Arduino-based chaotic generator with polar attractor

Hanna Zemlianukhina[1], Roman Voliansky[1], Nina Volianska[2], Yunifa Miftachul Arif[3] and Dmytro Halushko[4]

[1]National Technical University of Ukraine "Igor Sikorsky Kyiv Polytechnic Institute", 37, Prospect Beresteiskyi, Kyiv, 03056, Ukraine
[2]Taras Shevchenko National University of Kyiv, 24, Bohdan Havrylyshyn Str., Kyiv, 04116, Ukraine
[3]Universitas Islam Negeri Maulana Malik Ibrahim Malang, Jl. Gajayana No.50, Dinoyo, Kec. Lowokwaru, Kota Malang, Jawa Timur 65144, Indonesia
[4]Zaporizhzhia National University, 66, Zhukovsky Str., Zaporizhzhia, 69600, Ukraine

Abstract

The paper addresses the design of methodological frameworks to construct a chaotic system with a predefined attractor. These frameworks are based on an algebraic attractor equation, which defines a set of system state coordinates in the phase plane and the interrelations between them. We consider using polar curves equations to describe a system attractor, but it is also possible to use equations in other coordinate systems. In all cases, one should define the interrelations between system state variables and the attractor's coordinates as some algebraic dependencies. Differentiating these dependencies with respect to time yields the equations of motion for a new dynamical system describing motion along the horizontal and vertical axes. Due to the strong dependency between system state variables and attractor coordinates, it is possible to construct a novel dynamical system with separate and interconnected channels. Such systems can be implemented in continuous and discrete time domains using various analog and digital devices and elements. We demonstrate the application of our approach by designing a chaotic system based on the well-known Mackey-Glass equation and a desired cardioid-like attractor implemented on an Arduino Due board. Our studies show hidden chaos in the designed system while the desired attractor is formed.

Keywords

Chaotic system, hidden chaos, polar curve, coordinate transformation, trigonometric functions

1. Introduction

Nowadays, control theory is gaining popularity due to its use in various areas of engineering and scientific applications [1]. It becomes possible due to the significant advances in mathematics, which allow us to construct precise models and algorithms [2]. The main feature of such models, which are based on ordinary differential equations, is the possibility to study both regular and chaotic motions [3] that define the dynamics of various technical systems in different applications.

One of the most important applications is the route planning for autonomous mobile robots, which finds its use to solve different tasks [4]. Depending on the performed tasks, robot motions should be regular periodical and irregular chaotic [5] ones. The last motions are produced using chaotic systems showing complex, unpredictable behaviour that depends heavily on initial conditions [6]. A lot of papers devoted to design and impelemntation of various chaotict systems by using modern microcontrollers and Arduino-based boards are among them [7]. The implemnted in such a way systems have been studied for decades, often through the analysis of their attractor [8].

Most known integer and fractional-order chaotic attractors are self-excited [9]. Their basins of attraction are connected to unstable equilibrium points. This property makes them easier to find. A

CS&SE@SW 2025: 8th Workshop for Young Scientists in Computer Science & Software Engineering, November 21, 2025, Kryvyi Rih, Ukraine
https://doi.org/10.55056/cssesw2025/paper34
annzemlya@gmail.com (H. Zemlianukhina); volianskyi.roman@lll.kpi.ua (R. Voliansky); ninavolianska@gmail.com (N. Volianska); yunif4@ti.uin-malang.ac.id (Y. M. Arif); dmitriy.galushko99@gmail.com (D. Halushko)
https://epa.kpi.ua/department/staff/zemlyanukhina/ (H. Zemlianukhina)
0000-0002-9653-8416 (H. Zemlianukhina); 0000-0001-5674-7646 (R. Voliansky); 0000-0001-5996-2341 (N. Volianska); 0000-0002-2183-0762 (Y. M. Arif); 0009-0008-9451-9343 (D. Halushko)

small disturbance near an unstable equilibrium typically excites the attractor. As a result, they are well understood and widely used in modeling and design.

In contrast, hidden attractors are much harder to detect [10]. Their basins of attraction are not connected to any equilibrium, and starting a simulation near an equilibrium will never reveal the attractor. Instead, the hidden attractor may be reached only from specific initial conditions, often far from any equilibrium point [11]. This fact makes them invisible to traditional numerical methods and requires more careful analysis.

Hidden attractors were discovered only in the past decade [12]. They have sparked interest because of their surprising and sometimes dangerous behavior. In mechanical systems, they can cause unexpected vibrations. In electronics, they may lead to unpredictable oscillations. In secure communication, their unpredictability can be used as an advantage. Because they are hard to locate, hidden attractors may also hide instabilities that designers are unaware of.

Studying hidden attractors helps us understand the full range of behaviours a system can show. It also forces us to improve our simulation techniques. New methods for detecting hidden attractors are being developed, including analytical approaches and refined numerical searches. These tools are helping researchers uncover attractors that were previously missed.

In our paper, we use a predefined attractor equation that defines a hidden attractor of the system and is expressed in polar coordinates. Such an approach allows us to use some known chaotic system's output as an argument for polar curves and define chaotic motions along these curves by transforming system motions from polar coordinates to orthogonal ones and differentiating components of system position.

2. Method

2.1. Continuous-time infinite-dimensional dynamical system in polar coordinates

Let us consider a motion equation of the generalized nonlinear dynamical system with delayed feedback

$$\dot{\theta} = f(\theta, \theta_\tau), \tag{1}$$

here θ is the state variable of the system, τ is a time delay, θ_τ denotes the delayed state variable, t is a system time, $f(.)$ is some nonlinear function.

It is well known that delayed feedback is required information about the system's previous states. From the control theory viewpoint, such a system is infinite-dimensional dynamical system. The trajectories of such systems lie in an infinite-dimensional phase space formed by the values of the state variable at different past instants. On the other hand, the considered system is defined by using only one state variable; thus, its motion can be localized by the position of the system representative point in some coordinate axis.

It is clear that in the most general case, it is possible to consider (1) as an evolution equation governing the motion. Since, in the most general case, such a system can have no physical meaning or have already been technically implemented, we can consider it as the mathematical foundation for generating signals that can be used in different applications.

Since some systems have no physical nature, considering (1) simultaneously with studying its properties raises the problem of interpretation of the obtained data because this data can describe different processes. Clearly, in the case of the generalized system study, it is possible to consider its output at will. We propose to interpret the solution (1) as some driving signal.

In our paper, we propose consider the above-mentioned transitional signal θ as an argument for the generalized polar curve equation

$$\rho = g(\theta), \tag{2}$$

here we consider ρ as some virtual nonlinear-dependent state variable to increase the number of used state variables that define system motions.

Such a signal interpretation gives us the possibility to define a new signal rho, which is some nonlinear function $g(.)$ of the transitional signal θ. Using signals ρ and $theta$ makes it possible to define some polar axes and study system motions in a new coordinate plane by using angular and linear positions of the system representative point. These positions can be considered nonlinear-dependent, but contrary to solution (1), they define system position in a plane instead of one axis. Thus, using (1) and (2) allows us to increase the coordinate system's order, where the system trajectory's current values are studied.

One can consider (1) and (2) as differential-algebraic equations. Obtaining the general solution for such a system is very hard, so we propose use known numerical methods instead. The joint solution of (1) and (2) allows us to study the motion of the considered nonlinear systems in polar coordinates. It is very convenient to use such coordinates while the motion of a body in a plane is implemented due to the possibility of operating its angular and linear speeds.

If the problem of path planning is solved, it is possible to find that using the above-mentioned approach is quite tricky since it defines body motion but does not show the coordinates in the plane where the body occurs. We solve this problem by performing a transformation from polar to Cartesian coordinates as follows

$$\dot{\theta} = f(\theta, \theta_\tau),\ \rho = g(\theta),\ x = \rho \cos\theta,\ y = \rho \sin\theta. \tag{3}$$

It is clear that (3) defines system polar coordinates and then its Cartesian ones in a series way, and due to the considered dependence of the system's linear position on the angular one, (3) can be rewritten as follows

$$\dot{\theta} = f(\theta, \theta_\tau),\ x = g(\theta)\cos\theta,\ y = g(\theta)\sin\theta \tag{4}$$

to decrease the number of system variables and thus reduce the hardware resources used to implement the considered system.

From a control theory viewpoint, (4) is a multi-output dynamical system that produces two nonlinear output signals at the same time. Contrary to (1)-(2), (4) defines system position by using these outputs only, so the second and third expressions in (4) can be considered as the observability equations for a dynamical system whose motion is defined by the first equation in (4). Using (4) to simulate system motion and define its position is very convenient since the numerical integration procedures should be used for only one equation per calculation cycle.

At the same time, the use of analytical and numerical methods to study system dynamics without performing simulation of its motions but using methods which are based on considering system frequency response and its derivatives, as well as algebraic methods which are based on considering system eigenvalues, Lyapunov functions, etc., requires operating with differential equations. That is why we offer to rewrite (4) in the form of differential equations, by differentiating the observability equations

$$\dot{x} = \frac{\partial g(\theta)}{\partial \theta}\dot{\theta}\cos\theta - g(\theta)\dot{\theta}\sin\theta;\ \dot{y} = \frac{\partial g(\theta)}{\partial \theta}\dot{\theta}\sin\theta + g(\theta)\dot{\theta}\cos\theta;\ \dot{\theta} = f(\theta, \theta_\tau). \tag{5}$$

The main feature of (5) is the dependencies of all linear system coordinates from its angular speed and position, which have a strong pattern that is the multiplication of the system angular speed by some nonlinear function that depends on the system position. At the same time, if one divides the second equation (5) by the first one

$$\frac{\dot{y}}{\dot{x}} = \frac{\frac{\partial g(\theta)}{\partial \theta}\sin\theta + g(\theta)\cos\theta}{\frac{\partial g(\theta)}{\partial \theta}\cos\theta - g(\theta)\sin\theta};\ \dot{\theta} = f(\theta, \theta_\tau), \tag{6}$$

the motion speed in the phase plane can be defined. We call (6) the differential equation of the system phase portrait.

As we can see, this speed depends only on the system's angular position and does not depend on its angular speed. However, it is possible to find the angular position θ by integrating the second equation (6).

The above-mentioned studied system's feature requires defining one system's polar coordinate before defining its Cartesian ones. One can find that it is redundant while the system's motion in a plane is studied.

We offer to avoid the above-mentioned redundancy by using the following generalized algorithm:

- At first, we substitute the third equation of (5) in the first and second ones

$$\dot{x} = \frac{\partial g(\theta)}{\partial \theta} f(\theta, \theta_\tau) \cos\theta - g(\theta) f(\theta, \theta_\tau) \sin\theta; \; \dot{y} = \frac{\partial g(\theta)}{\partial \theta} f(\theta, \theta_\tau) \sin\theta + g(\theta) f(\theta, \theta_\tau) \cos\theta. \quad (7)$$

  Equations show that in the case when the coordinate of the initial dynamical system is considered as its angular position, we can define the generalized equations for the components of the system's linear speed in its phase plane as a function of the current and previous system angular positions.
- Then, we solve second and third equations of (4) for system angular position θ

$$\theta = g_x^{-1}(x), \; \theta = g_y^{-1}(y), \quad (8)$$

  here $g_x^{-1}(x)$ and $g_y^{-1}(y)$ are solution of the above-mentioned algebraic equations from (4). Determination of functions $g_x^{-1}(x)$ and $g_y^{-1}(y)$ is possible only in a few particular cases. In most cases, the solution (8) can be obtained only numerically.
- At third, we substitute (8) into (7) and define system motions in such a way

$$\begin{aligned} \dot{x} = {} & \tfrac{\partial g(g_x^{-1}(x))}{\partial g_x^{-1}(x)} f(g_x^{-1}(x), g_x^{-1}(x_\tau)) \cos(g_x^{-1}(x)) - \\ & - g(g_x^{-1}(x)) f(g_x^{-1}(x), g_x^{-1}(x_\tau)) \sin(g_x^{-1}(x)); \\ \dot{y} = {} & \tfrac{\partial g(g_y^{-1}(y))}{\partial g_y^{-1}(y)} f(g_y^{-1}(y), g_y^{-1}(y_\tau)) \sin(g_y^{-1}(y)) + \\ & + g(g_y^{-1}(y)) f(g_y^{-1}(y), g_y^{-1}(y_\tau)) \cos(g_y^{-1}(y)). \end{aligned} \quad (9)$$

Analysis of (9) allows us to claim that the motion of the considered system is defined in Cartesian coordinates only, and no other coordinates are used. Moreover, the definition of system angular position θ as a function of x and y makes it possible to define two independent equations that can be solved separately to find a corresponding system coordinate. This approach allows us to solve each of (9) using different devices, cores, and computational units to improve system performance. Also, we can find that only one equation from (9) can be solved at the current time. This allows us to solve one equation from (9), store its solution as some array of numbers, and then solve another equation using the same device.

At the same time, the separate solution of (9) can cause some computational errors due to the discrete nature of numerical integration routines. These errors can be different for each equation in (9) and, thus, cause a significant difference from the desired motion trajectories. We offer to consider possible computational errors and define some interrelations between the first and second equations in (9) to compensate for them.

One of the ways to establish interrelations between system coordinates is by using each solution (8) in each equation (9). Generally speaking, we can define various interrelations that are out of the scope of the current paper and the topic of our future studies.

Here we only designate the symmetrical and asymmetrical forms of interrelated equations.

The symmetrical one can be given as follows:

$$\begin{aligned} \dot{x} &= \tfrac{\partial g(g_x^{-1}(x))}{\partial g_x^{-1}(x)} f(g_x^{-1}(x), g_x^{-1}(x_\tau)) \cos(g_x^{-1}(x)) - g(g_y^{-1}(y)) f(g_y^{-1}(y), g_y^{-1}(y_\tau)) \sin(g_y^{-1}(y)); \\ \dot{y} &= \tfrac{\partial g(g_y^{-1}(y))}{\partial g_y^{-1}(y)} f(g_y^{-1}(y), g_y^{-1}(y_\tau)) \sin(g_y^{-1}(y)) + g(g_x^{-1}(x)) f(g_x^{-1}(x), g_x^{-1}(x_\tau)) \cos(g_x^{-1}(x)). \end{aligned} \quad (10)$$

These equations define the second summands of each equation as cross-links to transform the information about the coordinate x channel, which forms another one y, and vice versa.

The asymmetrical one can be rewritten in such a way:

$$\dot{x} = \frac{\partial g(g_x^{-1}(x))}{\partial g_x^{-1}(x)} f(g_x^{-1}(x), g_x^{-1}(x_\tau)) \cos(g_x^{-1}(x)) - g(g_x^{-1}(x)) f(g_x^{-1}(x), g_x^{-1}(x_\tau)) \sin(g_x^{-1}(x)); \tag{11}$$
$$\dot{y} = \frac{\partial g(g_y^{-1}(y))}{\partial g_y^{-1}(y)} f(g_y^{-1}(y), g_y^{-1}(y_\tau)) \sin(g_y^{-1}(y)) + g(g_x^{-1}(x)) f(g_x^{-1}(x), g_x^{-1}(x_\tau)) \cos(g_x^{-1}(x)).$$

One can use information only about one coordinate x in the second summand in both equations in this case.

It is clear that, regardless of the form of the equations used, it is necessary to solve both equations simultaneously.

The main feature of (9)-(11), as well as (7), is the determination of their right-hand expressions with harmonic functions of current values of system state variables. Since these harmonic functions are defined explicitly, we claim the definition of a dynamical system with a periodic nonlinear function in the right-hand expressions.

Analysis of the given equations shows that, in all cases, considering the dynamical system output in polar coordinates with subsequent transformation into Cartesian coordinates enables to increase the number of variables to study the system dynamics.

2.2. Discrete-time models of infinite-dimensional systems in polar coordinates

All above-obtained equations are given in continuous time domain, which reflects the continuous nature of the system's operating time. At the same time, the technical implementation of such systems poses challenges when high-precision analogue circuitry is required, especially when the studied systems do not have real technical analogues and are constructed from scratch using only some mathematical equations.

Nowadays, such systems are typically implemented using digital programmable devices such as MCUs, DSPs, FPGAs, and others, which are discrete-time devices driven by some oscillations at fixed time intervals.

Contrary to the continuous-time systems, whose motions are defined by differential equations, the motions of discrete-time systems are defined using finite-difference equations. This type of equations differs from differential ones by using a finite-difference approximation of derivatives instead of differentiating functions and signals.

In our paper, we define the following generalized approximation operator

$$\frac{\partial q_1}{\partial q_2} \approx D(q_1, z^{-1}q_1, \ldots, z^{-n}q_1, q_2, z^{-1}q_2, \ldots, z^{-m}q_2), \tag{12}$$

here q_1 and q_2 are two signals that define some derivative, z is a shift operator, and z^{-i} means that the previous values of system state variable are taken for the time moment $t - iT_s$, T_s is a sample time, n and m are number of used previous values of system state variables.

It is clear that (12) is the most general operator to approximate signal derivatives using its current and previous values. The high orders of previous values can be used to correct the approximated derivative and smooth it out using various numerical integration methods. At the same time, it is possible to easily simplify (12) in such a way

$$\frac{\partial q_1}{\partial q_2} \approx D_1(q_1, z^{-1}q_1, q_2, z^{-1}q_2) \tag{13}$$

by using only the current and previous values of signals to define their first derivative.

Approximations (12) and (13) define the approximation of the first derivative of one signal by another, and that is why we use both of these signals as the arguments of the proposed approximation operator. We offer to use them to define the partial derivatives in the first summands (9)–(11).

At the same time, we can analyze these equations and find derivatives of signals x and y concerning time. We offer to approximate such derivatives by using the following generalized operators

$$\frac{dq_1}{dt} \approx D_t(q_1, z^{-1}q_1, \dots, z^{-n}q_1), \tag{14}$$

and

$$\frac{dq_1}{dt} \approx D_{t1}(q_1, z^{-1}q_1) \tag{15}$$

which are similar to previously defined ones by using information only for the values of one signal, which is differentiating.

Specific definitions of the considered approximation operators depend on the finite difference scheme and algorithms. In the simplest case, we can use a linear finite difference formula, which we generalize for the case of approximating some generalized derivative as follows

$$D_1(q_1, z^{-1}q_1, q_2, z^{-1}q_2) = \tfrac{q_1 - z^{-1}q_1}{q_2 - z^{-1}q_2}, \tag{16}$$

$$D_{t1}(q_1, z^{-1}q_1) = \tfrac{q_1 - z^{-1}q_1}{T_s}. \tag{17}$$

In a more complex case, it is possible to use some nonlinear combinations of system state variables to improve the accuracy of the considered discrete-time system and make its motions close to the motions of the initial nonlinear system.

Another feature of the considered infinite-dimensional systems is the use of time-delayed signals. Contrary to a continuous-time system, which can use any time delay values, in discrete-time systems, these values should be quantized in time. Thus, we use the trivial formulas to approximate the time delay τ

$$\tau \approx \tau_D = T_s \left[\frac{\tau}{T_s}\right] = T_s j. \tag{18}$$

Here, using square brackets means taking the integer part of a fraction, and j means the address of a previously defined system state variable in the discrete-time system's storage element.

In case of known and continuous sample time, we can define the time-delayed signals by using the shift operator as follows

$$x_\tau \approx z^{-j}x,\ y_\tau \approx z^{-j}y. \tag{19}$$

If one takes into mind approximations (19), (15), and (13), he can rewrite (9)–(11) in following way

$$\begin{aligned}
D_{t1}(x, z^{-1}x) &= D_1(g(g_x^{-1}(x)), g(g_x^{-1}(z^{-1}x)), g_x^{-1}(x), g_x^{-1}(z^{-1}x)) f(g_x^{-1}(x), g_x^{-1}(z^{-j}x)) \times \\
&\times \cos(g_x^{-1}(x)) - g(g_x^{-1}(x)) f(g_x^{-1}(x), g_x^{-1}(z^{-j}x)) \sin(g_x^{-1}(x)); \\
D_{t1}(y, z^{-1}y) &= D_1(g(g_y^{-1}(y)), g(g_y^{-1}(z^{-1}y)), g_y^{-1}(y), g_y^{-1}(z^{-1}y)) f(g_y^{-1}(y), g_y^{-1}(z^{-j}y)) \times \\
&\times \sin(g_y^{-1}(y)) + g(g_y^{-1}(y)) f(g_y^{-1}(y), g_y^{-1}(z^{-j}y)) \cos(g_y^{-1}(y)),
\end{aligned} \tag{20}$$

$$\begin{aligned}
D_{t1}(x, z^{-1}x) &= D_1(g(g_x^{-1}(x)), g(g_x^{-1}(z^{-1}x)), g_x^{-1}(x), g_x^{-1}(z^{-1}x)) f(g_x^{-1}(x), g_x^{-1}(z^{-j}x)) \times \\
&\times \cos(g_x^{-1}(x)) - g(g_y^{-1}(y)) f(g_y^{-1}(y), g_y^{-1}(z^{-j}y)) \sin(g_y^{-1}(y)); \\
D_{t1}(x, z^{-1}y) &= D_1(g(g_y^{-1}(y)), g(g_y^{-1}(z^{-1}y)), g_y^{-1}(y), g_y^{-1}(z^{-1}y)) f(g_y^{-1}(y), g_y^{-1}(z^{-j}y)) \times \\
&\times \sin(g_y^{-1}(y)) + g(g_x^{-1}(x)) f(g_x^{-1}(x), g_x^{-1}(z^{-j}x)) \cos(g_x^{-1}(x)).
\end{aligned} \tag{21}$$

$$\begin{aligned}
D_{t1}(x, z^{-1}x) &= D_1(g(g_x^{-1}(x)), g(g_x^{-1}(z^{-1}x)), g_x^{-1}(x), g_x^{-1}(z^{-1}x)) f(g_x^{-1}(x), g_x^{-1}(z^{-j}x)) \times \\
&\times \cos(g_x^{-1}(x)) - g(g_x^{-1}(x)) f(g_x^{-1}(x), g_x^{-1}(z^{-j}x)) \sin(g_x^{-1}(x)); \\
D_{t1}(x, z^{-1}y) &= D_1(g(g_y^{-1}(y)), g(g_y^{-1}(z^{-1}y)), g_y^{-1}(y), g_y^{-1}(z^{-1}y)) f(g_y^{-1}(y), g_y^{-1}(z^{-j}y)) \times \\
&\times \sin(g_y^{-1}(y)) + g(g_x^{-1}(x)) f(g_x^{-1}(x), g_x^{-1}(z^{-j}x)) \cos(g_x^{-1}(x)).
\end{aligned} \tag{22}$$

We call (20)–(22) as the discrete-time models of the generalized infinite-dimensional dynamical system and offer to implement these systems using various digital devices. To improve system properties and features, it is possible to use more complex approximations (12) and (13). Moreover, the simplest backward feedback differences (16) and (17) can be replaced with the Tustin transformation or more complex ones to increase the stability indicators for the designed discrete-time systems.

One can use (20)–(22) to define the current values of system state variables by solving these equations for x and y. Such a solution can be given as follows

$$x = h_x(z^{-1}x, z^{-j}x),\ y = h_y(z^{-1}y, z^{-j}y), \tag{23}$$

here h_x and h_y are solutions of (20)–(22).

In the most general case, due to the nonlinearities (20)–(22), the solutions (23) cannot be given analytically, and one should use the Newton-Raphson numerical methods to find the functions $h_x(.)$ and $h_y(.)$. Such an approach allows us to consider these functions in the class of iterative functions and thus generalize the considered discrete-time model into a class of models with iterations. Using such models to solve the inverse dynamic problem is quite interesting from the control theory viewpoint and control algorithm determination. We leave the design of such controllers for our future works and here remark that we can use (23) to implement the designed model in various digital programmable devices.

3. Results and discussion

3.1. Modeling of Mackey-Glass system with cardioid attractor

Let us consider using the proposed approach to define motion trajectories of a well-known Mackey-Glass system, whose dynamics we describe by the following differential equation

$$\dot{\theta} = \gamma\theta + \beta\frac{\theta_\tau}{1+\theta_\tau^n}, \tag{24}$$

here $\gamma = -1$ and $\beta = 2$ are system's weight factors, $n = 10$ is power factor, and $\tau = 4$ is system time delay. The values for the above-mentioned factors form a chaotic dynamic for the considered Mackey-Glass system.

In our paper, we show the use of our approach by considering a chaotic system design with a cardioid-like attractor. However, our approach allows us to use any polar curve even highly complex ones.

The cardioid curve is defined by the following equation

$$\rho = 1 + \cos(\theta). \tag{25}$$

Let us define projections of radius-vector ρ in Cartesian coordinate axes

$$x = \rho\cos(\theta);\ x = \rho\sin(\theta). \tag{26}$$

Substituting (25) into (26) yields following expressions

$$x = \cos(\theta) + \cos^2(\theta);\ y = \sin(\theta) + \sin(\theta)\cos(\theta), \tag{27}$$

we consider as observability equations for system (24).

According to the above-considered approach, it is necessary to use direct and inverse dependencies between the system position in different coordinates. That is why, before constructing the system model in Cartesian coordinates, we solve (27) for θ.

The first equation in (26) can be solved analytically for the considered system's angular position θ

$$\theta_{1x} = \arccos\left(-0.5 + 0.5\sqrt{1+4x}\right);\ \theta_{2x} = \pi - \arccos\left(0.5 + 0.5\sqrt{1+4x}\right). \tag{28}$$

At the same time, there is no analytical solution for the second equation of (24), and we propose use numerical routines from the SciPy library to solve it.

This solution is obtained by using following algorithm:

- Firstly, we introduce a new variable u which interrelates with $theta$ as follows

$$u = \sin(\theta) \tag{29}$$

and solve (29) as follows

$$\theta = \arcsin(u). \tag{30}$$

- Secondly, we rewrite second equation (24) in terms of u and y only

$$y = u + u\sqrt{1 - u^2}. \tag{31}$$

- Thirdly we analyze (25) and (26) as well as cardioid plot (figure 1)

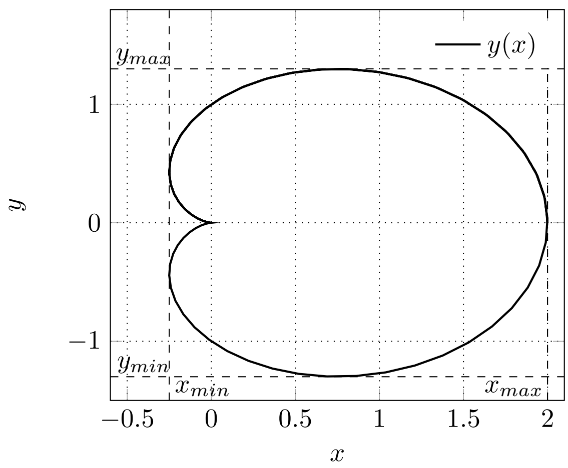

Figure 1: Cardioid in Cartesian coordinates.

and find intervals of possible values for x and y outputs

$$x \in [x_{min}, x_{max}];\ \ y \in [y_{min}, y_{max}]. \tag{32}$$

- At fourth, we use SciPy `fsolve` routine to find solution of (31) for y from interval (32). This solution is shown in figure 2

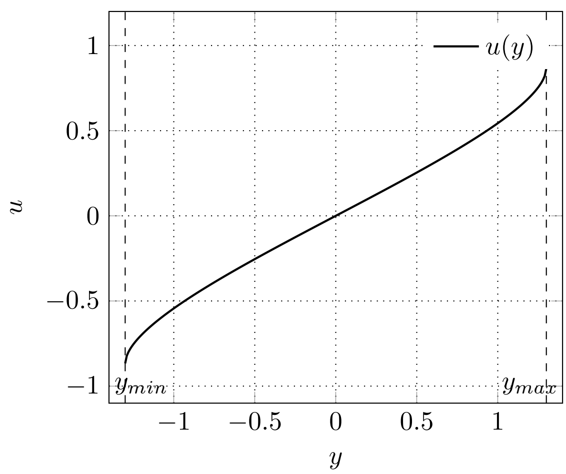

Figure 2: Results of numerical solution (31).

- At fifth, we use the given in Figure 2 function to define an approximated polynomial

$$\tilde{u} = a_3 y^3 + a_1 y_1, \tag{33}$$

here, the factors $a_3 = 0.0797$ and $a_1 = 0.475$ yielding a coefficient of determination $R^2 = 0.999$, which confirms the high accuracy of the approximation.
It is clear that for other cases, one should use other approximation functions and techniques, such as cubic splines, to get a highly accurate approximation of a nonlinear function.
- At last, if one substitutes (33) into (30), he can obtain the dependence of system angular coordinate θ from vertical component y of its position in Cartesian coordinates

$$\theta_y \approx \arcsin(a_3 y^3 + a_1 y). \tag{34}$$

It is possible to equalize (34) and (28) and write down equation which define interrelation between horizontal x and vertical y system's positions

$$\arcsin(a_3 y^3 + a_1 y) = \arccos\left(-0.5 + 0.5\sqrt{1+4x}\right), \tag{35}$$

here and further, we use the first simplest root of the first equation (27).

One can solve (35) to define the dependence between the considered system linear positions x and y.

Since all interrelations between system coordinates are defined now, we consider the design of the model for the considered system in Cartesian coordinates. According to (5) we can define such a models by differentiating system outputs

$$\dot{x} = -\sin(\theta)(2\cos(\theta)+1)\dot{\theta};\ \dot{y} = (2\cos^2(\theta)+\cos(\theta)-1)\dot{\theta}. \tag{36}$$

Expressions (36) define projections of the system's linear speed in coordinate axes and thus can be considered as system motion equations. These equations interrelate components of linear speed with angular ones by using complex trigonometric functions. Since, according to (24), system angular speed depends on its angular position, it is possible to use these derivatives to define components of linear speed as functions of system angular position by substituting (24) into (36)

$$\dot{x} = -\sin(\theta)(2\cos(\theta)+1)\left(\gamma\theta + \beta\frac{\theta_\tau}{1+\theta_\tau^n}\right);\ \dot{y} = (2\cos^2(\theta)+\cos(\theta)-1)\left(\gamma\theta + \beta\frac{\theta_\tau}{1+\theta_\tau^n}\right). \tag{37}$$

Analysis (37) proves the possibility of clearly defining system motion components.

At the same time, the main feature of models (37) and (36) is the dependency of components of the system's linear speed from its angular position. Thus, we can claim that the considered model is the parametric one and its usage requires solving both linear models (37) and angular one (24).

We offer avoiding this drawback by using (28) and (34) to rewrite models' equations in terms of linear coordinates only. As it is mentioned above and shown by (9)–(11), several cases for such models are possible.

At first, we define equations for the model in Cartesian coordinates without any interrelations between output signals. This model is obtained from (37) by substituting (28) and (34) in this equation

$$\begin{aligned}\dot{x} = &-\left(\sin\left(\arccos\left(-0.5+0.5\sqrt{1+4x}\right)\right)\sqrt{1+4x}\right)\times\\ &\times\left(\gamma\arccos\left(-0.5+0.5\sqrt{1+4x}\right)+\beta\frac{\arccos\left(-0.5+0.5\sqrt{1+4x_\tau}\right)}{1+\arccos\left(-0.5+0.5\sqrt{1+4x_\tau}\right)^n}\right);\\ \dot{y} = &\ (2\cos^2(\arcsin(a_3y^3+a_1y))+\cos(\arcsin(a_3y^3+a_1y))-1)\times\\ &\times\left(\gamma\arcsin(a_3y^3+a_1y)+\beta\frac{\arcsin(a_3y_\tau^3+a_1y_\tau)}{1+\arcsin(a_3y_\tau^3+a_1y_\tau)^n}\right).\end{aligned} \tag{38}$$

Equations (38) define motions of two independent dynamical systems. One can use the proposed approach to increase the number of system output channels that operate in parallel [13], and can be implemented by using various devices or parts of one device.

It should be mentioned that since no external signals are considered, the considered dynamical system is an autonomous one and its motions are caused by non-zero initial conditions, which for (24) are given in such a way

$$\theta(0) = \theta_0, \theta_\tau(0) = 0. \tag{39}$$

The usage (27) enables to define any interrelations between system coordinates and also use them, as well as (39), to write down initial values of the system's motion in Cartesian coordinates

$$x(0) = \cos(\theta_0) + \cos^2(\theta_0);\ y(0) = \sin(\theta_0) + \sin(\theta_0)\cos(\theta_0),\ x_\tau(0) = 2;\ y_\tau(0) = 0. \tag{40}$$

Analysis (40) shows the difference of system initial conditions in various coordinates. Moreover, as we can see, the transformation of a delayed-dynamical system into another coordinate system can also cause a change in the initial value of the delayed variable.

If one analyzes (38), he finds that these equations consist of two multipliers. The first one depends on coordinate transformations, and the second one is defined by system motion. This fact according to (10) and (11) enables to design dynamical systems with interconnected symmetrical

$$\begin{aligned}
\dot{x} = -&\left(\sin\left(\arccos\left(-0.5 + 0.5\sqrt{1+4x}\right)\right)\sqrt{1+4x}\right) \times \\
&\times\left(\gamma \arcsin(a_3y^3 + a_1y) + \beta\frac{\arcsin(a_3y_\tau^3 + a_1y_\tau)}{1+\arcsin(a_3y_\tau^3+a_1y_\tau)^n}\right); \\
\dot{y} = &\left(2\cos^2(\arcsin(a_3y^3+a_1y)) + \cos(\arcsin(a_3y^3+a_1y)) - 1\right) \times \\
&\times\left(\gamma \arccos\left(-0.5+0.5\sqrt{1+4x}\right) + \beta\frac{\arccos\left(-0.5+0.5\sqrt{1+4x_\tau}\right)}{1+\arccos\left(-0.5+0.5\sqrt{1+4x_\tau}\right)^n}\right);
\end{aligned} \tag{41}$$

and asymmetrical

$$\begin{aligned}
\dot{x} = -&\left(\sin\left(\arccos\left(-0.5 + 0.5\sqrt{1+4x}\right)\right)\sqrt{1+4x}\right) \times \\
&\times\left(\gamma \arcsin(a_3y^3 + a_1y) + \beta\frac{\arcsin(a_3y_\tau^3 + a_1y_\tau)}{1+\arcsin(a_3y_\tau^3+a_1y_\tau)^n}\right); \\
\dot{y} = &\left(2\cos^2(\arcsin(a_3y^3+a_1y)) + \cos(\arcsin(a_3y^3+a_1y)) - 1\right) \times \\
&\times\left(\gamma \arcsin(a_3y^3 + a_1y) + \beta\frac{\arcsin(a_3y_\tau^3 + a_1y_\tau)}{1+\arcsin(a_3y_\tau^3+a_1y_\tau)^n}\right);
\end{aligned} \tag{42}$$

channels.

It is necessary to say that we can modify (41) and (42) to define various combinations of horizontal and vertical coordinates because these coordinates are algebraically dependent according to (35).

Now we turn our attention to the implementation of the designed models.

We use backward finite difference approximation (17) for the derivative operator and (18)-(19) to define delayed signals. After applying these formulas to (38), the following discrete-time model is defined

$$\begin{aligned}
x = z^{-1}x - T_s&\left(\sin\left(\arccos\left(-0.5 + 0.5\sqrt{1+4z^{-1}x}\right)\right)\sqrt{1+4z^{-1}x}\right) \times \\
&\times\left(\gamma \arccos\left(-0.5+0.5\sqrt{1+4z^{-1}x}\right) + \beta\frac{\arccos\left(-0.5+0.5\sqrt{1+4z^{-j-1}x}\right)}{1+\arccos\left(-0.5+0.5\sqrt{1+4z^{-j-1}x}\right)^n}\right); \\
y = z^{-1}y + T_s&(2\cos^2(\arcsin(a_3(z^{-1}y)^3 + a_1z^{-1}y)) + \cos(\arcsin(a_3(z^{-1}y)^3 + a_1z^{-1}y)) - 1) \times \\
&\times\left(\gamma \arcsin(a_3(z^{-1}y)^3 + a_1z^{-1}y) + \beta\frac{\arcsin(a_3(z^{-j-1}y)^3 + a_1z^{-j-1}y)}{1+\arcsin(a_3(z^{-j-1}y)_\tau^3+a_1z^{-j-1}y)^n}\right).
\end{aligned} \tag{43}$$

If one analyses (43) and compares approximation (17) with known numerical methods, he finds that the usage of above-given finite-difference equations are similar to applying Euler backward integration method to initial continuous time system (38).

The similar models are obtained from (41) and (42)

$$x = z^{-1}x - T_s\left(\sin\left(\arccos\left(-0.5 + 0.5\sqrt{1+4z^{-1}x}\right)\right)\sqrt{1+4z^{-1}x}\right) \times \tag{44}$$

$$\times\left(\gamma\arcsin(a_3(z^{-1}y)^3+a_1z^{-1}y)+\beta\frac{\arcsin(a_3(z^{-j-1}y)^3+a_1z^{-j-1}y)}{1+\arcsin(a_3(z^{-j-1}y)^3_\tau+a_1z^{-j-1}y)^n}\right).$$

$$y=z^{-1}y+T_s(2\cos^2(\arcsin(a_3(z^{-1}y)^3+a_1z^{-1}y))+\cos(\arcsin(a_3(z^{-1}y)^3+a_1z^{-1}y))-1)\times$$

$$\times\left(\gamma\arccos\left(-0.5+0.5\sqrt{1+4z^{-1}x}\right)+\beta\frac{\arccos\left(-0.5+0.5\sqrt{1+4z^{-j-1}x}\right)}{1+\arccos\left(-0.5+0.5\sqrt{1+4z^{-j-1}x}\right)^n}\right);$$

$$x=z^{-1}x-T_s\left(\sin\left(\arccos\left(-0.5+0.5\sqrt{1+4z^{-1}x}\right)\right)\sqrt{1+4z^{-1}x}\right)\times \tag{45}$$

$$\times\left(\gamma\arcsin(a_3(z^{-1}y)^3+a_1z^{-1}y)+\beta\frac{\arcsin(a_3(z^{-j-1}y)^3+a_1z^{-j-1}y)}{1+\arcsin(a_3(z^{-j-1}y)^3_\tau+a_1z^{-j-1}y)^n}\right).$$

$$y=z^{-1}y+T_s(2\cos^2(\arcsin(a_3(z^{-1}y)^3+a_1z^{-1}y))+\cos(\arcsin(a_3(z^{-1}y)^3+a_1z^{-1}y))-1)\times$$

$$\times\left(\gamma\arcsin(a_3(z^{-1}y)^3+a_1z^{-1}y)+\beta\frac{\arcsin(a_3(z^{-j-1}y)^3+a_1z^{-j-1}y)}{1+\arcsin(a_3(z^{-j-1}y)^3_\tau+a_1z^{-j-1}y)^n}\right).$$

Contrary to the initial differential equations, the equations (43)–(45) are algebraic ones, so we can easily solve them with programmable devices and calculate the system's motion trajectories.

3.2. The designed systems' numerical simulation

At first, we study the classical Mackey-Glass system motion (figure 3).

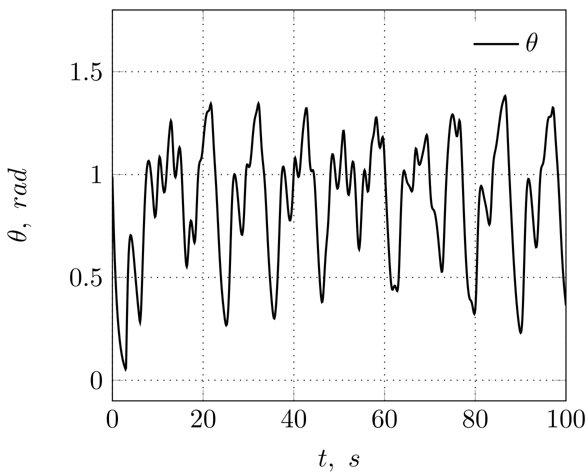

Figure 3: Results of numerical solution (24) with $\theta_0 = 1, \gamma = 1, \beta = 2, n = 10, j = 300$.

As shown in figure 3, the Mackey-Glass system has chaotic dynamics for a given classical set of parameters. Moreover, its motion trajectory is bounded and does not exceed 1.5. If one interprets Mackey-Glass system output as some angle, it is clear that only relatively small angular positions can be reached in this case. In our further studies, we increase the state variable θ by multiplying it by 6 to get from the angular position in the range from zero to 2π rad.

Such a multiplication enables to form a chaotic system radius-vector as shown in figure 4.

The given simulation results prove the above-given sentence about the bounded radius vector. Also, it is possible to see that any algebraic transformations of a chaotic signal lead to a new signal whose nature is also chaotic. Moreover, using trigonometric functions dramatically changes the produced signal compared to the initial one.

The last fact causes the necessity to study the formation of the system radius-vector for values of the multiplication factor k from the set $\{1, 2, 3, 4, 5, 6\}$, which defines the scaling of the system angle to reach various parts of the system trajectory. Corresponding simulation results are shown in figures 5–10.

Analysis given in figures 5–10 simulation results proves the dependence of system motions on the multiplication factor k and shows that increasing this factor makes system motions more nonlinear and

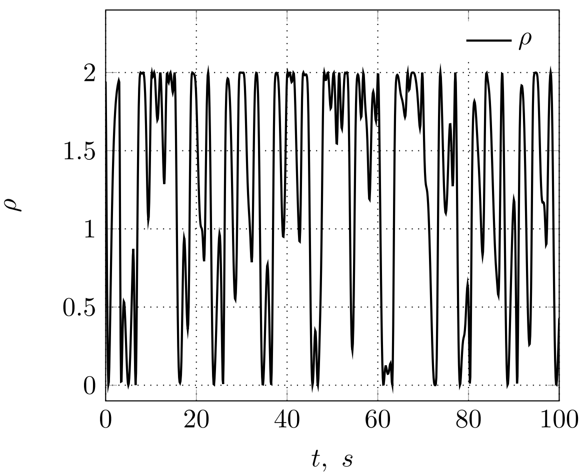

Figure 4: Studied system radius-vector ρ for $\theta \in [0, 2\pi]$.

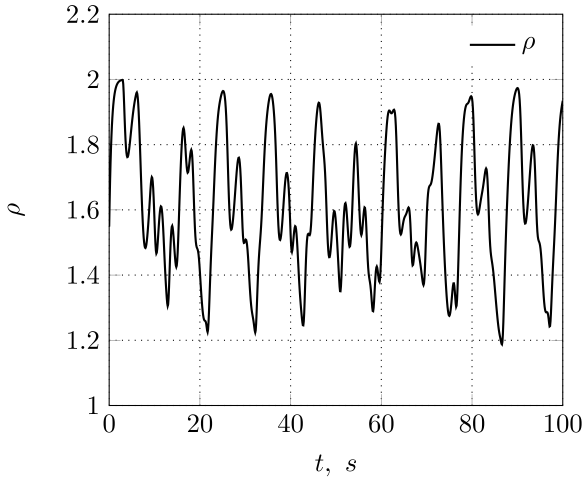

Figure 5: System radius-vector for $K = 1$.

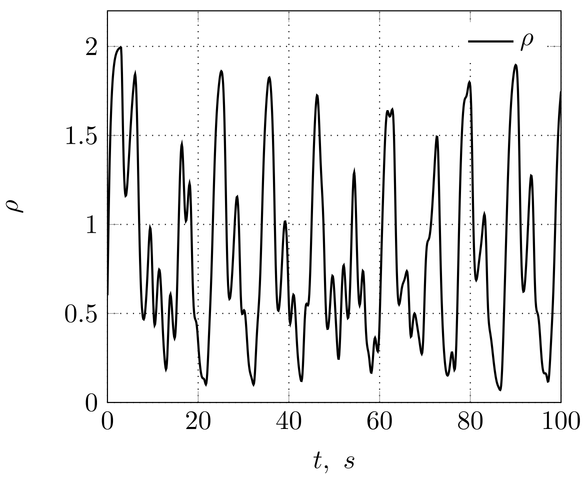

Figure 6: System radius-vector for $K = 2$.

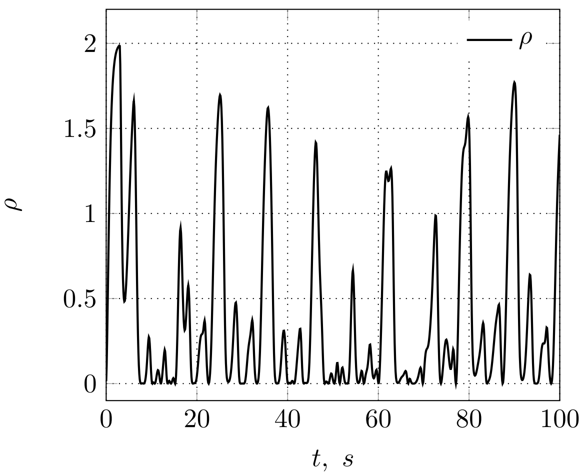

Figure 7: System radius-vector for $K = 3$.

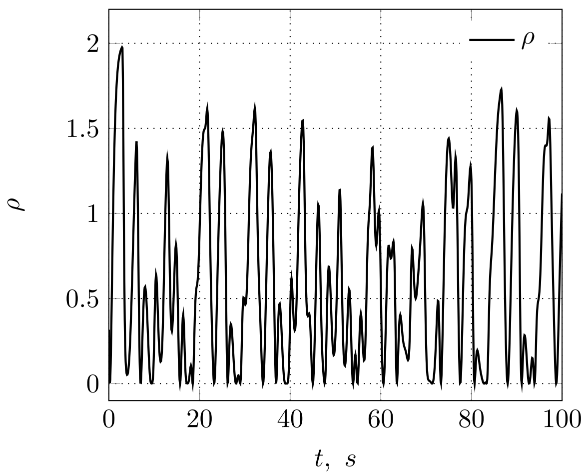

Figure 8: System radius-vector for $K = 4$.

increases the speed of radius-vector changing. We offer to consider this fact while a chaotic system with trigonometric functions is being designed.

The above-given numerical solutions for (24) and (25) are obtained by using the Arduino Due

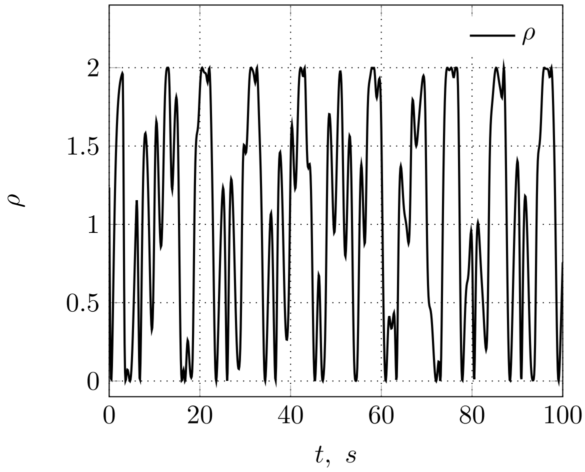

Figure 9: System radius-vector for $K = 5$.

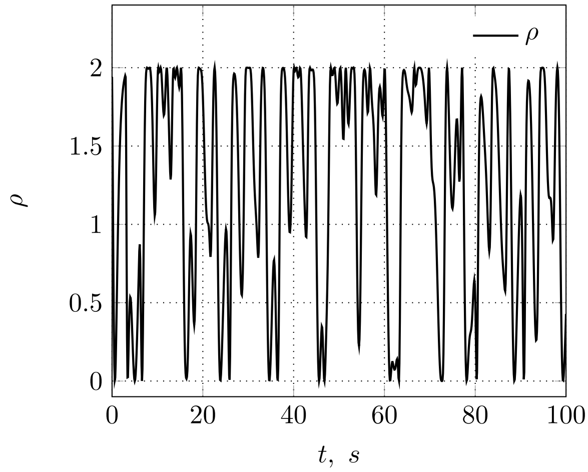

Figure 10: System radius-vector for $K = 6$.

developer board based on the Atmel AT91SAM3X8E ARM Cortex-M3 MCU as the calculation unit and transmitting data by using the serial communication interface to the PC. We use PgfPlot LaTeX package for visualising the obtained data.

In figures 11–28, we show simulation results for (43) which are obtained by using the same hardware implementation. These results are given for the different values of the multiplication factor K.

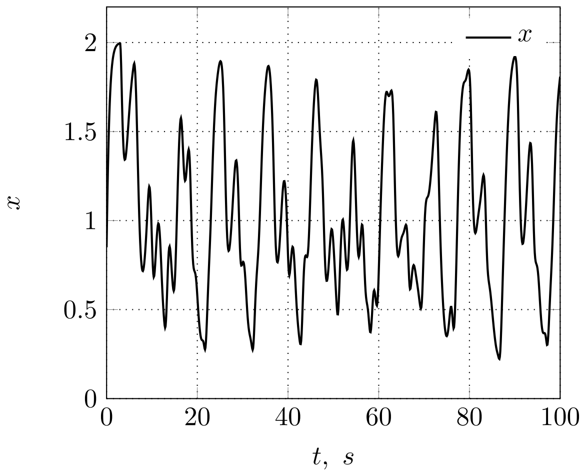

Figure 11: $x, K = 1$.

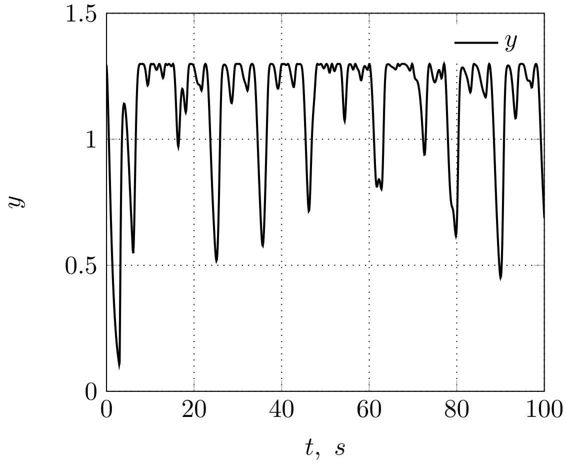

Figure 12: $y, K = 1$.

The simulation results show that the designed system has a cardioid-like localized attractor exhibiting hidden chaos. Moreover, it is possible to construct a system that forms only part of this attractor by using specific values of the multiplication factor K.

3.3. Parameters sensitivity analysis

To quantitatively assess the robustness of the proposed chaotic system, we perform a parameter sensitivity analysis and compare the results with the classical Mackey–Glass system. We plot bifurcation diagrams for both systems by varying a single parameter while keeping all other parameters fixed and equals to previous given values. This approach allows evaluating how small perturbations in key parameters affect the long-term dynamics and whether the chaotic regime remains structurally stable.

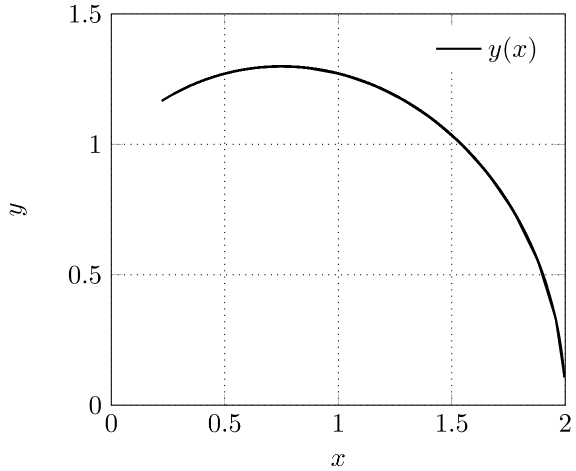

Figure 13: Phase portrait, $K = 1$.

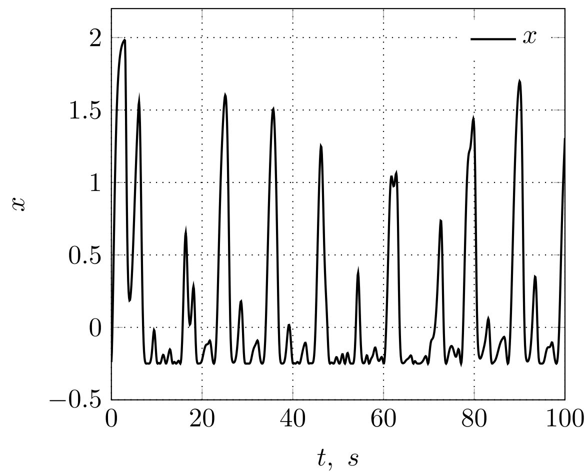

Figure 14: $x, K = 2$.

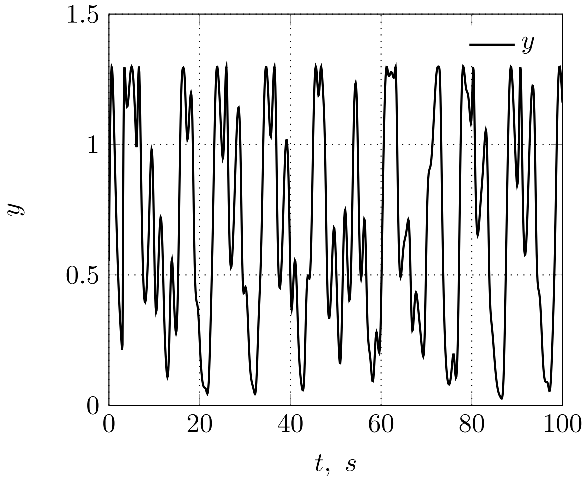

Figure 15: $y, K = 2$.

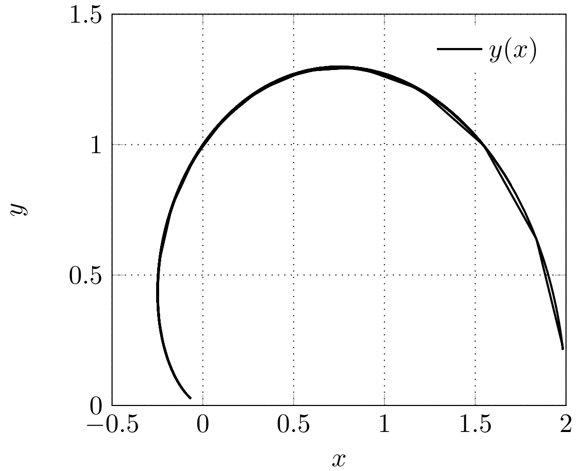

Figure 16: Phase portrait, $K = 2$.

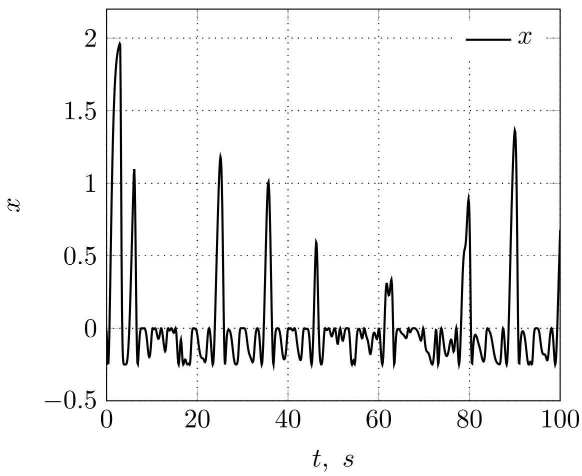

Figure 17: $x, K = 3$.

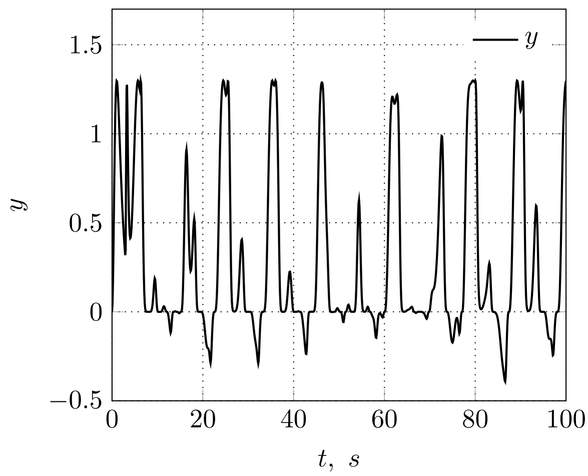

Figure 18: $y, K = 3$.

3.3.1. Sensitivity of the classical Mackey–Glass system

The classical Mackey–Glass equation is known for its high sensitivity to parameter variations, especially with respect to its parameters.

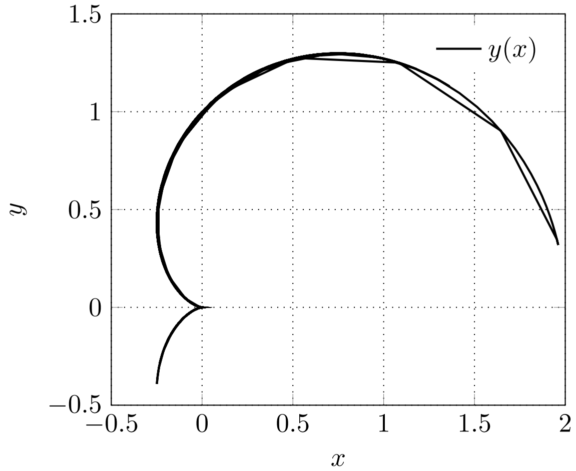

Figure 19: Phase portrait, $K = 3$.

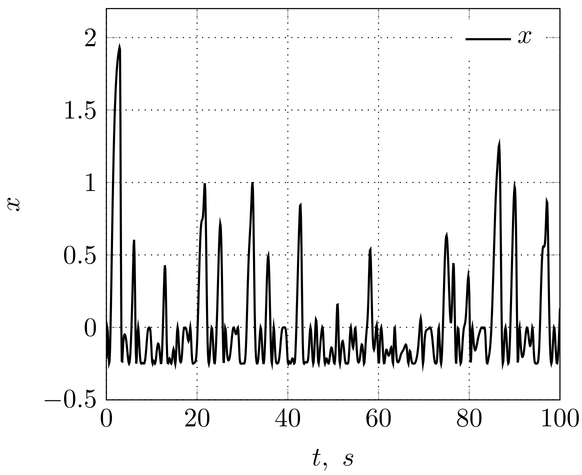

Figure 20: x, $K = 4$.

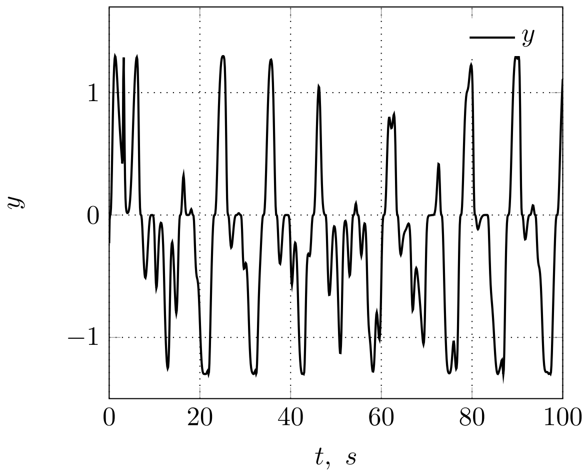

Figure 21: y, $K = 4$.

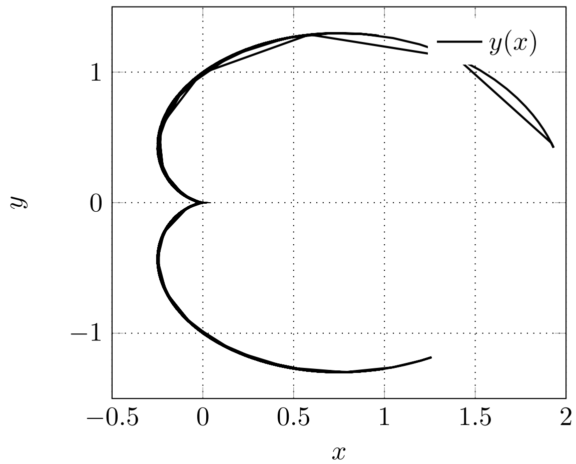

Figure 22: Phase portrait, $K = 4$.

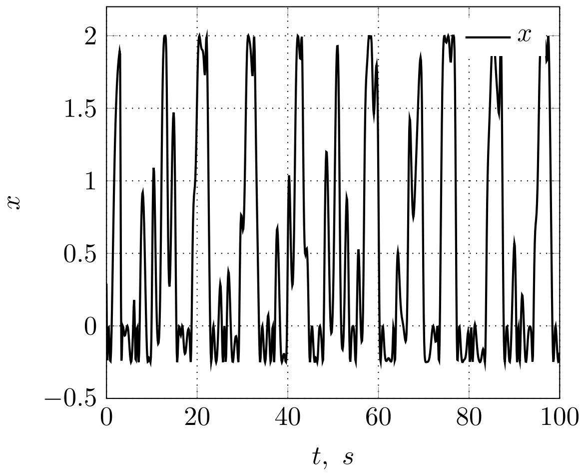

Figure 23: x, $K = 5$.

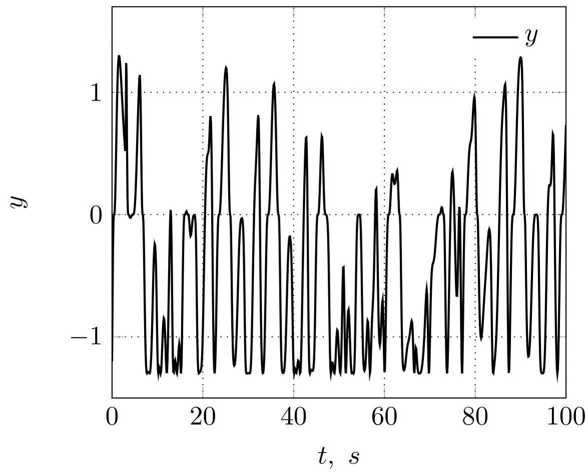

Figure 24: y, $K = 5$.

The bifurcation diagrams obtained for the reference model clearly demonstrate smooth periodic dynamics for small parameter values and the onset of fully developed chaos for sufficiently large values. In figures 29, 30, and 31 we show bifurcation diagrams to study system sensitivity for some factors which are the common for continuous and discrete time models. Due to the lack of space we do not

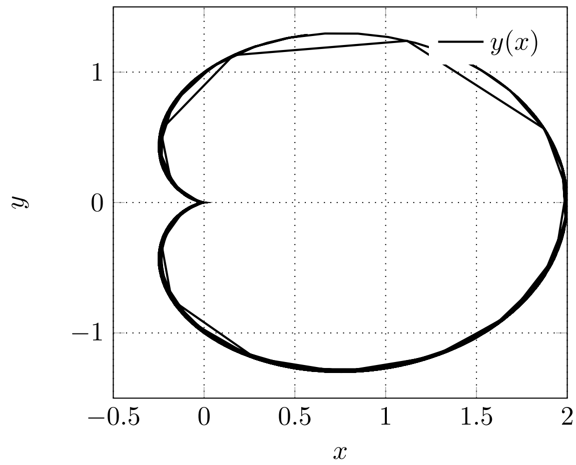

Figure 25: Phase portrait, $K = 5$.

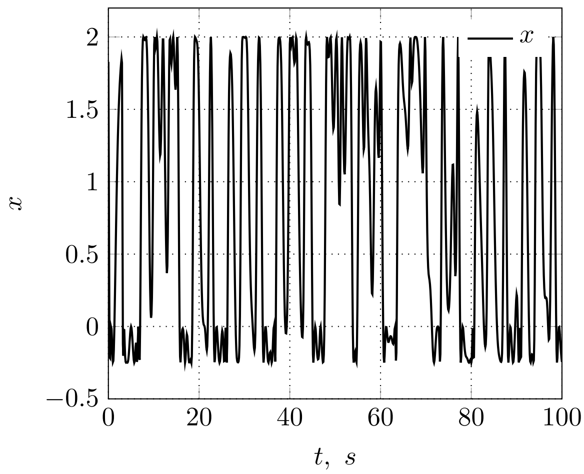

Figure 26: x, $K = 6$.

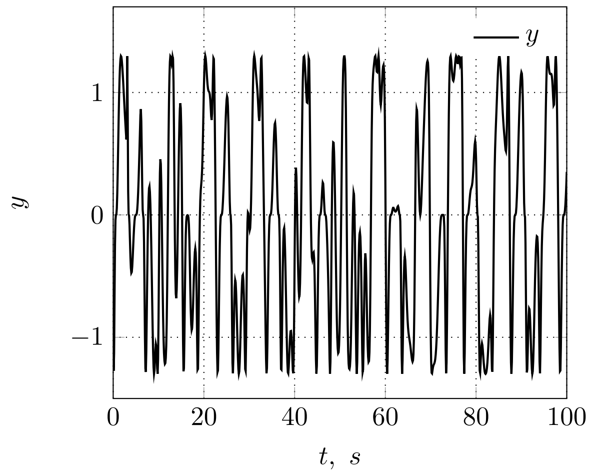

Figure 27: y, $K = 6$.

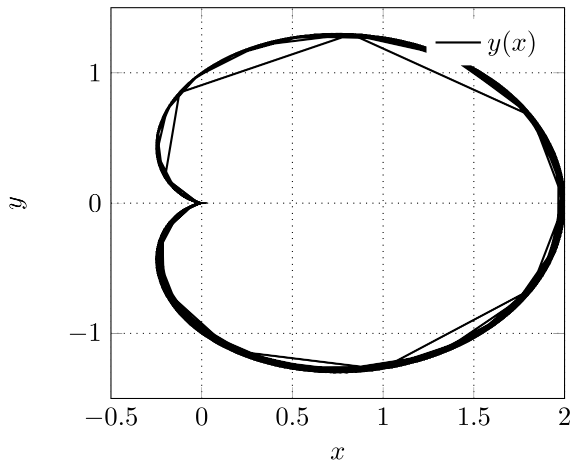

Figure 28: Phase portrait, $K = 6$.

show here the diagrams for specific factors of discrete-time model and leave them for future papers.

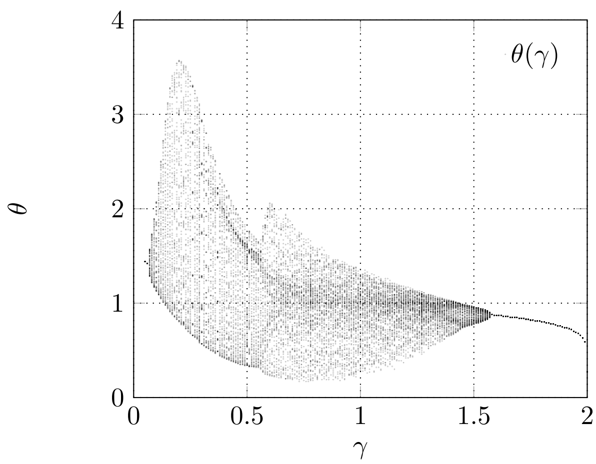

Figure 29: $\gamma = var$.

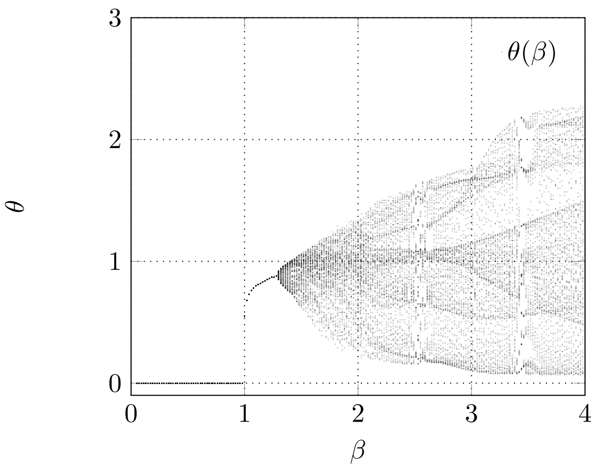

Figure 30: $\beta = var$.

These transitions are consistent with the well-documented behavior of the Mackey–Glass system, confirming that our numerical implementation reproduces the expected dynamical structure.

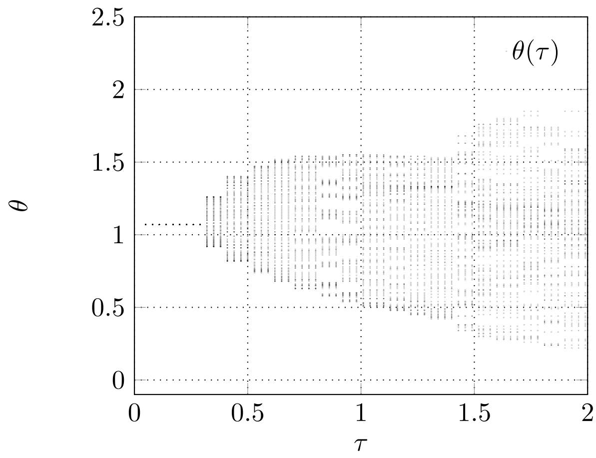

Figure 31: $\tau = var.$

3.3.2. Sensitivity of the proposed polar-attractor–based system

The proposed system, although structurally different, exhibits a qualitatively similar route to chaos. Its bifurcation diagrams show regions where the attractor remains single-valued and a broad chaotic band analogous to the chaotic zone of the Mackey–Glass system.

Compared to the classical model, the proposed system demonstrates a novel chaotic behavior (figures 32, 33, 34, 35, 36, and 37).

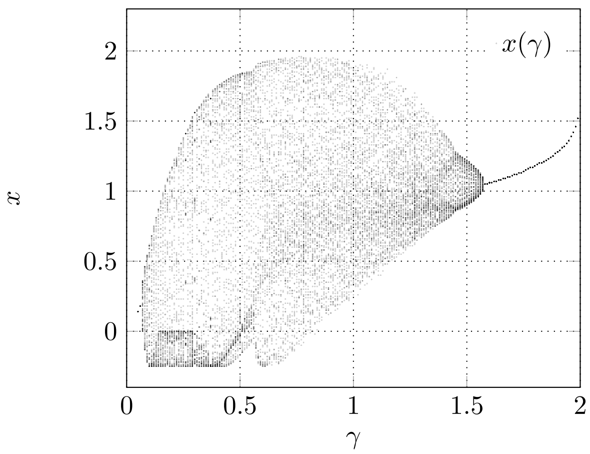

Figure 32: $\gamma = var.$

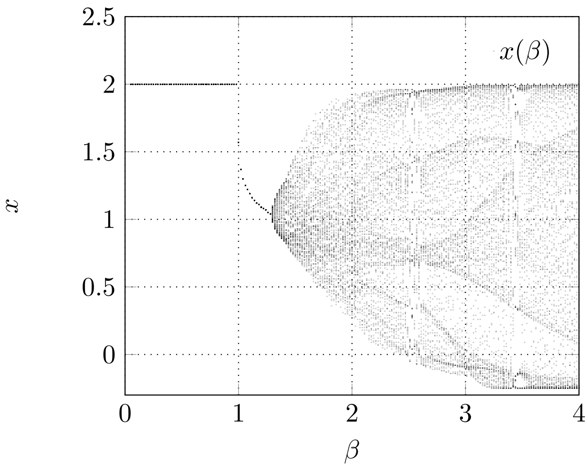

Figure 33: $\beta = var.$

This indicates that the chaos generated by the proposed model is less fragile with respect to parameter perturbations, an important property for hardware implementation where coefficients may slightly drift due to quantization or timing inaccuracies.

3.3.3. Comparative interpretation

By placing the two bifurcation diagrams side-by-side, several important observations can be made:

- Both systems share a similar qualitative bifurcation structure, confirming that the proposed design faithfully preserves the canonical characteristics of delay-based chaotic systems.
- The proposed system exhibits fewer narrow periodic windows, which typically complicate hardware implementation because they may be unintentionally triggered by quantization noise or clock jitter.

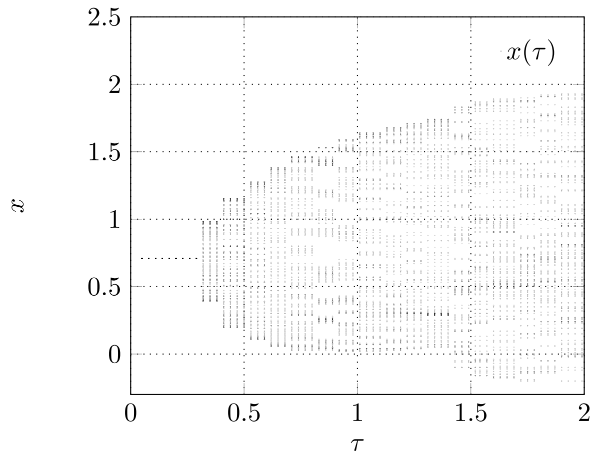

Figure 34: $\tau = var$.

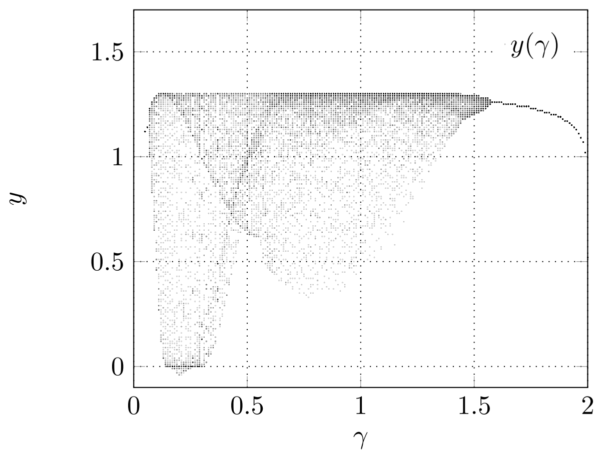

Figure 35: $\gamma = var$.

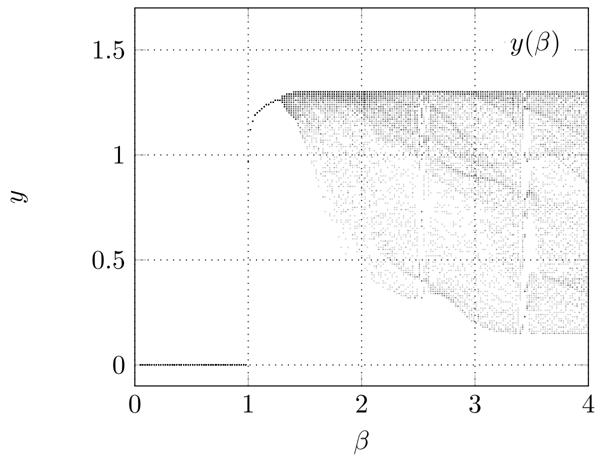

Figure 36: $\beta = var$.

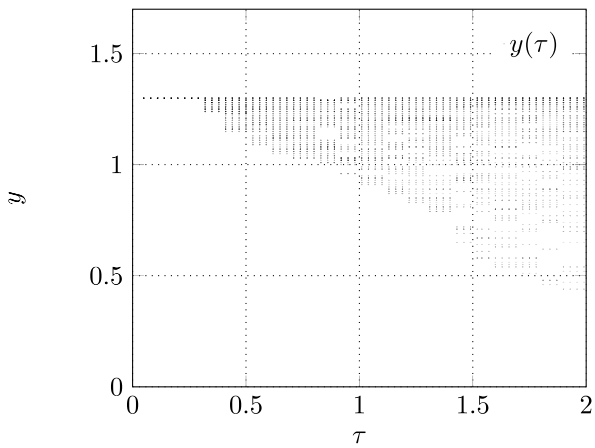

Figure 37: $\tau = var$.

These findings support the conclusion that the proposed chaotic system offers a more stable yet still strongly chaotic alternative to the classical Mackey–Glass system, particularly suitable for embedded implementations. Because the proposed system exhibits reduced parameter fragility, small deviations caused by ADC/DAC quantization, finite precision arithmetic, clock jitter in sampling period or limited resolution of system parameters do not significantly suppress the chaotic regime.

3.4. Influence of sampling period on stability and chaotic fidelity

All above-shown simulation results are obtained for the sample time $T_s = 10\ ms$. At the same time, it is clear, that a smaller sampling period improves the fidelity of the discrete-time realization to the continuous model but increases CPU load on the Arduino Due. This fact raises an important problem of numerical stability analysis of the studied system in term of sample time T_s.

We offer to perform this analysis by studying eigenvalues of discrete-time implementation of dynamical system which is linearized near stable point x_0, y_0

$$\dot{x} = a_{11}x + a_{12}y;\ \dot{y} = a_{21}x + a_{22}y, \tag{46}$$

here factors a_{ij} are components of system Jacobian which are taken in stable point

$$\mathbf{J} = \begin{pmatrix} \left.\frac{\partial f_1(x,y)}{\partial x}\right|_{x_0,y_0} & \left.\frac{\partial f_1(x,y)}{\partial y}\right|_{x_0,y_0} \\ \left.\frac{\partial f_2(x,y)}{\partial x}\right|_{x_0,y_0} & \left.\frac{\partial f_2(x,y)}{\partial y}\right|_{x_0,y_0} \end{pmatrix} = \begin{pmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{pmatrix}, \tag{47}$$

where $f_i(x, y)$ are left-hand expressions in (42).

One can define the stable point if he assumes in (42) $\dot{x} = \dot{y} = 0$, $y = y_\tau$ and solves

$$\begin{gathered} 0 = -\left(\sin\left(\arccos\left(-0.5 + 0.5\sqrt{1+4x}\right)\right)\sqrt{1+4x}\right) \times \\ \times\left(\gamma \arcsin(a_3y^3 + a_1y) + \beta\frac{\arcsin(a_3y^3+a_1y_)}{1+\arcsin(a_3y^3+a_1y)^n}\right); \\ 0 = (2\cos^2(\arcsin(a_3y^3 + a_1y)) + \cos(\arcsin(a_3y^3 + a_1y)) - 1) \times \\ \times\left(\gamma \arcsin(a_3y^3 + a_1y) + \beta\frac{\arcsin(a_3y^3+a_1y)}{1+\arcsin(a_3y^3+a_1y)^n}\right); \end{gathered} \tag{48}$$

for x and y with the above-defined system parameters.

This solution gives us set of various roots most of them are defined in the complex plane and thus should not be considered due to the real nature of signals x and y. Thus, only one real solution $x = 1,\ y = -0.11$, which corresponds the above-shown system motions, can be taken into further studies.

Now let us find factors a_{ij} by defining the derivatives in (47), substituting coordinates of stable point into their formulas, and equalizing the derivatives with corresponding factors

$$a_{11} = 0.012;\ a_{21} = 2.52;\ a_{21} = 0.11;\ a_{22} = -2.84. \tag{49}$$

These factors define eigenvalues for the continuous-time system (42)

$$\lambda_1 = -2.934;\ \lambda_2 = 0.106. \tag{50}$$

The considered system has both positive and negative eigenvalues of linearized dynamical system which can show the complex motion of initial system.

Now we apply (17) to (46) and rewrite these equations into discrete-time form

$$x = z^{-1}x(1 + T_sa_{11}) + T_sa_{12}z^{-1}y;\ y = T_sa_{21}z^{-1}x + (1 + T_sa_{22})z^{-1}y. \tag{51}$$

One can define i-th eigenvalue for this system by using known eigenvalue of continuous time system in such a way

$$\mu_i = 1 + T_s\lambda_i \tag{52}$$

It is clear that discrete system's solution is stable if following condition becomes true

$$|1 + T_s\lambda_a| < 1, \tag{53}$$

where λ_a is eigenvalue which takes maximal absolute value.

This condition enables to define following expression for sample time T_s

$$T_s < \frac{2}{|\lambda_a|} \text{ or } T_s < \frac{2}{|\lambda_1|} \text{ or} T_s < 0.68. \tag{54}$$

It is clear that this value is much greater when the used sample time. Thus, the problem of solution stability should not be raised for the considered system with given factors while MCU with small time of working cycle is used. Unstable solution can be reached if one increases the maximal system eigenvalue more than 70 time. However, such a raising cannot be physicaly reached in the studied system with the considered parameters.

4. Conclusion

Our paper proves the possibility of designing a chaotic system with a predefined phase portrait by by treating the parametric equations of the desired attractor as observability equations for the well-known or novel-designed chaotic systems, whose coordinates can be interpreted in different coordinate systems. The differentiating of the observability equations allows us to define a novel chaotic system, which can have as many outputs as the number of observability equations employed. One can use this fact to change the order of the designed system. Our approach shows the possibility of defining separate and interconnected system outputs. Since we consider the system transformation from polar to Cartesian coordinates, the main feature of the studied systems is the use of trigonometric functions to define their motion equations, which extend the class of nonlinearities to produce chaotic oscillations.

Since the system motions dramatically depend on the multiplication factor, which enables scaling the system angular position and use various ranges of trigonometric functions, we see the future development of our work in the design and investigation of chaotic systems that employ nonlinear scaling functions to transform the system angular position to desired ranges of trigonometric functions' arguments. We believe it will allow us to design novel systems with unique features and their quantitative chaos characteristic based on Lyapunov exponents and Kaplan–Yorke dimensions will be discussed. Also, it is possible to use our approach to design systems that produce attractors similar to known polar curves and study their cryptic features in various digital devices with different integration algorithms, discretization time, and data bus width.

Declaration on Generative AI

The authors have not employed any Generative AI tools.

References

[1] V. Chenchevoi, V. Kuznetsov, V. Kuznetsov, O. Chencheva, I. Zachepa, O. Chornyi, M. Kovzel, V. Kovalenko, M. Babyak, S. Levchenko, Development of mathematical models of energy conversion processes in an induction motor supplied from an autonomous induction generator with parametric non-symmetry, Eastern-European Journal of Enterprise Technologies 4 (2021) 67–82. doi:10.15587/1729-4061.2021.239146.

[2] K. Cherednichenko, V. Ivannikova, O. Sokolova, I. Ostroumov, O. Sushchenko, Y. Averyanova, M. Zaliskyi, O. Solomentsev, Y. Bezkorovainyi, O. Holubnychyi, B. Kuznetsov, I. Bovdui, T. Nikitina, R. Voliansky, Simulation Modelling for Urban Transport Infrastructure Optimization in Ukraine, in: O. Prentkovskis, I. Yatskiv (Jackiva), P. Skačkauskas, M. Karpenko, M. Stosiak (Eds.), TRANSBALTICA XV: Transportation Science and Technology, Lecture Notes in Intelligent Transportation and Infrastructure, Springer Nature Switzerland, Cham, 2025, pp. 367–380. doi:10.1007/978-3-031-85390-6_35.

[3] R. Voliansky, O. Kluev, O. Sadovoi, O. Sinkevych, N. Volianska, Chaotic Time-variant Dynamical System, in: 2020 IEEE 15th International Conference on Advanced Trends in Radioelectronics, Telecommunications and Computer Engineering (TCSET), 2020, pp. 606–609. doi:10.1109/TCSET49122.2020.235503.

[4] M. Imran, F. Kunwar, A hybrid path planning technique developed by integrating global and local path planner, in: 2016 International Conference on Intelligent Systems Engineering (ICISE), 2016, pp. 118–122. doi:10.1109/INTELSE.2016.7475172.

[5] X. Li, T. Huang, H. Chen, Y. Zhang, L. Xu, Y. Liu, Path Planning of Mobile Robot Based on Dynamic Chaotic Ant Colony Optimization Algorithm, in: 2022 IEEE 10th International Conference on Information, Communication and Networks (ICICN), 2022, pp. 515–519. doi:10.1109/ICICN56848.2022.10006459.

[6] Y. Wang, S. Kang, C. Lan, Y. Liang, J. Zhu, H. Gao, A five-dimensional chaotic system with a large parameter range and the circuit implementation of a time-switched system, in: 2016 11th International Conference on Reliability, Maintainability and Safety (ICRMS), 2016, pp. 1–6. doi:10.1109/ICRMS.2016.8050133.

[7] P. H. O. Silva, E. G. Nepomuceno, A. Vitorino, S. A. M. Martins, A visual chaotic system simulation in Arduino platform controlled by Android app, in: 2017 IEEE World Engineering Education Conference (EDUNINE), 2017, pp. 62–66. doi:10.1109/EDUNINE.2017.7918183.

[8] X. Shan, X. Huang, Y. Fan, Generation of Multi-Scroll Chaotic Attractor in a Simple State-Dependent Switched System: A Threshold Tuning Approach, in: 2019 Chinese Control Conference (CCC), 2019, pp. 150–154. doi:10.23919/ChiCC.2019.8866023.

[9] Y. Yang, L. Huang, J. Xiang, Q. Guo, Three-Dimensional Sine Chaotic System With Multistability and Multi-Scroll Attractor, IEEE Transactions on Circuits and Systems II: Express Briefs 69 (2022) 1792–1796. doi:10.1109/TCSII.2021.3126227.

[10] S. M. Aziz, S. F. Al-Azzawi, A Novel Simple 6D Hyperchaotic System with Hidden Attractors (2022) 7–12. doi:10.1109/CSASE51777.2022.9759660.

[11] G. Grassi, A. A. Khennaoui, A. Ouannas, V.-T. Pham, Three-dimensional Chaotic Fractional Maps without Fixed Points: Dynamics, Coexisting Hidden Attractors and Hardware Implementation, in: 2023 International Conference on Fractional Differentiation and Its Applications (ICFDA), 2023, pp. 1–6. doi:10.1109/ICFDA58234.2023.10153160.

[12] S. Chen, D. Ding, Z. Yang, Multistability and Hidden Attractors in a Fractional-Order Memristive Tabu Learning Neuron Model and Its Medical Image Encryption, in: 2023 IEEE International Conference on Memristive Computing and Applications (ICMCA), 2023, pp. 1–5. doi:10.1109/ICMCA59770.2023.10481226.

[13] A. Pranolo, Parallel mathematical models of dynamic objects, International Journal of Advances in Intelligent Informatics 4 (2018) 120–131. doi:10.26555/ijain.v4i2.229.