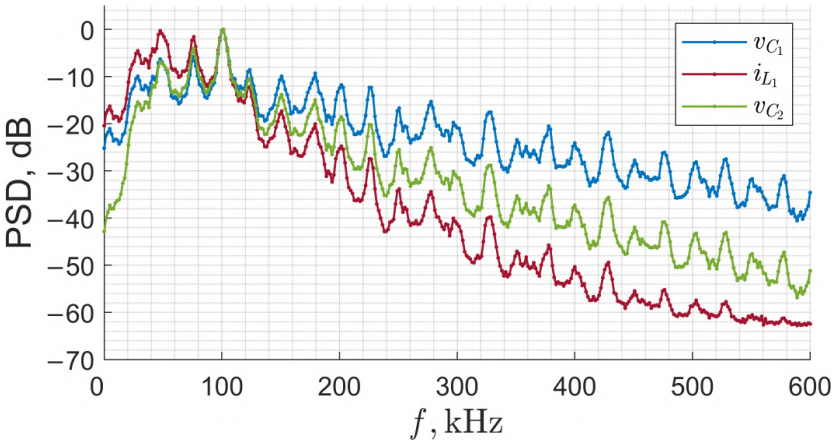


Fig. 1.18. PSD of v_{C_1}, i_{L_1} and v_{C_2} for the Colpitts chaotic oscillator.

Prototype

The fabricated hardware prototype of the Colpitts chaotic oscillator is presented in Fig. 1.19. The trim resistors in the prototype are introduced in place of R_1 and R_L. The original intention was to explore the chaotic dynamic of the oscillator by adjusting these resistors.

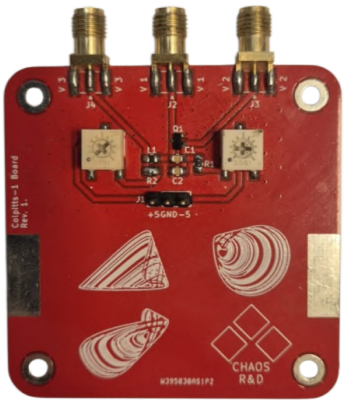
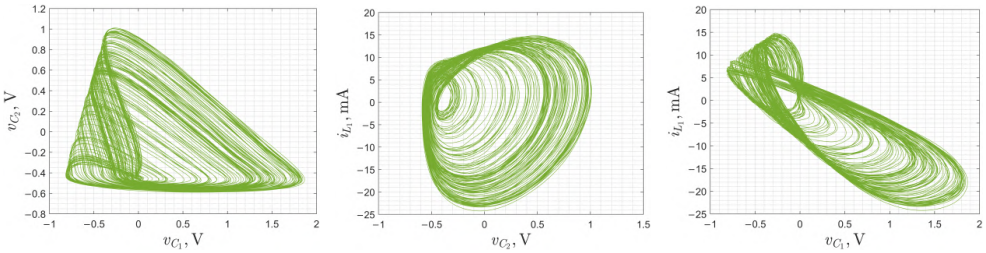


Fig. 1.19. Fabricated Colpitts chaotic oscillator PCB prototype.

The signals of the Colpitts chaotic oscillator were recorded using the ADP3450. The output voltages of the oscillator were acquired from the circuit nodes indicated in Fig. 1.15. These signals correspond to the observable variables of the system and allow reconstruction of the attractor projections that characterize the oscillator dynamics. It is important to note that the current i_{L_1} was acquired as the voltage drop across the resistor R_L. The attractor projections of the Colpitts oscillator are demonstrated in Fig. 1.20. The attractor of the hardware prototype coincides well with the LTspice's attractor.



(a) (b) (c)

Fig. 1.20. Two-dimensional attractor projections v_{C_1} and v_{C_2} (a), v_{C_2} and i_{L_1} (b), v_{C_1} and i_{L_1} (c) of the Colpitts chaotic oscillator PCB.

The acquired PSD of the chaotic signals are presented in Fig. 1.21. The spectra of the hardware signals also coincide with the LTspice signals.

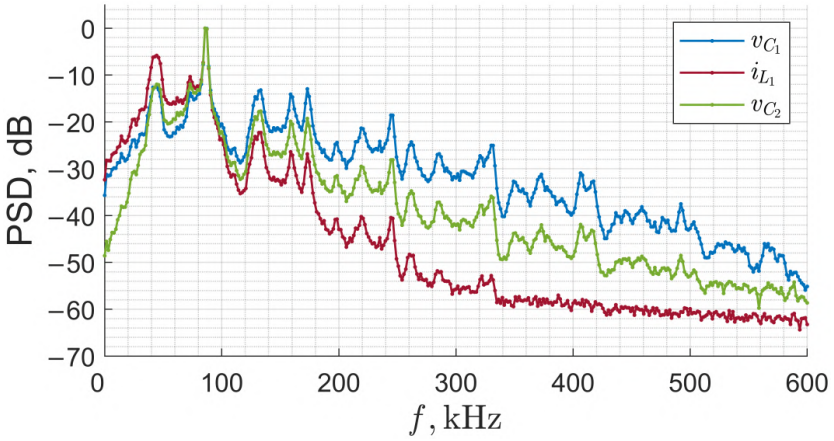


Fig. 1.21. PSD of v_{C_1}, i_{L_1} and v_{C_2} for the Colpitts chaotic oscillator prototype.

In the case of the Colpitts chaotic oscillator, the LTspice model and the fabricated prototype exhibit good agreement, as demonstrated by the close correspondence of the attractor projections and the associated PSD of the generated signals. Like with the Vilnius and RC chaos oscillators, the LTspice model is employed in the further studies of this Thesis to model the analog oscillator prior to hardware verification using the fabricated prototype.

2 DISCRETE CHAOTIC OSCILLATOR MODELS AND DIGITAL IMPLEMENTATION

This chapter presents a systematic procedure for transforming the mathematical models of the considered analog chaotic oscillators into their discrete-time counterparts. For each oscillator described in Section 1, a step-by-step derivation of the corresponding discrete model is provided, followed by its implementation on an FPGA platform.

The dynamics of chaotic oscillators are governed by nonlinear differential equations. While such equations are typically solved numerically on general-purpose computers, they can also be implemented on dedicated hardware platforms such as digital signal processors (DSP) and FPGA. This enables the digital realization of discrete-time chaotic oscillators suitable for real-time operation. These implementations can be used both for validating analog designs during early development stages and as standalone systems in applications such as secure communication, encryption, and pseudo-random sequence generation.

In the discrete domain, continuous-time dynamics are approximated using numerical integration methods, resulting in discrete-time systems that reproduce the qualitative behavior of their analog counterparts. Unlike analog circuits, digital implementations are not affected by temperature variations or component tolerances, and their parameters and initial conditions can be precisely controlled. However, they are subject to limitations imposed by finite word length, quantization effects, and sampling resolution. Furthermore, the characteristic time scale of a discrete oscillator is determined by the system clock rather than by a physical resonant frequency.

A key step in this process is the normalization of the governing equations. Dimensional normalization expresses the system in a dimensionless form, which reduces the dynamic range of variables, mitigates the risk of numerical overflow in fixed-point arithmetic, and facilitates efficient FPGA implementation. Additionally, it enables a consistent framework for comparing different oscillator models.

The section is organized as follows: Section 2.1 presents normalized equations of Vilnius, RC and Colpitts chaotic oscillators, highlighting the considerations for each case. Section 2.2 introduces the forward Euler numerical integration method to obtain the numeric solution to the normalized differential equations, resulting in difference equations suitable for digital implementation. Section 2.3 presents the digital design architecture. Section 2.4 demonstrates the implementation of the digital design on an FPGA.

2.1 Normalized Equations

The first step toward FPGA implementation of the chaotic oscillator models is the normalization of the differential equations. This process rescales the state variables and system parameters to dimen-

sionless quantities with well-defined amplitude ranges. In addition, the time variable is normalized with respect to a characteristic time constant of the system, resulting in a dimensionless time representation. This transformation simplifies the equations and removes explicit dependence on physical units, making the models more suitable for digital realization.

This normalization is essential for implementation on FPGA hardware, as it ensures that all signals remain within a bounded and predictable range compatible with the finite word length of the system. It also improves numerical stability, reduces the risk of overflow, and facilitates efficient utilization of hardware resources. Moreover, the use of dimensionless time allows straightforward control of the effective system dynamics through the selection of the discrete integration step size.

Vilnius Chaotic Oscillator

The original set of nonlinear differential equations of the Vilnius chaotic oscillator is given by:

$$\begin{cases} C_1 \dfrac{\mathrm{d}v_{C_1}}{\mathrm{d}t} = i_{L_1} \\ L_1 \dfrac{\mathrm{d}i_{L_1}}{\mathrm{d}t} = (k-1) \cdot R_1 \cdot i_{L_1} - v_{C_1} - v_{C_2} \\ C_2 \dfrac{\mathrm{d}v_{C_2}}{\mathrm{d}t} = i_0 + i_{L_1} - i_S \cdot \left(\mathrm{e}^{\frac{v_{C_2}}{V_T}} - 1 \right) \end{cases} . \tag{2.1}$$

In [47], a normalized (dimensionless) form of the system is already introduced. First, the state variables are scaled by the thermal voltage V_T. Additional dimensionless parameters are defined to eliminate dependence on the absolute values of C_1, C_2, and L_1, as given in (2.2).

$$\begin{aligned} x = \tfrac{v_{C_1}}{V_T}, \qquad & y = \tfrac{\rho \cdot i_{L_1}}{V_T}, \quad z = \tfrac{v_{C_2}}{V_T}, \\ V_T = \tfrac{k_B T}{e}, \qquad & \rho = \sqrt{\tfrac{L_1}{C_1}}, \quad \varepsilon = \tfrac{C_2}{C_1}, \\ a = (k-1)\tfrac{R_1}{\rho}, \quad & b = \tfrac{\rho \cdot i_0}{V_T}, \quad c = \tfrac{\rho \cdot i_S}{V_T}. \end{aligned} \tag{2.2}$$

Using these substitutions, the system can be expressed in normalized form:

$$\begin{cases} \frac{\mathrm{d}x}{\mathrm{d}t} = y \\ \frac{\mathrm{d}y}{\mathrm{d}t} = a \cdot y - x - z \\ \varepsilon \cdot \frac{\mathrm{d}z}{\mathrm{d}t} = b + y - c \cdot (\mathrm{e}^z - 1) \end{cases} . \tag{2.3}$$

Next, the time variable is normalized by introducing a dimensionless time θ, which is the time t scaled by the time constant τ (related to the fundamental frequency), resulting in (2.5).

$$\theta = \tfrac{t}{\tau}, \quad \tau = \sqrt{L_1 \cdot C_1}. \tag{2.4}$$

This transformation yields the final dimensionless system:

$$\begin{cases} \frac{\mathrm{d}x}{\mathrm{d}\theta} = y \\ \frac{\mathrm{d}y}{\mathrm{d}\theta} = a \cdot y - x - z \\ \varepsilon \cdot \frac{\mathrm{d}z}{\mathrm{d}\theta} = b + y - c \cdot (\mathrm{e}^z - 1) \end{cases} . \tag{2.5}$$

The resulting system is fully expressed in dimensionless variables and time, making it independent of physical units. This normalized representation is well suited for digital implementation, as it ensures bounded signal ranges and enables further realization of the Vilnius chaotic oscillator on an FPGA.

RC Chaotic Oscillator

In the case of the RC chaotic oscillator, the original set of nonlinear differential equations is given by:

$$\begin{cases} C\frac{\mathrm{d}v_{C_1}}{\mathrm{d}t} = -\frac{v_{C_1}}{R} + \frac{k_1-1}{R}v_{C_2} - \frac{k_1-1}{R}v_{C_3} \\ C\frac{\mathrm{d}v_{C_2}}{\mathrm{d}t} = -\frac{v_{C_1}}{R} + \frac{k_1-2}{R}v_{C_2} - \frac{k_1-1}{R}v_{C_3} \\ C\frac{\mathrm{d}v_{C_3}}{\mathrm{d}t} = \frac{k_1}{R}v_{C_2} - \frac{k_1+\alpha}{R}v_{C_3} + \frac{\beta}{R}(v_{C_3} - v^*)H(v_{C_3} - v^*) \end{cases} . \tag{2.6}$$

In the work [48], to obtain a dimensionless representation, the state variables are first normalized using a characteristic voltage v^*:

$$x = \frac{v_{C_1}}{v^*}, \quad y = \frac{v_{C_2}}{v^*}, \quad z = \frac{v_{C_3}}{v^*}. \tag{2.7}$$

Using these substitutions, the system can be rewritten in normalized form:

$$\begin{cases} \frac{\mathrm{d}x}{\mathrm{d}t} = -x + \left(k_1 - 1\right) \cdot y - \left(k_1 - 1\right) \cdot z \\ \frac{\mathrm{d}y}{\mathrm{d}t} = -x + \left(k_1 - 2\right) \cdot y - \left(k_1 - 1\right) \cdot z \\ \frac{\mathrm{d}z}{\mathrm{d}t} = k_1 \cdot y - \left(k_1 + \alpha\right) \cdot z + \beta \cdot (z - 1) \cdot H\left(z - 1\right) \end{cases} . \tag{2.8}$$

Next, the time variable is normalized by introducing the dimensionless time θ , defined using the characteristic time constant τ:

$$\theta = \frac{t}{\tau}, \quad \tau = R \cdot C. \tag{2.9}$$

This results in the final dimensionless system:

$$\begin{cases} \frac{\mathrm{d}x}{\mathrm{d}\theta} = -x + \left(k_1 - 1\right) \cdot y - \left(k_1 - 1\right) \cdot z \\ \frac{\mathrm{d}y}{\mathrm{d}\theta} = -x + \left(k_1 - 2\right) \cdot y - \left(k_1 - 1\right) \cdot z \\ \frac{\mathrm{d}z}{\mathrm{d}\theta} = k_1 \cdot y - \left(k_1 + \alpha\right) \cdot z + \beta \cdot (z-1) \cdot H\,(z-1) \end{cases} . \tag{2.10}$$

The Heaviside step function is transformed to:

$$H = \begin{cases} 0, & \text{if } z - 1 < 0 \\ 1, & \text{if } z - 1 \geq 0 \end{cases} \tag{2.11}$$

With the applied transformation, the resulting system is expressed in dimensionless variables and time. Like in the case of Vilnius chaotic oscillator, the differential equations of the RC chaotic oscillator in normalized representation are well suited for further digital implementation.

Colpitts Chaotic Oscillator

The transformation steps for the Colpitts chaotic oscillators differ slightly from those of other oscillators and were derived specifically for this Thesis. The original nonlinear differential equations of the Colpitts oscillator are:

$$\begin{cases} C_1 \frac{\mathrm{d}v_{C_1}}{\mathrm{d}t} = i_{L_1} - i_C \\ C_2 \frac{\mathrm{d}v_{C_2}}{\mathrm{d}t} = \frac{V_1 - v_{C_2}}{R_1} - i_{L_1} - i_B \\ L_1 \frac{\mathrm{d}i_{L_1}}{\mathrm{d}t} = V_2 - v_{C_1} - v_{C_2} - i_{L_1} \cdot R_L \end{cases} . \tag{2.12}$$

The piecewise-linear approximation for the transistor currents is:

$$i_B = \begin{cases} 0, & \text{if } -v_{C_2} \leq V_{TH} \\ -\frac{v_{C_2} + V_{TH}}{R_{ON}} \cdot \rho_1, & \text{if } -v_{C_2} > V_{TH} \end{cases} . \tag{2.13}$$

In the first transformation step, the following dimensionless variables are introduced:

$$\rho_1 = \sqrt{\frac{L_1}{C_1}}, \quad \omega_1 = \frac{1}{\sqrt{L_1 \cdot C_1}}, \quad \tau_1 = \omega_1 \cdot t, \tag{2.14}$$

where ω_1 is the characteristic frequency; ρ_1 is the characteristic impedance; τ is the dimensionless time. These parameters are then applied to normalize and simplify the equations, resulting in (2.15). It is important to note that in (2.15) V_1 is taken as 5 V, i.e. the sign is moved from the constant to the equation, for convenience. As for the time scale of the dimensionless equations, the

characteristic frequency w_1 was selected.

The parameters in (2.14) were used for the convenience of the final normalized equations. It is important to note that the choice of normalization does not change the dynamics of the oscillator, only the appearance of the equations in dimensionless form. The fundamental frequency of the Colpitts oscillator can also be utilized, resulting in a different appearance of normalized equations (different coefficients and constants) with preserved dynamics and chaotic attractor. This was confirmed in work [41].

Applying the introduced variables yields:

$$\begin{cases} \dfrac{\mathrm{d}v_{C_1}}{\mathrm{d}\tau_1} \cdot \omega_1 = \dfrac{i_L - \beta \cdot i_B}{C_1} \cdot \dfrac{\rho_1}{\rho_1} \\ \dfrac{\mathrm{d}v_{C_2}}{\mathrm{d}\tau_1} \cdot \omega_1 = -\dfrac{V_1 + v_{C_2}}{R_1 \cdot C_2 \cdot \dfrac{\rho_1}{\rho_1}} + \dfrac{i_L + i_B}{C_2} \cdot \dfrac{\rho_1}{\rho_1} \\ \dfrac{\mathrm{d}i_{L_1}}{\mathrm{d}\tau_1} \cdot \omega_1 \cdot \dfrac{\rho_1}{\rho_1} = \dfrac{V_2 - v_{C_1} - v_{C_2}}{L_1} - i_{L_1} \cdot \dfrac{R_L}{L_1} \cdot \dfrac{\rho_1}{\rho_1} \end{cases} . \tag{2.15}$$

In the second transformation step, the variables shown in (2.16) are introduced. The use of characteristic impedance ρ_1 transforms the current i_{L_1} into a voltage, which means that the normalized equations have all three state variables as voltages.

$$\begin{gathered} i_{L_1} \cdot \rho_1 = \tilde{i}_{L_1}, \quad i_B \cdot \rho_1 = \tilde{i}_B, \quad \frac{\omega_1}{\rho_1} = \frac{1}{L_1}, \\ \epsilon = \frac{C_2}{C_1}, \qquad \frac{R_L}{\rho_1} = \tilde{R}_L, \quad \frac{R_1}{\rho_1} = \tilde{R}_1, \\ x = v_{C_1}, \qquad y = v_{C_2}, \qquad z = \tilde{i}_{L_1}. \end{gathered} \tag{2.16}$$

The final simplified normalized equations of the Colpitts chaotic oscillator thus are:

$$\begin{cases} \dfrac{\mathrm{d}x}{\mathrm{d}\tau_1} = z - \beta \cdot \tilde{i}_B \\ \dfrac{\mathrm{d}y}{\mathrm{d}\tau_1} = -\dfrac{V_1 + y}{\tilde{R}_1 \cdot \epsilon} + \dfrac{z + \tilde{i}_B}{\epsilon} \\ \dfrac{\mathrm{d}z}{\mathrm{d}\tau_1} = V_2 - x - y - z \cdot \tilde{R}_L \end{cases} . \tag{2.17}$$

The piecewise-linear approximation for the base current with the introduced transformations is:

$$\tilde{i}_B = \begin{cases} 0, & \text{if } -y \leq V_{TH} \\ -\dfrac{y + V_{TH}}{R_{ON}} \cdot \rho_1, & \text{if } -y > V_{TH} \end{cases} . \tag{2.18}$$

The acquired system (2.17) expressed in dimensionless time is further used for digital implementation.

2.2 Forward Euler Numerical Integration

Since the differential equations describe a continuous-time oscillators and the task is to design discrete-time counterparts of the oscillators, the next step is to transform the differential equations to obtain the discrete solutions of the systems. In this step, the forward Euler numerical integration is applied to acquire the approximate discrete solution for the integration step $\Delta\theta$. The forward Euler method was selected for being computationally cheap per step. The simplicity of the Forward Euler method makes it well-suited for discrete hardware implementation. Although higher-order methods such as Runge–Kutta [49] are preferable for obtaining exact trajectories, the forward Euler method can reproduce the attractor's shape – provided that $\Delta\theta$ is chosen carefully – which is precisely the requirement for digital hardware implementation.

Vilnius Chaotic Oscillator

In the case of Vilnius chaotic oscillator, applying the forward Euler numerical integration yields the following difference equations:

$$\begin{cases} x[n+1] = y[n] \cdot \Delta\theta + x[n] \\ y[n+1] = (a \cdot y[n] - x[n] - z[n]) \cdot \Delta\theta + y[n] \\ z[n+1] = (b + y[n] - c \cdot (\mathrm{e}^{z[n]} - 1)) \cdot \frac{\Delta\theta}{\varepsilon} + z[n] \end{cases}, \tag{2.19}$$

where $a = 0.5$; $b = 5.769$; $c = 9.155 \cdot 10^{-5}$; $\varepsilon = 0.15$; $\Delta\theta = 2^{-10}$. It is important to note that in (2.2) c depends on the current i_S, meaning that in reality, this constant is variable. However, this constant is very small, and the source study [47] explicitly states that the chaotic mode is insensitive to c; thus, taking c as constant is valid.

RC Chaotic Oscillator

Similarly, the discrete equations (2.20) for the RC chaotic oscillator are derived by applying forward Euler numerical integration to (2.10):

$$\begin{cases} x[n+1] = \big(\big(k_1 - 1\big) \cdot y[n] - \big(k_1 - 1\big) \cdot z[n] - x[n]\big) \cdot \Delta\theta + x[n] \\ y[n+1] = \big(\big(k_1 - 2\big) \cdot y[n] - \big(k_1 - 1\big) \cdot z[n] - x[n]\big) \cdot \Delta\theta + y[n] \\ z[n+1] = \big(k_1 \cdot y[n] - \big(k_1 + \alpha\big) \cdot z[n] + \beta \cdot (z[n] - 1) \cdot H\left(z[n] - 1\right)\big) \cdot \Delta\theta + z[n] \end{cases}. \tag{2.20}$$

The Heaviside step function in this case is

$$H = \begin{cases} 0, & \text{if } z[n] - 1 < 0 \\ 1, & \text{if } z[n] - 1 \geq 0 \end{cases} . \tag{2.21}$$

The final equation has the following parameters: $k_1 = 5.5$; $\alpha = 10$; $\beta = 14.1026$; $\Delta\theta = 2^{-10}$.

Colpitts Chaotic Oscillator

In the case of Colpitts chaotic oscillator, applying forward Euler numeric integration results in the following system:

$$\begin{cases} x[n+1] = \left(z[n] - \beta \cdot \tilde{i}_B\right) \cdot \Delta\theta + x[n] \\ y[n+1] = \left(-\dfrac{y[n] + V_1}{\tilde{R}_1 \cdot \epsilon} + \dfrac{z[n] + \tilde{i}_B}{\epsilon}\right) \cdot \Delta\theta + y[n] \\ z[n+1] = \left(V_2 - x[n] - y[n] - z[n] \cdot \tilde{R}_L\right) \cdot \Delta\theta + z[n] \end{cases} . \tag{2.22}$$

The nonlinearity in this case is

$$\tilde{i}_B = \begin{cases} 0, & \text{if } -y[n] \leq V_{TH} \\ -\dfrac{y[n] + V_{TH}}{R_{ON}} \cdot \rho_1, & \text{if } -y[n] > V_{TH} \end{cases} . \tag{2.23}$$

The final equation has the following parameters: $\beta = 200$; $\epsilon = 1$; $\rho_1 = 43.033\ \Omega$; $\tilde{R}_L = 0.9295$; $\tilde{R}_1 = 9.295$; $R_{ON} = 100\ \Omega$; $V_1 = V_2 = 5$ V; $V_{TH} = 0.75$ V; $\Delta\theta = 2^{-10}$.

2.3 Digital Design

This section describes the implementation of the chaotic oscillators' difference equations in a digital design that is intended for an FPGA. The task is to create a unified design approach that is applicable to all three chaotic oscillators. The first step of creating the digital design is outlining the mathematical operations of the difference Equations (2.19), (2.20) and (2.22) from the perspective of sequential digital design. In all three chaotic oscillator difference equations acquired via forward Euler numerical integration the resulting system shows that the next value of the state variable is acquired from the current value of the state variable plus the derivative multiplied by the time step.

Difference equation of the Vilnius chaotic oscillator with highlighted operations:

$$
\begin{cases}
\underbrace{x[n+1]}_{\text{Next value}} = \underbrace{x[n]}_{\text{Current value}} + \underbrace{y[n]}_{\text{Derivative}} \cdot \underbrace{\Delta\theta}_{\text{Time step}} \\
\underbrace{y[n+1]}_{\text{Next value}} = \underbrace{y[n]}_{\text{Current value}} + \underbrace{(a \cdot y[n] - x[n] - z[n])}_{\text{Derivative}} \cdot \underbrace{\Delta\theta}_{\text{Time step}} \\
\underbrace{z[n+1]}_{\text{Next value}} = \underbrace{z[n]}_{\text{Current value}} + \underbrace{\left(\frac{b}{\epsilon} + \frac{1}{\epsilon} \cdot y[n] - \frac{c}{\epsilon} \cdot (\mathrm{e}^{z[n]} - 1)\right)}_{\text{Derivative}} \cdot \underbrace{\Delta\theta}_{\text{Time step}}
\end{cases}
. \tag{2.24}
$$

Difference equation of the RC chaotic oscillator with highlighted operations:

$$
\begin{cases}
\underbrace{x[n+1]}_{\text{Next value}} = \underbrace{x[n]}_{\text{Current value}} + \underbrace{((k_1 - 1) \cdot y[n] - (k_1 - 1) \cdot z[n] - x[n])}_{\text{Derivative}} \cdot \underbrace{\Delta\theta}_{\text{Time step}} \\
\underbrace{y[n+1]}_{\text{Next value}} = \underbrace{y[n]}_{\text{Current value}} + \underbrace{\left((k_1 - 2) \cdot y[n] - (k_1 - 1) \cdot z[n] - x[n]\right))}_{\text{Derivative}} \cdot \underbrace{\Delta\theta}_{\text{Time step}} \\
\underbrace{z[n+1]}_{\text{Next value}} = \underbrace{z[n]}_{\text{Current value}} + \underbrace{\left(k_1 \cdot y[n] - (k_1 + \alpha) \cdot z[n] + \beta \cdot (z[n] - 1) \cdot H\left(z[n] - 1\right)\right)}_{\text{Derivative}} \cdot \underbrace{\Delta\theta}_{\text{Time step}}
\end{cases}
. \tag{2.25}
$$

Difference equation of the Colpitts chaotic oscillator with highlighted operations:

$$
\begin{cases}
\underbrace{x[n+1]}_{\text{Next value}} = \underbrace{x[n]}_{\text{Current value}} + \underbrace{\left(z[n] - \beta \cdot \tilde{i}_B\right)}_{\text{Derivative}} \cdot \underbrace{\Delta\theta}_{\text{Time step}} \\
\underbrace{y[n+1]}_{\text{Next value}} = \underbrace{y[n]}_{\text{Current value}} + \underbrace{\left(-\frac{y[n] + V_1}{\tilde{R}_1 \cdot \epsilon} + \frac{z[n] + \tilde{i}_B}{\epsilon}\right)}_{\text{Derivative}} \cdot \underbrace{\Delta\theta}_{\text{Time step}} \\
\underbrace{z[n+1]}_{\text{Next value}} = \underbrace{z[n]}_{\text{Current value}} + \underbrace{\left(V_2 - x[n] - y[n] - z[n] \cdot \tilde{R}_L\right)}_{\text{Derivative}} \cdot \underbrace{\Delta\theta}_{\text{Time step}}
\end{cases}
. \tag{2.26}
$$

This form of the difference equations is subsequently mapped onto a digital circuit, as illustrated in Fig. 2.1. The implementation is based on registers that store and update the state variables at each rising edge of the clock signal clk. The current state values $x[n]$, $y[n]$, and $z[n]$ together with the system constants, are fed into the "Derivative calculation pipeline", which evaluates the derivative terms of the difference equations for each state variable with matched latency. The computed derivatives are then multiplied by the discrete time step $\Delta\theta$. The resulting increments are added to the corresponding state variables, producing the updated values $x[n+1]$, $y[n+1]$, and $z[n+1]$ for the next iteration.

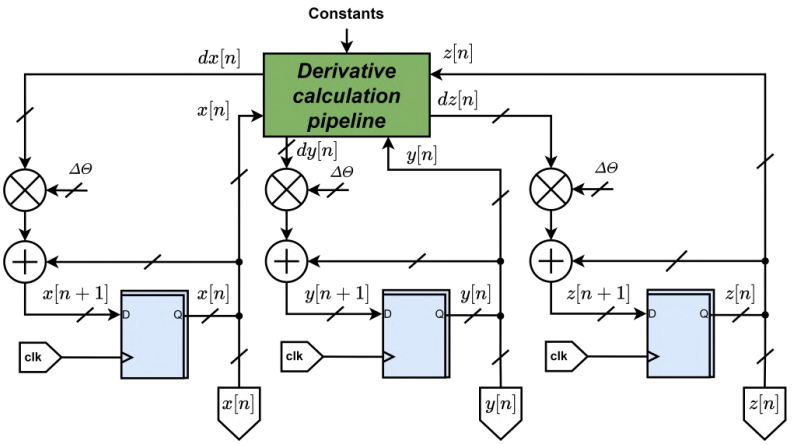


Fig. 2.1. Full digital design of chaotic oscillator system equations.

With the presented design, the implementation difference of each chaotic oscillator is the design of the derivatives calculation pipeline, which is addressed in the following subsections.

Derivative Calculation Pipeline for Vilnius Chaotic Oscillator

The pipelining approach means that the derivative calculation is split into operation stages followed by registers. In the case of the Vilnius chaotic oscillator, the derivative part of the difference equation is presented in (2.27), where the pipelining process is highlighted by outlining the order of discrete operations for every derivative. For example, calculating $\mathrm{d}y$ would need to perform a $a \cdot y[n]$ in the first clock cycle, while $x[n] - z[n]$ is done in parallel in the same clock cycle. After both results are registered, the difference of the two results is performed in the second clock cycle. It is notable that the increment calculations of the state variables differ by the number of clock cycles required. However, oscillators must update all state variables simultaneously, which means that additional registers are required to match the delay of the longest pipeline, $\mathrm{d}z$ in this case.

$$\begin{cases} \mathrm{d}x = \underbrace{y[n]}_{1} \\ \mathrm{d}y = \underbrace{\underbrace{a \cdot y[n]}_{1} - \underbrace{x[n] - z[n]}_{1}}_{2} \\ \mathrm{d}z = \underbrace{\underbrace{\underbrace{\underbrace{\frac{1}{\varepsilon} \cdot y[n]}_{1}}_{2} + \frac{b}{\varepsilon}}_{3} - \underbrace{\underbrace{\frac{c}{\varepsilon} \cdot (\mathrm{e}^{\underbrace{z[n]}_{1}} - 1)}_{2}}_{3}}_{4} \end{cases} . \tag{2.27}$$

Another consideration point is calculating the nonlinear function $e^{z[n]}$. To acquire the exponent value in one clock cycle, it was decided to use ROM with the pre-calculated $e^{z[n]}$ values for the possible $z[n]$ range. This idea was expanded further by implementing the $\frac{b}{\epsilon} - \frac{c}{\epsilon} \cdot (e^{z[n]} - 1)$ in ROM due to the fact that it contains the aforementioned exponents and constants. This allows simplifying the pipeline, resulting in (2.28).

$$\begin{cases} \mathrm{d}x = \underbrace{y[n]}_{1} \\ \mathrm{d}y = \underbrace{\underbrace{a \cdot y[n]}_{1} - \underbrace{x[n] - z[n]}_{1}}_{2} \\ \mathrm{d}z = \underbrace{\underbrace{\frac{1}{\epsilon} \cdot y[n]}_{1} + \underbrace{\mathrm{ROM}(z[n])}_{1}}_{2} \end{cases} . \tag{2.28}$$

The system design was initially implemented in MATLAB to accelerate debugging and verification. The system is designed using a fixed-point data format. The MATLAB environment enables convenient adjustment of the integer and fractional bit widths of signals, which is essential for fixed-point implementation on an FPGA. A ROM-based implementation of the nonlinear function is first developed and validated in MATLAB, and the resulting memory contents are subsequently reused in the implementation of the digital design in hardware.

The memory structure is illustrated in Figure 2.2. The integer and fractional bit widths are identical for all three state variables. The design process in MATLAB showed that the $x[n]$ requires at least 8 bits for the signed integer part to describe the dynamic range of the signal. Therefore, an 8-bit integer width is assigned to $x[n]$, $y[n]$, and $z[n]$. Additionally, 14 bits are required for the fractional part to ensure sufficient precision for maintaining chaotic dynamics. The resulting word length of the chaotic signals is 22 bits.

As shown in Figure 2.2, the ROM addressing is derived from the variable $z[n]$ where 6 bits from the integer part and 6 bits from the fractional part are used to form the address. Consequently, the ROM has a 12-bit address space, allowing storage of 4096 data entries, each represented with an 8-bit integer part and a 14-bit fractional part.

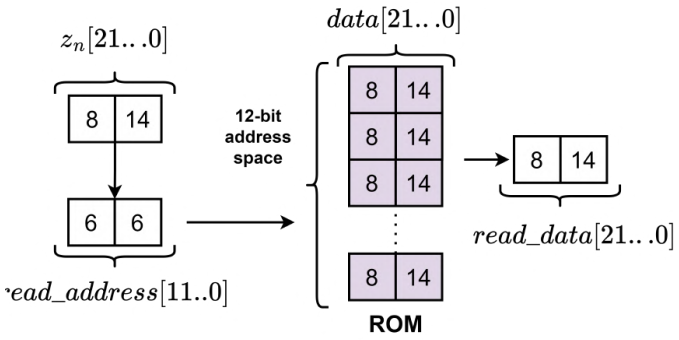


Fig. 2.2. The ROM structure for the Vilnius chaotic oscillator.

Fig. 2.3 demonstrates the contents of the ROM in the case of the Vilnius chaotic oscillator. The y-axis is used for the stored memory values of $\frac{b}{\epsilon} - \frac{c}{\epsilon} \cdot (e^{z[n]} - 1)$ for all possible read address ($z[n]$ rounded to 12 bits) values on the x-axis. The input range is from -32 to 31.984375. For the $z[n]$ inputs greater than 12, the in-memory expression's integer part exceeds the assigned 8-bit signed integer value range, so the stored values are truncated to $128 - 2^{-14}$ (maximal signed 8-bit integer and 14-bit fractional parts fixed point number).

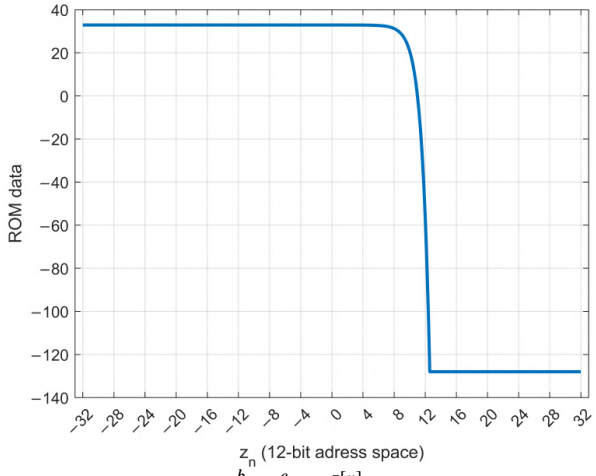


Fig. 2.3. Memory data of $\frac{b}{\epsilon} - \frac{c}{\epsilon} \cdot (e^{z[n]} - 1)$ for $z[n]$ input range.

Derivative Calculation Pipeline for RC Chaotic Oscillator

The equation demonstrates that the system (2.25) for the RC oscillator is very similar to (2.24) in the case of the Vilnius oscillator. The only difference is the operations performed in the "Derivative" part of the system. For this reason, it is possible to use the digital design presented in Fig. 2.1, only adjusting the operations done in the derivative calculation block. From (2.25), it can be seen that the nonlinearity (Heaviside function) has $z[n]$ as an argument. Following the Vilnius oscillator

implementation procedure described above, the $\left(k_1+\alpha\right)\cdot z[n]+\beta\cdot(z[n]-1)\cdot H\left(z[n]-1\right)$ is replaced with a ROM. The simplified pipeline is shown in (2.29).

$$\begin{cases} \mathrm{d}x = \underbrace{\underbrace{\underbrace{\left(k_1-1\right)\cdot y[n]}_{1}-\underbrace{\left(k_1-1\right)\cdot z[n]}_{1}}_{2}-\underbrace{\underbrace{x[n]}_{1}}_{2}}_{3} \\ \mathrm{d}y = \underbrace{\underbrace{\underbrace{\left(k_1-2\right)\cdot y[n]}_{1}-\underbrace{\left(k_1-1\right)\cdot z[n]}_{1}}_{2}-\underbrace{\underbrace{x[n]}_{1}}_{2}}_{3} \\ \mathrm{d}z = \underbrace{\underbrace{k_1\cdot y[n]}_{1}+\underbrace{\mathrm{ROM}(z[n])}_{1}}_{2} \end{cases}. \tag{2.29}$$

The ROM design is identical to Fig. 2.2. The ROM contents for the RC chaotic oscillator are shown in Fig. 2.4. In this case, for $z[n]<-8$, the in-memory expression values exceed the 8-bit signed integer value range, so the stored values are truncated to $128-2^{-14}$.

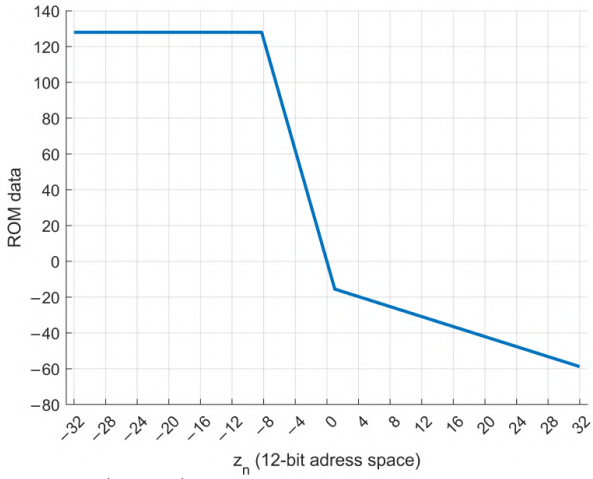


Fig. 2.4. Memory data of $-\left(k_1+\alpha\right)\cdot z[n]+\beta\cdot(z[n]-1)\cdot H\left(z[n]-1\right)$ for $z[n]$ input range.

Derivative Calculation Pipeline for Colpitts Chaotic Oscillator

The derivative calculation pipeline design for the Colpitts oscillator is addressed next. The "Derivative" part from (2.26) is shown in (2.30).

$$
\begin{cases}
\mathrm{d}x = \underbrace{\underbrace{z[n]}_{1} - \underbrace{\beta \cdot \tilde{i}_B}_{1}}_{2} \\
\mathrm{d}y = \underbrace{\underbrace{-\underbrace{\frac{y[n]}{\tilde{R}_1 \cdot \epsilon}}_{1} - \underbrace{\frac{V_1}{\tilde{R}_1 \cdot \epsilon}}_{1}}_{2} + \underbrace{\underbrace{\frac{z[n]}{\epsilon}}_{1} + \underbrace{\frac{\tilde{i}_B}{\epsilon}}_{1}}_{2}}_{3} \\
\mathrm{d}z = \underbrace{\underbrace{\underbrace{V_2}_{1} - \underbrace{x[n]}_{1}}_{2} - \underbrace{\underbrace{y[n]}_{1} - \underbrace{z[n]}_{1}}_{2}}_{3}
\end{cases}
. \tag{2.30}
$$

To implement the derivative calculation pipeline, the nonlinearity $\tilde{i}_B$ is approximated using ROM. The $\tilde{i}_B$ is used in two equations and it is most convenient to form two different ROMs. The first ROM contains $\beta \cdot \tilde{i}_B$. The second ROM contains $-\frac{y[n]}{\tilde{R}_1 \cdot \epsilon} - \frac{V_1}{\tilde{R}_1 \cdot \epsilon} + \frac{\tilde{i}_B}{\epsilon}$, as the $\tilde{i}_B$ itself has $y[n]$ as input argument. The final pipeline structure is:

$$
\begin{cases}
\mathrm{d}x = \underbrace{\underbrace{z[n]}_{1} - \underbrace{\mathrm{ROM}_1}_{1}}_{2} \\
\mathrm{d}y = \underbrace{\underbrace{\frac{z[n]}{\epsilon}}_{1} - \underbrace{\mathrm{ROM}_2}_{1}}_{2} \\
\mathrm{d}z = \underbrace{\underbrace{\underbrace{V_2}_{1} - \underbrace{x[n]}_{1}}_{2} - \underbrace{\underbrace{y[n]}_{1} - \underbrace{z[n]}_{1}}_{2}}_{3}
\end{cases}
. \tag{2.31}
$$

As shown in Fig. 2.5, in the case of the Colpitts chaotic oscillator, the ROM addressing is derived from the variable $y[n]$, where 6 bits from the integer part and 6 bits from the fractional part are used to form the address. Consequently, the ROM has a 12-bit address space, allowing storage of 4096 data entries, each represented with a 6-bit integer part and a 16-bit fractional part. Fig. 2.6 demonstrates the contents of the two ROMs.

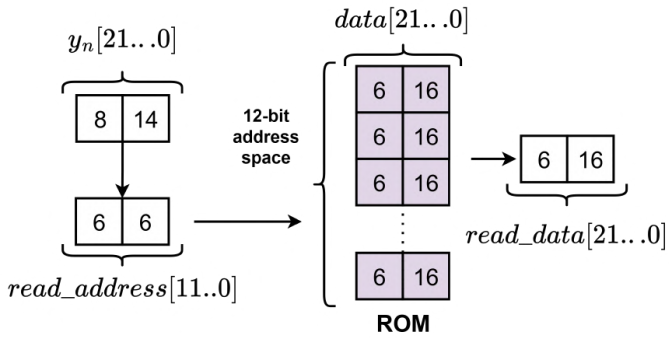
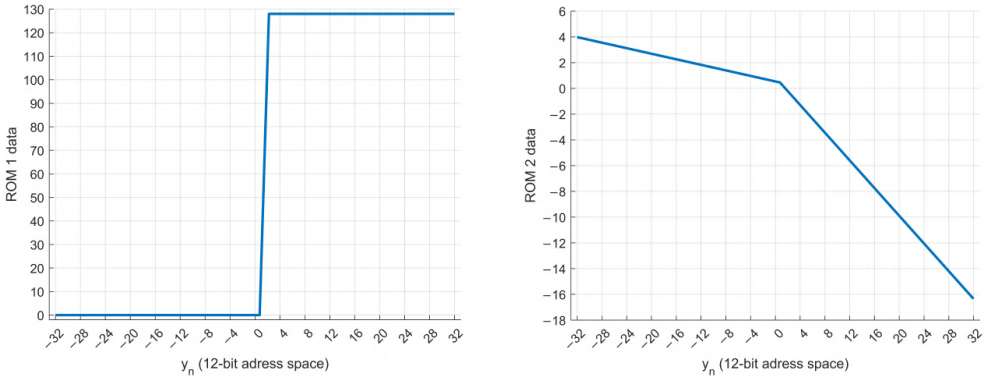


Fig. 2.5. The ROM design for the Colpitts chaotic oscillator.



(a) (b)

Fig. 2.6. ROM containing $\beta \cdot \tilde{i}_B$ (a) and ROM containing $-\frac{y[n]}{\tilde{R}_1 \cdot \epsilon} - \frac{V_1}{\tilde{R}_1 \cdot \epsilon} + \frac{\tilde{i}_B}{\epsilon}$ (b).

With the digital designs of the three chaotic oscillators established, the attention can be turned to the implementation of the digital design of the aforementioned oscillators on the FPGA platform.

2.4 Digital Design Implementation

After establishing the complete digital design of the oscillator with the block calculating the derivatives and the ROM storing the nonlinearity in MATLAB, the VHDL implementation was performed by translating the expressions. This subsection discusses the FPGA resource utilization and verifies the correspondence of the VHDL and MATLAB designs.

The FPGA implementation is targeted at the Altera (Intel) Cyclone V 5CSXFC6D6F31C6 device on the DE10-Standard development board. The resource utilization of the VHDL implementations of the chaotic oscillators is summarized in Table 2.1. The comparison includes the number of adaptive logic modules (ALMs), digital signal processing (DSP) blocks, and block random access memory

(BRAM) bits, as well as the percentage of utilized resources relative to the total available on the 5CSXFC6D6F31C6 device.

The results indicate that the designed oscillators occupy only a small fraction of the available FPGA resources. This suggests that the implementations are suitable for deployment on lower-capacity devices or can be readily integrated into more complex, resource-intensive systems that incorporate chaotic oscillators.

A comparison of the individual designs shows that the RC oscillator requires a slightly higher number of ALMs and DSP blocks compared to the other implementations due to a longer pipeline, although the difference remains modest. In terms of memory usage, the Colpitts oscillator exhibits approximately twice the BRAM utilization, which is attributed to the use of two ROMs in its design.

Table 2.1
FPGA resource utilization of chaotic oscillators

Chaotic oscillator	**ALMs**	**DSP blocks**	**BRAM bits**
Vilnius	67 (< 1 %)	1 (1 %)	90122 (2 %)
RC	115 (< 1 %)	5 (6 %)	90122 (2 %)
Colpitts	95 (< 1 %)	1 (1 %)	180224 (3 %)

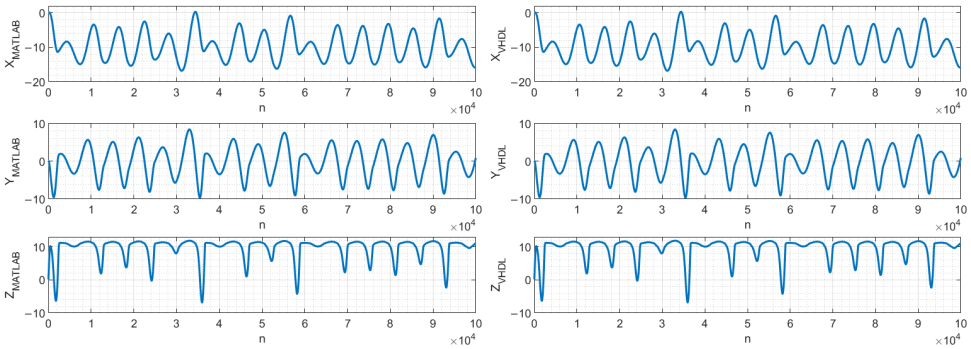
To provide vendor-independent comparison in Table 2.2, Altera ALMs are converted to equivalent LUTs assuming 1 ALM $\approx$ 2 four-input LUTs, following the Cyclone V ALM architecture. The table also includes the number of flip-flops (FFs).

Table 2.2
FPGA resource utilization of chaotic oscillators using vendor-independent metrics

Chaotic oscillator	**LUTs**	**FFs**	**DSP blocks**	**BRAM bits**
Vilnius	134	234	1	90122
RC	230	360	5	90122
Colpitts	190	330	1	180224

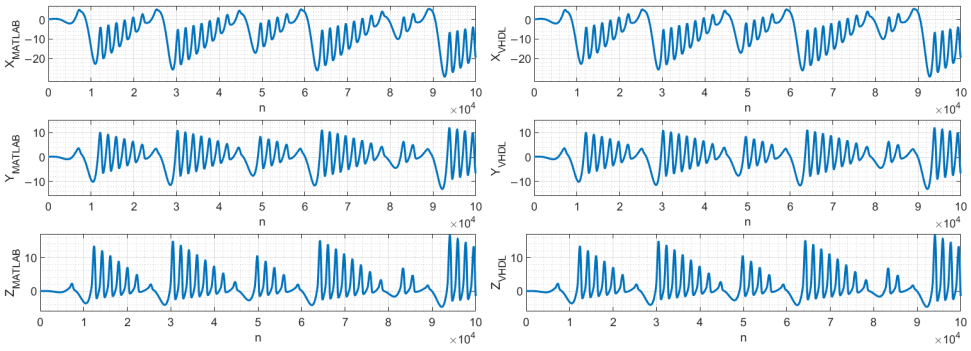
To verify that the VHDL-based design accurately reproduces the behavior of the MATLAB model, a functional simulation was performed using the Questa (ModelSim) simulation tool. The simulation results were subsequently exported to MATLAB for direct comparison.

Fig. 2.7 and 2.8 demonstrate that, for identical parameters and initial conditions, the signals generated by the MATLAB model and the VHDL-based implementation are identical and do not diverge over 10^5 clock cycles. This indicates that the MATLAB model precisely captures the behavior of the corresponding digital design. The same consistency was also observed in longer simulation runs.



(a) (b)

Fig. 2.7. MATLAB (a) and VHDL-based digital design (b) Vilnius chaotic oscillator state variable signals.



(a) (b)

Fig. 2.8. MATLAB (a) and VHDL-based digital design (b) RC chaotic oscillator state variable signals.

The next step is the verification of the digital designs in the hardware – the 5CSXFC6D6F31C6 FPGA chip on the DE10-Standard development board. It is important to note that in this FPGA implementation, the master clock frequency and the numerical integration step $\Delta\theta$ are not identical quantities. The FPGA system clock f_{clk} determines the rate at which arithmetic operations are executed, while $\Delta\theta$ is implemented as a constant multiplier in the discretized equations to control the effective integration step size. In this way, $\Delta\theta$ can be set independently of the hardware clock, allowing the same FPGA design to emulate different step sizes by adjusting only the scaling constants, thus ensuring flexibility in digital realization. To match the analog chaotic oscillator, the integration step must be the relation of the scaling frequency to the clock frequency (2.32). As the system is implemented in fixed-point arithmetic, the $\Delta\theta$ is rounded with the fixed-point precision.

$$\Delta\theta = \frac{f_0}{f_{clk}} \tag{2.32}$$

Fig. 2.9 presents the hardware verification setup. The FPGA-based chaotic oscillator operates on a DE10-Standard board. An ADA-HSMC daughter card is connected via a high-speed mezzanine card (HSMC) connector, providing a two-channel analog-to-digital converter (ADC) and a two-channel digital-to-analog converter (DAC). For each chaotic oscillator, the two DACs output a pair of state variable signals that are recorded using the oscilloscope channels of the ADP3450. This setup allows acquiring all attractor projections of the chaotic oscillators implemented in the FPGA.

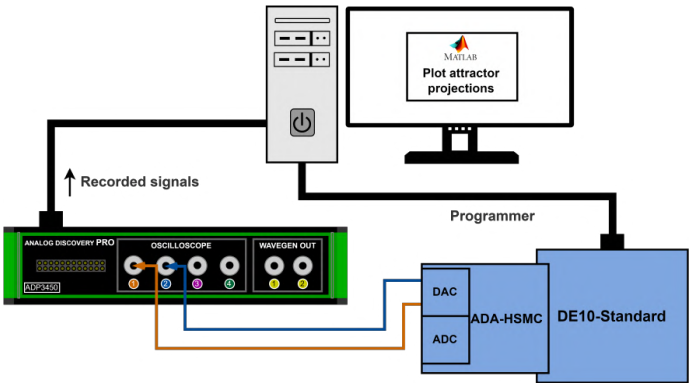
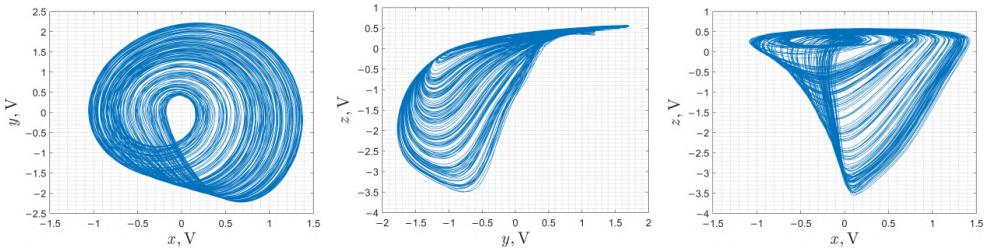


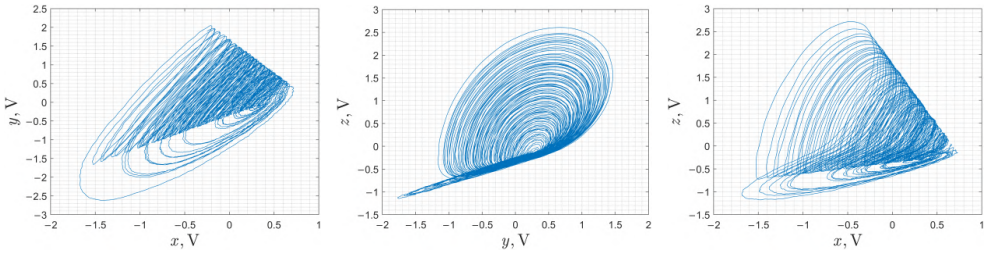
Fig. 2.9. Hardware verification of the chaotic oscillators implemented on an FPGA.

The recorded signals are exported to MATLAB for visualization. Fig. 2.10–2.12 compile the attractor projections of the three chaotic oscillators. The figures show attractors that are very similar to their analog counterparts, with minor discrepancies due to the model's finite precision. As demonstrated in the following studies, these discrepancies are not significant.



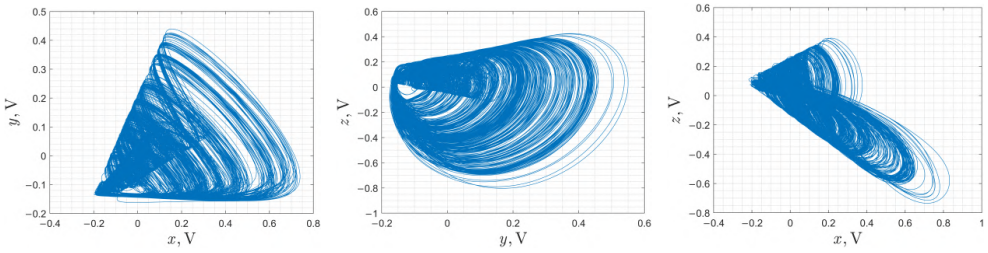
(a) (b) (c)

Fig. 2.10. Two-dimensional attractor projections x and y (a), y and z (b), x and z (c) of the Vilnius chaotic oscillator.



(a) (b) (c)

Fig. 2.11. Two-dimensional attractor projections x and y (a), y and z (b), x and z (c) of the RC chaotic oscillator.



(a) (b) (c)

Fig. 2.12. Two-dimensional attractor projections x and y (a), y and z (b), x and z (c) of the Colpitts chaotic oscillator.

The developed discrete chaotic oscillators, together with their analog counterparts, are used in subsequent studies. The presented results confirm that the chosen discretization approach, based on the forward Euler method, is sufficient to preserve the qualitative dynamics of the original continuous-time systems, including the characteristic structure of chaotic attractors. Furthermore, the consistency between MATLAB and VHDL-based implementations demonstrates that the fixed-point representation and hardware realization do not introduce significant numerical distortion or instability.

Consequently, the developed discrete models accurately reproduce the behavior of their analog counterparts and can be reliably used in further investigations, including synchronization studies between analog and discrete chaotic oscillators.

3 IMPLEMENTATION AND ANALYSIS OF CHAOTIC SYNCHRONIZATION IN ANALOG-DISCRETE SYSTEMS

The ability to synchronize two chaotic systems, despite their inherent sensitivity to initial conditions, has opened significant avenues in secure communications, cryptography, and sensor networks. While classical chaotic synchronization typically involves two identical continuous-time systems or two identical discrete-time maps, modern communication infrastructures increasingly rely on the interplay between physical analog hardware and digital processing units. This requires a deep understanding of hybrid chaotic synchronization, where a continuous-time analog oscillator must synchronize with a discrete-time numerical model or a digital hardware implementation and vice versa.

This chapter explores the feasibility of achieving bidirectional synchronization between analog and discrete chaotic oscillators using the Pecora-Carroll method. The study progresses through three levels of complexity:

1. **Numerical Simulation:** Investigating the interaction between continuous-time LTspice models and discrete-time MATLAB models.

2. **Hardware-in-the-Loop:** Establishing synchronization between physical PCB-based oscillators and MATLAB models via data acquisition systems.

3. **Full Hardware Integration:** Evaluating synchronization between analog circuits and digital implementations on FPGA platforms.

By using Pearson correlation as a quantitative metric, this section aims to demonstrate that the fundamental principles of chaotic substitution remain robust even when crossing the boundary between continuous and discrete time scales, despite the inevitable presence of quantization noise and component tolerances.

3.1 Pecora–Carroll Chaotic Synchronization

The first method for synchronizing chaotic systems was proposed in 1990 by Louis M. Pecora and Thomas L. Carroll in their study [50]. This synchronization technique was also called “Pecora–Carroll” synchronization. The principle of synchronization is outlined in Fig. 3.1 and is as follows: the drive system generates a chaotic signal y_1, which is then used to replace or “drive” the corresponding variable y_2 in the response system. As a result, the state variable x_2 repeats x_1, and z_2 repeats z_1. Synchronization occurs even if the initial conditions of the two chaotic systems differ. The term “substitution method” [51] is also widely used in the literature to refer to this synchronization technique.

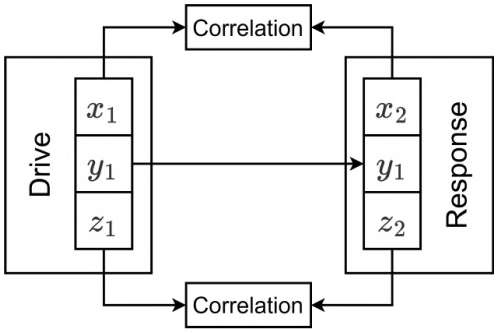


Fig. 3.1. Pecora–Carroll chaotic synchronization.

To evaluate synchronization quality, the Pearson correlation coefficient is often used to compute correlation. This metric quantifies the similarity of the drive and response oscillator's signals. For arbitrary a and b input signals, the coefficient, denoted as r, is calculated using the following equation:

$$r = \frac{\sum_{i=1}^{n}(a_i - \bar{a})(b_i - \bar{b})}{\sqrt{\sum_{i=1}^{n}(a_i - \bar{a})^2}\sqrt{\sum_{i=1}^{n}(b_i - \bar{b})^2}}, \tag{3.1}$$

where n is the number of samples, a_i and b_i are individual samples at index i, and $\bar{a}$ and $\bar{b}$ sample means over n points.

The Pearson correlation coefficient r ranges from -1 to 1. A value of $r = 1$ indicates a perfect positive linear relationship, which remains invariant to signal scaling or shifting. Conversely, $r = -1$ signifies a perfect negative linear relationship, while $r = 0$ indicates a complete absence of linear correlation between the analyzed signals.

3.2 Study Based on Numerical Simulation

This section investigates the feasibility of achieving Pecora–Carroll chaotic synchronization for the selected oscillators in two configurations. The first configuration considers analog-discrete synchronization, where the drive system is an anlog oscillator modeled in LTspice, and the response system is the corresponding discrete oscillator implemented in MATLAB. The second configuration examines discrete-analog synchronization, in which the drive system is the discrete oscillator implemented in MATLAB, while the response system is the analog oscillator modeled in LTspice.

This Numerical Simulation step is the first step toward investigating hybrid chaotic synchronization, focusing specifically on the interaction between analog and discrete oscillator models.

To ensure compatibility of the chaotic signals of the LTspice and discrete MATLAB models, transformations described for the derivation of the discrete normalized models were applied to the signals acquired from LTspice before interfacing them to the MATLAB model. The inverse transformations were applied to the MATLAB discrete model's signals before interfacing them in LTspice.

The transformation required converting the amplitudes from dimensionless values to volts, and the time axis from discrete dimensionless time to continuous time.

Analog-Discrete Synchronization

Fig. 3.2 demonstrates the analog-discrete synchronization configuration. The y_1 state variable of the LTspice-based drive oscillator substitutes y_2 of the MATLAB-based response discrete oscillator. As a result, x_2 should repeat the behavior of x_1, and z_2 should repeat the behavior of z_1. The synchronization's quality was evaluated using the aforementioned Pearson correlation coefficient.

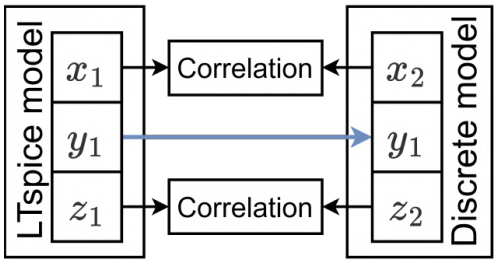


Fig. 3.2. Analog-discrete Pecora–Carroll synchronization using the LTspice and discrete models.

In the case of Vilnius and RC chaotic oscillators, this exact configuration was used – synchronization was achieved using the y state variable. In the case of Colpitts chaotic oscillators, the synchronization was done using the x state variables as synchronization signals, and synchronization was evaluated using the y and z state variables.

The synchronized x and z signals in the case of the Vilnius chaotic oscillator are presented in Fig. 3.3. The synchronization signal is applied at a discrete time $\theta = 100$. The figure shows that the two chaotic systems are initially asynchronous due to conceptual differences in their models, but after $\theta = 100$, synchronization is achieved. Overall, the discrete model reproduces the behavior of the LTspice model, but the signals are not a perfect match, showing differences in amplitude and mean value.

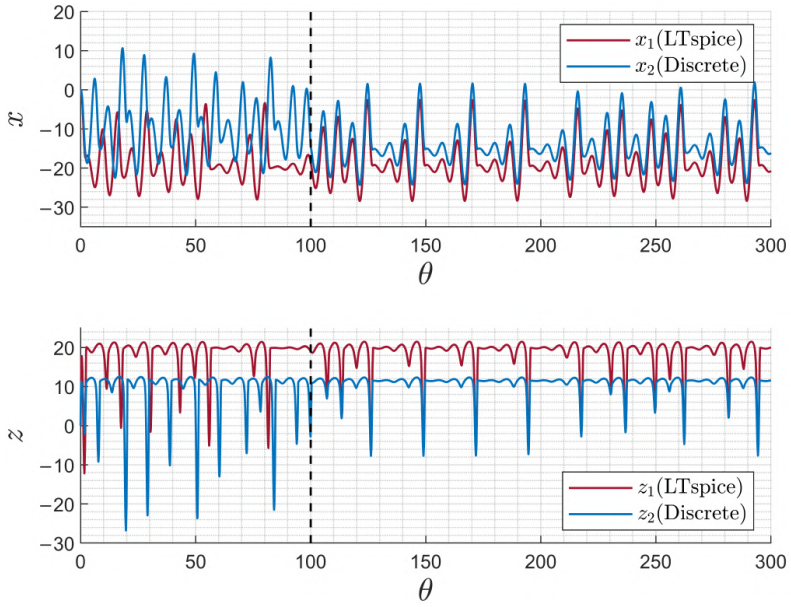


Fig. 3.3. Time diagrams of the synchronized signals in the case of Vilnius chaotic oscillator analog-discrete synchronization.

The Pearson correlation coefficient is computed for the 10^6 samples of synchronized signals and is presented in Table 3.1. The Pearson correlation coefficient is invariant to the mean values of the analyzed signals, since it is defined in terms of mean-centered variables. By subtracting the mean value from each signal, the coefficient quantifies only the linear relationship between their deviations from the mean. Consequently, any constant offset does not influence the resulting correlation. The results compiled in Table 3.1 confirm that the analog-discrete chaotic synchronization was achieved, indicated by the Pearson correlation coefficient values exceeding 0.9.

Table 3.1
Analog-discrete synchronization evaluation using Pearson correlation coefficient

Chaotic oscillator	x_1 **and** x_2	y_1 **and** y_2	z_1 **and** z_2
Vilnius	0.99	—	0.97
RC	0.99	—	0.99
Colpitts	—	0.99	0.99

Discrete-Analog Synchronization

Fig.3.4 demonstrates the discrete-analog synchronization configuration. The same Pecora-Carroll synchronization technique is used, but the synchronization direction is reversed compared to the previous configuration. The y_2 state variable of the MATLAB-based drive oscillator substitutes y_1 of the LTspice-based response oscillator. As a result, x_1 should repeat the behavior of x_2, and

z_1 should repeat the behavior of z_2. The synchronization's quality was evaluated using the Pearson correlation coefficient.

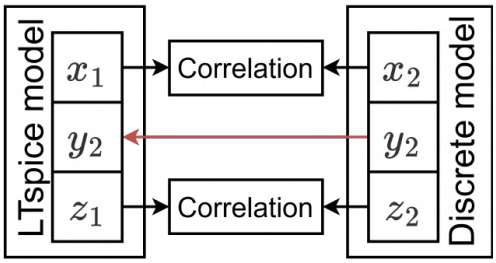


Fig. 3.4. Discrete-analog Pecora–Carroll synchronization using the LTspice and discrete models.

The synchronized x and z signals in the case of the Vilnius chaotic oscillator are presented in Fig.3.5. The synchronization signal is applied at 100 μs. The figure shows that the two chaotic systems are initially asynchronous, but after 100 μs, the LTspice model repeats the behavior of the discrete model. As in the previous study, the signals do not match perfectly after the synchronization signal is applied, demonstrating mismatches in amplitudes and mean values. For robust applications of chaotic synchronization, such as chaotic communications, an exact match is not needed.

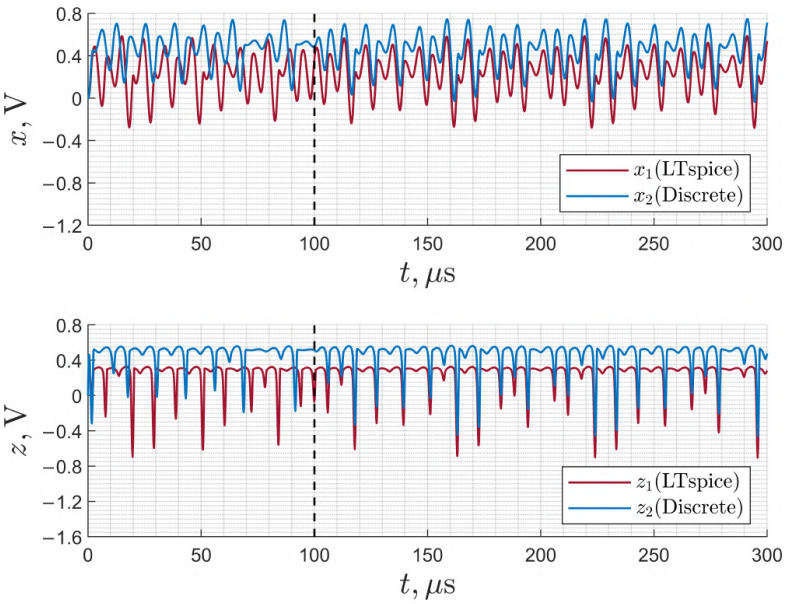


Fig. 3.5. Time diagrams of the synchronized signals in the case of Vilnius chaotic oscillator discrete-analog synchronization.

The Pearson correlation coefficient is computed for the 10^6 samples of synchronized signals. Table 3.2 compiles the results for the three chaotic oscillators. Table 3.2 confirms that the discrete-analog chaotic synchronization was achieved for all three chaotic oscillators, indicated by the Pearson

correlation coefficient values exceeding 0.9.

Table 3.2
Discrete-analog synchronization evaluation using Pearson correlation coefficient

Chaotic oscillator	x_1 **and** x_2	y_1 **and** y_2	z_1 **and** z_2
Vilnius	0.99	—	0.97
RC	0.99	—	0.99
Colpitts	—	0.99	0.99

3.3 Study Based on HiL

The next step in investigating Pecora–Carroll hybrid chaotic synchronization involves replacing the continuous-time numerical model with physical hardware prototypes implemented on a PCB. By interfacing the PCB-based oscillator with a discrete-time MATLAB model via a data acquisition system, this setup establishes a HiL framework. Consistent with previous studies, both analog-discrete and discrete-analog synchronization configurations are evaluated.

To maintain signal compatibility across the hybrid interface, normalization transformations -— previously derived for the discrete-time models —- were applied to the raw signals acquired from the hardware before being processed in MATLAB. Conversely, the discrete signals generated in MATLAB were inverse-transformed to restore the physical voltage and time scales before being injected into the hardware circuit. This bidirectional mapping ensures that the state variables remain within the operational range of both the digital algorithm and the analog components.

Analog-Discrete Synchronization

The block diagram for investigating the possibility of analog-discrete synchronization is outlined in Fig. 3.6. The y_1 state variable of the hardware-based drive oscillator substitutes y_2 of the MATLAB-based response discrete-time oscillator. As a result, x_2 should repeat the behavior of x_1, and z_2 should repeat the behavior of z_1, evaluated using Pearson correlation.

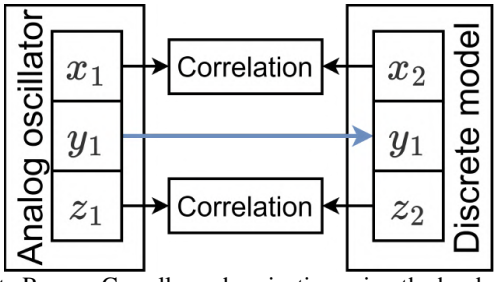


Fig. 3.6. Analog-discrete Pecora–Carroll synchronization using the hardware and discrete model.

The experimental setup for performing the study in the case of the Vilnius oscillator is shown in Fig. 3.7. The ADP3450 is used to record the samples of the analog circuit's state variables. The

current is obtained from the voltage across the resistor. The recorded signals are then transformed, and the synchronization signal is applied to the discrete model implemented in MATLAB. Finally, the correlation is computed in MATLAB.

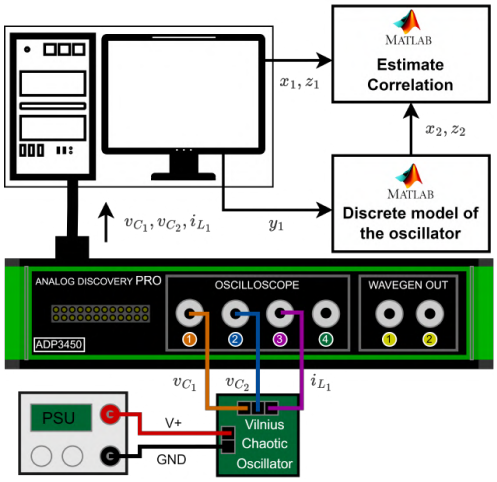


Fig. 3.7. Experimental setup for analog-discrete synchronization of the Vilnius chaotic oscillators.

The time diagrams of the x_1 and x_2, z_1 and z_2 in the case of the Vilnius chaotic oscillator are presented in Fig.3.8. The synchronization signal y_1 is applied at a discrete time $\theta = 100$. Time diagrams show that the two chaotic systems are initially asynchronous, but after the synchronization signal is applied, they become synchronized. Despite differences in amplitude and mean value, the similarity in behavior is clear.

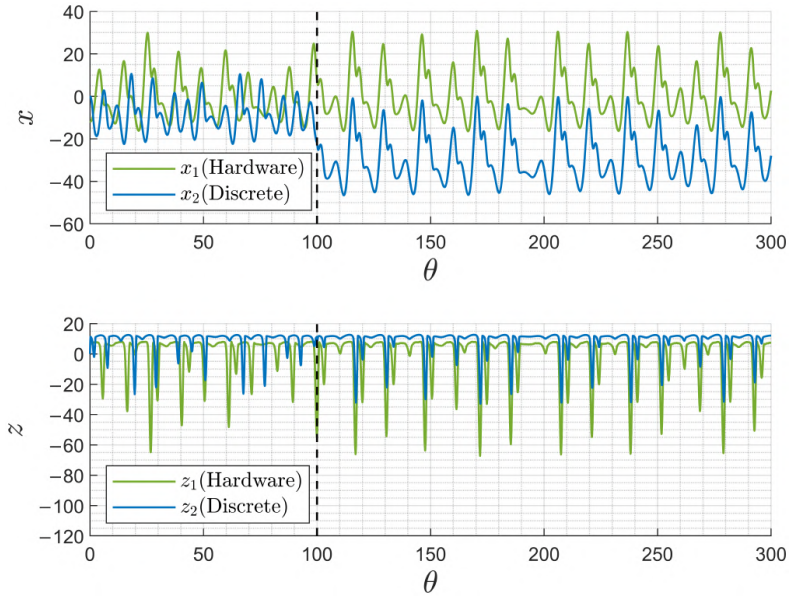


Fig. 3.8. Time diagrams of the synchronized signals in the case of Vilnius chaotic oscillator analog-discrete synchronization.

The results of computing the 10^6 sample Pearson correlation for the analog-discrete synchronization are shown in Table 3.3. The results compiled in Table 3.3 confirm that the analog-discrete chaotic synchronization was achieved, indicated by the Pearson correlation coefficient values exceeding 0.9 for all considered state variables of the chaotic oscillators.

Table 3.3
Analog-discrete synchronization evaluation using Pearson correlation coefficient

Chaotic oscillator	x_1 **and** x_2	y_1 **and** y_2	z_1 **and** z_2
Vilnius	0.99	—	0.95
RC	0.97	—	0.99
Colpitts	—	0.98	0.99

Discrete-Analog Synchronization

The block diagram for investigating the possibility of discrete-analog synchronization is outlined in Fig. 3.9. The y_2 of the hardware circuit substitutes the y_1 of the discrete model. This results in x_1 repeating the behavior of x_2 and z_1 repeating the behavior z_2. Like in the previous study, the synchronization quality is evaluated by computing the Pearson correlation for x_1 and x_2, z_1 and z_2.

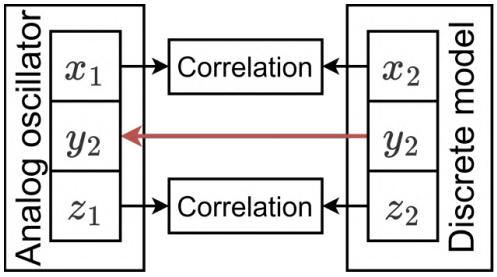


Fig. 3.9. Discrete-analog Pecora–Carroll synchronization using the hardware and discrete model.

The experimental setup for performing the study is shown in Fig. 3.10. First, the chaotic signals x_2, y_2, and z_2 are taken from the discrete model implemented in MATLAB and transformed back to voltages. Then synchronization signal y_2 is applied to the chaotic oscillator via the arbitrary waveform generator of the ADP3450. The applied voltage signal substitutes the corresponding state variable, thus performing the Pecora-Carroll synchronization. The oscilloscope channels of the ADP3450 record the voltages of the circuit. Finally, correlation is computed in MATLAB for the recorded signals.

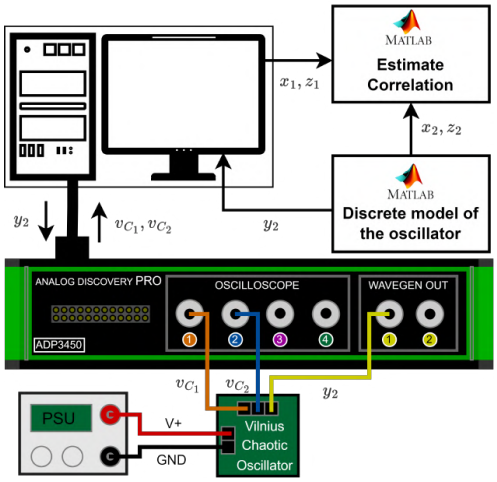


Fig. 3.10. Experimental setup for discrete-analog synchronization of the Vilnius chaotic oscillators.

The results of the discrete-analog synchronization are displayed in Fig. 3.11. In this case, the displayed x_1 and x_2, z_1 and z_2 signals are voltages, and the figure shows signals after the synchronization was applied. The moment of synchronization was not recorded, as it required a different experimental setup. Nevertheless, the time diagrams show that the behavior of the hardware circuit repeats that of the discrete model. As in the previous study, the signals are not identical and have amplitudes and mean value mismatches.

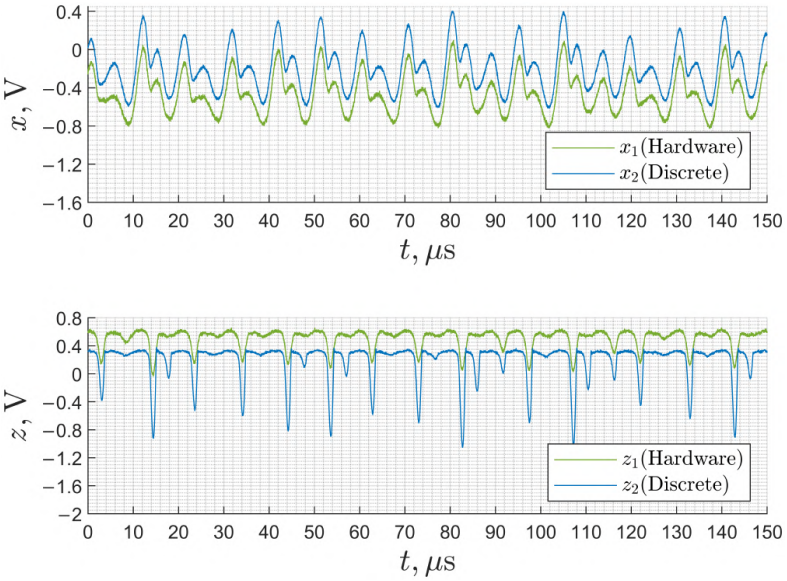


Fig. 3.11. Time diagrams of the synchronized signals.

The results of computing the 10^6 sample Pearson correlation for the discrete-analog synchronization are shown in Table 3.4. Pearson correlation coefficient in all cases equals or exceeds 0.9, thus confirming the high quality of the discrete-analog synchronization.

Table 3.4
Discrete-analog synchronization evaluation using Pearson correlation coefficient

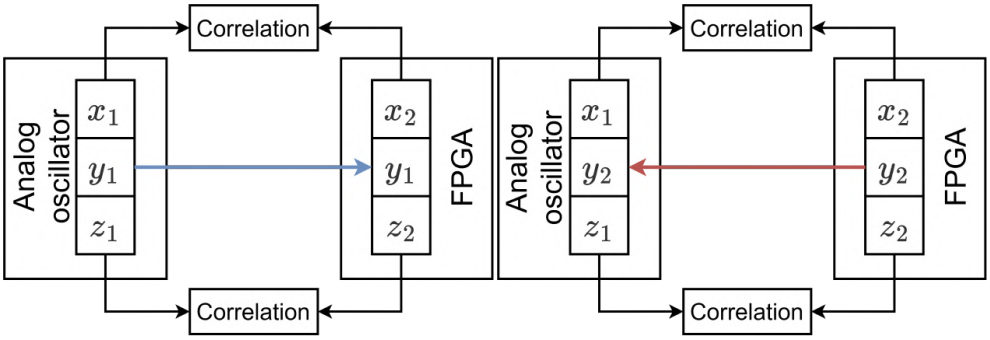
Chaotic oscillator	x_1 **and** x_2	y_1 **and** y_2	z_1 **and** z_2
Vilnius	0.99	—	0.90
RC	0.99	—	0.99
Colpitts	—	0.98	0.99

3.4 Full Hardware Integration

This subsection is devoted to verifying analog-discrete and discrete-analog synchronization using the analog chaotic oscillators implemented on PCB and FPGA-based discrete chaotic oscillators. The synchronization configurations of the studies are depicted in Fig. 3.12. The state variables of the analog oscillator are x_1, y_1, and z_1, while the state variables of the FPGA-based chaotic oscillator are x_2, y_2, and z_2.

In Fig. 3.12 (a), the analog oscillator is a drive system, while the FPGA-based chaotic oscillator is a response system. Synchronization is achieved by passing the y_1 state variable to the response system and replacing y_2. By doing this, x_2 starts following the behavior of x_1, and z_2 starts following the behavior of z_1. A discrete-analog synchronization is shown in Fig. 3.12 (b), where the FPGA-

based oscillator is a drive system, and the analog oscillator is a response system. Passing the y_2 to the response system and replacing y_1, x_1 starts following the behavior of x_2, and z_1 starts following the behavior of z_2. Signal similarity is evaluated by calculating the Pearson correlation between the corresponding signals.



(a) (b)

Fig. 3.12. Analog-discrete (a) and discrete-analog (b) Pecora–Carroll synchronization.

Fig. 3.13 demonstrates the experimental evaluation of discrete-analog synchronization. The FPGA-based chaotic oscillator operates on a DE10-Standard board. An ADA-HSMC daughter card is connected via a HSMC connector, providing a two-channel ADC and a two-channel DAC. The first DAC channel outputs a y_2 synchronization signal applied to the analog chaotic oscillator. The second DAC channel outputs x_2 in the first set of measurements and z_2 in the second set of measurements. ADP3450 oscilloscope channels are used to record and save x_1, x_2, and z_1, z_2. Next, the Pearson correlation between the corresponding signals is computed in MATLAB.

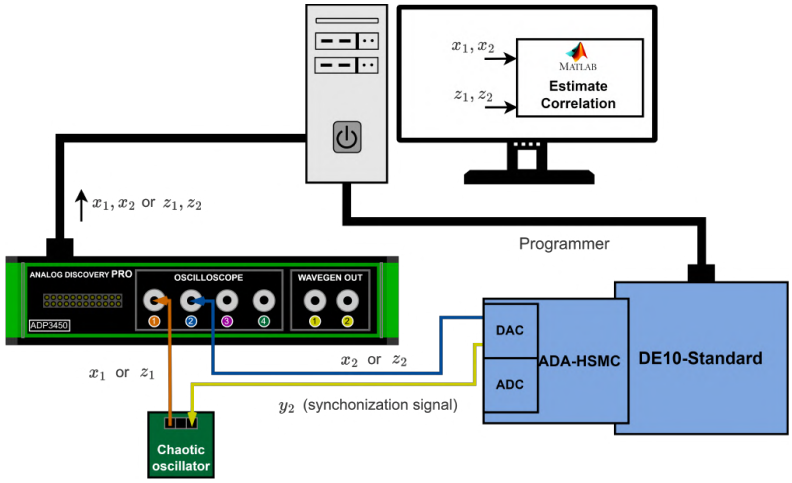


Fig. 3.13. Experimental verification of discrete-analog chaotic synchronization.

Fig. 3.14 demonstrates the experimental evaluation of analog-discrete synchronization. In this

case, y_1 signal is passed to the ADC input and forwarded then to the FPGA-based oscillator. FPGA signals x_2 and z_2 are passed to DAC. In this case, the ADP3450 oscilloscope channels record x_1, x_2 and z_1, z_2 simultaneously. The correlation is also estimated in MATLAB.

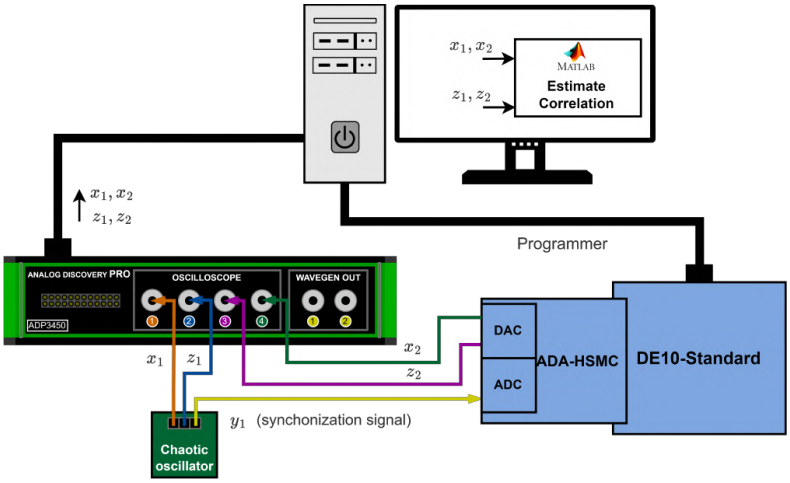
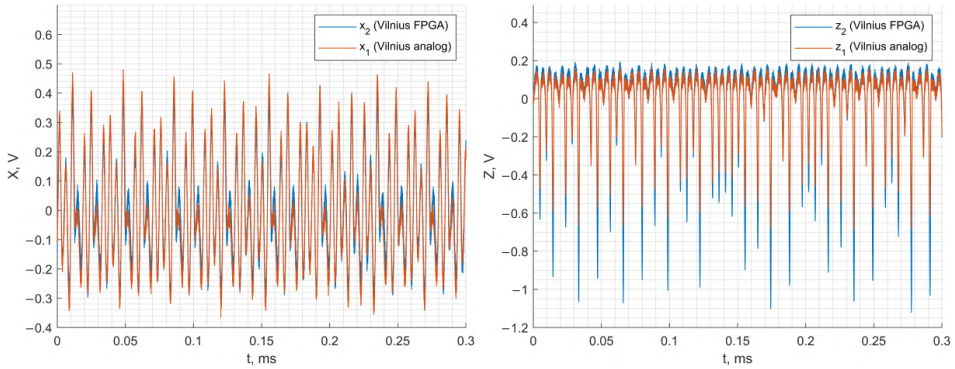


Fig. 3.14. Experimental verification of analog-discrete chaotic synchronization.

Fig. 3.15 demonstrate the recorded waveforms of the Vilnius chaotic oscillators with applied synchronization in the case of discrete-analog synchronization. The figures demonstrate that with the synchronization signal applied to the analog oscillator from the FPGA board, the analog oscillator behavior matches the behavior of the FPGA-based oscillator. In the case of the Vilnius chaotic oscillator, the x_1, x_2 and z_1, z_2 signals show very close agreement.

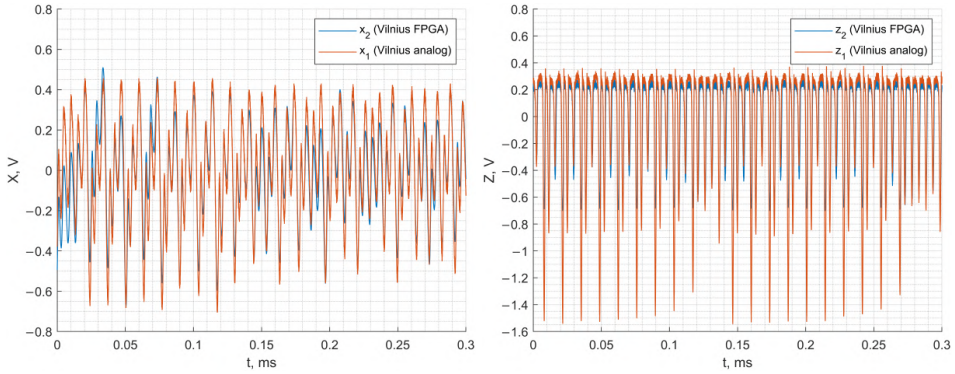


(a) (b)

Fig. 3.15. Discrete-analog synchronization of Vilnius chaotic oscillator x (a) and z (b) state variables.

Fig. 3.16 demonstrate the recorded waveforms of the Vilnius chaotic oscillators with applied synchronization in the case of analog-discrete synchronization. The figures demonstrate that with the

synchronization signal applied to the FPGA-based oscillator from the analog oscillator, the FPGA-based oscillator's behavior matches to the analog oscillators. The x_1, x_2 and z_1, z_2 signals match, demonstrating successful synchronization.



(a) (b)

Fig. 3.16. Analog-discrete synchronization of Vilnius chaotic oscillator x (a) and z (b) state variables.

Tables 3.5 and 3.6 compiles the correlation coefficients calculated for every Vilnius, RC and Colpitts chaotic oscillator state variable in the discrete-analog and analog-discrete synchronization cases. The results show that in most cases, the correlation coefficient is over 0.9 and very close to 1, thus confirming that analog-discrete and discrete-analog synchronization was achieved between analog and FPGA-based chaotic oscillators. The fact that the signals' coincidence is not perfect can be attributed to the mismatch of the parameters in the analog and FPGA-based oscillators. Another contributor to this is noise, whose influence is further examined in the next section.

Table 3.5
Synchronization evaluation using correlation coefficient

	Discrete-analog		**Analog-discrete**	
Chaotic oscillator	x_1 **and** x_2	z_1 **and** z_2	x_1 **and** x_2	z_1 **and** z_2
Vilnius	0.97	0.98	0.93	0.92
RC	0.72	0.97	0.89	0.90

Table 3.6
Synchronization evaluation using correlation coefficient

	Discrete-analog		**Analog-discrete**	
Chaotic oscillator	y_1 **and** y_2	z_1 **and** z_2	y_1 **and** y_2	z_1 **and** z_2
Colpitts	0.93	0.97	0.91	0.96

3.5 Discussion

In this section, the possibility of achieving hybrid analog-discrete and discrete-analog chaotic synchronization was successfully demonstrated across multiple platforms and oscillator topologies. Based on the experimental and simulation results, the following conclusions can be drawn:

- **Robustness of the Pecora–Carroll method:** The ”substitution method” proved effective for synchronizing hybrid systems. High correlation coefficients (typically $r > 0.9$) were consistently achieved for the Vilnius, RC, and Colpitts oscillators, regardless of whether the drive system was analog or discrete.
- **Cross-domain compatibility:** The study confirmed that by applying appropriate scaling transformations to account for voltage-to-dimensionless units and continuous-to-discrete time mapping, a mathematical discrete model can accurately track the dynamics of a physical circuit.
- **Hardware and FPGA realization:** The successful synchronization between PCB-based circuits and FPGA implementations validates the practical utility of these systems. While small discrepancies in amplitude and mean values were observed—attributed to parameter mismatches, and ADC/DAC quantization—the fundamental chaotic trajectories remained highly correlated.
- **Metric validity:** The use of the Pearson correlation coefficient provided a reliable measure of synchronization quality, as it remained invariant to the constant offsets and mean-value shifts inherent in analog-to-digital signal conversion.

Overall, the results presented in this chapter establish a solid foundation for the development of hybrid chaotic communication systems, where both analog and discrete chaotic oscillators are utilized.

4 PERFORMANCE OF ANALOG-DISCRETE AND DISCRETE-ANALOG CHAOTIC SYNCHRONIZATION

This chapter investigates the performance of Pecora–Carroll chaotic synchronization in hybrid configurations, primarily analog-discrete and discrete-analog. The analysis focuses on two key performance aspects: noise immunity and synchronization speed. These aspects are critical for evaluating the robustness and practical applicability of chaotic synchronization in real-world environments where noise and system imperfections are unavoidable.

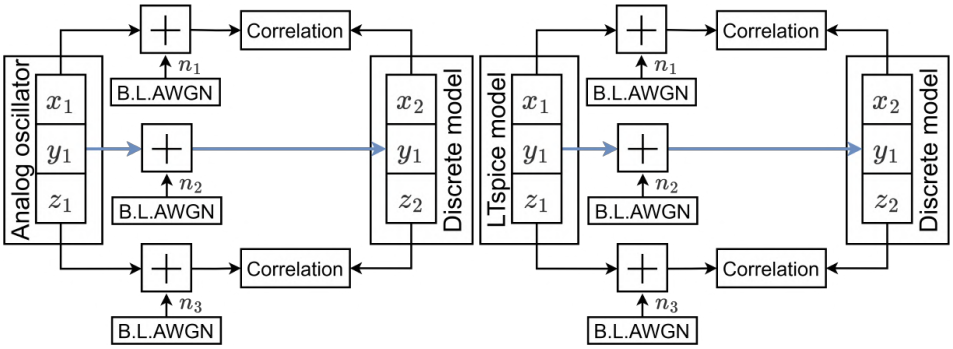
The studies are conducted using both simulation and experimental approaches, combining LTspice models, discrete-time MATLAB models of the oscillators, and hardware prototypes of the analog oscillators. The Vilnius and RC oscillators are considered to provide a comprehensive comparison across different dynamical behaviors. The first part of the section evaluates the impact of additive noise on synchronization quality, while the second part examines synchronization and desynchronization times under varying noise conditions. Together, these analyses provide insight into the reliability, stability, and implementation trade-offs of hybrid chaotic synchronization configurations.

4.1 Synchronization Noise Immunity

The first study on the performance of hybrid Pecora–Carroll synchronization is synchronization noise immunity. This study aims to assess the influence of noise on the quality of chaotic synchronization in each configuration. The following subsections elaborate more on the simulation (using the LTspice and discrete models) and experimental (using the hardware and discrete models) setups and analyze the acquired results. Unlike the previous study, this section considers only Vilnius and RC chaotic oscillators.

Analog-Discrete Synchronization

The first part of this study evaluates the noise immunity of analog-discrete synchronization. Fig. 4.1 illustrates the experimental setup. The drive chaotic oscillator is implemented either as an LTspice model or as a physical analog oscillator, while the response oscillator is represented by a discrete model. Additive B.L.AWGN is introduced into the chaotic signals of the drive oscillator. Each noise component is generated independently for each chaotic signal by applying low-pass filtering with a cutoff frequency matched to the bandwidth of the corresponding chaotic signal. This approach is based on the assumption that noise components overlapping with the signal bandwidth have the most significant impact on synchronization.



(a) (b)

Fig. 4.1. Chaotic oscillator's discrete model synchronization with LTspice model (a), and analog oscillator (b).

The injected noise affects the synchronization quality, leading to its degradation as the noise power increases relative to the signal power. The signal-to-noise ratio (SNR) metric is used in this study and is defined as:

$$\mathrm{SNR} = 10 \cdot \log_{10}\left(\frac{P_{\mathrm{signal}}}{P_{\mathrm{noise}}}\right), \tag{4.1}$$

where P_{signal} and P_{noise} are the average powers of the signals.

Fig. 4.2 presents the implementation of noise immunity analysis for discrete model synchronization using LTspice-generated signals. The LTspice outputs are exported to MATLAB, where they are resampled and appropriately scaled. Subsequently, noise corresponding to a predefined SNR is added to the signals. The perturbed synchronization signal, $y_1 + n_2$, is then applied to the discrete model. Finally, the correlation is evaluated between the signal pairs x_2 and $x_1 + n_1$, as well as z_2 and $z_1 + n_3$.

Fig. 4.3 shows the experimental evaluation of noise immunity for synchronization with a physical analog oscillator. The chaotic signals are acquired using ADP3450 oscilloscope channels; the figure presents an example based on the Vilnius chaotic oscillator. As in the LTspice case, the recorded signals are resampled and scaled before being used as inputs to the discrete model, after which the synchronization performance is quantified via correlation analysis.

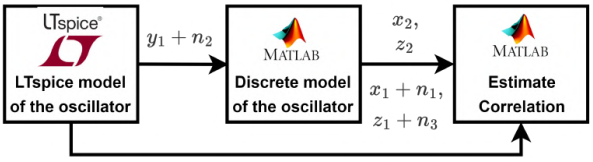


Fig. 4.2. Implementation of the discrete model synchronization with LTspice model.

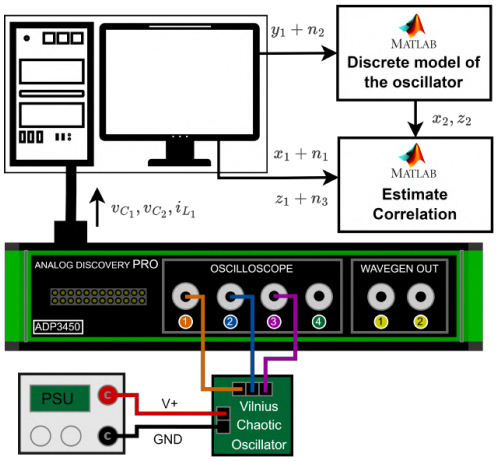
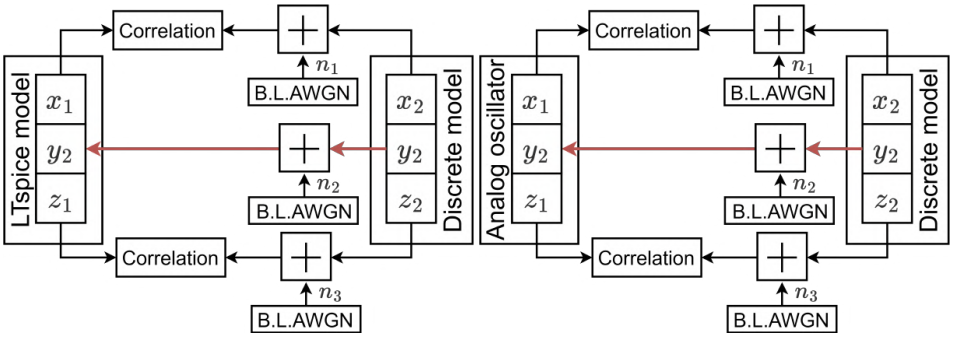


Fig. 4.3. Implementation of the discrete model synchronization with analog oscillator.

Discrete-Analog Synchronization

The second part of this study focuses on evaluating the noise immunity of discrete–analog synchronization. Fig. 4.4 illustrates the experimental setup. In this case, the drive chaotic oscillator is implemented as a discrete model, while the response oscillator is realized either as an LTspice model or as a physical analog oscillator. As in the first part of the study, B.L.AWGN is added to the chaotic signals of the drive oscillator.



(a) (b)

Fig. 4.4. Chaotic oscillator's LTspice model (a), and hardware prototype (b) synchronization with discrete model.

Fig. 4.5 presents the evaluation of noise immunity for synchronization between the discrete model and the LTspice implementation. The discrete signals are first resampled and scaled in MATLAB. Noise corresponding to a specified SNR is then added before exporting the signals to LTspice. For correlation analysis, all signals are subsequently rescaled to match the parameters of the discrete model.

Fig. 4.6 demonstrates synchronization between the discrete model and the hardware prototype under noisy conditions. In this setup, the signal generator channel of the ADP3450 is used to transmit the noisy synchronization signal from the discrete model to the analog chaotic oscillator. The resulting chaotic signals are captured using the oscilloscope channels. As in the previous cases, the recorded hardware signals are appropriately scaled before evaluating their correlation with the noisy discrete model signals.

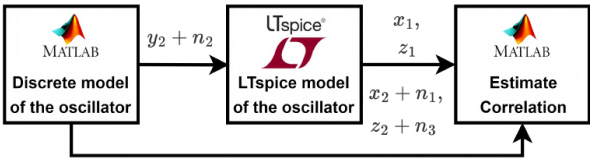


Fig. 4.5. Implementation of the LTspice model synchronization with the discrete model noise immunity.

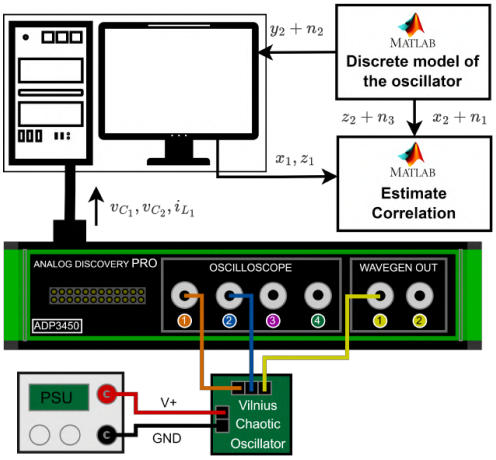


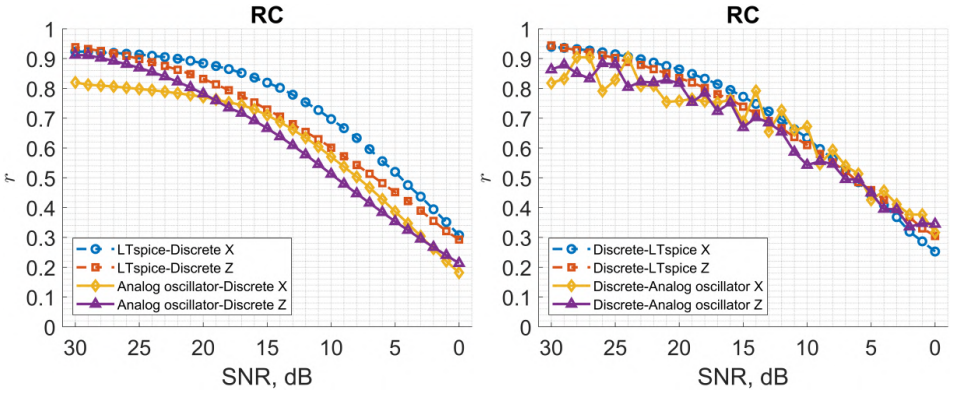
Fig. 4.6. Implementation of the hardware prototype synchronization with the discrete model noise immunity.

Study Results

This subsection summarizes the results of the analog-discrete and discrete-analog synchronization noise immunity study; the results are grouped by the oscillators. The results using the approximated curves for the RC chaotic oscillator are presented in Fig. 4.7 and the results for the Vilnius chaotic oscillator are presented in Fig. 4.8. As expected, the correlation level is mostly 0.8 to 0.95 for 30 dB SNR and the correlation level decreases with the decrease in SNR (i.e., increase of the noise level) down to 0.1–0.2 for 0 dB SNR. Analyzing the results compiled in the two figures, the match between the LTspice and hardware results are exceptionally close, with the differences explained by the additional noises and mismatches introduced in hardware. The overall close match between the

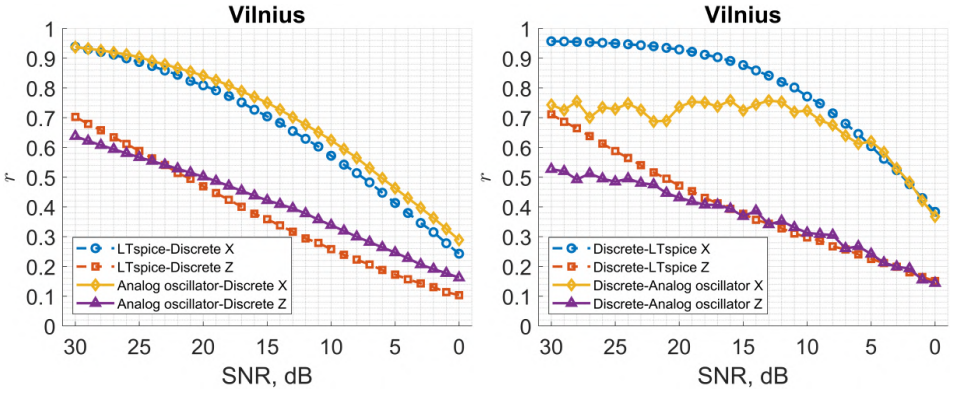
LTspice and hardware means that the LTspice is precise in terms of synchronization performance in hardware.

The RC chaotic oscillator results in Figs. 4.7 (a) and 4.7 (b)) show little difference in the behavior of x and z state variables and close match between the analog-discrete and discrete-analog chaotic synchronization performance. The Vilnius results in Figs. 4.8 (a) and 4.8 (b) show that the z state variable has lower correlation than x for the given noise levels. This means that this state variable is more sensitive to noise due to the waveform of the signal. Comparing the results of the two chaotic oscillators, the RC chaotic oscillator demonstrates better synchronization for the two state variables.



(a) (b)

Fig. 4.7. Analog-discrete (a) and discrete-analog (b) synchronization noise immunity of RC chaotic oscillator.



(a) (b)

Fig. 4.8. Analog-discrete (a) and discrete-analog (b) synchronization noise immunity of Vilnius chaotic oscillator.

Discrete-Discrete Synchronization Comparison

This subsection investigates and compares the noise immunity of discrete-discrete versus analog-discrete chaotic synchronization for both the Vilnius and RC chaotic oscillators. The performance is evaluated using the previously established methodology: the Pearson correlation coefficient r is computed across a range of SNR. The results are illustrated in Fig. 4.9 and 4.10.

As anticipated, the synchronization quality exhibits a monotonic degradation as the SNR decreases. However, the data reveals a clear performance hierarchy: discrete-discrete configurations consistently outperform their analog-discrete counterparts across the entire SNR spectrum. This superiority is primarily attributed to the perfect parametric alignment between the drive and response models in the purely discrete domain. In contrast, analog-discrete synchronization is hampered by model mismatch and the non-ideal parasitic behaviors inherent in physical analog circuit, which are not fully captured by the discrete mathematical model. Despite these practical limitations, analog-discrete synchronization remains robust, maintaining a correlation coefficient $r > 0.5$ for SNR levels above 12 dB.

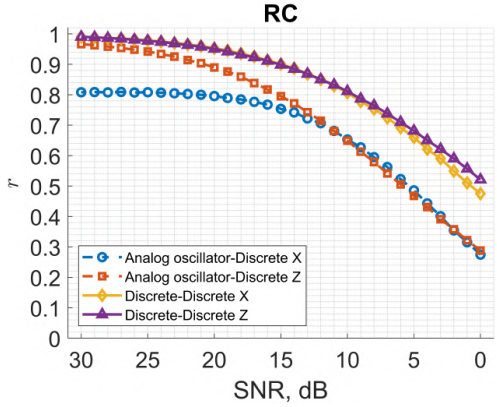


Fig. 4.9. Analog-discrete and discrete-discrete synchronization noise immunity of RC chaotic oscillator.

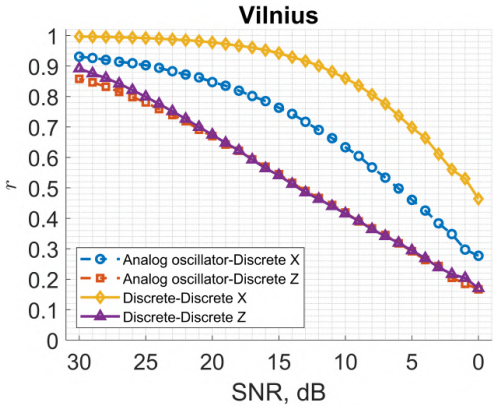


Fig. 4.10. Analog-discrete and discrete-discrete synchronization noise immunity of Vilnius chaotic oscillator.

4.2 Synchronization and De-synchronization Time

The second study on the performance of Pecora-Carroll analog-discrete and discrete-analog synchronization is synchronization and de-synchronization time. This study assesses the time required for the synchronization and de-synchronization at 30 dB SNR. The following subsections elaborate more on the study details and analyze the acquired results.

Analog-Discrete and Discrete-Analog Simulation and Experimental Setup

This study introduces a switch that controls when the synchronization signal is applied. Fig. 4.11 demonstrates the example with analog-discrete synchronization, where the synchronization signal is passed to the controlled switch before applying it to the discrete model.

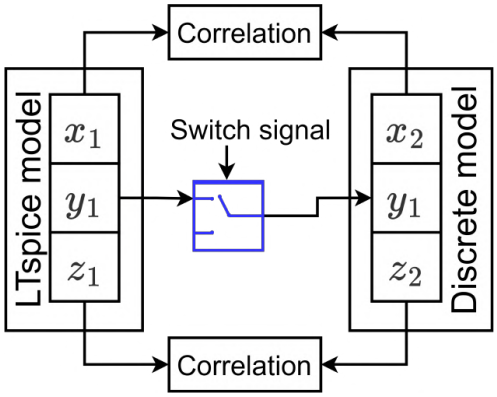


Fig. 4.11. Chaotic oscillator's discrete model synchronization with LTspice model example with the added switch to estimate synchronization and de-synchronization time.

In case the synchronization occurs in LTspice or MATLAB, only minor functional modifications

are required. The only major addition is to discrete-analog synchronization using the hardware prototype demonstrated in Fig. 4.12. The DG419 [52] analog switch circuit is introduced to perform the study. The synchronization signal from the discrete model is passed to the first input In1 of DG419 from the signal generator of ADP3450, while the second input In2 is left open. The switch control input is connected to the second signal generator channel of ADP3450. The generated 2 kHz rectangular control signal has 5 V on high and 0 V on low levels. The switch control signal is recorded on the 3rd channel of the oscilloscope to determine the start times of the synchronization and de-synchronization intervals.

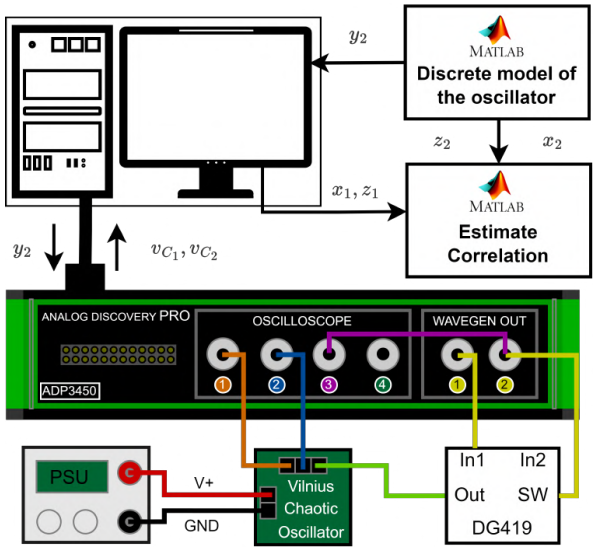


Fig. 4.12. Implementation of the hardware prototype synchronization with the discrete model for estimating synchronization and desynchronization time.

Fig. 4.13 demonstrates an example of the change in correlation coefficient over time in the case of RC physical analog prototype synchronization with the discrete model. The synchronization time is taken as the time between the start of synchronization (when the switch signal was set to 5 V) and the moment when correlation reached 80% of the maximum level of synchronization. The de-synchronization time is measured similarly—the time is taken between the end of the synchronization signal application and the moment when correlation reaches 20% of the maximum level. In every configuration the average times are estimated by performing 10 synchronization and de-synchronization trials.

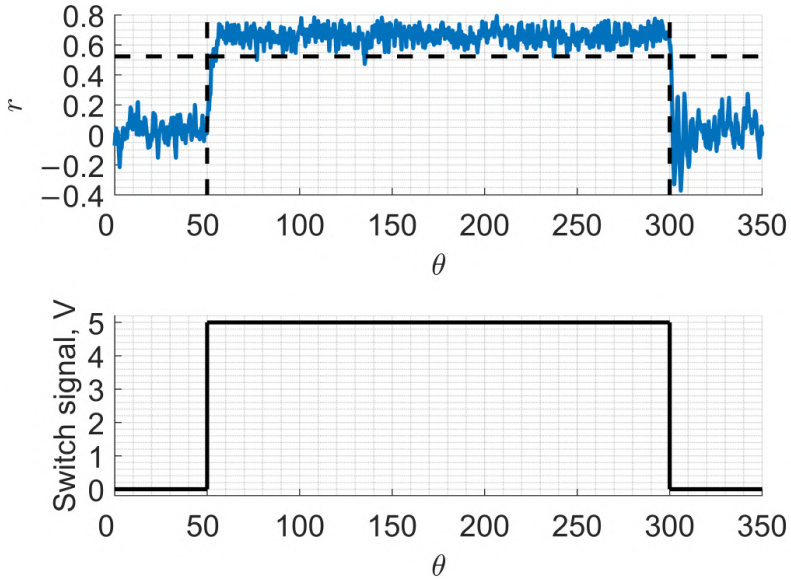


Fig. 4.13. Time diagram of window correlation of z state variables in the case of RC chaotic oscillator hardware prototype synchronization with the discrete model.

Study Results

This subsection compiles and analyzes the results of the analog-discrete and discrete-analog synchronization and de-synchronization times for the Vilnius and RC chaotic oscillators. The results in the case of analog-discrete synchronization are compiled in Table 4.1, while Table 4.2 presents the results in the case of the discrete-analog synchronization. The benefit of using the dimensionless time scale is that the results for two chaotic oscillators become comparable despite the different frequencies of oscillations.

Table 4.1 generally shows lower values, especially for the Vilnius chaotic oscillator. This suggests that the discrete-time system tracks the analog drive signal more efficiently or with less latency. Table 4.2 demonstrates significantly higher values, especially in the case of Vilnius oscillator in LTspice jump significantly (e.g., z desync rising from 0.50 to 8.74). This indicates that forcing an analog system with a discrete-time signal is inherently more ”difficult” for the system to settle, likely due to the discrete model being more simplified.

Another point apparent from the results is the simulation and the hardware disparity. In Table 4.2, in the case of Vilnius oscillator, LTspice predicts a very high de-sync value for z (8.74), while the hardware shows a much lower value (1.55). This suggests that the real-world analog circuit may have inherent damping properties that make it more resilient to de-synchronization than the idealized models in LTspice. In the case of RC chaotic oscillator x variable (Table 4.2), the hardware sync value (3.29) is over three times higher than the LTspice prediction (1.04). This implies that the hardware RC oscillator struggles much more to maintain synchronization in practice than the

simulation suggests.

Table 4.1
Analog-discrete synchronization and de-synchronization times, θ

Chaotic oscillator	Variable	LTspice		Analog oscillator	
		Sync	De-sync	Sync	De-sync
Vilnius	x	0.77	0.64	0.67	0.70
	z	1.00	0.50	1.00	0.33
RC	x	1.20	0.55	2.21	0.40
	z	0.97	0.20	1.08	0.11

Table 4.2
Discrete-analog synchronization and de-synchronization times, θ

Chaotic oscillator	Variable	LTspice		Analog oscillator	
		Sync	De-sync	Sync	De-sync
Vilnius	x	3.71	8.55	2.99	1.72
	z	2.37	8.74	4.85	1.55
RC	x	0.97	0.20	1.08	0.11
	z	1.04	1.52	3.29	0.76

4.3 Discussion

The results presented in this section demonstrate that both analog-discrete and discrete-analog chaotic synchronization schemes exhibit robust performance under noisy conditions, although their behavior differs depending on the system configuration and implementation domain. The noise immunity study confirms that synchronization quality degrades predictably with decreasing signal-to-noise ratio, while maintaining strong agreement between LTspice simulations and hardware experiments. Among the evaluated systems, the RC chaotic oscillator consistently shows higher resilience to noise compared to the Vilnius oscillator, particularly across multiple state variables.

A clear performance hierarchy is observed, with discrete-discrete synchronization providing the highest robustness due to ideal parameter matching, followed by analog-discrete and discrete-analog configurations. Despite the inherent limitations of hybrid systems – such as model mismatch and analog non-idealities – analog-discrete synchronization remains sufficiently robust for practical applications.

The analysis of synchronization and de-synchronization times further highlights fundamental differences between configurations. Analog-discrete synchronization generally achieves faster convergence, indicating efficient tracking of analog signals by discrete models. In contrast, discrete-analog synchronization exhibits slower dynamics and greater sensitivity to system imperfections, especially in experimental conditions. Notably, discrepancies between simulation and hardware results emphasize the importance of accounting for real-world effects such as parasitics and damping in analog implementations.

Overall, the findings provide a comprehensive evaluation of hybrid chaotic synchronization performance, offering valuable guidance for the design and implementation of practical systems where trade-offs between accuracy, robustness, and complexity must be carefully balanced.

5 CHAOS-BASED AUTHENTICATION

The security of modern communication systems increasingly relies on multifaceted protection layers, where physical-layer authentication (PLA) has emerged as a robust complement to traditional cryptographic methods. By exploiting the unique hardware characteristics or signal dynamics of a transmitter, PLA provides a non-algorithmic security barrier that is difficult to spoof or replicate. Among the various signal sources explored for this purpose, chaotic oscillators offer a compelling advantage due to their extreme sensitivity to initial conditions and internal parameters, alongside their inherent noise-like appearance.

This chapter focuses on the development and implementation of a hardware-efficient authentication scheme based on the Vilnius chaotic oscillator. Central to this scheme is the ability of a receiver to accurately track the internal dynamics of a transmitter using chaotic synchronization. Specifically, the digital design of a parameter estimator intended for FPGA deployment is studied. By utilizing an adaptive LMS filter architecture, the system is capable of estimating the oscillator's internal damping parameter, a.

The section details the mathematical derivation of the estimator, its pipelined digital implementation, and an extensive performance evaluation.

5.1 Chaos Oscillator Parameter Estimator-Based Hardware Authentication Scheme

The current section presents the concept of the chaos-based PLA scheme for wireless sensor network (WSN). The star-topology chaos-based WSN originally proposed in [37] is presented in Fig. 5.1. The key design element of the presented system is the use of analog chaotic oscillator circuits in each individual sensor node (SN) together with the discrete models of chaotic oscillators implemented in gateway (GW) for chaos-based wireless data transfer. Chaotic oscillators would be a cost-effective solution for an individual SN. On the GW, a single FPGA chip can host discrete models of multiple oscillators, thus making it possible to transfer information between the SNs and GW using chaotic waveforms. The previous sections demonstrated that it is possible to establish synchronization between analog chaotic oscillators and their discrete models hosted on an FPGA.

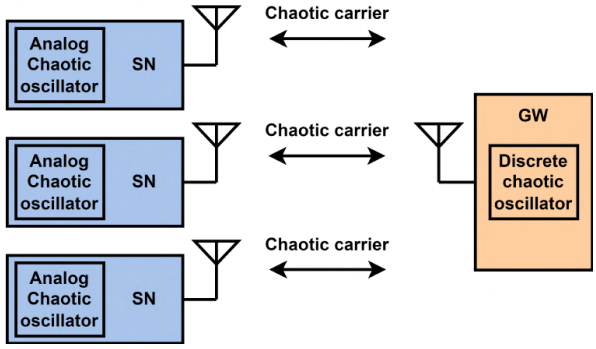


Fig. 5.1. Star-topology WSN based on application of analog and discrete chaos oscillators.

In order to understand the proposed PLA scheme, it is important to outline the key design elements of the chaos-based data transfer scheme shown in Fig. 5.2, which represents a generalized architecture based on works [53], [54]. The transmitter hosts the drive chaos oscillator with state variables x_1, y_1 and z_1. The state variable y_1 is used to establish chaotic synchronization using Pecora-Carroll approach [50]. The binary information signal $b(t)$ is transferred by applying a chaos shift keying (CSK) technique. The figure shows that the transmitted information is encoded by applying phase shift keying (PSK) to the chaotic state variable x_1, as in [54]. The transmitter then applies modulation to transfer the information-carrying and synchronization signals to the receiver through the wireless channel. Alternatively, the chaos-based data transmission system in [53] transfers the binary information signal $b(t)$ by CSK using x_1 and z_1, yet the core elements of the system are similar.

The receiver hosts the response chaotic oscillator with state variables x_2, y_2 and z_2. The demodulator outputs the received y_1 and information-carrying x_1. By Pecora-Carroll synchronization, the y_1 replaces the y_2 of the receiver's oscillator, thus establishing synchronization. When synchronized, x_2 and x_1, as well as z_2 and z_1 signals match. The detection of the received binary information signal $b'(t)$ is done by calculating Pearson's correlation coefficient using x_2 and the received information-carrying x_1. In the context of chaos-based WSN, the transmitter's chaotic oscillator is implemented as a circuit on PCB, and the receiver's chaotic oscillator is implemented as a discrete model on a FPGA.

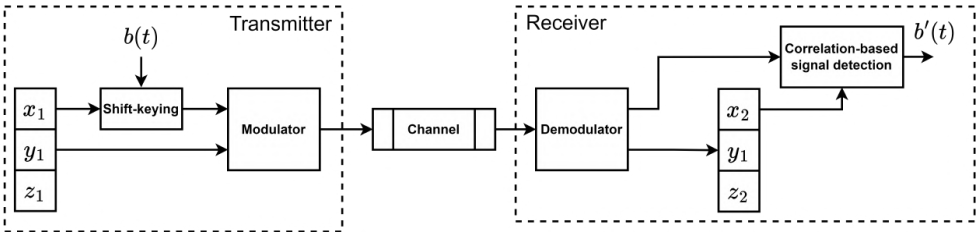


Fig. 5.2. Generalized chaos-based data transfer scheme.

The key enabling factor for such data transfer is the ability of chaotic oscillators to synchronize. The use of chaotic carriers alone enhances the security aspect on the hardware level, yet chaotic synchronization alone is not enough to fully secure the data transfer. The receiver can easily synchronize and receive data from another, possibly unauthorized transmitter with a similar oscillator. This leads to a necessity to supplement the previously proposed chaos-based data transfer schemes with a mechanism for distinguishing between different transmitters.

Chaotic synchronization is highly robust; data transmission requires two synchronized chaotic oscillator signals, and the oscillator dynamics are governed by parameterized equations directly related to the chaotic waveforms. Therefore, the proposed authentication scheme relies on estimating at least one parameter of the transmitter's chaotic oscillator at the receiver. The detailed outline of the proposed authentication scheme is presented in Fig. 5.3.

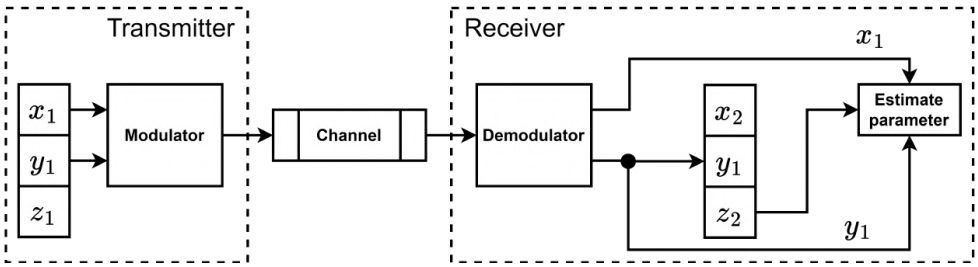


Fig. 5.3. Proposed chaos-based authentication scheme.

The figure demonstrates the transmitter sending the x_1 and y_1 signals to the receiver. Like in the data transfer case, the y_1 is used for Pecora-Carroll synchronization. In the receiver, the received x_1 and y_1 as well as the signal from the local chaotic oscillator z_2 are passed to the block that estimates one of the parameters of the transmitter's chaotic oscillator. The use of chaotic synchronization allows using z_2 that is in phase with z_1, thus there is no need to transmit all three chaotic signals in order to estimate the transmitter's oscillator parameter. This system can supplement the existing proposed modulation schemes in Fig. 5.2, enabling the identification of the transmitter's chaotic oscillator based on a selected control parameter. This estimation can be performed before the message transfer. Because the signals used in the estimation are transmitted over a public channel, the parameter must be dynamically adjusted according to a pre-agreed pattern, thereby preventing the data from being received from an unauthorized transmitter with a similar architecture.

The parameter estimation mechanism exploits the stable synchronization regime: once the state synchronization error is driven to zero by Pecora-Carroll synchronization, this enables the convergence of the estimator to a stable value. Although the present approach does not employ adaptive laws for synchronization as in adaptive synchronization schemes, its robustness relies on a similar feedback-driven error correction principle in the estimator itself. In this sense, the presented estimator shares conceptual similarities with adaptive synchronization [22], [55].

The following sections further elaborate on the implementation and performance evaluation of the oscillator parameter estimator in the case of the Vilnius chaotic oscillator.

5.2 Vilnius Chaotic Oscillator Parameter Estimator

The first step of designing the parameter estimator is to select the parameter of the chaotic oscillator that will be estimated based on the difference equations. The difference equation of the Vilnius chaotic oscillator with highlighted operations:

$$\begin{cases} \underbrace{x[n+1]}_{\text{Next value}} = \underbrace{x[n]}_{\text{Current value}} + \underbrace{y[n]}_{\text{Derivative}} \cdot \underbrace{\Delta\theta}_{\text{Time step}} \\ \underbrace{y[n+1]}_{\text{Next value}} = \underbrace{y[n]}_{\text{Current value}} + \underbrace{(a \cdot y[n] - x[n] - z[n])}_{\text{Derivative}} \cdot \underbrace{\Delta\theta}_{\text{Time step}} \\ \underbrace{z[n+1]}_{\text{Next value}} = \underbrace{z[n]}_{\text{Current value}} + \underbrace{\left(\frac{b}{\varepsilon} + \frac{1}{\varepsilon} \cdot y[n] - \frac{c}{\varepsilon} \cdot (e^{z[n]} - 1)\right)}_{\text{Derivative}} \cdot \underbrace{\Delta\theta}_{\text{Time step}} \end{cases} . \tag{5.1}$$

By observing the system (5.1), the most convenient parameter for estimation is a from the second term of the difference equations:

$$y[n+1] = y[n] + (a \cdot y[n] - x[n] - z[n]) \cdot \Delta\theta. \tag{5.2}$$

From this equation, the parameter a can be expressed as:

$$a_{estimated}[n] = \frac{\frac{y[n+1]-y[n]}{\Delta\theta} + x[n] + z[n]}{y[n]}. \tag{5.3}$$

The convenience is that the equation contains only one parameter and that the signal relations are linear, unlike other terms of the system (5.1). Relating to the analog circuit, the parameter a is defined as:

$$a = (k-1)\frac{R_1}{\rho}, \qquad \rho = \sqrt{\frac{L_1}{C_1}}, \tag{5.4}$$

where k is the closed-loop gain of the operational amplifier, and R_1 is the series resistor in the resonant loop. The variable ρ represents the characteristic impedance of the L_1C_1 resonant circuit.

The parameter a directly influences the effective damping of the resonant subsystem and therefore governs the qualitative behavior of the oscillator. In the circuit, this parameter can be adjusted either via the gain k or the resistor R_1. As demonstrated in earlier studies [47], varying a yields transitions between periodic oscillations, period-doubling cascades, and fully developed chaos.

Digital Implementation of the Parameter Estimator

The digital design of the a parameter estimator is divided into two components: first, the numerator is acquired from (5.3):

$$s[n] = \frac{y[n+1] - y[n]}{\Delta\theta} + x[n] + z[n], \tag{5.5}$$

second, the estimated value of a is acquired from the expression:

$$a_{estimated}[n] = \frac{s[n]}{y[n]}. \tag{5.6}$$

The digital implementation of the first component is presented in Fig. 5.4. The digital design is a pipeline set to work in Fixed-point arithmetic. The $s[n]$ is acquired by inputting the $x[n]$, $z[n]$, $\frac{y[n]}{\Delta\theta}$ and performing the required operations as described in (5.5). Because the time step is a multiple of two, the $\frac{y[n]}{\Delta\theta}$ division can be done with bit shifting. The acquisition of the $s[n]$ takes 4 registers, resulting in a 4 clock cycle delay. As shown in (5.6), the second component requires $s[n]$ and $y[n]$ in the same phase, so a digital delay line for delaying the $y[n]$ for 4 clock cycles is added, giving the $y_{delayed}[n]$ output signal.

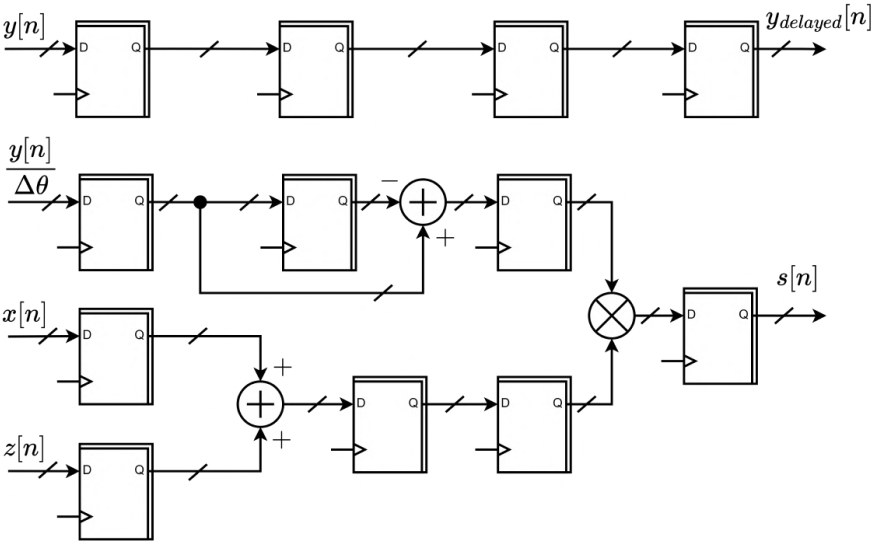


Fig. 5.4. Pipeline design for acquiring the numerator par of the a parameter estimator.

The digital implementation for the second component is not as straightforward because division of the two signals in the digital system cannot be done in a single clock cycle and requires an algorithm. The solution to acquire $a_{estimated}[n]$ is through the adaptive filter LMS (Wiener filter with minimum mean squared error (MMSE)). The equation (5.6) with the $y_{delayed}[n]$ can be expressed as

$$a_{estimated}[n] \cdot y_{delayed}[n] = s[n]. \tag{5.7}$$

Defining the cost function as a squared error, the following is obtained:

$$\mathcal{J}[n] = \mathbb{E}\left[\left(s[n] - a_{estimated}[n] \cdot y_{delayed}[n]\right)^2\right]. \tag{5.8}$$

Next, the equation is differentiated with respect to $a_{estimated}[n]$, resulting in

$$\frac{\mathrm{d}\mathcal{J}[n]}{\mathrm{d}a_{estimated}[n]} = -2 \cdot y_{delayed}[n] \cdot \left(s[n] - a_{estimated}[n] \cdot y_{delayed}[n]\right). \tag{5.9}$$

Thus, $a_{estimated}[n]$ can be assessed using the stochastic gradient as

$$a_{estimated}[n+1] = a_{estimated}[n] - \mu \frac{\mathrm{d}\mathcal{J}[n]}{\mathrm{d}a_{estimated}[n]}. \tag{5.10}$$

The final equation for the digital implementation is acquired by applying (5.9) in (5.10), resulting in

$$a_{estimated}[n+1] = a_{estimated}[n] + 2\mu \cdot y_{delayed}[n] \cdot \left(s[n] - a_{estimated}[n] \cdot y_{delayed}[n]\right). \tag{5.11}$$

The constant μ is the step size (also known as the convergence speed). This constant depicts how fast the algorithm converges to the Wiener MMSE solution. This constant needs to be selected accordingly, as setting μ too low results in slow convergence, while setting it too high produces output oscillations. A sufficient condition for mean-square stability is

$$0 < \mu < \frac{2}{\mathbb{E}[y[n]^2]}. \tag{5.12}$$

Fig. 5.5 presents the digital implementation of the LMS adaptive filter (5.11). The first two registers are the output registers of the numerator calculation pipeline from Fig. 5.4, so they output the $s[n]$ and $y_{delayed}[n]$ for the LMS adaptive filter. The filter is a pipeline feedback structure that performs the mathematical operations defined in (5.11) with additional delays to ensure that the operations are performed on the correct signal samples. A critical design aspect is the aforementioned convergence speed μ and multiplication with 2μ. In the system, 2μ is set to 1×10^{-5}. The system is implemented in fixed-point arithmetic, so the 2μ constant is floored, making it a multiple of two and allowing multiplication to be replaced with bit-shifting. Because the 2μ is a very small constant, the output $a_{estimated}[n]$ requires 20 fractional bits. The expected integer range is from $0 < a_{estimated}[n] < 2$, so devoting 2 integer bits would be enough, yet the system has 4 integer bits for some headroom.

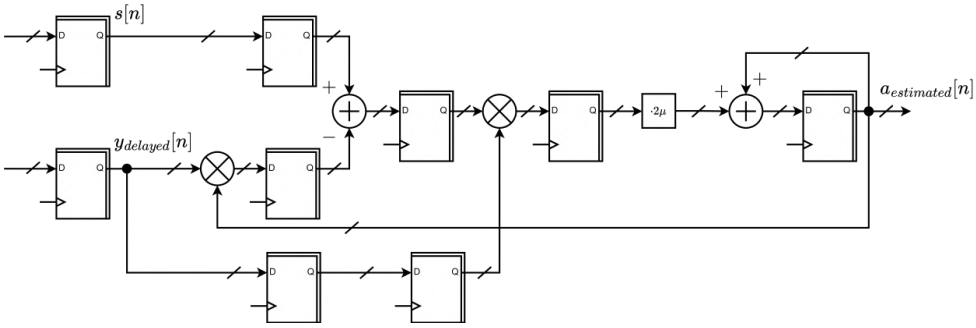


Fig. 5.5. The digital implementation of the LMS adaptive filter for parameter a estimation.

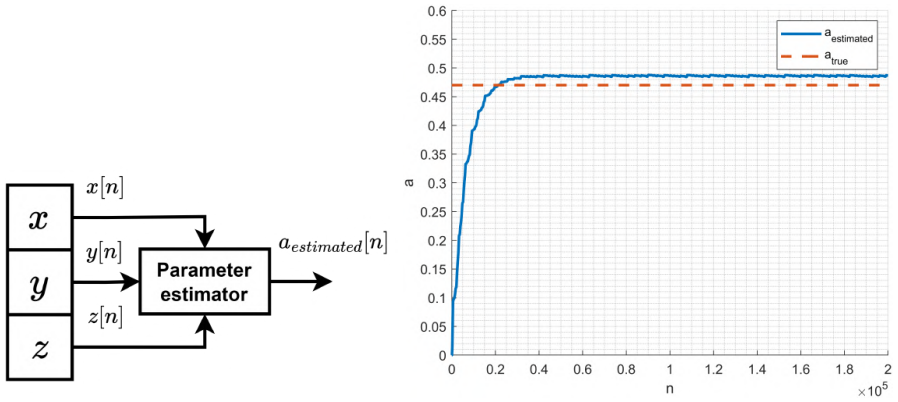
The FPGA resource utilization of the presented estimator's design, together with the resource utilization of the Vilnius chaos oscillator, is presented in Table 5.1. The design was compiled for an Altera (Intel) Cyclone 5CSXFC6D6F31C6 chip. To provide vendor-independent comparison, Intel ALMs are converted to equivalent LUTs assuming 1 ALM $\approx$ 2 four-input LUTs, following the Cyclone V ALM architecture.

Table 5.1
FPGA resource utilization of the estimator together with Vilnius chaos oscillator.

Resource	**Utilization**	**Available**	**Utilization, %**
LUT	614	83820	0.7
BRAM	90122	5662720	1.6
FF	550	166000	0.33
DSP	6	112	5.36

Performance Evaluation

This subsection is devoted to studying the performance of the designed parameter estimator for the Vilnius chaotic oscillator. The first test of the estimator is demonstrated in Fig. 5.6. The block diagram in Fig. 5.6 (a) shows that the signals $x[n]$, $y[n]$, and $z[n]$ of the discrete oscillator are passed to the estimator. The parameter a of the oscillator is set to 0.47 and is kept constant. The output of the estimator is presented in Fig. 5.6 (b). The initial value of $a_{estimated}[n]$ was set to 0, and it is seen that the $a_{estimated}[n]$ is rapidly approaching the stable value with insignificant ripples, reaching it after 4×10^4 clock cycles. The diagram also displays the true value of the a parameter, overlaying it as the level with a dashed line. The estimator's output $a_{estimated}[n]$ level equals 0.485, which does not match exactly the transmitter's a parameter. This is because the LMS does not converge to the true physical parameter but to the best linear MMSE estimator of $s[n]$ from $y_{delayed}[n]$.



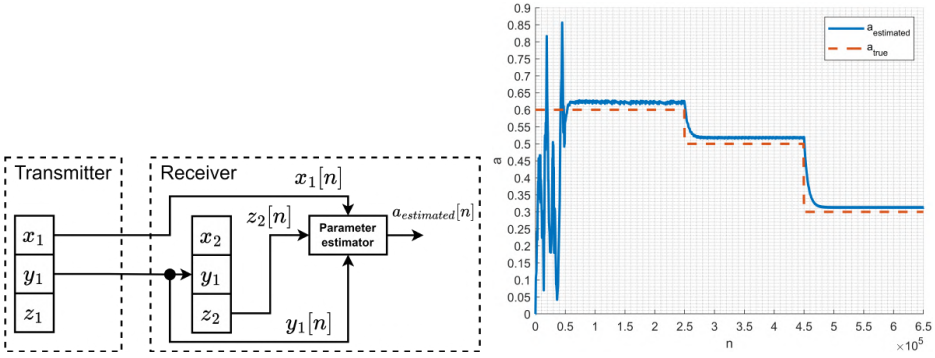
(a) (b)

Fig. 5.6. Experimental setup for verifying the a parameter estimator (a) and the output of the a parameter estimator (b).

The second test evaluates the estimator's performance when it is tasked with estimating the parameter a of the transmitter with the receiver's oscillator synchronized to the transmitter's oscillator. The block diagram of the verification setup is presented in Fig. 5.7 (a). The transmitter sends the $x_1[n]$ and $y_1[n]$ signals to the receiver, where $y_1[n]$ is used to achieve chaotic synchronization using the Pecora-Carroll approach. The synchronization signal is applied after 5×10^4 clock cycles. The received $x_1[n]$ and $y_1[n]$ as well as the signal $z_2[n]$ from the receiver's chaos oscillator are passed to estimate the transmitter's a parameter. This setup exactly reflects the proposed authentication scheme presented in Fig. 5.3. In the receiver's oscillator, a is fixed at 0.47, while the transmitter's oscillator is switched to 0.6, 0.5, and 0.3, after every 2×10^5 clock cycles following synchronization.

Fig. 5.7 (b) shows the $a_{estimated}[n]$ output signal of the estimator. In the first 5×10^4 clock cycles, the receiver's and transmitter's chaos oscillators are asynchronous, so $a_{estimated}[n]$ does not output a stable level, which is expected. After the synchronization signal is applied, the $a_{estimated}[n]$ reaches a stable level equal to 0.61, then after 2×10^5 clock cycles, $a_{estimated}[n]$ level switches to 0.51, and after another 2×10^5 clock cycles, $a_{estimated}[n]$ level switches to 0.31. When synchronized, the estimator tracks the a parameter of the transmitter's chaos oscillator, which is also displayed on the graph. The reached $a_{estimated}[n]$ levels exhibit minor ripples and approximately 4×10^4 cycles of transition time, like in Fig. 5.6 (b).

In the current configuration the 4×10^4 cycles of convergence time at 50 MHz FPGA system clock yields an estimated time of 800 μs. Chaos-based data systems can operate at 60 kb/s, comparable to low-bitrate long-range IoT that operate in 0.3–300 kb/s range, which the developed chaos-based data transfer paired with PLA targets. The estimator, in its current form, has very little overhead given the low bitrate and can be further adjusted and used for the PLA.



(a) (b)

Fig. 5.7. Experimental setup for verifying the *a* parameter estimator in synchronous mode (a) and the output of the *a* parameter estimator (b).

Overall, although the estimator does not produce the exact values of the transmitter's *a*, it outputs levels that are clearly distinguishable and close to the original values. Furthermore, the difference between actual and estimated *a* appears to be constant. Another observation made in the Fig. 5.7 (b) results is that the ripples on the estimated level in the case of transmitter's *a* equal 0.6 are more than in the case of the other *a* values. This raises the question of what distinguishable *a* levels can be reliably used for authentication.

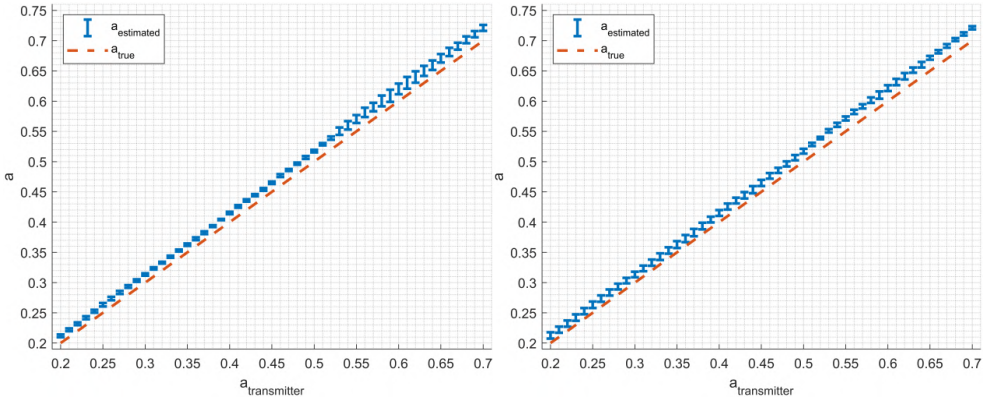
The next study examines how the estimated average value and minimum-maximum range of $a_{estimated}$ vary based on the transmitter's *a*. For the study, the transmitter's *a* is varied from 0.2 to 0.7 in increments of 0.01. The two cases of receiver's *a* are taken - in the first case receiver's *a* equals 0.47, like in Fig. 5.7; in the second case receiver's *a* equals 0.6.

The results of the study are presented in Fig. 5.8. The results in the case of receiver's *a* equals 0.47 in Fig. 5.8 demonstrate a very small minimum-maximum ranges when transmitter's *a* is from 0.2 to 0.52, making them well distinguishable. The $a_{estimated}$ minimum-maximum range then increases, reaching its peak at 0.61 and then decreases. Within this *a* value range, the minimum-maximum ranges of the neighboring $a_{estimated}$ values overlap, meaning that they are indistinguishable when used for authentication. Every third *a* value must be used instead.

Alternatively, if the receiver's *a* is set to 0.6, the $a_{estimated}$ minimum-maximum ranges are different across the transmitter's *a* value range. For the transmitter's *a* range 0.2 to 0.52, the minimum-maximum ranges increase, resulting in a slight overlap, so every second $a_{estimated}$ is distinguishable. The range above 0.52 has decreased minimum-maximum ranges, also making every second $a_{estimated}$ distinguishable instead of every third, as in the case when the receiver's *a* was set to 0.47.

The results present several key points. First, it is possible to select a set of distinguishable parameter *a* values to use in the authentication. Second, the minimum-maximum range of the estimator's output depends on the receiver's parameter *a*, making this a crucial and controllable parameter when selecting the *a* values used for authentication. Finally, the difference between the estimated and true

values of a is the same across the entire parameter range, as seen by overlaying the true values with a dashed line. The maximum relative error of the average a in Fig. 5.8 does not exceed 6 % if the linear shift is not compensated. By compensating the estimated average 0.017 shift, the relative error then does not exceed 1.01 %. This highlights the necessity for calibration.

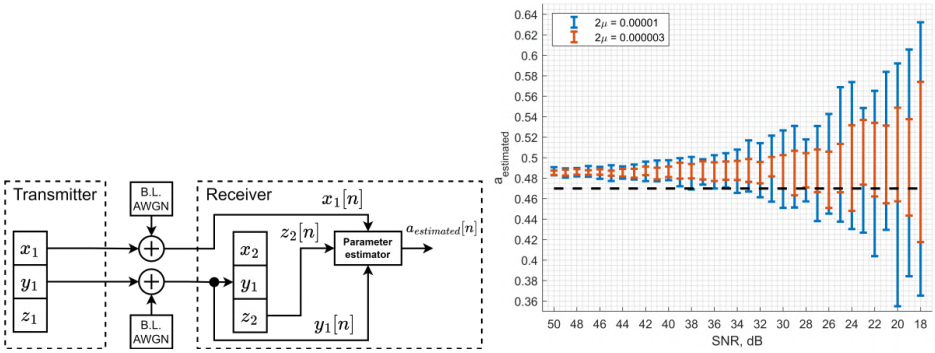


(a) (b)

Fig. 5.8. The output of the a parameter estimator average and minimum-maximum range for different transmitter a values in the case of receiver's $a = 0.47$ (a) and in the case of receiver's $a = 0.6$ (b).

The next study aims to investigate how additive noise affects the performance of the estimator. The setup for this evaluation is demonstrated in Fig. 5.9 (a). The notable modification is the inclusion of B.L.AWGN, which is added to the transmitter oscillator's signals $x_1[n]$ and $y_1[n]$. The added noise signals are in-band with the related chaotic signals, representing the worst-case scenario, as the out-of-band noise can be filtered, thus improving the SNR. Since the parameter estimation system is designed to operate with synchronous receivers' and transmitters' chaotic oscillators, this noise will affect both the oscillators' synchronization and the estimation quality. The transmitter's and receiver's a are both set to 0.47, and the noise signal's SNR is varied from 18 to 50 dB.

The results of this study are presented in Fig. 5.9 (b). The first set of results was obtained for the convergence speed 2μ set to 1×10^{-5}, which is the same as in previous studies. The results present the $a_{estimated}$ average value and minimum-maximum range across the SNR range, demonstrating that the minimum-maximum range noticeably increases as the SNR decreases. This behavior is expected and indicates that the noise will affect the ability to distinguish the levels of the parameter. Improving the noise performance of the estimator is possible by decreasing the convergence speed, i.e, increasing the window of the adaptive filter. The second set of results was acquired for the convergence speed 2μ set to 3×10^{-6} and demonstrates a noticeably lower $a_{estimated}$ minimum-maximum range for the same SNR levels.



(a) (b)

Fig. 5.9. Experimental setup for verifying the *a* parameter estimator in B.L.AWGN channel (a) and the output of the *a* parameter estimator (b).

5.3 Discussion

This section introduced a novel scheme based on chaotic oscillator parameter estimation for PLA. The possibility of setting and estimating the parameter of the chaotic oscillator provides a powerful authentication mechanism that can be integrated into chaos-based data transfer systems.

The performance evaluation of the designed estimator identified several key aspects for successful deployment. The primary factor is the tradeoff between estimation time and precision, governed by the convergence speed. Faster convergence reduces estimation time but increases output fluctuations, making it more difficult to distinguish discrete parameter ranges used for authentication. Noise immunity analysis further showed that these fluctuations increase with noise level, requiring convergence speed adaptation according to the channel state.

In the proposed chaos-based data transfer system, the channel state can be estimated using the already implemented Pearson correlation coefficient, which evaluates synchronization quality by measuring the similarity between local and received chaotic signals. This enables dynamic adjustment of the parameter estimation process under noisy conditions.

Another important factor is the receiver oscillator's native parameter, which determines which transmitter parameter ranges can be reliably distinguished. This allows the system to be highly configurable by selecting the receiver's native parameter and adjusting convergence speed to achieve the desired authentication precision. Combined, these features create a highly parameterized system capable of mitigating unauthorized data transmission from similar architectures.

CONCLUSIONS

This Doctoral Thesis has presented a comprehensive study on the analysis, design, and experimental validation of analog-discrete chaotic systems and their synchronization.

The results demonstrate that chaotic synchronization between analog circuits and FPGA-based digital models is both feasible and reliable. High levels of synchronization accuracy were achieved, confirming that discrete-time implementations can effectively reproduce the dynamics of continuous-time chaotic systems and vice versa when appropriate scaling and normalization techniques are applied.

The research further shows that FPGA-based implementations using fixed-point arithmetic are capable of preserving essential chaotic properties, making them suitable for real-time and hardware-efficient applications. Among the investigated configurations, analog–discrete synchronization exhibited the highest robustness, while discrete-analog configuration also provided satisfactory performance under practical conditions. Although noise, quantization effects, and component tolerances introduce deviations from ideal behavior, the systems maintained stable synchronization within acceptable limits, demonstrating their robustness in real-world environments.

In addition, the feasibility of chaos-based authentication was validated through parameter estimation techniques. The results indicate that chaotic systems can be effectively utilized for physical-layer security applications, offering a promising alternative to conventional methods.

In summary, this Thesis establishes a unified framework for the development and implementation of analog-discrete chaotic systems, contributing to the advancement of secure communication technologies and hardware-based security solutions.

BIBLIOGRAPHY

[1] *Ericsson Mobility Report November 2024*. en. Tech. rep. 2024. URL: https://www.ericsson.com/4adb7e/assets/local/reports-papers/mobility-report/documents/2024/ericsson-mobility-report-november-2024.pdf (visited on 05/01/2025).

[2] S. Ullah et al. "Types of Lightweight Cryptographies in Current Developments for Resource Constrained Machine Type Communication Devices: Challenges and Opportunities." In: *IEEE Access* 10 (2022), pp. 35589–35604. ISSN: 2169-3536. DOI: 10.1109/ACCESS.2022.3160000.

[3] C. Silva, N. Tenório, and J. Bernardino. "Lightweight Encryption Algorithms for IoT." en. In: *Computers* 14.12 (Dec. 2025), p. 505. ISSN: 2073-431X. DOI: 10.3390/computers14120505.

[4] K. U. Sarker. "A systematic review on lightweight security algorithms for a sustainable IoT infrastructure." en. In: *Discover Internet of Things* 5.1 (Apr. 2025), p. 47. ISSN: 2730-7239. DOI: 10.1007/s43926-025-00150-4.

[5] A. S. Demirkol et al. "Real time hybrid medical image encryption algorithm combining memristor-based chaos with DNA coding." In: *Chaos, Solitons & Fractals* 183 (June 2024), p. 114923. ISSN: 0960-0779. DOI: 10.1016/j.chaos.2024.114923.

[6] T. Bonny and W. Al Nassan. "NeuroChaosCrypt: Revolutionizing Chaotic-Based Cryptosystem With Artificial Neural Networks—A Comparison With Traditional Cryptosystems." en. In: *IEEE Access* 12 (2024), pp. 62030–62046. ISSN: 2169-3536. DOI: 10.1109/ACCESS.2024.3395527.

[7] C. Zhang et al. "Double Image Encryption Algorithm Based on Parallel Compressed Sensing and Chaotic System." en. In: *IEEE Access* 12 (2024), pp. 54745–54757. ISSN: 2169-3536. DOI: 10.1109/ACCESS.2024.3389975.

[8] D. Migwi and R. S.Romaniuk. "Lightweight and Scalable Security for Wireless IoT Systems: Challenges and Research Directions." en. In: *International Journal of Electronics and Telecommunication* 71.3 (July 2025), pp. 1–8. ISSN: 2300-1933. DOI: 10.24425/ijet.2025.155451.

[9] L. He et al. "A Modulated Position-Indexed Differential Chaos Shift Keying System With Enhanced BER and Throughput." en. In: *IEEE Communications Letters* 30 (2026), pp. 612–616. ISSN: 1089-7798, 1558-2558, 2373-7891. DOI: 10.1109/LCOMM.2025.3646108.

[10] R. U. Almada-Prieto, J. C. Núñez-Pérez, and A. R. Díaz-Rizo. "Chaos-Based Hardware Trojan Covert Channel for Synchronized Chaotic Cryptosystems." en. In: *2026 IEEE 17th Latin America Symposium on Circuits and System (LASCAS)*. Arequipa, Peru: IEEE, Feb. 2026, pp. 1–5. ISBN: 979-8-3315-7097-2. DOI: 10.1109/LASCAS67804.2026.11457091.

[11] M. Siino, S. Mangione, and I. Tinnirello. “Compact Attention-Augmented Neural Decoder for Chaos-Based Wireless Communications.” en. In: *IEEE Communications Letters* 30 (2026), pp. 1275–1279. ISSN: 1089-7798, 1558-2558, 2373-7891. DOI: 10.1109/LCOMM.2026.3670578.

[12] S. Wei et al. “Adaptive secure communications with key updating using 4D hyper-chaos in digital mobile fronthaul.” en. In: *IEEE Photonics Technology Letters* (2026), pp. 1–1. ISSN: 1041-1135, 1941-0174. DOI: 10.1109/LPT.2026.3681645.

[13] J. Yang et al. “High-Sensitivity Chaotic System for Image Security: Applications to Multi-ROI Encryption.” en. In: *IEEE Internet of Things Journal* (2026), pp. 1–1. ISSN: 2327-4662, 2372-2541. DOI: 10.1109/JIOT.2026.3682306.

[14] D. Oikawa et al. “Random-Number Generation Using Chaos in RF-Irradiated Stacked Intrinsic Josephson Junction.” en. In: *IEEE Transactions on Applied Superconductivity* 36.5 (Aug. 2026), pp. 1–5. ISSN: 1051-8223, 1558-2515, 2378-7074. DOI: 10.1109/TASC.2025.3650523.

[15] J. Petrzela. “Chaos in Analog Electronic Circuits: Comprehensive Review, Solved Problems, Open Topics and Small Example.” en. In: *Mathematics* 10.21 (Jan. 2022), p. 4108. ISSN: 2227-7390. DOI: 10.3390/math10214108.

[16] X.-k. Hu et al. “A discrete Lorenz-like map and its pseudo-Hamiltonian energy.” In: *Physica Scripta* 100 (July 2025), p. 075232. ISSN: 0031-8949. DOI: 10.1088/1402-4896/ade0f9.

[17] L. Wang et al. “Estimation-Correction Modeling and Chaos Control of Fractional-Order Memristor Load Buck-Boost Converter.” en. In: *Complex System Modeling and Simulation* 4.1 (Mar. 2024), pp. 67–81. ISSN: 2096-9929, 2097-3705. DOI: 10.23919/CSMS.2024.0002.

[18] L. Benadero et al. “Period Doubling Route to Chaos in Open Loop Boost Converters under Constant Power Loading and Discontinuous Conduction Mode Conditions.” en. In: *2020 IEEE International Symposium on Circuits and Systems (ISCAS)*. Seville, Spain: IEEE, Oct. 2020, pp. 1–5. ISBN: 978-1-72813-320-1. DOI: 10.1109/ISCAS45731.2020.9180647.

[19] R. Cristiano et al. “Chaos Through Sliding Bifurcations in a DC–DC Boost Power Converter.” en. In: *International Journal of Bifurcation and Chaos* (Nov. 2024). DOI: 10.1142/S0218127424300337.

[20] J. R. C. Piqueira. “Hopf bifurcation and chaos in a third-order phase-locked loop.” In: *Communications in Nonlinear Science and Numerical Simulation* 42 (Jan. 2017), pp. 178–186. ISSN: 1007-5704. DOI: 10.1016/j.cnsns.2016.06.001.

[21] Z. Duan et al. “A Fully Integrated Memristive Chaotic Circuit Based on Memristor Emulator with Voltage-Controlled Oscillator.” en. In: *Micromachines* 16.3 (Mar. 2025), p. 246. ISSN: 2072-666X. DOI: 10.3390/mi16030246.

[22] F. Yu et al. “Dynamics analysis, synchronization and FPGA implementation of multiscroll Hopfield neural networks with non-polynomial memristor.” In: *Chaos, Solitons & Fractals* 179 (Feb. 2024), p. 114440. ISSN: 0960-0779. DOI: 10.1016/j.chaos.2023.114440.

[23] D. N. Butusov et al. “Synchronization of analog and discrete Rössler chaotic systems.” en. In: *2017 IEEE Conference of Russian Young Researchers in Electrical and Electronic Engineering (EIConRus)*. St. Petersburg and Moscow, Russia: IEEE, 2017, pp. 265–270. ISBN: 978-1-5090-4865-6. DOI: 10.1109/EIConRus.2017.7910544.

[24] T. Karimov et al. “Accurate Synchronization of Digital and Analog Chaotic Systems by Parameters Re-Identification.” en. In: *Electronics* 7.7 (July 2018), p. 123. ISSN: 2079-9292. DOI: 10.3390/electronics7070123.

[25] Y.-H. An et al. “Design and implementation of master-slave synchronization for chaotic Lur’e systems using Chua’s circuit.” en. In: *2024 43rd Chinese Control Conference (CCC)*. Kunming, China: IEEE, July 2024, pp. 682–686. ISBN: 978-988-758-158-1. DOI: 10.23919/CCC63176.2024.10661450.

[26] H. Cheng et al. “A deep reinforcement learning method to control chaos synchronization between two identical chaotic systems.” In: *Chaos, Solitons & Fractals* 174 (Sept. 2023), p. 113809. ISSN: 0960-0779. DOI: 10.1016/j.chaos.2023.113809.

[27] H. G. Chou et al. “A fuzzy-model-based chaotic synchronization and its implementation on a secure communication system.” In: *IEEE Transactions on Information Forensics and Security* 8.12 (2013), pp. 2177–2185. ISSN: 15566013. DOI: 10.1109/TIFS.2013.2286268.

[28] L. Du et al. *Chaos Synchronization of a Class of Chaotic Systems via Linear State Error Feedback Control*. Jan. 2016. DOI: 10.2991/ameii-16.2016.295.

[29] T. Addabbo et al. “A New Class of Chaotic Sources in Programmable Logic Devices.” en. In: *2020 IEEE International Workshop on Metrology for Industry 4.0 & IoT*. Roma, Italy: IEEE, June 2020, pp. 6–10. ISBN: 978-1-72814-892-2. DOI: 10.1109/MetroInd4.0IoT48571.2020.9138256.

[30] O. Karataş, K. Demir, and S. Ergün. “Chaotic Sampling of Double Scroll Chaos for Digital Random Number Generation.” en. In: *2022 IEEE Asia Pacific Conference on Circuits and Systems (APCCAS)*. Shenzhen, China: IEEE, Nov. 2022, pp. 424–427. ISBN: 978-1-66545-073-7. DOI: 10.1109/APCCAS55924.2022.10090396.

[31] M. S. Azzaz, R. Kaibou, and A. Smahi. “FPGA Implementation using Novel Co-Design Approach of Real-Time Speech Chaos based Crypto-Watermarking Prototype.” en. In: *2023 International Conference on Advances in Electronics, Control and Communication Systems (ICAECCS)*. BLIDA, Algeria: IEEE, Mar. 2023, pp. 1–6. ISBN: 978-1-66546-310-2. DOI: 10.1109/ICAECCS56710.2023.10104859.

[32] R. Hobincu and O. Datcu. “FPGA Implementation of a Chaos Based PRNG Targetting Secret Communication.” en. In: *2018 International Symposium on Electronics and Telecommunications (ISETC)*. Timisoara, Romania: IEEE, Nov. 2018, pp. 1–4. ISBN: 978-1-5386-5925-0. DOI: 10.1109/ISETC.2018.8583863.

[33] R. R. Babu and R. Karthikeyan. “Adaptive synchronization of novel chaotic system and its FPGA implementation.” In: *2015 International Conference on Smart Technologies and Management for Computing, Communication, Controls, Energy and Materials, ICSTM 2015 - Proceedings* May (2015), pp. 449–454. DOI: 10.1109/ICSTM.2015.7225459.

[34] O. Guillén-Fernández et al. “On the synchronization techniques of chaotic oscillators and their FPGA-based implementation for secure image transmission.” en. In: *PLOS ONE* 14.2 (2019), e0209618. ISSN: 1932-6203. DOI: 10.1371/journal.pone.0209618.

[35] R. Babajans et al. “Study on Analog and Discrete Chaos Oscillators Synchronization.” en. In: *16th Chaotic Modeling and Simulation International Conference*. Ed. by C. H. Skiadas and Y. Dimotikalis. Cham: Springer Nature Switzerland, 2024, pp. 33–46. ISBN: 978-3-031-60907-7. DOI: 10.1007/978-3-031-60907-7_4.

[36] R. Babajans et al. “Experimental Study on Analog and Discrete Chaos Oscillators Synchronization.” In: *2023 Workshop on Microwave Theory and Technology in Wireless Communications (MTTW)*. Oct. 2023, pp. 24–28. DOI: 10.1109/MTTW59774.2023.10319995.

[37] R. Babajans et al. “Synchronization of Analog-Discrete Chaotic Systems for Wireless Sensor Network Design.” en. In: *Applied Sciences* 14.2 (Jan. 2024), p. 915. ISSN: 2076-3417. DOI: 10.3390/app14020915.

[38] R. Babajans, D. Cirjulina, and D. Kolosovs. “Discrete Sequential Implementation of Chaos Oscillators for FPGA Integration.” In: *2024 IEEE Workshop on Microwave Theory and Technology in Wireless Communications (MTTW)*. Riga, Latvia: IEEE, Oct. 2024, pp. 33–36. ISBN: 979-8-3315-3317-5. DOI: 10.1109/MTTW64344.2024.10742168.

[39] R. Babajans, D. Cirjulina, and D. Kolosovs. “Field-Programmable Gate Array-Based Chaos Oscillator Implementation for Analog–Discrete and Discrete–Analog Chaotic Synchronization Applications.” en. In: *Entropy* 27.4 (Apr. 2025), p. 334. ISSN: 1099-4300. DOI: 10.3390/e27040334.

[40] R. Babajans et al. “Noise Performance Study of FPGA-based and Analog Chaos Oscillator Synchronization.” en. In: *2025 IEEE 12th Workshop on Advances in Information, Electronic and Electrical Engineering (AIEEE)*. Vilnius, Lithuania: IEEE, 2025, pp. 1–5. DOI: 10.1109/AIEEE66149.2025.11050878.

[41] D. Cirjulina et al. “Nonlinear Dynamics and Hybrid Synchronization of DC Biased Colpitts Chaotic Oscillators.” en. In: *Electronics* 14.20 (Jan. 2025), p. 4005. ISSN: 2079-9292. DOI: 10.3390/electronics14204005.

[42] R. Babajans et al. “Chaos-Based Dynamical Parameter Estimation for Physical Layer Authentication in Wireless IoT Networks.” en. In: *Electronics* 15.4 (Jan. 2026), p. 748. ISSN: 2079-9292. DOI: 10.3390/electronics15040748.

[43] O. Rössler. “An equation for continuous chaos.” In: *Physics Letters A* 57.5 (July 1976), pp. 397–398. ISSN: 0375-9601. DOI: 10.1016/0375-9601(76)90101-8.

[44] K. M. Cuomo and A. V. Oppenheim. “Circuit implementation of synchronized chaos with applications to communications.” en. In: *Physical Review Letters* 71.1 (July 1993), pp. 65–68. ISSN: 0031-9007. DOI: 10.1103/PhysRevLett.71.65.

[45] M. Kennedy. “Chaos in the Colpitts oscillator.” en. In: *IEEE Transactions on Circuits and Systems I: Fundamental Theory and Applications* 41.11 (Nov. 1994), pp. 771–774. ISSN: 10577122. DOI: 10.1109/81.331536.

[46] J. C. Sprott. “A new class of chaotic circuit.” In: *Physics Letters, Section A: General, Atomic and Solid State Physics* 266.1 (2000), pp. 19–23. ISSN: 03759601. DOI: 10.1016/S0375-9601(00)00026-8.

[47] A. Tamaševičius et al. “A simple chaotic oscillator for educational purposes.” In: *European Journal of Physics* 26.1 (Nov. 2004), p. 61. DOI: 10.1088/0143-0807/26/1/007.

[48] A. Namajunas and A. Tamasevicius. “Simple RC chaotic oscillator.” In: *Electronics Letters* 32 (June 1996), pp. 945–946. DOI: 10.1049/el:19960682.

[49] *FPGA implementation of a chaotic oscillator using RK4 method*. en-US. URL: https://ieeexplore.ieee.org/document/5782646 (visited on 09/30/2025).

[50] L. M. Pecora and T. L. Carroll. “Synchronization in chaotic systems.” en. In: *Physical Review Letters* 64.8 (Feb. 1990), pp. 821–824. ISSN: 0031-9007. DOI: 10.1103/PhysRevLett.64.821.

[51] L. M. Pecora et al. “Fundamentals of synchronization in chaotic systems, concepts, and applications.” eng. In: *Chaos (Woodbury, N.Y.)* 7.4 (Dec. 1997), pp. 520–543. ISSN: 1089-7682. DOI: 10.1063/1.166278.

[52] *DG419 Datasheet and Product Info | Analog Devices*. URL: https://www.analog.com/en/products/dg419.html#product-overview (visited on 12/30/2023).

[53] D. Cirjulina, R. Babajans, and D. Kolosovs. “Design Particularities of Quadrature Chaos Shift Keying Communication System with Enhanced Noise Immunity for IoT Applications.” en. In: *Entropy* 27.3 (Mar. 2025), p. 296. ISSN: 1099-4300. DOI: 10.3390/e27030296.

[54] R. Babajans et al. “Impact of the Chaotic Synchronization’s Stability on the Performance of QCPSK Communication System.” In: *Electronics* 10.6 (2021), p. 640. ISSN: 2079-9292. DOI: 10.3390/electronics10060640.

[55] A. Karimov et al. “Adaptive Generalized Synchronization between Circuit and Computer Implementations of the Rössler System.” en. In: *Applied Sciences* 11.1 (Dec. 2020), p. 81. ISSN: 2076-3417. DOI: 10.3390/app11010081.

Ruslans Babajans was born in 1998 in Riga, Latvia. He received an Academic Bachelor's degree in Electrical Engineering (2020) and a Professional Master's degree in Electronics and the qualification of Lead Electronics Engineer (2022) from Riga Technical University (RTU). Since 1 March 2019, he has been a Research Assistant at RTU, and since September 2022, a Researcher at the Institute of Photonics, Electronics and Telecommunications, where he participates in research projects and teaches students.

During his doctoral studies, he participated in several international mobilities and professional visits at KTH Royal Institute of Technology (Sweden), Politecnico di Torino (Italy), and Aristotle University of Thessaloniki (Greece). His scientific and academic achievements during doctoral studies have been recognised with the *IEEE* Best Paper Award (2023).

His research interests focus on chaos-based data transmission systems for *IoT* applications and field-programmable gate arrays.