

Delay difference feedback memristive map: dynamics, hardware implementation and application in path planning

Quanli Deng, Chunhua Wang, Yichuang Sun *Senior Member, IEEE*, Gang Yang

Abstract—The delay of state variable plays a crucial role in chaotic systems. However, it has not received sufficient attention in discrete memristor-based maps. This paper presents a study on the effects of delay feedback in the discrete memristive system, proposing a generalized delay difference feedback memristive map. The dynamical behaviors influenced by control parameters, delay length and initial conditions, are explored through four discrete memristive maps. The Kaplan-Yorke dimension is utilized as an indicator to investigate the chaotic dynamic variations induced by the delay length within memristive maps. Furthermore, digital circuits for the proposed systems are designed and implemented, with hardware experimental results that are consistent with numerical simulations, thereby verifying the effectiveness of the digital circuit-based system and providing a foundation for hardware-based delay difference system design. Additionally, the chaotic series are integrated into the particle swarm optimization for tackling obstacle avoidance path planning. The superiority of the designed delay difference feedback memristive maps is highlighted through comparisons with several classical chaotic maps, showcasing their enhanced performance in terms of the speed and cost efficiency in solving the path planning task.

Index Terms—discrete memristor, delay feedback, hyperchaos, hardware implementation, path planning.

I. INTRODUCTION

CHAOS is a ubiquitous phenomenon in nature emerging from nonlinear systems. Chaotic systems exhibit several unique properties, including sensitivity to initial conditions, topological transitivity, and a density of periodic orbits. These unique features have inspired extensive research across various academic fields and have led to applications in a wide range of industrial domains [1]–[3]. Chaotic systems can be simulated using two main approaches: differential equations for continuous time variables and difference equations for discrete time variables. Discrete-time chaotic maps offer several advantages in modeling chaotic phenomena, including

algorithmic simplicity, rapid computation, and flexibility in hardware implementation [4]–[6]. Consequently, research on discrete-time chaotic maps holds an significant position in both academic and industrial applications [7]–[9].

The memristor, a novel electronic component characterized by both nonlinearity and memory properties, has seen increasing applications in recent years such as in-memory computing [10]–[12], bio-inspired circuit [13]–[15], and chaotic systems [16]–[18], contributing significantly to advancements in these fields. Particularly, memristor models based on discrete-time maps have attracted widespread attention. Bao *et al.* proposed a circuit model comprising a sampling switch, a memristor, and a capacitor, constructing a discrete-time iterative map using the discrete Euler algorithm and exploring the chaotic characteristics of the model through various numerical simulations [19]. Fu *et al.* proposed a method that uses the difference in state variables as the input to the memristor to construct a memristor-based Hénon map model, based on which they subsequently designed the iterative map with analog circuits [20]. To address the issues of discontinuous chaotic intervals and low Lyapunov exponents in chaotic maps, Lai *et al.* introduced a memristor-based hyperchaotic map capable of generating cubic attractors and ultra-boosting behaviors, effectively increasing the complexity of the chaotic system [21]. Yuan *et al.* focused on networks composed of discrete memristors with various topological structures, proposing different network configurations coupled with chaotic maps via memristors, and demonstrated the large chaotic parameter space of the proposed models through numerical simulations [22]. Despite the vigorous research and promising results in the application of memristor in chaotic maps, several critical issues remain unresolved. Notably, the phenomenon of time delay, which is prevalent in dynamic systems, has not yet received adequate attention.

Time-delay is a common phenomenon in various physical systems, such as biological systems, chemical reactions, and mechanical systems. It arises due to various factors such as signal transmission, processing time, or physical distances. Time-delay can profoundly affect the stability and dynamic behavior of nonlinear systems. In particular, the introduction of time-delay feedback into chaotic systems can lead to the emergence of complex dynamics, including chaos, hyperchaos, and even hyperchaos with multiple positive Lyapunov exponents [23]–[25]. Unlike continuous systems, where the introduction of time delay results in an infinite-dimensional

Manuscript received Month xx, 2xxx; revised Month xx, xxxx; accepted Month x, xxxx. This work was supported in part by the National Natural Science Foundation of China under Grant 62271197, and in part by the Guangdong Basic and Applied Basic Research Foundation under Grant 2024A1515011910.

Quanli Deng, Chunhua Wang, and Gang Yang are with the College of Information Science and Engineering, Hunan University, Changsha, 410082, China (Corresponding author: Chunhua Wang, e-mail: wch1227164@hnu.edu.cn).

Chunhua Wang is also with the Greater Bay Area Institute for Innovation, Hunan University, Guangzhou, 511300, China.

Yichuang Sun is with the School of Engineering and Computer Science, University of Hertfordshire, Hatfield AL10 9AB, U.K.

state space, the incorporation of time delay in discrete-time maps expands the system's state space while maintaining it within finite dimensions [26]. Buchner and Zebrowski studied the stability and chaos of the Logistic chaotic map under delay feedback, and explained the reason why the control the delayed chaotic system failed due to enhancement chaotic dynamics. [27]. Gentile *et al.* investigated the impact of time-delay feedback in discrete-time maps through frequency domain analysis, deriving analytical conditions under which the system undergo period-doubling bifurcations and Neimark-Sacker bifurcations [28]. Liu and Miao introduced a method for incorporating time delays in discrete-time chaotic systems to eliminate the dynamical degeneration phenomenon observed in digital chaotic maps, successfully producing a delayed discrete chaotic map with high complexity [29]. Liu *et al.* further employed the time-delay feedback to propose a novel method for constructing hyperchaos based on one-dimensional chaotic maps combined with modulo operations, resulting in a chaotic system with enhanced complex properties [30]. It can be observed that the introduction of time-delay into discrete systems can facilitate the enhancement of chaotic dynamics with more intricate characteristics.

However, the aforementioned researches on time-delay chaotic map have not yet incorporated memristors. As a non-linear device, integrating of memristor into chaotic maps can enhance the dynamic characteristics of these systems. To date, time-delay memristive maps remain unreported. Consequently, in order to improve the dynamic characteristics of memristive chaotic maps, we propose the study of delay feedback memristive map. Firstly, we design a generalized delay difference feedback memristive map (DDFMM). A stability analysis of the equilibrium points of the DDFMM is conducted. Furthermore, four distinct discrete memristor models incorporating hyperbolic tangent, absolute value, quadratic, and exponential functions in their conductance relationships are utilized as case studies to explore the impact of delay feedback. Moreover, the dynamical behaviors are examined numerically through various perspectives, including phase portraits, Lyapunov exponents (LEs), bifurcation diagrams (BDs), and Kaplan-Yorke dimensions (D_{KY}). The influence of delay on dynamics of discrete memristive systems is intuitively captured by D_{KY} , where an increase in D_{KY} signifies enhanced dynamical behaviors. Calculations reveal that for all four DDFMMs, D_{KY} consistently increases with the delay length, indicating a continuous rise in the dynamical behaviors of these systems. Finally, the obstacle avoidance path planning task is tackled by DDFMMs generated chaotic series-based optimization algorithm, demonstrating the superiority of DDFMMs in enhancement of chaotic dynamics. The contributions of this article are outlined as follows.

1) The time-delay induced dynamical behavior enhancement in discrete memristive system is studied, with four maps employing distinct memristor models serving as illustrative case studies.

2) Hyperchaotic attractors generated by the DDFMMs are experimentally implemented on a Field-Programmable Gate Array platform.

3) The DDFMM-based particle swarm optimization tack-

les the obstacle avoidance path planning task, showcasing the DDFMMs' superior randomness compared to traditional chaotic maps.

The remainder of this study is organized as follows. Section II describes the discrete memristor models and the generalized delay difference feedback memristive map. Section III investigates the dynamical behaviors of the delay difference feedback map across four different memristor models. Section IV focuses on the digital implementation using FPGA technology. Section V designs the DDFMM-based particle swarm optimization for obstacle avoidance path planning. Finally, Section VI summarizes the whole work and provides further research directions.

II. MODEL DESCRIPTION

A. Discrete memristor models

From the perspective of circuit theory, a memristor is a two-terminal nonlinear device that can exhibit pinched hysteresis loops under periodic stimulus [31]. The discretized memristor (DM) model can be obtained by applying the Euler difference method to continuous memristor formulations. An ideal discrete memristor can be mathematically represented by

$$\begin{cases} v_n = M(q_n)i_n \\ q_{n+1} = q_n + i_n \end{cases} \quad (1)$$

where v_n , i_n , and q_n are the sampling values of voltage, current, and charge at the n -th iteration, and $M(q_n)$ represents the sampling value of memristance at the n -th iteration.

The following investigation delves into the dynamics of memristive maps when subjected to delay difference feedback, employing four distinct differential memristor models. These discrete memristor formulations encompass the hyperbolic tangent function-based memristor (T-DM), the absolute value function-based memristor (A-DM), the quadratic function-based memristor (Q-DM), and the exponential function-based memristor (E-DM). The mathematical representations of these four DM models are listed in Table I. Notably, the formulations for A-DM and Q-DM stem from the work presented in [32]. For a comprehensive understanding of the dynamical behaviors specific to the A-DM and Q-DM, readers can refer to the content in [32].

TABLE I
MATHEMATICAL MODELS OF DMS

DM name	DM equation
T-DM	$\begin{cases} v_n = (\tanh(q_n - 1) \\ - \tanh(q_n + 1) + 0.5)i_n \\ q_{n+1} = q_n + i_n \end{cases}$
A-DM	$\begin{cases} v_n = (\text{abs}(q_n) - 1)i_n \\ q_{n+1} = q_n + i_n \end{cases}$
Q-DM	$\begin{cases} v_n = (2.4q_n^2 - 0.8)i_n \\ q_{n+1} = q_n + i_n \end{cases}$
E-DM	$\begin{cases} v_n = (0.025q_n - e^{-q_n^2})i_n \\ q_{n+1} = q_n + i_n \end{cases}$

Drawing upon the test scheme outlined in [33], we undertake the hysteresis loop analysis in the v - i plane by connecting a discrete current source with the input terminals of each

of the four differential DM models. Specifically, we employ a discrete current signal denoted as $i_n = 0.1 \sin(\omega n)$, where n represents the discrete time index. By varying the frequency ω , we numerically simulate the resulting voltage-current relationships using MATLAB and present the outcomes in Fig.1. The observations reveal that all four DM models are capable of exhibiting the hysteresis loops centered at the origin in the v - i plane. Furthermore, as the frequency ω increases, the area enclosed by the hysteresis loop lobes decreases monotonically. These numerical findings conclusively manifest that these four DM models possess the characteristic properties of memristors.

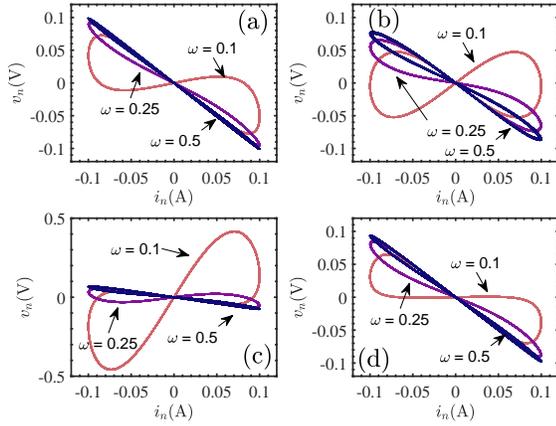


Fig. 1. Frequency-dependent pinched hysteresis loops of four DMs by applying a discrete current $i_n = 0.1 \sin(\omega n)$ A (a) T-DM, (b) A-DM, (c) Q-DM, and (d) E-DM.

B. Delay difference feedback memristive mapping model

Given their unique nonlinear characteristics, DMs have been harnessed to generate chaotic maps. In this study, we introduce the delay effect into the DM-based maps, resulting in a novel structure known as delay difference feedback memristive map (DDFMM), as illustrated in Fig.2. This framework involves utilizing the difference between the state variable x at the current iteration n and its value from D iteration prior, $n - D$, as the input to the DM. Subsequently, the output of the DM, scaled by a proportional controller k , serves as the input for the subsequent iteration of the model. By adopting this schematic structure, the delay effects on the DM maps are explored.

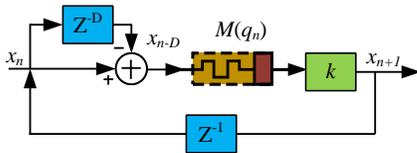


Fig. 2. Schematic structure of the DDFMM.

With the schematic structure depicted in Fig.2, where Z^{-D} represents the discrete delay of D iteration steps and Z^{-1} denotes one iteration step delay. The equation of the simple

delay difference feedback memristive map can be described as

$$\begin{cases} x(n+1) = kM(q(n))(x(n) - x(n-D)) \\ q(n+1) = q(n) + x(n) - x(n-D) \end{cases} \quad (2)$$

where k denotes a proportional controller, D is the delay length, x_n and y_n are the state variables at the n -th iteration.

The map (2) can be formally expanded [27] to a set of $D+2$ equations as

$$\begin{cases} x(n+1) = kM(q(n))(x(n) - z_1(n)) \\ q(n+1) = q(n) + x(n) - z_1(n) \\ z_1(n+1) = z_2(n) \\ \dots \\ z_{D-1}(n+1) = z_D(n) \\ z_D(n+1) = x(n) \end{cases} \quad (3)$$

The stability of a discrete iteration map can be characterized using its fixed points. According to (3), the fixed points of the system can be calculated by

$$\begin{cases} x = kM(q)(x - z_1) \\ q = q + x - z_1 \\ z_1 = z_2 \\ \dots \\ z_{D-1} = z_D \\ z_D = x \end{cases} \quad (4)$$

Clearly, the fixed point can be expressed as

$$S = (x^*, q^*, z_1^* \dots z_D^*) = (0, \eta, 0) \quad (5)$$

where η represents an arbitrary constant. The Jacobian matrix at the fixed point S can be derived as

$$J = \begin{bmatrix} kM(\eta) & 0 & -kM(\eta) & 0 & 0 & \dots & 0 \\ 1 & 1 & -1 & 0 & 0 & \dots & 0 \\ 0 & 0 & 0 & 1 & 0 & \dots & 0 \\ 0 & 0 & 0 & 0 & 1 & \dots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & 0 & 0 & 0 & 0 & \dots & 0 \end{bmatrix} \quad (6)$$

The characteristic polynomial can be deduced from the Jacobian matrix at S as

$$P(\lambda) = (1-\lambda)[(-\lambda)^D(kM(\eta) - \lambda) + (-1)^{D+7}kM(\eta)]. \quad (7)$$

In the given equation (7), $\lambda_1 = 1$ is explicitly stated, and the remaining eigenvalues $\lambda_{2 \dots (D+2)}$ are solutions to the polynomial $(-\lambda)^D(kM(\eta) - \lambda) + (-1)^{D+7}kM(\eta) = 0$. According to the theory of stability of iteration maps [34], a fixed point is considered stable if the absolute values of all eigenvalues of the Jacobian matrix at the fixed point are less than 1. In this condition, we have one eigenvalue λ_1 which lies exactly on the unit circle. The stability of the fixed point, therefore, depends on the locations of the other eigenvalues $\lambda_{2 \dots (D+2)}$ relative to the unit circle. The positions of these eigenvalues are influenced by the proportional controller gain k , the delay length D and memristor function $M(\eta)$. Therefore, the fixed points for the model may be unstable or critically stable. This underscores the importance of carefully selecting these parameters to achieve desired dynamical behaviors in the delay difference feedback memristive map.

III. NUMERICAL SIMULATION RESULTS

We substitute the different forms of DM models in Table I into (3) to obtain four distinct DDFMM models. These models are denoted as T-DDFMM, A-DDFMM, Q-DDFMM, and E-DDFMM, respectively.

A. Delay length-dependent dynamics

Set the parameter $k=1$ and initial values for x and q both as 0.1. Keep the history values of the delay variables at zero. Numerical simulations are conducted by varying the delay length D . The resulting attractors are depicted in Fig.3, obtained through iterating the discrete maps for 5000 steps. For the T-DDFMM, quasi-periodic attractors are observed when the delay length D is set to 1, and 2. Continuing to increase D to 3, the system transitions to exhibiting chaotic attractor. For the A-DDFMM, Q-DDFMM and E-DDFMM models, increase in D induces a sequential transition from periodic to quasi-periodic and subsequently to chaotic dynamics.

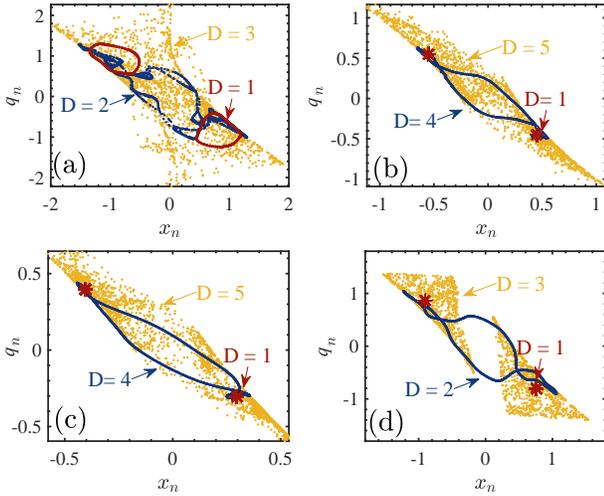


Fig. 3. Phase portraits of attractors of (a) T-DDFMM with $D=1$ (quasi-periodic), 2(quasi-periodic), 3(chaotic); (b) A-DDFMM with $D=1$ (stable), 4(quasi-periodic), 5(chaotic); (c) Q-DDFMM with $D=1$ (stable), 4(quasi-periodic), 5(chaotic); (d) E-DDFMM with $D=1$ (stable), 2(quasi-periodic), 3(chaotic).

The Kaplan-Yorke dimension serves as a critical metric for analyzing dynamical systems, particularly within the domain of chaos theory. It provides a quantitative evaluation of the fractal dimension of attractors present in the systems. The D_{KY} , which is derived from the Lyapunov exponents λ , can be computed using the following formula

$$D_{KY} = k + \frac{\sum_{i=1}^k \lambda_i}{|\lambda_{k+1}|} \quad (8)$$

where k is the index of the largest LE that is positive, and λ_i represents the i -th LE.

The D_{KY} of the four DDFMMs, calculated by varying the delay length D , are illustrated in Fig.4. Each plot displays numerical result for D_{KY} as dots, while the accompanying line represents the polynomial fitting curve. The fitted parameters for these curves are summarized in Table II. From the results

of polynomial fitting, it can be observed that the relationship between the delay length D and the D_{KY} aligns closely with a first-order linear relationship. The systems exhibit increasingly pronounced nonlinear dynamics as D grows, highlighting the intricate interplay between the delay length and the dynamics.

According to the definition of LE, it is given by the equation:

$$LE_i = \lim_{N \rightarrow \infty} \frac{1}{N} \ln(|\lambda_i(\Phi_N)|) \quad (9)$$

where $\lambda_i(\Phi_N)$ represents the i -th eigenvalue of the matrix $\Phi_N = J_0 J_1 \dots J_{N-1}$, with J denoting the Jacobian matrix of the system (3). Utilizing the algebraic theory, the determination of a matrix is equivalent to the product of its eigenvalues, leading to

$$|J| = \prod_{i=1}^{D+2} \lambda_i(J) = (-1)^{D+2} (A - B) \quad (10)$$

where $A = kM(q)$, $B = kM'(q)(x - z_1)$. Assuming square metrics M_1, M_2, \dots, M_N are of the same dimension, we derive the equation of $|M_1 M_2 \dots M_n| = |M_1| \cdot |M_2| \dots |M_n|$. Consequently, we can obtain:

$$\begin{aligned} |\Phi_N| &= \prod_{i=1}^{D+2} \lambda_i(\Phi_N) \\ &= |J_0| \cdot |J_1| \dots |J_{N-1}| \\ &= (-1)^{(D+2)N} (A - B)^N \end{aligned} \quad (11)$$

Combing (9) with (11), we have

$$\begin{aligned} \sum_{i=1}^{D+2} LE_i &= \sum_{i=1}^{D+2} \lim_{N \rightarrow \infty} \frac{1}{N} \ln |\lambda_i(\Phi_N)| \\ &= \lim_{N \rightarrow \infty} \frac{1}{N} \ln \left| \prod_{i=1}^{D+2} \lambda_i(\Phi_N) \right| \\ &= \lim_{N \rightarrow \infty} \frac{1}{N} \ln |(-1)^{(D+2)N} (A - B)^N| \\ &= \ln |A - B| \end{aligned} \quad (12)$$

Applying the same analytical approach, we can determine the total value of LEs for the system with an increased delay length D to $(D + \Delta D)$, where ΔD represents the increment. The calculation is presented as follows:

$$\begin{aligned} \sum_{i=1}^{(D+\Delta D)+2} LE_i &= \sum_{i=1}^{(D+\Delta D)+2} \lim_{N \rightarrow \infty} \frac{1}{N} \ln |\lambda_i(\Phi_N)| \\ &= \lim_{N \rightarrow \infty} \frac{1}{N} \ln \left| \prod_{i=1}^{(D+\Delta D)+2} \lambda_i(\Phi_N) \right| \\ &= \lim_{N \rightarrow \infty} \frac{1}{N} \ln |(-1)^{((D+\Delta D)+2)N} (A - B)^N| \\ &= \ln |A - B| \end{aligned} \quad (13)$$

The sum of LEs after increasing D by ΔD can also be expressed as:

$$\sum_{i=1}^{(D+\Delta D)+2} LE_i = \sum_{i=1}^{D+2} LE_i + \sum_{i=D+3}^{(D+\Delta D)+2} LE_i \quad (14)$$

Combining (12), (13) with (14), we have

$$\sum_{i=D+3}^{(D+\Delta D)+2} LE_i = 0 \quad (15)$$

From the characteristic equation of the Jacobian matrix, increasing in D does not result all eigenvalues being equal to 1, which implies that not all of the increased LE will be zero. It can be inferred that two cases satisfying the equation (15) as follows:

$$\begin{cases} LE_{D+3} = 0 & \text{for } \Delta D = 1 \\ \exists LE_i > 0, i \in [D+3, (D+\Delta D)+2] & \text{for } \Delta D > 1 \end{cases} \quad (16)$$

Therefore, when the increment $\Delta D > 1$, the values of $LE_{D+3} \dots LE_{(D+\Delta D)+2}$ must include at least one positive number. Consequently, as D increase, the count of positive LEs increases, ultimately leading to an increase of D_{KY} with increasing D .

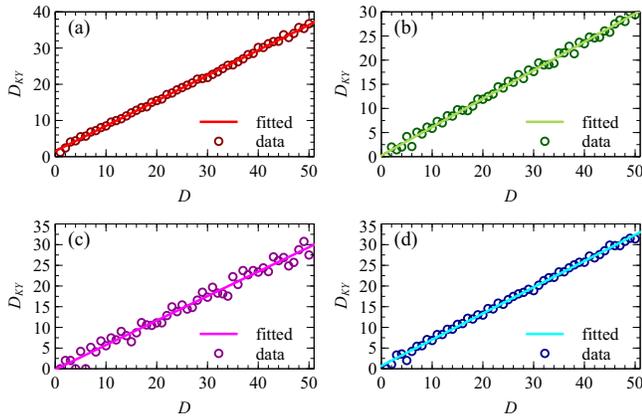


Fig. 4. Delay length dependent Kaplan-Yorke dimension for (a) T-DDFMM, (b) A-DDFMM, (c) Q-DDFMM, and (d) E-DDFMM.

TABLE II
PARAMETERS OF THE POLYNOMIAL FITTED CURVE

DM name	slope	y-intercept
T-DM	1.55	0.68
A-DM	0.25	0.57
Q-DM	0.16	0.58
E-DM	0.71	0.62

B. Parameter-dependent dynamics

With the delay length fixed at 5 and the initial conditions for state variables x and q set to 0.1 and 0.1, the historical values of the delay variables are maintained at zero. The dynamics influenced by the proportional controller k are examined through the Lyapunov exponent spectrum and bifurcation diagram of state variable x . The LEs are calculated utilizing the Wolf's Jacobian matrix-based algorithm, which provides a rapid and accurate numerical computation. The bifurcation diagram is obtained by evaluating the local maximum value of the state variable x . The numerical simulation results are presented in

Fig.5, where for clarity, only the first three LEs are displayed. For the T-DDFMM, as depicted in Fig.5(a), periodic attractors emerge when k is set within the interval $[0.5, 0.68]$. Quasi-periodic attractors are observed for k values in the range $(0.68, 0.79]$. A chaotic attractor with one positive LE arises for k in the range $(0.79, 0.8]$, and hyperchaotic attractors with two positive LEs are produced in the range $(0.8, 1]$, with the exception of the periodic window at $k \in [0.85, 0.865]$. In the case of the A-DDFMM, as shown in Fig.5(b), a chaotic region is identified for $k \in [0.947, 0.96]$ and a hyperchaotic region with two positive LEs for $k \in [0.97, 1]$. The A-DDFMM, as illustrated in Fig.5(c), exhibits a narrow chaotic parameter range for $k \in [0.96, 1]$. Within this range, chaotic attractors with a single positive LE are generated for k in $[0.96, 0.972]$, and hyperchaotic attractors with two positive LEs are produced within $k \in [0.972, 0.987] \cup [0.995, 1]$. For the E-DDFMM, as shown in Fig.5(d), chaotic behavior is observed for k in the interval $[0.867, 0.965]$, and hyperchaotic attractors with two positive LEs are obtained for k in the range $[0.965, 0.997]$. The bifurcation diagrams for variable x corroborate the findings from the LEs, thereby validating the correctness of the numerical simulations of LEs.

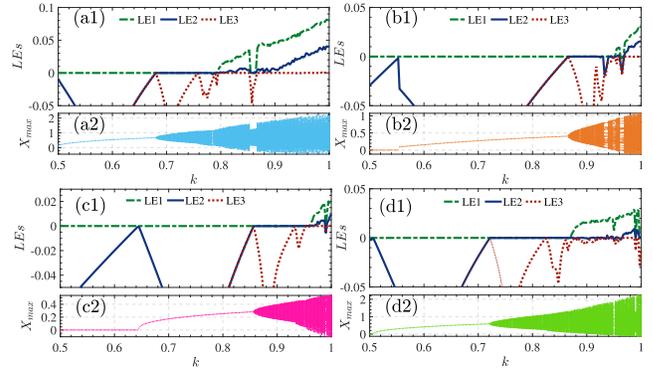


Fig. 5. Parameter k -relied LEs and state variable x bifurcation diagrams with $D=5$, for (a) T-DDFMM, (b) A-DDFMM, (c) Q-DDFMM (d) E-DDFMM.

The variation of D_{KY} with respect to parameter k is illustrated in Fig.6. By observing of the figure, it can be seen that as the parameter k increase, the system changes dynamically from periodicity to chaos. The variation of D_{KY} with k is consistent with the variation of LEs. For further investigating the variation of D_{KY} with respect to parameters, a two dimensional map is calculated by varying D and k , as depicted in Fig.7. In this figure, the color intensity of the space corresponds to the magnitude of D_{KY} , with brighter shades of blue indicating the larger values of D_{KY} . It is evident from the visualization that an increase in both D and k generally results in a trend towards the maximum values of D_{KY} . This observation suggests that the geometric complexity and fractal nature of the chaotic attractors within the DDFMMs are influenced by the simultaneous increase in the delay length and the proportional control parameter, indicative of a more intricate dynamical behavior as these parameters are adjusted.

In addition to using the methods of D_{KY} and LEs to capture the complexity of the system, we can further quantify it by

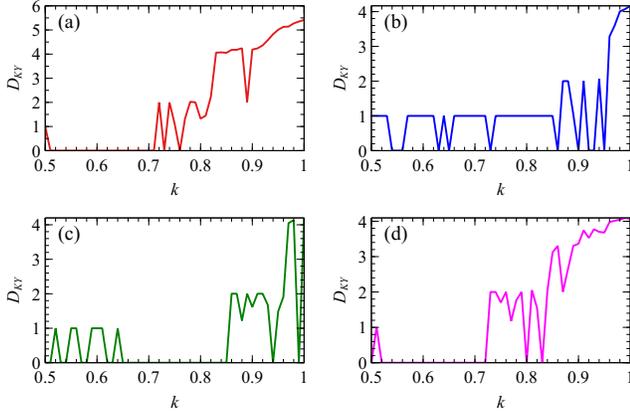


Fig. 6. Parameter k -relied D_{KY} with $D=5$, for (a) T-DDFMM, (b) A-DDFMM, (c) Q-DDFMM (d) E-DDFMM.

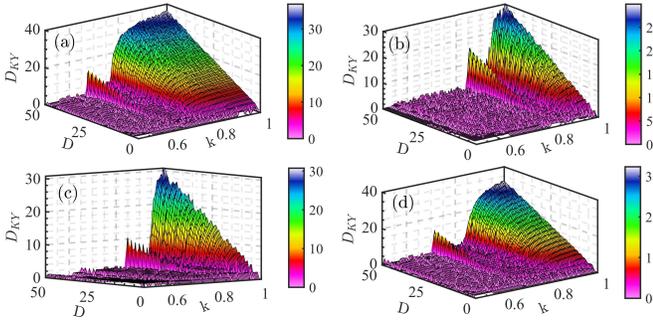


Fig. 7. Delay length and parameter k -relied Kaplan-Yorke dimensions for (a) T-DDFMM, (b) A-DDFMM, (c) Q-DDFMM (d) E-DDFMM.

calculating the entropy of the system's state variables. In this study, we select spectral entropy and C0 complexity, two indicators for assessing the complexity of chaotic sequences, to investigate how variations in the parameter k influence the system's complexity. Fig.8 illustrates the effects of modifying the parameter k on the spectral entropy and C0 complexity of the four DDFMMs. The results indicate that the spectral entropy and C0 complexity of the DDFMMs increase with the parameter k . This suggests that the complexity of the DDFMMs is enhanced as the parameter k is adjusted, which is consistent with the results obtained from the D_{KY} and LEs analyses.

C. Initial value-dependent dynamics

With the delay length D and parameter k set to 5 and 1, respectively, and the historical values of the delay variables maintained at zero, the dynamics that depend on the initial values of state variables are examined. Distinct values for the state variables are chosen to investigate their influence on the dynamical behaviors. Fig.9 illustrates the coexisting attractors when $x(0)=0.1$ and $q(0)=\pm 0.5$, respectively. The figure demonstrates that coexisting symmetric chaotic attractors can be obtained in all four DDFMMs by selecting appropriate

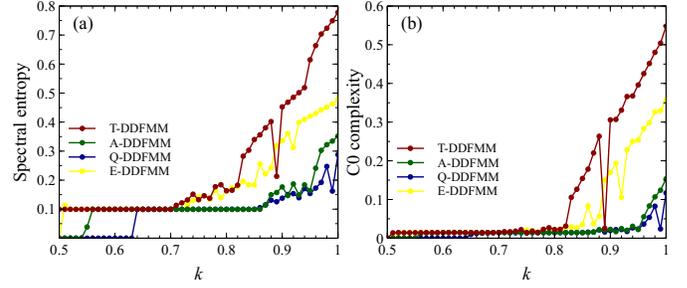


Fig. 8. Results of complexity variation with respect to k reflected by (a) spectral entropy, (b) C0 complexity.

initial conditions. In the figure, the attractors depicted in light blue, referred to as downward chaos (DCH), represent the trajectories that originate from $q(0)=-0.5$. Conversely, the attractors portrayed in bright purple, named as upward chaos (UCH), correspond to trajectories initiated from $q(0)=0.5$.

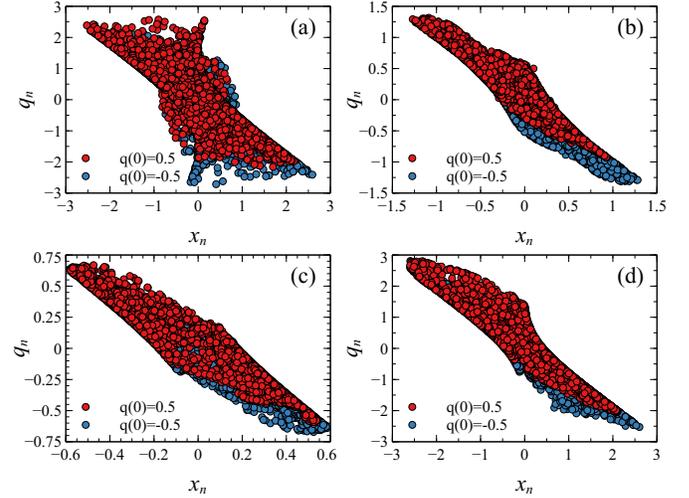


Fig. 9. Coexisting symmetric chaotic attractors for (a) T-DDFMM, (b) A-DDFMM, (c) Q-DDFMM (d) E-DDFMM, where light blue dots indicate trajectories initiating from $q(0)=-0.5$ and bright purple dots represent trajectories initiating from $q(0)=0.5$.

For the specified parameter settings, the attraction basins of the DDFMMs are drawn by examining every initial state in the $x(0)-q(0)$ plane. The results of this analysis are graphically represented in Fig.10. In this figure, the bright and dark blue regions correspond to the downward chaos and upward chaos, respectively. The orange region indicates the presence of stable points, while the white region in Fig.10(c) denotes unbounded regions. It is evident that the coexistence of symmetric chaotic attractors in the DDFMMs is significantly influenced by the initial values of $q(0)$.

IV. FPGA-BASED HARDWARE IMPLEMENTATION

The implementation of chaotic maps in FPGA-based digital circuits has garnered significant interest due to the advantages, such high computational speed, robust stability, and the ease

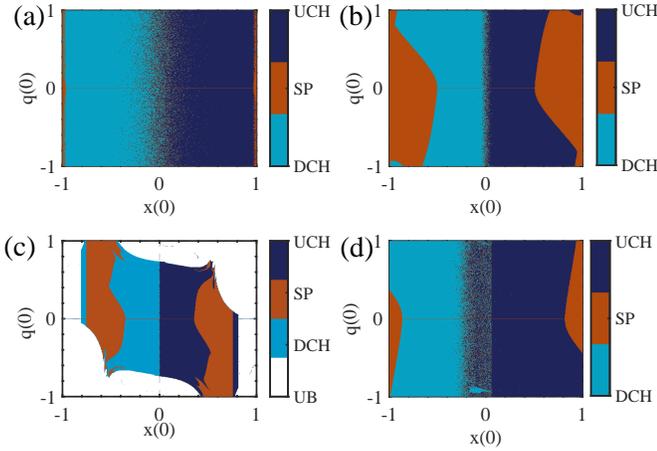


Fig. 10. Basins of attraction for (a) T-DDFMM, (b) A-DDFMM, (c) Q-DDFMM (d) E-DDFMM.

with altering system parameters and initial conditions [35]–[38]. This study presents the FPGA-based implementation of the proposed DDFMMs. The flow block diagram for the FPGA-based implementation of the model described by (3) is shown in Fig.11. There are three blocks in the FPGA-based hardware implementation. The DDFMM Module is used for iteration calculation of state variables. The Data Transfer Module is used for preparing data for digital to analog transformation. And the DAC Module is used for the analog output.

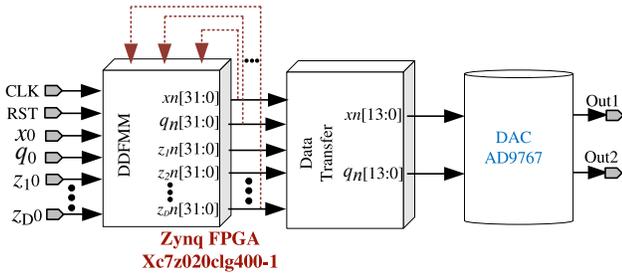


Fig. 11. Block diagram of FPGA-based DDFMM.

The iterative calculations and output preparation operations are executed on the Xilinx xc7z020clg400-1 platform. The designed discrete iteration systems operate at a clock frequency of 50 MHz, with the Verilog Hardware Description Language employed to develop the program code. A 32-bit fixed-point decimal format, comprising of 1 sign bit, 6 integer bits, and 25 decimal bits, is employed. As illustrated in Fig.9, it is noteworthy that the maximum values of stable variables x and q are both less than 4. Therefore, using an integer bit greater than 2 can avoid the risk of data overflow during the calculation process. To enhance the operational robustness, we adopt 6 integer bits, thereby guaranteeing that data overflow is avoided. Additionally, in addressing rounding errors, a consistent downward rounding approach is employed, discarding a portion exceeding 25 decimal bits. While this introduces some degree of error, it is acceptable for implementation of

this digital system. However, in specific applications, such as image encryption and secure communication, which have stringent requirements for data integrity, further optimization of the decimal processing is deemed necessary.

In the DDFMM Module, the state variables x and q at the n -th iteration are denoted as x_n and y_n , respectively. The delay state variables at the n -th iteration are represented by $z_1 n \dots z_D n$, with D being the delay length. For the hardware implementation, we chose a delay length of $D=5$, a parameter $k=1$ and initialized the values of x and y as 0.1 and 0.5, respectively. To ensure the orderly output of internal data within the module, we adopted a finite state machine approach to implement operations such as multiplication and addition in the DDFMM models. For multiplication, we employed fixed-point multiplication, which involves multiplying 32-bit signed numbers and storing the result in a 64-bit temporary variable. Subsequently, we truncated the sign bit and bits ranging from the 56th to the 25th position of this temporary variable to form the final 32-bit multiplication result. After completing the multiplication operation and function computations, all data is subjected to addition and subtraction according to the model's equation to obtain the final output result.

Furthermore, the hyperbolic tangent function of the T-DDFMM is achieved through an approximation technique as delineated in [39]. The absolute value function, a component of the A-DDFMM, is efficiently implemented in FPGA by assessing the sign bit of the state variable q_n . For the implementation of Q-DDFMM in FPGA, a fixed-point multiplication approach is adopted. The implementation of the E-DDFMM is similar to aforementioned methods, but one of its unique difficulties lies in achieving the exponential function while optimizing hardware resource expenditure. In this context, the methodology proposed in [40] is embraced, which facilitates the transformation of the exponential function with base e into one with base 2 for computation purposes.

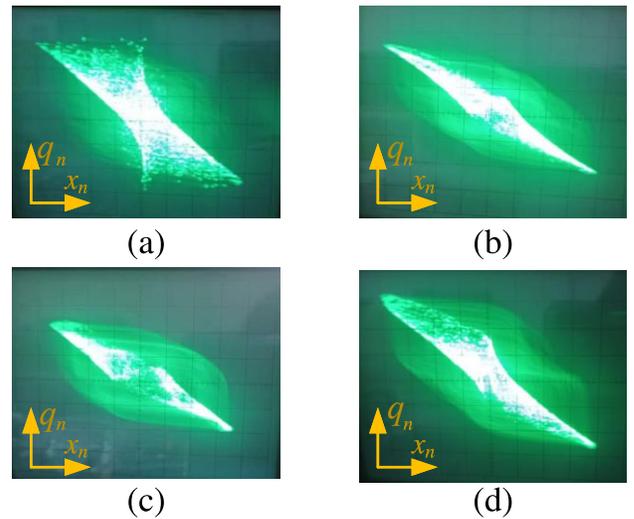


Fig. 12. FPGA-based implementation for (a) T-DDFMM, (b) A-DDFMM, (c) Q-DDFMM (d) E-DDFMM, with delay length $D=5$.

In the DAC module, we used the dual-port DAC device, AD9767, which has a precision of 14 bits. To meet this

requirements, we retain the sign bit and concatenate the bits 27 to 14 to form the output data of DAC.

The oscilloscope captured hardware experimental results for each DDFMM model are shown in Fig.12. Observations from the hardware implemented phase portraits in $xn - qn$ plane aligns with the numerical simulated red trajectories in Fig.9. These experimental results demonstrate the feasibility and simplicity of digital FPGA-based implementation, and also manifest the correctness of the mathematical models of DDFMMs.

V. APPLICATION IN CHAOTIC PSO-BASED PATH PLANNING

Obstacle avoidance path planning is a critical challenge for autonomous systems, requiring efficient and robust algorithm to navigate complex dynamic environments. Chaotic systems, characterized by their sensitivity to initial conditions, unpredictability, and ergodicity, offer unique advantages for generating diverse and adaptive trajectories. The randomness inherent in chaotic maps is utilized to initialize populations in optimization algorithm, significantly enhancing efficiency and enabling the algorithm to avoid local optima [41]. Furthermore, the integration of chaotic systems with dynamic window approaches have proven effective in improving the robot's navigation capabilities through narrow passages and in avoiding obstacles in dynamic environments [42]. Additionally, the variability introduced by chaotic systems aids in generating smoother trajectories, thereby avoiding sharp turns and abrupt changes in direction [43]. Therefore, integrating chaotic dynamics into algorithms for path planning can enhance the system's adaptability, robustness, and efficiency.

The particle swarm optimization (PSO) algorithm was inspired by the swarm behavior such as fish and bird schooling in nature [44]. The PSO-based algorithm has emerged as a promising approach in addressing obstacle avoidance path planning. However, traditional PSO algorithms often suffer from premature convergence and limited global search capabilities, which can hinder their effectiveness in highly nonlinear and multimodal search spaces. To address these limitations, chaotic PSO variants have been developed, leveraging the inherent randomness, ergodicity, and sensitivity to initial conditions of chaotic systems. These characteristics enable chaotic PSO to enhance global exploration, avoid local optima, and improve the diversity of solutions [45]. In this study, we propose a DDFMM-based PSO algorithm designed to execute the obstacle avoidance path planning tasks. By using chaotic sequences to initialize the initial position and initial velocity of the particle swarm, the random initialization characteristics of chaos are integrated into the PSO algorithm. During the iterative process of particle swarm optimization, each update of particle position incorporates a chaotic random sequence, introducing randomness into the update process to enhance the algorithm's ability to explore the state space.

Creating a swarm of particles with population of N , where each particle represents a potential solution. The initial values of particle population are crucial, as they influence the subsequent search trajectory and the quality of final solution. To ensure the ergodicity and randomness in the initialization

process, chaotic series generated by the DDFMM are utilized for setting the initial positions and velocities of the particles. Let the velocity and position of the i -th particle at time $t+1$ be denoted as v_i^{t+1} and x_i^{t+1} , respectively. Each particle maintains an individual best position ($pbest_i$) that it has visited so far during its search. Additionally, a global best position ($gbest$) is tracked, representing the best position found among all particles in the swarm. The particles move towards both their individual and global best positions with a certain probability, aiming to discover the authentic global optimum in the solution space. The equations for updating the position and velocity of the i -th particle are as follows.

$$\begin{aligned} v_i^{t+1} &= v_i^t + c_1 \cdot r_1 \cdot (pbest_i - x_i^t) + c_2 \cdot r_2 \cdot (gbest - x_i^t) \\ x_i^{t+1} &= x_i^t + v_i^{t+1} \end{aligned} \quad (17)$$

where learning factors c_1 and c_2 regulate the influence of the individual best position and global best position on the movements, respectively. The random number r_1 and r_2 , which lie between 0 and 1, introduce randomness in the solution search process. These random numbers are obtained by performing a modulo 1 operation on the state variable in DDFMM.

The convergence analysis of the PSO algorithm is a significant and intricate task. Clerc and Kennedy have introduced an analytical approach for assessing PSO convergence by simplifying the PSO system (17) down into a one-dimensional particle system [46]. During the convergence analysis, the randomness terms, which are confined within the range $[0,1]$, can be disregarded [46]. By applying the method proposed by Clerc and Kennedy, which involves substituting $c = c_1 + c_2$ and $u_t = p - x_{v+1}$ (where p is the global best position), we can obtain the following equation:

$$Y_{t+1} = AY_t \quad (18)$$

where

$$A = \begin{bmatrix} 1 & c \\ -1 & 1 - c \end{bmatrix} \quad (19)$$

The convergence of the PSO algorithm is determined by the eigenvalues of the matrix A , which are computed as follows:

$$\lambda_{1,2} = 1 - \frac{c}{2} \pm \frac{\sqrt{c^2 + 4c}}{2} \quad (20)$$

Crucially, when $c=4$, a bifurcation occurs in (18). The convergence of the PSO algorithm is guaranteed if $0 < c < 4$. To ensure the convergence of the PSO algorithm in this work, we set c_1 and c_2 both to 1.5. This choice of learning factors ensures that the PSO algorithm converges to the global optimum.

To demonstrate the global optimization capabilities of the DDFMM-based PSO algorithm, we conducted a simulation experiment in which the algorithm was used to find the minimum value of the Griewank function. Finding the global minimum of the function serves as evidence that the algorithm has obtained the global optimal solution [47]. The Griewank function is formulated as follows:

$$f(x) = 1 + \frac{1}{4000} \sum_{i=1}^n x_i^2 - \prod_{i=1}^n \cos\left(\frac{x_i}{\sqrt{i}}\right) \quad (21)$$

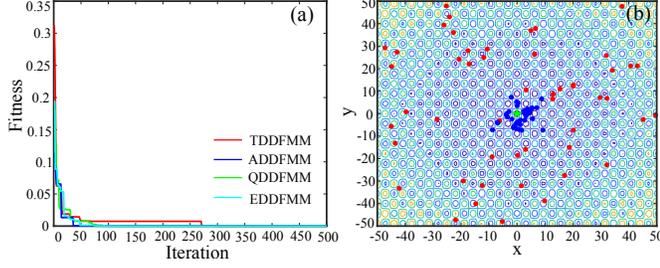


Fig. 13. Global optimization of the Griewank function using DDFMM-based PSO algorithm (a) fitness variations with iteration (b) particle initial positions with red color and final positions with blue color.

where n is the dimension of the search space. This function achieves its global minimum value of 0 when each x_i equals 0 [48]. The Griewank function is a multimodal function, featuring numerous local optima, which makes it is often used in the research of PSO algorithm to verify the global optimization ability of the algorithm [49]. The simulation results are shown in Fig.13. The fitness variations with iteration are depicted in Fig.13(a), which shows that the fitness value decreases to 0 within 500 iterations, indicating that the PSO algorithm is converging to the global optimum. For further illustrate, we take the EDDFMM-based approach as an example to illustrate the initial positions and the final converged positions of its particles in Fig.13(b). The particle initial positions are shown in red, while the final positions are displayed in blue in Fig.13(b). The final positions of the particles are concentrated around the global optimum, demonstrating the effectiveness of the DDFMM-based PSO algorithm in global optimization.

The procedure of the DDFMM-based PSO in obstacle avoidance path planning is outlined in Algorithm 1.

Algorithm 1 DDFMM-based PSO for Obstacle Avoidance Path Planning

- 1: Initialize positions and velocities of particle population with DDFMM.
 - 2: Set parameters: maximum number of iterations T , number of particles N , learning factors c_1, c_2 , inertia weight ω .
 - 3: **for** $t = 1$ to T **do**
 - 4: **for** each particle i **do**
 - 5: Evaluate the cost function C_i of particle i based on the path length and obstacle avoidance.
 - 6: **if** C_i is lower than the personal best $pbest_i$ **then**
 - 7: Update $pbest_i$ to the current position of particle i .
 - 8: **end if**
 - 9: **if** C_i is lower than the global best $gbest$ **then**
 - 10: Update $gbest$ to the current position of particle i .
 - 11: **end if**
 - 12: Update the velocity v_i and position x_i of particle i based on (17) with chaotic series.
 - 13: **end for**
 - 14: **end for**
 - 15: Return the path corresponding to the global best position $gbest$.
-

The performance evaluation in this application is carried out using a cost function that assesses three critical factors: the path length, the path smoothness and the degree of obstacle violations. These factors are combined to form the total cost, which guides the optimization process. The path length is a fundamental metric that measures the total distance traveled along the planned path. It is calculated by summing the Euclidean distance between consecutive points on the path:

$$L = \sum_{i=1}^{n-1} \sqrt{(x_{i+1} - x_i)^2 + (y_{i+1} - y_i)^2} \quad (22)$$

where (x_i, y_i) and (x_{i+1}, y_{i+1}) are the coordinates of consecutive points on the path. The path smoothness is another important factor that evaluates the curvature changes along the path, ensuring that the path is not overly jagged or erratic. It is calculated as the sum of the absolute difference in the second-order differences of the path coordinates:

$$S = \sum_{i=2}^{n-1} (|(x_{i+1} - 2x_i + x_{i-1})^2 + (y_{i+1} - 2y_i + y_{i-1})^2|). \quad (23)$$

This metric penalizes paths with high curvature changes, promoting smoother trajectories. The obstacle violation term ensures that the planned path does not intersect with any obstacles. This term is calculated differently for circular and polygonal obstacles. For circular obstacles, the violation degree is calculated based on the distance from each point on the path to the obstacle center. If the distance d is less than the obstacle radius r , a penalty is imposed

$$v = \max(1 - \frac{d}{r}, 0) \quad (24)$$

The total violation penalty for circular obstacle is the sum of these penalties for all points on the path that are within the obstacle radius. For polygonal obstacles, the violation is determined by checking if any segment of the path intersect with any edge of the polygon. For each path segment (x_i, y_i) to (x_{i+1}, y_{i+1}) and each edge of the polygon (x_j, y_j) to (x_{j+1}, y_{j+1}) , we use the line-line intersection formula to check for intersections:

$$t = \frac{(x_i - x_j) \cdot (y_j - y_{j+1}) - (y_i - y_j) \cdot (x_j - x_{j+1})}{(x_i - x_{i+1}) \cdot (y_j - y_{j+1}) - (y_i - y_{i+1}) \cdot (x_j - x_{j+1})} \quad (25)$$

$$u = \frac{(x_i - x_j) \cdot (y_i - y_{i+1}) - (y_i - y_j) \cdot (x_i - x_{i+1})}{(x_i - x_{i+1}) \cdot (y_j - y_{j+1}) - (y_i - y_{i+1}) \cdot (x_j - x_{j+1})}.$$

If $t \geq 0$ and $t \leq 1$ and $u \geq 0$ and $u \leq 1$, then the path segment intersects with the polygon edge. In this case, a predefined penalty value is added to the total violation penalty. The total violation penalty is the sum of violation penalties for both circular and polygonal obstacles. Therefore, total cost z is calculated by combining these factors with appropriate weight parameters:

$$z = L \cdot (1 + \beta \cdot \text{violation} + \gamma \cdot S) \quad (26)$$

where β and γ are weight parameters that balance the importance of avoiding obstacles and maintaining path smoothness, respectively. A higher value of β indicates a greater emphasis on obstacle avoidance, while a higher value of γ prioritizes path smoothness. These parameters are set as $\beta=0.7$ and $\gamma=0.3$,

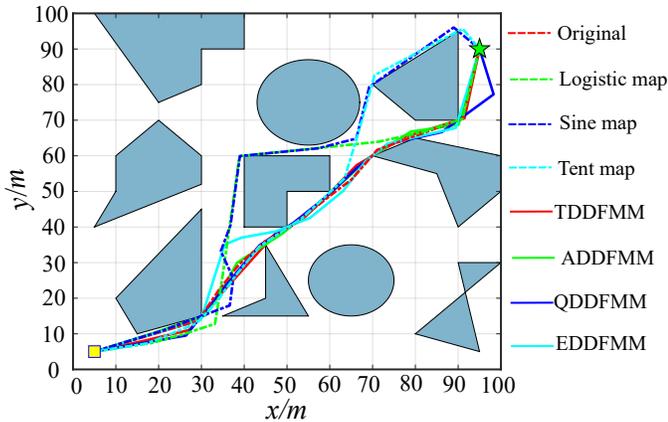


Fig. 14. Simulation of PSO-based obstacle avoidance path planning.

which indicate a prioritization of obstacle avoidance over path smoothness in this study.

Set the maximum number of iterations T to 50 and the number of particles N to 30. A simulation of the PSO-based path planning is conducted in MATLAB, where ten obstacles with random positions and shapes are placed within a 100×100 square space. Fig.14 demonstrates the trajectories obtained using original, classical chaotic map-based and DDFMM-based PSO algorithms, under the aforementioned parameter settings. It can be observed that all the paths in the figure successfully avoid obstacles, achieving the obstacle avoidance path planning functionality. A summary of the path length, smoothness, and success rates over 50 independent experiments for different methods is presented in Table III. The DDFMM-based method demonstrate superior performance in terms of path length, smoothness and success rate compared to the original PSO and other classical chaotic map-based PSO variants. For instance, the EDDFMM-PSO achieves the shortest path length and the highest success rate while maintaining a relatively low smoothness value. These results underscore the superior randomness and exploration capabilities of the DDFMMs, which enable the PSO algorithm to find more efficient and feasible paths in complex environments.

TABLE III
RESULTS COMPARISON FOR DIFFERENT METHODS BASED-PSO

Method	path length	smoothness	success rate
Original-PSO	150.217	154.242	0.80
Logistic-PSO	154.174	97.478	0.76
Sine-PSO	140.307	132.316	0.80
Tent-PSO	136.939	83.988	0.84
TDDFMM-PSO	132.438	57.368	0.90
ADDFMM-PSO	132.540	50.506	0.88
QDDFMM-PSO	136.189	76.781	0.92
EDDFMM-PSO	131.833	73.694	0.94

VI. CONCLUSION AND OUTLOOK

The time-delay effect in nonlinear systems can increase the number of positive LEs, thereby engendering hyperchaos

with multiple positive LEs and enriching the dynamical behaviors of nonlinear systems. In discrete memristive maps, the role of delay dynamics has not been thoroughly investigated. This study introduces a novel discrete memristive system with delayed feedback and examines its dynamical behaviors across four distinct memristor models. Numerical simulations, focusing on control parameters, delay length, and initial conditions, reveal that the dynamics of the system varies with increasing the delay length. Additionally, the proposed model is realized using a FPGA-based digital platform, with hardware experimental results substantiating the design's practicality and providing a foundation for subsequent hardware-oriented research on discrete delay memristive maps. This study also presents a DDFMM-based PSO algorithm to address the obstacle avoidance path planning. The simulation results demonstrate the superior randomness of the DDFMMs. Looking ahead, our future research will focus on the design of intricate memristive delay systems with multiple positive LEs and the investigation of chaos control strategies.

REFERENCES

- [1] L. Zhang, J. Riem, J. Chen, H. Mackay, T. Lan, N. D. Bastian, and G. C. Adam, "Multi-memristor based distributed decision tree circuit for cybersecurity applications," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 71, no. 8, pp. 3526–3537, 2024.
- [2] Q. Deng, C. Wang, Y. Sun, Z. Deng, and G. Yang, "Memristive tabu learning neuron generated multi-wing attractor with FPGA implementation and application in encryption," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 72, no. 1, pp. 300–311, 2025.
- [3] F. Yu, Y. Lin, W. Yao, S. Cai, H. Lin, and Y. Li, "Multiscroll Hopfield neural network with extreme multistability and its application in video encryption for IIoT," *Neural Networks*, vol. 182, p. 106904, 2025.
- [4] H. Bao, Y. Su, Z. Hua, Q. Xu, M. Chen, and B. Bao, "2-D threshold hyperchaotic map and application in timed one-time password," *IEEE Transactions on Industrial Informatics*, vol. 20, no. 11, pp. 13 058–13 068, 2024.
- [5] Y. Zhang, Z. Hua, H. Bao, and H. Huang, "Multi-valued model for generating complex chaos and fractals," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 71, no. 6, pp. 2783–2796, 2024.
- [6] W. Deng and M. Ma, "Analysis of the dynamical behavior of discrete memristor-coupled scale-free neural networks," *Chinese Journal of Physics*, vol. 91, pp. 966–976, 2024.
- [7] W. Feng, J. Zhang, Y. Chen, Z. Qin, Y. Zhang, M. Ahmad, and M. Wozniak, "Exploiting robust quadratic polynomial hyperchaotic map and pixel fusion strategy for efficient image encryption," *Expert Systems with Applications*, vol. 246, p. 123190, 2024.
- [8] X. Ma, Z. Wang, and C. Wang, "An image encryption algorithm based on Tabu search and hyperchaos," *International Journal of Bifurcation and Chaos*, vol. 34, no. 14, p. 2450170, 2024.
- [9] W. Feng, Q. Wang, H. Liu, Y. Ren, J. Zhang, S. Zhang, K. Qian, and H. Wen, "Exploiting newly designed fractional-order 3D Lorenz chaotic system and 2D discrete polynomial hyper-chaotic map for high-performance multi-image encryption," *Fractal and Fractional*, vol. 7, no. 12, p. 887, 2023.
- [10] Q. Deng, C. Wang, J. Sun, Y. Sun, J. Jiang, H. Lin, and Z. Deng, "Nonvolatile CMOS memristor, reconfigurable array, and its application in power load forecasting," *IEEE Transactions on Industrial Informatics*, vol. 20, no. 4, pp. 6130–6141, 2024.
- [11] L. Zhang, J. Riem, J. Chen, H. Mackay, T. Lan, N. D. Bastian, and G. C. Adam, "Multi-memristor based distributed decision tree circuit for cybersecurity applications," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 71, no. 8, pp. 3526–3537, 2024.
- [12] Q. Deng, C. Wang, and Z. Deng, "Memristive circuit of quaternion multiplication and its application in aircraft attitude computation," *IEEE Transactions on Circuits and Systems II: Express Briefs*, vol. 71, no. 8, pp. 3970–3974, 2024.
- [13] J. Sun, Y. Zhai, P. Liu, and Y. Wang, "Memristor-based neural network circuit of operant conditioning with bridging and conditional reinforcement," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 71, no. 8, pp. 3514–3525, 2024.

- [14] Z. Deng, C. Wang, H. Lin, Q. Deng, and Y. Sun, "Memristor-based attention network for online real-time object tracking," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 44, no. 2, pp. 684–695, 2025.
- [15] X. Li, J. Sun, Y. Sun, and J. Zhang, "A power-adaptive neuron model and circuit implementation," *Nonlinear Dynamics*, vol. 113, pp. 3763–3780, 2025.
- [16] D. Luo, C. Wang, Q. Deng, and Y. Sun, "Dynamics in a memristive neural network with three discrete heterogeneous neurons and its application," *Nonlinear Dynamics*, vol. 113, pp. 5811–5824, 2025.
- [17] X. Kong, F. Yu, W. Yao, S. Cai, J. Zhang, and H. Lin, "Memristor-induced hyperchaos, multiscroll and extreme multistability in fractional-order HNN: Image encryption and FPGA implementation," *Neural Networks*, vol. 171, pp. 85–103, 2024.
- [18] Z. Liao, J. Ouyang, and M. Ma, "Dula-attribute ring-star neural network," *Nonlinear Dynamics*, vol. Early Access, 2024.
- [19] B.-C. Bao, H. Li, H. Wu, X. Zhang, and M. Chen, "Hyperchaos in a second-order discrete memristor-based map model," *Electronics Letters*, vol. 56, no. 15, pp. 769–770, 2020.
- [20] L. Fu, X. Wu, S. He, H. Wang, and K. Sun, "A memristive Hénon map based on the state variable difference and its analog circuit implementation," *IEEE Transactions on Industrial Electronics*, vol. 71, no. 8, pp. 9668–9676, 2024.
- [21] Q. Lai, L. Yang, and G. Chen, "Design and performance analysis of discrete memristive hyperchaotic systems with stuffed cube attractors and ultraboosting behaviors," *IEEE Transactions on Industrial Electronics*, vol. 71, no. 7, pp. 7819–7828, 2024.
- [22] F. Yuan, S. Zhang, G. Xing, and Y. Deng, "Parameter control methods for discrete memristive maps with network structure," *IEEE Transactions on Industrial Informatics*, vol. 20, no. 5, pp. 7194–7204, 2024.
- [23] X. F. Wang, G.-Q. Zhong, K.-S. Tang, K. Man, and Z.-F. Liu, "Generating chaos in Chua's circuit via time-delay feedback," *IEEE Transactions on Circuits and Systems I: Fundamental Theory and Applications*, vol. 48, no. 9, pp. 1151–1156, 2001.
- [24] C. Cai, Z. Xu, and W. Xu, "Melnikov's analysis of time-delayed feedback control in chaotic dynamics," *IEEE Transactions on Circuits and Systems I: Fundamental Theory and Applications*, vol. 49, no. 12, pp. 1724–1728, 2002.
- [25] X. Du, M. Xiao, J. Qiu, Y. Lu, and J. Cao, "Stability and dynamics analysis of time-delay fractional-order large-scale dual-loop neural network model with cross-coupling structure," *IEEE Transactions on Neural Networks and Learning Systems*, vol. Early Access, pp. 1–15, 2024.
- [26] K. Pyragas, "Continuous control of chaos by self-controlling feedback," *Physics Letters A*, vol. 170, no. 6, pp. 421–428, 1992.
- [27] T. Buchner and J. Zebrowski, "Logistic map with a delayed feedback: Stability of a discrete time-delay control of chaos," *Physical review E*, vol. 63, no. 1, p. 016210, 2000.
- [28] F. S. Gentile, A. L. Bel, M. Belen DAmico, and J. L. Moiola, "Effect of delayed feedback on the dynamics of a scalar map via a frequency-domain approach," *Chaos: An Interdisciplinary Journal of Nonlinear Science*, vol. 21, no. 2, 2011.
- [29] L. Liu and S. Miao, "Delay-introducing method to improve the dynamical degradation of a digital chaotic map," *Information Sciences*, vol. 396, pp. 1–13, 2017.
- [30] W. Liu, K. Sun, H. Wang, and B. Li, "Delayed feedback chaotification model and its hardware implementation," *IEEE Transactions on Industrial Electronics*, vol. 71, no. 10, pp. 13 002–13 011, 2024.
- [31] Z. Yuan, Y. Wu, C. Ou, P. Zhong, X. Zhao, and M. Ma, "Dynamical behavior of SW-SW neural networks," *Chinese Journal of Physics*, vol. 94, pp. 108–120, 2025.
- [32] H. Bao, Z. Hua, H. Li, M. Chen, and B. Bao, "Discrete memristor hyperchaotic maps," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 68, no. 11, pp. 4534–4544, 2021.
- [33] S. He, B. Yan, X. Wu, H. Wang, M. Wang, and H. H.-C. Lu, "Spatiotemporal chaos in a sine map lattice with discrete memristor coupling," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 71, no. 3, pp. 1039–1049, 2024.
- [34] Y. Li, C. Li, T. Lei, Y. Yang, and G. Chen, "Offset boosting-entangled complex dynamics in the memristive Rulkov neuron," *IEEE Transactions on Industrial Electronics*, vol. 71, no. 8, pp. 9569–9579, 2024.
- [35] C. Wang, D. Luo, Q. Deng, and G. Yang, "Dynamics analysis and FPGA implementation of discrete memristive cellular neural network with heterogeneous activation functions," *Chaos, Solitons & Fractals*, vol. 187, p. 115471, 2024.
- [36] Q. Deng, C. Wang, Y. Sun, and G. Yang, "Memristive multi-wing chaotic Hopfield neural network for LiDAR data security," *Nonlinear Dynamics*, vol. Early Access, 2025.
- [37] Y. Yuan, F. Yu, B. Tan, Y. Huang, S. Cai, and H. Lin, "A class of n-D hamiltonian conservative chaotic systems with three-terminal memristor: Modeling, dynamical analysis, and FPGA implementation," *Chaos*, vol. 35, no. 1, p. 013121, 2025.
- [38] C. Wang, Y. Li, and Q. Deng, "Discrete-time fractional-order local active memristor-based Hopfield neural network and its FPGA implementation," *Chaos, Solitons & Fractals*, vol. 193, p. 116053, 2025.
- [39] H. K. Kwan, "Simple sigmoid-like activation function suitable for digital hardware implementation," *Electronics Letters*, vol. 28, pp. 1379–1380, 1992.
- [40] M. A. Hussain, S.-W. Lin, and T.-H. Tsai, "An area-efficient and high throughput hardware implementation of exponent function," in *2022 IEEE International Symposium on Circuits and Systems (ISCAS)*, 2022, pp. 3369–3372.
- [41] Vikas and P. Dayal, R., "Chaos-based optimal path planning of humanoid robot using hybridized regression-gravity search algorithm in static and dynamic terrains," *Applied Soft Computing*, vol. 140, p. 110236, 2023.
- [42] Y. Yang, S. Qin, and S. Liao, "Ultra-chaos of a mobile robot: A higher disorder than normal-chaos," *Chaos, Solitons & Fractals*, vol. 167, p. 113037, 2023.
- [43] F. Ahuraka, P. Mcnamee, Q. Wang, Z. N. Ahmadabadi, and J. Hudack, "Chaotic motion planning for mobile robots: Progress, challenges, and opportunities," *IEEE Access*, vol. 11, pp. 134 917–134 939, 2023.
- [44] J. Kennedy and R. Eberhart, "Particle swarm optimization," in *Proceedings of ICNN'95 - International Conference on Neural Networks*, vol. 4, 1995, pp. 1942–1948.
- [45] S. He, L. Fu, Y. Lu, X. Wu, H. Wang, and K. Sun, "Analog circuit of a simplified tent map and its application in sensor position optimization," *IEEE Transactions on Circuits and Systems II: Express Briefs*, vol. 70, no. 3, pp. 885–888, 2023.
- [46] M. Clerc and J. Kennedy, "The particle swarm - explosion, stability, and convergence in a multidimensional complex space," *IEEE Transactions on Evolutionary Computation*, vol. 6, no. 1, pp. 58–73, 2002.
- [47] D. Zhou, X. Gao, G. Liu, C. Mei, D. Jiang, and Y. Liu, "Randomization in particle swarm optimization for global search ability," *Expert Systems with Applications*, vol. 38, no. 12, pp. 15 256–15 364, 2011.
- [48] H. Essam H., G. Ahmed G., H. Kashif, and S. Ponnuthurai Nagaratnam, "Major advances in particle swarm optimization: Theory, analysis and applications," *Swarm and Evolutionary Computation*, vol. 63, p. 100868, 2021.
- [49] Y. Duan, N. Chen, L. Chang, Y. Ni, S. V. N. S. Kumar, and P. Zhang, "Capso: Chaos adaptive particle swarm optimization algorithm," *IEEE Access*, vol. 10, pp. 29 393–29 405, 2022.



Quanli Deng received the B.S. degree in microelectronics from the School of Physics and Optoelectronics, Xiangtan University, Xiangtan, China, in 2016, the M.S. in information and communication engineering from the College of Computer Science and Electronic Engineering, Hunan University, Changsha, China, in 2020 and the Ph.D. degree in computer science and technology from the College of Computer Science and Electronic Engineering, Hunan University, Changsha, China, in 2024.

He is currently a Postdoctoral Research Fellow with the College of Computer Science and Electronic Engineering, Hunan University. His research interests include modeling and analysis of neural systems, fundamental theory of nonlinear systems and circuits, and analog implementation of neuromorphic systems.



Chunhua Wang received the M.S. degree in microphysics from Zhengzhou University, Zhengzhou, China, in 1994, and the Ph.D. degree in microelectronics and solid-state electronics from the Beijing University of Technology, Beijing, China, in 2003.

He is currently a Professor with the College of Computer Science and Electronic Engineering, Hunan University, Changsha, China, where he is a Doctor Tutor and Director of the Advanced Communication Technology Key Laboratory. He has presided more than eight national and provincial projects and

authored or coauthored more than 200 papers retrieved by SCI, among which 20 papers were highly cited. His research interests include chaotic circuit, memristor circuit, chaotic encryption, neural networks based on memristor, complex networks, and current-mode circuit.

Dr. Wang is the Director of the Chaos and Nonlinear Circuit Professional Committee of Circuit and System Branch of the China Electronic Society.



Yichuang Sun (Senior Member, IEEE) received the B.Sc. and M.Sc. degrees from Dalian Maritime University, Dalian, China, in 1982 and 1985, respectively, and the Ph.D. degree from the University of York, York, U.K., in 1996, all in communications and electronics engineering.

Dr. Sun is currently Professor of Communications and Electronics, Head of Communications and Intelligent Systems Research Group, and Head of Electronic, Communication and Electrical Engineering Division in the School of Engineering and Computer Science of the University of Hertfordshire, UK. He has published over 330 papers and contributed 10 chapters in edited books. He has also published four text and research books: *Continuous-Time Active Filter Design* (CRC Press, USA, 1999), *Design of High Frequency Integrated Analogue Filters* (IEE Press, UK, 2002), *Wireless Communication Circuits and Systems* (IET Press, 2004), and *Test and Diagnosis of Analogue, Mixed-signal and RF Integrated Circuits - the Systems on Chip Approach* (IET Press, 2008). His research interests are in the areas of wireless and mobile communications, RF and analogue circuits, microelectronic devices and systems, and machine learning and deep learning.



Gang Yang received the B.S. degree in communications engineering and the M.S. degree in artificial intelligence from the Jiangxi University of science and technology, Ganzhou, China, in 2020 and 2023, respectively. He is currently working toward the Ph.D. degree with the College of Computer Science and Electronic Engineering, Hunan University, Changsha, China.

His current research interests include chaotic systems and circuits, memristor neural networks, and memristor systems and circuits.