IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 1 December 2022, accepted 19 December 2022, date of publication 22 December 2022,
date of current version 29 December 2022.

Digital Object Identifier 10.1109/ACCESS.2022.3231610

RESEARCH ARTICLE

FPGA Realization of Two Different Fractional-Order Time-Delay Chaotic System With Predefined Synchronization Time

LIXIONG LIN[1], QING WANG[2], AND GUOWEI CAI[3]
[1]School of Ocean Information Engineering, Jimei University, Xiamen, Fujian 361021, China
[2]VOYAH Automobile Technology Company, Wuhan, Hubei 430050, China
[3]Fujian Chuanzheng Communications College, Fuzhou 350007, China

Corresponding author: Lixiong Lin (elelinlixiong@139.com)

This work was supported in part by the Jimei University Startup Research Project under Grant ZQ2022002; in part by the Natural Science Foundation of Fujian Province, China, under Grant 2019J05024; and in part by the Xiamen Key Laboratory of Marine Intelligent, China, under Grant B18208.

ABSTRACT To solve the problem that the fractional-order system is difficult to implement on FPGA, the chaotic behavior of a single fractional-order Chen system is realized on the FPGA platform in this article by using Laplace transform and Bode-domain approximation to transform the fractional-order operator. Then the System Generator model of two different fractional-order systems is studied, and FTS/PTS (fixed/predefined time synchronization) algorithms are designed to realize the FTS/PTS of the FOTD (fractional-order time-delay) chaotic systems on FPGA. By changing the initial conditions and parameters of the FOTD chaotic systems, several groups of experiments are carried out on FPGA. The experimental results show that the FTS/PTS controllers of two different FOTD chaotic systems are feasible on FPGA.

INDEX TERMS FPGA, fixed time synchronization, fractional-order time-delay chaotic system, predefined time synchronization.

I. INTRODUCTION

Fractional-order chaotic system, as the product of the combination of chaos and fractional calculus, can establish mathematical models closer to the actual situation and describe various models more accurately, such as weather phenomena, financial research, wind power generation, neural networks, etc. [1], [2], [3]. Many ways of synchronization are found, such as complete synchronization, anti-phase synchronization, finite time synchronization, FTS, etc. [4], [5], [6], [7], [8]. Due to the strong stability and robustness of FTS/PTS methods, the related research has attracted the attention of many scholars in recent years [6]. FTS makes the system reach stable in a given time under initial conditions unknown. Ni et al. [9] proposed a non-singular terminal sliding mode control method for fractional-order chaotic systems with uncertain disturbances to make the fractional-order Liu system reaching a stable state in a fixed time. To solve the problem that the settling time of fixed time stability is difficult to adjust according to actual needs, a special fixed time stability, namely predefined time stability, was introduced in [10]. Predefined time stability allows the system to reach stable within a settling time, which depends only on predefined parameters [11], [12], [13], [14]. The predefined time synchronization is not only independent of the initial conditions of the system, but also independent of parameters of the system. Anguiano-Gijón et al. [15] designed a PTS controller based on Lyapunov function stability method, and applied it in Rossler system and Lorenz system to make them reach the PTS. Lin et al. [14] proposed a new trajectory tracking predefined time controller combined with sliding mode method for nonholonomic mobile robots. At present, the implementation of different synchronization methods is mainly verified by simulation, and rarely

The associate editor coordinating the review of this manuscript and approving it for publication was Guillermo Botella Juan.

implemented on FPGA. The main difficulty lies in the complexity of discretization of control methods and the complexity of program.

Fractional-order chaotic systems contain complex factors such as the nonlinear part and the fractional derivative. Analog implementations suffer from low sensitivity of analog components, while digital implementations suffer from accuracy problems caused by the small number of bits to perform arithmetic operations. The emergence of FPGA opens the way to solve these problems. FPGA is a programmable electronic device composed of various logic gate circuits, which consists of programmable independent logic modules, I/O modules and hardware circuits, etc.. The resources are combined in different ways to build a variety of hardware logic circuits [16], [17], such as multipliers, registers, etc.. Commonly used FPGA chips are also equipped with other hardware components. Various modules are optimized and embedded into the hardware devices of the chip and development board to facilitate the implementation of specific functions, such as graphics processing, digital signal processing, state machine, etc.. FPGA has higher flexibility and can meet more design requirements and actual scene needs. Shen et al. [18] realized the chaotic behavior of multistable fourth-order autonomous Chua system on FPGA based on embedded engineering application development and floating-point format numerical algorithm. Malik and Mir [19] used the fractional-order operator Laplace transform method to realize the conversion of fractional-order neurons on FPGA, and successfully operated. Abdelaty et al. [20] proposed an FPGA realization of the fractional order operator based on the product integration rules with a modification in the PI rules. Mohamed et al. [21] proposed FPGA realization of an IP core for generic fractional-order derivative based on Grünwald-Letnikov approximation. Monir et al. [22] implemented the fractional order differentiator and integrator of Grünwald Letnikov definition on FPGA for different fractional orders. Clemente-Lopez et al. [23] explored three types of the embedded and non-embedded implementation of a 3D fractional order chaotic system and the elaboration of a chaos based true random number generator. From above analysis, most of the current scholars' research is to realize the chaotic behavior characteristics of nonlinear systems on FPGA [18], [19], [20], [21], [22]. The time delay in fractional order chaotic systems is also not considered. How to solve the problem of uncertain initial values in FTS of FOTD, and how to solve the problem of adjustable settling time in synchronization of FOTD? Therefore, this article will focus on the problem of the implementation of synchronization of FOTD chaotic system on FPGA. This study has the following contributions:

1) Fractional-order nonlinear system is decomposed by Laplace transform and Bode-domain approximation, and the chaotic behavior of fractional-order chaotic system is realized on FPGA.
2) Clemente-Lopez et al. [23], [24] realized fractional order chaotic system through IP Bolcks on a Xilinx Zynq-7000 XC7Z020 SoC. This article studies the system generator model of two different FOTD chaotic systems, and combines MATLAB with FPGA to achieve the combination of simulation results and implementation hardware.
3) Considering time delay in chaotic system, the FTS of two different FOTD chaotic systems are implemented on FPGA. On this basis, considering adjustable parameter of the settling time, the PTS of two different FOTDs is realized on FPGA, which broadens the application of chaotic systems synchronization in practical scenarios.

II. PRELIMINARIES

Consider the system

$$\dot{x} = g(t, x), \quad x(0) = x_0, \tag{1}$$

where $x \in \mathbb{R}^n$ denotes the state vector and $g : \mathbb{R}_+ \times \mathbb{R}^n \to \mathbb{R}^n$ represents a smooth nonlinear function.

Definition 1: [25] If system (1) *is globally finite time stable and the settling time is bounded, i.e.* $\exists T_{max} > 0$: $\forall x_0 \in R^n$, $T(x_0) \leq T_{max}$. *Then system* (1) *is globally fixed time stable.*

Lemma 1: [26] A positive definite and continuous radially function $V(x) : R^n \to R$ *for system* (1) *satisfies* $\alpha, \beta, \gamma, \eta > 0$, $\gamma\eta > 1$ *for any* $V(x) > 0$, *such that*

$$\dot{V}(x) \leq -(\alpha V(x)^\gamma + \beta)^\eta, \quad x(t) \in R^n \backslash \{0\}. \tag{2}$$

Then system (1) *is stable in fixed time and the settling time is* T^1_{max}, *which is described as*

$$T(x_0) \leq T^1_{max} \triangleq \frac{1}{\beta^\eta}\left(\frac{\beta}{\alpha}\right)^{\frac{1}{\gamma}}\left(1 + \frac{1}{\gamma\eta - 1}\right). \tag{3}$$

Definition 2: [12] If system (1) *is globally fixed time stable and the settling time* $T(x_0)$ *is*

$$T(x_0) \leq T_c, \forall x_0 \in \mathbb{R}^n,$$

where T_c *is a adjustable parameter called a predefined time. Then system* (1) *is predefined time stable.*

Lemma 2: [26] A positive definite and continuous radially function $V(x) : R^n \to R$ *for system* (1) *satisfies* $\alpha, \beta, \gamma, \eta > 0$, $\gamma\eta > 1$ *for any* $V(x) > 0$, *such that*

$$\dot{V} \leq -\frac{T^2_{max}}{T_c}(\alpha V^\gamma + \beta)^\eta \tag{4}$$

with

$$T(x_0) \leq T^2_{max} = \frac{\beta^{\frac{1}{\gamma} - \eta}}{\alpha^{\frac{1}{\gamma}}\gamma} B\left(\frac{1}{\gamma}, \eta - \frac{1}{\gamma}\right), \tag{5}$$

where $B(\sigma, \theta)$ *is the complete beta function. Then system* (1) *is stable in predefined time and the predefined time is* T_c.

Common simulation methods in fractional order systems include Caputo method, G-L (Grunwald-Letnikov) method, Adams Bashforth Coulton method, R-L (Riemman-Liouville) method, etc.. In addition, under certain conditions, the dynamic analysis required for fractional order chaotic

systems can be evaluated by system entropy, Lyapunov exponent, Kaplan–York dimension, etc. [27]. It is found that the G-L definition and the Caputo definition are equivalent under certain conditions, and the R-L definition and the Caputo definition can replace each other in the same period of time. Caputo's Derivative Operator can be applied not only to fractional order chaotic systems, such as Lorenz-Stenflo chaotic systems and Colpitts oscillator [28], but also to the analysis of fractional order models in economics [29]. Therefore this article uses Caputo derivative definition.

Definition 3: [30] The definition of Caputo derivative for $h(t)$ is given as following

$$ {}_{t_0}^{C}D_t^{\beta}h(t) = \begin{cases} \dfrac{1}{\Gamma(n-\beta)}\displaystyle\int_{t_0}^{t}\frac{h^{(n)}(s)}{(t-s)^{\beta-n+1}}ds, \\ n-1<\beta<n, \\ \dfrac{d^n h(t)}{dt^n}, \beta = n, \end{cases} $$

where $\beta > 0$ is the order of derivative; t_0 is the initial time and $t \geq t_0$; n is integer and $n-1 < \beta \leq n$.

Lemma 3: Laplace transform of fractional order operators

$$ D^{\alpha}f(t) = \int_0^{\infty} e^{-st}\,{}_0D_t^{\alpha}f(t)dt = s^{\alpha}F(s) $$

where α is the order of the system; $s = \sigma + j\omega$ is the complex variable; $F(s)$ is the corresponding complex function.

III. IMPLEMENTATION OF FRACTIONAL-ORDER CHAOTIC SYSTEM ON FPGA

A. FRACTIONAL-ORDER CHEN CHAOTIC SYSTEM

The fractional-order Chen chaotic system is

$$ \begin{cases} D^{\alpha_1}x_1 = a(x_1 - x_2) \\ D^{\alpha_2}x_2 = -x_1x_3 + cx_2 + (c-a)x_1 \\ D^{\alpha_3}x_3 = -bx_3 + x_1x_2 \end{cases} \tag{6} $$

where a, b, c are the system parameters, with values of $a = 35, b = 3, c = 28$ respectively. Order is $\alpha_1 = \alpha_2 = \alpha_3 = \alpha = 0.9$, and initial value is $X(0) = [x_1(0), x_2(0), x_3(0)] = [2, 1, 3]^T$. To solve the problem of fractional operator transformation, according to the Baud domain approximation method, Ahmad and Sprott [31] obtained the approximate expression of the transfer function in the frequency domain after a lot of calculations: $H(s) = H(0)/s^{\alpha}$. When $\alpha = 0.9$, $1/s^{\alpha}$ can be approximately estimated as:

$$ H(s) = \frac{1}{s^{0.9}} = \frac{2.2675(s+215.4)(s+1.292)}{(s+2.145)(s+359.4)(s+0.01292)}, \tag{7} $$

where the step size is 0.1 and the maximum error is $2dB$. By substituting the parameters into the system (6), and performing Laplace transform on the system (6), one can obtain that

$$ \begin{cases} s^{0.9}x_1(s) = 35x_2(s) - 35x_1(s) \\ s^{0.9}x_2(s) = -u(s) - 7x_1(s) + 28x_2(s) \\ s^{0.9}x_3(s) = w(s) - 3x_3(s) \end{cases} \tag{8} $$

where $u(s) = \zeta[x_1(s), x_3(s)]$, $w(s) = \zeta[x_1(s), x_2(s)]$. In [32], the embedded hardware implementation of a fractional order switching system is realized by approximating fractional order chaotic systems to integer order chaotic systems through time-frequency-time and Forward-Euler method. The implementation of fractional order continuous time chaotic systems on FPGA and embedded systems is reviewed in [24], especially in the calculation and algorithm of solution methods. To reduce the computation, this article adopts the Bode-domain approximation method. The left and right sides of system (8) are simultaneously multiplied by $1/s^{0.9}$, and the system (8) can be fitted to a nine-dimensional first-order differential system by the Bode-domain approximation and formula (7).

$$ \begin{cases} \dot{x}_1 = x_2 \\ \dot{x}_2 = x_3 \\ \dot{x}_3 = -(35gl+p)x_1 - (35gk+n)x_2 \\ -(35g+m)x_3 + 35g(lx_4 + kx_5 + x_6) \\ \dot{x}_4 = x_5 \\ \dot{x}_5 = x_6 \\ \dot{x}_6 = -g(x_3x_7 + x_1x_9 + 2x_2x_8 + lx_1x_7 \\ +k(x_2x_7 + x_1x_8)) + (28gl - p)x_4 \\ +(28gk - n)x_5 + (28g - m)x_6 \\ -7g(lx_1 + kx_2 + x_3) \\ \dot{x}_7 = \dot{x}_8 \\ \dot{x}_8 = \dot{x}_9 \\ \dot{x}_9 = g(x_1x_6 + x_3x_4 + 2x_2x_5 + lx_1x_4 \\ +k(x_1x_5 + x_2x_4)) - (3gl+p)x_7 \\ -(3gk+n)x_8 - (3g+m)x_9 \end{cases} \tag{9} $$

where $g = 2.2675$, $k = 216.692$, $l = 278.2968$, $m = 361.567$, $n = 778.819$ and $p = 10$. The Forward Euler Method is selected as the discrete formula

$$ y_{n+1} = f(y_n, t_n) \cdot dt + y_n, y(t_0) = y_0 \tag{10} $$

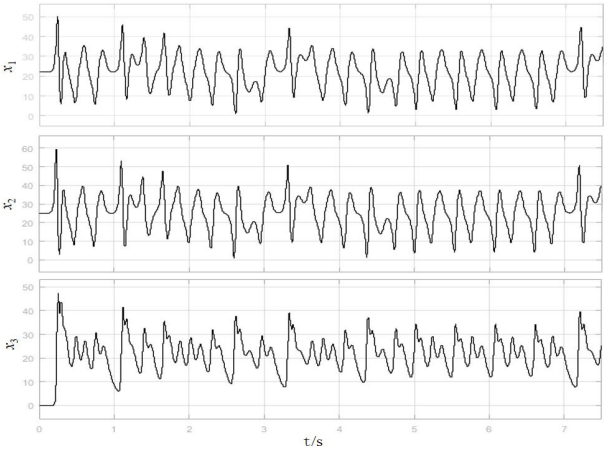

FIGURE 1. State trajectories.

where $f(y_n, t_n) = y'_n$ represents the first derivative of the function, and $dt = t_{n-1} - t_n$ represents the interval step between discrete points. Discretizing the system by the Euler method, one can obtain that

$$\begin{cases} x_1(t+1) = x_2(t) \cdot dt + x_1(t) \\ x_2(t+1) = x_3(t) \cdot dt + x_2(t) \\ x_3(t+1) = (-(35gl + p)x_1(t) - (35gk + n)x_2(t) \\ \quad - (35g + m)x_3(t) + 35g(lx_4(t) + kx_5(t) \\ \quad + x_6(t))) \cdot dt + x_3(t) \\ x_4(t+1) = x_5(t) \cdot dt + x_4(t) \\ x_5(t+1)x_6(t) \cdot dt + x_5(t) \\ x_6(t+1) = ((28gl - p)x_4(t) + (28gk - n)x_5(t) \\ \quad + (28g - m)x_6(t) - 7g(lx_1(t) + kx_2(t) \\ \quad + x_3(t)) - g(x_3(t)x_7(t) + x_1(t)x_9(t) \\ \quad + 2x_2(t)x_8(t) + lx_1(t)x_7(t) + k(x_2(t) \cdot \\ x_7(t) + x_1(t)x_8(t)))) \cdot dt + x_6(t) \\ x_7(t+1) = x_8(t) \cdot dt + x_7(t) \\ x_8(t+1) = x_9(t) \cdot dt + x_8(t) \\ x_9(t+1) = (g(x_1(t)x_6(t) + x_3(t)x_4(t) + 2x_2(t)x_5(t) \\ + lx_1(t)x_4(t) + k(x_1(t)x_5(t) + x_2(t)x_4(t))) \\ \quad - (3gl + p)x_7(t) - (3gk + n)x_8(t) \\ \quad - (3g + m)x_9(t)) \cdot dt + x_9(t) \end{cases} \tag{11}$$

The initial value of system (11) is $X(0) = [0, 0, 2, 0, 0, 1, 0, 0, 3]^T$. After a series of processing, the fractional-order Chen chaotic system has been transformed into a system that can be directly processed by FPGA.

B. SYSTEM GENERATOR MODEL BUILDING

To improve the utilization of FPGA hardware, the coordinate translation of system variables is carried out without changing the dynamic characteristics of the system. According to system (11), the fractional-order Chen chaotic system model is constructed with the help of System Generator. To simplify the program, the floating point arithmetic mode is used for programming. The discrete formulas in (11) are built with nine subsystems and connected according to the relationship between variables. Finally the complete system model is established. Then the state variables of the fractional-order Chen chaotic system are output. The simulation results of the model are shown in FIGURE 1 and FIGURE 2. FIGURE 1 is the trajectories of the fractional-order Chen chaotic system (11). FIGURE 2 is the $x_1 - x_2$ and $x_2 - x_3$ phase diagrams. The basic building blocks of the IP core for fractional-order derivative based on system generator are applied to achieve the reconfigurability of the fractional-order chaotic systems using the steps illustrated in the flowchart in FIGURE 3. The experimental process of realizing fractional-order chaotic system and its synchronization based on FPGA is mainly divided into three steps. Firstly, according to the system formula, the System Generator model is established and designed by using Matlab/Simulink, and

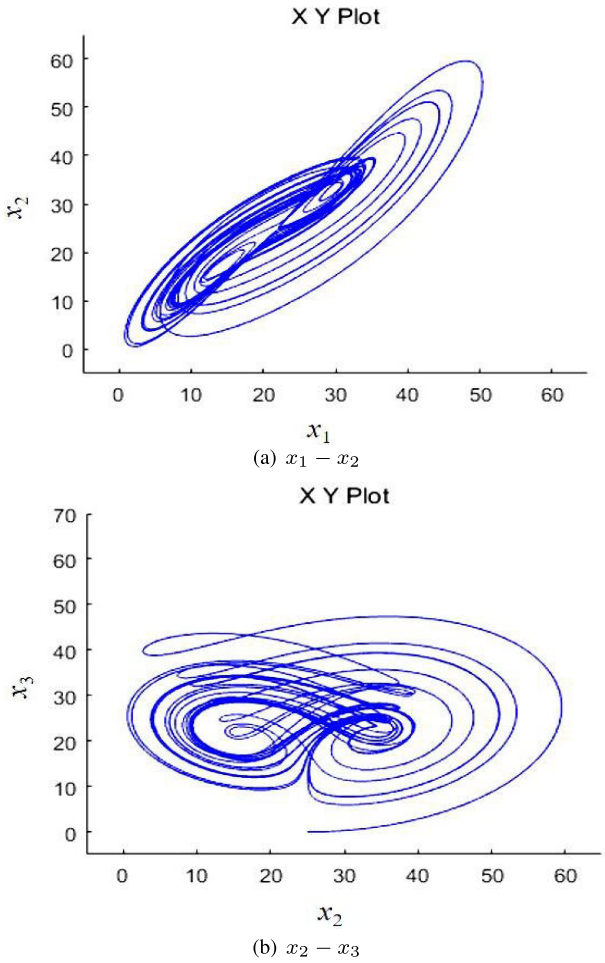

FIGURE 2. State variables phase diagram.

the simulation test is carried out. Secondly, the built model is transformed into a project file, and the integrated design is completed on Vivado, then the bitstream file is generated. Finally, the hardware experiment is completed on the FPGA development board.

C. IMPLEMENTATION OF FRACTIONAL-ORDER CHEN CHAOTIC SYSTEM ON FPGA

In engineering, the wider the bit width of the data captured by the ILA, the greater the sampling depth. The more accurate the captured data trace, the closer it is to the actual value, but the more hardware resources are required. It is particularly important to select appropriate bit width and sampling depth. In order to realize chaotic behavior of the fractional-order Chen chaotic system on FPGA, three probes are set in the project. Set the sampling depth of each probe to 8192 and the sampling bit width to [0, 11]. The results of the implementation of the fractional-order Chen chaotic system on the FPGA are shown in FIGURE 4. The three-dimensional trajectory diagram of the fractional Chen chaotic system (11) is shown in FIGURE 5. According to the relationship between state variables, the output variables still have obvious chaotic

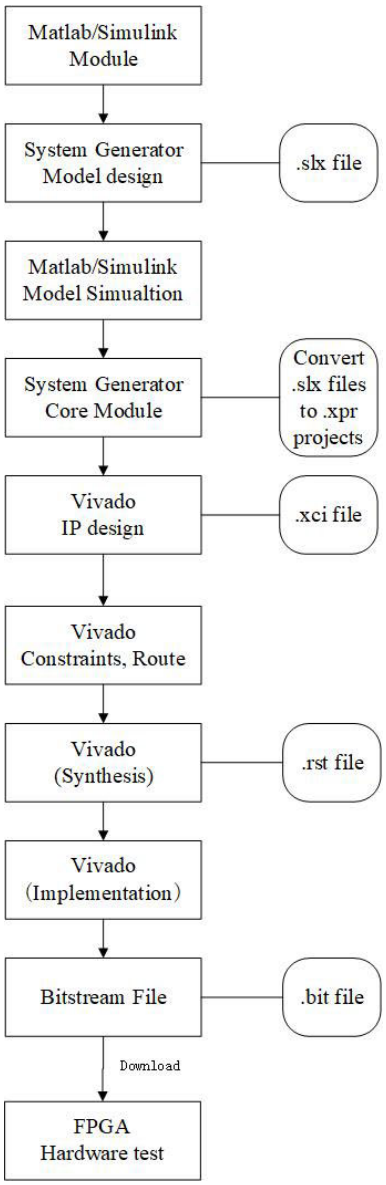

FIGURE 3. Flowchart of the IP core for fractional-order derivative based on system generator with a chaotic system application.

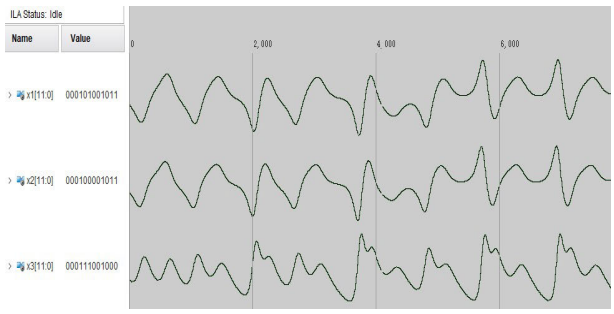

FIGURE 4. Implemented on FPGA.

behavior, which is the same as the chaotic attractor of the original system before transformation. The experimental results in FIGURE 5 show that the proposed method is suitable

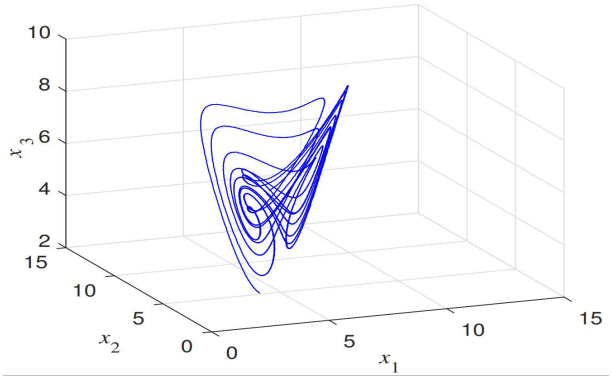

FIGURE 5. Three dimensional trajectory diagram.

for the implementation of fractional-order chaotic system on FPGA.

IV. IMPLEMENTATION OF FTS ON FPGA

The FPGA experimental environment is shown in FIGURE 6, using $XC7Z020-2CLG400I$ chip. The drive FOTD chaotic system is described as [26]

$$\begin{cases} D^{\alpha}x_1 = -\sigma_1 x_1(t) + 0.1\tanh(x_1(t-\tau)) \\ +a(x_2(t)-x_1(t)) \\ D^{\alpha}x_2 = -\sigma_2 x_2(t) + 0.1\tanh(x_2(t-\tau)) \\ +bx_1(t) - dx_1(t)x_3(t) + x_4(t) \\ D^{\alpha}x_3 = -\sigma_3 x_3(t) + 0.1\tanh(x_3(t-\tau)) \\ +h(x_1(t))^2 - cx_3(t) + x_4(t) \\ D^{\alpha}x_4 = 0.1\tanh(x_4(t-\tau)) - \sigma_4 x_4(t) - rx_2(t) \end{cases} \tag{12}$$

where $0 < \alpha < 1$ is the order of drive system; $i = 1, 2, 3, 4$, σ_i represents the self-inhibition of drive system and $\sigma_i > 0$; $x_i(t)$ represents the state variates of drive system; $\tau > 0$ represents the time-delay term. The response system is described as

$$\begin{cases} D^{\alpha}y_1 = -\sigma_1 y_1(t) + a_1(y_2(t) - y_1(t)) \\ + y_4(t) + 0.1\tanh(y_1(t-\tau)) + u_1(t) \\ D^{\alpha}y_2 = -\sigma_2 y_2(t) + b_1 y_1(t) - y_2(t) - y_1(t)\cdot \\ y_3(t) + 0.1\tanh(y_2(t-\tau)) + u_2(t) \\ D^{\alpha}y_3 = -\sigma_3 y_3(t) + y_1(t)y_2(t) - c_1 y_3(t) \\ + 0.1\tanh(y_3(t-\tau)) + u_3(t) \\ D^{\alpha}y_4 = -\sigma_4 y_4(t) - y_2(t)y_3(t) - r_1 y_4(t) \\ + 0.1\tanh(y_4(t-\tau)) + u_4(t) \end{cases} \tag{13}$$

where $y_i(t)$ represents the state variates of response system. The FTS controller for the drive FOTD system (12) and the response FOTD system (13) is designed as:

$$\begin{aligned} u_i(t) = {} & \sigma_i e_i(t) - h_i(e_i(t)) - sign(e_i(t))L \\ & -D^{\alpha-1}\left(\frac{2^{k-1}}{N^{1-qk}}\cdot\alpha_1 sign(e_i(t))\cdot|e_i(t)|^{qk}\right. \\ & \left.+2^{k-1}\lambda_1 sign(e_i(t))\right) \end{aligned} \tag{14}$$

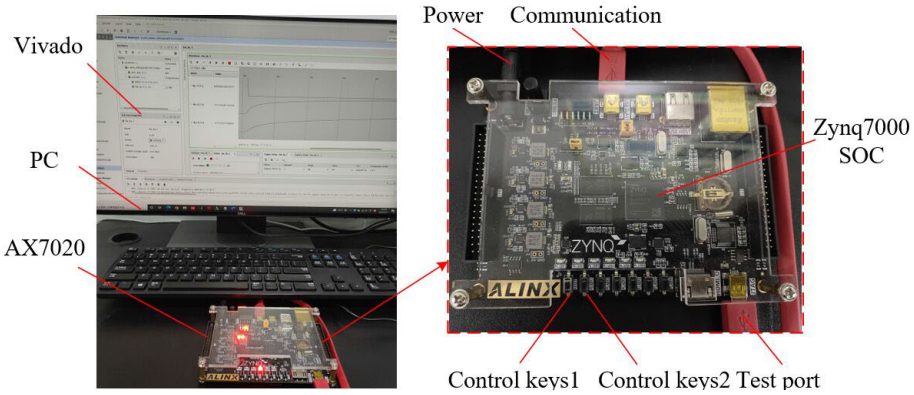

FIGURE 6. FPGA experimental equipment.

where α_1, λ_1, q, k are positive constants and $qk > 1$, L satisfies the Assumption 1 in [26]. The response FOTD chaotic system (13) after adding the controller (14) is:

$$\begin{cases} D^{\alpha}y_1 = a(x_2(t) - x_1(t)) - D^{\alpha-1}(\frac{2^{k-1}}{N^{1-qk}}\alpha_1 \cdot \\ sign(e_1(t))|e_1(t)|^{qk} + 2^{k-1}\lambda_1 sign(e_1(t))) \\ - sign(e_1(t))L + 0.1\tanh(y_1(t-\tau)) \\ D^{\alpha}y_2 = bx_1(t) - dx_1(t)x_3(t) + x_4(t) - sign(e_2(t))L \\ - D^{\alpha-1}(\frac{2^{k-1}}{N^{1-qk}}\alpha_1 sign(e_2(t))|e_2(t)|^{qk} \\ + 2^{k-1}\lambda_1 sign(e_2(t))) + 0.1\tanh(y_2(t-\tau)) \\ D^{\alpha}y_3 = h(x_1(t))^2 - cx_3(t) + x_4(t) - D^{\alpha-1}(\frac{2^{k-1}}{N^{1-qk}} \cdot \\ \alpha_1 sign(e_3(t))|e_3(t)|^{qk} + 2^{k-1}\lambda_1 sign(e_3(t))) \\ - sign(e_3(t))L + 0.1\tanh(y_3(t-\tau)) \\ D^{\alpha}y_4 = -rx_2(t) - D^{\alpha-1}(\frac{2^{k-1}}{N^{1-qk}}\alpha_1 sign(e_4(t)) \cdot \\ |e_4(t)|^{qk} + 2^{k-1}\lambda_1 sign(e_4(t))) \\ -sign(e_4(t))L + 0.1\tanh(y_4(t-\tau)) \end{cases} \tag{15}$$

The parameters of the controller are chosen as $\alpha = 0.9$, $q = 0.9$, $k = 2.9$, $N = 4$, $L = 0.2$, $\sigma_i = 0.1(i = 1, 2, 3, 4)$, $\lambda_1 = 0.003$, $\alpha_1 = 0.007$. Using the same method mentioned above, formula (15) can be converted into 9 formulas and implemented with FPGA. Due to the fast operation speed of FPGA and the short time required for system synchronization, it is necessary to add a trigger signal to control the initial condition of system operation. Setting the *trigger* as external input, FPGA starts to enter the operation state when *trigger* $= 0$. The initial condition of drive FOTD system (12) is $X(0) = [1.67, 3, 0.5, -0.54]^T$. The initial condition of response system (14) with FTS controller (14) is $Y(0) = [5.51, -2, -2.5, 1.46]^T$ and dt=0.0005. The FTS errors of FOTD chaotic systems are shown in FIGURE 7 and FIGURE 8. The trajectory of $e_1(t)$ is shown in FIGURE 7. The specific value of $e_1(t)$ is shown in the form of analog quantity and the specific change of each bit after expansion is shown in the form of digital quantity. The trajectories of e_1, e_2 and e_3, from initial values to stabilization, are shown in FFIGURE 8, where the errors exist in the form of analog quantity and *trigger*[1 : 0] is the trigger signal. From FIGURE 7 and FIGURE 8, in the process of system errors tending to zero, the convergence speed is faster at the initial stage and slows down at the later stage, then converges to near zero within a certain period of time.

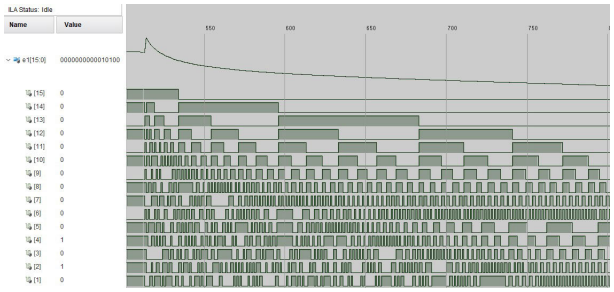

FIGURE 7. Data of e_1.

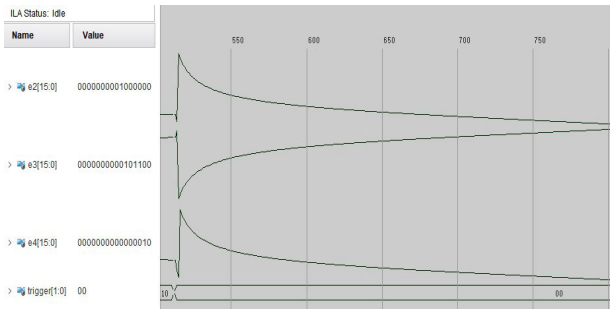

FIGURE 8. Error variation trajectories.

By varying the initial conditions of the FOTD drive-response systems to $X(0) = [-0.4, 3, 5.3, -1]^T$, $X(0) = [-0.7, -0.8, -3.5, -1]^T$, $Y(0) = [1.8, 3, -0.5, -2]^T$ and

$Y(0) = [0.5, 5, 7, -5]^T$, multiple groups of experiments are set up to obtain the FTS error trajectories under different conditions. The results are shown in FIGURE 9. Under the condition of three different initial values, the synchronization error converges to zero with the increasing of time. Due to the accuracy problem between the binary conversion, it can be considered that the drive-response FOTD systems have achieving synchronization in fixed time. According to Definition 1, the drive-response FOTD chaotic systems have reached fixed time stable. According to the estimation formula (2) of the upper bound of the stabilization time in Lemma 1, the upper bound of the settling time is estimated to be $T(x_0) = 0.231$, which is obviously larger than the convergence time $T = 0.18$. In other words, the FTS of the drive response FOTD systems on FPGA has been achieved.

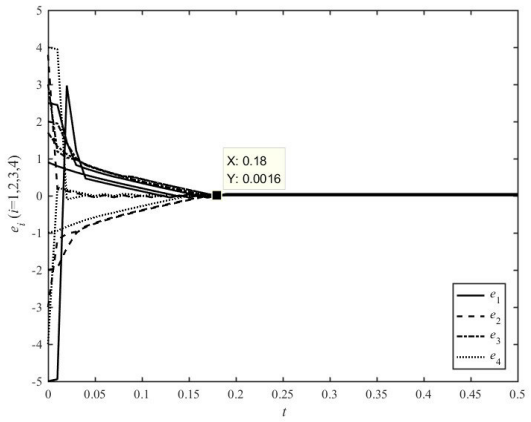

FIGURE 9. Synchronization error trajectories.

V. IMPLEMENTATION OF PTS ON FPGA

A. HARDWARE RESOURCE ANALYSIS

According to the drive FOTD system (12), the response FOTD system (13) and Lemma 2, the PTS controller is designed as:

$$u_i(t) = -h_i(e_i(t)) + \sigma_i e_i(t) - sign(e_i(t))L - D^{\alpha-1}\left(\frac{C_v}{T_c}\cdot (\alpha_2 sign(e_i(t))|e_i(t)|^{qk} + \lambda_2 sign(e_i(t)))\right) \quad (16)$$

where α_2, λ_2, q , k are positive constants and $qk > 1$. The response FOTD system with the PTS controller (16) is as follows:

$$\begin{cases} D^{\alpha} y_1 = a(x_2(t) - x_1(t)) - D^{\alpha-1}\left(\frac{C_v}{T_c}\cdot(\mu sign(e_1(t))\cdot |e_1(t)|^{qk} + \omega sign(e_1(t)))\right) - sign(e_1(t))L \\ \quad + 0.1\tanh(y_1(t-\tau)) \\ D^{\alpha} y_2 = b x_1(t) - k x_1(t) x_3(t) + x_4(t) - D^{\alpha-1}\left(\frac{C_v}{T_c}\cdot (\mu sign(e_2(t))|e_2(t)|^{qk} + \omega sign(e_2(t)))\right) \\ \quad - sign(e_2(t))L + 0.1\tanh(y_2(t-\tau)) \\ D^{\alpha} y_3 = h(x_1(t))^2 - c x_3(t) + x_4(t) - D^{\alpha-1}\left(\frac{C_v}{T_c}\cdot (\mu sign(e_3(t))|e_3(t)|^{qk} + \omega sign(e_3(t)))\right) \\ \quad - sign(e_3(t))L + 0.1\tanh(y_3(t-\tau)) \\ D^{\alpha} y_4 = -r x_2(t) - D^{\alpha-1}\left(\frac{C_v}{T_c}(\mu sign(e_4(t))\cdot |e_4(t)|^{qk} + \omega sign(e_4(t)))\right) - sign(e_4(t))L \\ \quad + 0.1\tanh(y_4(t-\tau)) \end{cases} \quad (17)$$

where the controller parameters are chosen as $q = 0.5$, $k = 5.2$, $\beta_2 = 12.9$, $\lambda_2 = 11$, $L = 0.2$, $\sigma_i = 0.1$ ($i = 1, 2, 3, 4$). In this case, according to [26], $C_v = 0.0534$ is obtained. Compared with the FTS controller (14), the PTS controller (16) adds a tuning parameter, which makes the synchronization process faster and easier to adjust. At this time, there is also more demand for hardware consumption. Its consumption needs to be analyzed to ensure the effective implementation of PTS of two different FOTD chaotic systems on FPGA. After comprehensive analysis, the total on-chip power is $0.3W$ and the thermal margin is $4.7W$ from the netlist analysis. At this point, the allocation of FPGA hardware resources and power consumption are shown in TABLE 1, FIGURE 10 and FIGURE 11. The location allocation of hardware resources required in the development board is shown in FIGURE 10. At the same time, the occupied power and proportion of each part obtained from the constraint, simulation and vector free analysis, are shown in FIGURE 11. From FIGURE 10 and FIGURE 11, the hardware resources in the FPGA development board are divided into six regions according to the location. The resources occupied include most regions of X1Y0, X1Y1, and X0Y0 and a small part of X0Y1. The hardware resources occupied are about half of the total resources. In the total on-chip power, the static power of the device is 0.109W, accounting for 36% of the total power; dynamic power is 0.191W, accounting for 64% of the total power. The power occupied by dynamic equipment includes 0.005W for clocks, 0.059W for signals, 0.055W for logic operations, 0.001W for BRAM, 0.019W for digital signal processing (DSP), and 0.052W for input/output ports (I/O). On the whole, the resources

TABLE 1. Hardware resource allocation.

Resource	Amount occupied	Available Quantity	Utilization
LUT	12856	53200	24.17
LUTRAM	154	17400	0.89
FF	3183	106400	2.99
BRAM	4.5	140	3.21
DSP	78	220	35.45
IO	73	125	28.40
BUFG	2	32	6.25

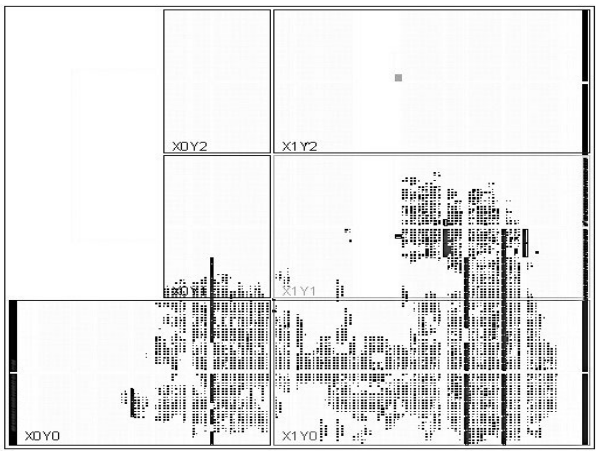

FIGURE 10. Hardware resource allocation.

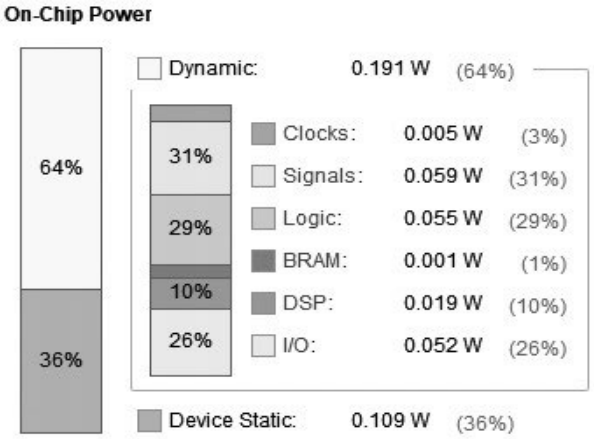

FIGURE 11. Power consumption and distribution.

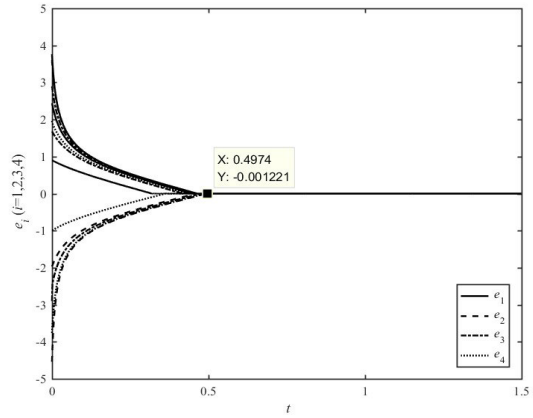

FIGURE 12. Predefined time synchronization error trajectories with $T_c = 1$.

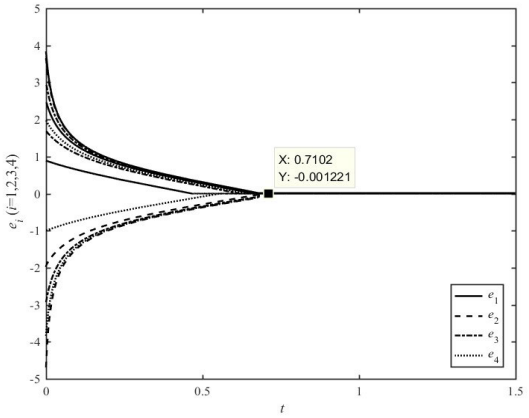

FIGURE 13. Predefined time synchronization error trajectories with $T_c = 1.5$.

required for the predefined time synchronization of two different FOTD chaotic systems are more, but the FPGA device of XC7Z020-2CLG400I chip has more hardware resources, which is enough to realize the experiments related to the predefined time synchronization of fractional order chaotic systems.

B. PREDEFINED TIME SYNCHRONIZATION

When the initial values of the drive FOTD chaotic system are $X(0) = [1.67, 3, 0.5, -0.54]^T$, $X(0) = [-0.7, -0.8, -3.5, -1]^T$ and $X(0) = [-0.4, 3, 5.3, -1]^T$, and the initial values of the response FOTD chaotic system are $Y(0) = [5.51, -2, -2.5, 1.46]^T$, $Y(0) = [1.8, 3, -0.5, -2]^T$ and $Y(0) = [0.5, 5, 7, -5]^T$, keeping other parameters unchanged and changing the value of the tuning parameters T_c, we observe the synchronization process of the drive-response FOTD systems and whether T_c can control the upper bound of the system synchronization time. The simulation results are shown in FIGURE 12 and FIGURE 13. FIGURE 12 is the synchronization error trajectories with $T_c = 1.0$. FIGURE 13 is the synchronization error trajectories with $T_c = 1.5$. When the predefined time $T_c = 1$, the convergence time of the system $T = 0.4974$ is less than T_c. When the predefined time $T_c = 1.5$, the convergence time of the system $T = 0.7102 < 1.5$. That is, under the conditions of different initial values of the system and different tuning parameters, FOTD chaotic systems (12) and (13) reach a stable within predefined time. Combined with the experimental results, the PTS process of FOTD chaotic system can be realized on FPGA by the above method.

VI. CONCLUSION

Based on the FPGA hardware experimental platform, the FTS/PTS of two different FOTD chaotic systems have been studied in this article. The fractional-order Chen chaotic system is transformed by the Bode-domain approximation method and the forward Euler method, and its chaotic behavior has been realized on FPGA. The system generator models of the drive system is constructed. The proposed synchronization method for two different FOTD chaotic systems is verified by experiments with multiple sets of initial values. The experimental results show that the driving-response systems errors under the FTS/PTS controllers converges

rapidly in the initial stage and then slows down slightly in the later stage, but it can still converge to zero within the predefined time. In future research, we will make efforts in the following aspects

1) More attention will be paid to the computational and algorithmic aspects of the solution method of the fractional differential equation.
2) The above solutions are applied to the synchronization of two different fractional order systems, and the power consumption, clock freq, throughput, Bit rate, and hardware resource utilization of different synchronization methods will be studied.
3) We will make full use of the advantages of fuzzy control in solving model uncertainty and consider the combination of predefined time stability and fractional-order multiple-Model Type-3 Fuzzy control [33]. It is hoped that it can be applied to real objects, such as autonomous vehicles [34].

REFERENCES

[1] A. Tian, C. Fu, H.-T. Yau, X.-Y. Su, and H. Xiong, "A new methodology of soil salinization degree classification by probability neural network model based on centroid of fractional Lorenz chaos self-synchronization error dynamics," *IEEE Trans. Geosci. Remote Sens.*, vol. 58, no. 2, pp. 799–810, Feb. 2020.

[2] K. A. Alattas, J. Mostafaee, A. Sambas, A. K. Alanazi, S. Mobayen, M. T. Vu, and A. Zhilenkov, "Nonsingular integral-type dynamic finite-time synchronization for hyper-chaotic systems," *Mathematics*, vol. 10, no. 1, p. 115, Dec. 2021.

[3] J. Li, L. Chen, W. Cai, J. Xiao, J. Zhu, Y. Hu, and K. Wen, "Holographic encryption algorithm based on bit-plane decomposition and hyper-chaotic Lorenz system," *Opt. Laser Technol.*, vol. 152, Aug. 2022, Art. no. 108127.

[4] L. Wang, T. Dong, and M.-F. Ge, "Finite-time synchronization of memristor chaotic systems and its application in image encryption," *Appl. Math. Comput.*, vol. 347, pp. 293–305, Apr. 2019.

[5] W. S. Sayed and A. G. Radwan, "Generalized switched synchronization and dependent image encryption using dynamically rotating fractional-order chaotic systems," *AEU Int. J. Electron. Commun.*, vol. 123, Aug. 2020, Art. no. 153268.

[6] M. Shirkavand, M. Pourgholi, and A. Yazdizadeh, "Robust global fixed-time synchronization of different dimensions fractional-order chaotic systems," *Chaos, Solitons Fractals*, vol. 154, Jan. 2022, Art. no. 111616.

[7] B. Wang, W. Chen, B. Zhang, and Y. Zhao, "Regulation cooperative control for heterogeneous uncertain chaotic systems with time delay: A synchronization errors estimation framework," *Automatica*, vol. 108, Oct. 2019, Art. no. 108486.

[8] Z. Li, T. Xia, and C. Jiang, "Synchronization of fractional-order complex chaotic systems based on observers," *Entropy*, vol. 21, no. 5, p. 481, May 2019.

[9] J. Ni, L. Liu, C. Liu, and X. Hu, "Fractional order fixed-time nonsingular terminal sliding mode synchronization and control of fractional order chaotic systems," *Nonlinear Dyn.*, vol. 89, no. 3, pp. 2065–2083, Aug. 2017.

[10] J. D. Sanchez-Torres, E. N. Sanchez, and A. G. Loukianov, "A discontinuous recurrent neural network with predefined time convergence for solution of linear programming," in *Proc. IEEE Symp. Swarm Intell.*, Dec. 2014, pp. 1–5.

[11] J. D. Sánchez-Torres, E. N. Sanchez, and A. G. Loukianov, "A discontinuous recurrent neural network with predefined time convergence for solution of linear programming," in *Proc. IEEE Symp. Swarm Intell.*, Dec. 2014, pp. 226–230.

[12] J. D. Sánchez-Torres, D. Gómez-Gutiérrez, E. López, and A. G. Loukianov, "A class of predefined-time stable dynamical systems," *IMA J. Math. Control Inf.*, vol. 35, pp. i1–i29, Mar. 2018.

[13] L. Lin, "Predefined-time synchronization of 5D Hindmarsh–Rose neuron networks via backstepping design and application in secure communication," *Nonlinear Anal., Modeling Control*, vol. 27, no. 4, pp. 630–649, Apr. 2022.

[14] L. Lin, P. Wu, B. He, Y. Chen, J. Zheng, and X. Peng, "The sliding mode control approach design for nonholonomic mobile robots based on non-negative piecewise predefined-time control law," *IET Control Theory Appl.*, vol. 15, no. 9, pp. 1286–1296, Jun. 2021.

[15] C. A. Anguiano-Gijón, A. J. Muñoz-Vázquez, J. D. Sánchez-Torres, G. Romero-Galván, and F. Martínez-Reyes, "On predefined-time synchronisation of chaotic systems," *Chaos, Solitons Fractals*, vol. 122, pp. 172–178, May 2019.

[16] J. Zhang, Q. Yin, W. Hu, Y. Li, H. Li, N. Ye, and B. Cao, "EPA : The effective pipeline architecture for CNN accelerator with high performance and computing efficiency based on FPGA," *Concurrency Comput., Pract. Exper.*, vol. 2021, p. e6198, Mar. 2021.

[17] S. He, D. Zhan, H. Wang, K. Sun, and Y. Peng, "Discrete memristor and discrete memristive systems," *Entropy*, vol. 24, no. 6, p. 786, Jun. 2022.

[18] H. Shen, B. He, Y. Song, B. Yin, S. Cai, L. Liu, Z. Zhang, F. Yu, Y. Huang, and S. Du, "CCII and FPGA realization: A multistable modified fourth-order autonomous Chua's chaotic system with coexisting multiple attractors," *Complex.*, vol. 2020, Mar. 2020, Art. no. 5212601.

[19] S. A. Malik and A. H. Mir, "FPGA realization of fractional order neuron," *Appl. Math. Model.*, vol. 81, pp. 372–385, May 2020.

[20] A. M. Abdelaty, M. Roshdy, L. A. Said, and A. G. Radwan, "Numerical simulations and FPGA implementations of fractional-order systems based on product integration rules," *IEEE Access*, vol. 8, pp. 102093–102105, 2020.

[21] S. M. Mohamed, W. S. Sayed, L. A. Said, and A. G. Radwan, "Reconfigurable FPGA realization of fractional-order chaotic systems," *IEEE Access*, vol. 9, pp. 89376–89389, 2021.

[22] M. S. Monir, W. S. Sayed, A. H. Madian, A. G. Radwan, and L. A. Said, "A unified FPGA realization for fractional-order integrator and differentiator," *Electronics*, vol. 11, no. 13, p. 2052, Jun. 2022.

[23] D. Clemente-Lopez, J. J. Rangel-Magdaleno, J. M. Munoz-Pacheco, and L. Morales-Velazquez, "A comparison of embedded and non-embedded FPGA implementations for fractional chaos-based random number generators," *J. Ambient Intell. Hum. Comput.*, vol. 2022, pp. 1–15, Sep. 2022.

[24] D. Clemente-López, J. M. Munoz-Pacheco, and J. D. J. Rangel-Magdaleno, "A review of the digital implementation of continuous-time fractional-order chaotic systems using FPGAs and embedded hardware," *Arch. Comput. Methods Eng.*, vol. 2022, pp. 1–33, Oct. 2022.

[25] S. P. Bhat and D. S. Bernstein, "Finite-time stability of continuous autonomous systems," *SIAM J. Control Optim.*, vol. 38, no. 3, pp. 751–766, Jan. 2000.

[26] L. Lin, Q. Wang, B. He, Y. Chen, X. Peng, and R. Mei, "Adaptive predefined-time synchronization of two different fractional-order chaotic systems with time-delay," *IEEE Access*, vol. 9, pp. 31908–31920, 2021.

[27] Z. Youan, Y. Jian, and L. Jingmao, *Control and Synchronization of Fractional-Order Chaotic Systems*. China Industry and Information Technology Publishing & Media Group, 2018.

[28] H. Wang, D. Zhan, X. Wu, and S. He, "Dynamics of a fractional-order colpitts oscillator and its FPGA implementation," *Eur. Phys. J. Special Topics*, vol. 231, nos. 11–12, pp. 2467–2476, Aug. 2022.

[29] H. U. Rehman, M. Darus, and J. Salah, "A note on Caputo's derivative operator interpretation in economy," *J. Appl. Math.*, vol. 2018, pp. 1–7, Oct. 2018.

[30] I. Petráš, "Chaos in the fractional-order Volta's system: Modeling and simulation," *Nonlinear Dyn.*, vol. 57, nos. 1–2, pp. 157–170, 2009.

[31] W. M. Ahmad and J. C. Sprott, "Chaos in fractional-order autonomous nonlinear systems," *Chaos, Solitons Fractals*, vol. 16, no. 2, pp. 339–351, Mar. 2003.

[32] E. Zambrano-Serrano, J. M. Muñoz-Pacheco, and E. Campos-Cantón, "Chaos generation in fractional-order switched systems and its digital implementation," *AEU Int. J. Electron. Commun.*, vol. 79, pp. 43–52, Sep. 2017.

[33] A. Mohammadzadeh, O. Castillo, S. S. Band, and A. Mosavi, "A novel fractional-order multiple-model Type-3 fuzzy control for nonlinear systems with unmodeled dynamics," *Int. J. Fuzzy Syst.*, vol. 23, no. 6, pp. 1633–1651, Sep. 2021.

[34] A. Mohammadzadeh and H. Taghavifar, "A robust fuzzy control approach for path-following control of autonomous vehicles," *Soft Comput.*, vol. 24, no. 5, pp. 3223–3235, Mar. 2020.

LIXIONG LIN received the B.S. degree in electrical engineering and automation from Huaqiao University, China, in 2008, and the M.S. degree in detection technology and the Ph.D. degree in control theory and control engineering from Xiamen University, China, in 2011 and 2016, respectively. He was a joint Ph.D. Student at the UAV Research Group, Department of Electrical and Computer Engineering, Temasek Laboratories, National University of Singapore, from 2015 to 2016. He was a Software Engineer at Kehua Company Ltd., Xiamen, China, from 2011 to 2013. He was also an Assistant Professor at the School of Mechanical Engineering and Automation, Fuzhou University, from 2017 to 2021. He is currently an Associate Professor at the School of Ocean Information Engineering, Jimei University. His research interests include fault detection, multi-sensor fusion, robotic control systems, and autonomous navigation.

QING WANG received the B.S. degree from the Hubei University of Arts and Sciences, China, in 2019, and the M.S. degree in mechanical engineering and automation from Fuzhou University, China, in 2022. She is currently a Software Engineer at VOYAH Automobile Technology Company, Hubei, China. Her research interests include system synchronization and fault detection.

GUOWEI CAI received the B.Sc. degree from Tianjin University, in 2002, and the Ph.D. degree from the National University of Singapore, in 2009. He is currently a Professor at the Fujian Chuanzheng Communications College. His main research interests include integrated navigation and UAV control.