ISSN: 2687 - 4539

CHAOS THEORY AND APPLICATIONS

IN APPLIED SCIENCES AND ENGINEERING

VOLUME 6, ISSUE 4, NOVEMBER 2024

AN INTERDISCIPLINARY JOURNAL OF NONLINEAR SCIENCE

ISSN: 2687 - 4539

CHAOS

THEORY AND APPLICATIONS

IN APPLIED SCIENCES AND ENGINEERING

Chaos Theory and Applications (CHTA)
Volume:6 – Issue No:4(November 2024)
https://dergipark.org.tr/en/pub/chaos/issue/88057

Editorial Advisory Board Members

Language Editors

Technical Coordinator

ISSN: 2687 - 4539

CHAOS THEORY AND APPLICATIONS

IN APPLIED SCIENCES AND ENGINEERING

Chaos Theory and Applications (CHTA)
Volume: 6 – Issue No: 4 (November 2024)
https://dergipark.org.tr/en/pub/chaos/issue/88057

Contents

CHAOS
Theory and Applications
in Applied Sciences and Engineering

e-ISSN: 2687-4539
RESEARCH ARTICLE
Vol.6 / No.4 / 2024 / pp.228-236
https://doi.org/10.51537/chaos.1462132

Nonlinear Dynamics Induced by Coil Heat in the PMDC Motor and Control

Arnaud Ngonting Topy [*] [α,1], **Justin Roger Mboupda Pone** [*,2], **Alex Stephane Kemnang Tsafack** [α,3] **and Andre Cheukem** [*,4]
[*]Research Unit of Automation and Applied Computers (RU-AIA), Electrical Engineering Department of UIT-FV, University of Dschang, P.O. Box: 134, Bandjoun, Cameroon, [α]Research Unit of Condensed Matter of Electronics and signal Processing (RU-MACETS). Department of Physics, Faculty of Sciences, University of Dschang, P.O. Box 67, Dschang, Cameroon.

ABSTRACT In this paper, the interesting dynamics of chaos induced by the effect of the variation of internal average heat during operation in the DC motor control by the full bridge drive are analyzed. By using simple powerful tools of analyzing nonlinear dynamical systems like phase portraits, time traces and frequency spectrum in the MATLAB-SIMULINK environment, we showed that under certain conditions, the PMDC motor develops different behaviors as periodic limit cycles, and chaotic attractors, when the motor drive different form of external load torque and the windings resistance variation. This paper presents the first studies on the variation of the average heat of the motor and the amplitude of the triangular load torque to produce the strange phenomena like chaos as far as our knowledge go. A chaos control of the unstable regime is proposed to stabilize the PMDC motor in a desire regime. This contribution is very important in industry because some unexplained dynamical behaviors of the DC motor driven by a full bridge now can be avoided.

KEYWORDS
PMDC motor
Triangular load torque
Windings coil heat
Phase portraits
Periodic oscillations
Torus
Chaos
Chaos control

INTRODUCTION

The electric motor (EM) is the most used actuator in industry (Yu *et al.* 2011; Dalcalı 2018) . The electromotive force (EMF) powers most of the electro-mechanical actuators in the industry. In these systems the torque is created by using the Laplace laws. Among various type of these EM actuators, there is the permanent magnet DC electrical motors that are sufficiently used in industry today due to its linear torque characteristics, adjustable speed simplicity of control less noise operation and longer durability (Klein and Kenyon 1984; Arat 2018; Gieras 2009). Generally, The actuator in this contribution is a type of DC motor that uses a permanent magnet (PM) to create the electromotive force (EMF) useful for its operation(Öztürk 2020; Pillay and Krishnan 1989). This electromechanical actuator is useful in applications like automotive industrial robots, production automated system, agriculture, aerospace just to name some sample contributions (Liao *et al.* 1995).

In order to fulfill their industrial requirements, a bridge controlled is widely used in this purpose. In PMDC electrical motor drive, power transistors provide nonlinearity in the overall system due to their on/off state (Parsa and Toliyat 2005). Some nonlinear dynamical contributions have been intensively studied and have proven to exhibit stable, regular, oscillation behaviors and chaos (Öztürk 2020). Some studies show that nonlinear system can entered chaos regime via complex behavior while one or two parameters are varied (Poliashenko and Aidun 1995; Ayan and Kurt 2018). Some simple condition to entering chaos in systems including few fundamental electrical drive systems are described by (Chau and Wang 2011) .These chaos regimes are usually unwanted because they could be harmful to the industrial application (Tahir *et al.* 2017). Among these parameters are the load the DC source, the control speed, the duty ratio, or the parameter of the feedback control system .

Okafor *et al.* (2010b) reported the multistability of P-1 and P-3 attracting sets and fractal basin boundaries in dc drives controlling a PMDC motor by using a 4 quadrants DC/DC converter circuit. Using the well-known Filippov condition, the stability nature of P-1 and P-3 attractors was computed to explain the presence of the competing limit cycles. His contribution aimed to prevent the occurrence of this behavior by adding a controller that extend the parameter range for safe limit cycle operation. Öztürk (2020) shows

Manuscript received: 18 April 2024,
Revised: 2 August 2024,
Accepted: 13 September 2024.

[1]ngontingtopyarnaur@gamil.com
[2]mboupdapone00@gmail.com (**Corresponding author**)
[3]alexstphanekemnang@gmail.com
[4]acheukem05@gmail.com

by using MATLAB/Simulink model, that under chaotic load the asynchronous machine develops chaos. As far as concerned the chaotic effect in the motor, we note that other authors described the behaviors of dynamics and control of chaos in the IFOC of 3-Φ IM. They show simple criterion for the motor to develops periodic and chaotic behavior (Tsafack *et al.* 2020). Okafor *et al.* (2010a, 2015) analyzed and controlled period cascade to chaos in DC/DC converter employing a full-bridge using Filippov's algorithm. The authors showed that the monodromy matrix and its eigenvalues could be computed by the periodic orbit stability analyses. The previous result can be modified by the appropriate computation of the saltation matrix of the system. Hence, they designed a controller to stabilize the behavior to period-l operation of the PMDC motor.

Abdullah *et al.* (2016) and Tahir *et al.* (2017) suppressed the chaos regime by using the sliding mode control method. Recently, Moustafa *et al.* (2021) study the Floquet theory to control the system, they experienced oscillations when varying the load torque without qualitative variation of the dynamics. The control the speed of the driven load and obtained strong stability of the system. Tsafack *et al.* (2020) suppressed chaos by applying a self-feedback controller in 3phases IFOC IM. In other hand, it is well known that different losses (friction, Joule effect, copper losses...) contributed to motor heat increasing and reduce the motor efficiency, (speed, torque ...) resulting in a thermal aging process and eventually destroy the motor. Therefore, reducing thermal effect and increasing motor cooling systems result to the longevity and reliability of the EM (Bonnett 2001). Some good standing papers have focused on the thermal effects on the torque speed performance of a Brushless Direct Current Motor (BLDCM) (Fussell 1993). Another contribution analyzed the transient thermal network model that could be used to predict some sensitivity of the model towards design variables (Junak *et al.* 2008; Minghui and Weiguo 2010).

In summary of the previous contributions and as far as out knowledge goes, no contribution has been interested on windings currents heat consequences behavior on the nonlinear effect of the PMDC motor that could explained some industrial chaotic behavior of the driven process. This paper tries to bridge this gap. We observed that till now, no-contribution has focused on the influence of the thermal behavior of the rotor resistance to the dynamic behavior of the PMDC motor: we set this as the main objective of this paper. The chaotic behavior due to effect of the internal heat in the motor when the motor drive different type of load torque is proposed and some interesting dynamics like limit cycles, torus, chaos due to the heat or resistance or the variation of inductance are shown. These results are encouraging because some unexplained dynamical behaviors of the DC motor driven by a full bridge now can be explained and avoided (Pisarchik and Feudel 2014).

The main highlights of this contributions are as follows:

- The variation of the amplitude value of a triangle load is investigated and showed some chaotic regime in the PMDC Motor.
- We discovered that the thermal heat on the winding's resistance induced interesting nonlinear behaviors like limit cycles, period doubling route to chaos, torus, and chaotic attractors.
- We showed the existence of the nonlinear dynamics induced by the heat on the magnet field not yet explained.
- A chaos control of the unstable regime due to these novel dynamics is proposed to stabilize the PMDC motor in a desire regime.

The outline of this article is as follows: the next section 2 introduces the description of mathematical model of PMDC motor. In section 3, the dynamical behaviors of the motor induced by the load torque are presented. The section 4 highlights the effect of the winding resistance variation due to heat with different variation. The section 5 exhibits the control of the chaos using the load in the PMDC motor. Finally, the section 6 concludes this paper.

MATHEMATICAL MODEL OF THE PMDC DRIVE

The mathematical model of the PMDC motor driven by a full bridge converter is well documented (Okafor *et al.* 2010b; Okafor 2013; Abdullah *et al.* 2016) and the schematic diagram is recalled in Figure (1).

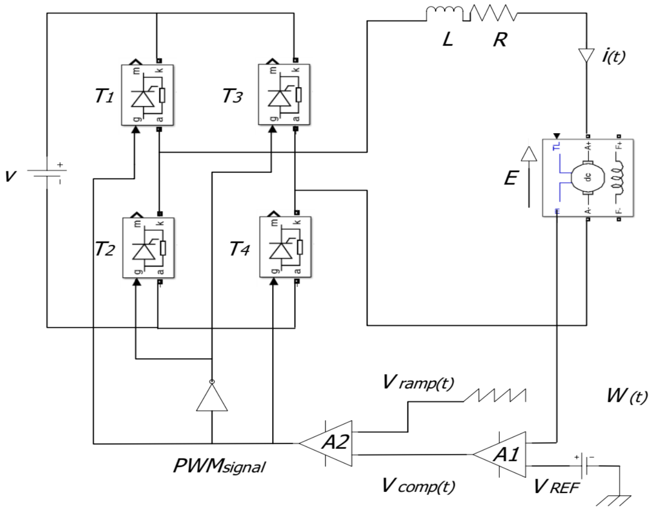


Figure 1 The PMDC driven by a full bridge converter.

The system is built by connecting the power converter bridge, a control electronic and permanent magnet DC motor. After a long-term running, the system in Figure (1) toggled between 2 subsystems in accordance with the state of the output of the second operational amplifier A2 used in switched mode. In the Figure (1), V represent the supply voltage $T1$, $T2$, $T3$, and $T4$ are the semiconductor controlled rectifier, (SCR) L is the inductance, R is the resistance, i(t) is the current, E is the voltage across the motor, Vramp(t) is the sawtooth signal, $\omega(t)$ is the speed of the motor, Vref represents the value of the reference speed of the motor, $Vcomp(t)$ is the difference between the reference speed and the actual speed developed by the motor; PWM signal Pulse Width Modulated voltage.

Operational amplifier Switch A2 is high

The differential equations governing the evolution of the current and the speed of the motor are given in equations (1).

$$\begin{cases} \frac{dw(t)}{dt} = \dfrac{T_e - T_L - bw(t)}{J} \\ \frac{di(t)}{dt} = \dfrac{-Ri(t) - E_b - Vin}{L} \end{cases} \qquad (1)$$

where we have

$$T_e = k_t i(t), \; E_b = k_e w(t) \qquad (2)$$

After replacing the equation (2) in equations (1), we have the equations (3).

$$\begin{cases} \dfrac{dw(t)}{dt} = \dfrac{k_t i(t) - T_L - bw(t)}{J} \\ \dfrac{di(t)}{dt} = \dfrac{-Ri(t) - k_e w(t) + Vin}{L} \end{cases} \tag{3}$$

Note that $i(t)$ is the current absorbed by the motor, $w(t)$ is the motor output speed, R is the sum of resistance of the armature and the field coil inductances. L L is the sum of the armature and field coil inductances, k_e and $k_t$are the back electromotive force (EMF) constant and the torque constant, respectively; b is the friction coefficient, J is the moment of inertia; T_L is the load torque. In this sub-part of the work, the system settings are chosen as follows:$R = 7.2\Omega$, $L = 0.0917H$, $T_L = 0.2N_m$, $T = 10ms$, $k_e = k_t = 0,1236N.m/A^2$, $J = 7,046e^{-4}kg.m^2$, $b = 4e^{-4}Nm.rads.s$, $V_L = 0V$, $V_u = 8V$ and $w_{ref} = 100rad/s$ the expression of $V_{com}(t) = A1(w(t) - w_{ref})$ (Abdullah *et al.* 2016).

Operational amplifier Switch A2 is low

In this subsection, the switch is off the generator signal is equal to zero $(V_{in} = 0V)$. Equations. (3) becomes equations (4) as shown as follows.

$$\begin{cases} \dfrac{dw(t)}{dt} = \dfrac{k_t i(t) - T_L - bw(t)}{J} \\ \dfrac{di(t)}{dt} = \dfrac{-Ri(t) - k_e w(t)}{L} \end{cases} \tag{4}$$

The parameters values of equations (4) are the same as in section 1. This is the model that will be used throughout the paper.

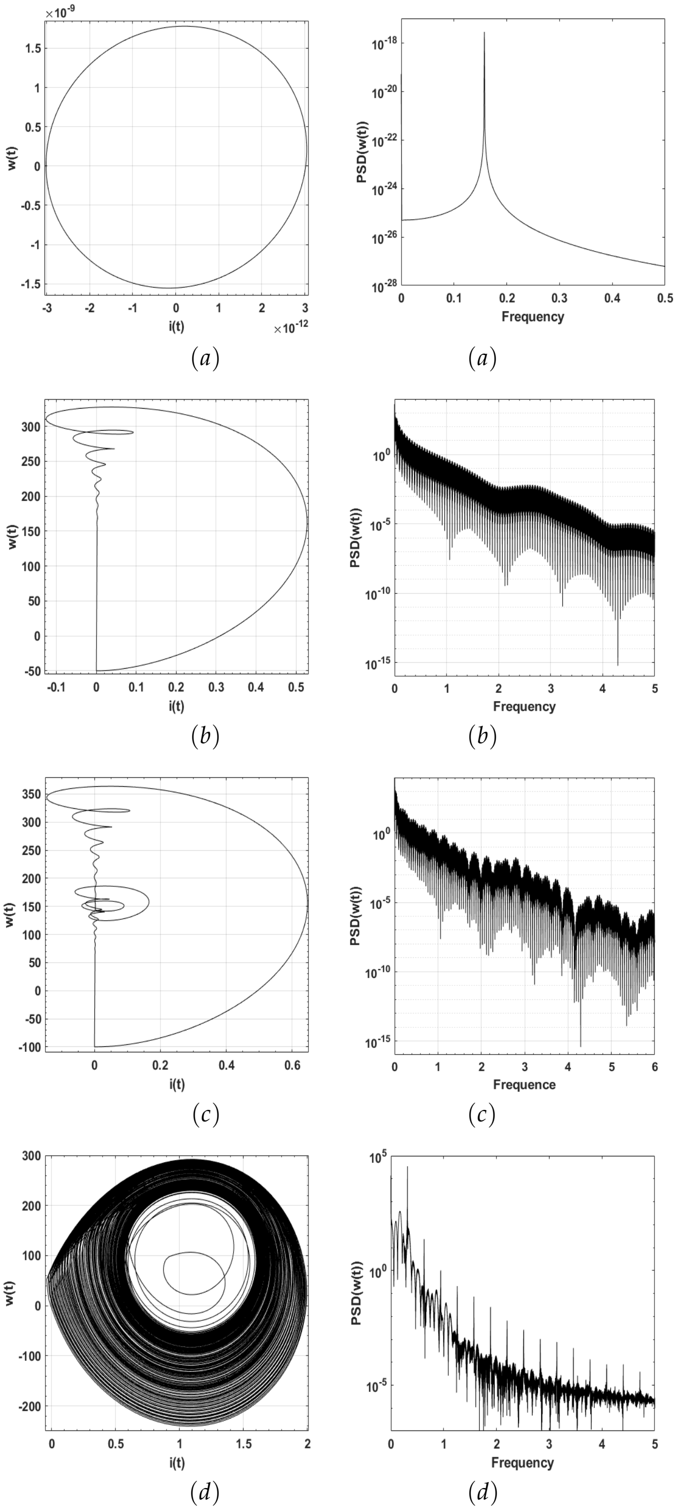
DYNAMICAL BEHAVIORS ON LOAD VARIATION OF PMDC OPERATION

This section is devoted to the numerical study of the PMDC motor in normal operation driving an external load and a triangular load.

Chaos behavior induced by an external load torque

The investigation of the external load torque effect of the general behavior of the PMDC motor is given as follows:

Firstly, when the motor drives the constant load torque (the normal driving) the behavior of the system dynamics quality changes as TL varies. The system in Figure (1) is run in Matlab-Simulink for a long time and the transient phase is omitted. The current $i(t)$ and the output speed $\omega(t)$ are recorded and plotted in Figure (2). The Figure (2) shows the different dynamics observed. We considered that the general parameter of the motor is unchanged. We recalled the parameters values: $R = 50\Omega$; $J = 0,000010388kgm^2$; $b = 0,00025N.m.s$; $K_t = 1,8N.m/A$; $L = 1,2mH$;The external load torque is varied between 0 to $5N.m$.



(a) (a)

(b) (b)

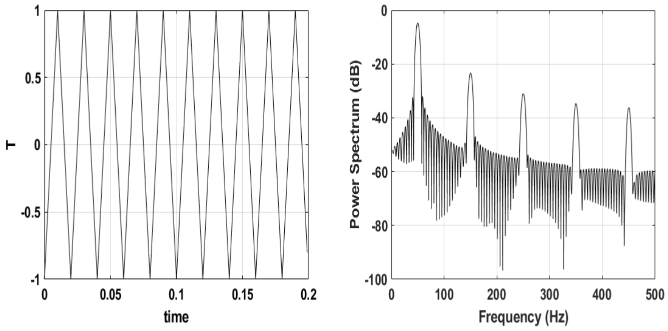
(c) (c)

(d) (d)

Figure 2 Phase portraits (i) of different dynamical behaviors of the PMDC motor driven by a full bridge converter and the corresponding Power Spectral Density (PSD) of the speed $(\omega(t))$ (ii) under variation of the amplitude of the external linear$T$. The values of T are given as (a) $T = 0,00001N.m$ period-1; (b) $T = 0,0125N.m$ periodic bursting; (c) $T = 0,5N.m$ periodic bursting; (d) $T = 1,95N.m$ chaotic motion. The rest of the parameters of the system are the ones of Figure (1). The Figure (2) shows that the behavior of the motor changes from the period limit cycle, to periodic bursting, to chaotic behaviors. The PSD plots highlighted the different kinds of the observed attractors.

Effect of the amplitude variation of the triangular T

In the industry, some load applied to the PMDC motor are non-constant, but varies continuously. To mimic this type of load, we introduced here a triangular load. The triangular signal is periodic linear in pieces and continuous, it contains odd harmonics. It can be seen as the absolute value of the saw-tooth signal and for a period of time as follows. $x(t) = 2\left|\frac{t}{a} - \left[\frac{t}{a} + \frac{1}{2}\right]\right|$. In this case, we used the signal equation as: $x(t) = A * [sawtooth(2.\pi.50.t, 1/2)]$. available in Matlab-Simulink. The triangular load torque and the corresponding PSD graph are shown in Figure (3).



(a) (b)

Figure 3 Triangular waveform T (a); and the corresponding PSD (b). The amplitude Ttri=1N.m; and the frequency is $F = 50H_z$. Sub-harmonics are visible. $A = 1$.

The effects of the amplitude variation of the triangular external load torque are investigated in this sub-section. For this purpose, the system in Figure (1) is run in Matlab-Simulink with load torque T replaced by the triangular one $Tri(t)$ for a long time and only the steady state data are recorded. The parameters of the motor are given as: $R = 50\Omega$; $J = 0,000010388kgm^2$; $b = 0,00025N.m.s$; $K_t = 1,8N.m/A$; $L = 1,2mH$;$T = 2N.m$; We plotted different current $i(t)$ and speed $\omega(t)$ recorded as in sub-section 3.1 to describe the situation. The variation of T highlights some unobserved streaking dynamical phenomena.

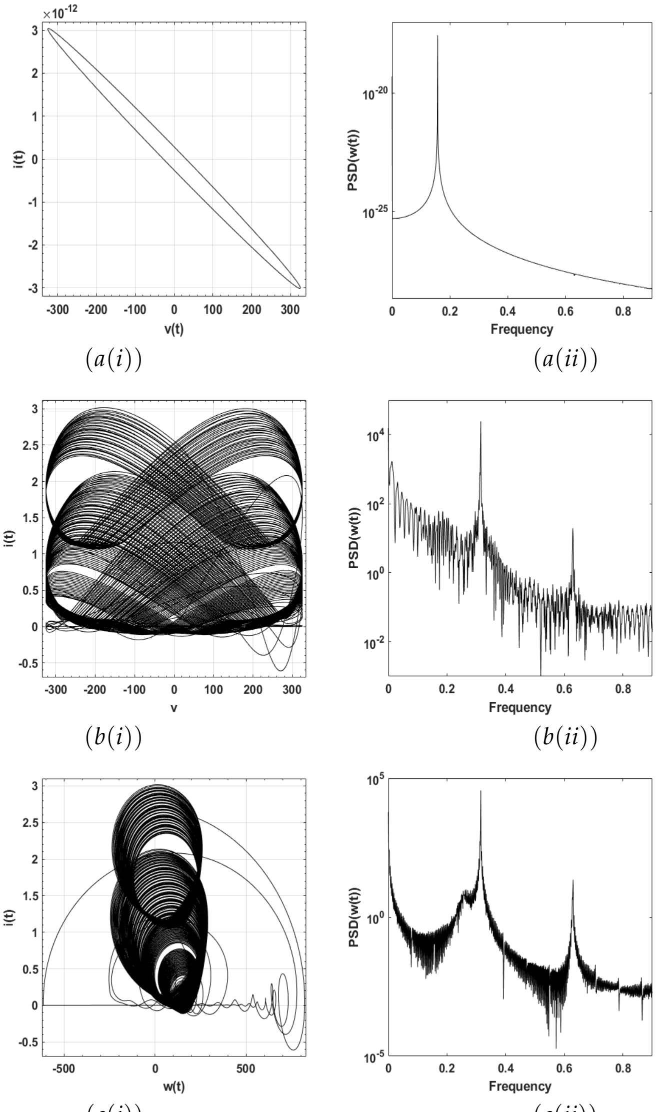
In the light of Figure (4), it is easy to see that when the total load torque driving by the PMDC motor have a triangle form, the motor develops different behavior like periodic, torus and chaos behaviors.

Dynamical behaviors induced by the windings heat during PMDC operation

In the industry the load driven by the system (1) transforms electrical energy into mechanical energy for the manufacturing process. Energy laws states that this transformation cannot occurred without loss of energy. In this paper we focused on the Joule energy effects dissipated by the winding's coils heat due to current flowing in these elements.

VARIATION OF THE WINDING'S COIL RESISTANCE OF THE PMDC MOTOR

When the motor drive different type of load torques, the general temperature of the systems increases, and some parameters are affected such as resistance and reactance winding coil of the motor. The general equation of the resistance as function of the temperature is give as $R = R_0\left(1 + A(\theta - \theta_0) + B(\theta - \theta_0)^2 + ...\right)$



$(a(i))$ $(a(ii))$

$(b(i))$ $(b(ii))$

$(c(i))$ $(c(ii))$

Figure 4 Phase portraits (i) and the corresponding PSD (ii) describing the dynamics of the system under variation of the amplitude of the external triangular load torque A. The values of A for each case are given as: $(a(i))$ $A = 0,00001N.m$, period-1; $(b(i))$ $A = 0,025N.m$, torus; $(c(i))$: $A = 5N.m$, chaos the initial condition are $(i(0) = 0,01; \omega(0) = 0,01)$.

(Malvino 1993) and the simplified form of this equation can be deduced as:

$$R = R_0(1 + \alpha\theta) \tag{5}$$

.

Where R_0 is a value of the resistance when $\theta = \theta_0$, A et B are the thermal coefficients depending of the type of the motor resistance coil used. In the case the temperature is varying between $0°C$ to $100°C$ and when the reference temperature is equal to zero, the thermal coefficient is replaced by the average thermal coefficient α.

Using equation (5), we replaced the resistor R in equations (1) and we derived the new equations of system in Figure (1), depending on the internal heat of the motor give in equations (6) and (7) as follows:

$$\begin{cases} \frac{d\omega(t)}{dt} = \frac{(K_t i(t) - T_L - b\omega(t))}{J} \\ \frac{di(t)}{dt} = \frac{(-i(t)R_0(1+\alpha\theta) - K_e\omega(t) + Vin)}{L} \end{cases} \tag{6}$$

when $A2$ is in the high and

$$\begin{cases} \frac{d\omega(t)}{dt} = \frac{(K_t i(t) - T_L - b\omega(t))}{J} \\ \frac{di(t)}{dt} = \frac{(-i(t)R_0(1+\alpha\theta) - K_e\omega(t))}{L} \end{cases} \tag{7}$$

When the $A2$ output is low.

By defining the state vector $X(t) = [x_1(t), x_2(t)]^t = [\omega(t), i(t)]^t$,The DC drive + full bridge can be expressed in standard form as in Equation. (8).

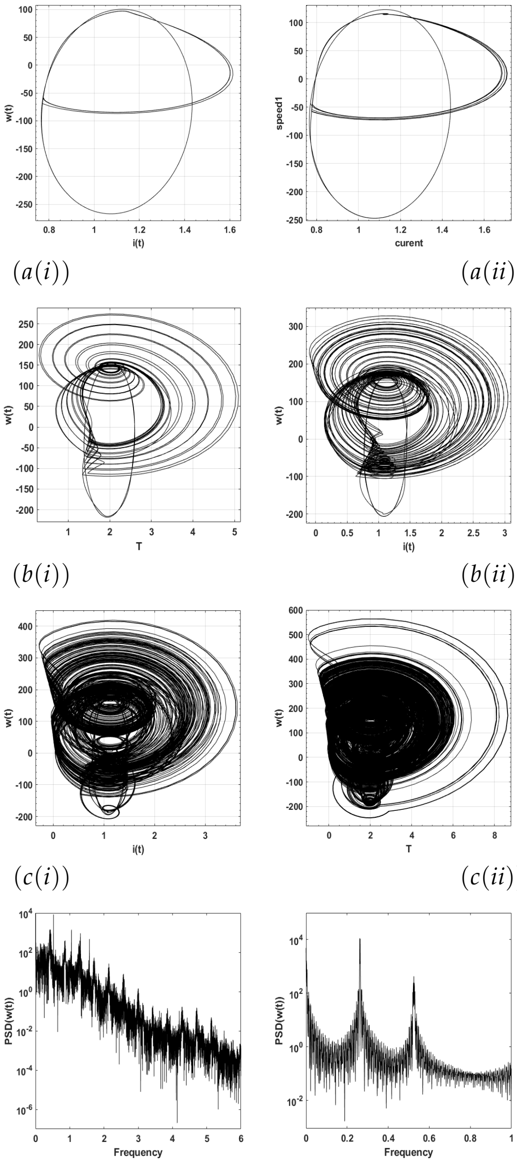
$$\begin{cases} \dot{X} = A_{on}X(t) + Von \\ \dot{X} = A_{off}X(t) + Voff \end{cases} ; where X(t) = \begin{bmatrix} x_1(t) \\ x_2(t) \end{bmatrix} = \begin{bmatrix} \omega(t) \\ i(t) \end{bmatrix} \tag{8}$$

$$A_{on} = A_{off} = A = \begin{bmatrix} \frac{-b}{J} & \frac{k_t}{J} \\ \frac{-k_e}{L} & \frac{-R_0(1+\alpha\theta)}{L} \end{bmatrix}; \qquad V_{on} = \begin{bmatrix} \frac{-T_L}{J} \\ \frac{Vin}{L} \end{bmatrix}; V_{off} = \begin{bmatrix} \frac{-T_L}{J} \\ 0 \end{bmatrix} \tag{9}$$

Where the system parameters are the same as in section 1.

Chaos behavior induced by the resistance variation

The system (1) is used in Matlab-Simulink with the resistor as in equation (5) and the motor is powered by the inverter that drives a linear load. The curves in the following Figures are obtained by varying the total resistance of the machine as a function of temperature. It is noted that in an electric motor driving a load torque, the energy is dissipated by thermal effect in the form $J = RI^2$ This dissipation of heat acts at the level of the different resistances of the motor windings coil due to the modification of temperature according to law of equation (5). (the resistance at an initial temperature which is the ambient temperature in our case it is 37° C) We remarked that this variation of the temperature in an electric motor driving a load can be at the origin of strange phenomena such as periodic limit cycles, torus and chaos as illustrated in section 3. Noted that the general form of the total impedance of the motor is give as: $Z = R + jX$ where R is the total resistance and X is the total inductive part. In this special case we suppose firstly that the total reactance of the motor is unchanged and the total resistance of the motor is varied by the different degree of the temperature when the system drive the linear load torque as shown in equation (5) assuming that the parameters of the motor in this study is given as: $L = 1,2mH$; $J = 0,000010388kgm^2$; $b = 0,00025N.m.s$; $K_t = 1,8N.m/A$;$T = 2N.m$; and R vary between $1,2\Omega to 3000\Omega$. After long-term running of system (1) in the MATLAB-Simulink, we obtained some phase portraits showing different form of attractors of dynamical the behaviors of the motor in the plane $(w(t), i(t))$ in the Figure (5).



$(a(i))$ $(a(ii))$

$(b(i))$ $(b(ii))$

$(c(i))$ $(c(ii))$

$(d(i))$ $(d(ii))$

Figure 5 Phase portraits describing the different behaviors of the motor and the corresponding PSD when the resistance R is varied between 1.2Ω to 3000Ω due to heat: $(a(i))$ $R = 135\Omega$;$(a(ii))$; $R = 100\Omega$:$(b(i))$ $R = 50\Omega$; $(b(ii))$ $R = 30\Omega$; $(c(i))$ $R = 10\Omega$; $(c(ii))$ $R = 1,2\Omega$; $(d(i))$ PSD for $R = 1,2\Omega$; $(d(ii))$ PSD for $R = 100\Omega$ the initial conditions are: $(i(0) = 0,01; \omega(0) = 0,01)$;

In the light of the curves in Figure (5), the reader can see that when the DC motor + full bridge drives some external load torque, their internal temperature changes and affects the global resistance of the system and induced some interesting dynamics like period limit cycles and chaotic attractors. These interesting behaviors are found in many physical systems but not yet observed in this system caused by windings coil heat. This contribution falls in the second criteria of publishing novel contribution of chaotic system (Sprott 2011) and important to shared.

Chaos behavior induced by the reactance variation

In this section, we set the resistance constant to $R = 100\Omega$ (no longer influenced by the coil heat) and then varying the reactance L. For the simplicity of this paper the phenomena causing the variation of $L$is not illustrated. The permanent magnet is a magnetic material (strong) is characterised by the large hysteresis cycle the variation of the reducing induction as a function of the temperature is governed by the following equation; This equation is valid in a certain temperature range. In this equation is the remnant induction at temperature T ,B_{r0} is the induction at temperature T_0 and β is the temperature coefficient. It should be noted that this coefficient is negative, which is why the induction decreases when the temperature increases The greater this coefficient (in absolute value), the greater the drop in induction B.

$$B_r = B_{r0}\left(1 + \beta\left(T - T_0\right)\right) \tag{10}$$

where $B = L = Xw$ in this case we have:

$$\begin{cases} \frac{d\omega(t)}{dt} = \frac{(K_t i(t) - T_L - b\omega(t))}{J} \\ \frac{di(t)}{dt} = \frac{(-i(t)R_0 - K_e\omega(t) + Vin)}{L(1+\alpha\theta)} \end{cases} \tag{11}$$

when the $A2$ output is hiegh, and

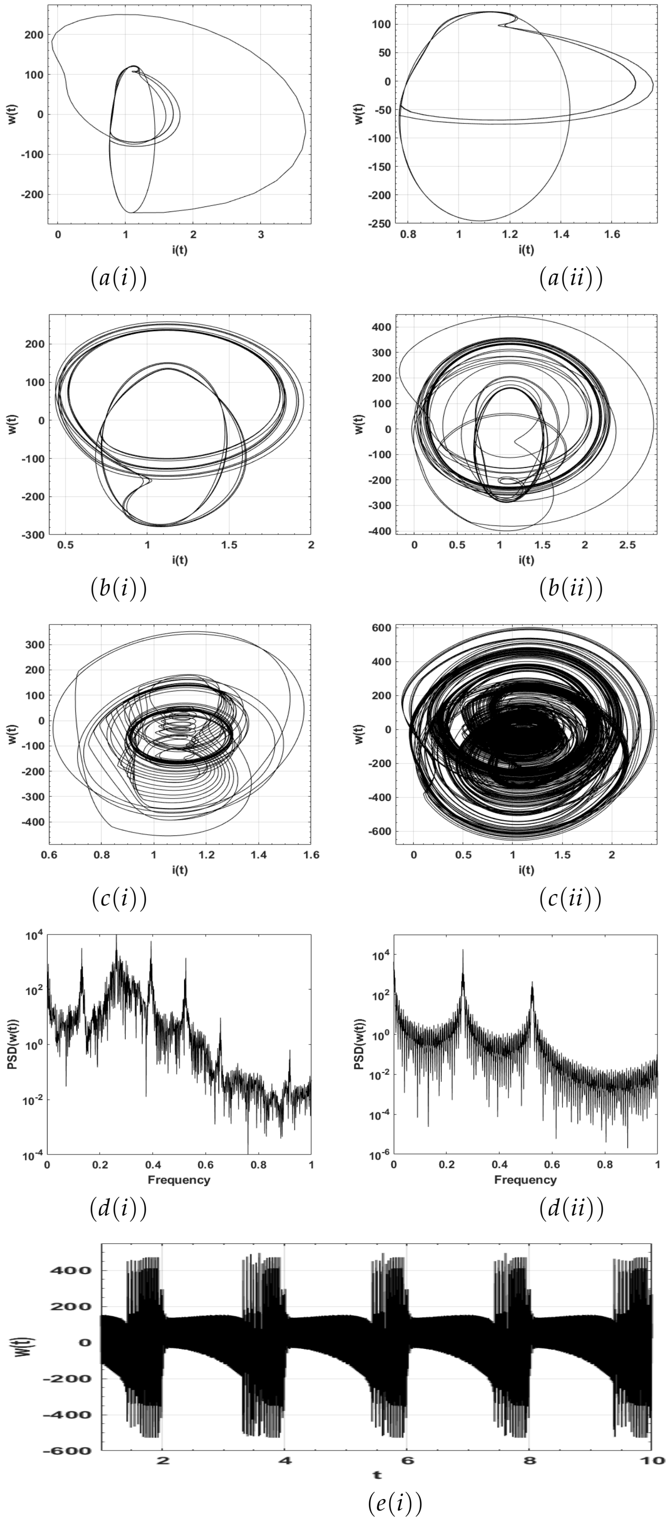
$$\begin{cases} \frac{d\omega(t)}{dt} = \frac{(K_t i(t) - T_L - b\omega(t))}{J} \\ \frac{di(t)}{dt} = \frac{(-i(t)R_0 - K_e\omega(t))}{L(1+\alpha\theta)} \end{cases} \tag{12}$$

when the $A2$ output is low. By defining the state vector $X(t) = [x_1(t), x_2(t)]^t = [\omega(t), i(t)]^t$. the DC drive + full bridge can be expressed in matrix form as given in equation (13)

$$\begin{cases} \dot{X} = A_{on}X(t) + Von \\ \dot{X} = A_{off}X(t) + Voff \end{cases}$$
$$where X(t) = \begin{bmatrix} x_1(t) \\ x_2(t) \end{bmatrix} = \begin{bmatrix} \omega(t) \\ i(t) \end{bmatrix} \tag{13}$$

$$A_{on} = A_{off} = A = \begin{bmatrix} \frac{-b}{J} & \frac{k_t}{J} \\ \frac{-k_e}{L(1+\alpha\theta)} & \frac{-R_0}{L(1+\alpha\theta)} \end{bmatrix};$$
$$V_{on} = \begin{bmatrix} \frac{-T_L}{J} \\ \frac{Vin}{L(1+\alpha\theta)} \end{bmatrix}; V_{off} = \begin{bmatrix} \frac{-T_L}{J} \\ 0 \end{bmatrix}; \tag{14}$$

The phase portraits describing the different behaviors of the motor when the inductance varied are given in the Figure (6) .



$(a(i))$ $(a(ii))$

$(b(i))$ $(b(ii))$

$(c(i))$ $(c(ii))$

$(d(i))$ $(d(ii))$

$(e(i))$

Figure 6 phase portraits in plane $(w(t), i(t))$ and the PSD describing the different behaviors of the motor when the inductance value L is varied between $0,042mH$ to $10,2mH$ $(a(i))$ $L = 0,042mH$ period-1,$(a(ii))$ $L = 0,085mH$ period-2, $(b(i))$ $L = 0,5mH$ period-8, $(b(ii))$ $L = 0,7mH$ period-16,$(c(i))$ $L = 0,85mH$ period-32, $(c(ii))$ $L = 1,952395mH$ chaos PSD of speed $\Omega(t)$ $(d(i))$ $L = 1,952395mH$ chaos and $(d(ii))$ $L = 0,085mH$ periodic,$(e(i))$ the time series of the speed $w(t)$,the initials conditions are $(i(0) = 0,01; \omega(0) = 0,01$; and the rest of parameters are $R = 100\Omega$; $J = 0,000010388kgm^2$; $b = 0,00025N.m.s$; $K_t = 1,8N.m/A$;$T = 2N.m$; .

In the light of Figure (6), it is easy to see that when the inductance value changes, some interesting dynamics occur in the system (1) such as limit cycle, torus and chaos.

CHAOS SUPPRESSION IN THE PMDC MOTOR

This section is focus on the chaos suppression in the studied system. Note that in the literature, different type of controllers to suppressed chaos have been presented. For example, the time delay controller, sliding mode controller, indirect field controller and so on; a new control structure for applications such as electric traction without mechanical sensors studied is the Direct Torque Control. This technique has many advantages over classic oriented flow vector control (FOC) structure. Indeed, while this typically requires three control loops, a PWM current generator and coordinate transformations, DTC requires only one pair of comparators to hysteresis to perform dynamic flow and torque control. The objective of this part of work is the study the effect of this controller in electric drive.

The Direct Torque Control (DTC, or DTFC) method from appointment The Anglo-Saxon "Direct Torque (and Flux) Control" was developed in 1985 by Takahashi and Depenbrock especially for asynchronous machines (Chergui *et al.* 2020) In this technique, it is no longer needed to use the position of the rotor to choose the voltage vector, this particularity defines DTC as a well-suited method for the Mechanical sensorless control of AC machines. The DTC control of an asynchronous machine is based on the direct determination of the Control sequences applied to switches in a voltage inverter from the calculated values of the stator flux and the torque . (Toufouti *et al.* 2007) So the state of the switches is linked directly to the evolution of the electromagnetic state of the motor. Direct torque control of the machine provides a satisfactory solution robustness problems encountered in conventional control technology based on the orientation of the rotor flow.

On the other hand, DTC, is as simple, interesting given its simplicity; in particular, by the fact that it does not requires no real-time speed measurement or complex modulation control Pulse Width (PWM) close to the inverter. Under certain condition, the motor drive exhibit chaotic phenomena and the chaos suppression from the system is to be forcing the system from chaos regime to periodic oscillation. In this situation, we use the gain of the different between electromagnetic torque produced by the motor and the resistive external torque. The block diagram described the form of the controller apply in this system in our cases is shown in the Figure (7).

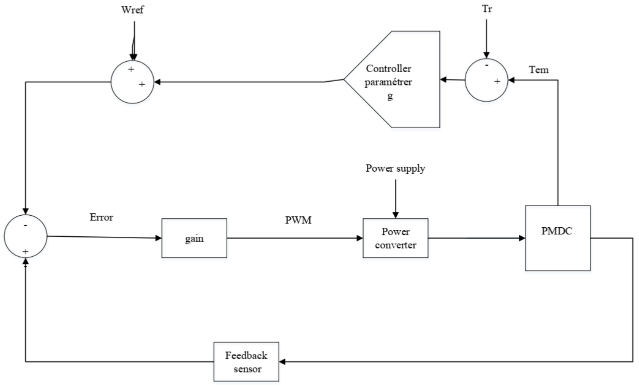
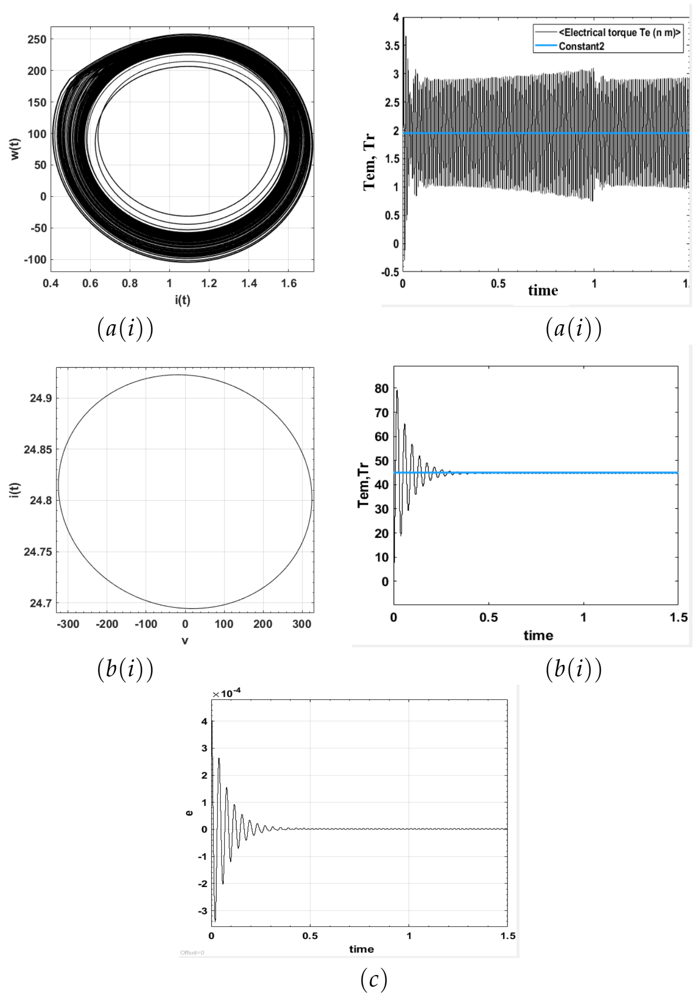


Figure 7 Block diagram of the PMDC drive with a controller parameter

The general form of the mechanical equation of the motor is giving by apply the second low of newton as follows:

$$j\frac{dw}{dt} + fw = T_{em} - T_r \tag{15}$$

where j being the inertia moment of the motor, T_{em} is the electromagnetic torque produced by the motor, f is the coefficient viscous of friction, T_r is the external torque . we chose the load torque equation as: $T = T_{em} - T_r$ in our case; we use the gain of the different between the load torque to force the system from chaotic regime to the periodic oscillations. The equation describing this situation is given as $T = g(T_{em} - T_r)$ where g is the gain. From Matlab-Simulink environment when the system (1) is ran for long times, the controlled parameter varying here between 0 to 1.5 .We realized that when $g = [0.0001; 0.05]$ the system is under control; the special case is when $g = 0.0025$ after $0.398s$ the total load torque is equal to zero and the behavior of the motor changes from chaotic to periodic regime. the time series in the Figure (8) bellow describes the situation where in Figure (8(b)) the chaos is suppressed and the system is periodic.



$(a(i))$ $(a(i))$

$(b(i))$ $(b(i))$

(c)

Figure 8 Phase portrait of chaotic behavior a(i) and the time series of the electromagnetic torque +with external torque superimposed $g = 1.02$ Periodic phase portrait after tuning parameter g b(i) and the corresponding time traces of T_{em} and T_r superimposed (ci) is the error between the too different load torque when the gain g is $g = 0.00025$. the initials conditions of the system are $(i(0) = 0,01; w(0) = 0,01)$.

In the light of Figure (8), it is easy to see that the proposed controller is effective and therefore, the system is forced to exit from chaotic regime and settle in the periodic oscillations after some small time of 0.395s.

CONCLUSION

In this paper the full bridge converter driving the PMDC motor with load is analysed under particular situation where the temperature affects the resistance of the inductive windings of the engine. We used simple power full tool of analysing nonlinear dynamical systems like phase portrait, time traces and power density spectrum diagram. In Matlab-Simulink environment, numerical simulations of the PMDC motor show that under certain conditions of the temperature of the internal resistance and of some value of the inductance and the value of the amplitude of triangular and linear load torque, the PMDC motor and the full bridge drive under investigation shows periodic limit cycle and chaotic attractors. A chaos control of the unstable regime is proposed to stabilize the PMDC motor in a desire periodic regime. These results mean that the variation of the average internal temperature and the variation of the amplitude of different load torque influence the general behavior of the motor drive. These results would have industrial application of the PMDC motor driven by a full bridge converter because the application is widely used.

Funding

This research received no specific grant from any funding agency in the public or in the private.

Authors' contributions

A.N.T and J.R.M.P. developed the theoretical formalism. A.N.T and A.S.K.T contributed to sample preparation. A.N.T and A.C planned and carried out the simulations. A.N.T and A.S.K.T. contributed to the interpretation of the results. J.R.M.P. and A.N.T took the lead in writing the manuscript. All authors provided critical feedback and helped shape the research, analysis and manuscript.

Acknowledgments

The authors would like to thank Prof Sifeu Takougang for carefully reading the manuscript.

Data availability statement

The data generated during this study will be made available at reasonable request.

Conflicts of interest

The authors declare that they have no conflict interest regarding the publication of this paper.

LITERATURE CITED

Abdullah, M. A., F. R. Tahir, and K. M. Abdul-Hassan, 2016 Sliding mode control-based chaos stabilization in pm dc motor drive. Iraqi Journal for Electrical & Electronic Engineering **12**.

Arat, H. T., 2018 Numerical comparison of different electric motors (im and pm) effects on a hybrid electric vehicle. Avrupa Bilim ve Teknoloji Dergisi pp. 378–387.

Ayan, S. Ş. and A. Kurt, 2018 Computational bifurcation analysis to find dynamic transitions of the corticotroph model. Communications Faculty of Sciences University of Ankara Series A2-A3 Physical Sciences and Engineering **60**: 41–64.

Bonnett, A. H., 2001 Operating temperature considerations and performance characteristics for ieee 841 motors. IEEE Transactions on Industry Applications **37**: 1120–1131.

Chau, K. and Z. Wang, 2011 Introduction to chaos theory and electric drive systems .

Chergui, H. *et al.*, 2020 *Commande Directe Du Couple d'un Moteur Asynchrone avec la Reconstitution de l'Algorithme Des Commutations par la Logique Floue*. Ph.D. thesis.

Dalcalı, A., 2018 Optimal design of high performance interior pm motor for electric vehicle. The International Journal of Energy and Engineering Sciences **3**: 46–54.

Fussell, B., 1993 Thermal effects on the torque-speed performance of a brushless dc motor. In *Proceedings of Electrical/Electronics Insulation Conference*, pp. 403–411, IEEE.

Gieras, J. F., 2009 *Permanent magnet motor technology: design and applications*. CRC press.

Junak, J., G. Ombach, and D. Staton, 2008 Permanent magnet dc motor brush transient thermal analysis. In *Proceedings of the 2008 International Conference on Electrical Machines*, pp. 978–1.

Klein, F. N. and M. E. Kenyon, 1984 Permanent magnet dc motors design criteria and operation advantages. IEEE transactions on industry applications pp. 1525–1531.

Liao, Y., F. Liang, and T. A. Lipo, 1995 A novel permanent magnet motor with doubly salient structure. IEEE transactions on industry applications **31**: 1069–1078.

Malvino, A., 1993 Electronic principles.

Minghui, Z. and L. Weiguo, 2010 Transient coupled electromagnetic thermal analysis of a permanent magnet brushless dc motor. In *2010 International Conference on Computer, Mechatronics, Control and Electronic Engineering*, volume 4, pp. 221–224, IEEE.

Moustafa, E., A.-A. Sobaih, B. Abozalam, and A. S. A. Mahmoud, 2021 Period-doubling bifurcation analysis and chaos control for load torque using flc. Complex & Intelligent Systems **7**: 1381–1389.

Okafor, N., 2013 *Analysis and control of nonlinear phenomena in electrical drives*. Ph.D. thesis, Newcastle University.

Okafor, N., D. Giaouris, and B. Zahawi, 2015 Analysis and control of fast scale bifurcation in series connected dc drive operating in continuous conduction mode. In *IECON 2015-41st Annual Conference of the IEEE Industrial Electronics Society*, pp. 000667–000671, IEEE.

Okafor, N., D. Giaouris, B. Zahawi, and S. Banerjee, 2010a Analysis of fast-scale instability in dc drives with full-bridge converter using filippov's method .

Okafor, N., B. Zahawi, D. Giaouris, and S. Banerjee, 2010b Chaos, coexisting attractors, and fractal basin boundaries in dc drives with full-bridge converter. In *Proceedings of 2010 IEEE International Symposium on Circuits and Systems*, pp. 129–132, IEEE.

Öztürk, H., 2020 A novel chaos application to observe performance of asynchronous machine under chaotic load. Chaos Theory and Applications **2**: 90–97.

Parsa, L. and H. A. Toliyat, 2005 Five-phase permanent-magnet motor drives. IEEE Transactions on Industry applications **41**: 30–37.

Pillay, P. and R. Krishnan, 1989 Modeling, simulation, and analysis of permanent-magnet motor drives. i. the permanent-magnet synchronous motor drive. IEEE Transactions on industry applications **25**: 265–273.

Pisarchik, A. N. and U. Feudel, 2014 Control of multistability. Physics Reports **540**: 167–218.

Poliashenko, M. and C. K. Aidun, 1995 A direct method for computation of simple bifurcations. Journal of Computational Physics **121**: 246–260.

Sprott, J. C., 2011 A proposed standard for the publication of new chaotic systems. International Journal of Bifurcation and Chaos **21**: 2391–2394.

Tahir, F. R., K. M. Abdul-Hassan, M. A. Abdullah, V.-T. Pham, T. M. Hoang, *et al.*, 2017 Analysis and stabilization of chaos in permanent magnet dc motor driver. International Journal of Bifurcation and Chaos **27**: 1750173.

Toufouti, R., S. Meziane, and H. Benalla, 2007 Direct torque control strategy of induction motors. Acta Electrotechnica et Informatica **1**: 1–7.

Tsafack, A. S. K., R. Kengne, A. Cheukem, J. R. M. Pone, and G. Kenne, 2020 Chaos control using self-feedback delay controller and electronic implementation in ifoc of 3-phase induction motor. Chaos Theory and Applications **2**: 40–48.

Yu, J., T. Zhang, and J. Qian, 2011 *Electrical motor products: international energy-efficiency standards and testing methods*. Elsevier.

How to cite this article: Topy, A. N., Pone, J.R.M., Tsafack, A.S.K., and Cheukem, A. Nonlinear Dynamics Induced by Coil Heat in the PMDC Motor and Control *Chaos Theory and Applications*, 6(4), 228-236, 2024.

CHAOS
Theory and Applications
in Applied Sciences and Engineering

e-ISSN: 2687-4539
RESEARCH ARTICLE
Vol.6 / No.4 / 2024 / pp.237-248
https://doi.org/10.51537/chaos.1539877

Detection of Two-Phase Slug Flow Film Thickness by Ultrasonic Reflection

Lalu Febrian Wiranata [*,α,1], **Narendra Kurnia Putra** [*,2] **and Deddy Kurniadi** [*,3]
[*]Engineering Physics, Institut Teknologi Bandung, Jawa Barat, Indonesia, [α]Engineering Automation, Politeknik Negeri Bali, Bali, Indonesia.

ABSTRACT This study aims to detect and analyze slug flow film thickness in two-phase flow, providing detailed structural flow information. The ultrasonic or Doppler reflection method is employed to identify slug flow and detect detailed thickness. Additionally, electrical resistance tomography is used to image and confirm the presence of slug flow. A high-speed camera records the slug flow's shape in real-time, validating its existence. The ultrasonic reflection method offers high accuracy, with a measurement error rate of less than 1% based on experimental results. The study uses a homogeneous block calibration method to measure slug flow thickness. Graphical results reveal apparent differences between the slug flow regime, inner pipe wall, and outer pipe wall, with the first echo of slug flow being easily observable. The accuracy of results is attributed to the combination of field programmable gate array instruments and measurement methods, showcasing the study's novel approach. This research introduces a new perspective or novelty on slug flow in multiphase flow studies, highlighting an innovative method for detecting film thickness.

KEYWORDS
Slug flow film thickness
Ultrasonic reflection
Velocity flow
Electrical resistance tomography
Two-phase flow

INTRODUCTION

Slug flow is a common phenomenon in closed pipes, occurring in two-phase flow where liquid and gas phases mix, impacting the custody transfer or the efficiency of chemical reactions. Where, the slug flow pattern is a multiphase flow condition in a pipe system when the liquid phase and gas phase flow together in the form of slugs (bars or large oval bubbles), which are not homogeneous and can be accompanied by tiny bubbles. Usually slug flow occurs in piping systems that are arranged horizontally or vertically. To better understand the characteristics of slug flow conditions that often occur, several slug flow conditions are shown in (Falcone *et al.* 2010; Zhai *et al.* 2021), where slug flow can be categorized into several types of flow, namely plug, low-aerated, high-aerated to pseudo slug flow. This condition indicates that the slug combines flow rates with large and several tiny bubbles. The slug flow category is also shown in (Zhai *et al.* 2023), which maps the gas-liquid flow pattern in horizontal pipe conditions.

One of the main detrimental impacts of flow rate is increased pressure loss in a pipe or flow system. This is caused by the interaction between the liquid and gas phases, which produces turbulence in the flow. A slug moving faster than the average flow can cause a significant pressure drop in the system. Additionally, slug flow can impact the flow capacity of the system, where fluctuating flow velocities and changes in the flow profile can cause variations in the flow capacity acceptable to the system. In systems susceptible to slug flow, flow fluctuations can cause problems in unstable operations (Villarreal *et al.* 2006). This may affect system reliability and operational security. For example, in the oil and gas industry, slug flow can cause problems such as pipe erosion, vibration and risk of equipment failure (Al-Safran 2009).

There are also influences in phase separation applications, such as in oil and gas phase separators. Slug flow can also affect phase separation efficiency. This can result in a less effective phase mixture, requiring unique designs or controls to overcome this problem. To overcome the impact of slug flow, various control strategies, and technologies can be used, including better pipe design, the use of phase separators, the use of slug-dampening devices, and the use of advanced measurement technology. Understanding flow characteristics and slug flow modeling is also important to manage their impacts effectively. This is the primary motivation for developing a thickness of slug flow detection system, where the characteristics of slug flow must be understood to provide

Manuscript received: 29 August 2024,
Revised: 4 October 2024,
Accepted: 8 October 2024.

[1] febrian.nata@gmail.com(**Corresponding author**).
[2] narend@itb.ac.id
[3] kurniadi@itb.ac.id

essential data for detecting flow transition patterns.

In the development of slug flow detection, Doppler ultrasonic technology was chosen for its ability to provide detailed and precise information about slug flow characteristics. Doppler can detect changes in the frequency of sound waves caused by the movement of slug flow within a fluid, enabling measurements of velocity, spatial distribution, and other characteristics of multiphase flows. Its advantages include compatibility with various fluid types, including gases and liquids, and the provision of high-resolution data. Consequently, the application of Doppler in this study enhances our understanding of flow properties and establishes a robust foundation for identifying gas-liquid flow regimes. Numerous academic references explore the flow characteristics related to Doppler signals. For example, Nnabuife *et al.* (2019) employed an ultrasonic Doppler sensor and a machine learning approach to predict gas-liquid flow regimes.

Wang *et al.* (2019) focused on detecting the spatial and temporal distribution of liquid velocity in horizontal gas-liquid flow, proposing a bubble flow identification method based on the maximum speed and maximum velocity difference ratio. Additionally, Nnabuife *et al.* (2020) utilized a deep artificial neural network to identify gas-liquid flow regimes in an S-shaped pipe, using features extracted from Doppler signals. Weiling *et al.* (2021) analyzed the Doppler spectrum via a continuous wave ultrasonic Doppler sensor, with some extracted statistical features being input into a multi-class SVM classification model to classify five oil-water flow patterns in horizontal pipes. Recent advancements have also been made in measuring velocity vector profiles using ultrasonic methods (Hitomi *et al.* 2021; Obayashi *et al.* 2008; Tiwari and Murai 2021; Zhang *et al.* 2022). Furthermore, Lin and Hanratty (1987); Shimomoto *et al.* (2021) presents a method for detecting slug flow in gas-liquid mixtures using pressure sensors, which effectively distinguishes between slugs and pseudo-slugs.

The method's key strengths include a practical experimental approach and simple technology, achieving high detection rates for intermittent slug flow. However, it faces challenges such as scalability issues, complex data analysis, double detection in intermittent flows, and reduced accuracy for continuous slug flow in simulations. Overall, while Doppler ultrasonic technology provides significant advantages in detecting slug flow through its high-resolution data and versatility across various fluid types, it also encounters challenges, including the need for advanced algorithms for complex data interpretation, scalability concerns in larger systems, issues with double detection in intermittent flows, lower accuracy in continuous slug flows in real-time industrial applications.

This study presents a new perspective method for measuring film thickness in gas-liquid slug flow using an Ultrasonic Doppler or Ultrasonic Reflection (U_R) system, with the U_R method demonstrating superiority detail in detecting slug flow structures. In contrast to the research conducted by Zhai *et al.* (2021, 2023), which primarily showcased the cut-off signal form without providing detailed signal representations during slug flow or bubble detection, resulting in suboptimal numerical graph presentations, this research approach emphasizes a more comprehensive detailed analysis of slug flow thickness signal.

The technique employed is based on homogeneous calibration block measurements. This system utilizes a Field-Programmable Gate Array (FPGA) for data acquisition from the ultrasonic transducer. The FPGA offers a relatively high sampling rate of 64 MSPS (Mega-Samples Per Second) for each data loop process, allowing for detailed information retrieval from the ultrasonic transducer. Additionally, we incorporate Electrical Resistance Tomography (ERT) as an imaging system to visualize the presence of two distinct phases in slug flow. For ERT, we employ an ARM-type microcontroller, which achieves a superior and more stable sampling rate compared to other types, delivering a sampling rate of 2.4 MSPS for each data collection session. The results indicate a commendable level of accuracy, with the U_R method effectively detecting the thickness of slug flow. Moreover, it was observed that the flow velocity, derived from the combination of ERT and U_R measurements, significantly influences the composition of slug flow, particularly its thickness.

In addition, the following parts of this article are organized as follows. Section 2 explains the primary ERT-U_R sensor's method, working principle, and corresponding to the detection slug flow film-thickness method. Section 3 is related to measurement systems and data acquisition, where the combination of FPGA instruments and measurement methods has great potential to be applied and is one of the novelty or new perspectives in this study. Apart from that, there is also an ERT system that functions as confirmation of the existence of different phases. Section 4 shows the results of the experimental verification and validation of the system used, where in this study, the first echo is very easy to observe. This can also be a problem-solving limitation, as detecting the first echo in the slug flow takes a lot of work. Therefore, measuring film thickness in a gas-liquid becomes easy to explore. Section 5 contains the conclusions of the study obtained.

MATERIAL AND METHODS

This section focuses on the basic concept of ultrasonic reflection or ultrasonic Doppler to collect surface (inner pipe wall) distance data to the ultrasonic transducer. Apart from that, f_d can also calculate the velocity of liquid and gas flow. The expected result that we want to know is the time of flight between the transmitter of ultrasonic waves and the time of receiver ultrasonic waves. However, in some situations, such as the presence of slug flow at the flow rate, it is challenging to find the desired information, thus the process of searching and identifying first echoes must be carefully observed. The next concept is the image reconstruction process using the ERT technique, and the main ERT aim is to identify essential differences between two different phases. However, of course, the ERT technique is not accurate enough to consistently determine the size and position of the slug flow, therefore in this study, ERT is only used to detect differences in images of the two objects and the average two-phase fluid velocity flow rate to ensure that it can support strengthening the presence of slugs in the fluid flow rate.

Principle of Ultrasonic Reflection or Ultrasonic Doppler

This section explains the basic concepts used to find ToF due to slug flow in the pipe cross-section. As seen in Figure 1, f_0 is the received frequency, f_d is the shift frequency, θ is a degree of transducer position in the pipe (Falcone *et al.* 2010; Zhang *et al.* 2022). For the c_{water} value, it is 1497 m/s (Bao *et al.* 2022). Apart from that, there is also h, which is the position or depth of the echo pulse from the first echo detection in slug flow conditions, where Δt is the time difference between the transmitter and receiver, which can be stated that the value of f is proportional to the inverse of t.

$$f_d = f_r - f_0 = \frac{2U_R}{\cos\theta} f_0 \tag{1}$$

$$U_R = \frac{f_d c_{\text{water}}}{2f_0 \cos\theta} \tag{2}$$

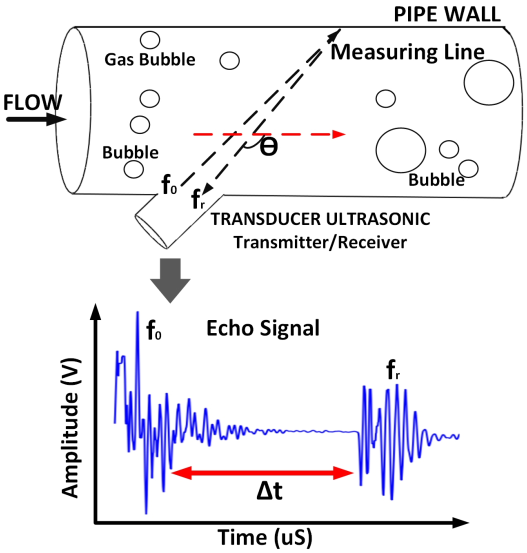


Figure 1 Principle of transduser ultrasonic reflection.

To find the average of flow velocity $\overline{U}_R$, Equation (3) is used, where A is the pipe cross-sectional area, R is the pipe radius, and y is the pipe diameter (Liu *et al.* 2018a).

$$\overline{U}_R = \frac{\int_0^{2R} U_R(y) \cdot 2\sqrt{y(2R-y)}\,dy}{A} \tag{3}$$

Meanwhile, to find the distance between the position of the transducer and the pipe, it is used Equation (4) (Zhai *et al.* 2021).

$$h = \frac{\Delta t \cdot c_{\text{water}}}{2} \tag{4}$$

Principle of Electrical Resistance Tomography Detection

Electrical resistance tomography (ERT) is an imaging technique that focuses on measuring the electrical resistance of an object or medium. The basic principle of this method is to measure the difference in resistance in a specific material or liquid, which has different characteristics from different resistance values. It can be done by exciting an AC electric current into the electrode to measure the characteristic value with a different electrode at another position. As shown in Figure 2, there are two ways to excite current into the vessel, and both of these methods are good for obtaining different signals from objects in the vessel. However, on this study, Figure 2(b) utilized opposite paired excitation because the electric current will generally pass through the container, allowing for the effective detection of the characteristics of non-uniform objects.

Opposite pair excitation method could be weak if the container is large enough. The distance between positive AC electric current and negative AC electric current will hamper the excitation current. In this study, the circle's diameter in the trial was relatively small, allowing the current to cross the object ideally. In other words, every method has its shortcomings and weaknesses. Therefore, it can be concluded that the advantage of the opposite pair method is that the sensitivity is quite good, provided that the distance between the sending and receiving electrodes is not too far, or it can be searched manually by moving objects on the vessel, where the graphic results will show a more sensitive response in any change in the movement of the object being measured.

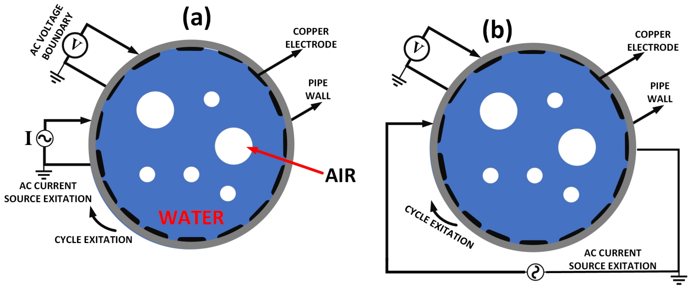


Figure 2 (a) adjacent pair excitation, (b) opposite pair excitation

Figure 3 explains the basic concept of ERT in reconstructing images, which initially come from several number value frames and then convert them into images with particular objects. For the first step, the AC electric current source is excited by rotating the 16 electrodes used, where the AC Boundary data takes the excitation data obtained 13 times with 16 measurements. Thus, we obtain 208 measurement limits for each resulting data frame. Next, the data is obtained to be used in the calculation values boundary. The data obtained is then searched for the potential distribution value with certain field boundaries, where the objective value is calculated and compared with the measurement data. If it is deemed to have met the appropriate threshold limit, then the image can be displayed. However, the iteration still needs to be increased. In that case, it is necessary to repeat the calculation of the resistance distribution values by carrying out updated calculations (conductivity updates) in the reconstruction equation (Liu *et al.* 2018b). In general, to solve the ERT problem, an approach using a forward problem is used (see Figure 3, finite element section), which aims to produce a mathematical model and can connect the distribution of resistance in the object with voltage measurements on the electrodes, where the model of the forward method in ERT is derived from the equation Maxwell which is written in Equation 5 (Liu *et al.* 2018b).

$$\nabla \cdot \sigma(x)\nabla u(x) = 0, \quad x \in \Omega \tag{5}$$

With value, $\nabla\cdot$ shows the divergence value, and ∇ is the gradient value, $\Omega \in \mathbb{R}^n$ is the data distribution in a specific domain (the region of interest). The conductivity value $\sigma(x)$ and electrical potential $u(x)$ depend on the domain where the x value is located. In simple terms, governing equations for the forward model can function to describe the relationship between the distribution of electrical resistance in an object and the measurement of voltage produced by electrodes placed around the object. Next, it is assumed that the current (I) is injected at the electrode with a specific domain boundary, which is written as $\partial\Omega$ in Equation (6) and Equation (7).

$$\int_{E_l} \sigma \frac{\partial u(x)}{\partial n} dS = I_1, \quad l = 1, 2, \ldots, L \tag{6}$$

$$\sigma \frac{\partial u(x)}{\partial n} = 0, \quad \text{on } \partial\Omega \setminus \bigcup_{l=1}^{L} E_l \tag{7}$$

Equation (6) can be said to be the integral of the electrode with current flow (integral of current density), where E_l is the field at the l-th electrode, I_1 is the current injected into E_l, L is the total number of electrodes, n is the outward normal vector, and dS is the surface of the element. Equation (7) is expressed as the equation for absence of current between one electrode and another, thus U_l

can be expressed as the l-th electrode potential, written in Equation (8).

$$u = U_l \quad \text{on } E_l, \quad l = 1, 2, \ldots, L \tag{8}$$

However, in some cases, Equation (8) fails to calculate due to the double layer, as well as the high resistance contact between the electrode surface and the surface of specimen, which is the conductor in container. This concept in electrochemistry can prevent the transfer of electrical energy, which is modeled as the electrode resistance/impedance z_1. As a result, Equation (9) is obtained (Demidenko *et al.* 2011):

$$u + z_1 \sigma \frac{\partial u(x)}{\partial n} = U_l \quad \text{on } E_l, \quad l = 1, 2, \ldots, L \tag{9}$$

Finally, the charge conservation equation is expressed in Equation (10) and Equation (11) for the ERT potential (Somersalo *et al.* 1992):

$$\sum_{l=1}^{L} I_l = 0 \tag{10}$$

$$\sum_{l=1}^{L} U_l = 0 \tag{11}$$

Equation (10) and Equation (11) are chosen as the design equations of the ERT because each induced current corresponds closely to the magnitude and direction of the opposite sign, enabling Equation (10) and Equation (11) to calculate the average current value without being induced by the proximity of electrode pair.

The Gauss-Newton method in the ERT case is used to estimate the inverse non-linear value, with the main aim of minimizing the value of objective function $J(\sigma)$, written in Equation (12) (Ruan 2016):

$$J(\sigma) = \| U_{l,\text{meas}} - F(\sigma) \|_2^2 \tag{12}$$

where $U_{l,\text{meas}}$ is the potential electrical measurement vector value and F is the forward problem, which relates to the distribution vector of voltage values in an unknown area. The measured and calculated potential values are compared with the difference value (ε), written in Equation (13):

$$U_l = U_{l,\text{meas}} + \varepsilon \tag{13}$$

This is based on the frequent occurrence of ill-posed problems, which needs to be explicitly addressed using Tikhonov regularization (Graham 2007). To minimize the ERT function, it is necessary to add several basic conductivity constants (σ_0) as prior information or obtained from direct measurements. The resistance distribution calculation function can then be written as Equation (14), where λ is the hyperparameter value (Brinckerhoff 2018).

$$J_\lambda(\sigma_{k+1}) = \| U_{l,\text{meas}} - F(\sigma_k) \|_2^2 + \lambda \| \sigma_k - \sigma_0 \|_2^2 \tag{14}$$

In addition, there is a parameter for updating the conductivity value, related to the conductivity update (σ_{k+1}), written in Equation (15):

$$\sigma_{k+1} = \sigma_k + \Delta\sigma \tag{15}$$

To calculate the value of $\Delta\sigma$, Equation (16) is used, which involves the Jacobian matrix calculation, discussed in more detail in (Brinckerhoff, 2018) :

$$\Delta\sigma = (J^T J + \lambda^2 I)^{-1} J^T (U_{l,\text{meas}} - F(\sigma)) \tag{16}$$

MEASUREMENT SYSTEM

FPGA and ERT system

This section explains the primary system used to generate transmitter/receiver signals from ultrasonic waves, where the ultrasonic transducer type uses Sonatest with a central frequency of 10 MHz, which is embedded in the bottom wall of the pipe to ensure that it forms an angle of 900. As seen in Figure 4, the pipe is attached to the wall, which aims to ensure that the transducer surface is in direct contact with two phases (liquid and air), thereby enabling maximum detection of the time of flight (ToF). This system uses a lattice iCE40HX4K FPGA type with 8Mb RAM. The working principle is that the FPGA sends high and low signals in a few microseconds to the MOSFET to open it to activate the DC-to-DC boost converter. Next, the boost converter or pulser generating the ultrasonic transducer generator will be active.

In the boost converter (NMT0572SC), there are 3 outputs 24 V, 48V, and 72 V, which have a low power consumption of 3W, and in the experiment, 48V was used to generate pulses from the ultrasonic transducer. The 48V voltage is sufficient to supply or generate the crystal in the transducer, and the amplitude can be adjusted under programming conditions using Python Code. Previously, the VHDL programming was communicated via FTDI is used to read the FPGA port, allowing the output to be directly converted into graphics.

The next step is data processing, carried out by the MCP4881, which is a component of time gain compensation (TGC), where TGC aims to compensate for the decrease in the amplitude of the ultrasonic signal, which occurs along with increasing depth when the signal propagates through different tissues or structures. The analog signal is then converted using the ADC100065 and processed briefly into a signal of a certain amplitude with a resulting sample rate of 128 MSPS each time the FPGA processes the data. The signal processed data is then sent via FTDI to USB, which in this case will be compensated into graphics. In this case, the ADC provides a 10-bit resolution, which is sufficient to analyze the output signal.

Figure 5(a) explains the configuration of the ERT system to detect the presence of slug flow or phase differences in closed flow pipes. The primary system used for control is an STM32 ARM microcontroller chip with a 32-bit Cortex CPU and 2.4 MSPS ADC, which provides a level of accuracy and precision that is reliable enough to execute DAC control, where the DAC is used to control mux/demux. In this case, an additional relay aims to separate the VCCS (voltage to control current) excitation because of the ability of the mux (multiplexer), which cannot pass AC current. The data is then received by the 12-bit RMS ADC by the demux (demultiplexer) in the form of different potential values for each electrode installed. It can be seen in Figure 5(b) that the composition form of the current excitation is colored blue while the potential difference receiver is colored green. For speed, the frequency used in executing the RMS ADC is activated at 108 MHz to ensure the data obtained can be executed quickly and accurately.

System configuration of flow measurement

Figure 6 explains the system configuration used, where the transducer is a Sonatest 10-MHz, the pipe has a diameter of 1.5 inches, a length of 2 meters, and a height of 1 meter. This system is embedded with a 3-unit detection system, namely ERT as the initial detection of two different phases, an ultrasonic transducer, and a high-speed camera with a resolution of 1080 × 2400 pixels and 960 frames per second, which aims to capture real-time images of the slug flow.

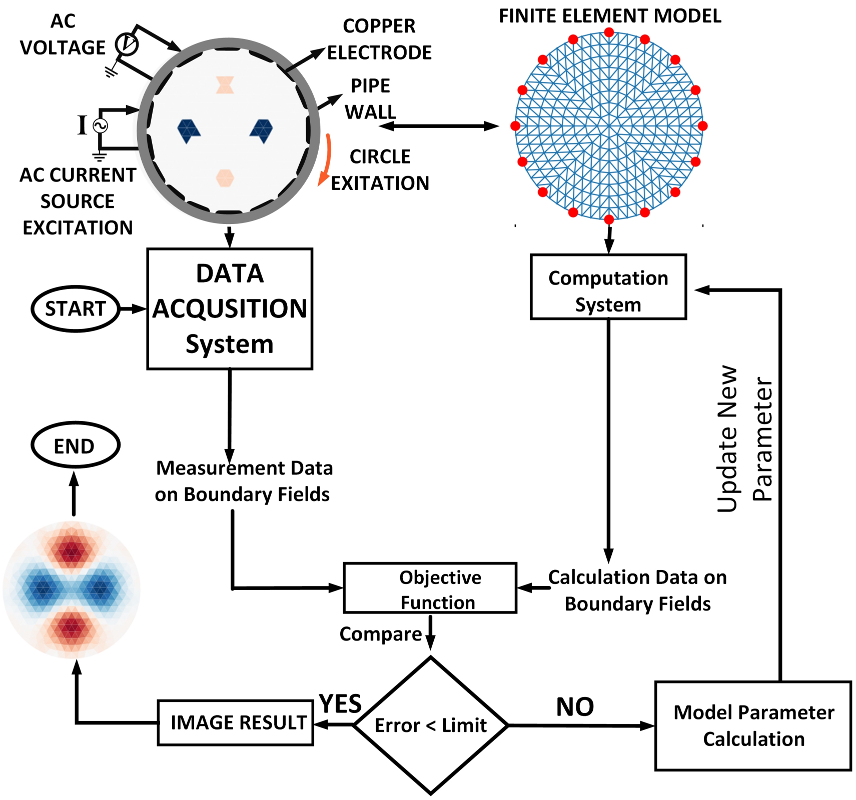


Figure 3 Data reconstruction image profile

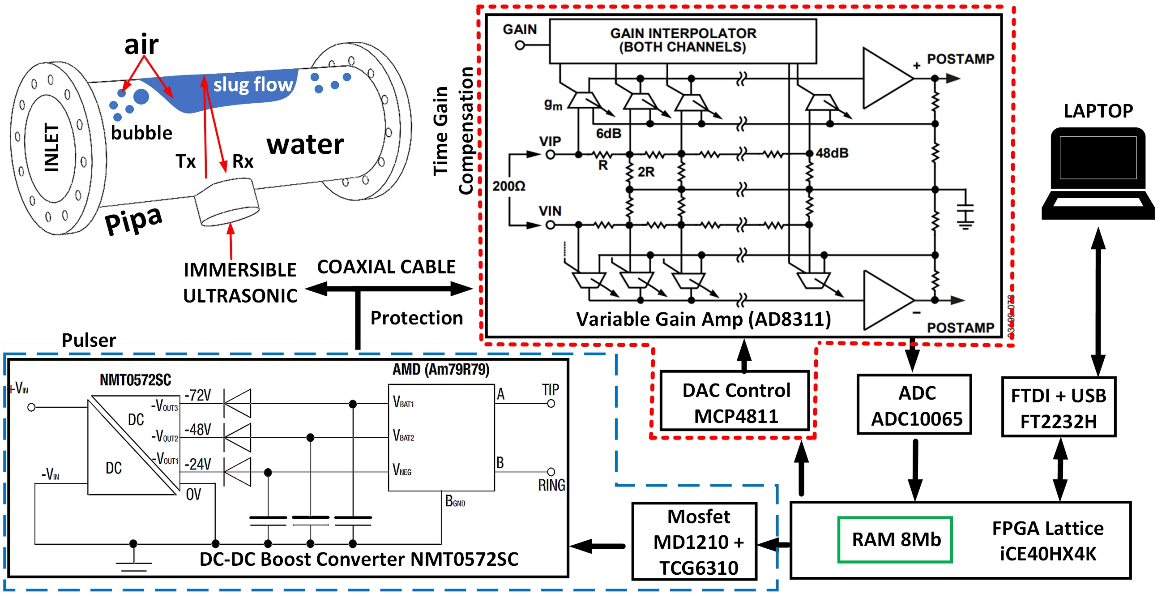


Figure 4 FPGA for ultrasonic reflection system.

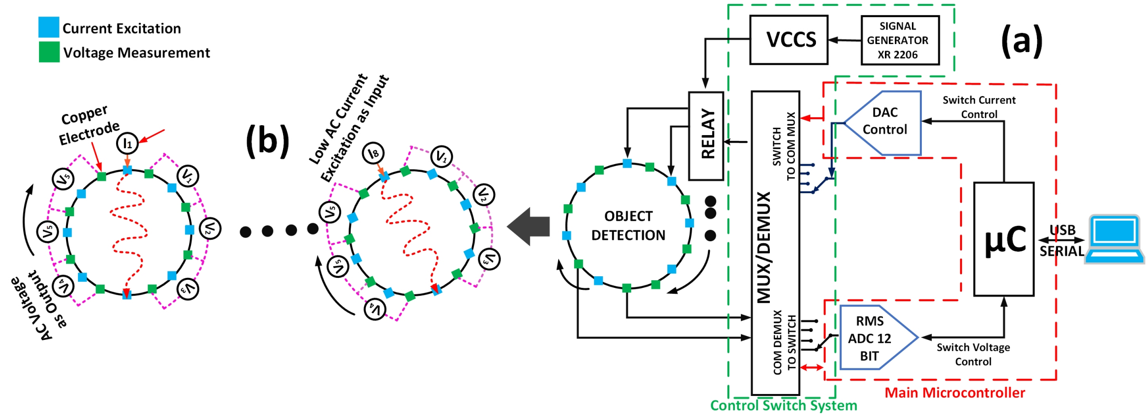


Figure 5 (a) ERT measurement system, (b) Measuring pipe tube (Wiranata et al., 2023).

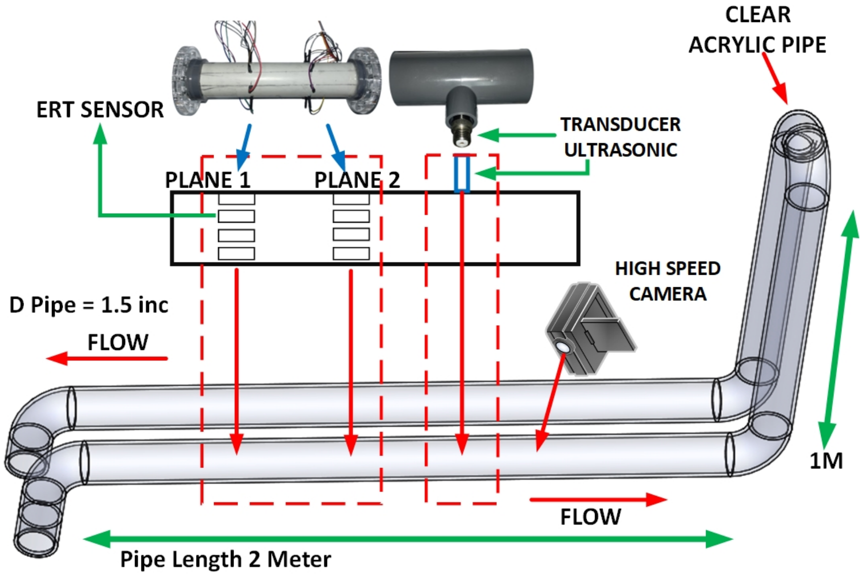


Figure 6 . Configuration System of Ultrasonic Reflection.

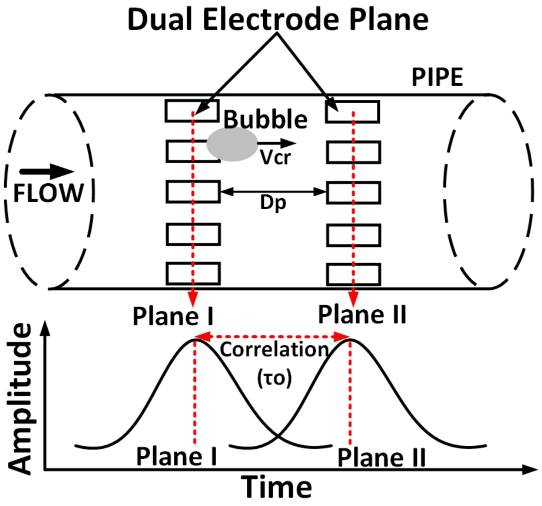


Figure 7 ERT dual plane.

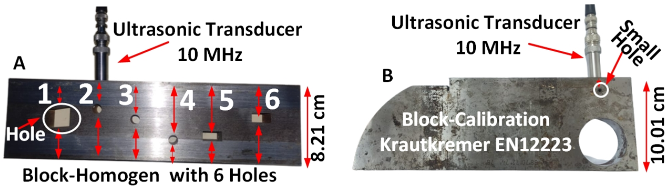


Figure 8 (a) Block homogen 6 holes, (b) Block calibration.

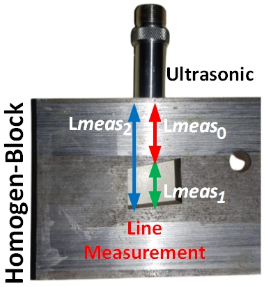


Figure 9 Block homogen measurement.

Table 1 Parameter measurement

Parameter	Values
Excitation Current AC	2 mA
Excitation Frequency	50 kHz
Electrode Number	8 Received & 8 Excitation
Data Speed Acquisition	10 frames/s
U_R	
Diameter Transducer	0.5 inch
Excitation Frequency	10 MHz
ADC Sampling Rate FPGA	64 MSPS
Excitation Voltage	48 V

Figure 7 explains how to obtain the velocity flow in a closed flow pipe using ERT, where D_p is the distance between planes I and II, and τ is the cross-sectional correlation between plane I and plane II. In Equation (18), $\overline{V_{CR}}$ is the average flow rate using dual ERT, with $V_{CR(n)}$ being the plane section (plane I or plane II).

$$V_{CR(n)} = \sum_{i,j} \frac{(V_{ij} - V_{ij0})}{V_{ij0}} \tag{17}$$

$$\overline{V_{CR}} = \lim_{T \to \infty} \left(\frac{1}{T} \int_0^T V_{CR1}(\lambda) V_{CR2}(\lambda + \tau)\, dt \right) \tag{18}$$

Placing the distance between plane I and plane II is very important for cross-correlation velocity measurement. If the distance is too far, the flow characteristics are challenging to detect, and one or two times the length of the pipe diameter is used (Deng *et al.* 2001). Table 1 explains several parameter values used in this research.

RESULT AND DISCUSSION

This section explains the results obtained from the experiment as a first step and as proof to find out the slug flow thickness and become a new perception in measuring the thickness of slug flow or become a novelty in this research, where the slug flow thickness measurement technique is adopted from the block homogeneous measurement process. The process starts from hollowing out a homogeneous block into 6 parts (see Figure 8(a)). Then, try measuring the width or thickness of the hole using ultrasonic waves transducer, as seen in Figure 8 technique used in the hole width search process. Meanwhile, the equation used refers to Equation (4) with c_{steel} 5920 m/s. The next step, namely looking for the presence of slug flow in the closed pipe cross-section, is the primary motivation for this research. Apart from that, there are also image results from photos using a high-speed camera to record the exact structure of the slug flow, and there are also imaging results from ERT as a comparison for detecting the presence of different particles in the slug flow.

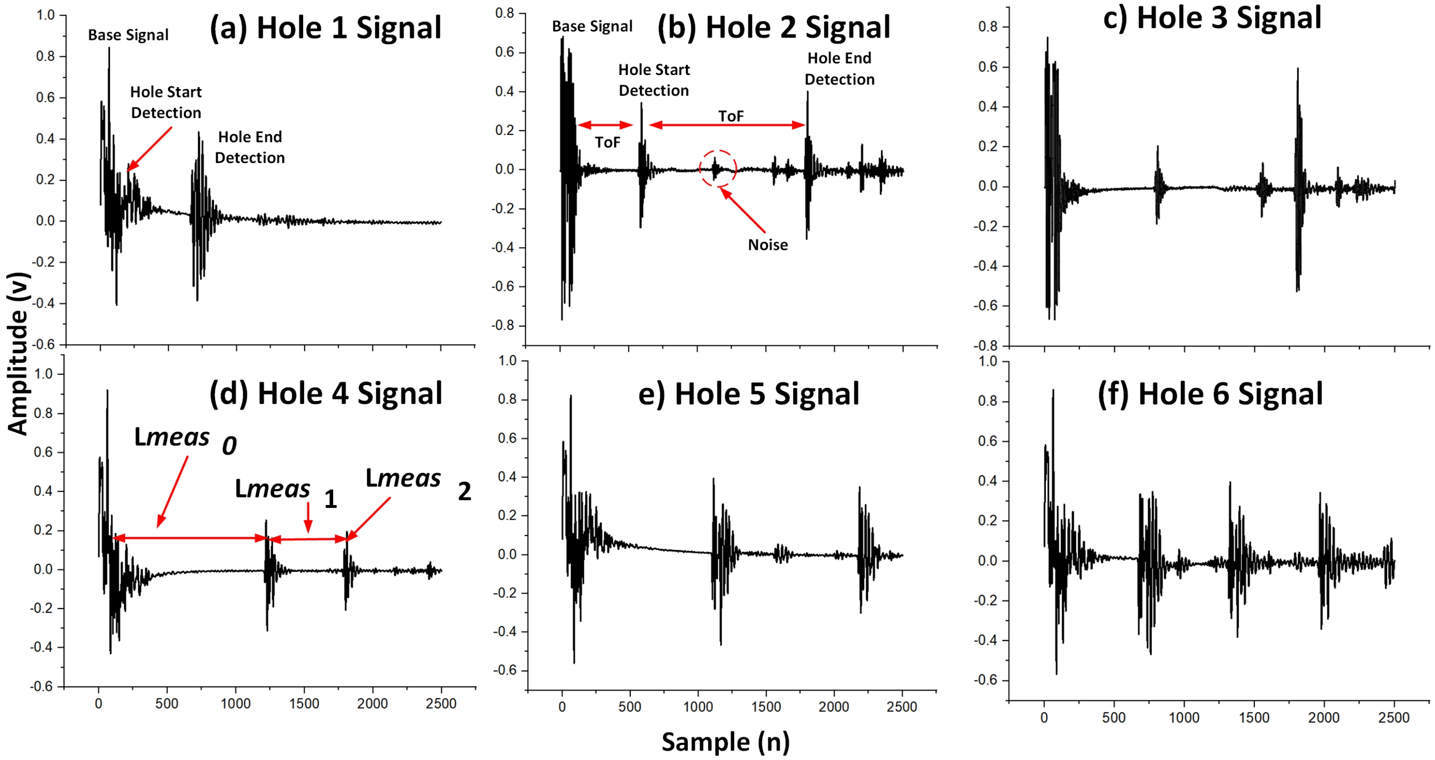


Figure 10 Percentage error between reference and measurement (a) 0.7346%. (b) 0.53066%. (c) 0.89116%. (d) 0.71103%. (e) 0.86957%. (f) 0.35891%.

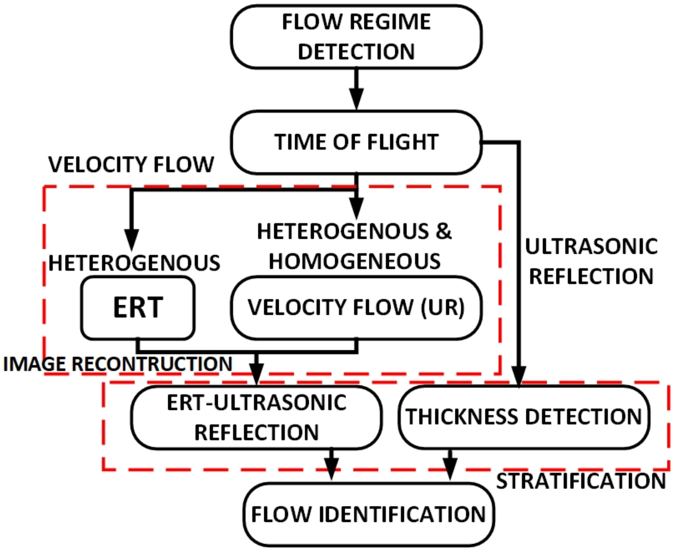


Figure 11 Flow regime identification.

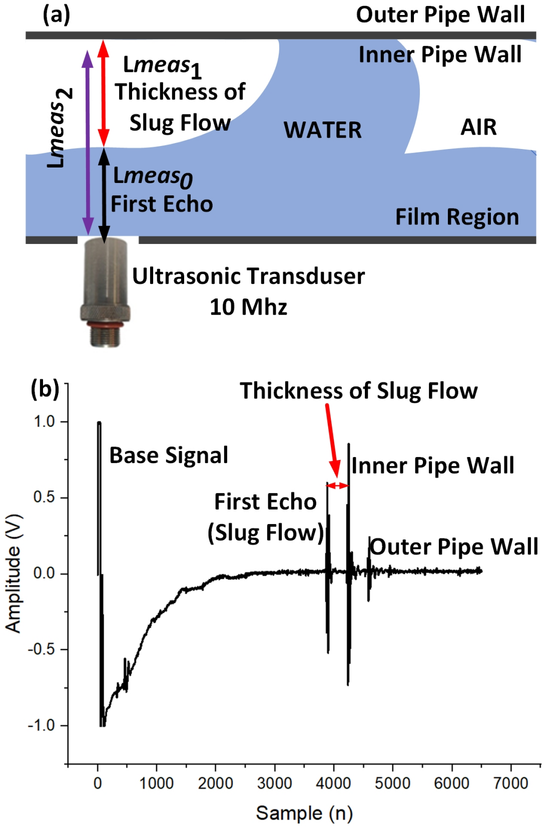


Figure 12 U_R Slug flow thickness detection and identification.

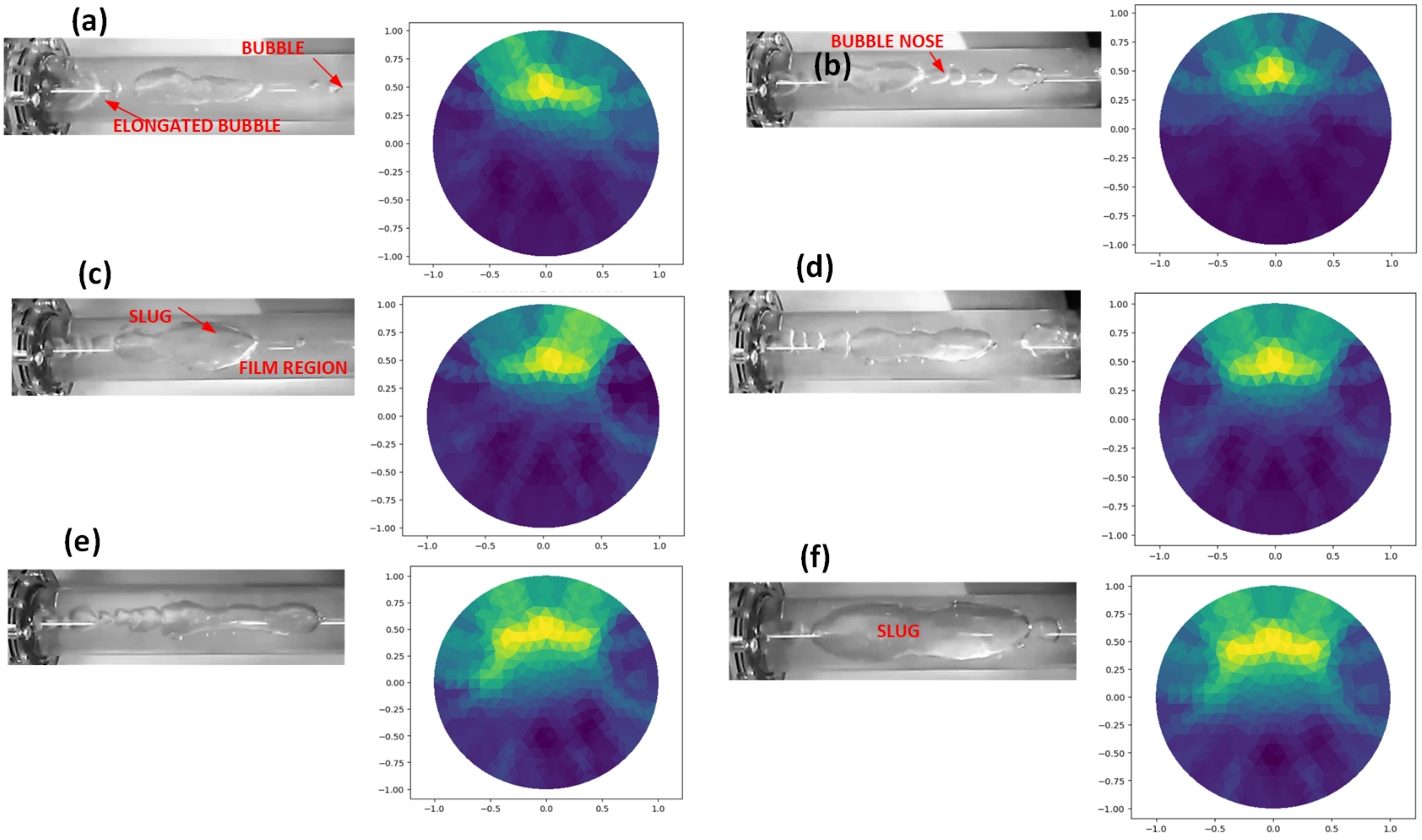


Figure 13 Velocity Flow (a) 0.73 m/s. (b) 0.76 m/s. (c) 0.95 m/s. (d) 0.98 m/s. (d) 1.18 m/s. (e) 1.2 m/s.

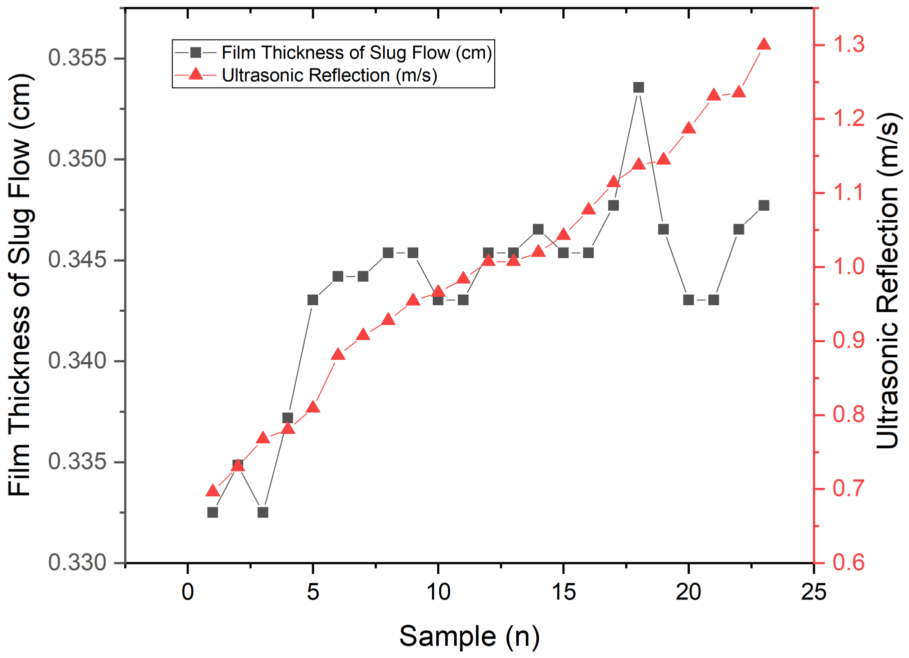


Figure 14 Film thickness of slug flow vs U_R Velocity Flow.

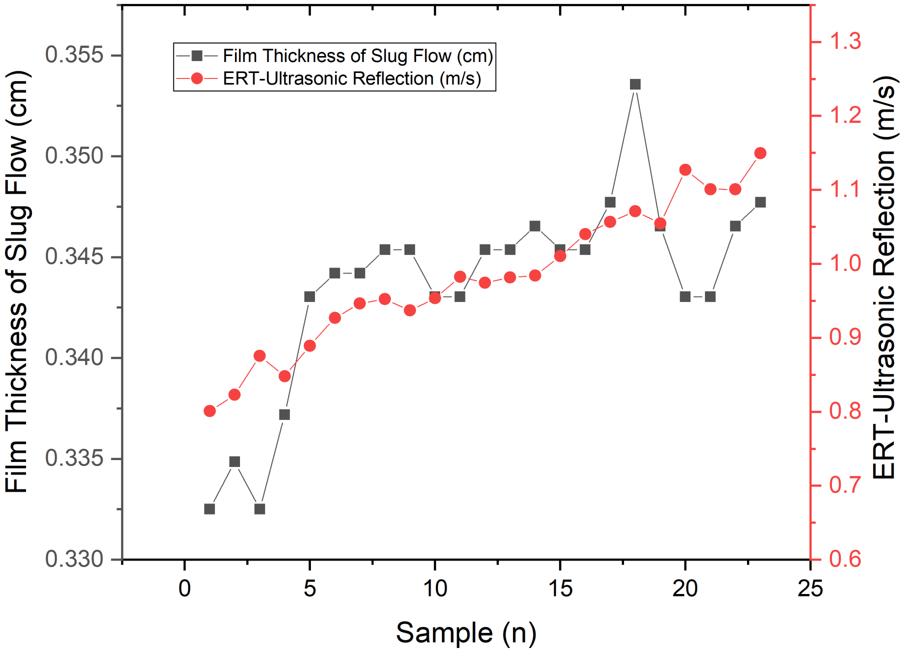


Figure 15 Film thickness of slug flow vs ERT-U_R Velocity Flow.

Validation method for measuring ultrasonic slug flow thickness

The use of homogeneous block calibration is a way to calibrate ultrasonic transducers to determine the level of error and accuracy produced by ultrasonic waves (ultrasonic transducers) to measure the thickness of objects using the ToF method, especially for the Sonatest immersible transducer type used. As seen in Figure 8. (b) The transducer is first tested on a homogeneous block calibration standard using Krautkremer EN12223. In the second step, the ultrasonic transducer was tested on a homogeneous block in which various holes were made at several points, the image of which is shown in Figure 8 (a), where the percentage error obtained by the measurement compared to the reference is no more than 1%.

To find the thickness from Figure 8(a), Equation (19) is used to determine the percentage error with Equation (20) when compared with the reference length in centimeters ($L_{\text{ref}_{\text{meas}}}$) which is the reference length from direct measurements using a digital clipper. Following Figure 9 is a detail of the measurement process.

$$L_{\text{meas}_1} = (L_{\text{meas}_2} - L_{\text{meas}_0}) - (L_{\text{meas}_2} - L_{\text{ref}_{\text{meas}}}) \tag{19}$$

$$\%\text{Error} = \frac{(L_{\text{meas}_1} - L_{\text{ref}_{\text{meas}_1}})}{L_{\text{ref}_{\text{meas}_1}}} \times 100 \tag{20}$$

Figure 10 is a graph of test results from U_R, where Figure 10 (a) to (f) represent the hole shape thickness in Figure 8(a) hole from 1 to 6. It can be seen in Figure 10 (d) and (e) that the error value is almost close to 1%. This occurs because the hole in the homogeneous block is quite far from the sensor, and there are other holes or measuring points close to other neighbor holes, which is the nature of the transducer ultrasonics have a spreading beam. Besides, the further the sound waves propagate, the bigger the beam. However, the results will be more reliable if the holes in Figure 8 (a) are reversed when measuring the hole width. This occurs because other holes do not interfere with the measuring point.

Result of Slug Flow Thickness Detection

This section will explain the method or process of flow regime detection, where the essential information is obtained using the ToF method. As seen in Figure 11, ToF is the primary key to the information obtained, where the U_R method can detect heterogeneous and homogeneous presence from two different phases. At the same time, ERT is only able to detect heterogeneously. This is because ERT generally calculates the average flow velocity along the pipe cross-section. Meanwhile, calculating the thickness of slug flow using the ToF method can immediately be the primary reference. Thus, the final result obtained is in identifying and detecting slug flow thickness and flow rate in the pipe cross-section, which can be done using the U_R method or with ERT-U_R for a combination of speed and slug flow thickness.

Figure 12(a) explains the shape of the slug flow illustration in general, which often occurs in closed pipe cross-sections. That is then continued with Figure 12(b), which explains the process of identifying and determining the thickness of slug flow in pipe cross-sections. The difference between slug flow, inner pipe wall, and outer pipe wall is visible, as a result the measurements of the slug flow's thickness can be done quickly without looking for the threshold hold of the signal obtained. In general, to detect the thickness of the threshold slug flow, it takes time to identify it, especially if there is a lot of noise specifically. In several other studies, we have to look for the first echo or the location where the slug flow occurs, which takes longer compared to the concept in this study. This is the motivation and novelty of this research, namely information on the existence of slug flow, which is entirely valid. This was obtained because the transducer used had a very high frequency, namely 10 MHz, which is rarely used to test the presence of slug flow in closed pipes. Apart from that, there is assistance from FPGA, which can process graphics significantly. In addition, the thickness of slug flow is difficult to identify in the first echo because there is phase mixing between water and air, making ToF characteristics challenging to observe. This, of course, requires accuracy to find the thickness of slug flow, but because of the accuracy of the transducer and FPGA from the system, the slug flow can be detected quite well, where the differences between each condition can be known. In addition, during the measurement process, not all flow conditions have slug flow because the characteristics of slug flow tend to change in pipe conditions that do not comply with standards. In this study, it was deliberately arranged to avoid following the rules for placing ultrasonic sensors, such as the conditions at 10xD and 5xD or other restrictions for placing sensors. This becomes a challenge due to conditions in the field, which cannot guarantee 100% laminar flow. However, if the flow is not symmetrical, it can still be detected well. Moreover, the graphic characteristics between the pipe wall and slug flow in ultrasonic transducers are very different, as seen in Figure 12, which is the characteristic shape of slug flow. In contrast, the specific characteristics of bubble flow will be explained in the following study.

Figure 13 explains the results of the combination of a high-speed camera and the imaging process using ERT seen in Figure 13 (a) – (f); there are variations in speed, where ERT is generally able to identify or detect the presence of two different phases with a yellow contrast light indicates high resistance while dark blue indicates low resistance. Or it can be stated that the bright yellow color contains the air fraction while the blue contains the water fraction, but due to spatial resolution problems, ERT is not good at imaging the shape of the slug flow in detail. Figure 13 also shows the forms of slug flow, which often occur in the trial dimension, to ensure that the general structure of slug flow can be depicted during the data collection process. If observed and compared with the reference, in this study, the form of slug flow observed was a low-aerated slug flow type, where the main profile contained a slug surrounded by a small amount of bubble flow.

Figure 14 explains the results obtained from the experiment between the number of sample film thickness of slug flow and the velocity flow results from U_R. It can be seen that the trend of the ultrasonic flowmeter is increasing because, in this case, the speed of water and air is increased slowly, where, in this case, the experiment can only be carried out by increasing both speeds simultaneously, aiming to see an increase in the same volume in each condition. In the beginning, the thickness of the film increased slightly even though it was found in a reasonably small range because the type of pipe had a diameter that was not too large. Then, it tends to stabilize at 0.345 cm. This proves that, in general, the slug flow thickness is difficult to know verifiable in the configurations tested in this research.

It can be seen in Figure 15 that the speed is slightly less linear between the addition of the flow rate and the process of measuring the flow rate using the ERT-U_R method. This is because ERT has a sensitivity that is not as good as U_R but has the potential to measure the average heterogeneous flow regime signal, where there is a slight pattern following the structure of the film thickness. This happens because the flow rate and void fraction of the fluid flow rate in heterogeneous conditions strongly influence slug flow. In detail, the concepts of fraction void and flow velocity will be presented in future research.

CONCLUSION

This study uses a combination of U_R and ERT methods to detect the presence of slug flow or two different phases, where the configuration uses a 1.5-inch diameter pipe. The concept of measuring the thickness of a homogeneous block is adopted in the U_R method to measure the thickness of slug flow. With the help of a transducer and FPGA, the location and thickness of the slug flow can be detected consistently. In addition, the U_R method can monitor the appearance of homogenous and heterogeneous differences in slug flow conditions, making this method optimal for use in two-phase states. Following are some summaries obtained:

- The slug flow measurement technique is adopted from the block-homogeneous measurement process with error less than 1%.
- It can be stated that the bright yellow color contains the water fraction while the blue water fraction is in the ERT results, but due to spatial resolution problems, ERT is not good at imaging in detail the shape of the slug flow
- A high-speed camera is used to record the shape of the slug flow through images definitively, and the results of the ERT imaging technique are used to compare and detect the presence of different particles in the slug flow.
- If observed and compared with the reference (Zhai *et al.* 2021), in this study, the form of slug flow observed is a low-aerated slug flow type, where the main profile contains a slug surrounded by a small distribution amount of bubble flow.
- From the results of the combination of the ERT-U_R method, it was found that ERT has a sensitivity that is not as good as U_R but has the potential to measure the average heterogeneous flow regime signal, where a slight pattern can be seen following the structure of the film thickness. This happens because the flow rate, velocity, and void fraction of the fluid flow rate in heterogeneous conditions strongly influence slug flow.
- A new approach is introduced to determine liquid film thickness by utilizing the instantaneous velocity profile. The liquid film thickness can be reliably measured by identifying the size of slug flow.
- The FPGA system developed is capable of delivering significant results due to the accuracy and reliability of the system's design.
- In general, the main results show that ERT is used to depict the fundamental structure of the velocity profile, while U_R is applied to measure the velocity and provide detailed measurements of the slug flow film thickness.

Finally, this study presents a new perspective method to detect multi-scale slug flow structures, where the speed and thickness of slug flow can influence each other or vice versa. Besides, seeing the consistency of slug flow can also be the basis for modelling and developing flow transition models. Moreover, the U_R method is the most suitable method for determining the thickness of slug flow. It is more economical because it only requires one type of transducer, making it very prospective for use in slug flow conditions. Additionally, by knowing the liquid film thickness of the slug, the void fraction development process better becomes where the composition of different phases can be learned such that it can be implemented across various industrial sectors, including the upstream oil and gas industry, pharmacy industry, freshwater distribution or freshwater management, metering systems, and custody transfer for volumetric systems.

Acknowledgments

This is work funded in part by Center Education Services (Pusat Layanan Pendidikan) Ministry of Education, Culture, Research and Technology of the Republic of Indonesia, and also LPDP under Beasiswa Kemendibudristek Grant 202209091514.

Availability of data and material

The data collected in this study are available from the corresponding author upon reasonable request.

Conflicts of interest

The authors declare that there is no conflict of interest regarding the publication of this paper.

Ethical standard

The authors have no relevant financial or non-financial interests to disclose.

LITERATURE CITED

Al-Safran, E., 2009 Investigation and prediction of slug frequency in gas/liquid horizontal pipe flow. Journal of Petroleum Science and Engineering **69**: 143–155.

Bao, Y., C. Tan, and F. Dong, 2022 Oil–water two-phase flow volume fraction measurement based on nonlinear ultrasound technique. IEEE Transactions on Instrumentation and Measurement **71**: 1–9.

Brinckerhoff, M., 2018 Comparison of electrical impedance tomography reconstruction algorithms with eidors reconstruction software.

Demidenko, E., A. Borsic, Y. Wan, R. J. Halter, and A. Hartov, 2011 Statistical estimation of eit electrode contact impedance using a magic toeplitz matrix. IEEE Transactions on Biomedical Engineering **58**: 2194–2201.

Deng, X., F. Dong, L. J. Xu, X. P. Liu, and L. A. Xu, 2001 The design of a dual-plane ert system for cross correlation measurement of bubbly gas/liquid pipe flow. Measurement Science and Technology **12**: 1024–1031.

Falcone, G., G. F. Hewitt, and C. Alimonti, 2010 *Multiphase flow metering*. Elsevier.

Graham, B. M., 2007 Enhancements in electrical impedance tomography (eit) image reconstruction for three-dimensional lung imaging.

Hitomi, J., S. Nomura, Y. Murai, G. De Cesare, Y. Tasaka, *et al.*, 2021 Measurement of the inner structure of turbidity currents by ultrasound velocity profiling. International Journal of Multiphase Flow **136**: 103540.

Lin, P. Y. and T. J. Hanratty, 1987 Detection of slug flow from pressure measurements. International Journal of Multiphase Flow **13**: 13–21.

Liu, B., B. Yang, C. Xu, J. Xia, M. Dai, *et al.*, 2018a pyeit: A python based framework for electrical impedance tomography. SoftwareX **7**: 304–308.

Liu, W., C. Tan, X. Dong, F. Dong, and Y. Murai, 2018b Dispersed oil–water two-phase flow measurement based on pulse-wave ultrasonic doppler coupled with electrical sensors. IEEE Transactions on Instrumentation and Measurement **67**: 2129–2142.

Nnabuife, S., B. Kuang, J. Whidborne, and Z. Rana, 2020 Non-intrusive classification of gas-liquid flow regimes in an s-shaped pipeline riser using a doppler ultrasonic sensor and deep neural networks. Chemical Engineering Journal **403**: 126401.

Nnabuife, S., K. E. Pilario, L. Lao, Y. Cao, and M. Shafiee, 2019 Identification of gas-liquid flow regimes using a non-intrusive doppler ultrasonic sensor and virtual flow regime maps. Flow Measurement and Instrumentation **68**.

Obayashi, H., Y. Tasaka, S. Kon, and Y. Takeda, 2008 Velocity vector profile measurement using multiple ultrasonic transducers. Flow Measurement and Instrumentation **19**: 189–195.

Ruan, T., 2016 Development of an automated impedance tomography system and its implementation in cementitious materials.

Shimomoto, Y., K. Ikeda, H. Ogawa, Y. Nakatsu, and I. Yamamoto, 2021 Detection of slug flow generated in horizontal pipeline. Sensors and Materials **33**: 947.

Somersalo, E., M. Cheney, and D. Isaacson, 1992 Existence and uniqueness for electrode models for electric current computed tomography. SIAM Journal on Applied Mathematics **52**: 1023–1040.

Tiwari, N. and Y. Murai, 2021 Ultrasonic velocity profiler applied to explore viscosity–pressure fields and their coupling in inelastic shear-thinning vortex streets. Experiments in Fluids **62**: 185.

Villarreal, J., D. Laverde, and C. Fuentes, 2006 Carbon-steel corrosion in multiphase slug flow and co2. Corrosion Science **48**: 2363–2379.

Wang, S.-Q., K.-W. Xu, and H.-B. Kim, 2019 Slug flow identification using ultrasound doppler velocimetry. International Journal of Heat and Mass Transfer **148**: 119004.

Weiling, L., C. Tan, and F. Dong, 2021 Doppler spectrum analysis and flow pattern identification of oil-water two-phase flow using dual-modality sensor. Flow Measurement and Instrumentation **77**: 101861.

Zhai, L., H. Xia, H. Xie, and J. Yang, 2021 Structure detection of horizontal gas–liquid slug flow using ultrasonic transducer and conductance sensor. IEEE Transactions on Instrumentation and Measurement **70**: 1–10.

Zhai, L., B. Xu, H. Xia, and N. Jin, 2023 Simultaneous measurement of velocity profile and liquid film thickness in horizontal gas–liquid slug flow by using ultrasonic doppler method. Chinese Journal of Chemical Engineering **58**: 323–340.

Zhang, H., F. Dong, and C. Tan, 2022 Liquid–solid two-phase flow rate measurement by electrical and ultrasound doppler sensors. IEEE Transactions on Instrumentation and Measurement **71**: 1–9.

How to cite this article: Wiranata, L.F., Putra N.K., and Kurniadi D. Detection of Two-Phase Slug Flow Film Thickness by Ultrasonic Reflection *Chaos Theory and Applications*, 6(4), 237-248, 2024.

CHAOS
Theory and Applications
in Applied Sciences and Engineering

e-ISSN: 2687-4539
RESEARCH ARTICLE
Vol.6 / No.4 / 2024 / pp.249-256
https://doi.org/10.51537/chaos.1466919

Performance Evaluation of FPGA-Based Design of Modified Chua Oscillator

Muhammed Furkan Taşdemir[*β,1], **Anna Litvinenko**[α,2], **İsmail Koyuncu**[β,3] **and Filips Capligins**[α,4]

[*]SpacESPro Lab – Space Electronics and Signal Processing Laboratory, Riga Technical University, Riga, Latvia, [α]Institute of Radioelectronics, Riga Technical University, Riga, Latvia, [β]Department of Electrical Electronics Engineering, Faculty of Technology, Afyon Kocatepe University, Afyon, Türkiye.

ABSTRACT The chaotic systems are among the most important areas that have increased in popularity and are actively used in several fields. One of the most essential components in chaotic systems is the chaotic oscillator which generates chaotic signals. IQ-Math and floating point number systems are preferred number standards. In this study, the Modified Chua chaotic oscillator has been designed to work on FPGA chips using fixed point and floating point number representations, and both system version performances are compared. Euler numeric algorithm has been used to design the Modified Chua chaotic oscillator. In the first section of the study, the Modified Chua chaotic system based on fixed point has been composed the model in the Matlab Simulink and converted to VHDL with the help of Matlab HDL Coder Toolbox. In the second section of the study, the Modified Chua chaotic oscillator has been designed with VHDL based on floating point. Modified Chua chaotic oscillators which are composed with two different number standards have been tested using Xilinx ISE Design Tools in VHDL. Modified Chua chaotic oscillators which have two different number standards and designed, are synthesized for Virtex-6 on ML605 FPGA development board using Xilinx ISE Design Tools 14.2 program. The values that are achieved from the process of synthesizing and the process of maximum operating frequency have been presented. As a result, the study has found that fixed-point representation achieved a maximum operating frequency of 50.242 MHz, while the floating-point representation achieved 273.631 MHz.

KEYWORDS
Chaotic oscillators
Euler algorithm
FPGA chips
VHDL

INTRODUCTION

Nonlinear systems are structures where space nonlinearity and linearity exist only within certain limits. Behaviors that seem simple or trivial in a nonlinear system can lead to unpredictable changes and results. Chaotic systems are among the most actively used and researched areas within this broad spectrum. In recent years, chaotic systems have played an important role in solving increasing security problems worldwide. Economic and technological developments have led to an increase in the speed and capacity of information, which has, in turn, caused security problems. While massive amounts of information are transmitted without loss, each piece of information needs to be stored and encrypted according to the area in which it is used. In this context, the use of chaotic systems and the renewed focus on these systems created by chaotic fluctuations present promising approaches for solving these problems. Chaotic systems do not work as periodical systems, so the next results cannot be predicted (Litvinenko 2017a). Chaotic systems produce new values at each step, giving different results from previous values and generating unique subsequent values from earlier values. Chaotic systems are actively used in such areas as cybersecurity (Amir Anees 2018), voice and image processing (Fatih Özkaynak 2013; Mohamed L. Barakat 2013; Gabr 2023), optimization algorithms (Erkan Tanyıldızı 2017), the defense industry (Vasyuta *et al.* 2019), biomedical applications (Zhengxing Huang 2014), and mechatronics (Jorge Pomares 2014).

A study conducted by Linsheng Zhang et al. developed an automated system that transitions from a floating-point number system to a fixed-point number system based on Extreme Value Theory (Linsheng Zhang 2009). Babajans et al. achieved synchronization between two chaotic systems using the Vilnius chaotic oscillator, with results shared by researchers (Ruslans Babajans 2020). Litvinenko's research thesis illustrates the usage of generated chaotic systems such as logistic map, Bernoulli map, and tent map for DS-CDMA (Direct-Sequence Code Division Multiple Access) systems (Litvinenko 2017b). In another study, Litvinenko et al. proposed Chaos Shift Keying (CSK) based on the Modified

Manuscript received: 10 May 2024,
Revised: 11 October 2024,
Accepted: 31 October 2024.

[1]furkantasdemir123@gmail.com (**Corresponding author**).
[2]anna.litvinenko@rtu.lv
[3]ismailkoyuncu@aku.edu.tr
[4]filips.capligins@rtu.lv

Chua chaotic system, achieving synchronization between a transmitter and two receivers for data transmission (Litvinenko and A. 2019). Additional work by M. K. Gabr implemented an encryption and decryption system that employs the Chua circuit as one of the chaos generators (Gabr 2023). Arpaci's thesis introduced video encryption and decryption systems using a Modified Chua chaotic circuit for enhanced data transmission security (Arpaci 2019).

The daily pace of humanity has increased with the development of technology. When comparing the last decade's technological acceleration with that of the last 20 years, even though the time span is only twice as long, technological development has more than doubled. One important technology for meeting ever-growing humanity's needs, whether in academia or industry, is the FPGA (Field Programmable Gate Array). Nowadays, the active use of FPGA chips is crucial for digital system design. FPGA chips, which allow fast prototyping, parallel signal processing, and high working frequency capabilities in a whole-circuit structure, are preferred in resourceful signal processing tasks, such as chaotic system applications. FPGA chips have been extensively used in signal processing and real-time applications that require high performance with high processing power due to their significant advantages such as pipelined processing, low cost, re-programmability, and high throughput. FPGA chips are actively used in both academia and industry in fuzzy logic (Fatih Karataş 2020), image processing (Koyuncu 2022; Taşdemir 2020), biomedical systems (Fatih Karataş 2021), artificial neural networks (Murat Alcin 2016), and communication systems (Filips Capligins 2021).

FPGA, ASIC (Application Specific Integrated Circuit), and DSP (Digital Signal Processor) chips are preferred structures due to their high-frequency parallelism and special capabilities. These chips are generally differentiated from each other based on their production purpose and cost. While FPGA chips stand out with their re-programmability and flexibility, ASICs are more cost-effective than FPGA and DSP chips when produced in large quantities, but they do not have re-programmability features. As a result, they are industrially preferred over other chips. DSP chips excel in signal processing for more specialized areas such as image and sound processing. In one study, a stable modified fourth-order autonomous Chua chaotic system was developed using the Virtex 6 FPGA chip, achieving a frequency of 180.180 MHz with the RK4 (Fourth-order Runge-Kutta) numerical integration algorithm based on 32-bit IQ-Math floating-point numbers (Fei Yu 2020). In another study, Capligins et al. programmed an FPGA chip using a Modified Chua chaos generator for high-security networks and wireless communication methods (Filips Capligins 2021).

In this study, unlike classical number base studies, the performance analysis of two number bases has been performed. To ensure fair conditions and make the performance analysis more efficient, the comparison was made with a single-type chaotic system produced with two different number bases. A Modified Chua chaotic system was generated with both fixed-point and floating-point number bases. This approach reduces the impact of external factors when analyzing performance and equalizes conditions between both representations. This study provides an analysis of the importance of the materials to be used when designing a chaotic oscillator and how these materials should be selected, such as memory, experience, and cost requirements. Conversely, it provides guidance on how to produce the most effective chaotic oscillator with readily available materials.

The chip statistics, including comparison results with different studies related to Xilinx (AMD) and Altera (Intel) FPGA chips, are given in Table 1. This table includes critical parameters, such as maximum operating frequencies and numerical methods regarding fixed-point and floating-point number systems based on Lu-Chen, Lorenz, Liu, Chen, Chua; the Modified Chua used in this article, and CO and HO from the literature.

As a result of the study, the fixed point number-based chaotic oscillator provides less memory usage than the results that shared in the literature, while the floating point number-based oscillator provides clearer and more accurate results. These results explain the importance of chaotic oscillators considering the importance of past and present communication systems. In the second part of the study, general information about fixed and floating point number systems, Modified Chua chaotic oscillator and FPGA-based Modified Chua chaotic system is given. In the Third Section, 32-bit fixed point and floating point-based Modified Chua chaotic oscillator unit designs on FPGA chip and chip statistics obtained from the designs are presented. In the last section, the results obtained from the studies are evaluated.

MATERIAL AND METHODS

In this section, general information is given about technical structures in which used in the system at FPGA chip, chaotic systems, fixed and floating point number systems.

Fixed and Floating Number Systems and Modified Chua Chaotic Oscillator

The whole universe is moved with frequency domain. In this direction frequency and signal processing methods form the basis of each electronically systems. Digital signal processing methods move with two essential number bases as fixed point bases and floating point bases. These number systems thought storing methods and transmission methods of computer-based data. The floating point number system represents the large scale according to a fixed point. This situation not only represents to maximum and minimum range of number value but also the width of the range of decimal values is also represented. While floating point number system using more resources on the chips, the fixed point number system was seen more economical.

The mathematical function of the Modified Chua chaotic system is represented by the system of differential equations (1), where p_1, p_2, p_3, p_4 are the four state variables and σ, γ, θ, c, d are the system coefficients.

$$\begin{aligned}
\frac{dp_1}{dt} &= -g(p_1, p_3)(p_1 - p_3) - p_2, \\
\frac{dp_2}{dt} &= p_1 + \gamma p_2, \\
\frac{dp_3}{dt} &= \theta\left(g(p_1, p_3)(p_1 - p_3) - p_4\right), \\
\frac{dp_4}{dt} &= \sigma p_3.
\end{aligned} \tag{1}$$

$g(p_1, p_3)$ is a nonlinear function with the parameters c and d, as defined in Equation (2).

$$g(p_1, p_3) = c(p_1 - p_3 - d)(p_1 - p_3) > d(p_1 - p_3) \leq d \tag{2}$$

The Forward Euler method is one of the most important numerical analysis techniques. It is commonly used for the numerical analysis of differential equations, which are employed to model changing variables in dynamic systems, such as chaotic systems. The essential principle of the Forward Euler method involves updating system variables regularly. This step-by-step process predicts changes in the system variables between discrete time steps.

Table 1 Floating and fixed point-based chaotic oscillators on FPGA in literature

Study	Chaotic Generator	Max. Operating Frequency (MHz)	Numerical Method	Platform	Number System
Fei Yu (2020)	Modified Chua	180.180	RK4	Xilinx ZYNQ-XC7Z020	32-bit Floating Point
Murat Tuna (2019)	Lu-Chen	464.688	Heun	Xilinx Virtex-6	32-bit Floating Point
Mohamed Salah Azzaz (2013)	Lorenz	38.86	Euler	Xilinx Virtex-2	32-bit Floating Point
Mohammed F. Tolba (2017)	Liu	137.561	Fractional order	Xilinx Virtex-5	Fixed Point
Sadoudi Said (2009)	Chen	22.850	RK4	Xilinx Virtex-2	32-bit Floating Point
Luciana De Micco (2011)	Lorenz	3.676 - 125	RK4	Altera Cyclone-III	32-bit Floating Point
Akif Akgul (2015)	Lorenz	373.134	RK4	Xilinx ISE and Labview Simulator	32-bit Floating Point
E. Tlelo-Cuautle (2015)	Chua	70.943, 69.500, 58.648, 48.209	RK4 / Forward Euler	Xilinx Spartan-3	Fixed Point
Omar Guillén-Fernández (2021)	CO1, CO2, CO3, HO4, HO5	90.88, 102.75, 58.55, 79.77, 82.7	Forward Euler	Altera Cyclone-IV	Fixed Point
Present	Modified Chua	273.631 - 50.242	Forward Euler	Xilinx Virtex-6	Floating Point and Fixed Point

The Forward Euler method can be expressed with a simple equation:

$$y_{n+1} = y_n + h \cdot f(t_n, y_n) \tag{3}$$

Different initial conditions and system parameter values can be used for the Modified Chua system to exhibit the desired nonlinear dynamic behavior. The initial conditions and parameters used in this study for the implementation of the Modified Chua chaotic system are provided in Table 2.

Table 2 Initial Conditions and System Parameters of the Modified Chua Chaotic System

Initial Conditions				Parameters				
p_1	p_2	p_3	p_4	σ	γ	θ	c	d
0.1415	-2.073	-0.252	0.829	0.5	10	1.5	3	1

The Modified Chua chaotic system has been modeled numerically using the Euler algorithm. Figure 1 shows the phase portraits of the system state variables. Figure 2 shows the time series of the system state variables. For the mathematical calculations of the Modified Chua chaotic oscillator, the step size of the algorithm is chosen as $h = 0.005$. For $k = 0$, the initial values are set to $P_1(k) = 0.1415$, $P_2(k) = -2.073$, $P_3(k) = -0.252$, and $P_4(k) = 0.829$. The system parameters are considered as $\sigma = 0.5$, $\gamma = 10$, and $\theta = 1.5$.

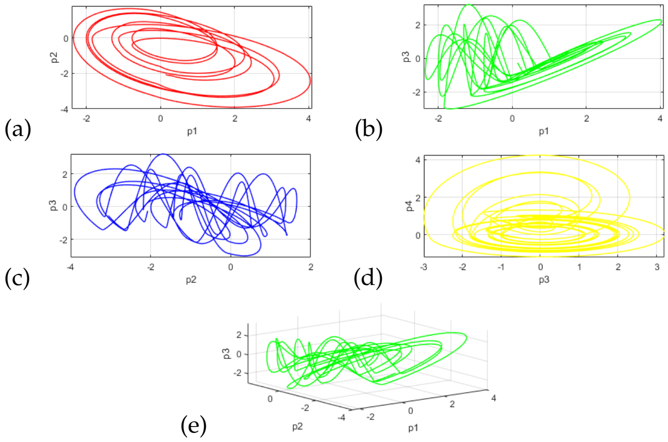


Figure 1 The phase portraits of the Modified Chua chaotic system state variables: (a) p_1-p_2, (b) p_1-p_3, (c) p_2-p_3, (d) p_3-p_4, (e) p_1-p_2-p_3.

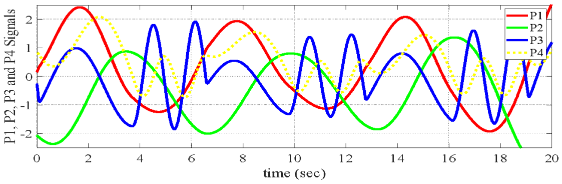


Figure 2 The time series of the Modified Chua chaotic system state variables for P_1, P_2, P_3, and P_4.

FPGA-Based Modified Chua Chaotic System

FPGA chips are widely used for low power consumption, reprogrammability, fast prototyping, and parallel signal processing capabilities in academic and industrial fields. FPGA chips are designed with languages such as VHDL, Verilog, and System C. Compared to specially designed graphics cards, supercomputers, and parallel computers, these chips are low-cost and easier to access. Chaotic systems are one of the very sensitive areas in digital signal processing. This sensitivity and the high energy consumption caused by it can be mitigated with the parallel processing and low power consumption capabilities of FPGA chips.

Figure 3 shows the parts of a basic FPGA chip. A basic FPGA chip consists of three main components: Configurable Logic Block (CLB), Input-Output Blocks (IOBs), and interconnections. According to the digital circuit designed by the user, the logic blocks and the connections between them are configured. The CLB provides functional elements for the logic circuit that the user wants to create. The IOB provides an interface between the internal signal lines of the chip and the pins of the chip. Interconnections are used for configuring the connections between the CLB and IOB.

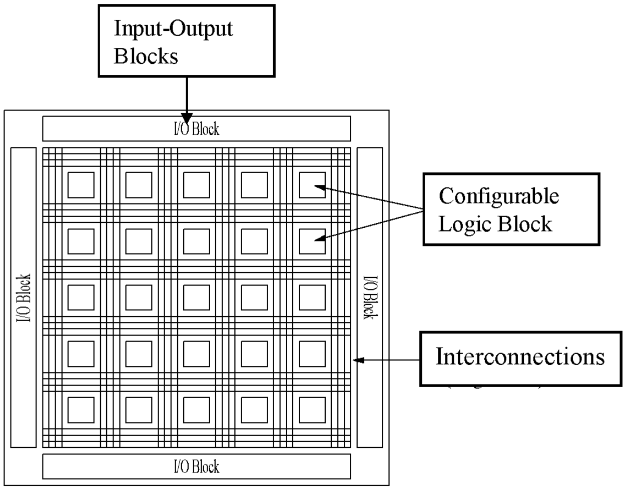


Figure 3 The parts of a basic FPGA chip: Configurable Logic Block (CLB), Input-Output Blocks (IOBs), and interconnections.

Chaotic systems exhibit complex behavior without any particular order or predictability. They are dynamical systems that can be expressed in terms of nonlinear equations and can respond to small changes over time. One of the most important things is that tiny changes at initial conditions can have enormous effects over time. Because of these effects, chaotic systems need not only memory space but also parallelism to calculate fast on generating new values. Therefore FPGA chips are generally preferred for chaotic systems. The Chua circuit is an electrical circuit using passive elements such as resistors, capacitors, and active elements such as diodes, transistors, and amplifiers. This circuit is a system that can exhibit chaotic behavior under certain conditions.

FPGA-BASED CHAOTIC SYSTEMS

In this section, the structure of the Modified Chua chaotic oscillator is described using both floating-point and fixed-point bases. The 32-bit IEEE-754-1985 single-precision standard is used for floating-point number representation, while the 16I-16Q format is used for IQ-Math with fixed-point numbers.

Fixed Point Based Modified Chua Chaotic Oscillator Unit on FPGA

Fixed-point implementation for fractional calculus in FPGA is the most straightforward approach since it uses a predefined number of bits for all of the signal nets. From the perspective of FPGA, the system works with binary numbers without knowing about fraction points; it is implemented in the initial design stage and therefore does not require any additional resources for calculation processing, unlike floating-point format. In the present work, 32 bits are used, from which 1 bit is reserved for the sign, 15 bits for representing integer values, and 16 bits for representing fractional values. This way, such a format can represent numbers from $-2^{15} = -32768$ to $(2^{15} - 2^{-16}) = 32767.99998474121$ with precision equal to $2^{-16} = 1.52587890625 \times 10^{-5}$. The integer part has a large reserve for further processing needs, while the fractional part, which limits the absolute maximum number of chaos discrete values, provides acceptably high precision values for the obtained chaos to exhibit itself.

To implement the continuous chaotic system in a digital system, the ordinary differential equations need to be solved using some discrete integration method. In this work, the forward Euler method is used for fixed-point Modified Chua chaos generator implementation, with a time step of 1/1024. Other methods (such as 4th order Runge-Kutta) may provide more accurate results in some cases but are more complex to implement and require more hardware resources. Since with the chosen integration time step and method, the solutions of the chaos system's differential equations always converge, this approach is feasible for practical use in experimental studies. The Matlab Simulink model of the fixed-point based Modified Chua oscillator is shown in Figure 4.

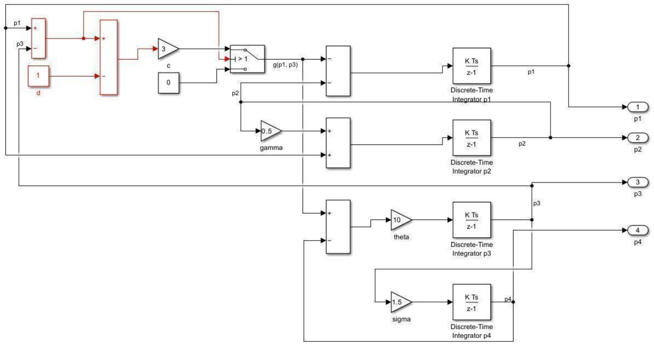

p1
p3
+
-
1
d
+
-
3
c
0
| > 1
g(p1, p3)
-
-
p2
0.5
gamma
+
+
K Ts
z-1
Discrete-Time Integrator p1
p1
Discrete-Time Integrator p2
p2
1
p1
2
p2
3
p3
p3
+
-
10
theta
Discrete-Time Integrator p3
4
p4
1.5
sigma
Discrete-Time Integrator p4
p4

Figure 4 Matlab Simulink model of the fixed-point based Modified Chua chaotic oscillator.

The Modified Chua chaotic oscillator has been designed in Matlab Simulink, and the design has been verified to work as expected. After this step, the VHDL code was generated using the Matlab HDL Coder Toolbox. In this study, the fixed-point number standard was used for converting the model in Matlab Simulink into VHDL language. A testbench was designed and tested using Xilinx ISE Design Tools to verify the design. The test results of the fixed-point based Modified Chua chaotic oscillator are shown in Figure 5.

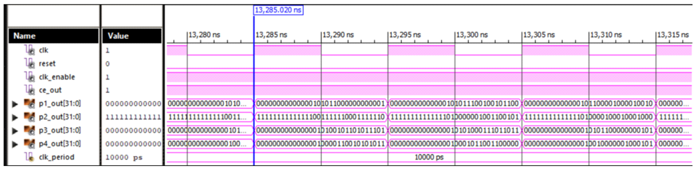


Figure 5 Modified Chua chaotic oscillator test results based on fixed point obtained from Xilinx ISE Design Tools.

Figure 6 shows the top-level block diagram of the fixed-point implementation of the Modified Chua chaotic oscillator unit. This unit has three inputs: a 1-bit *clk*, a 1-bit *clk_enable*, and a 1-bit *reset*. It also has five outputs: 32-bit *P1_out*, *P2_out*, *P3_out*, *P4_out*, and a 1-bit *ce_out*. When the *clk* signal has a rising edge, the system is activated and generates new values. The chaotic signal values are output through *P1_out*, *P2_out*, *P3_out*, and *P4_out*, and the system's functionality is verified by monitoring the output of the *ce_out* signal. The schematic for the second-level blocks is not presented due to its complexity and overly detailed nature.

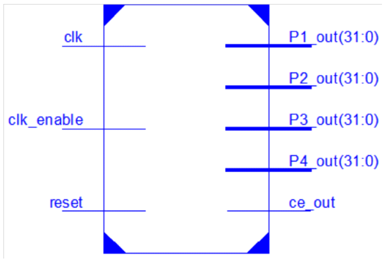


Figure 6 Top-level block diagram of the fixed-point based Modified Chua chaotic oscillator unit on FPGA.

The VIRTEX-6 ML605 FPGA evaluation board (Device: XC6VLX240T, Package: FF1156, Speed: -1) was used for the implementation. The maximum operating frequency for the design was 50.242 MHz. FPGA utilization statistics and the maximum frequency for the fixed-point implementation of the Modified Chua chaotic oscillator unit are presented in Table 3.

A testbench was written in VHDL language to analyze the results generated by the fixed-point based Modified Chua chaotic oscillator for the P_1, P_2, P_3, and P_4 signals. The design was run in the Xilinx ISE Design Suite, and 4,000 values for each signal were saved in a .txt file. The related values were converted to real values to generate the time series graph. The time series of the Modified Chua chaotic oscillator for P_1, P_2, P_3, and P_4 signals are shown in Figure 7.

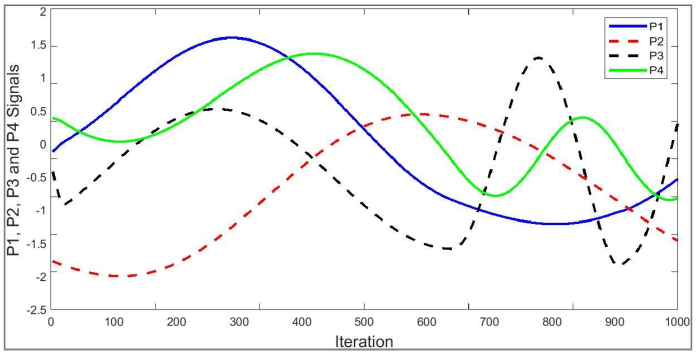


Figure 7 Time series of the fixed-point based Modified Chua chaotic system for P_1, P_2, P_3, and P_4 signals on FPGA.

Floating Point Based Modified Chua Chaotic Oscillator Unit on FPGA

In this section, the discretized model of the Modified Chua chaotic system is obtained using the Euler numerical differential equation solution method. Next, the related chaotic system is designed using VHDL and Xilinx ISE Design Tools. After the design process, the chaotic system is tested with a testbench written in VHDL and using Xilinx ISE Design Tools to verify the Floating Point based Modified Chua Chaotic Oscillator unit. The chip statistics for the Floating Point based Modified Chua Chaotic Oscillator unit are presented. The top-level block diagram of the floating-point based Modified Chua Chaotic Oscillator Unit on FPGA is shown in Figure 8.

The top-level block diagram includes a 1-bit *clk*, *Reset*, and *Start* signals as inputs, and 32-bit *P1_out*, *P2_out*, *P3_out*, *P4_out*, and a 1-bit *Result_Ready* signal as outputs.

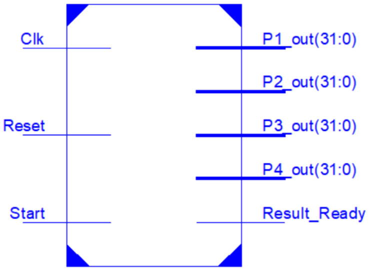


Figure 8 Top-level block diagram of the floating-point based Modified Chua Chaotic Oscillator Unit on FPGA.

Figure 9 shows the second-level block diagram of the floating-point based *Modified_Chua_Euler* unit. This includes *x4mux* and *Modified_Chua_Euler* units in the second-level design. The *x4mux* unit has inputs grouped as 32-bit 2X4. The first four groups represent the initial conditions required for starting operation. When the *Modified_Chua_Euler* unit begins generating results, the second four groups send all data either to the system output as *P1_out*, *P2_out*, *P3_out*, *P4_out* or to the *x4mux* unit as the next initial conditions. The *shys* signal is used as a control process. When the value of *shys* is '0', the *x4mux* unit sends initial conditions to the *Modified_Chua_Euler* unit. If the value of *shys* is '1', the *x4mux* unit sends the generated values to the *Modified_Chua_Euler* unit.

The third-level block schematic for the floating-point based *Modified Chua Chaotic Oscillator* unit is shown in Figure 10. The design includes IP-Core units, a *modified_Chua_core* unit, and a filter unit for multiplication and addition operations in 32-bit floating-point standards. The *Modified_Chua_core* unit and the Chua chaotic system are implemented in VHDL for FPGA realization. The filter unit is created to prevent unintended consequences of the *Chua_Chaotic_Oscillator* unit.

Table 3 FPGA utilization statistics of the fixed-point based Modified Chua Chaotic Oscillator Unit

	Number of Slice Registers	Number of Slice LUTs	Number of Occupied Slices	Number of DSP48E1s	Number of IOBs	Maximum Clock Frequency (MHz)
Used	128	642	158	8	132	
Utilization (%)	1	1	1	1	22	50.242
Available	301,440	150,720	37,680	768	600	

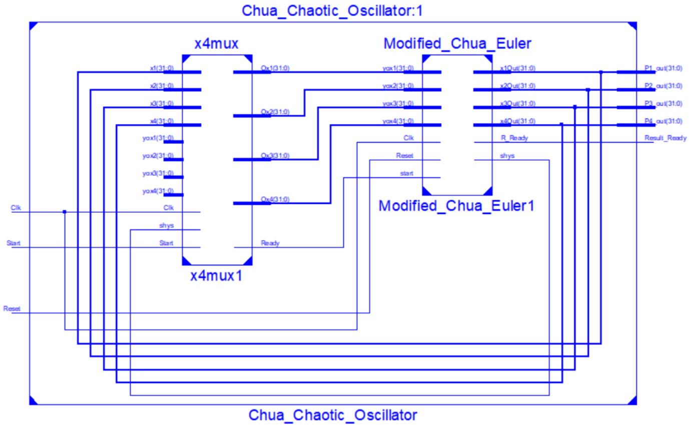


Figure 9 Second-level block diagram of the floating-point based Modified Chua Chaotic Oscillator Unit on FPGA.

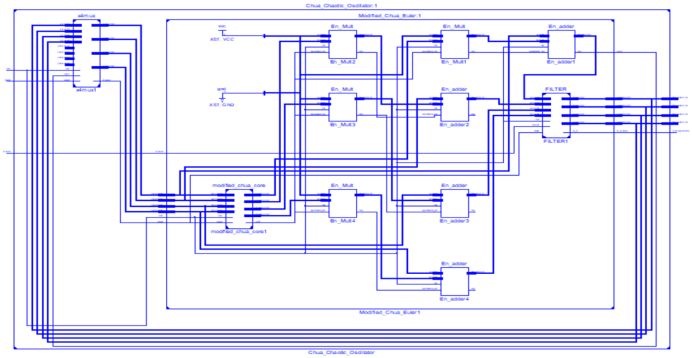
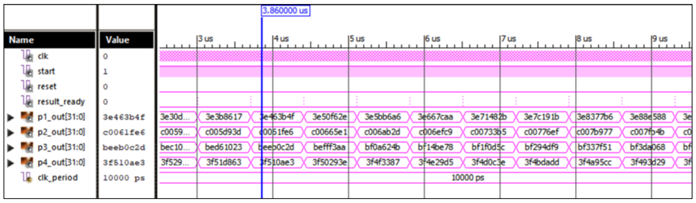


Figure 10 Third-level block diagram of the floating-point based Modified Chua Chaotic Oscillator Unit on FPGA.

The simulation results of the floating-point based *Modified Chua Chaotic Oscillator* unit are shown in Figure 11. The designed unit sends signals to the outputs (*P1_out*, *P2_out*, *P3_out*, and *P4_out*) on every 69th clock pulse. While the *Chua_Chaotic_Oscillator* unit is generating results, the *Result_Ready* signal is set to ’1’; otherwise, it remains ’0’.



Name	Value
clk	0
start	1
reset	0
result_ready	0
p1_out[31:0]	3e463b4f
p2_out[31:0]	c0061fe6
p3_out[31:0]	beeb0c2d
p4_out[31:0]	3f510ae3
clk_period	10000 ps

Figure 11 Simulation results of the floating-point based Modified Chua Chaotic Oscillator unit on FPGA.

FPGA utilization statistics and the maximum frequency have been presented for the realization of the floating-point based *Modified Chua Chaotic Oscillator* unit in FPGA. The VIRTEX-6 ML605 (Device: XC6VLX240T, Package: FF1156, Speed: -1) evaluation board was used for realization, achieving a maximum frequency of 273.631 MHz. The results are shown in Table 4.

A testbench was written in VHDL language to analyze the results generated by the floating-point based Modified Chua chaotic oscillator for the P_1, P_2, P_3, and P_4 signals. The design was run in the Xilinx ISE Design Suite, and 4,000 values for each signal were saved in a .txt file. The related values were converted to real values to generate the time series graph. The time series of the Modified Chua chaotic oscillator for P_1, P_2, P_3, and P_4 signals are shown in Figure 12.

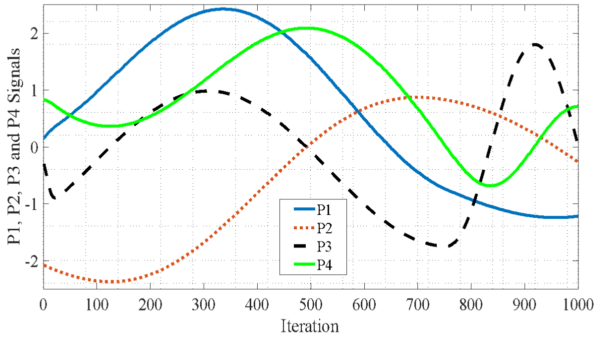


Figure 12 Time series of the floating-point based Modified Chua chaotic oscillator for P_1, P_2, P_3, and P_4 signals on FPGA.

Table 4 FPGA utilization statistics of the floating-point based Modified Chua Chaotic Oscillator Unit

	Number of Slice Registers	Number of Slice LUTs	Number of Occupied Slices	Number of DSP48E1s	Number of IOBs	Maximum Clock Frequency (MHz)
Used	6,827	6,395	2,325	9	132	
Utilization (%)	2	4	6	1	22	273.631
Available	301,440	150,720	37,680	768	600	

CONCLUSION

In this study, the Modified Chua chaotic system has been designed using the 32-bit IQ-Math number system and the 32-bit IEEE-754-1985 standard for implementation on FPGA chips. The Modified Chua chaotic oscillator based on the floating-point number system was designed using VHDL, while the fixed-point version was modeled in Matlab Simulink and then converted to VHDL using the Matlab HDL Coder Toolbox. Both designs of the Modified Chua chaotic oscillators were tested using a testbench composed in VHDL, and the successful simulation results were presented in this study.

The two designs of the Modified Chua chaotic oscillators were synthesized with the Xilinx ISE Design Tools 14.2 for the Virtex-6 chip on the ML605 FPGA development board. The fixed-point based Modified Chua chaotic system achieved a maximum frequency of 50.242 MHz, while the floating-point based version achieved a maximum frequency of 273.631 MHz. Although the floating-point design provided a higher maximum frequency, the fixed-point design was more favorable in terms of resource utilization.

With the findings presented in this study, a high-speed random number generator can be developed using the floating-point based Chua chaotic oscillator. Furthermore, a low-cost, customized FPGA-based chaotic random number generator can be implemented using the fixed-point based Chua chaotic oscillator. It has been demonstrated that the Modified Chua chaotic oscillators based on both floating and fixed-point number systems can be used safely in secure communication and cryptographic applications. In future studies, a random number generator may be developed for secure communication and cryptographic applications using FPGA-based Modified Chua chaotic oscillators designed with both number systems.

Availability of data and material

Not applicable.

Conflicts of interest

The authors declare that there is no conflict of interest regarding the publication of this paper.

Ethical standard

The authors have no relevant financial or non-financial interests to disclose.

LITERATURE CITED

Akif Akgul, H. C., 2015 Chaos-based engineering applications with a 3d chaotic system without equilibrium points. Nonlinear Dynamics .

Amir Anees, I. H., 2018 A novel method to identify initial values of chaotic maps in cyber security. MDPI .

Arpaci, B., 2019 *Color Image Encryption System Designed with The Modified Chua Circuit For Security Data Transmission*. Ph.D. thesis, Gazi University, Ankara.

E. Tlelo-Cuautle, J. R.-M.-A.-R.-P., 2015 Fpga realization of multi-scroll chaotic oscillators. Communications in Nonlinear Science and Numerical Simulation **27**: 66–80.

Erkan Tanyıldızı, T. C., 2017 Kaotik haritalı balina optimizasyon algoritmaları. Science and Engineering Journal of Fırat University pp. 307–317.

Fatih Karataş, K., 2020 Implementation of fuzzy logic membership functions on fpga. Bilgisayar Bilimleri ve Teknolojileri Dergisi pp. 1–9.

Fatih Karataş, K., 2021 Design and implementation of fpga-based arrhythmic ecg signals using vhdl for biomedical calibration applications. International Advanced Researches and Engineering Journal pp. 362–371.

Fatih Özkaynak, A. B., 2013 Security analysis of an image encryption algorithm based on chaos and dna encoding. IEEE .

Fei Yu, H. S., 2020 Ccii and fpga realization: A multistable modified fourth-order autonomous chua's chaotic system with coexisting multiple attractors. Complexity, Dynamics, Control, and Applications of Nonlinear Systems with Multistability pp. 1–17.

Filips Capligins, A. L., 2021 Fpga implementation and study of synchronization of modified chua's circuit-based chaotic oscillator for high-speed secure communications. In *IEEE 8th Workshop on Advances in Information, Electronic And Electrical Engineering*.

Gabr, M. K., 2023 *Image Encryption Based On Base-N PRNGs Key Application And Parallel Base-N S-Boxes*. The German University, Cairo.

Jorge Pomares, I. P., 2014 Dynamic visual servoing with chaos control for redundant robots. IEEE/ASME Transactions On Mechatronics pp. 423–430.

Koyuncu, T. M. F.-A.-M.-T.-M. C. E., I., 2022 Real time realization of image processing algorithms on fpga. Journal of BAUN Institute of Science and Technology **24**: 125–137.

Linsheng Zhang, Y. Z., 2009 Floating-point to fixed-point transformation using extreme value theory. In *IEEE/ACIS International Conference on Computer and Information Science*, pp. 271–276.

Litvinenko, A., 2017a *Use of Chaotic Sequences For Data Transmission System*. Riga Technical University.

Litvinenko, A., 2017b *Use of Chaotic Sequences for Data Transmission Systems*. Ph.D. thesis, Riga Technical University, Faculty of Electronics and Telecommunications, Institute of Radioelectronics.

Litvinenko, A. and A. A., 2019 Advanced chaos shift keying based on a modified chua's circuit. In *IEEE Microwave Theory And Techniques In Wireless Communications (MTTW)*, pp. 17–22.

Luciana De Micco, H. A., 2011 Fpga implementation of a chaotic oscillator using rk4 method. In *Southern Conference On Programmable Logic*, pp. 185–190.

Mohamed L. Barakat, A. S., 2013 Hardware stream cipher with controllable chaos generator for colour image encryption. IETL .

Mohamed Salah Azzaz, C. T., 2013 A new auto-switched chaotic system and its fpga implementation. Communications in Nonlinear Science and Numerical Simulation pp. 1792–1804.

Mohammed F. Tolba, A. M., 2017 Fpga implementation of two fractional orders chaotic systems. AEU-International Journal of Electronics and Communications **78**: 162–172.

Murat Alcin, I. P., 2016 Hardware design and implementation of a novel ann-based chaotic generator in fpga. Elsevier pp. 37–58.

Murat Tuna, M. A., 2019 High speed fpga-based chaotic oscillator design. Microprocessors and Microsystems **73–80**.

Omar Guillén-Fernández, M. F.-L.-C., 2021 Issues on applying one- and multi-step numerical methods to chaotic oscillators for fpga implementation. Mathematics .

Ruslans Babajans, D. D., 2020 Noise immunity of substitution method-based chaos synchronization in vilnius oscillator. IEEE Microwave Th and Techniq in Wireless Communications .

Sadoudi Said, M. S., 2009 An fpga real-time implementation of the chen's chaotic system for securing chaotic communications. International Journal of Nonlinear Science pp. 467–474.

Taşdemir, I. K., M. F., 2020 Real-time fast corner detection algorithm based image processing application on fpga. In *International Asian Congress on Contemporary Sciences-III*, pp. 39–46, Konya.

Vasyuta, K., F. Zots, and I. Zakharchenko, 2019 Building the air defense covert information and measuring system based on orthogonal chaotic signals. Peer-Reviewed Open Access Scientific Journal pp. 33–43.

Zhengxing Huang, W. D., 2014 Similarity measure between patient traces for clinical pathway analysis: Problem, method, and applications. IEEE Journal of Biomedical and Health Informatics pp. 4–14.

How to cite this article: Taşdemir, M. F., Litvinenko, A., Koyuncu, I., and Capligins, F. Performance Evaluation of FPGA-Based Design of Modified Chua Oscillator. *Chaos Theory and Applications*, 6(4), 249-256, 2024.

CHAOS
Theory and Applications
in Applied Sciences and Engineering

e-ISSN: 2687-4539
RESEARCH ARTICLE
Vol.6 / No.4 / 2024 / pp.257-263
https://doi.org/10.51537/chaos.1552511

Hybrid Function Projective Synchronization of Hyperchaotic Financial Systems via Adaptive Control

Vikash [*,1], **Ayub Khan** [α,2] **and Khursheed Alam** [*,3]
[*]School of Basic Sciences and Research, Sharda University, Greater Noida, Uttar Pradesh, 201306, India, [α]Department of Mathematics, Jamia Millia Islamia, New Delhi, India.

ABSTRACT In this manuscript, we establish hybrid function projective synchronization of a new hyperchaotic system using an adaptive control technique with unknown system parameters. In order to prevent either from deriving from participants in the single hyperchaotic financial system, identical master and slave systems are chosen. We design an adaptive controller to achieve global chaos synchronization between these master and slave systems. The synchronization results are based on adaptive control theory and Lyapunov stability theory. Additionally, we outline the basic dynamic characteristics of both hyperchaotic financial systems. Numerical simulations performed in Matlab validate our results excellently.

KEYWORDS
Fundamental dynamical properties
Adaptive control
Synchronization
Chaotic systems
Lyapunov stability theory

INTRODUCTION

In nonlinear science, chaotic dynamics is an interesting area of research that has been a lot of exploration in current decades. Chaotic events have an impact on a wide range of domains, including secure communication, computer science, quantum physics, biological systems, chemical systems, power converters, electrical engineering, psychology, and so on (Chen and G.ed. 1999) .Complex dynamics with unique characteristics, like topological mixing, dense periodic orbits, unusual attractors, broad Fourier transform spectra, limited and fractal motion qualities in phase space, and great sensitivity to beginning circumstances, are characteristics of a hyperchaotic system (Farivar,F. and Teshnehlab 2012). L. M. Pecora and Carroll (1990) established the master-slave idea for synchronization of chaotic systems in 1990.

Given the extensive practical applications of chaotic dynamical systems in the fields mentioned above, numerous theoretical and experimental studies have been conducted on controlling chaos and achieving synchronization (Abd-Elouahab and Wang 2010; Chen, L. and Wu 2011). To be more precise, synchronization of nonlinear dynamical systems allows for a deeper comprehension of collective dynamical behaviour in systems that are physical, chemical, biological, and other. Numerous mathematical, physical, sociological, physiological, and biological systems have been shown to exhibit synchronous behaviour (Koronovskii, A.A. and Hramov 2013).

Many techniques for controlling and synchronizing, have been designed comparable and non-identical chaotic systems have been developed in an effort to improve ways for chaos management and synchronization. Such methods include backstepping control (Li, S.Y. and Chiu 2012), adaptive control (Khan, A. and Shikha. 2017), linear feedback (Ma, M. and Cai 2012), optimal control (Li, Y. and Li 2013),(Cai, G. and Fang 2013) active control (Kareem, S.O. and Njah 2012), active sliding control (Khan, A. and Prasad 2016), passive control (Motallebzadeh, F. and Cherati 2012), and so on. To derive the controller in these published works, one has to be aware of the values of the system's parameters.

Nevertheless, these factors are frequently unknown in real-world scenarios. Subsequently deriving an adaptive controller, therefore, is an important problem for the control and synchronization of hyperchaotic financial systems with unknown system parameters (Vaidyanathan, S. 2015). For the purpose of synchronizing hyperchaotic financial systems, a number of synchronization techniques have been developed such as complete synchronization (CS) (Chen, H. and Guo 2021), generalized synchronization (GS) (Zheng, Z. and Hu 2000), projective synchronization (PS), and hybrid synchronization (HS) (Wu, X. and Li 2012). Due to its ability to achieve speedier communication with its proportional features There are now two positive Lyapunov exponents (LE) that point to hyperchaotic behaviour. This dissipative hyperchaotic system's mathematical characteristics are shown both theoretically and statistically, including Lyapunov exponents (Al-Azzawi, S and Hasan 2024).

Manuscript received: 18 September 2024,
Revised: 14 October 2024,
Accepted: 21 October 2024.

[1] vbvikasrohilla@gmail.com
[2] akhan12@jmi.ac.in
[3] khursheed.alam@sharda.ac.in (**Corresponding author**).

In order to have a thorough understanding of the dynamics of the suggested system, we have examined Hamilton energy and competitive modes at various parameter values. One of the simplest concepts for gaining a deeper knowledge of the dynamics or stability of a chaotic system is Hamilton energy. Research reveals that if one can totally control the energy flow in a chaotic system, one can successfully manage its stability. By handling the Hamilton energy, they can do this. The significance of the Hamilton energy for the formation of nonlinear oscillations has been examined in recent research on chaotic systems (Khan, A. and ALi 2024) projective synchronization has garnered substantial attention and has been thoroughly explored among the numerous types of chaos synchronization (Li., Z. and Xu 2004).

The synchronization of the master and slave systems up to a scaling factor is the defining feature of projective synchronization. Chen and Li have considered the function projective synchronization (FPS), a unique synchronization approach (Chen, Y. and Li 2007) . As opposed to projective synchronization, function projective synchronization (FPS) allows synchronization between the response and drive systems up to a certain scaling function rather than a constant. Projective synchronization (PS) or complete synchronization (CS), respectively, can be achieved by choosing the scaling function to be either a constant or unity. Thus, a more inclusive definition of projective synchronization is function projective synchronization. Function projective synchronization is very helpful for safe communications because of the unpredictable nature of the scaling function, which can further improve communication security.

In more general terms, though, not every element in the vector can synchronize to the required scaling function. All of the vector's scaling functions differ in hybrid function projective synchronization (HFPS) (Ojo, K.S.. and Omeike 2014) which increase complexity and fortifies secure communication even more. The primary benefit of employing the adaptive control technique is that it enables controllers to accomplish drive and response system synchronization without requiring knowledge of parameter values. This method effectively synchronizes the systems with less information needed. The active control technique is used to create synchronization and anti-synchronization between the drive and response systems. Controller design requires parameter values. These days, secure communication is a major concern. Hybrid function projective synchronization (HFPS), as previously mentioned, increases controller complexity and makes it more difficult for hackers to interpret communications. This combination enhances secure communication.

Additionally, most reported research on hybrid function projective synchronization achieve synchronization between two hyperchaotic financial systems that are both part of the unified hyperchaotic financial system. Inspired by the above discussion, in this work, we address the HFPS via adaptive control. Complete synchronization (CS), projective synchronization (PS), anti-synchronization (AS), and hybrid projective synchronization (HPS) are the subcases of hybrid function projective synchronization.

This manuscript organized as: The problem of statements for the hyperchaotic financial system's synchronization are covered in Section 2. In section 3 A description of the hyperchaotic financial system's basic dynamical features is given. Section 4 is succeeded by the hyperchaotic financial system's hybrid function projective synchronization (HFPS) via adaptive control. A numerical simulations and discussions Section 5. Finally, conclusion is delivered in Section 6.

PROBLEM STATEMENT FOR SYNCHRONIZATION OF CHAOTIC SYSTEM

Assume that a hyperchaotic financial system with a state vector is a driving system.
$X_m \in R^n$ and $P \in R^{n\times n}$ is system matrix given by

$$\dot{X_m} = PX_m + f\left(X_m\right) \tag{1}$$

Where $f\left(X_m\right) : R^n \to R^n$ is the system's nonlinear part. Another highly hyperchaotic financial system can be thought of as a slave system with a state vector. The system matrix $Y_s \in R^n$ and $Q \in R^{n\times n}$ with controller is provided by P

$$\dot{Y_s} = QY_s + g\left(Y_s\right) + \sigma\left(X_m, Y_s\right) \tag{2}$$

Where $g\left(Y_s\right) : R^n \to R^n$ is nonlinear part of the slave system and σ is the adaptive controller added in slave system for synchronization of the systems (1) and (2).

For hybrid function projective synchronization, the error $e \in R^n$ between states X_m and Y_s is defined as:

$$e = Y_s - A(t)X_m \tag{3}$$

Where $A(t) = \text{diag}\left(\eta_1(t), \eta_2(t), \ldots\ldots., \eta_n(t)\right)$ is the diagonal matrix and $\eta_i(t) : R^n \to R(i = 1, 2, \ldots\ldots, n)$ are functions that are bounded and continuously differentiable, $\eta_i(t) \neq 0\forall t$.

From (1) and (3) error dynamics as:

$$\dot{e} = QY_s + g\left(Y_s\right) + \sigma\left(X_m, Y_s\right) - PX_m - f\left(X_m\right) \tag{4}$$

Therefore, for hybrid function projective synchronization, the goal is to determine the controller $\sigma\left(X_m, Y_s\right)$, so that $\lim_{t\to\infty} \|e(t)\| = 0, \forall e \in R^n$.

Remark 1: If $A(t) = \text{diag}\left(\eta_1(t), \eta_2(t), \ldots\ldots\ldots, \eta_n(t)\right)$ where $\eta_i(t) \in R$ are constants, then hybrid function projective synchronization simplifies to hybrid projective synchronization. Furthermore, when all $\eta_i(t)$ are identical, the problem reduces to projective synchronization.

FUNDAMENTAL DYNAMICAL PROPERTIES OF THE SYSTEM

Consider the novel financial system:

$$\begin{aligned}
\dot{x_1} &= {x_3}^2 + \left(x_2 - a\right)x_1 + x_4\\
\dot{x_2} &= 1 - bx_2 - {x_1}^2\\
\dot{x_3} &= -x_1x_2 - cx_3\\
\dot{x_4} &= -0.05x_1{x_3}^2 - dx_4
\end{aligned} \tag{5}$$

Where the interest rate (x_1), investment demand (x_2), price index (x_3), and average profit margins (x_4) are the four state variables for which the system specifies the temporal evolution. Differentiation with respect to time t is indicated by the dot and $a \geq 0$ the saving amount, $b \geq 0$ the cost per investment, $c \geq 0$ is the elasticity, and $d \geq 0$ is positive systems parameter.

The values of the parameters $a = 0.9, b = 0.2, c = 1.5$ and $d = 0.17$ the Lyapunov exponents are $\lambda_1 = 1.1605, \lambda_2 = 0.6589, \lambda_3 = -0.7145$ and $\lambda_4 = -2.0642$ as shown in Fig 1.
$\sum_{\lambda=1}^{4} \lambda_i = -0.9593 \leq 0$. The considered system is hyperchaotic based on our calculation of the Lyapunov exponent for the system witnessing the two positive Lyapunov exponents. Here two Lyapunov exponents are positive and two are negative, positive Lyapunov exponents shows that system 5 is hyperchaotic.

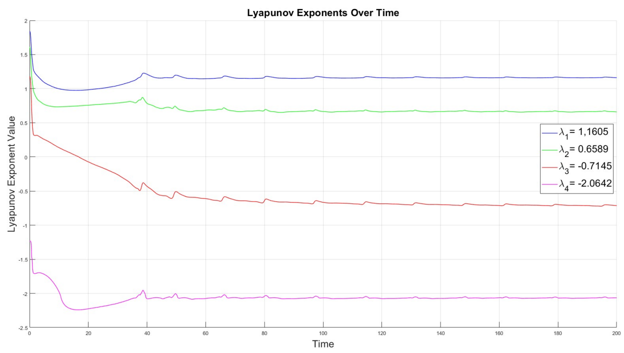


Figure 1 Lyapunov exponents of the 5 hyperchaotic financial system

Lyapunov exponent spectrum

In a dynamical system, the rate at which infinitesimally close paths separate is quantified by the Lyapunov exponent spectrum. The average exponential rate of divergence or convergence in particular directions inside the system's phase space is represented by each of the exponents that make up this representation. When neighbouring trajectories diverge, a positive Lyapunov exponent suggests chaotic activity; conversely, a negative one suggests convergence to a stable point or periodic orbit. Zero exponents frequently imply neutral stability, as in the case of a conservative system's trajectory. The complete range of Lyapunov exponents sheds light on the general stability of the system as well as the characteristics of its attractors.

Bifurcation analysis

Bifurcation analysis as parameter a increases from 0.5 to 2, the system transitions from stable behavior to hyperchaotic oscillations in the interest rate x. For values of parameter b ranging from 0.1 to 0.5, the system exhibits very large fluctuations in the steady state of x reaching magnitudes on the order 10^{12} suggesting instability in the system. As parameter c increases from 1 to 2, the system shows hperchaotic behavior initially, but the fluctuations in x reduce as c increases. Parameter d, ranging from 0.1 to 0.3, leads to hyperchaotic behaviour for the most of its range , with multiple steady-state value of x. The systems remains hyperchaotic.

Dissipation

The divergence of the system 5 is

$$\nabla V = \left(\frac{\partial \dot{x}}{\partial x}\right) + \left(\frac{\partial \dot{y}}{\partial y}\right) + \left(\frac{\partial \dot{z}}{\partial z}\right) + \left(\frac{\partial \dot{w}}{\partial w}\right)$$

$$= -a - b - c - d = -(a + b + c + d) < 0.$$

Since $a, b, c, d \geq 0$, the dynamical system (3.1) is a dissipative system, and

$$\dot{V}(t) = e^{-(a+b+c+d)}.$$

This indicates that as t increases, each volume carrying the trajectory of this dynamical system (3.1) shrinks to zero at an exponential rate of $-(a + b + c + d)$. Hence, the asymptotic motion settles onto an attractor of the new dynamical system (3.1), thereby limiting all of the orbits of the system to a certain subset with zero volume. (Wu, X. and Li 2012).

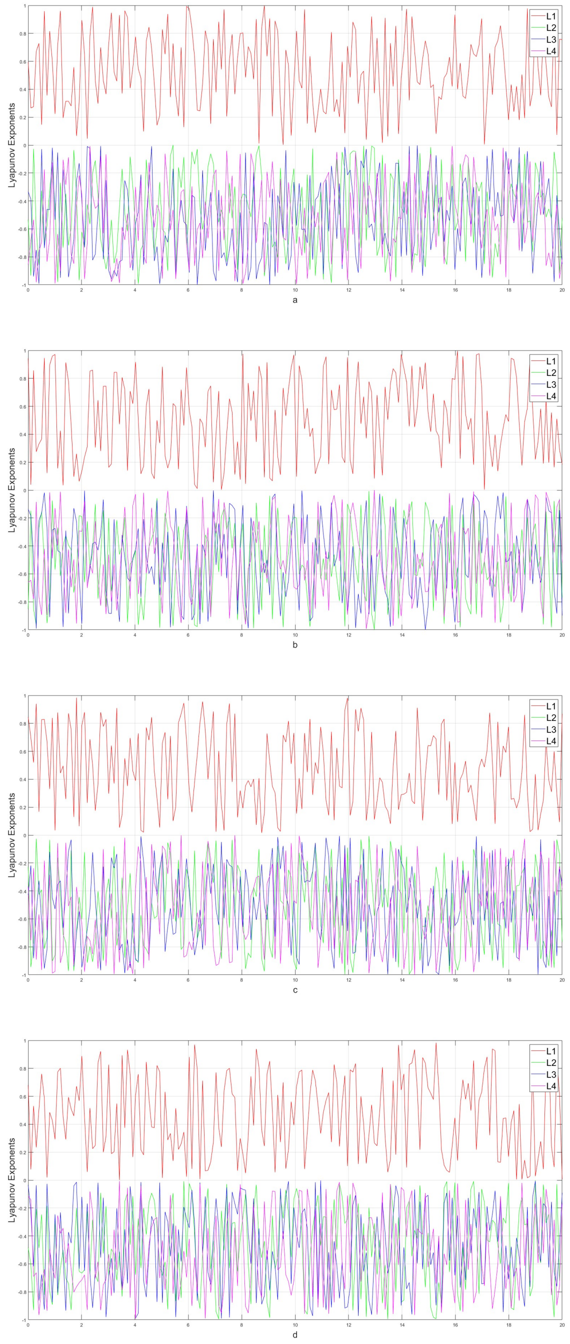


Figure 2 Lyapunov exponent spectrum of the new system 5 and L1 represented largest lyapunov exponent in all cases. (a) represented versus parameter a, (b) represented versus parameter b, (c) represented versus parameter c, (d) represented versus parameter d

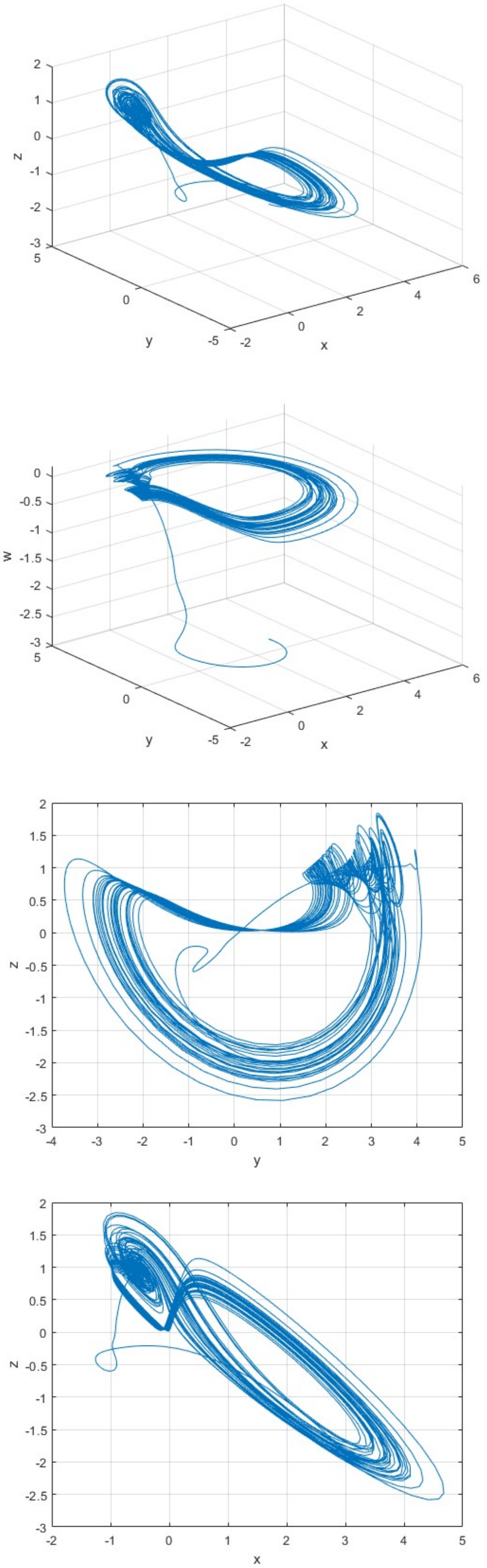


Figure 3 Phase portraits of 5 hyperchaotic financial system (a) in x-y-z space, (b) in the x-y-w space, (c) projection on the y-z plane, and (d) projection on x-z plane

Equilibrium and Stability

The equilibrium of system 5 can be obtained by solving equations:

$$x_3^2 + (x_2 - a)x_1 + x_4 = 0$$
$$1 - bx_2 - x_1^2 = 0$$
$$-x_1x_2 - cx_3 = 0$$
$$-0.05x_1x_3^2 - dx_4 = 0$$

system 5 has a trivial equilibrium points $E_0 = (0, 0, 0, 0)$ and two nontrivial equilibrium points $E_1 = (0.9, 0.7, -0.4, -0.05)$, and $E_2 = (2.6, 5, -9.06, -3.53)$. Therefore, E_0 is stable and E_1 and E_2 are unstable equilibrium points.

HYBRID FUNCTION PROJECTIVE SYNCHRONIZATION OF HYPERCHAOTIC FINANCIAL SYSTEM VIA ADAPTIVE CONTROL

Our aim is to achieve hybrid function projective synchronization between master and slave hyperchaotic systems using the method of adaptive control. Is that regard, we consider the master and slave system, follows:

$$\begin{aligned} \dot{x_1} &= {x_3}^2 + (x_2 - a)\, x_1 + x_4 \\ \dot{x_2} &= 1 - bx_2 - {x_1}^2 \\ \dot{x_3} &= -x_1x_2 - cx_3 \\ \dot{x_4} &= -0.05x_1{x_3}^2 - dx_4 \end{aligned} \tag{6}$$

Where x_1, x_2, x_3 and x_4 are typical profit margins, price index, investment demand, and interest rate. and a, b, c and d are positive parameters. The above system 6 has already been seen as hyperchaotic for the specified values of parameterise.

The slave system is described as:

$$\begin{aligned} \dot{y_1} &= y_3^2 + (y_2 - a)\, y_1 + y_4 + u_1 \\ \dot{y_2} &= 1 - by_2 - y_1^2 + u_2 \\ \dot{y_3} &= -y_1y_2 - cy_3 + u_3 \\ \dot{y_4} &= -0.05y_1y_3^2 - dy_4 + u_4 \end{aligned} \tag{7}$$

Where y_1, y_2, y_3 and y_4 are typical profit margins, price index, investment demand, and interest rate. and a, b, c and d are positive parameters and u_1, u_2, u_3 and u_4 , continuously differentiable, non-zero scaling functions. The error dynamics is expressed as the derivative of 8 is

$$e_i = y_i - \eta_i x_i, \quad \text{where } i = 1, 2, 3, 4 \tag{8}$$

and $\eta'{}_i s(i = 1, 2, 3, 4)$ are bounded, continuously differentiable, non-zero scaling functions. The error states' time derivative of 8 is

$$\dot{e_l} = \dot{y_i} - \dot{\eta_l}(t)x_i - \eta_i(t)\dot{x_l} \tag{9}$$

Using 6 ,7 and 9 we obtain

$$\begin{aligned} \dot{e_1} &= y_3^2 + (y_2 - a)\, y_1 + y_4 + u_1 - \dot{\eta_1}(t)x_1 - \eta_1(t)\left(x_3^2 + (x_2 - a)\, x_1 + x_4\right) \\ \dot{e_2} &= 1 - by_2 - y_1^2 + u_2 - \dot{\eta_2}(t)x_2 - \eta_2(t)\left(1 - bx_2 - x_1^2\right) \\ \dot{e_3} &= -y_1y_2 - cy_3 + u_3 - \dot{\eta_3}(t)x_3 - \eta_3(t)\left(-x_1x_2 - cx_3\right) \\ \dot{e_4} &= -0.05y_1y_3^2 - dy_4 + u_4 - \dot{\eta_4}(t)x_4 - \eta_4(t)\left(-0.05x_1x_3^2 - dx_4\right) \end{aligned} \tag{10}$$

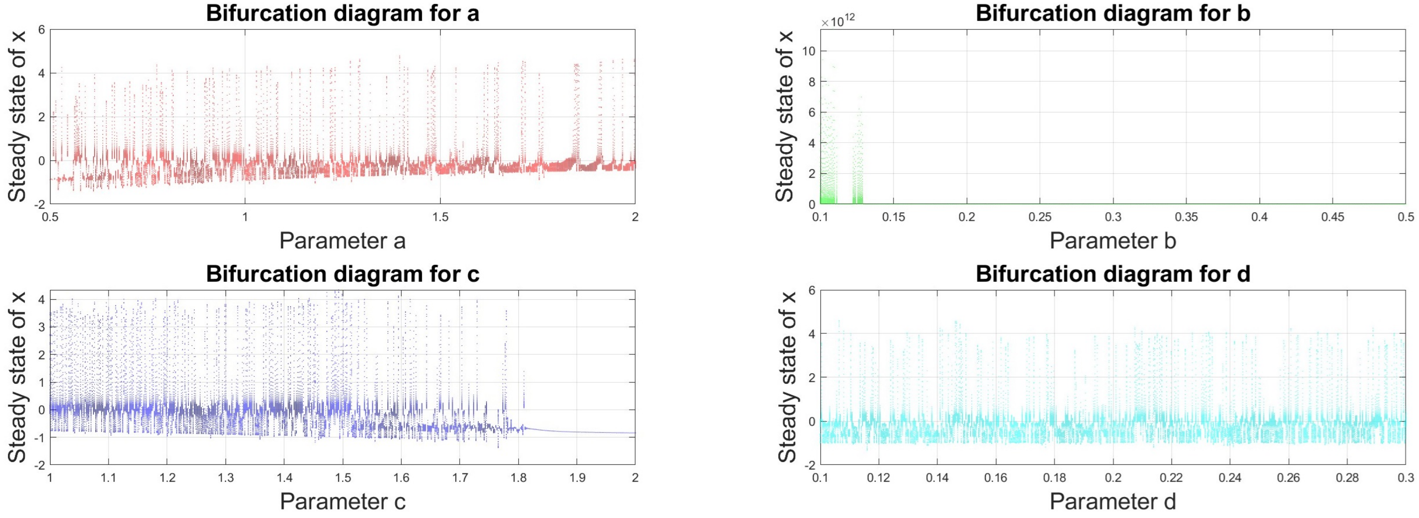


Figure 4 Bifurcation for Parameter a, Chaotic Behaviour with Increasing Parameter a . Bifurcation for Parameter b, Extreme Oscillations and Instability with Parameter b. Bifurcation for Parameter c, Transition to Stability with Increasing Parameter c. Bifurcation for Parameter d, Chaotic Dynamics with Varying Parameter d

To achieve HFPS between master and slave hyperchaotic financial systems with uncertain parameters and arbitrary initial conditions, we need to design appropriate controllers $u_i(t)(i = 1, 2, 3, 4)$ and a parameter update rule. In this way, the unknown parameters will be estimated simultaneously with the assurance that the error dynamical system 10 is asymptotically stable at the origin, and HFPS between the slave system 7 and the master system 6 will be achieved. As a result, for the error dynamical system 10, the synchronization problem is transformed into a stability challenge.
The controller are designed as follows:

$$u_1 = -{y_3}^2 - y_2y_1 + \dot{\eta}_1(t)x_1 + \eta_1(t)x_1 + \eta_1(t)x_2x_1 + \hat{a}e_1 + \eta_1(t)x_4 - k_1e_1$$
$$u_2 = -1 + {y_1}^2 + \dot{\eta}_2(t)x_2 + \eta_2(t) - n_2(t){x_1}^2 + \hat{b}e_2 - k_2e_2 \quad (11)$$
$$u_3 = y_1y_2 + \dot{\eta}_3(t)x_3 - \eta_3(t)x_1x_2 + \hat{c}e_3 - k_3e_3$$
$$u_4 = 0.05y_1{y_3}^2 + \dot{\eta}_4(t)x_4 - \eta_4(t)0.05x_1{x_3}^2 + \hat{d}e_4 - k_4e_4$$

and the following is how the parameter updating rules are made:

$$\begin{aligned} \dot{\hat{a}} &= -\left(y_1 - \eta_1(t)x_1\right)e_1 - k_5e_a \\ \dot{\hat{b}} &= -\left(y_2 - \eta_2(t)x_2\right)e_2 - k_6e_b \\ \dot{\hat{c}} &= -\left(y_3 - \eta_3(t)x_3\right)e_3 - k_7e_c \\ \dot{\hat{d}} &= -\left(y_4 - \eta_4(t)x_4\right)e_4 - k_8e_d \end{aligned} \quad (12)$$

Where the control gain $k_i > 0(i = 1, 2, \ldots, 8), \hat{a}, \hat{b}, \hat{c}$, and $\hat{d}$ are the parameters for the estimated variable that are unknown. $e_a = \hat{a} - a, e_b = \hat{b} - b, e_c = \hat{c} - c$, and $e_d = \hat{d} - d$ are corresponding parameter errors. We select the Lyapunov function that satisfies the requirements of Lyapunov stability theory for the parameter update methods designed above. That will demonstrate the stability of the faulty dynamical system and the achievement of the necessary synchronization. However, we select the subsequent Lyapunov function candidate for the error system 10:

$$V(t) = \frac{1}{2}\left({e_1}^2 + {e_2}^2 + {e_3}^2 + {e_4}^2 + {e_a}^2 + {e_b}^2 + {e_c}^2 + {e_d}^2\right) \quad (13)$$

Undoubtedly, $V(t) > 0$. Along the trajectories of the error system 10, the time derivative of V(t) equals

$$\dot{V}(t) = e_1\dot{e_1} + e_2\dot{e}_2 + e_3\dot{e_3} + e_4\dot{e_4} + e_a\dot{e_a} + e_b\dot{e_b} + e_c\dot{e}_c + e_d\dot{e_d}$$

$$\begin{aligned} \dot{V}(t) = e_1 \Big(y_3^2 + (y_2 - a)\,y_1 + y_4 + u_1 - \dot{\eta}_1(t)x_1 - \eta_1(t)\Big(x_3^2 + \\ (x_2 - a)\,x_1 + x_4)\Big) + e_2\Big(1 - by_2 - {y_1}^2 + u_2 - \dot{\eta}_2(t)x_2 - \\ \eta_2(t)\left(1 - bx_2 - {x_1}^2\right)\Big) + e_3\left(-y_1y_2 - cy_3 + u_3 - \dot{\eta}_3(t)x_3 - \right. \\ \eta_3(t)\left(-x_1x_2 - cx_3\right)) + e_4\Big(-0.05y_1{y_3}^2 - dy_4 + u_4 - \dot{\eta}_4(t)x_4 - \\ \eta_4(t)\left(-0.05x_1{x_3}^2 - dx_4\right)\Big) + e_a\dot{e_a} + e_b\dot{e_b} + e_c\dot{e_c} + e_d\dot{e_d} \end{aligned} \quad (14)$$

Using 11, 12 and 14, we obtain

$$\begin{aligned} \dot{V}(t) = e_1\left(e_a\ \left(y_1 - \eta_1(t)x_1\right)\right) - k_1{e_1}^2 + e_2\left(e_b\left(y_2 - \eta_2(t)x_2\right)\right) - k_2{e_2}^2 \\ + e_3\left(e_c\left(y_3 - \eta_3(t)x_3\right)\right) - k_3{e_3}^2 + e_4\left(e_d\left(y_4 - \eta_4(t)x_4\right)\right) - k_4{e_4}^2 \\ + e_a\left(-\left(y_1 - \eta_1(t)x_1\right)e_1 - k_5e_a\right) + e_b\left(-\left(y_2 - \eta_2(t)x_2\right)e_2 - k_6e_b\right) \\ + e_c\left(-\left(y_3 - \eta_3(t)x_3\right)e_3 - k_7e_c\right) + e_d\left(-\left(y_4 - \eta_4(t)x_4\right)e_4 - k_8e_d\right) \\ = -k_1{e_1}^2 - k_2{e_2}^2 - k_3{e_3}^2 - k_4{e_4}^2 - k_5{e_a}^2 - k_6{e_b}^2 - k_7{e_c}^2 - k_8{e_d}^2 \\ = eKe < 0 \end{aligned}$$

Where $e = (e_1, e_2, e_3, e_4, e_a, e_b, e_c, e_d)$ and $K = diag\left(k_1, k_2, k_3, k_4, k_5, k_6, k_7, k_8\right)$.
Based on the Lyapunov stability theory, the error vector e asymptotically converges to zero, meaning that $\lim_{t\to\infty} \|e(t)\| = 0$. since $\dot{V}(t) < 0$ It also suggests that the unknown parameters are approximated simultaneously the hybrid function projective synchronization of the master and slave systems is both globally and asymptotically synchronized.

NUMERICAL SIMULATIONS

To demonstrate the viability and validity of the proposed synchronization technique, numerical simulations are carried out. The selected parameter of the hyperchaotic financial system as $a = 0.9, b = 0.2, c = 1.5$, and $d = 0.17$. The initial condition of master and slave system are chosen as $x(0) = (3, 1, -2, -3)$ and $y(0) = (5, 3, -6, -3)$. The scaling functions $\eta_1 = \sin(t), \eta_2 = 0.5\cos(t), \eta_3 = 1 + \sin(t)$, and $\eta_4 = \cos(0.1t)$ are chosen at random. It is assumed that the control gains are $k_i = 0.11 \forall i = 1, 2, \ldots, 8$. and Figs. 5 and 6 display the outcomes of the simulation. Figure 4 illustrates how the error dynamics approach zero as t approaches infinity. As seen in Figure3 exhibit that values of the unknown parameters also tend to $\hat{a} \to a, \hat{b} \to b, \hat{c} \to c, \hat{d} \to d$ Consequently, the intended hybrid function projective synchronization between the slave and master systems is achieved.

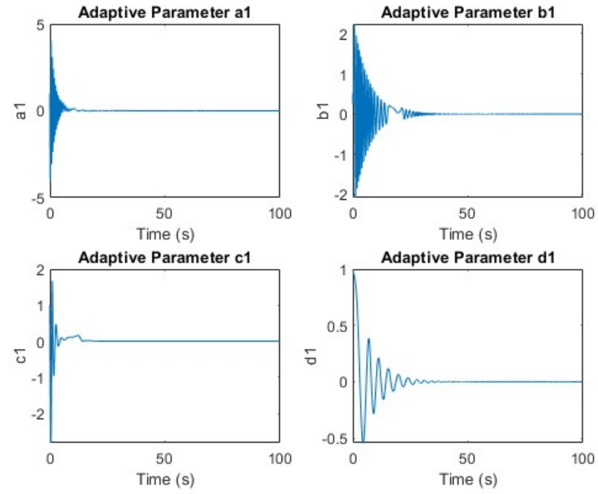


Figure 5 The estimated value of the parameters that are unknown $\hat{a}, \hat{b}, \hat{c}$ and $\hat{d}$ as hybrid function projection synchronization occurs

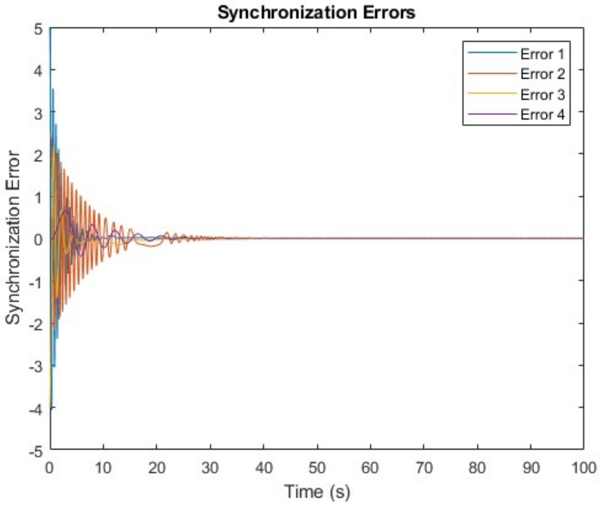


Figure 6 Error in synchronization between the slave and master system states

CONCLUSION

By including more terms and increasing one more variable, average profit margins x_4, based on the chaotic system described in (Wu, X. and Li 2012) is achieved. Both theoretical and numerical analyses are performed on a few fundamental dynamical features, including the Lyapunov exponent spectrum, bifurcations, equilibria, and hyperchaotic dynamical behaviours. This manuscript successfully demonstrates hybrid function projective synchronization (HFPS) of a novel hyperchaotic system utilizing an adaptive control technique, even in the presence of unknown system parameters. The chosen master and slave systems are carefully selected to ensure that they are distinct from any members of the unified chaotic financial system. An adaptive controller is meticulously designed to ensure global chaos synchronization between the master and slave systems. The synchronization is carefully proven using Lyapunov stability theory and adaptive control theory, ensuring theoretical soundness. Additionally, the manuscript provides a detailed analysis of the fundamental dynamical properties of the hyperchaotic financial systems. The effectiveness and accuracy of the proposed synchronization strategy are further confirmed through numerical simulations conducted in MATLAB, validating the theoretical findings and demonstrating practical applicability.

Acknowledgments

This work was supported by Sharda University Scientific Research Projects Unit under Grants 0000-00-00-001, 0000-00-00-002.

Availability of data and material

Not applicable.

Conflicts of interest

The authors declare that there is no conflict of interest regarding the publication of this paper.

Ethical standard

The authors have no relevant financial or non-financial interests to disclose.

LITERATURE CITED

Chen, H., L. W. Y., Yu and M. Guo, 2021 Synchronization of a hyperchaotic finance system. Complexityl p. 6618435.

Khan, A. and R. Prasad, 2016 Hybrid synchronization of hyperchaotic CAI systems via sliding mode control. Journal of Engineering Thermophysics **25(1)**: 151–157.

Khan, A. and Shikha., 2017 Hybrid function projective synchronization of chaotic systems via adaptive control. International Journal of Dynamics and Control **5**: 1114–1121.

Motallebzadeh, F., M., Motlagh and Z. Cherati, 2012 Synchronization of different order chaotic systems: Adaptive active vs. optimal control. Communications in Nonlinear Science and Numerical Simulation **17(9)**: 3643–3657.

Vaidyanathan, S., 2015 Anti-synchronization of Brusselator chemical reaction systems via adaptive control. International Journal of ChemTech Research **8(6)**: 759–768.

Abd-Elouahab, H. N. E., M.S. and J. Wang, 2010 Chaos control of a fractionalorder financial system. Mathematical Problems in Engineering **2010**: 270646.

Al-Azzawi, S and A. Hasan, 2024 A new 4D hidden hyperchaotic system with higherlargest Lyapunov exponent and its synchronization. International Journal of Mathematics Statistics and Computer Sciencel **2**: 63–74.

Cai, G., L. H. P., Yao and X. Fang, 2013 Adaptive full state hybrid function projective synchronization of financial hyperchaotic systems with uncertain parameters. Discrete and Continuous Dynamical Systems-Series B **18(8)**: 2019–2028.

Chen and G.ed., 1999 Controlling chaos and bifurcation in engineering systems. CRC Press .

Chen, L., Y., Chai and R. Wu, 2011 Control and synchronization of fractional-order financial system based on linear control. Discrete Dynamics in Nature and Society **958393**: 958393.

Chen, Y. and X. Li, 2007 Function projective synchronization between two identical chaotic systems. International journal of modern physics C **18(05)**: 883–888.

Farivar,F., M. N. M., Aliyari Shoorehdeli and M. Teshnehlab, 2012 Chaos control and modified projective synchronization of unknown heavy symmetric chaotic gyroscope systems via Gaussian radial basis adaptive backstepping control. Nonllnear Dynamics **64**: 1913–1941.

Kareem, S.O., K., Ojo and A. Njah, 2012 Function projective synchronization of identical and nonidentical modified finance and Shimizu Morioka systems. Pramana **79**: 71–79.

Khan, A. and S. ALi, 2024 Hamilton energy, competitive modes and ultimate bound estimation of a new3D chaotic system, and its application in chaos synchronization. Physica Scripta **99(11)**: 115205.

Koronovskii, A.A., O. S. S., Moskalenko and A. Hramov, 2013 Generalized synchronization in discrete maps. New point of view on weak and strong synchronization. Chaos, Solitons and Fractals **46**: 12–18.

L. M. Pecora and T. L. Carroll, 1990 A new class of chaotic circuit. Pysical review lette **266**: 821–824.

Li, S.Y., C. L. C. K. L., Yang and T. Chiu, 2012 Adaptive synchronization of chaotic systems with unknown parameters via new backstepping strate. Nonlinear Dynamics **70**: 2129–2143.

Li, Y., S., Tong and T. Li, 2013 Adaptive fuzzy output feedback control for asinglelink flexible robot manipulator driven DC motor via backstepping. Nonlinear Analysis: Real World Applications **14(1)**: 483–494.

Li., Z. and D. Xu, 2004 A secure communication scheme using projective chaos synchronization. Chaos, Solitons and Fractals **22(2)**: 477–481.

Ma, M., J., Zhou and J. Cai, 2012 Synchronization of different order chaotic systems: Adaptive active vs. optimal controll. International Journal of Modern Physics C **23(11)**: 1250073.

Ojo, K.S.., A. O. O., Njah and M. Omeike, 2014 Generalized reducedorder hybrid combination synchronization of three Josephson junctions via backstepping technique. Nonlinear Dynamics **77**: 583–595.

Vaidyanathan, S. and A. T. Azar, 2016 Adaptive control and synchronization of Halvorsen circulant chaotic systemsl. In Advances in chaos theory and intelligent control **337**: 225–247.

Wu, X. and S. Li, 2012 Dynamics analysis and hybrid function projective synchronization of a new chaotic system. Nonlinear Dynamics **69**: 1979–1994.

Zheng, Z. and G. Hu, 2000 Generalized synchronization versus phase synchronization. Physical Review E **62(6)**: 7882.

How to cite this article: Vikash, Khan A., and Alam K. Hybrid Function Projective Synchronization of Hyperchaotic Financial Systems via Adaptive Control *Chaos Theory and Applications*, 6(4), 257-263, 2024.

CHAOS
Theory and Applications
in Applied Sciences and Engineering

e-ISSN: 2687-4539
RESEARCH ARTICLE
Vol.6 / No.4 / 2024 / pp.264-272
https://doi.org/10.51537/chaos.1486869

Tax Audit in Turkiye: Simulation and Estimations Based on Kernel and Weight Functions

Mehmet Niyazi Çankaya[*,1] **and Murat Aydın**[α,2]
[*]Faculty of Applied Sciences, Department of International Trading and Finance, Uşak University, Uşak, Turkiye, [α]Faculty of Applied Sciences, Department of Accounting Finance and Management, Uşak University, Uşak, Turkiye.

ABSTRACT

This research examines the use of kernel estimation and `FindDistribution` methods in `Mathematica` software to analyze the ratio of taxpayer audits to total taxpayers, focusing on two large populations: one with approximately 80,000 audits per 100,000 taxpayers and the other with 4.5 million audits per 6 million taxpayers. Comparing the maximum statistics, the study shows that a larger number of taxpayers leads to more audits. The dataset also includes a weighted average for audits and taxpayers with a maximum of around 75,000 and 4 million respectively. These numerical values have been determined using the simulation carried out after modeling the real data sets of the total number of taxpayers and their audits from the years 2012 to 2023. These results show that different taxpayer populations require the targeted audit strategies and highlight the importance of the statistical models with corresponding estimation method to better understand complex distributions and improve tax audit processes.

KEYWORDS
Inference
Non-parametric models
Robust statistics
Simulation
Taxpayers

INTRODUCTION

Kernel density estimation (KDE) has been widely used in various fields, including income distribution analysis (Papatheodorou *et al.* 2004), poverty assessment (Minoiu and Reddy 2008), and population variance estimation (Hanif and Shahzad 2019). However, its application to grouped data has been found to introduce biases in poverty estimates (Minoiu and Reddy 2008). To address this, a method that combines auxiliary information with a kernel estimate has been proposed (Kuk 1993). Furthermore, a bipartite recursive algorithm based on KDE has been developed for measuring the scale of a given income population (Chen and Wang 2011).

KDE stands as a versatile tool widely deployed across diverse fields, ranging from income distribution analysis to poverty assessment and population variance estimation. Papatheodorou *et al.* (2004) underscores its efficacy in unveiling nuanced disparities within income distributions across different European countries, shedding light on the ramifications of income polarization and concentration. However, Minoiu and Reddy (2008) brings attention to the inherent biases introduced when KDE is applied to grouped data, particularly in poverty estimation, urging for caution in parameter selection. Addressing this concern, Kuk (1993) proposes a method amalgamating auxiliary information with kernel estimation to mitigate such biases, while Chen and Wang (2011) devises a bipartite recursive algorithm grounded in KDE for gauging the scale of specific income populations. This confluence of research highlights the promise KDE holds in estimating taxpayer numbers and scrutinizing taxpayer audit. Nevertheless, the discourse underscores the imperative of meticulous consideration of data characteristics and parameter choices to ensure robust and reliable estimations. A merge between a parametric model used for distribution of error term in the polynomial regression model and the polynomial movement on the data set as time series is studied by (Çankaya and Aydın 2024).

In Turkiye, the tax audit process is managed by the Presidency of the Tax Audit Board under the Ministry of Treasury and Finance. An important element of tax audit is expressed by the term "tax inspection" as it is understood in the activity reports of the Presidency. Article 134 of the Tax Procedure Act states that the main objective of tax inspection is to investigate and ensure the correctness of tax payments. Accordingly, tax inspectors check whether taxpayers have fulfilled their tax obligations in accordance with the legislation and whether they have correctly determined the

Manuscript received: 20 May 2024,
Revised: 8 October 2024,
Accepted: 8 November 2024.

[1] mehmet.cankaya@usak.edu.tr
[2] murat.aydin@usak.edu.tr (**Corresponding author**).

actual tax base. Tax inspection is not limited to the detection of tax evasion, but also includes the purpose of informing taxpayers of their tax obligations and verifying the elements of their tax returns.

A range of studies have explored the distribution of taxpayers and taxpayer audit. Chamberlain and Prante (2007); Piketty *et al.* (2018); Serikova *et al.* (2020) both highlight the progressive nature of the U.S. tax system, with the former emphasizing the impact of government spending on this progressive. Johns and Slemrod (2010); Davidson and Duclos (1997) delve into the distributional consequences of income tax noncompliance and the statistical inference for measuring the incidence of taxes and transfers, respectively. Ruggles and O'Higgins (1981); Piketty *et al.* (2017) both examine the distributive impact of government expenditures, with the latter focusing on the distribution of national income. Chotikapanich (2008); Perese (2015) provide methodological approaches for estimating income distributions and analyze the distribution of household income and federal taxes, respectively. Tax audit outcomes can lead to considerable adjustments in how companies recognize and value tax benefits, ultimately affecting their financial statements and tax strategies (Brushwood *et al.* 2018; Cowx and Vernon 2023).

The organization of the paper is given in the following order: The first section is for the introductory knowledges from literature. The second section gives real data. The method and objective are represented by third section. The forthcoming sections provide the statistical evaluations and their numerical results. The last section is divided into section for the conclusion.

DATA ON TAXPAYERS IN TURKIYE

Within the scope of the study, the data were obtained from the annual reports published on the official website of the Presidency of the Tax Audit Board. The reports covering the period between 2012 and 2023 contain informations which are total number of taxpayers and taxpayer audit.

Table 1 Taxpayers and their Audit by Years (VDK 2023)

Year	Total number of taxpayers	Taxpayer audit
2012	2,422,975	46,845
2013	2,460,281	71,352
2014	2,472,658	55,284
2015	2,527,084	58,676
2016	2,541,016	49,817
2017	2,636,370	44,182
2018	2,727,208	44,376
2019	2,813,452	40,763
2020	3,004,329	47,597
2021	3,221,084	54,065
2022	3,443,964	77,610
2023	3,621,478	60,242

Table 1 presents the total number of taxpayers and the number of taxpayers audited for certain years. In general, the table shows that the number of taxpayers increases each year and that the number of taxpayer audit generally shows an increasing trend. This may imply that tax controls cannot be applied to all taxpayers due to the limited resources of the tax administration or other priorities. In particular, there can be a significant decrease in the numbers of audit in years 2013-2014, 2015-2016, 2018-2019 and 2022-2023. There is an increase in trending of taxpayer audit from years 2012-2013 and 2021-2022. These numbers may indicate that the tax administration's strategies or resources have changed or that it has turned to other ways of administrative process of tax management system.

The continuous increase in the total number of taxpayers may reflect the expansion of the tax system or the fact that more people are becoming taxpayers as the economy grows. However, low numbers of audit may indicate that tax compliance is not at the desired level or that the tax administration is not using its audit resources effectively. The next section provides the modeling of data sets in Table 1 and the artificial data sets generated from the estimated functions determined by modeling.

METHOD AND OBJECTIVE

Kernel Estimation Method

One of the key advantages of the kernel mixture distribution is its ability to fit complex and multimodal data distributions. Unlike traditional parametric models, which make assumptions about the underlying data distribution, the kernel mixture distribution is non-parametric, meaning that it can adapt to the shape and structure of the data without imposing strict constraints, that is, it is data-adaptive and so the smoothness property will guarantee to fit data set well. This flexibility makes it particularly suitable for analyzing data sets with different patterns and characteristics.

It also provides a versatile framework for a variety of statistical tasks, including kernel mixture distribution, density estimation, clustering, and anomaly detection. By adjusting parameters such as the bandwidth of the kernels and the number of components in the mixture, analysts can fine-tune the distribution to capture different aspects of the data and achieve the desired level of granularity (Wand and Jones 1994).

In cases where the sample size is small, it is considered prudent to use applied techniques and artificial datasets to avoid bad effects. It should be noted, however, that for the smoothing technique, the alternative smoothing function can also be tried to obtain a possibly more accurate modeling; the figures will be close to the results already obtained, since the number of replications is increased to generate the artificial data. In addition, the parametric model proposed provides a comparison between the parametric function and the smoothing function. The results of the proposed distribution show that the smooth function is able to perform an accurate fit compared to the trimodal normal distribution as a parametric model (Vila *et al.* 2024b).

Various techniques and measures are used in data analysis to overcome the difficulties of working with small sample sizes. The use of artificial datasets can be very useful in such cases, allowing the creation of additional data points to supplement the original sample. This can help to correct for irregularities or gaps in the data and increase the robustness of the analysis. In addition, the use of smoothing techniques can improve the modelling process by reducing noise and highlighting important patterns in the data. Exploring alternative smoothing functions can further improve the modelling process and potentially lead to more accurate results. The comparison between parametric models and smooth functions, as proposed by Vila *et al.* (2024b), sheds light on the effectiveness of different modelling approaches. The smooth function appears to outperform the parametric model, especially when compared to the tri-modal normal distribution.

Kernel estimation as a non-parametric method is a statistical method used to estimate the probability density function, $f(x)$, of a random variable based on a sample of data. It uses kernel functions to set a smoothness in order to fit the data and create estimates for the parameters such as location, scale, etc. of the underlying distribution.

Given a data sample, $x_1, x_2, \ldots, x_n$, the kernel density estimate of the function $f(x)$ at a point x is calculated as:

$$f(x) = \frac{1}{nh}\sum_{i=1}^{n} K\left(\frac{x - x_i}{h}\right) \tag{1}$$

- $f(x)$ is the estimated density at point x for the function $f(x)$.
- n is the number of data points in the sample.
- h is the bandwidth parameter, which controls the width of the kernel.
- $K(\cdot)$ is the kernel function, a smooth, symmetric function centered around zero. Common choices for the kernel function include the Gaussian, Epanechnikov, and uniform kernels. The choice of kernel affects the shape of the estimated density (Wand and Jones 1994; Wolfram 2003).

In `Mathematica`, the `KernelMixtureDistribution` function is used to create a kernel mixture model for density estimation. By default, `KernelMixtureDistribution` uses a Gaussian (normal) kernel for the estimation (Wand and Jones 1994; Wolfram 2003).

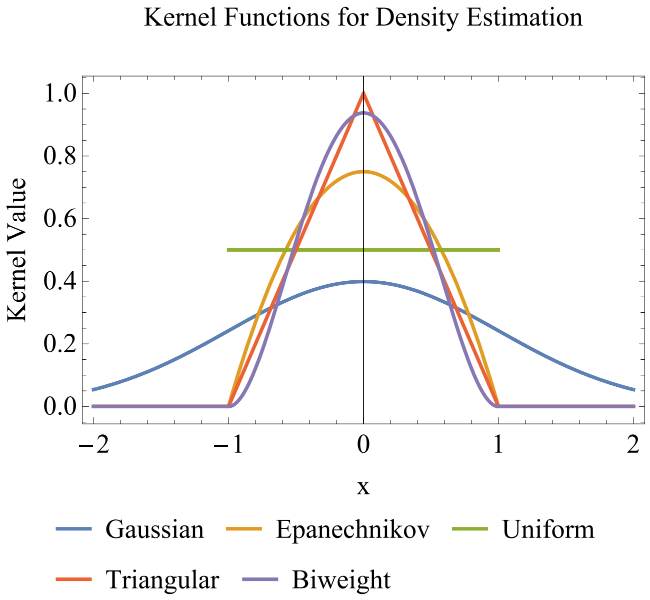


Figure 1 Kernel functions

Since the kernel functions used in `Mathematica` are very close to each other, the mixture form of the normal distribution was preferred due to the number of pages and the complexity of the results in the Figures at this paper. On the other hand, Gaussian (normal) kernel have an ability to fit the data where take the values at the interval $[-2, 2]$ when compared with other kernel functions such as Epanechnikov, Triangular, Biweight, etc. In addition, the function `FindDistribution` is also used to be able to perform a precise evaluation while getting the weights from differences of cumulative distribution function (see codes in Appendix).

Robust Estimations for Location and Scale Parameters

Robust statistics and kernel estimation share the goal of dealing with non-standard data distributions and mitigating the effects of outliers. Robust statistics focuses on developing methods that are resistant to outliers and deviations from standard assumptions. Robust estimators, such as the median or trimmed mean, are less affected by extreme values than traditional estimators such as the mean (Maronna *et al.* 2019).

Kernel estimation, often used in non-parametric density estimation, involves smoothing the data using a kernel function to estimate the underlying probability density function. This approach provides flexibility in modelling complex data distributions without assuming a specific parametric form. However, kernel estimation can be sensitive to outliers, leading to biased estimates, especially in regions of sparse data (Wand and Jones 1994).

The connection between robust statistics and kernel estimation lies in their complementary roles in dealing with challenging data scenarios. While kernel estimation provides flexibility and adaptability in modelling diverse data distributions, robust statistical techniques provide stability and resistance to outliers. By combining the principles of robust statistics with kernel estimation, researchers can develop methods that are both flexible and robust, enabling more reliable inference and analysis in the presence of non-standard data distributions and outliers.

The log-likelihood form of location and scale family is used to obtain the weighted mean and the weighted variance. The mean is given by

$$\text{weightedMean}(w) = \frac{\sum_{i=1}^{n} (\text{Sort}(\text{ND}(w))_i \cdot \text{weights}(w)_i)}{\sum_{i=1}^{n} \text{weights}(w)_i} \tag{2}$$

- $\text{ND}(w)$: Function that returns the numerical data derived from w.
- $\text{weights}(w)$: Function that returns the weights corresponding to the elements of $\text{ND}(w)$.
- $\text{Sort}(\text{ND}(w))$: Sorted version of $\text{ND}(w)$ in ascending order.

The square root of the weighted variance is defined as weighted standard deviation given by the following form:

$$\text{weightedStD}(w) = \sqrt{\frac{1}{n}\sum_{i=1}^{n} \text{weights}(w)_i\,(\text{Sort}(\text{ND}(w))_i - \text{weightedMean}(w))^2} \tag{3}$$

- $\text{ND}(w)$: Function that returns the numerical data derived from w.
- $\text{weights}(w)$: Function that returns the weights corresponding to the elements of $\text{ND}(w)$.
- $\text{Sort}(\text{ND}(w))$: Sorted version of $\text{ND}(w)$ in ascending order.
- $\text{weightedMean}(w)$: Weighted mean of w, as defined previously.
- n: Length of $\text{ND}(w)$.

The theory of robust statistics is based on weights from the assumed or the chosen function (Maronna *et al.* 2019). In our case, the weights come from two separate functions. One is the kernel estimator and the other is the `FindDistribution` function included with `Mathematica` software `12.0.0.0`. The `FindDistribution` is a powerful tool for fitting a probabilistic model to a given dataset. Implemented in version 12.0.0.0, this function automatically identifies the most appropriate distribution from a set of candidate distributions using the statistical goodness-of-fit tests. It allows users to quickly determine the underlying statistical properties of their data, simplifying the statistical modelling (Wolfram 2003).

Kernel Mixture and Find Distributions in `Mathematica`

In `Mathematica`, the functions `KernelMixtureDistribution` and `FindDistribution` can be used to estimate a distribution based on a data sample and a chosen kernel function.

The syntaxes for the function are:

```
KernelMixtureDistribution[data,Automatic,"SemiCircle"]
FindDistribution[data]
```

- 'data' is the input data sample.
- 'Automatic' allows `Mathematica` to automatically select an appropriate bandwidth.
- '"SemiCircle"' specifies the semi-circle kernel function, which may be useful for specific types of data.
- The function `FindDistribution` returns the name or symbolic representation of the distribution that best fits the data. It can handle a wide range of distribution families, including but not limited to normal, uniform, exponential, gamma, beta, and many others.

The 'KernelMixtureDistribution' function returns a nonparametric distribution that can be used for further analysis. Consideration of factors such as numerical optimization and manufacturing process is crucial to ensure the validity and reliability of the results. The numerical values obtained from Figures 2-13 are likely to represent the results of these optimization processes and can give an idea of the performance of the modeling techniques automatically performed by the functions in `Mathematica 12.0.0.0`. By following these steps, you will have a comprehensive understanding of the characteristics of your synthetic datasets and be able to analyze their statistical properties effectively (Wolfram 2003).

Algorithmic Schema in Order

1. Transfer data set into the case where the unit interval is set: If your original data is not in the unit interval (i.e., the range [0, 1]), you'll need to scale it to fit within this range. The number of taxpayer audit is proportioned to the total number of taxpayers, thus obtaining data will be in the unit interval. If the randomly generated ratio values from estimated density, $f(x)$, are multiplied by the total number of taxpayers, then the number of taxpayer audit is obtained. If the number of taxpayers is divided by the randomly generated ratio values from estimated density, $f(x)$, the total number of taxpayers is obtained.
2. Model the unit interval data set: Once your data is in the unit interval, you can model it using a kernel estimation method. Since using `Mathematica`, the function, 'SK=KernelMixtureDistribution[x,Automatic,"SemiCircle"]', could be used to create a smooth kernel density estimate of your data. Using the model generated in the previous step, generate a synthetic dataset with the same characteristics as your original data.
3. Generate artificial data set with sample size $n = 12$: 'RandomVariate[SK,n=12]'; Use the 'RandomVariate' function in `Mathematica` to generate random samples from your estimated density, $f(x)$.
4. Once you have generated one synthetic dataset, replicate this process 10,000 times by using 'SK'. For each iteration, generate a new synthetic dataset. Multiplication by total taxpayers with generated ratio values gives the taxpayer audit and division of taxpayer audit with ratio values gives the total number of taxpayers (see also step 1).
5. Provide statistics: Once you have your 10,000 synthetic datasets, calculate statistics such as the first moment (mean), scale estimate (standard deviation), minimum, maximum, 1th, 25th, 75th and 99th percentiles for each dataset. You can also use functions like first moment and scale estimated from the estimated density, 'Min', 'Max', 'Quantiles (1%, 25%, 75% and 99%)', etc., in `Mathematica` to evaluate these statistics summarizing the general representation of the generated data set from estimated density.

When the smooth function from kernel method is used, the corresponding statistics such as first moment, scale estimate, etc. are calculated. Therefore, these statistics are more accurate due to the precise fitting performed with the smooth function. Since the artificial data set is replicated with a sample size of $n = 12$, the minimum and maximum values are selected for each set. The same process is done for the data sets at the 1th, 25th, 75th and 99th percentiles so that we can observe the behavior of the data set at these percentiles as probability values indicating what the values generated for these percentiles are. In other words, we can see the overall picture of the data generated for these values. Note that the computational and methodological processes are also used by references (Vila *et al.* 2024b; Özen and Çankaya 2023; Aydın and Çankaya 2024).

STATISTICAL EVALUATIONS

The kernel smoothing method in `Mathematica` is capable of performing a fitting on the data set. Further, since the assumed nonparametric density in this software is used to generate artificial data sets, we have a well-defined computational schema for evaluating various statistics and properties of your synthetic datasets. To summarize:

1. Empirical First Moment and Scale Estimate from Data Generated SK: These statistics are computed using built-in functions in `Mathematica` ($'Moment[data,1]'$ and $'Sqrt[Moment[data,2]-Moment[data,1]^2]'$) and are considered more accurate due to the precise fitting performed by the smooth function.
2. Minimum and Maximum Values: For each replicated artificial dataset, the minimum and maximum values are chosen. This provides insight into the range of values generated by the model.
3. Quantiles at 1th, 25th, 75th and 99th Percentiles: Similarly, for each replicated artificial dataset, the values at 1th, 25th, 75th and 99th percentiles are determined. This provides a picture of the overall distribution of the generated data and allows the behavior of the dataset around these quantities to be observed. It should also be noted that this calculation scheme is used with references (Vila *et al.* 2024b; Özen and Çankaya 2023; Aydın and Çankaya 2024), which shows its validity and suitability in practice. By following this scheme, you can effectively analyze the characteristics and behavior of your synthetic datasets and help your optimization process.
4. Weighted Statistics for Location Scale Parameters: The weighted mean and the weighted standard deviation based on the differences of cumulative distribution function from kernel smooth (KS) and FindDistribution (FD) are calculated.

The statistical evaluations are detailed quantitatively in the following section 'Numerical Results'. This section presents the empirical findings, providing a comprehensive analysis of the data and their implications for the study.

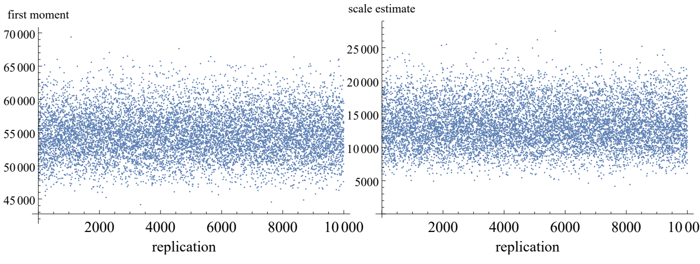
NUMERICAL RESULTS OF STATISTICAL EVALUATIONS

The generation of accurate simulated data can have a significant impact on various stakeholders, including policymakers, researchers, and industry professionals. These stakeholders rely on the quality and precision of the data to make fully informed decisions. While the simulated data are confidential, it is important to thoroughly and fully document the methods and approaches used to generate them. This documentation allows for reproducibility and validation of the outputs of simulation by other researchers or analysts in the field.

Each of Figures 2-13 shows summary statistics from the artificially generated data set for the sample size $n = 12$, replicated 10,000 times. There are two types of Figures. One of them represents the statistical values for taxpayer audit given by Figs. 2 - 5. The second one represents the statistical values for taxpayer given by Figs. 6 - 9. Further, note that we focus on the maximum values of the generated data set for the sake of the fact that the future probable prediction can also be evaluated and suggested as well. In such case, the maximum, 1%, 25%, 75% and 99% as order statistics for taxpayer audit and taxpayer are an open issue which should be studied intensively. We prefer to omit the topic about order statistics and Figures 2-13 give the general appearance, as mentioned above. Consequently, the numerical results can be the values shown in Figures 2-13, taking into account the numerical optimization and generation procedure from the estimated density, $f(x)$ estimated by using two estimation method which are non-parametric being kernel smooth and parametric being FindDistribution. The simulated data are therefore considered confidential as they are the best possible match to the observed data.

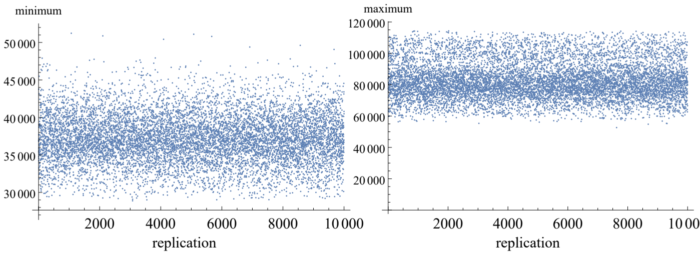
Figures 2a-2b and 6a-6b show the empirical first moment on average from the measure of central tendency and the scale estimate from the measure of dispersion. Figures 3 and 7 represent the minimum and maximum values of the data at a sample size of $n = 12$. Figures 4-5 and 8-9 represent the simulated data for $n = 12$ at 1%, 25%, 75% and 99% cut-offs, respectively.

When comparing Figures 3b and 7b for maximum of the artificial numbers from taxpayers audit and taxpayers, respectively, the more taxpayers lead to have detection of the more taxpayer audit, as expected.



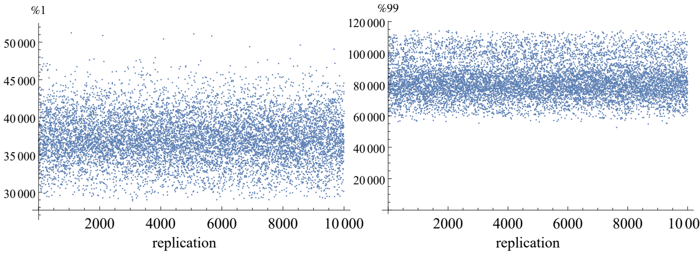
(a) Empirical first moment **(b)** Scale estimate

Figure 2 The simulated data for statistics of the taxpayer audit within years 2012-2023



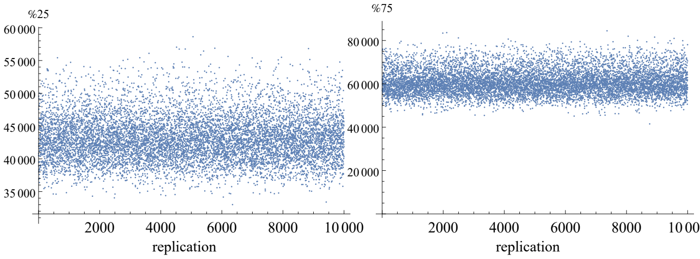
(a) Minimum values of artificial data **(b)** Maximum values of artificial data

Figure 3 The simulated data for minimum and maximum of the taxpayer audit within years 2012-2023



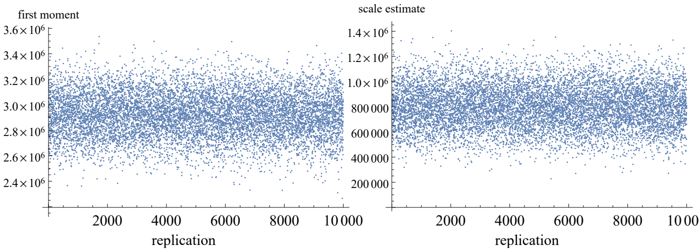
(a) Quantile at 1% of artificial data **(b)** Quantile at 99% of artificial data

Figure 4 The simulated data for quartiles at 1% & 99% of the taxpayer audit within years 2012-2023



(a) Quantile at 25% of artificial data **(b)** Quantile at 75% of artificial data

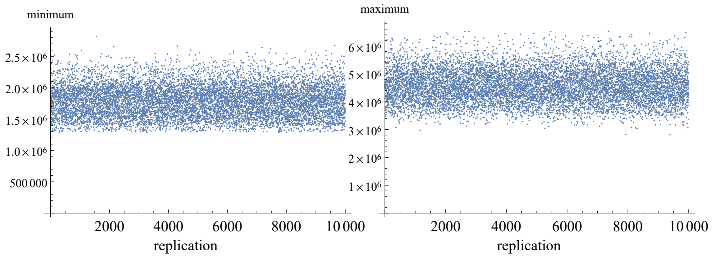
Figure 5 The simulated data for quartiles at 25% & 75% of the taxpayer audit within years 2012-2023



(a) Empirical first moment **(b)** Scale estimate

Figure 6 The simulated data for statistics of the taxpayers within years 2012-2023

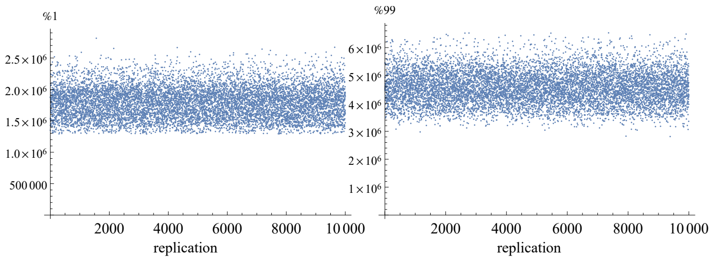
Figures 10-13 show the robust estimates replicated at 10,000 times for location and scale parameters. When Figure 10a is compared with Figure 11a, the results show that the values of weighted mean from FD tends to take lower values, which shows that the chosen function for fitting data set affects the results we will get, because the function chosen plays role in determining the weights for the robust estimation. In addition, the scale estimate given by Figure 2b have values which are bigger than that of values in Figure 10b. The same situation for taxpayers at Figures 6b and 12b is observed. Figures 10-11 represent the case where the numbers of taxpayer audit are around. Figures 12-13 represent the case



(a) Minimum values of artificial data

(b) Maximum values of artificial data

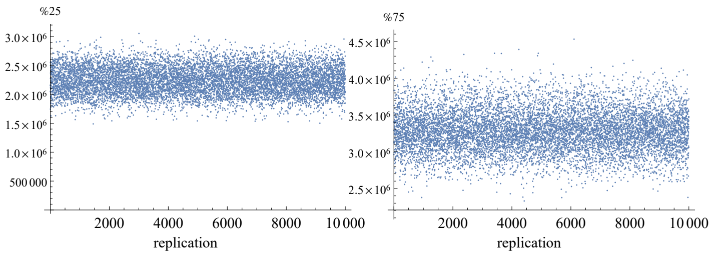
Figure 7 The simulated data for minimum and maximum of the taxpayers within years 2012-2023



(a) Quantile at 1% of artificial data

(b) Quantile at 99% of artificial data

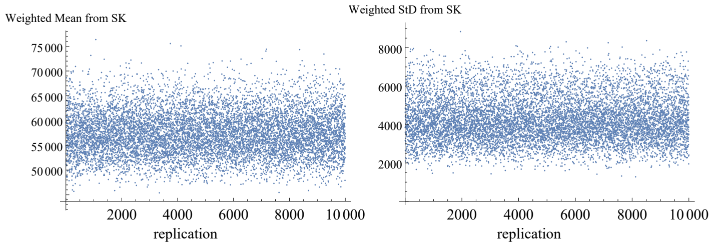
Figure 8 The simulated data for quartiles at 1% & 99% of the taxpayers within years 2012-2023



(a) Quantile at 25% of artificial data

(b) Quantile at 75% of artificial data

Figure 9 The simulated data for quartiles at 25% & 75% of the taxpayers within years 2012-2023

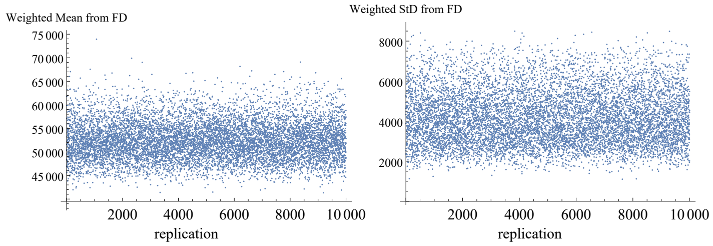


(a) Weighted mean from Smooth Kernel(SK)

(b) Weighted standard deviation from Smooth Kernel

Figure 10 The weighted forms of location and scale estimates from Smooth Kernel, $n = 12$ for the taxpayer audit

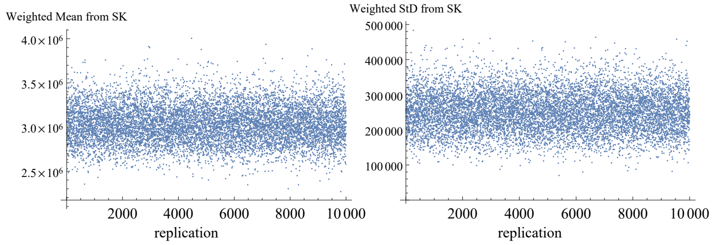
where the numbers of taxpayer are around.



(a) Weighted mean from Find Distribution (FD)

(b) Weighted standard deviation from Find Distribution

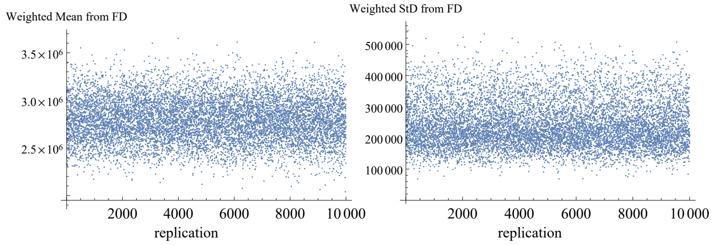
Figure 11 The weighted forms of location and scale estimates from Find Distribution, $n = 12$ for the taxpayer audit



(a) Weighted mean from Smooth Kernel

(b) Weighted standard deviation from Smooth Kernel

Figure 12 The weighted forms of location and scale estimates from Smooth Kernel, $n = 12$ for the taxpayers



(a) Weighted mean from Find Distribution

(b) Weighted standard deviation from Find Distribution

Figure 13 The weighted forms of location and scale estimates from Find Distribution, $n = 12$ for the taxpayers

Number of Taxpayers at per square Kilometer of Turkiye

To find the number of taxpayers per square kilometre, you must first find the total number of taxpayers in a given area, and then divide that number by the total area of that area, measured in square kilometre. This calculation will give you the density of taxpayers, which indicates how many taxpayers live or are based in each square kilometre of the area in question.

Considering the maximum value in the simulated data from kernel estimation method, there are approximately 6 million taxpayers from Figure 7b. Turkiye, with a population of 85.8 million, is in the taxpayer's role at a rate of $6/85.8 = 0.06993$.

Since there are 110 people per square kilometre in Turkey, $110 \cdot 0.06993 = 7.69$, approximately 8 out of 110 people per square kilometre will be identified as taxpayers if they are evenly distributed across the regions of Turkey (Wolfram 2003). This measure is essential for understanding the distribution of taxpayers across the country, which can help in effective policy making, resource allocation and economic planning.

CONCLUSION

The study has highlighted the importance of using sophisticated statistical methods to accurately model these complex, multimodal data distributions which can be modeled by using the kernel estimation methods which provide robust and versatile framework for statistical analysis. They can handle complex/multimodal data distributions such as the ratio values between the taxpayer audit and its total numbers. Firstly, the kernel estimation method has been used. After that, the `FindDistribution` function included with `Mathematica` software `12.0.0.0` is used to model the generated data artificially from `SK=KernelMixtureDistribution` with `RandomVariate[SK, n=12]`.

The values of ratio give an advantage for us to fit the data set well. In addition, the total taxpayers and the taxpayer audit can be calculated from the values of ratio. When the results for the maximum number of taxable persons and taxpayers are compared, the more taxable persons lead to the detection of the more taxpayer audit, as expected. According to the maximum statistic of the simulated data, the total numbers of taxpayers can go up to 6 million, which can occur in the near future. The taxpayer audit will be around 120,000 from maximum statistics. The simulation results for the taxpayer audit have shown that there are two blocks for the numbers which are 80,000 and 100,000. The values around 100,000 are few when compared with that of 80,000, which shows that the audit on the taxpayers is commented as two populations and some precautions for tax audit can be necessary when the maximum statistics are taken into account.

In the same way, when observing the number of taxpayers, there can be two populations which are 4.5 million and 6 million. Additional statistics, including the weighted means for taxpayer audits and overall taxpayer population, which are approximately 75,000 and 4 million respectively, are also provided to summarize general characteristics of the data set. As a result, more taxpayers should be surveyed. Further, by improving our understanding of these populations; policymakers and tax authorities can implement more effective policies to optimize tax audit processes and ensure fair tax compliance among different taxpayer groups. Our ability for modeling, estimating and understanding the number of taxpayers and its audit form with precision and confidence intervals will continue to improve with the continued research and improvements in this area.

APPENDIX

The `Mathematica` codes for computation and statistical evaluations

`Mathematica`, developed by Wolfram Research, is a comprehensive computational software system widely used in various fields of science, engineering, mathematics and computing. It features a high-level programming language, powerful computational capabilities, and a wide range of built-in functions, making it an indispensable tool for research, education, and industrial applications.

The codes were used to model the proportional data using the kernel mixture distribution in `Mathematica` 12.0.0.0 software.

```
For[w=1, w <= rep, w++,
  (* Fitting via Smooth Kernel Method *)
SK[w]=KernelMixtureDistribution[x, Automatic, "SemiCircle"];

(* Vector for scaling the generated random numbers *)
  vec={data};

(* Generate random numbers based on the kernel
distribution and scale them by vec *)
  ND[w]=RandomVariate[SK[w], 12] * vec;

(* Calculate the CDF of the kernel distribution for
sorted ND[w] *)
 CDFSK[w]=CDF[KernelMixtureDistribution[ND[w]],Sort[ND[w]]];

(* Calculate the CDF of a fitted distribution for
  sorted ND[w] *)
 CDFFD[w]=CDF[FindDistribution[ND[w]],Sort[ND[w]]];

(* Calculate weights based on the differences in the CDF
  for the kernel distribution *)
  weights1SK[w]=Differences[CDFSK[w]];

(* Calculate the remaining weight to ensure the weights
  sum to 1 *)
weights2SK[w]=1 - Total[weights1SK[w]];

(* Combine the weights and ensure they sum up to 1 *)
  weightsSK[w]=Join[weights1SK[w], {weights2SK[w]}];

(* Calculate weights based on the differences in the CDF
  for the fitted distribution *)
  weights1FD[w]=Differences[CDFFD[w]];

(* Calculate the remaining weight to ensure the weights
  sum to 1 *)
  weights2FD[w]=1 - Total[weights1FD[w]];

(* Combine the weights and ensure they
  sum up to 1 *)
  weightsFD[w]=Join[weights1FD[w], {weights2FD[w]}];

(* Calculate various statistics for ND[w] *)
  sta1[w]:=Moment[ND[w], 1];  (* First moment (mean) *)
  ta1=Table[sta1[w], {w, rep}];

  sta2[w]:=Mean[ND[w]];  (* Mean *)
  ta2=Table[sta2[w], {w, rep}];

  sta3[w]:=Median[ND[w]];  (* Median *)
  ta3=Table[sta3[w], {w, rep}];

 (* Standard deviation based on moments *)
  sta4[w]:=Sqrt[Moment[ND[w], 2] - Moment[ND[w], 1]^2];
  ta4=Table[sta4[w], {w, rep}];

  (* Standard deviation *)
  sta5[w]:=StandardDeviation[ND[w]];
  ta5=Table[sta5[w], {w, rep}];

  (* Median absolute deviation from the median *)
  sta6[w]:=Median[Abs[ND[w] - Median[ND[w]]]];
  ta6=Table[sta6[w], {w, rep}];

   (* Median absolute deviation from the mean *)
  sta61[w]:=Median[Abs[ND[w] - sta1[w]]];
  ta61=Table[sta61[w], {w, rep}];

  (* Mean absolute deviation from the mean *)
  sta62[w]:=Mean[Abs[ND[w] - sta1[w]]];
  ta62=Table[sta62[w], {w, rep}];

  (* Minimum value *)
  sta7[w]:=Min[ND[w]];
  ta7=Table[sta7[w], {w, rep}];

  (* Maximum value *)
  sta8[w]:=Max[ND[w]];
  ta8=Table[sta8[w], {w, rep}];

  (* Calculate various quantiles *)

  (* 1st percentile *)
  staquan1[w]:=Quantile[ND[w], 0.01];
  taga1=Table[staquan1[w], {w, rep}];

  (* 25th percentile *)
  staquan2[w]:=Quantile[ND[w], 0.25];
  taga2=Table[staquan2[w], {w, rep}];

  (* 50th percentile / median *)
  staquan3[w]:=Quantile[ND[w], 0.5];
  taga3=Table[staquan3[w], {w, rep}];

  (* 75th percentile *)
  staquan4[w]:=Quantile[ND[w], 0.75];
  taga4=Table[staquan4[w], {w, rep}];

  (* 99th percentile *)
  staquan5[w]:=Quantile[ND[w], 0.99];
  taga5=Table[staquan5[w], {w, rep}];

  (* Calculate the weighted mean based on
  the smooth kernel (SK) distribution weights *)
  weightedMeanSK[w]:=
    Total[Sort[ND[w]] * weightsSK[w]]
    /
    Total[weightsSK[w]];
  taWeMeSK=Table[weightedMeanSK[w], {w, rep}];

  (* Calculate the weighted standard deviation
  based on the smooth kernel distribution weights *)
  weightedStDSK[w]:=
Sqrt[Total[weightsSK[w]*(Sort[ND[w]]-weightedMeanSK[w])^2]
    /
    Length[ND[w]]];
  taWeSDSK=Table[weightedStDSK[w], {w, rep}];

  (* Calculate the weighted mean based
  on the fitted distribution (FD) weights *)
  weightedMeanFD[w]:=
    Total[Sort[ND[w]] * weightsFD[w]]
    /
    Total[weightsFD[w]];
  taWeMeFD=Table[weightedMeanFD[w], {w, rep}];

  (* Calculate the weighted standard deviation
  based on the fitted distribution weights *)
  weightedStDFD[w]:=
Sqrt[Total[weightsFD[w]*(Sort[ND[w]]-weightedMeanFD[w])^2]
    /
    Length[ND[w]]];
  taWeSDFD=Table[weightedStDFD[w], {w, rep}];
]
```

Acknowledgments

We appreciate the editorial board's and reviewers' valuable comments on the paper.

Availability of data and material

The real data set is included by the paper.

Conflicts of interest

The authors declare that there is no conflict of interest regarding the publication of this paper.

Ethical standard

The authors have no relevant financial or non-financial interests to disclose.

LITERATURE CITED

Alleva, G. and A. E. Giommi, 2016 *Topics in Theoretical and Applied Statistics*. Springer.

Aydın, M. and M. N. Çankaya, 2024 Assessing the regulatory impact of the turkish competition authority on market dynamics: A statistical approach using kernel estimation and its simulation. Journal of Mehmet Akif Ersoy University Economics and Administrative Sciences Faculty **11**: 837–853.

Brushwood, J. D., D. M. Johnston, and S. J. Lusch, 2018 The effect of tax audit outcomes on the reporting and valuation of unrecognized tax benefits. Advances in Accounting **42**: 1–11.

Çankaya, M. N., 2020a M-estimations of shape and scale parameters by order statistics in least informative distributions on q-deformed logarithm. Journal of the Institute of Science and Technology **10**: 1984–1996.

Çankaya, M. N., 2020b On the robust estimations of location and scale parameters for least informative distributions. Turkish Journal of Science and Technology **15**: 71–78.

Çankaya, M. N. and O. Arslan, 2020 On the robustness properties for maximum likelihood estimators of parameters in exponential power and generalized t distributions. Communications in Statistics-Theory and Methods **49**: 607–630.

Çankaya, M. N. and M. Aydın, 2024 Future prediction for tax complaints to turkish ombudsman by models from polynomial regression and parametric distribution. Chaos Theory and Applications **6**: 63–72.

Çankaya, M. N. and J. Korbel, 2018 Least informative distributions in maximum q-log-likelihood estimation. Physica A: Statistical Mechanics and its Applications **509**: 140–150.

Çankaya, M. N. and R. Vila, 2023 Maximum log q likelihood estimation for parameters of weibull distribution and properties: Monte carlo simulation. Soft Computing **27**: 6903–6926.

Çankaya, M. N., A. Yalçınkaya, Ö. Altındağ, and O. Arslan, 2019 On the robustness of an epsilon skew extension for burr iii distribution on the real line. Computational Statistics **34**: 1247–1273.

Çankaya, M. N., 2021 Derivatives by ratio principle for q-sets on the time scale calculus. Fractals **29**: 2140040.

Chamberlain, A. and G. Prante, 2007 Who Pays Taxes and Who Receives Government Spending? An Analysis of Federal, State and Local Tax and Spending Distributions, 1991-2004. SSRN Electronic Journal .

Chen, Y. and H. Wang, 2011 Construction and application of bipartite recursive algorithm based on kernel density estimation: A new non-parametric method to measure the given income population scale. In *Statistics & Information Forum*, pp. 3–8.

Chotikapanich, D., 2008 *Modeling income distributions and Lorenz curves*, volume 5. Springer Science & Business Media.

Cowx, M. and M. Vernon, 2023 Accounting for tax uncertainty over time. Available at SSRN 4678373 .

Davidson, R. and J.-Y. Duclos, 1997 Statistical Inference for the Measurement of the Incidence of Taxes and Transfers. Econometrica **65**: 1453.

Hanif, M. and U. Shahzad, 2019 Estimation of population variance using kernel matrix. Journal of Statistics and Management Systems **22**: 563–586.

Johns, A. and J. Slemrod, 2010 The distribution of income tax noncompliance. National Tax Journal **63**: 397–418.

Kuk, A. Y., 1993 A kernel method for estimating finite population distribution functions using auxiliary information. Biometrika **80**: 385–392.

Maronna, R. A., R. D. Martin, V. J. Yohai, and M. Salibián-Barrera, 2019 *Robust statistics: theory and methods (with R)*. John Wiley & Sons.

Minoiu, C. and S. Reddy, 2008 Kernel density estimation based on grouped data: The case of poverty assessment. IMF Working Papers **08**: 1.

Özen, E. and M. N. Çankaya, 2023 Estimation of the turkish stock investor numbers based on kernel method. In *Competitivitatea şi inovarea în economia cunoaşterii*, pp. 445–454.

Papatheodorou, C., P. Peristera, and A. Kostaki, 2004 Kernel density techniques as a tool for estimating and comparing income distributions: a cross european–country study. Journal of Income Distribution **13**: 2–2.

Perese, K., 2015 The distribution of household income and federal taxes, 2011. Current Politics and Economics of the United States, Canada and Mexico **17**: 695.

Piketty, T., E. Saez, and G. Zucman, 2017 Distributional National Accounts: Methods and Estimates for the United States*. The Quarterly Journal of Economics **133**: 553–609.

Piketty, T., E. Saez, and G. Zucman, 2018 Distributional national accounts: methods and estimates for the united states. The Quarterly Journal of Economics **133**: 553–609.

Ruggles, P. and M. O'Higgins, 1981 The distribution of public expenditure among households in the united states. Review of Income and Wealth **27**: 137–164.

Serikova, M., L. Sembiyeva, K. Balginova, G. Alina, A. Shakharova, *et al.*, 2020 Tax revenues estimation and forecast for state tax audit. Entrepreneurship and Sustainability Issues **7**: 2419–2435.

VDK, T., 2023 Vdk annual reports. https://en-vdk.hmb.gov.tr/annual-reports, [Online; accessed 10-May-2024].

Vila, R., L. Alfaia, A. F. Menezes, M. N. Çankaya, and M. Bourguignon, 2024a A model for bimodal rates and proportions. Journal of Applied Statistics **51**: 664–681.

Vila, R. and M. N. Çankaya, 2022 A bimodal weibull distribution: properties and inference. Journal of Applied Statistics **49**: 3044–3062.

Vila, R., V. Serra, M. N. Çankaya, and F. Quintino, 2024b A general class of trimodal distributions: properties and inference. Journal of Applied Statistics **51**: 1446–1469.

Wand, M. P. and M. C. Jones, 1994 *Kernel smoothing*. CRC press.

Wolfram, S., 2003 *The mathematica book*. Wolfram Research, Inc.

How to cite this article: Cankaya, M.N., and Aydin M. Tax Audit in Turkiye: Simulation and Estimations Based on Kernel and Weight Functions *Chaos Theory and Applications*, 6(4), 264-272, 2024.

CHAOS
Theory and Applications
in Applied Sciences and Engineering

e-ISSN: 2687-4539
RESEARCH ARTICLE
Vol.6 / No.4 / 2024 / pp.273-283
https://doi.org/10.51537/chaos.1513080

A New 6D Two-wing Hyperhaotic System: Dynamical Analysis, Circuit Design, and Sinchronization

Michael Kopp[*,1] **and Inna Samuilik**[α,β,2]
[*]Institute for Single Crystals, NAS Ukraine, Nauky Ave. 60, Kharkiv 61072, Ukraine, [α]Institute of Life Sciences and Technologies, Daugavpils University, 13 Vienibas Street, LV-5401 Daugavpils, Latvia, [β]Institute of Applied Mathematics, Riga Technical University, LV-1048 Riga, Latvia.

ABSTRACT This paper introduces a novel 6D dynamic system derived from modified 3D Lorenz equations of the second type using state feedback control. While the original 3D equations are formally simpler than the classical Lorentz equations, they produce topologically more complex attractors with a two-winged butterfly structure. The proposed system contains the fewest terms compared to existing literature. These terms comprise two cross-product nonlinearities, two piecewise linear functions, six linear terms, and one constant. The new 6D hyperchaotic system exhibits a rich array of dynamic characteristics, including hidden attractors and dissipative behavior. A thorough dynamic analysis of this system was performed. In particular, bifurcation diagrams were constructed, Lyapunov exponents and dimensions were calculated, and multistability and offset boosting control were analyzed to understand the systems behavior further. An electronic circuit of the 6D hyperchaotic two-winged butterfly system was developed in the Multisim computer environment. The designed electronic circuit showed excellent agreement with the simulation results of the new 6D dynamic system. Synchronization of two identical 6D hyperchaotic systems was achieved using the active control method.

KEYWORDS
Two-wing attractors
Chaotic behavior
Multistability
Offset boosting control
Circuit implementation
Active control synchronization

INTRODUCTION

Since Lorenz's discovery of a three-dimensional (3D) chaotic system (Lorenz 1963), chaos researchers have increasingly focused on studying dynamic systems with dimensions higher than three. This trend is driven by several reasons. Firstly, many physical phenomena cannot be adequately modeled by three-dimensional systems. Higher-dimensional systems can capture the more complex behaviors and interactions observed in fields such as hydrodynamic turbulence theory (Bohr *et al.* 1998), climate modeling (Soldatenko *et al.* 2021), and neurodynamics (Yin *et al.* 2022). Another reason is that higher-dimensional complex systems are often employed in cryptographic applications due to their increased unpredictability and difficulty in being reverse-engineered. This makes them ideal for secure communications and information encryption (Ramakrishnan 2018).

Numerous 4D hyperchaotic systems have been thoroughly documented in the literature. These include Lorenz's hyperchaotic system (Jia 2007), Chen's hyperchaotic system (Chen *et al.* 2006), Liu's hyperchaotic system (Li 2009), the hyperchaotic Wang system (Wang and Chen 2008), the hyperchaotic Newton-Leipnik system (Ghosh and Bhattacharya 2010), and the hyperchaotic Vaidyanathan system (Vaidyanathan 2013). When constructing new hyperchaotic models, it is essential to consider several factors: the presence of multiple positive Lyapunov exponents, maintaining the smallest number of terms to meet the simplicity criteria established by researcher Sprott, and achieving the highest Kaplan-Yorke dimension.

Moreover, noteworthy among the issues in chaos theory are those of chaos control and synchronization. Chaos control for practical systems has been the subject of extensive research. The master, or drive system, and the slave, or response system, are two systems whose synchronization is the subject of the chaos synchronization problem. Control laws are created to address this issue by ensuring that, asymptotically over time, the output of the slave system tracks the output of the master system. Numerous techniques have been proposed, including active control (Jung *et al.* 2019; Bhat and Shikha 2019), adaptive control (Zhang *et al.* 2020; Tohidi *et al.* 2020; Vaidyanathan *et al.* 2014; Vaidyanathan and Volos 2015), backstepping control (Chu and Hu 2016), sliding mode control (Rajagopal *et al.* 2017a,b; Yousefpour *et al.* 2020), and

Manuscript received: 9 July 2024,
Revised: 31 October 2024,
Accepted: 9 November 2024.

[1]michaelkopp0165@gmail.com
[2]inna.samuilika@rtu.lv (**Corresponding author**).

Table 1 List of recently cited 6D dynamical systems.

Reference	Total terms	Number nonlinear terms	Nature system
Benkouider *et al.* (2020)	17	2	Dissipative
Sabaghian *et al.* (2020)	15	2	Dissipative
Yang *et al.* (2020)	14	3	Dissipative
Al-Azzawi and Al-Obeidi (2021)	17	2	Dissipative
Aziz and Al-Azzawi (2022)	13	3	Dissipative
Al-Talib and Al-Azzawi (2022)	12	4	Dissipative
Al-Obeidi and Al-Azzawi (2022)	17	3	Dissipative
Michael Kopp and Andrii Kopp (2022)	17	2	Dissipative
Al-Talib and Al-Azzawi (2023a)	12	4	Dissipative
Al-Azzawi and Al-Obeidi (2023)	17	3	Dissipative
Kopp *et al.* (2023)	21	4	Dissipative
Khattar *et al.* (2024)	12	4	Dissipative
This work	**11**	**2**	**Dissipative**

so on. Recently, a passive control method has also been presented in the literature. In paper (Adıyaman *et al.* 2020), a passive control method was presented to stabilize a new 4D hyperchaotic system at zero equilibrium and synchronize two identical new 4D hyperchaotic systems with different initial conditions. In another paper (Emiroglu *et al.* 2022), a passive control method was described to stabilize and suppress chaos in a chaotic system. These control techniques can also be used to achieve different types of real chaos synchronization.

Recently, there has been a trend towards constructing hyperchaotic models with higher dimensions, such as 5D models with three positive Lyapunov exponents (Hu 2009; Yang and Chen 2013; Al-Azzawi and Hasan 2023), 6D models with four positive Lyapunov exponents (Al-Talib and Al-Azzawi 2023b), and 7D models with five positive Lyapunov exponents (Yang *et al.* 2018). Compared to standard 3D and 4D models, these higher-dimensional chaotic models exhibit greater unpredictability and complexity. As can be seen from Table 1, most 6D dynamical systems (Benkouider *et al.* 2020; Yang *et al.* 2020; Al-Azzawi and Al-Obeidi 2021; Al-Obeidi and Al-Azzawi 2022; Kopp *et al.* 2023) consist of 12 or more terms with dissipative nature, and no simple dissipative 6D hyperchaotic system consisting of only 11 terms has been found. In addition, the proposed 6D system has a simple structure, containing only two control parameters. This motivated us to search for a new hyperchaotic system that contains the smallest number of terms.

This manuscript consists of the following sections. The Introduction provides a brief overview of the current state of the problem. Section 2 gives the derivation of the new 6D hyperchaotic dynamic system using state feedback control. In Section 3, we examine the dynamic characteristics of the new 6D nonlinear system by analyzing the fixed points, constructing bifurcation diagrams, and determining the spectrum and Lyapunov dimension. This section also delves into multistability and offset boosting control for the new system. Section 4 is dedicated to developing an electronic circuit for a hyperchaotic chaos generator using the Multisim environment. The circuit's operation was tested, and the simulation results were compared with those obtained in the Mathematica environment. Finally, in Section 5, we extend our focus to the numerical analysis of synchronization between two identical 6D hyperchaotic systems. We utilized the active control method (see, for example, (Jung *et al.* 2019; Bhat and Shikha 2019)) to achieve synchronization. The Conclusions section presents the main results obtained in this article.

DERIVATION OF A NEW 6D HYPERCHAOTIC DYNAMIC SYSTEM

In this section, we outline a method to derive a new six-dimensional (6D) dynamical system from a modified Lorenz system (Elwakil *et al.* 2002) of the following form:

$$\begin{cases} \dfrac{dx_1}{dt} = a(-x_1 + x_2) \\ \dfrac{dx_2}{dt} = -x_3\mathrm{sgn}(x_1) \\ \dfrac{dx_3}{dt} = |x_1| - 1 \end{cases} \tag{1}$$

Here $|x|$ is the absolute value function, signum function $\mathrm{sgn}(x)$ of

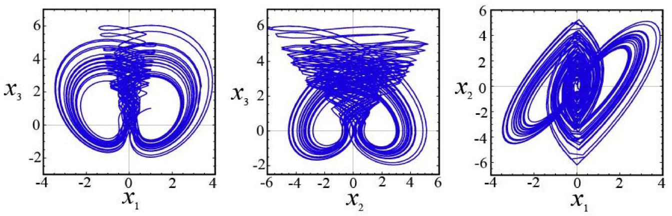


Figure 1 Plots depict two-wing butterfly attractors of system (1) in phase planes x_1x_3, x_2x_3, and x_1x_2, respectively.

a real number x is a piecewise function which is defined as follows:

$$\mathrm{sgn}(x) = \begin{cases} -1 \text{ if } x < 0 \\ \ \ 0 \text{ if } x = 0 \\ \ \ 1 \text{ if } x > 0 \end{cases} \tag{2}$$

Figure 1 shows typical two-wing butterfly attractors in different phase planes for system (1) with $a = 0.6$ and initial conditions $x_1(0) = x_2(0) = x_3(0) = 1$. The corresponding Lyapunov exponents are:

$$LE_1 = 0.191212, LE_2 \approx 0, LE_3 = -0.799337, \tag{3}$$

and the corresponding Kaplan-Yorke (or Lyapunov) dimension $D_{KY} = 2.239$. By incorporating a state variable x_4 into the first equation of system (1) with a feedback strategy, we derive a four-dimensional (4D) dynamic system:

$$\begin{cases} \dfrac{dx_1}{dt} = a(-x_1 + x_2) + x_4 \\ \dfrac{dx_2}{dt} = -x_3\mathrm{sgn}(x_1) \\ \dfrac{dx_3}{dt} = |x_1| - 1 \\ \dfrac{dx_4}{dt} = -bx_1 \end{cases} \tag{4}$$

Here b is the new control parameter. Using a coupling strategy by adding state variables x_5 and x_6, a six-dimensional (6D) hyperchaotic model is constructed, described as follows:

$$\begin{cases} \dfrac{dx_1}{dt} = a(-x_1 + x_2) + x_4 \\ \dfrac{dx_2}{dt} = -x_3\mathrm{sgn}(x_1) \\ \dfrac{dx_3}{dt} = -1 + |x_1| \\ \dfrac{dx_4}{dt} = -bx_1 \\ \dfrac{dx_5}{dt} = -x_5 + x_1x_4 \\ \dfrac{dx_6}{dt} = -x_6 + x_1x_3 \end{cases} \tag{5}$$

The resulting new 6D model of the dynamic system contains only 11 terms, which is one less than in recent paper (Al-Talib and Al-Azzawi 2023b). In addition to the linear terms, system (5) includes two nonlinearities (x_1x_4, x_1x_3) and two functions: $\mathrm{sgn}(x_1)$ and $|x_1|$. In this paper, we found that system (5) is hyperchaotic when the system parameters take the values $a = 0.77$ and $b = 0.45$. For these parameter values and initial conditions (ICs)

$$x_1(0) = x_2(0) = x_3(0) = x_4(0) = x_5(0) = x_6(0) = 1, \tag{6}$$

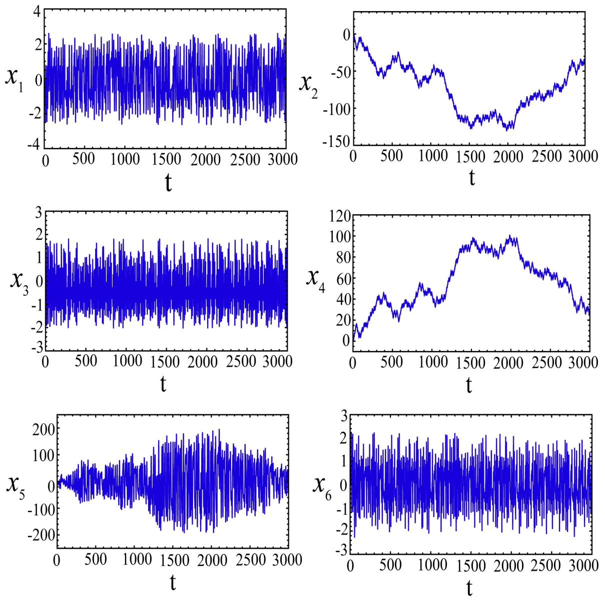


Figure 2 Temporal diagrams for variables $x_1, x_2, x_3, x_4, x_5, x_6$.

all Lyapunov exponents of the new system (5) were calculated in the following form:

$$LE_1 = 0.13238,\ LE_2 = 0.01280,\ LE_3 = 0.00580 \approx 0,$$

$$LE_4 = -0.88594,\ LE_5 = -1.01495,\ LE_6 = -1.02017. \tag{7}$$

It is also of interest to obtain time series data for the new 6D dynamic system (5) with ICs (6). In the context of dynamic systems, time series data reflects the behavior or evolution of a system over time. Time series analysis can be used to study the state variables of the new model x_i $(i = (1, 2, 3, 4, 5, 6))$ over time, as shown in Figure 2. Here, the random nature of the dependence of the variables x_i on time t is clearly visible.

Next, we start the dynamic analysis of the recently introduced systems (5).

DYNAMICAL ANALYSIS

In this section, we explore some fundamental dynamic properties of the new proposed 6D system.

Symmetry and dissipativity of the system

It is easy to verify that system (5) satisfies the following coordinate transformation **T**:

$$\mathbf{T}:\ (x_1, x_2, x_3, x_4, x_5, x_6) \to (-x_1, -x_2, x_3, -x_4, x_5, -x_6).$$

This means that each trajectory is symmetrical about x_3 and x_5 axises, and the system (5) is invariant for a given transformation **T**.

The divergence of the vector field $\mathbf{\Phi}(\dot{x}_1, \dot{x}_2, \dot{x}_3, \dot{x}_4, \dot{x}_5, \dot{x}_6)$ of the system (5) can be calculated as:

$$\mathrm{div}\mathbf{\Phi} = \frac{\partial \dot{x}_1}{\partial x_1} + \frac{\partial \dot{x}_2}{\partial x_2} + \frac{\partial \dot{x}_3}{\partial x_3} + \frac{\partial \dot{x}_4}{\partial x_4} + \frac{\partial \dot{x}_5}{\partial x_5} + \frac{\partial \dot{x}_6}{\partial x_6} = -(a+2) < 0,$$

$$\dot{x}_i \equiv \frac{dx_i}{dt},\ i = (1, 2, 3, 4, 5, 6). \tag{8}$$

Table 2 Lyapunov exponents for different values of the parameter a.

a	Lyapunov Exponents $(LE_1, LE_2, LE_3, LE_4, LE_5, LE_6)$	Signs	Behavior
0.005	**0.0340**,-0.0048,-0.0025,-0.0147,-1.00094,-1.0159	(0,-,-,-,-,-)	Periodic
0.15	**0.0199**,-0.0016,-0.0897,-0.0680,-1.0020,-1.0083	(0,-,-,-,-,-)	Periodic
0.3	0.0913,**0.0020**,**-0.0017**,-0.3713,-1.0116,-1.0086	(+,0,0,-,-,-)	Chaotic 2-torus
0.6	0.1484,0.0111,**0.0076**,-0.7418,-1.0150,-1.0103	(+,+,0,-,-,-)	Hyperchaotic
0.75	0.1367,**0.0014**,**-0.0005**,-0.8653,-1.0074,-1.0148	(+,0,0,-,-,-)	Chaotic 2-torus
0.77	0.1323,0.0127,**0.0058**,-0.8859,-1.0149,-1.0201	(+,+,0,-,-,-)	Hyperchaotic
1.5	**0.0085**,-0.0155,-0.0235,-0.9894,-1.0088,-1.4712	(0,-,-,-,-,-)	Periodic

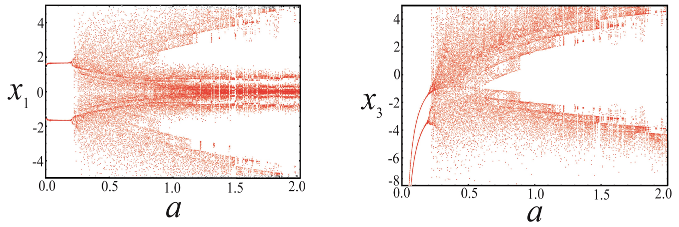


Figure 3 Bifurcation diagrams for x_1, x_3 components of the system (5).

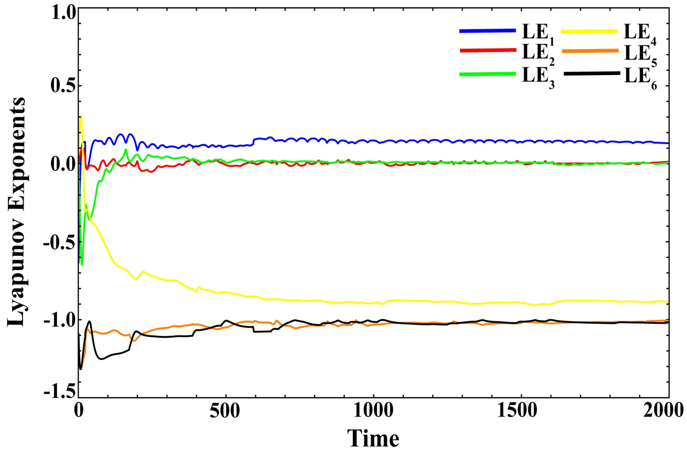


Figure 4 Lyapunov exponents for system (5).

According to Liouville's theorem, the phase volume V changes over time as follows:

$$\frac{dV}{dt} = \int ... \int \operatorname{div}\mathbf{\Phi} dx_1 dx_2 dx_3 dx_4 dx_5 dx_6 = (-(a+2))V(t). \quad (9)$$

In this case, the phase volume exponentially diminishes to zero as time t approaches infinity: $V(t) = V(0)\exp((-(a+2)t)$. As a result, system (5) is dissipative, allowing for the emergence of attracting sets, or attractors.

Equilibrium points

The equilibrium states of a dynamic system (5) are found from the left-hand sides of the equations by setting $\dot{x}_1 = \dot{x}_2 = \dot{x}_3 = \dot{x}_4 = \dot{x}_5 = \dot{x}_6 = 0$:

$$\begin{cases} 0 = a(-\tilde{x}_1 + \tilde{x}_2) + \tilde{x}_4 \\ 0 = -\tilde{x}_3 \operatorname{sgn}(\tilde{x}_1) \\ 0 = -1 + |\tilde{x}_1| \\ 0 = -b\tilde{x}_1 \\ 0 = -\tilde{x}_5 + \tilde{x}_1\tilde{x}_4 \\ 0 = -\tilde{x}_6 + \tilde{x}_1\tilde{x}_3 \end{cases} \quad (10)$$

Solving the equations (10) under the assumption that a and b are non-zero parameters results in $x_1 = 0$ from the fourth equation. Substituting this value into the third equation produces a contradiction $-1 = 0$, indicating the absence of equilibrium points in the system. Consequently, all attractors generated by system (5) are considered hidden attractors.

Bifurcation diagrams, analysis of Lyapunov exponents and dimension

Bifurcation diagrams represent changes in state variables of nonlinear dynamic systems graphically. They provide insights into qualitative changes as control parameters are adjusted. We use Mathematica software to solve the equations outlined in (5) with the initial conditions from (6). In our analysis, we control the parameter a in the system (5), while keeping the parameter b fixed at $b = 0.45$. Figure 3 displays bifurcation diagrams for the x_1 and x_3 components of the system (5) as a varies within the interval $a \in [0, 2]$. These diagrams help identify stable regions and regular behaviors (represented by individual points) within the system. They can also indicate areas where the system exhibits periodic or quasi-periodic behavior. Each branch in the diagram may correspond to different periodic orbits, reflecting various vibration modes. Additionally, bifurcation diagrams can illustrate period-doubling bifurcations as the parameter a changes. These bifurcations represent a sequence in which the system transitions from one periodic state to a period-doubling state, which can continue and may ultimately lead to chaotic behavior. As shown in the bifurcation diagram in Figure 3, there are two branches of regular oscillations: a lower (left) branch and an upper (right) branch. The left periodic attractor undergoes a period-doubling bifurcation, and a similar bifurcation occurs for the right attractor at the same value of a. In other words, the left and right attractors are mirror images of each other.

As is known, Lyapunov exponents (LEs) are an important criterion for describing the behavior and stability of dynamic systems. LEs characterize the rate of divergence or convergence of neighboring trajectories in a dynamic system. A dynamic system is assumed to be unstable or exhibit chaotic behavior when the LE is positive, and a negative exponent indicates a tendency toward stable equilibrium. Thus, by examining the sign of the LEs, one can classify the system's behavior as regular, quasi-regular (2-torus, 3-torus), chaotic, or hyperchaotic. The number of Lyapunov exponents corresponds to the dimensionality of the dynamical system. In the case of our system (5), there are six such indicators. Following the methodology (Binous and Zakia 2008), we computed LEs for specific values of parameter a at fixed parameter $b = 0.45$ and ICs (6). Lyapunov exponents offer deep insights into how the system's dynamic behavior evolves with changes in the parameter a. According to Table 2, the dynamical behaviors of system (5) can be categorized into the following groups based on the Lyapunov exponents. In the future, we will be interested in the hyperchaotic behavior of the system (5) for the parameter value $a = 0.77$. In this case, the sum of all six Lyapunov exponents is $L_1 + L_2 + L_3 + L_4 + L_5 + L_6 = -2.77 < 0$. This suggests that system (5) exhibits dissipative behavior (see, for example, (Kozlovska *et al.* 2024)). It is easy to verify that a hyperchaotic system (5) at parameters $a = 0.77, b = 0.45$ satisfies the condition (Singh and Roy 2016):

$$\sum_{i=1}^{6} LE_i = \mathrm{div}\boldsymbol{\Phi} = -2.77. \tag{11}$$

Figure 4 illustrates the dynamics of the Lyapunov exponents ().

One of the most frequently used characteristics in the numerical modeling of dynamic systems is the Lyapunov dimension, proposed by Kaplan and Yorke (Fredrickson *et al.* 1983). The Lyapunov dimension helps to identify the fractal dimension of a chaotic system, which is a measure of the complexity and entanglement of the system's attractor. Higher Lyapunov dimensions typically indicate more complex systems. For convenience, let us present the spectrum of the Lyapunov exponents in descending order: $LE_1 > LE_2 > LE_3 > LE_4 > LE_5 > LE_6$ and calculate the Lyapunov Kaplan-Yorki dimension according to the following formula:

$$D_{KY} = \xi + \frac{1}{|LE_{\xi+1}|}\sum_{i=1}^{\xi} LE_i = 3 + \frac{0.1509}{0.8859} \approx 3.17, \tag{12}$$

where ξ is determined from the conditions

$$\sum_{i=1}^{\xi} LE_i > 0 \;\Rightarrow\; \sum_{i=1}^{3} LE_i = 0.1509, \quad \sum_{i=1}^{\xi+1} LE_i = -0.735 < 0.$$

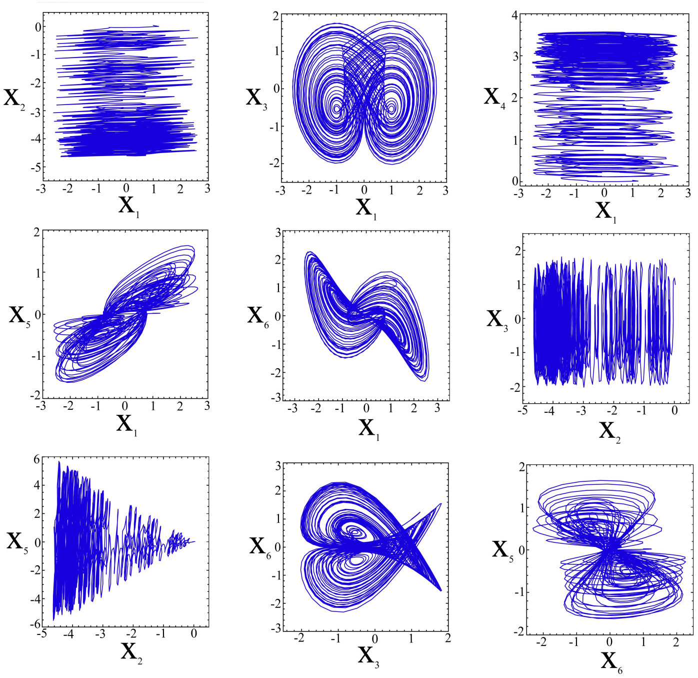


Figure 5 Hidden attractors of the new 6D rescaled system (13) in different planes.

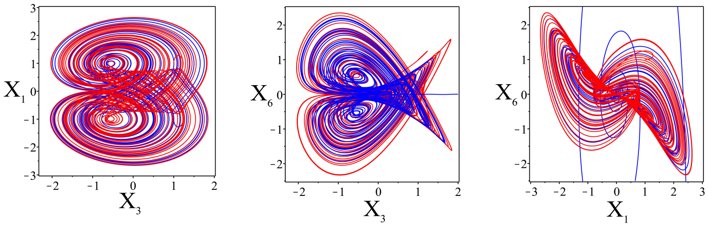


Figure 6 Plots demonstrating the multistability of two attractors in different phase planes with two different ICs given in Table 3.

Here ξ is the number of first non-negative exponents Lyapunov in the spectrum. From (12), it is evident that the Lyapunov dimension is fractal, indicating that the trajectories of system (5) exhibit complex behavior.

Visualizing phase portraits in the rescaled 6D dynamic system

As mentioned in the previous subsection, the dynamic system (5) can display hyperchaotic behavior. This makes the visual analysis of phase portraits for hyperchaotic attractors especially insightful. It is easy to see from Figure 2 that the temporal diagrams for the variables x_1, x_2, x_3, x_4, x_5, and x_6 exhibit an aperiodic structure, a defining feature of chaotic systems. Implementing the hyperchaotic system (5) in an electronic circuit presents challenges, as the dynamic variables x_2, x_4, and x_5 exceed the operational amplifiers' power supply limits. To address this, we scale the variables in the system () by setting $x_2 = 30X_2, x_4 = 30X_4$, and $x_5 = 40X_5$, while keeping $x_1 = X_1, x_3 = X_3$, and $x_6 = X_6$. This transformation results in the modified form of the hyperchaotic system (5) and ICs (6) in the following form:

Table 3 Multistability of the system (13) with fixed parameters a=0.77, b=0.45 and various ICs.

Figure planes	Initial Conditions	Color	Sign of LEs
X_3X_1	$\left(1, \frac{1}{30}, 1, \frac{1}{30}, \frac{1}{40}, 1\right)$	red	(+,+,0,-,-,-) hyperchaotic
X_3X_1	$\left(-1, -\frac{1}{30}, 1, -\frac{1}{30}, -\frac{1}{40}, 1\right)$	blue	(+,+,0,-,-,-) hyperchaotic
X_3X_6	$\left(1, \frac{1}{30}, 1, \frac{1}{30}, \frac{1}{40}, 1\right)$	red	(+,+,0,-,-,-) hyperchaotic
X_3X_6	$\left(5.2, -\frac{1}{3}, 3.5, \frac{1}{30}, -\frac{1}{40}, 1\right)$	blue	(+,+,0,-,-,-) hyperchaotic
X_1X_6	$\left(1, \frac{1}{30}, 1, \frac{1}{30}, \frac{1}{40}, 1\right)$	red	(+,+,0,-,-,-) hyperchaotic
X_1X_6	$\left(-5.2, -\frac{1}{3}, 3.5, -\frac{1}{30}, \frac{1}{40}, 1\right)$	blue	(+,+,0,-,-,-) hyperchaotic

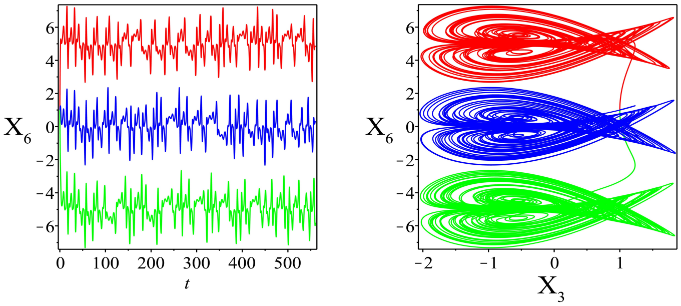


Figure 7 Signal X_6 and phase portrait in the plane X_3X_6 for different values of the offset boosting controller k: $k = 0$ (blue), $k = 5$ (green), $k = -5$ (red).

$$\begin{cases} \dfrac{dX_1}{dt} = a(-X_1 + 30X_2) + 30X_4 \\ \dfrac{dX_2}{dt} = -\dfrac{X_3}{30}\operatorname{sgn}(X_1) \\ \dfrac{dX_3}{dt} = -1 + |X_1| \\ \dfrac{dX_4}{dt} = -\dfrac{b}{30}X_1 \\ \dfrac{dX_5}{dt} = -X_5 + \dfrac{3}{4}X_1X_4 \\ \dfrac{dX_6}{dt} = -X_6 + X_1X_3 \end{cases} \tag{13}$$

$$X_1(0) = 1,\ X_2(0) = 1/30,\ X_3(0) = 1,\ X_4(0) = 1/30, \\ X_5(0) = 1/40,\ X_6(0) = 1\,. \tag{14}$$

Systems (5) and (13) are equivalent, as the linear transformation adjusts the variables without affecting the intrinsic properties of the nonlinear system. Figure 5 illustrates the solutions of the transformed equations (13) given the initial conditions () and parameter values $a = 0.77, b = 0.45$. The phase portraits show hidden attractors in the different planes. Notably, the dynamic range of the variables x_2, x_4, x_5 is considerably reduced compared to Figure 2. This reduction facilitates the practical implementation of electronic circuits using operational amplifiers, which operate within the typical voltage limits of -15V to +15V.

Multistability and offset boosting control

Dynamic systems, as mathematical constructs used to describe complex phenomena across various scientific fields, can possess multiple attractors. These attractors, which may be points, cycles, tori, or more complex chaotic structures, represent distinct states of the system. The specific attractor to which a system converges depends on its initial conditions, meaning small changes in these conditions can lead to different long-term behaviors. This leads to the concept of multistability, where several attractors coexist within the same set of system parameters.

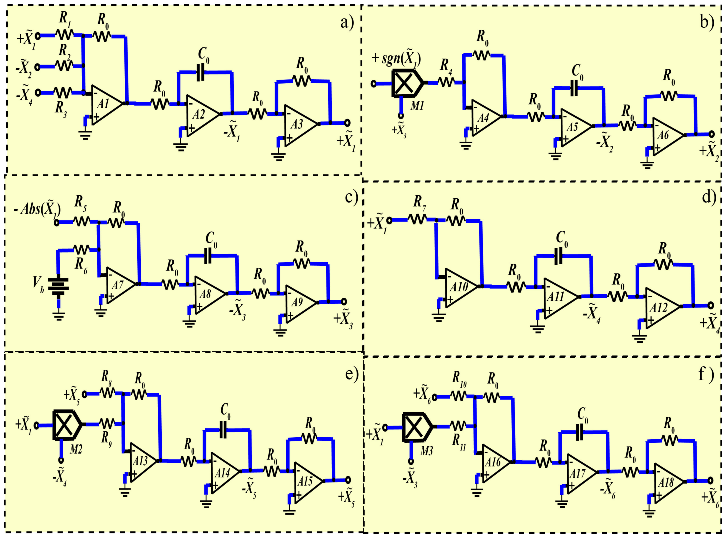


Figure 8 Circuit modules implemented based on a system of equations (13): a) $\widetilde{X}_1$, b) $\widetilde{X}_2$, c) $\widetilde{X}_3$, d) $\widetilde{X}_4$, e) $\widetilde{X}_5$, f) $\widetilde{X}_6$.

In this subsection, we examine the multistability property of system (5), demonstrating how different attractors can coexist under the same system parameters when initial conditions are varied. Table 3 provides data for two attractors obtained by solving system (13) with identical control parameters $a = 0.77, b = 0.45$ but different initial conditions. Figure 6 clearly illustrates the behavior of these two attractors based on the data from Table 3.

Offset boosting control has numerous applications in hyperchaotic systems. This method allows for flexible shifting of the attractor in a specific direction by introducing an offset, which holds significant engineering application value (Wen *et al.* 2021). Note that the state variable X_6 appears only in the sixth equation of the proposed system, making it easy to control. Consequently, the X_6 variable can be increased by introducing a k offset boosting controller, replacing X_6 with $X_6 + k$. The sixth differential equation

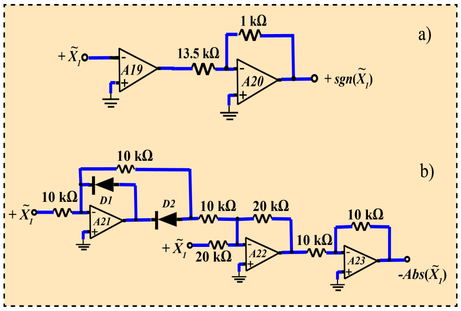


Figure 9 Schematic diagrams for the implementation of functions: a) signum sgn$(\cdot)$; b) absolute value $|\cdot|$.

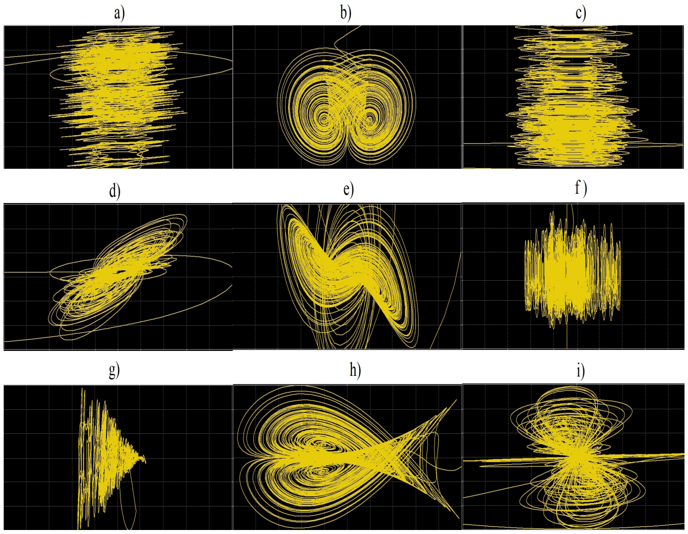


Figure 10 Chaotic phase trajectories of a electronic circuit (Fig. 8) displayed in Multisim oscilloscopes: a) $\widetilde{X}_1\widetilde{X}_2$, b) $\widetilde{X}_1\widetilde{X}_3$, c) $\widetilde{X}_1\widetilde{X}_4$, d) $\widetilde{X}_1\widetilde{X}_5$, e) $\widetilde{X}_1\widetilde{X}_6$, f) $\widetilde{X}_2\widetilde{X}_3$, g) $\widetilde{X}_2\widetilde{X}_5$, h) $\widetilde{X}_3\widetilde{X}_6$, i) $\widetilde{X}_6\widetilde{X}_5$.

of system (13) can then be rewritten as follows:

$$\frac{dX_6}{dt} = -(X_6 + k) + X_1X_3. \tag{15}$$

Figure 7 depicts several positions of hyperchaotic attractors boosted with different k values in the X_3X_6 plane. As shown on the left side of Figure 7, adjusting the bias gain control k converts the signal X_6 from bipolar to unipolar. For a positive value of k, the attractors are shifted in the negative direction, while for a negative value of k, the attractors are shifted in the positive direction.

CIRCUIT IMPLEMENTATION

For the practical implementation of the proposed new 6D hyperchaotic system (5) (or (13)), circuit modeling must be performed using Multisim software. According to Kirchhoff's law for electrical circuits, we can write the electrical analogue of the system (13) as follows:

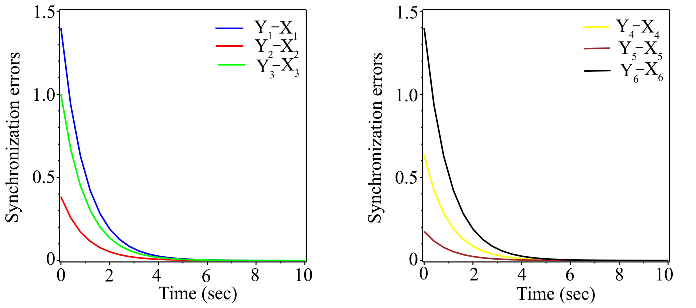


Figure 11 Synchronization error behavior for 6D hyperchaotic drive and response systems.

$$\begin{cases} C_1\dfrac{dU_1}{d\tau} = -\dfrac{U_1}{R_{11}} + \dfrac{U_2}{R_{12}} + \dfrac{U_4}{R_{13}} \\ C_2\dfrac{dU_2}{d\tau} = -\dfrac{U_3\text{sgn}(U_1)}{R_{21}K} \\ C_3\dfrac{dU_3}{d\tau} = \dfrac{|U_1|}{R_{31}} - \dfrac{\widetilde{V}_b}{R_{32}} \\ C_4\dfrac{dU_4}{d\tau} = -\dfrac{U_1}{R_{41}} \\ C_5\dfrac{dU_5}{d\tau} = \dfrac{U_1U_4}{R_{51}K} - \dfrac{U_5}{R_{52}} \\ C_6\dfrac{dU_6}{d\tau} = \dfrac{U_1U_3}{R_{61}K} - \dfrac{U_6}{R_{62}} \end{cases} \tag{16}$$

where $\widetilde{V}_b$ is a stable DC voltage source to implement the constant (=1) in a system (5), R_{ij} are resistors $(i,j) = 1,2,3,4,5,6$, $U_i(\tau)$ are voltage values, C_i are capacitors, and K is a scaling coefficient for the multiplier. We choose the normalized resistor as $R_0 = 100\text{k}\Omega$ and the normalized capacitor as $C_0 = 1\text{nF}$. Then the time constant is equal to $t_0 = R_0C_0 = 10^{-4}\text{s}$. We rescale the state variables of the system (16) as follows $U_1 = U_0\widetilde{X}_1, U_2 = U_0\widetilde{X}_2, U_3 = U_0\widetilde{X}_3, U_4 = U_0\widetilde{X}_4, U_5 = U_0\widetilde{X}_5, U_6 = U_0\widetilde{X}_6, K = U_0K'$, and $\tau = t_0t$. Next, we can write equations (16) in a dimensionless form. By substituting R_0, $C_1 = C_2 = C_3 = C_4 = C_5 = C_6 = C_0$, and $K' = 10$ into (16) and comparing numerical values before the output voltages of the system (13), we get the value of resistors:

$$\begin{cases} \dfrac{d\widetilde{X}_1}{dt} = -\dfrac{100\text{k}}{R_1}\widetilde{X}_1 + \dfrac{100\text{k}}{R_2}\widetilde{X}_2 + \dfrac{100\text{k}}{R_3}\widetilde{X}_4 \\ \dfrac{d\widetilde{X}_2}{dt} = -\dfrac{100\text{k}}{R_4\cdot 10}\widetilde{X}_3\text{sgn}(\widetilde{X}_1) \\ \dfrac{d\widetilde{X}_3}{dt} = \dfrac{100\text{k}}{R_5}|\widetilde{X}_1| - \dfrac{100\text{k}}{R_6}V_b \\ \dfrac{d\widetilde{X}_4}{dt} = -\dfrac{100\text{k}}{R_7}\widetilde{X}_1 \\ \dfrac{d\widetilde{X}_5}{dt} = \dfrac{100\text{k}}{R_9\cdot 10}\widetilde{X}_1\widetilde{X}_4 - \dfrac{100\text{k}}{R_8}\widetilde{X}_5 \\ \dfrac{d\widetilde{X}_6}{dt} = \dfrac{100\text{k}}{R_{11}\cdot 10}\widetilde{X}_1\widetilde{X}_3 - \dfrac{100\text{k}}{R_{10}}\widetilde{X}_6 \end{cases} \tag{17}$$

where

$$R_1 = 129.87\text{k}\Omega,\ R_2 = 4.329\text{k}\Omega,\ R_3 = 3.333\text{k}\Omega,\ R_4 = 300\text{k}\Omega,$$

$$R_5 = R_6 = 100\text{k}\Omega,\ R_7 = 6.666\text{M}\Omega,\ R_8 = 100\text{k}\Omega,$$

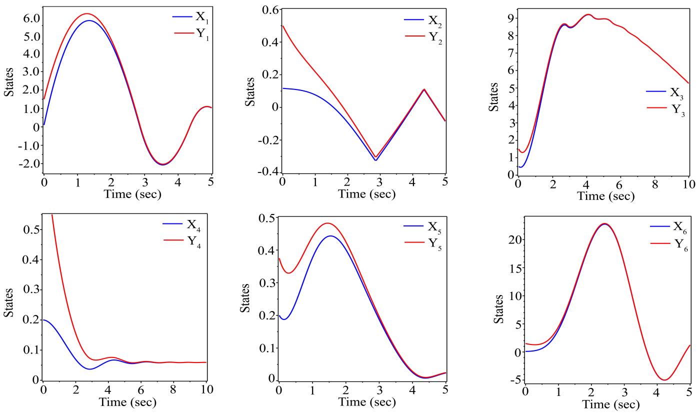


Figure 12 Synchronization of the states for 6D hyperchaotic drive and response systems.

$$R_9 = 13.33\text{k}\Omega,\ R_{10} = 100\text{k}\Omega,\ R_{11} = 10\text{k}\Omega.$$

Figure 8 presents analog circuit modules for the equations of system (17), which consists of standard components such as resistors (R), capacitors (C), diodes D1,D2 (1N4001), multipliers M1-M3 (AD633), operational amplifiers A1-A23 (TL084ACN), and a supply voltage of ±15V. The constant 1 is implemented using a constant voltage source $V_b = 1V$. In the modules shown in Figures 8b and 8c, we used standard electronic circuits that simulate the signum $\text{sgn}(\cdot)$ (see, for example, (Yu *et al.* 2008)) and absolute value $|\cdot|$ functions (Sedra and Smith 1998), which are presented in Figure 9. Figure 10 presents the simulation results from Multisim of an electronic circuit, displaying hyperchaotic attractors of system (17) in various planes. These results align with those from the Mathematica simulation shown in Figure 5, confirming the feasibility of the proposed circuit.

ACTIVE CONTROL SYNCHRONIZATION

Certainly, after developing a new chaotic oscillator based on 6D nonlinear dynamic equations, it is crucial to investigate the synchronization capabilities of this system to ensure its practical applicability. In this section, we examine the active control synchronization of two identical 6D hyperchaotic systems. System (13) was chosen as the drive system, while the response system is described as follows:

$$\begin{cases} \dfrac{dY_1}{dt} = a(-Y_1 + 30Y_2) + 30Y_4 + u_1 \\ \dfrac{dY_2}{dt} = -\dfrac{Y_3}{30}\text{sgn}(Y_1) + u_2 \\ \dfrac{dY_3}{dt} = -1 + |Y_1| + u_3 \\ \dfrac{dY_4}{dt} = -\dfrac{b}{30}Y_1 + u_4 \\ \dfrac{dY_5}{dt} = -Y_5 + \dfrac{3}{4}Y_1Y_4 + u_5 \\ \dfrac{dY_6}{dt} = -Y_6 + Y_1Y_3 + u_6 \end{cases} \tag{18}$$

where $Y_1, Y_2, Y_3, Y_4, Y_5, Y_6$ are the states and $u_1, u_2, u_3, u_4, u_5, u_6, u_7$ are active controllers that we will define later. Our objective is to synchronize the signals of both the drive and response systems, even when their initial conditions differ. The state errors are defined as $e_i(t) = Y_i(t) - X_i(t)$, for $(i = 1, 2, 3, 4, 5, 6)$. By subtracting the drive system (13) from the response system (18), we obtain the error system as follows:

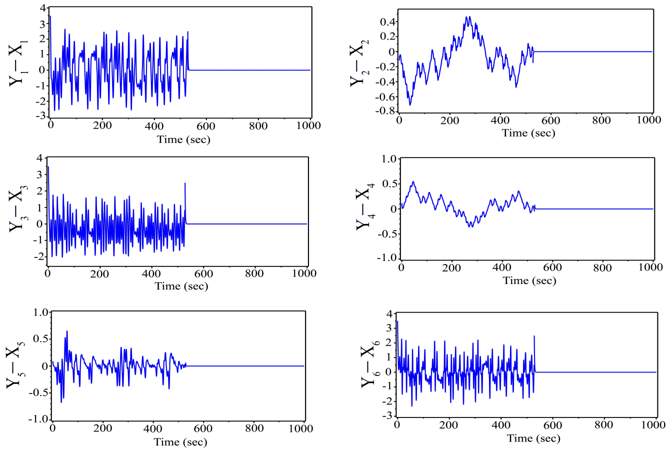


Figure 13 Time evolution of the synchronization errors with controllers deactivated ($t < 520$s) and activated ($t > 520$s).

$$\begin{cases} \dot{e}_1 = a(-e_1 + 30e_2) + 30e_4 + u_1 \\ \dot{e}_2 = -\frac{1}{30}(Y_3\text{sgn}(Y_1) - X_3\text{sgn}(X_1)) + u_2 \\ \dot{e}_3 = |Y_1| - |X_1| + u_3 \\ \dot{e}_4 = -\frac{b}{30}e_1 + u_4 \\ \dot{e}_5 = \frac{3}{4}(Y_1Y_4 - X_1X_4) - e_5 + u_5 \\ \dot{e}_6 = (Y_1Y_3 - X_1X_3) - e_6 + u_6 \end{cases} \tag{19}$$

Next, we define active control functions aimed at producing an asymptotically stable error system, thereby achieving synchronization of the novel 6D hyperchaotic systems. The selected active control functions are detailed below:

$$\begin{cases} u_1 = -e_1 + ae_1 - 30ae_2 - 30e_4 \\ u_2 = -e_2 + \frac{1}{30}(Y_3\text{sgn}(Y_1) - X_3\text{sgn}(X_1)) \\ u_3 = -e_3 - (|Y_1| - |X_1|) \\ u_4 = -e_4 + \frac{b}{30}e_1 \\ u_5 = -\frac{3}{4}(Y_1Y_4 - X_1X_4) \\ u_6 = -(Y_1Y_3 - X_1X_3) \end{cases} \tag{20}$$

Then, the dynamic equations of the error system are as follows:

$$\begin{cases} \dot{e}_1 = -e_1 \\ \dot{e}_2 = -e_2 \\ \dot{e}_3 = -e_3 \\ \dot{e}_4 = -e_4 \\ \dot{e}_5 = -e_5 \\ \dot{e}_6 = -e_6 \end{cases} \tag{21}$$

Upon applying the proposed active control functions (20), the error system transforms into a linear form. For convenience, we express this in matrix form as follows:

$$\begin{vmatrix} \dot{e}_1 \\ \dot{e}_2 \\ \dot{e}_3 \\ \dot{e}_4 \\ \dot{e}_5 \\ \dot{e}_6 \end{vmatrix} = \begin{vmatrix} -1 & 0 & 0 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 0 & 0 & -1 \end{vmatrix} \begin{vmatrix} e_1 \\ \dot{e}_2 \\ e_3 \\ e_4 \\ e_5 \\ e_6 \end{vmatrix} \tag{22}$$

It can be easily verified that all eigenvalues of the state matrix (22) are negative. Therefore, according to the Routh-Hurwitz criterion, the error system is stable, ensuring synchronization between the drive system (13) and the response system (18).

Numerical simulation

The nonlinear equations (13) and (18) were solved using the 4th-5th order Runge-Kutta-Fehlberg (rkf45) method in the Maple computing environment with the fixed parameters $a = 0.77, b = 0.45$. The drive system (13) was initialized with the following conditions:

$$X_1(0) = \frac{1}{10}, \; X_2(0) = \frac{7}{60}, \; X_3(0) = \frac{1}{2}, \; X_4(0) = \frac{1}{5}, \\ X_5(0) = \frac{1}{5}, \; X_6(0) = \frac{1}{10}, \tag{23}$$

and the response system was initialized with:

$$Y_1(0) = \frac{3}{2}, \; Y_2(0) = \frac{1}{2}, \; Y_3(0) = \frac{3}{2}, \; Y_4(0) = \frac{5}{6}, \\ Y_5(0) = \frac{3}{8}, \; Y_6(0) = \frac{3}{2}. \tag{24}$$

Figure 11 illustrates the error curves resulting from the synchronization between the drive and response systems, showcasing the exponential convergence of synchronization errors e_i to zero over time. In Figure 12, the behavior of each state in both the drive and response systems is depicted, demonstrating the convergence of trajectories within a short time and indicating synchronization in these hyperchaotic systems.

For a clear representation of synchronization using the active control method, we select the following initial conditions for the drive system (13) and response system (18):

$$X_1(0) = 1. \; X_2(0) = \frac{1}{30}, \; X_3(0) = 1, \; X_4(0) = \frac{1}{30}, \\ X_5(0) = \frac{1}{40}, \; X_6(0) = 1, \\ Y_1(0) = \frac{7}{2}, \; Y_2(0) = -\frac{7}{60}, \; Y_3(0) = \frac{7}{2}, \; Y_4(0) = \frac{7}{60}, \\ Y_5(0) = \frac{7}{80}, \; Y_6(0) = \frac{7}{2}. \tag{25}$$

Ensure that the active controllers are switched on at $t = 520$ seconds. The results depicted in Figure 13 indicate that the error system states exhibit chaotic behavior over time when the active controllers are deactivated (at $t < 520$ s), suggesting a lack of synchronization. At $t \geq 520$ s, the controllers are activated, and we can see that the synchronization error states quickly converge to zero.

Thus, simulation findings demonstrate the ability of the active controllers (20) to synchronize two identical 6D hyperchaotic systems starting from various initial conditions.

CONCLUSION

This work obtained a new 6D dynamic system with the smallest number of terms (only 11) compared to the existing 6D dynamic systems in the literature (see Tab. 1). It was found that the new dynamic system has no equilibrium points, which may lead to the formation of hidden attractors. For specific values of the system parameters, a hyperchaos regime was discovered, for which all Lyapunov exponents and the Kaplan-York dimension were calculated. The presence of two positive Lyapunov exponents indicates the complexity of the new 6D dynamic system. In addition, extensive studies of the dynamic properties of the system were carried out, including bifurcation diagrams, phase portraits, Lyapunov exponents, multistability, and offset boosting control. The electronic circuit of the proposed 6D system was designed using the Multisim 14 software. The results of the electronic circuit simulation are consistent with those obtained in the Mathematica environment. Finally, the synchronization between the two identical new 6D hyperchaotic systems was achieved by developing appropriate active controllers. The new system has promising applications in the field of encryption and decryption of information signals.

Acknowledgments

We thank three anonymous reviewers for their valuable suggestions and comments.

Availability of data and material

Not applicable.

Conflicts of interest

The authors declare that there is no conflict of interest regarding the publication of this paper.

Ethical standard

The authors have no relevant financial or non-financial interests to disclose.

LITERATURE CITED

Adıyaman, Y., S. Emiroğlu, M. K. Uçar, and M. Yıldız, 2020 Dynamical analysis, electronic circuit design and control application of a different chaotic system. Chaos Theory and Applications **2**: 10–16.

Al-Azzawi, S. F. and A. S. Al-Obeidi, 2021 Chaos synchronization in a new 6d hyperchaotic system with self-excited attractors and seventeen terms. Asian-European Journal of Mathematics **14**: 2150085.

Al-Azzawi, S. F. and A. S. Al-Obeidi, 2023 Dynamical analysis and anti-synchronization of a new 6d model with self-excited attractors. Applied Mathematics-A Journal of Chinese Universities **38**: 27–43.

Al-Azzawi, S. F. and A. M. Hasan, 2023 New 5d hyperchaotic system derived from the sprott c system: Properties and anti synchronization. Journal of Intelligent Systems and Control **2**: 110–122.

Al-Obeidi, A. S. and S. F. Al-Azzawi, 2022 A novel six-dimensional hyperchaotic system with self-excited attractors and its chaos synchronisation. International Journal of Computing Science and Mathematics **15**: 72–84.

Al-Talib, Z. S. and S. F. Al-Azzawi, 2022 A new simple 6d hyperchaotic system with nonhyperbolic equilibrium and its electronic circuit. In 2022 Int. Conf. Computer Sci. Software Engineering (CSASE) pp. 369–374.

Al-Talib, Z. S. and S. F. Al-Azzawi, 2023a A new simple 6d hyperchaotic system with hyperbolic equilibrium and its electronic circuit. Iraqi Journal for Computer Science and Mathematics **4**: 155–166.

Al-Talib, Z. S. and S. F. Al-Azzawi, 2023b A new simple 6d hyperchaotic system with hyperbolic equilibrium and its electronic circuit. Iraqi Journal For Computer Science and Mathematics **4**: 155–166.

Aziz, S. M. and S. F. Al-Azzawi, 2022 A novel simple 6d hyperchaotic system with hidden attractors. In 2022 Int. Conf. Computer Sci. Software Engineering (CSASE) pp. 7–12.

Benkouider, K., T. Bouden, M. E. Yalcin, and S. Vaidyanathan, 2020 A new family of 5d, 6d, 7d and 8d hyperchaotic systems from the 4d hyperchaotic vaidyanathan system, the dynamic analysis of the 8d hyperchaotic system with six positive lyapunov exponents and an application to secure communication design. International Journal of Modelling, Identification and Control **35**: 241–257.

Bhat, M. A. and M. Shikha, 2019 Complete synchronisation of non-identical fractional order hyperchaotic systems using active control. International Journal of Automation and Control **13**: 140–157.

Binous, H. and N. Zakia, 2008 An improved method for lyapunov exponents computation. https://library.wolfram.com/infocenter/MathSource/7109/ .

Bohr, T., M. H. Jensen, G. Paladin, and A. Vulpiani, 1998 *Dynamical Systems Approach to Turbulence*. Cambridge Nonlinear Science Series, Cambridge University Press.

Chen, A., J. Lu, J. Lu, and S. Yu, 2006 Generating hyperchaotic lu attractor via state feedback control. Physica A **364**: 103–110.

Chu, J. and W. W. Hu, 2016 Control chaos for permanent magnet synchronous motor base on adaptive backstepping of error compensation. International Journal of Automation and Computing **9**: 163–174.

Elwakil, A. S., S. Ozoguz, and M. P. Kennedy, 2002 Creation of a complex butterfly attractor using a novel lorenz-type system. IEEE Transactions on Circuits and Systems I **49**: 527–530.

Emiroglu, S., A. Akgül, Y. Adı yaman, T. E. Gümüš, Y. Uyaroglu, *et al.*, 2022 A new hyperchaotic system from t chaotic system: dynamical analysis, circuit implementation, control and synchronization. Circuit World **48**: 265–277.

Frederickson, P., J. L. Kaplan, E. D. Yorke, and J. A. Yorke, 1983 The liapunov dimension of strange attractors. Journal of differential equations **92**: 185–207.

Ghosh, D. and S. Bhattacharya, 2010 Projective synchronization of new hyperchaotic system with fully unknown parameters. Nonlinear Dynamics **61**: 11–21.

Hu, G., 2009 Generating hyperchaotic attractors with three positive lyapunov exponents via state feedback control. International Journal of Bifurcation and Chaos **19**: 651–660.

Jia, Q., 2007 Hyperchaos generated from the lorenz chaotic system and its control. Physics Letters A **366**: 217–222.

Jung, W., S. J. Elliot, and J. Cheer, 2019 Local active control of road noise inside a vehicle. Mechanical Systems and Signal Processing **121**: 144–157.

Khattar, D., N. Agrawal, and M. Sirohi, 2024 Qualitative analysis of a new 6d hyper-chaotic system via bifurcation, the poincare notion, and its circuit implementation. Indian Journal of Physics **98**: 259–273.

Kopp, M. I., A. V. Tur, and V. V. Yanovsky, 2023 Chaotic dynamics of magnetic fields generated by thermomagnetic instability in a nonuniformly rotating electrically conductive fluid. Journal of Physical Studies **27**: 2403.

Kozlovska, O., F. Sadyrbaev, and I. I. Samuilik, 2024 A new 3d chaotic attractor in gene regulatory network. Mathematics **12**: 100.

Li, X., 2009 Modified projective synchronization of a new hyperchaotic system via nonlinear control. Communications in Theoretical Physics **52**: 274–278.

Lorenz, E. N., 1963 Deterministic nonperiodic flow. Journal of atmospheric sciences **20**: 130–141.

Michael Kopp and Andrii Kopp, 2022 A new 6d chaotic generator: Computer modelling and circuit design. International Journal of Engineering and Technology Innovation **12**: 288–307.

Rajagopal, K., L. Guessas, S. Vaidyanathan, A. Karthikeyan, and A. Srinivasan, 2017a Dynamical analysis and fpga implementation of a novel hyperchaotic system and its synchronization using adaptive sliding mode control and genetically optimized pid control. Mathematical Problems in Engineering **2017**: 1–14.

Rajagopal, K., G. Laarem, A. Karthikeyan, and A. Srinivasan, 2017b Fpga implementation of adaptive sliding mode control and genetically optimized pid control for fractional-order induction motor system with uncertain load. Advances in Difference Equations **2017**: 1–20.

Ramakrishnan, R., 2018 *Chaos and its applications to Communication Systems*. Scholars' Press, Cambridge.

Sabaghian, A., S. Balochian, and M. Yaghoobi, 2020 Synchronisation of 6d hyper-chaotic system with unknown parameters in the presence of disturbance and parametric uncertainty with unknown bounds. Connection Science **32**: 362–383.

Sedra, A. S. and K. C. Smith, 1998 *Microelectronics Circuits, 4th ed*. Oxford University Press, New York.

Singh, J. P. and B. K. Roy, 2016 The nature of lyapunov exponents is (+, +, -, -), is it a hyperchaotic system? Chaos, Solitons & Fractals **92**: 73–85.

Soldatenko, S., A. Bogomolov, and A. Ronzhin, 2021 Mathematical modelling of climate change and variability in the context of outdoor ergonomics. Mathematics **9**.

Tohidi, S., Y. Yildiz, and I. Kolmanovsky, 2020 Adaptive state observers for incrementally quadratic nonlinear systems with application to chaos synchronization. Automatica **121**: 1–11.

Vaidyanathan, S., 2013 A ten-term novel 4d hyperchaotic system with three quadratic nonlinearities and its control. International Journal of Control Theory and Applications **6**: 97–109.

Vaidyanathan, S. and C. K. Volos, 2015 Analysis and adaptive control of a novel 3-d conservative no-equilibrium chaotic system. Archives of Control Sciences **25**: 333–353.

Vaidyanathan, S., C. K. Volos, and V. T. Pham, 2014 Hyperchaos, adaptive control and synchronization of a novel 5-d hyperchaotic system with three positive lyapunov exponents and its spice implementation. Archives of Control Sciences **24**: 409–446.

Wang, J. and Z. Chen, 2008 A novel hyperchaotic system and its complex dynamics. International Journal of Bifurcation and Chaos **18**: 3309–3324.

Wen, J., Y. Feng, X. Tao, , and Y. Cao, 2021 Dynamical analysis of a new chaotic system: Hidden attractor, coexisting-attractors, offset boosting, and dsp realization. IEEE Access **9**: 167920–167927.

Yang, L., Q. Yang, and G. Chen, 2020 Hidden attractors, singularly degenerate heteroclinic orbits, multistability and physical realization of a new 6d hyperchaotic system. Communications in Nonlinear Science and Numerical Simulation **90**: 105362.

Yang, Q. and C. Chen, 2013 A 5d hyperchaotic system with three positive lyapunov exponents coined. International Journal of

Bifurcation and Chaos **23**: 1350109.

Yang, Q., D. Zhu, and L. Yang, 2018 A new 7d hyperchaotic system with five positive lyapunov exponents coined. International Journal of Bifurcation and Chaos **28**: 1850057.

Yin, X., J. Chen, W. Yu, Y. Huang, W. Wei, *et al.*, 2022 Five-dimensional memristive hopfield neural network dynamics analysis and its application in secure communication. Circuit World **50**: 67–81.

Yousefpour, A., A. H. Hosseinloo, M. R. H. Yazdi, and A. Bahrami, 2020 Disturbance observer-based terminal sliding mode control for effective performance of a nonlinear vibration energy harvester. Journal of Intelligent Material Systems and Structures **31**: 1495–1510.

Yu, S., W. K. S. Tang, J. Lu, and G. Chen, 2008 Multi-wing butterfly attractors from the modified lorenz systems. 2008 IEEE International Symposium on Circuits and Systems (ISCAS), Seattle, WA, USA pp. 768–771.

Zhang, H., W. Zhang, Y. Zhao, and M. Ji, 2020 Adaptive state observers for incrementally quadratic nonlinear systems with application to chaos synchronization. Circuits, Systems, and Signal Processing **39**: 1290–1306.

How to cite this article: Kopp, M. I., and Samuilik, I. A New 6D Two-wing Hyperhaotic System: Dynamical Analysis, Circuit Design, and Sinchronization. *Chaos Theory and Applications*, 6(4), 273-283, 2024.

CHAOS Theory and Applications in Applied Sciences and Engineering

e-ISSN: 2687-4539
RESEARCH ARTICLE
Vol.6 / No.4 / 2024 / pp.284-293
https://doi.org/10.51537/chaos.1517688

A New Data Coding Algorithm for Secure Communication of Image

Hidayet Oğraş*

*Department of Electronics Communication Technology, Batman University, Central Campus, Batman, Turkiye.

ABSTRACT This paper proposes a new entropy-sensitive based data coding algorithm for the secure communication of image information between transceiver systems. The proposed algorithm utilizes chaos theory and the image information content of the reference image to create sensitivity on the decoding side for a high level of secrecy. It successfully recovers secret images at the receiver's side using secret code series derived from both the secret and reference images, instead of direct transmission of secret image. The image information can be retrieved only through the same reference image, the same system parameters and identical code series using the proper decoding technique at the receiver. Quantitative results indicate that the average coding time for 128x128 images is approximately 0.27 seconds, while the extraction time averages 0.19 seconds, yielding impressive rates of 0.487 Mbps and 0.677 Mbps, respectively. Moreover, according to qualitative results, even a single-bit change in the reference image leads to a complete inability to decode the secret image, highlighting the robustness and security of the algorithm. Experimental results on various images show that the proposed algorithm is reliable, fast and effective in securing confidential image information.

KEYWORDS
Chaos theory
Logistic map
Data coding
Image entropy

INTRODUCTION

Today, with the rapid development of data communication techniques and network technologies, many multimedia data have been transmitted and shared over open networks such as internet. Most of this information transmitted or shared over public networks need to be protected privately. Cryptography is a common approach used for the protection of data security by making the original message to unintelligible form (Sharma *et al.* 2022; Gurunathan and Rajagopalan 2020). However, one of the greatest challenges in cryptography is the secure generation, distribution and storage of the keys used in encryption and decryption processes (Rana *et al.* 2023). Cryptography has no ability to protect against vulnerabilities and threats resulting from poor design of systems, protocols and algorithms. Strongly encrypted, authentic and digitally signed information can sometimes be difficult to access, even for an authorized user. Moreover, creating suitable secret keys that meet sufficient security conditions is not easy in terms of circuit complexity, resource and time costs (Rana *et al.* 2023; Abba *et al.* 2024).

Another approach that can be used in secure communication is steganography. In steganography the message is hidden into another medium in a way that cannot be noticed by eavesdropper during communication (Jaradat *et al.* 2021; Mishra and Bhanodiya 2015). One of the major disadvantages of steganography is that there is a large overhead for hiding a very small amount of information. It provides communication secrecy with limited data capacity. Digital multimedia data such as images, audio and video are used as the cover medium where confidential information can be hidden (Pradhan *et al.* 2018). Among them, image files are very ideal as cover media due to the having large amount of redundant space (Ogras 2019). Image data hiding techniques can be classified into two important types: Spatial domain and Transform domain. Spatial domain techniques deal with image pixels and all of them directly replace some bits in pixel values while hiding the data (Hussain and Hussain 2013). Most commonly used technique in spatial domain is Least-Significant-Bit (LSB) embedding in which the secret message inside an image by replacing least significant bit of cover image with the bits of message to be hidden (Abba *et al.* 2024). For instance, LSB technique embeds a secret message into the cover image with one bit, so the modified pixel value is increased or decreased by 1 according to the used algorithm.

However, even if there is a tinny change in pixel values, total difference can be easily detected statistically with known analysis methods without much effort and so the secret information can be easily revealed. Another popular spatial domain method used in image steganography is the pixel value differencing (PVD) technique in which secret data is embedded into a cover image depend-

Manuscript received: 17 July 2024,
Revised: 14 October 2024,
Accepted: 21 November 2024.

[1] hidayet.ogras@batman.edu.tr (**Corresponding author**).

ing on pixel neighbourhood differences. Some studies (Hosam and Ben Halima 2016; Swain 2018a,b) based on PVD methods are proposed in image steganography. However, both techniques are not strong against some steganographic attacks as PVD can be detected by pixel difference histogram analysis.

A good data hiding algorithms aims at two important purposes: Payload capacity and imperceptibility. Payload capacity means the maximum amount of secret information that can be hidden inside a cover medium (Jaradat *et al.* 2021). Imperceptibility refers the visual quality of a stego image. After hiding secret message into a cover medium such as an image, the visual quality will decrease compared to cover image that results a slight distortion on the stego image. This distortion should be at an unnoticeable level; otherwise the risk of confidential information being detected will increase (Huang and Wang 2020). If zero distortion occurs as a result of such data coding process, then this will be interpreted as maximum imperceptibility, hence no analysis methods can be applied to detect secret information.

Many existing studies in the field of secure image communication rely heavily on traditional cryptographic techniques or steganographic methods that often introduce additional complexities, such as encryption algorithms or compression schemes. These approaches can lead to increased processing times and power consumption, making them less suitable for real-time applications. In contrast, chaotic systems have gained substantial interest in secure communication due to their inherent unpredictability, sensitivity to initial conditions and complex dynamic behavior (Liu *et al.* 2022). In such systems, slight changes in initial parameters can lead to significantly different outcomes, a characteristic often described as the "butterfly effect" (Bonny and Al Nassan 2024). This sensitivity makes chaotic systems an ideal tool for secure data transmission, as even a minor deviation in transmission parameters can prevent unauthorized decoding attempts, which is a key strength of chaos-based communication systems in ensuring high security. These systems are also advantageous for real-time applications due to their computational efficiency, relying on simple mathematical operations compared to traditional security methods (Khan and Waseem 2024). This enables fast encoding and decoding, supporting secure, high-speed image transmission without requiring heavy computational resources. The proposed algorithm demonstrates this efficiency with low coding and decoding times, making it suitable for real-time image transmission.

The application of chaos theory in secure communication benefits from the complex, non-linear behavior of chaotic systems to create robust encoding mechanisms, particularly for applications involving image data (Zhang and Liu 2023). For instance, chaotic maps such as the Logistic, Henon, and Tent maps are widely used due to their simplicity and effectiveness in generating high-entropy sequences. These maps can efficiently scramble image data, ensuring that the encrypted data appears random to unauthorized observers. Because chaotic maps require only a few key parameters to function, they simplify key management in encryption systems, reducing the complexity associated with traditional secret key distribution. Unlike traditional encryption methods, chaotic systems do not depend on large keys for security; instead, they use a set of initial parameters and iterative processes to generate unpredictable sequences. This property makes chaos-based encryption systems less susceptible to brute-force attacks and highly efficient in terms of computational resources (Khan and Waseem 2024). For image data in particular, chaos theory provides distinct advantages, as the encoding process can utilize the image's inherent complexity (entropy), making the encoded message directly tied to the image data. In this study, the entropy of a reference image is used along with chaos-based coding, which provides a higher level of security, as both the reference image and chaotic map parameters are essential for decoding. On the other hand, chaotic systems also present certain challenges in secure communication applications. One concern is the accurate reproduction of chaotic sequences at the decoding end, which requires that both the sender and receiver systems be perfectly synchronized (Liu *et al.* 2022). Any mismatch in parameters could cause the chaotic system to diverge, leading to decoding errors. Additionally, chaotic systems can sometimes exhibit periodic behavior, especially if the system parameter is not properly selected, which may compromise their unpredictability.

This paper introduces a new entropy-sensitive data coding algorithm that addresses these shortcomings by using a unique combination of chaos theory and reference image entropy without altering the reference image. The main contributions of this research include:

- The proposed method enhances security by requiring the correct reference image and codes for recovering secret images, which makes unauthorized access difficult.
- The high sensitivity of the system to even very small changes prevents successful decoding.
- The algorithm minimizes processing times, making it suitable for real-time applications while maintaining the integrity of the reference image and ensuring robustness against detection methods.

The structure of the paper is as follows: first, the definition of chaos and the chaotic map utilized in the algorithm are introduced. Next, the proposed coding and decoding algorithms are described in detail. Then, experimental results, including the running time of the coding and decoding processes, along with evaluation comparisons of the proposed algorithm, are given. Finally, the paper concludes with a summary of findings.

CHAOS

Chaos means a state of total confusion with disorder (Ahmad and Shin 2021). All systems that contain chaos exhibit extreme sensitivity to initial condition and control parameters (Ozkaynak 2020; Effah-Poku *et al.* 2018). Although chaotic systems have deterministic structure, their long-term behavior cannot be predicted (Umoh and Wudil 2016). Hence, these properties can be used to transform obvious events into an irregular and unpredictable form in some engineering fields such as cryptography (Roy *et al.* 2021; Gafsi *et al.* 2020; Irsan and Antoro 2019) and secure communication (Sharafi *et al.* 2021; Kumar and Raghava 2019; Ismail *et al.* 2020; Oğraş and Türk 2013). In this paper, the well-known chaotic Logistic map (LM) is used to generate secret key for the proposed coding algorithm.

Logistic map

LM is a simple but frequently used system for generating pseudo-random sequences. It has a simple iterative structure having a dynamical equation as in Equation 1:

$$x_{n+1} = r \cdot x_n \cdot (1 - x_n) \qquad (1)$$

where r refers to the control parameter of the map defined in (0,4]. If the r is between 3.57 and 4, then the map behaves chaotically, exhibiting chaotic properties such as non-periodicity and sensitivity to initial conditions. In this case, the output is distributed randomly within the range from 0 to 1. As a result, the LM can produce unpredictable series like a key generator, which can be used in fields of subjects where the randomness is needed. For instance, the output generated from the LM with $x_0 = 0.1234$ and $r = 3.9999$ after 500 iterations is shown in Figure 1.

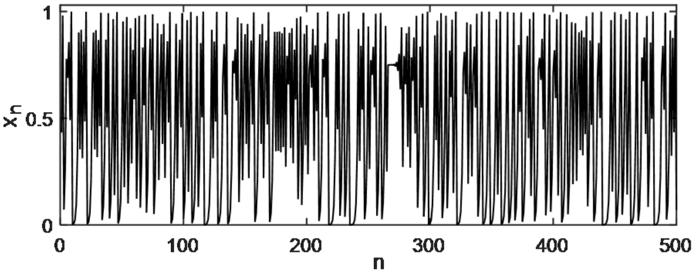


Figure 1 The output of chaotic LM

PROPOSED CODING ALGORITHM

Generating secret key and coding process

In this study, no data containing of the secret image information is sent directly to the receiver. Instead, unlike the cryptographic approach, secret image is mixed with a series of codes obtained from the reference image and the LM system through the proposed coding algorithm. The reference image in here is an arbitrary public image with which the secret image will be associated randomly. The secret image is a target image that will be decoded at the receiver side by being sensitively related to the reference image along with the secret code series. The coding process is carried out by associating the secret image with the secret codes using the reference image. The general block scheme of the proposed algorithm is shown in Figure 2

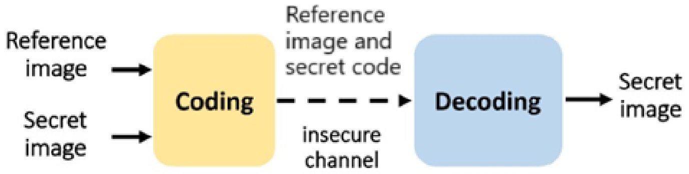


Figure 2 Block diagram of a proposed scheme

The algorithm uses a series of high-precision secret codes to decode the secret image. In addition to the parameters of the LM system, the reference image also affects the secret key over its entropy value as in Equation 2.

$$x_0' = (x_0 + \text{Entropy}(\text{reference\_img})) \mod 1 \tag{2}$$

x_0 and x_0' indicate the first and actual initial values of the LM, respectively. The equation introduces a perturbation to the initial value of the LM using the entropy of an image. Since entropy measures the amount of information or randomness in an image, this introduces variability based on the content of the image. The result is that x_0' will be slightly different from x_0 depending on the image characteristics. The use of the modulus operation ensures that x_0' remains within the range $[0, 1]$. This is crucial for LM, which typically require initial values in this interval. It is well known that chaotic systems are extremely sensitive to initial values. As a result of such an approach, the generated key series will be depended on the first initial value of the LM system and the entropy of the reference image with high sensitivity. This sensitivity on the coding side will also be reflected on the decoding side, which will play a major role in correctly decoding the encoded secret image information. Binary series are generated from the LM by using a simple mathematical transformation given in Equation 3.

$$b_n = \begin{cases} 1 & \text{if } x_n \geq 0.5 \\ 0 & \text{otherwise} \end{cases} \tag{3}$$

A simple bit is obtained for each corresponding of x_n value. Notice that, the orders of the generated series are completely dependent on the initial condition and control parameter of the LM. Then, bitstreams of reference image are mixed with the generated binary series according to the bitstreams of the secret image. The whole processes for the generation of the secret codes are explained in detail as follows:

Step 1) A secret image to be coded has the same size with the reference image selected and then it is converted to serial bitstream format.

Step 2) According to the values in the serial bitstream, XOR (Exclusively-OR) or XNOR (not XOR) operations are performed up to the length of the bitstream for 1 and 0, respectively.

Step 3) Chaotic binary series are generated by the LM with a chosen control parameter, initial condition and the entropy value from the reference image.

Step 4) Reference image is reshaped to the length of size and then converted to serial bitstream like secret image.

Step 5) According to the bitstreams value of the secret image, if '1' occurs, XOR operation is performed with chaotic series and the bitstreams of the reference image. Otherwise, XNOR operation is performed.

Step 6) Step 5 is continued by considering all of the bitstream values for the secret image. Finally, the output represents modified chaotic series that will be used as secret codes in the proposed algorithm.

Schematic diagram for generating secret code is shown in the Figure 3. In the algorithm, size of the secret image is same as the reference image which results maximum payload capacity indeed. Block diagram of the coding process is shown in Figure 4.

Decoding process

On the receiver side, a secret image can be recovered with the same reference image, identical secret key and correct system parameters of the LM. If there is a slight change in any one of them, the secret image will not be decoded correctly, even 1-bit change occurs in the reference image. Block diagram for the decoding process is shown in Figure 5.

All system parameters of the LM must be unknown except the receiving side. Keeping these values secret plays an important key role in integrity for decoding the secret image. In addition, the secret codes must be transmitted through a channel to the receiving side knowing which reference image is associated with the secret image. As a result, the secret image can be correctly decoded through the inverse algorithm of the proposed scheme. Schematic diagram of the inverse algorithm for decoding process is shown in the Figure 6.

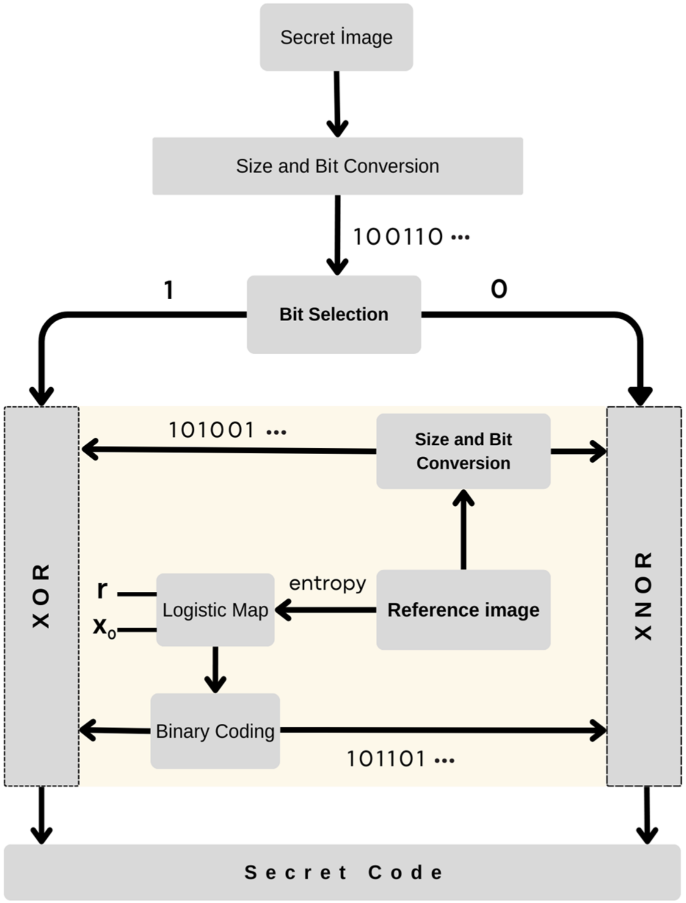


Figure 3 Schematic diagram for the generating secret code

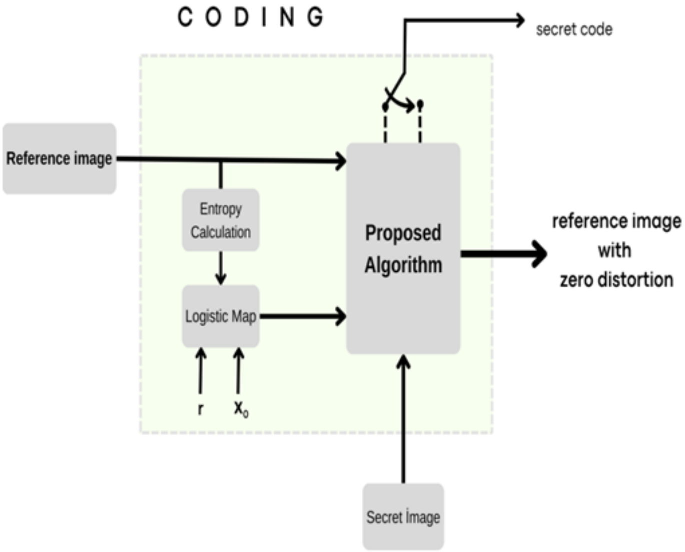


Figure 4 Block diagram of the coding process

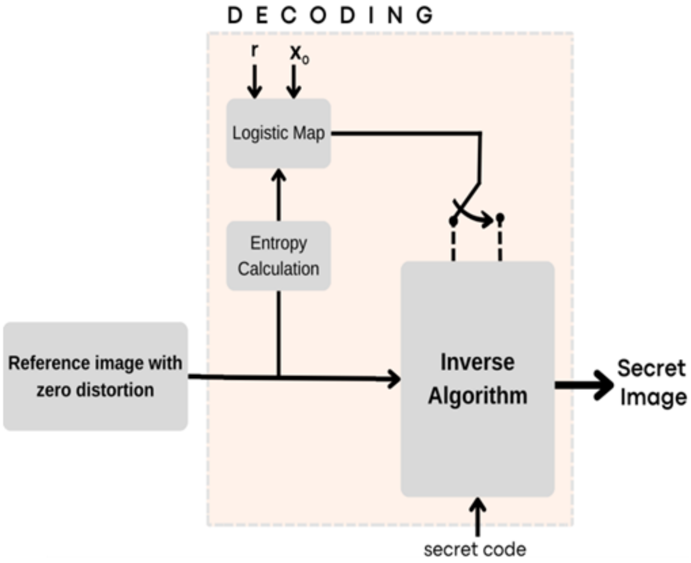


Figure 5 Block diagram of the decoding process

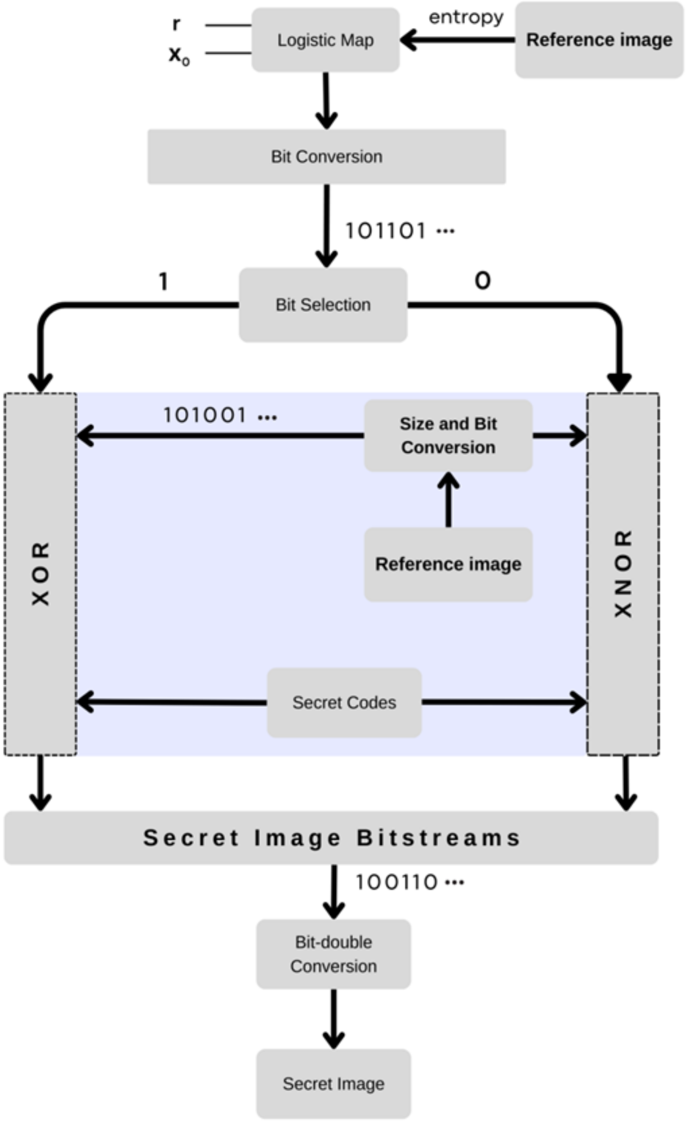


Figure 6 Schematic diagram of the inverse algorithm

EXPERIMENTAL RESULTS

Twelve grey test images are used to evaluate the performance of the proposed coding algorithm. The properties of the images are listed in Table 1, and some are shown in Figure 7.

To evaluate the performance of the proposed algorithm, sufficient number of simulation is performed with various test images by using Matlab software. The visual analysis results, as well as the running speed and decoding rate of the algorithm, are calculated using an Intel Core i3 2.13 GHz processor with 4 GB of RAM.

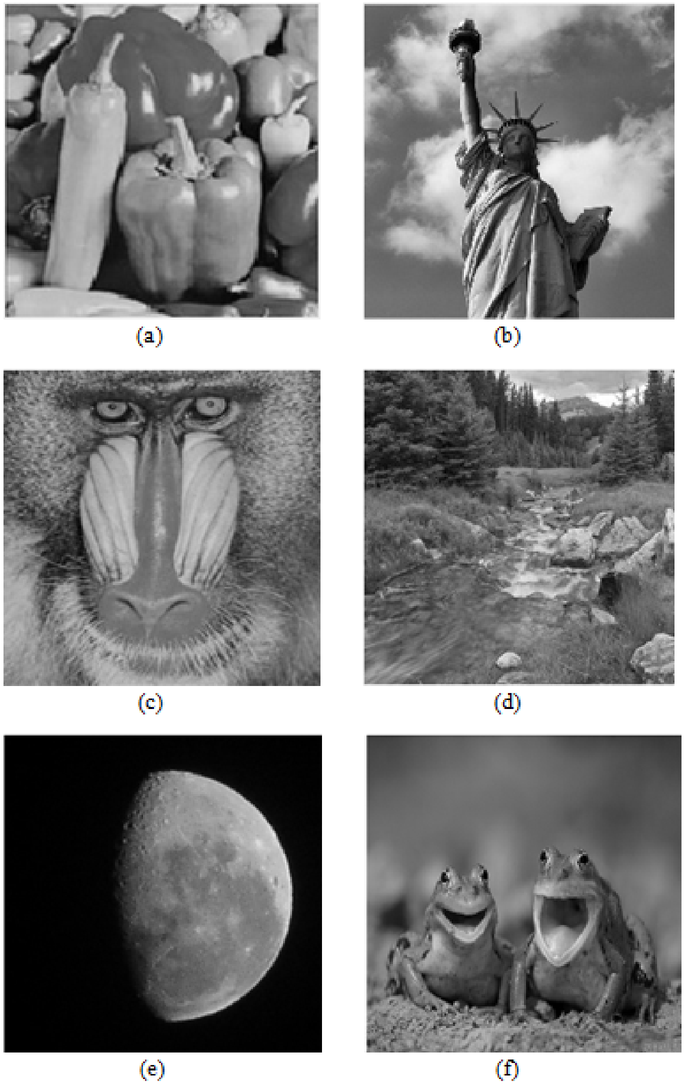


Figure 7 Some test images used in the proposed algorithm (a) Peppers (b) Liberty (c) Baboon (d) Landscape (e) Moon (f) Frog

Sensitivity analysis

In the proposed scheme, changing the entropy value of the reference image directly affects the initial value of the LM which results significant change in secret codes due to the chaos effect. Therefore, exactly the same reference image must be used in the receiver side to decode the secret image. The reference image is associated with the secret image at bit-level by using Bitwise operations and that correlation can only be appeared with the same secret key used at the receiver side. Furthermore, the secret image can only be detected under exactly the same reference image due to the entropy sensitivity of the algorithm. If a different reference image is used in the receiver side, then the secret image should not be decoded correctly. For this case, a test image of "Frog.jpg" is used as secret image and "Moon.jpg" image is used as reference image in the proposed algorithm. LM parameters are selected as $r = 4$ and $x_0 = 0.12345$. Then, on the receiver side, the secret image is tried to be decoded with the same LM parameters but using "Landscape.jpg" as reference image. The visual result is shown in Figure 8. According to the results, the secret image can only be decoded correctly for the associated correct reference image. If there is even a 1-bit pixel change in the correct reference image, the secret image should not still be decoded correctly. The secret "Frog" image is tried to be decoded by making only 1-bit change in the middle pixel of the correct reference image in Figure 8(c). In other words, the middle pixel value of 121 is changed to 122 in correct "Moon" image. The analysis result is shown in Figure 9.

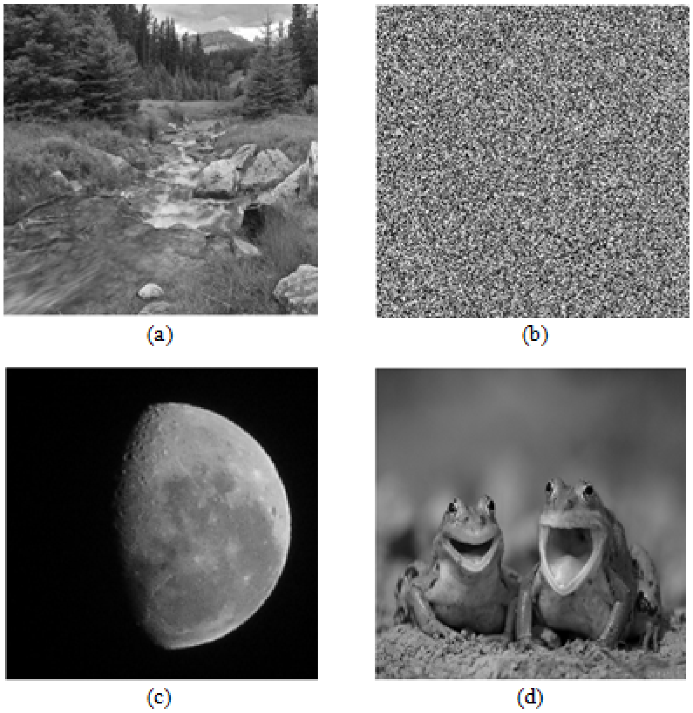


Figure 8 Reference image sensitivity analysis (a) Incorrect reference image (b) Incorrectly decoded image (c) Correct reference image (d) Correctly decoded image

In this analysis, the entropy of the correct reference image is 5.35447619; for the other one is 5.35447489 where 1-bit change is made in the middle pixel value. The results of the reference image sensitivity analyses show that the proposed algorithm is highly sensitive to the reference image even with only a single pixel change on it. In security manner, the proposed algorithm should also be sensitive to the system parameters of r and x_0. For this analysis, "Baboon" image is used as a secret image and "Liberty" image is used as a reference image. "Baboon" and "Liberty" images are used in the proposed algorithm for the parameters of $r = 4$ and $x_0 = 0.12345$. On the receiving side, a tinny change of 10^{-6} is applied to the one of the parameters while other remains same, and performs the inverse algorithm of the proposed scheme to decode secret image. The results are shown in Figure 10. According to the results, the proposed algorithm is highly sensitive to the all system

Table 1 Entropy values of different test images

Size	Name	Purpose of usage	Entropy
128x128	Flowers	Secret	7.44118
128x128	Frog	Reference	6.92897
256x256	Cameraman	Reference	7.10514
256x256	Frog	Secret	6.96951
256x256	Moon	Reference	5.35447
256x256	Baboon	Secret	7.22002
256x256	Landscape	Reference	7.34015
512x512	Baboon	Secret	7.18316
512x512	Liberty	Reference	7.49462
512x512	Peppers	Reference	7.59386
1024x1024	Cat	Secret	7.20682
1024x1024	Airplane	Reference	7.20493

parameters which enhance the security of the secret image.

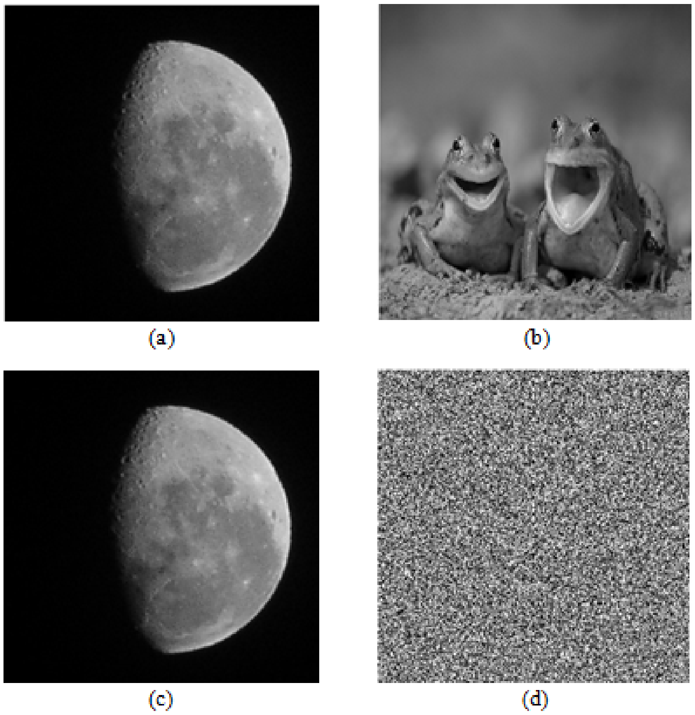


Figure 9 Entropy sensitivity analysis (a) Correct reference image (b) Correct decoded image (c) Similar reference image with only 1-bit change in the middle pixel value (d) Incorrectly decoded image

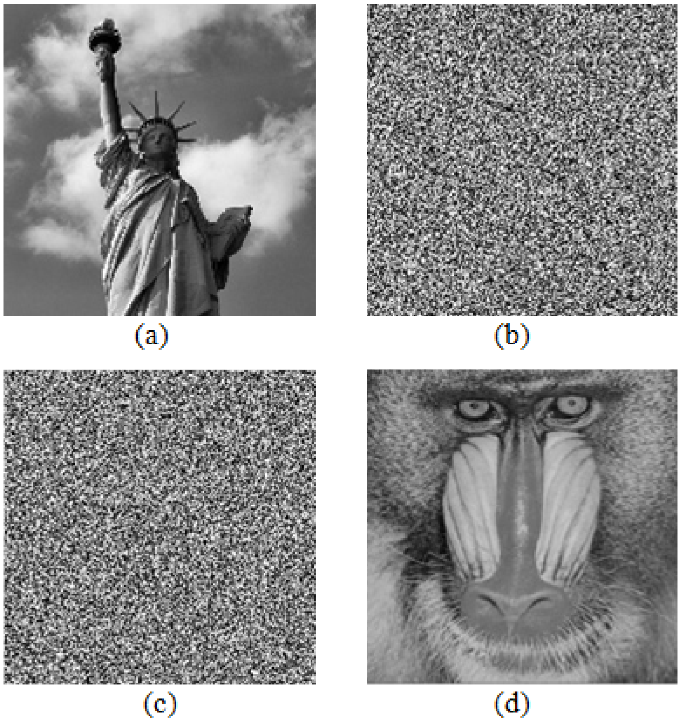


Figure 10 Parameters sensitivity analysis (a) Reference image coding with $r = 4$ and $x_0 = 0.12345$ (b) Incorrect decoded image using with $r = 4$ and $x_0 = 0.123451$ (c) Incorrect decoded image using with $r = 3.999999$ and $x_0 = 0.12345$ (d) Correct decoded image with $r = 4$ and $x_0 = 0.12345$

Entropy attack analysis

The entropy of an image measures the level of randomness or information content in that image, and is calculated as in Equation 4.

$$H = -\sum_{i=0}^{L-1} p(i)\log_2(p(i)) \tag{4}$$

where H is the entropy value, L is the total number of levels in an image and p(i) is the probability of the i-th level. In the proposed algorithm, the entropy value is utilized to directly modify the initial value of the LM, thereby affecting the generation of the chaotic binary sequence. However, it is important to note that even if the entropy values are the same, the pixel-level distribution and arrangement of bits within the image may still differ, which might still affect the encoding and decoding process. Moreover, the entropy value alone does not guarantee that the images are identical in terms of their content or structure and even slight differences in their pixel values can lead to variations in the generated secret codes. To perform an entropy attack analysis, the proposed algorithm uses two different images with the same entropy values. The Arnold Cat Map (ACM) method is then applied to permute a reference test image, generating a scrambled image with identical entropy. ACM is a method used in image processing to scramble the pixel positions of an image (Bhardwaj and Bhagat 2018). It is a transformation technique, as defined in Equation 5, which rearranges the pixel locations while preserving the image's overall properties, such as entropy and pixel values.

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} 1 & 1 \\ 1 & 2 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} \mod N \tag{5}$$

(x, y) and (x', y') represent the original coordinates of a pixel and the new coordinates after the transformation, respectively. N denotes the size of the image. For the entropy analysis, the test image of 'cameraman.jpg' with entropy value 7.10514 is chosen as reference image to be used for the secure transmission of 'baboon.jpg' image in the proposed scheme. During the decoding process of the secret image on the receiver side, the scrambled image obtained by applying the ACM method to the 'cameraman' image just one iteration is used. This leads to no change in entropy, and the visual results are shown in Figure 11.

The correlation coefficient between the reference image and the scrambled image is found to be 0.0529, while the correlation coefficient between the secret image and the decoded image is calculated as -0.0337. According to the entropy results, the secret image can only be decoded correctly if the exact same reference image, secret key and system parameters are used. The decoding process is highly sensitive to the reference image's entropy; even if a different image has the same entropy value, incorrect decoding may still occur because the actual pixel values can differ. This result verifies the sensitivity of the proposed algorithm to both the entropy and the precise structure of the reference image.

Differential analysis

NPCR (Number of Pixels Change Rate) and UACI (Unified Average Changing Intensity) are two metrics used to evaluate the differential analyses for the sensitivity of particular algorithms, especially in the context of image encryption (Louzzani *et al.* 2021). They are defined in Equations 6 and 7.

$$\text{NPCR} = \left(\frac{N_{\text{changed}}}{N_{\text{total}}}\right) \times 100 \tag{6}$$

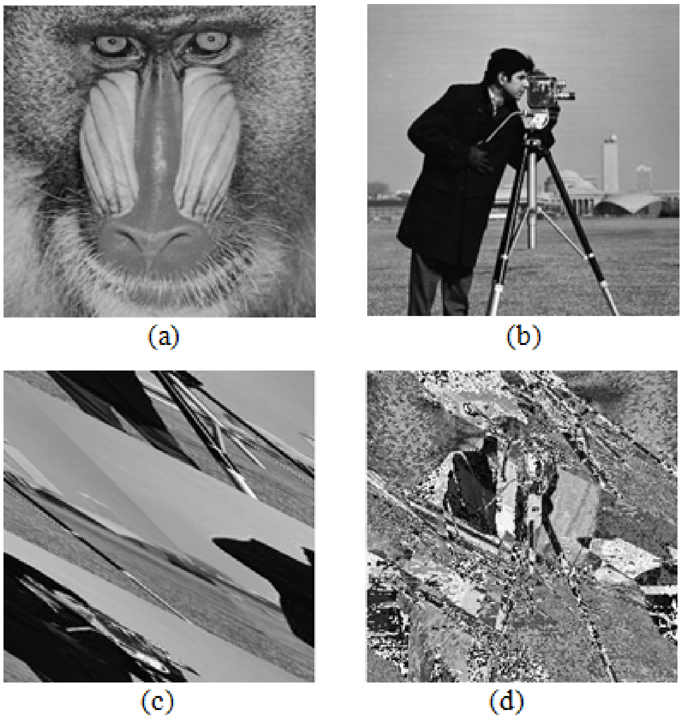


Figure 11 Entropy attack analysis (a) Secret image (b) Reference image (c) Scrambled image (d) Incorrectly decoded secret image

N_{changed} is the number of pixels that differ between the original and decoded image after a small change is made to the reference image. N_{total} is the total number of pixels in the secret image. A higher NPCR value, ideally close 100% indicates better sensitivity of the coding algorithm (Alawida 2023). UACI measures the average change intensity of the pixels. It gives an indication of how much the output changes with a small change in the input (Haridas *et al.* 2024).

$$\text{UACI} = \left(\frac{1}{N_{\text{total}}}\sum_{i=1}^{N_{\text{total}}}\frac{|C_1(i) - C_2(i)|}{255}\right) \times 100 \tag{7}$$

$C_1(i)$ and $C_2(i)$ are the pixel values of the original and decoded images, respectively. A higher UACI values, typically around 33%, indicates that the algorithm has strong sensitivity to changes in the input image. The theoretical values for NPCR and UACI are 99.60% and 33.46%, respectively (Jin *et al.* 2024). Random test images of different sizes are chosen to calculate the correlation coefficients, NPCR and UACI values for the proposed coding algorithm. First, all the secret test images are used to perform differential analyses with a key through the algorithm. Then, the pixel value in the middle of each reference image is incremented by one. Afterwards, the modified reference image is applied to the receiver to decode the secret image using the same system parameters. The corresponding results are listed in Table 2.

The results show that both NPCR and UACI values are very close to their ideal values, suggesting that the proposed algorithm is highly sensitive to the reference image. Moreover, the correlation coefficient is very close to 0, which means there is no relationship at all between the original and decoded images.

Table 2 Differential analysis of the proposed coding algorithm

Size	Secret Image	Cover Image	NPCR (%)	UACI (%)	Correlation Coefficient
128x128	Flowers	Frog	99.6459	32.4016	-0.004564
256x256	Baboon	Landscape	99.5986	29.5655	0.006156
512x512	Goldhill	Liberty	99.6208	31.6547	0.000108
1024x1024	Cat	Airplane	99.6023	30.5923	-0.001699

Table 3 Speed analyses of the proposed algorithm

Size of secret image	Average Coding time (sec)	Average Decoding time (sec)	Coding Rate (Mbps)	Decoding Rate (Mbps)
128x128	0.2691	0.1934	0.487	0.677
256x256	0.9380	0.7422	0.558	0.706
512x512	3.8149	2.3508	0.549	0.892
1024x1024	16.3716	10.4283	0.512	0.804

Speed analysis

In order to evaluate the running speed of the coding and decoding processes, 20 test images with different sizes are used in the proposed algorithm. Then average coding and decoding rates are calculated by using Matlab R2015a software. The average running time for the results can be found in Table 3.

Comparison with related works

The proposed coding algorithm can be considered as an alternative approach to steganographic algorithms. Generally, in steganographic algorithms, confidential information is hidden by manipulating the cover image in time or frequency domain. The coding process performed here is the modification of the random-like chaotic series against the secret image through bitwise operations. It is known that a secure cryptographic algorithm is extremely sensitive to system parameters. In the proposed algorithm, whole system sensitivity is achieved by affecting the initial value of the chaotic system by the information content of the reference image.The proposed scheme is compared with the closely related works of (Elshoush *et al.* 2021) and (Bai *et al.* 2017) in the field of zero image steganography. Table 4 summarizes the comparison of proposed scheme with the related works. This study (Elshoush *et al.* 2021), has proposed a zero distortion steganographic method that uses ASCII code matching for the character of secret message in a cover image without changing image size and pixel values. Here, only the positions of the secret message are noted in a mapping table. In extraction process, the ASCII codes of the secret messages are provided from the positions arrays in the cover image.

All related works of (Elshoush *et al.* 2021), (Bai *et al.* 2017) and (Bilal *et al.* 2013), include additional operations such as encryption, compression or mathematical transformation to increase the security of secret data. However, these processes should take complex structure, high power consumption and can increase time duration of hiding and extracting processes. The proposed scheme uses grayscale images for simpler implementation and efficient processing, particularly where color is unnecessary. Unlike methods that depend on color images, which increase complexity and computational demands, this approach omits encryption while maintaining security through chaos theory and inherent sensitivity. This ensures significant output changes from minor input variations, protecting secret images from unauthorized access. By avoiding encryption-related overhead, the scheme achieves high confidentiality, making it ideal for applications where speed and simplicity are essential. For instance, in (Elshoush *et al.* 2021), the secret message size with 2000 Bytes takes 9.5 seconds in “Cat1” image with a size of 200x200. For the proposed scheme, average time of different secret test images with 128x128 in size (16384 Bytes) takes about 0.27 seconds for coding process.

Evaluation of the proposed algorithm

The imperceptibility of a steganographic algorithm is the most important feature to consider. The ability of unnoticed as a first requirement is closely related to the strength of the steganographic algorithm. In the proposed algorithm, payload capacity is same as the volume of the reference image. Reference image can be thought as a cover image in this study. Therefore, larger volume of reference image means more payload capacity. Secret message should not leave a trace on the cover medium; otherwise it can be detected by steganalysis methods. The proposed algorithm makes no any change to the reference image and brings zero distortion, so it has a perfect imperceptibility and robustness. The evaluation of the proposed algorithm is given in Table 5.

Table 4 Comparison of other references

Criteria	Proposed scheme	Elshoush et al.	Bai et al.	Bilal et al.
Usage of RGB Component	Grayscale (8 bits)	All RGB (24 bits)	All RGB (24 bits)	Grayscale (8 bits)
Domain	Spatial	Spatial	Frequency	Spatial and Frequency
Hiding Data Type	Image	Message	Message	Image
Extracting Process	Bit by bit	Byte by byte	DCT Transform	DCT transform
Compression	No	Yes (Huffman algorithm)	No	Yes (Hash function)
Encryption	No	Yes (AES-128)	Yes (AES)	Yes (RSA)
Usage of positions	No	Yes	No	No
Invisibility	No change	No change	No change	Change
Usage of key	Yes	No	No	No
Precision sensitivity	Yes	No	-	Yes

Table 5 Evaluation of the proposed algorithm

Criteria	Evaluation
Invisibility	No change in reference image
Payload Capacity	As the volume of the reference image
Robustness against statistical attacks	Very high (zero distortion)
Independent of file format	Any format readable by Matlab
Unsuspicious files	Very high as no change in image

CONCLUSION

This paper introduces a new data coding algorithm designed to enhance the security of image communication through the innovative application of chaos theory. A key feature of this algorithm is its ability to recover secret images using only a reference image, specific system parameters and identical code series, rather than relying on direct transmission of secret data. This approach not only improves security but also minimizes the risk of unauthorized access to confidential information. Furthermore, the algorithm demonstrates remarkable robustness and security, exhibiting extreme sensitivity to changes in the reference image. Even minor alterations, such as a single-bit change, can lead to complete failure in decoding the secret image. This resilience is supported by metrics like NPCR and UACI, confirming the algorithm's strong defences against potential attacks. Theoretical and experimental results verify that the proposed algorithm is very efficient and usable, demonstrating its capability to securely transmit image data while maintaining rapid processing speeds and high levels of confidentiality. It is part of the future plans to design similar coding algorithms for different types of multimedia data and to use them in practical applications.

Availability of data and material

The data collected in this study are available from the corresponding author upon reasonable request.

Conflicts of interest

The author declares that there is no conflict of interest regarding the publication of this paper.

Ethical standard

The author has no relevant financial or non-financial interests to disclose.

LITERATURE CITED

Abba, A., J. S. Teh, and M. Alawida, 2024 Towards accurate keyspace analysis of chaos-based image ciphers. Multimedia Tools and Applications pp. 1–20.

Ahmad, I. and S. Shin, 2021 A novel hybrid image encryption–compression scheme by combining chaos theory and number theory. Signal Processing: Image Communication **98**: 116418.

Alawida, M., 2023 A novel chaos-based permutation for image encryption. Journal of King Saud University-Computer and Information Sciences **35**: 101595.

Bai, D., X. Chen, and M. Tian, 2017 A satellite communication zero steganography algorithm. Multimedia Tools and Applications **76**: 26447–26462.

Bhardwaj, R. and D. Bhagat, 2018 two level encryption of grey scale image through 2d cellular automata. Procedia Computer Science **125**: 855–861.

Bilal, M., S. Imtiaz, W. Abdul, and S. Ghouzali, 2013 Zero-steganography using dct and spatial domain. In *2013 ACS international conference on computer systems and applications (AICCSA)*, pp. 1–7, IEEE.

Bonny, T. and W. Al Nassan, 2024 Optimizing security and cost efficiency in n-level cascaded chaotic-based secure communication system. Applied System Innovation **7**: 107.

Effah-Poku, S., W. Obeng-Denteh, and I. Dontwi, 2018 A study of chaos in dynamical systems. Journal of Mathematics **2018**: 1808953.

Elshoush, H. T., I. A. Ali, M. M. Mahmoud, and A. Altigani, 2021 A novel approach to information hiding technique using ascii mapping based image steganography. J. Inf. Hiding Multim. Signal Process. **12**: 65–82.

Gafsi, M., N. Abbassi, M. A. Hajjaji, J. Malek, and A. Mtibaa, 2020 Improved chaos-based cryptosystem for medical image encryption and decryption. Scientific Programming **2020**: 6612390.

Gurunathan, K. and S. Rajagopalan, 2020 A stegano-visual cryptography technique for multimedia security. Multimedia Tools and Applications **79**: 3893–3911.

Haridas, T., S. Upasana, G. Vyshnavi, M. S. Krishnan, and S. S. Muni, 2024 Chaos-based audio encryption: Efficacy of 2d and 3d hyperchaotic systems. Franklin Open **8**: 100158.

Hosam, O. and N. Ben Halima, 2016 Adaptive block-based pixel value differencing steganography. Security and Communication Networks **9**: 5036–5050.

Huang, D. and J. Wang, 2020 High-capacity reversible data hiding in encrypted image based on specific encryption process. Signal Processing: Image Communication **80**: 115632.

Hussain, M. and M. Hussain, 2013 A survey of image steganography techniques. International Journal of Advanced Science and Technology **54**: 113–124.

Irsan, M. and S. C. Antoro, 2019 Text encryption algorithm based on chaotic map. In *Journal of Physics: Conference Series*, volume 1341, p. 062023, IOP Publishing.

Ismail, S. M., A. M. Ghidan, and P. W. Zaki, 2020 Novel chaotic random memory indexing steganography on fpga. AEU-International Journal of Electronics and Communications **125**: 153367.

Jaradat, A., E. Taqieddin, and M. Mowafi, 2021 A high-capacity image steganography method using chaotic particle swarm optimization. Security and Communication Networks **2021**: 6679284.

Jin, B., L. Fan, B. Zhang, R. Lei, and L. Liu, 2024 Image encryption hiding algorithm based on digital time-varying delay chaos model and compression sensing technique. Iscience **27**.

Khan, M. and H. M. Waseem, 2024 An efficient confidentiality scheme based on quadratic chaotic map and fibonacci sequence. AIMS Mathematics **9**: 27220–27246.

Kumar, A. and N. Raghava, 2019 Chaos-based steganography technique to secure information and integrity preservation of smart grid readings using wavelet. International Journal of Computers and Applications **44**: 57–63.

Liu, X., C. Li, S. S. Ge, and D. Li, 2022 Time-synchronized control of chaotic systems in secure communication. IEEE Transactions on Circuits and Systems I: Regular Papers **69**: 3748–3761.

Louzzani, N., A. Boukabou, H. Bahi, and A. Boussayoud, 2021 A novel chaos based generating function of the chebyshev polynomials and its applications in image encryption. Chaos, Solitons & Fractals **151**: 111315.

Mishra, R. and P. Bhanodiya, 2015 A review on steganography and cryptography. In *2015 International Conference on Advances in Computer Engineering and Applications*, pp. 119–122, IEEE.

Ogras, H., 2019 An efficient steganography technique for images using chaotic bitstream. International Journal of Computer Network and Information Security **15**: 21.

Oğraş, H. and M. Türk, 2013 Utilizing simulink and matlab graphical user interface in modelling and simulation of chaos-based digital modulation techniques. International Journal of Electrical Engineering Education **50**: 19–33.

Ozkaynak, F., 2020 A novel random number generator based on fractional order chaotic chua system. Elektronika ir Elektrotechnika **26**: 52–57.

Pradhan, A., K. R. Sekhar, and G. Swain, 2018 Digital image steganography using lsb substitution, pvd, and emd. Mathematical Problems in Engineering **2018**: 1804953.

Rana, S., F. K. Parast, B. Kelly, Y. Wang, and K. B. Kent, 2023 A comprehensive survey of cryptography key management systems. Journal of Information Security and Applications **78**: 103607.

Roy, M., S. Chakraborty, and K. Mali, 2021 A chaotic framework and its application in image encryption. Multimedia Tools and Applications **80**: 24069–24110.

Sharafi, J., Y. Khedmati, and M. Shabani, 2021 Image steganography based on a new hybrid chaos map and discrete transforms. Optik **226**: 165492.

Sharma, D. K., N. C. Singh, D. A. Noola, A. N. Doss, and J. Sivakumar, 2022 A review on various cryptographic techniques & algorithms. Materials Today: Proceedings **51**: 104–109.

Swain, G., 2018a Digital image steganography using eight-directional pvd against rs analysis and pdh analysis. Advances in multimedia **2018**: 4847098.

Swain, G., 2018b High capacity image steganography using modified lsb substitution and pvd against pixel difference histogram analysis. Security and communication networks **2018**: 1505896.

Umoh, E. A. and T. Wudil, 2016 Engineering applications of chaos. In *12th International Conference and Exhibition on Power and Telecommunications (ICEPT 2016)*, pp. 39–49.

Zhang, B. and L. Liu, 2023 Chaos-based image encryption: Review, application, and challenges. Mathematics **11**: 2585.

How to cite this article: Ogras, H. A New Data Coding Algorithm for Secure Communication of Image. *Chaos Theory and Applications*, 6(4), 284-293, 2024.