Research Paper
Computational Methods for Differential Equations
http://cmde.tabrizu.ac.ir
Vol. 11, No. 2, 2023, pp. 207-224
DOI:10.22034/cmde.2022.51132.2127

Optimal fractional order PID controller performance in chaotic system of HIV disease: particle swarm and genetic algorithms optimization method

Shaban Mohammadi* **and Seyed Reza Hejazi**
Faculty of mathematical sciences, Shahrood university of technology, Shahrood, Semnan, Iran.

Abstract

The present study aims to investigate the optimal fractional order PID controller performance in the chaotic system of HIV disease fractional order using the Particle Swarm optimization and Genetic algorithm method. Differential equations were used to represent the chaotic behavior associated with HIV. The optimal fractional order of the PID controller was constructed, and its performance in the chaotic system with HIV fractional order was tested. Optimization methods were used to get PID control coefficients from particle swarm and genetic algorithms. Findings revealed that the equations for the HIV disease model are such that the system's behavior is greatly influenced by the number of viruses produced by infected cells, such that if the number of viruses generated by infected cells exceeds 202, the disease's behavior is such that the virus and disease spread. For varying concentrations of viruses, the controller created for this disease does not transmit the disease.

Keywords. Chaotic system, Optimal fractional order controller, Genetic algorithm, particle swarm optimization algorithm, HIV.
2010 Mathematics Subject Classification. 49J20, 68W50.

1. Introduction

In the second half of 1994, it has been estimated that 17 million people worldwide were living with HIV. Today, HIV is a huge social burden as well as an infectious and immunological illness. CD4 + T cells, which are vital for the immune system, are affected by the virus. HIV is the most prevalent pathogenic illness of the twentieth century, affecting or killing millions of people worldwide in less than 30 years. Unfortunately, there has been no definitive cure for this disease yet. Only antiviral medication therapy can prevent HIV from progressing to AIDS, allowing a person to enjoy a relatively healthy life for several years [42].

Antiviral medication loses efficacy after a few years in some people, and in others, they live with HIV for decades, but they are obliged to take powerful drugs for the rest of their lives, which can have terrible side effects. Long-term usage of the medicine, on the other hand, creates severe negative effects on the patient's body. To overcome this challenge, mathematical modeling and control systems approaches may be used to combine existing information and offer a theoretical foundation for current models, which can be a useful tool for determining the best control strategy. Various strategies for optimum disease control have been offered based on different disease models and cost functions utilized in the articles, which eventually decrease the drug's adverse effects, keep the virus population at a low level, and extend life expectancy. These approaches are also employed in other medical engineering disciplines, such as cancer therapy. The use of fuzzy logic or neural networks has been used to control HIV in simulations, but the primary drawbacks of these approaches are first, the problem design of these controllers in the actual world, and second, the poor speed of system operation owing to the huge computational volume [12, 43].

Optimal LQR control has been employed in various additional approaches. The primary disadvantages of this strategy are, first, the system's linearity and, second, the need to adjust all system parameters and relationships completely [3]. Another flaw in the approaches discussed above is the system relations as an integer order. Given

Received: 14 April 2022 ; Accepted: 02 October 2022.
∗ Corresponding author. Email: arashmoh2019@gmail.com.

that all systems in the actual world are in fractional order, studying the system in the integer order will not aid in the accuracy of the conclusions. The concepts of derivation and integration have been beneficial in advancing natural and artificial systems, science, and technology. Fractional calculus is a branch of mathematics dedicated to furthering these ideas. Because many natural events have fractional orders and may be represented as fractional order differential equations, fractional order conversion functions, or fractional order state-space, fractional computations have been employed for a variety of applications [14, 20]. Fractional order systems have more degrees of freedom than integer order systems to be more versatile based on order, hence there is an unlimited number of matching integer order systems, just as there is an endless number of fractional order systems. In a variety of applications, such as electronic power converter control, systems with uncertain parameters, heat dissipation, and viscoelastic processes, robust modeling and control are studied [17].

Real-world systems are better described by fractional order differential equations. Furthermore, the use of fractional order differential equations opened up new possibilities in control engineering. In both control systems and medicine, HIV control has long been a major concern. Accordingly, fractional calculus is extensively utilized and developed; however, numerous experts have expressly declared that systems in the actual world are in fractional order. For this purpose, the HIV fractional order model was employed in the present research to make the simulation findings more realistic.

1.1. **Chaotic fractional-order systems.** This article investigates the parameters and conditions for which the fractional-order system could have chaotic behavior [13]. In this section, two relevant theorems for fractional-order systems are stated [23, 34, 41]. The theorem is about proportional fractional-order systems [1, 5, 8, 9, 23, 25, 30, 32, 34, 37].

Theorem 1.1. *In an autonomous system we have:*

$$\frac{d^\alpha x}{dt^\alpha} = Ax, \qquad x(0) = x_0, \tag{1.1}$$

i) *By considering* $0 < \alpha < 1$ *and* $x \in \mathbb{R}^{n\times n}$, *matrix* $A \in \mathbb{R}^{n\times n}$ *is asymptotically stable if and only if* $|\arg(\lambda)| > \frac{\alpha\pi}{2}$ *is valid. In this equation* λ *is the eigenvalue of matrix* A. *In addition, this matrix is stable if and only if* $|\arg(\lambda)| > \frac{\alpha\pi}{2}$.

ii) *The equilibrium point in fractional-order systems is calculated as in ordinary differential equations as below:*

$$\frac{d^\alpha x}{dt^\alpha} = f(x), \qquad f(x) = 0. \tag{1.2}$$

In the equation above, we have $0 < \alpha < 1$ *and* $x \in \mathbb{R}^{n\times n}$. *The equilibrium point is achieved by solving the equation is asymptotically stable if the calculated eigenvalue* λ *related to the Jacobian matrix* $J = \frac{df}{dt}$ *satisfies the following equation in equilibrium point:*

$$|\arg(\lambda)| > \frac{\alpha\pi}{2}. \tag{1.3}$$

Theorem 1.2. *The n-dimensional dynamic fractional-order system could be specified as follows:*

$$\begin{cases} \frac{d^{\alpha_1} x_1}{dt^{\alpha_1}} = a_{11}x_1 + a_{12}x_2 + \cdots + a_{1n}x_n, \\ \frac{d^{\alpha_2} x_2}{dt^{\alpha_2}} = a_{21}x_1 + a_{22}x_2 + \cdots + a_{2n}x_n, \\ \vdots \\ \frac{d^{\alpha_n} x_n}{dt^{\alpha_n}} = a_{n1}x_1 + a_{n2}x_2 + \cdots + a_{nn}x_n. \end{cases} \tag{1.4}$$

In the equation above, all α_i *coefficients have values between 0 and 1. It is assumed that* M *is the least common multiple of* u_i *that is expressed as* $\alpha_i = \frac{v_i}{u_i}$. *Here* $(u_i, v_i) = 1$ *and* $u_i, v_i \in \mathbb{Z}^+$ *for* $i = 1, ..., n$. $\Delta(\lambda)$ *is described as*

below [2]:

$$\Delta(\lambda)=\begin{pmatrix} \lambda^{M_{\alpha_1}}-a_{11} & -a_{12} & \cdots & -a_{1n} \\ -a_{12} & \lambda^{M_{\alpha_2}}-a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & \vdots & \vdots \\ -a_{n1} & -a_{n2} & \cdots & \lambda^{M_{\alpha_n}}-a_{nn} \end{pmatrix}. \tag{1.5}$$

The system response described in (1) is asymptotically stable if all roots λ of equation $\det\Delta(\lambda)=0$ *satisfy the condition* $|\arg(\lambda)|>\frac{\alpha\pi}{2}$. *The matrix represents the characteristic matrix of the system (1.1), and* $\det\Delta(s)$ *corresponds to its polynomial characteristic.*

Definition 1.3. The fractional-order system is considered as follows [31]:

$$\frac{d^{\alpha_i}x_i}{dt_{\alpha_i}}=f_i(x_1,x_2,...,x_i), \qquad i=1,...,n. \tag{1.6}$$

In the equation above, all α_i coefficients have values between 0 and 1. The equilibrium point of the system (1.7) is acquired by solving the following equation [29]:

$$f_i(x_1,x_2,...,x_i)=0, \qquad i=1,...,n. \tag{1.7}$$

It is assumed that $x_1^*=(x_1^*,x_2^*,...,x_n^*)$ is the equilibrium point of the system (1.7) meaning $f_i=(x_1^*,x_2^*,...,x_n^*)$ [35]. Considering the values for i, the equation below is defined to evaluate the stability of the equilibrium point:

$$\varepsilon_i=x_i-x_i^*, \qquad i=1,...,n. \tag{1.8}$$

As the Caputo differentiation by a constant value is zero, we would conclude:

$$\frac{d^{\alpha_i}\varepsilon_i}{dt^{\alpha_i}}=f_i(x_1^*+\varepsilon_i,x_2^*+\varepsilon_i,...,x_n^*+\varepsilon_i), \qquad i=1,...,n. \tag{1.9}$$

If the second partial differentiation of function f_i around the equilibrium point x^* exists in the n-dimensional space of $\mathbb{R}^n$, the right-hand side of equation (1.9) could be rewritten as [35]:

$$f_i(x_1^*+\varepsilon_i,x_2^*+\varepsilon_i,...,x_n^*+\varepsilon_i)=f_i(x_1^*,x_2^*,...,x_n^*)+\left(\frac{\partial f_i}{\partial x_1}\Big|_{x^*},\frac{\partial f_i}{\partial x_2}\Big|_{x^*},...,\frac{\partial f_i}{\partial x_n}\Big|_{x^*}\right)\varepsilon+\bar{f}_i(\varepsilon). \tag{1.10}$$

In the equation above, $\varepsilon=(\varepsilon_1,\varepsilon_2,...,\varepsilon_n)^T$ and $\bar{f}_i(\varepsilon)$ consist of the higher-order terms of Taylor expansion that are neglected. In addition, it is assumed that we have $f_i(x_1^*,x_2^*,...,x_n^*)=0$ for $i=1,...,n$. As a result, we could conclude:

$$f_i(x_1^*+\varepsilon_i,x_2^*+\varepsilon_i,...,x_n^*+\varepsilon_i)\approx+\left(\frac{\partial f_i}{\partial x_1}\Big|_{x^*},\frac{\partial f_i}{\partial x_2}\Big|_{x^*},...,\frac{\partial f_i}{\partial x_n}\Big|_{x^*}\right)\varepsilon+\bar{f}_i(\varepsilon). \tag{1.11}$$

Furthermore, we could assume the following equation:

$$J\varepsilon=\begin{pmatrix} \frac{d^{\alpha_1}x_1}{dt_1^{\alpha}} \\ \frac{d^{\alpha_2}x_2}{dt_2^{\alpha}} \\ \vdots \\ \frac{d^{\alpha_n}x_n}{dt_n^{\alpha}}, \end{pmatrix}, \tag{1.12}$$

where we have $f=(f_1,...,f_n)^T$ and $J=\frac{\partial f}{\partial x}|_{x^*}$. It is assumed that M is the least common multiple of α_i that is defined as $\alpha_i=\frac{v_i}{u_i}$, $(u_i,v_i=1)$, and $u_i,v_i\in\mathbb{Z}^+$ for $i=1,...,n$.

According to Theorem 1.2, if $|\arg(\lambda)|>\frac{\alpha\pi}{2}$ for all λ calculated by the equation below, the equilibrium point $x=x^*$ of the system (1.7) is asymptotically stable [26]:

$$\det\left(\operatorname{diag}\left(\lambda^{M_{\alpha_1}} \quad \lambda^{M_{\alpha_2}} \quad \lambda^{M_{\alpha_n}}\right)-J\right)=0. \tag{1.13}$$

It should be noted that $\operatorname{diag}(m_1 \quad m_2 \quad \cdots \quad m_n)$ is represented by a square $n \times n$ matrix as below:

$$\operatorname{diag}(m_1 \quad m_2 \quad \cdots \quad m_n) = \begin{pmatrix} m_1 & 0 & \cdots & 0 \\ 0 & m_2 & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & \cdots & m_n \end{pmatrix}. \tag{1.14}$$

1.2. **The required conditions for the presence of chaos in a fractional-order system.** The saddle point is an equilibrium point in a three-dimensional integer-order system with at least one eigenvalue at the stable region (the left-hand part of the imaginary axis) [6, 11, 16] and at least one eigenvalue in the unstable area (the right-hand part the imaginary axis). This saddle point is called saddle point of kind one if one of the eigenvalues is unstable and the others are stable. Alternatively, if one eigenvalue is stable while two others are unstable [22], the saddle point is of kind two [11, 18, 36].

The chaotic behavior in a chaotic system is demonstrated around a saddle point of kind two. The chaotic behavior could also be observed around a saddle point of kind 2 in a three-dimensional fractional-order system, just as the three-dimensional integer order one [24, 38]. It is considered that the chaotic three-dimensional system of the form $\dot{x} = f(x)$ have chaotic attractors. It is also assumed that Ω is a set of equilibrium points of the system surrounded by a twisting [10]. On the other hand, the $D^\alpha x = f(x)$ system with defined $D^\alpha = \left(\frac{d_1^\alpha}{dt_1^\alpha}, \frac{d_2^\alpha}{dt_2^\alpha}, \frac{d_3^\alpha}{dt_3^\alpha}\right)$ and the system $\dot{x} = f(x)$ have equal equilibrium points [18]. Therefore, the required condition for a fractional order system of $D^\alpha x = f(x)$ to have chaotic attractor is stated as the following equation [40]:

$$\frac{\pi}{2M} - \min|\arg(\lambda_i)| \geq 0, \tag{1.15}$$

where λ_i are the roots of the equation below:

$$\det\left(\left(\lambda^{M_1^\alpha} \quad \lambda^{M_2^\alpha} \cdots \lambda^{M_n^\alpha}\right)\right) - \frac{\partial f}{\partial x}\Big|_{x-x^*} = 0, \qquad x^* \in \Omega. \tag{1.16}$$

The system's behavior around this point cannot tend to a chaotic attractor if the system has a stable equilibrium point, and the initial conditions related to the system do not lie inside the attracting region [28]. In other words, the system cannot have a chaotic behavior for any initial condition, and some of the initial conditions do not actually represent chaotic behavior. In general, there is not a specific mathematical relation to the present attracting region. By assuming $\alpha_1 = \alpha_2 = \alpha_3 = \alpha$, the condition of being chaotic for the fractional-order system of (1.7) could be stated as follows:

$$\alpha \geq \frac{2}{\pi}\min|\arg(\lambda_i)|, \tag{1.17}$$

where λ_i are the eigenvalues of the Jacobian matrix that is defined as $\frac{\partial f}{\partial x}\Big|_{x-x^*} = 0$ for every $x^* \in \Omega$ [35]. The relation (1.17) states the necessary condition for chaos to occur in a fractional-order system. This relation could be used to acquire the minimum order of the system for which the chaotic behavior could occur.

1.3. **Optima control.** Optimal control of a system means changing the inputs and properties of the system to obtain the best outputs or results. Classical methods of designing control systems are usually the trial and error methods [27]. In creating a controller for a structure, we use iterative analytical methods, which are very tedious. The trial and error method is not acceptable for systems with multiple inputs or outputs satisfying different criteria or functions. In solving such optimization problems, we apply the optimal control method in which defining variables and optimization goals are of great importance. In most optimization problems, minimizing the value of a cost function is considered, and selecting the appropriate objective function is a primary optimization step that depends on the nature of the problem. It is important to note that there are sometimes different solutions to a problem. To select the optimal solution, we should compare them with the objective function. In general, there are three steps in solving the optimal control problem.1. It is necessary to achieve an accurate understanding of the system, which means a definite observation of system performance, a reasonable judgment about system variables, and a clear comprehension of interaction between variables; in other words, the ability to determine the specific mathematical model for the system.2. The optimization

goal is to improve the system behavior to meet the expected optimization criteria. We must define the optimization criteria for the system, which is a mathematical function. When the criterion function becomes minimum (maximum), the system is in its desired state.3. We search for the optimal values of the system variables after determining an appropriate method for solving the optimal control. In analyzing the optimal control of a system, the mathematical model of the system should be expressed in the form of state-space because state space is applicable for both linear and nonlinear systems; in addition, it is easier to work with state variables in numerical calculations and when using computers [18].

2. Research method

In this chapter, we studied the optimal control problem of the HIV cell fractional-order system. After considering a model for HIV from reference, the model was performed based on ordinary differential equations [15]. We selected the system parameters so that the system was in a chaotic zone. Then, using the stability theorem for fractional-order systems, the system order was estimated so that it was in its chaotic region [21]. Eventually, we were ready to solve the specified fractional-order model by the particle swarm optimization and genetic algorithms.

2.1. **HIV modelling.** Although numerous efforts have been made to treat HIV, there is still no definite therapy for the condition, which can merely extend the patient's life expectancy. According to current studies, there are two kinds of medications for treating this disease: RTI, which inhibits infected cells from being produced, and PTI, which prevents the formation of free viruses in the blood. Different models for the viral propagation model have been developed, with the model with three state variables being employed in this work. In equation (2.1), the integer order used in the present research has been expressed [10, 13, 41]:

$$\begin{cases} \dfrac{dx(t)}{dt} = \lambda - \delta x(t) - (1-\gamma)\beta x(t)z(t), \\ \dfrac{dy(t)}{dt} = (1-\gamma)\beta x(t)z(t) - \alpha y(t), \\ \dfrac{z(t)}{dt} = (1-\eta)N\alpha y(t) - uz(t) - (1-\gamma)\beta x(t)z(t). \end{cases} . \tag{2.1}$$

As shown, the concerned system has three state variables $x(t), y(t)$, and $z(t)$ each indicates the number of uninfected cells, infected cells, and viruses, respectively. Table 1 displays the remaining system parameters and simulation values:

Table 1. System values and parameters

Parameters		Values
x_0	Initial concentration of uninfected cells	1000 dm^{-3}
y_0	Initial concentration of infected cells	1000 dm^{-3}
z_0	Initial concentration of virus particles	1000 dm^{-3}
$(1-\gamma)$	The reverse transcriptase inhibitor drug effect	0.5
$(1-\eta)$	The protease transcriptase inhibitor drug effect	0.5
λ	The total rate of production of healthy cells	10^6 day^{-1} dm^{-3}
β	Rate uninfected become infected with virus	10^{-8} day^{-1} dm^{-3}
δ	Nature death rate of healthy cells	0.1 day^{-1}
α	Nature death rate of infected cells	0.5 day^{-1}
u	Nature death rate of infective virus particles	5 day^{-1}
N	Number of virus produced by infected cells	Varies

The majority of HIV control research and simulations have employed integer order-based systems, however, the simulations in this article have exploited the fractional order of the disease (1.7), the equations of which are as follows

Algorithm 1 Particle swarm optimization

Initialize Swarm;
repeat
  forall *particles* **do**
    calculate fitness f
  end
  forall *particles* **do**
    $v_{t+1} = v_t + \phi_1(b - x) + \phi_2(g - x)$;
    $x_{t+1} = x_t + v_{t+1}$;
  end
  $t = t + 1$
until *stopping criteria*;

FIGURE 1. Algoritm of particle swarm optimization [39].

[44]:

$$\begin{cases} D^\alpha x(t) = \lambda - \delta x(t) - (1-\gamma)\beta x(t)z(t), \\ D^\alpha y(t) = (1-\gamma)\beta x(t)z(t) - \alpha y(t), \\ D^\alpha z(t) = (1-\eta)N\alpha y(t) - uz(t) - (1-\gamma)\beta x(t)z(t). \end{cases} \tag{2.2}$$

2.2. **Particle swarm optimization algorithm.** Evolutionary algorithms have been successfully applied to solve many complex optimization engineering problems. Together with genetic algorithms, the PSO algorithm, proposed by Kennedy and Eberhart, has achieved considerable success in solving optimization problems [19]. The PSO algorithm was proposed originally in [19]. This optimization technique is inspired by the way swarms behave and its elements move in a synchronized way, both as a defensive tactic and for searching food. An analogy is established between a particle and a swarm element. The particle movement is characterized by two vectors, representing its current position x and velocity v. Since 1995, many techniques were proposed to refine and/or complement the original canonical PSO algorithm, namely by analyzing the tuning parameters and by considering hybridization with other evolutionary techniques. In literature, some work embedding FC and PSO algorithms can be found. Algorithm 1 illustrates a standard PSO algorithm. The basic algorithm begins by initializing the swarm randomly in the search space. As it can be seen in the pseudo-code, were t and $t+1$ represent two consecutive iterations, the position x of each particle is updated during the iterations by adding a new velocity v. This velocity is evaluated by summing an increment to the previous velocity value. The increment is a function of two components representing cognitive and the social knowledge [33]. The cognitive knowledge of each particle is included by evaluating the difference between its best position found so far b and the current position x. On the other side, the social knowledge, of each particle, is incorporated through the difference between the best swarm global position achieved so far g and its current position x. The cognitive and the social knowledge factors are multiplied by randomly uniformly generated terms ϕ_1 and ϕ_2, respectively. PSO is an optimization algorithm that proves to be efficient, robust and simple. However, if no care is taken the velocities may attain large values, particularly when particles are far away from local and global bests. Some approaches were carried out to eliminate this drawback. Eberhat proposed a clamping function to limit the velocity, through the expression [4, 7]:

$$v_{ij}(t+1) = \begin{cases} v'_{ij}(t+1) & \text{if } v'_{ij}(t+1) < v_{\max j}, \\ v_{\max j} & \text{if } v'_{ij}(t+1) \geq v_{\max j}, \end{cases}$$

where $v'_{ij}(t+1)$ results from $v'_{ij}(t+1) = v_{ij}(t) + \phi_1(b-x) + \phi_2(g-x)$ for the parameter j of particle i at iteration $t+1$. Later, a constant, the inertia weight, was introduced to control the velocity from exploding (1.5). The inertia

weight ω is very important to ensure convergence behavior over evolution by adopting the equation:

$$v_{i+1} = \omega v_i + \phi_1(b - x) + \phi_2(g - x).$$

Some empirical and theoretical studies were made to determine the best inertia value in order to obtain better PSO behavior. Later the same authors indicated that initializing ω to 0.9 and reducing it linearly to 0.4, during the evolution, would allow initial exploration toward a local search as iterations follow. An alternative technique, to prevent velocity explosion, proposed by Clerc and Kennedy uses a constriction χ in place of clamping the velocity:

$$v_{i+1} = \chi(v_i + \beta_1\phi_1(b - x) + \beta_2\phi_2(b - x)), \qquad \chi = \frac{2}{\left|2 - \phi - \sqrt{\phi^2 - 4\phi}\right|},$$

where $\phi = \phi_1 + \phi_2 > 4$.

The algorithm is given in the following steps:

Step 1: Initialize swarm: Randomly generate bounded real values to form an initial swarm of particles. Each particle represents the unknown parameters of the neural network. The initial swarm is scattered enough for better search space of the algorithm.

Step 2: Initialization: Following parameter values assigned for algorithm execution. Set the number of flights. Set the fitness limit and start cycle count. Set the values of individual best and global best acceleration factors. Set the value of inertia weight / and maximum velocity v_{max}.

Step 3: Fitness Evaluation: Calculate fitness by using the fitness function.

Step 4: Ranking: Rank each particle of the swarm on the basis of minimum fitness values. Store the best fitness particle.

Step 5: Termination Criteria: Terminate the algorithm if either predefined fitness value, i.e., MSE 10-08 for linear FDEs and 10-04 for non-linear FDEs is achieved or number of maximum flights/cycles is reached. If yes go to Step 7 else continue.

Step 6: Renewal: Update the Velocity using the equation. Update the position using the equation and then Repeat the procedure from Step 3 to Step 6 until the total number of flights is reached.

Step 7: Storage: Store the best fitted particle so far and name it as the global best particle for this run.

Step 8: Refinement: `MATLAB` optimization toolbox is used for simulating annealing algorithm for further fine-tuning by taking the best fitted particle as the start point of the algorithm. Store the value of the fitness along with the best particle for this run. Stop. The steps of the particle swarm optimization algorithm and its flowchart can be seen in Figure 2.

3. Simulation of chaotic model

The behavior of many physical systems can be modeled by fractional dynamics. In general, the description of systems with fractional differential equations provides researchers with a better and more accurate description. In the field of system control, fractional order controllers have more flexibility than classical controllers. To control HIV disease, an optimal fractional order controller is used and its simulation is used with `MATLAB` software. For optimal control of controller parameters, particle swarm optimization algorithm and genetic algorithm are used.

3.1. **Steps of performing a genetic algorithm.** Get a series of required inputs to run the algorithm. Initialize the problem. Set up empty vectors to store position values (optimized parameter values) and the value of the objective function. Create a starting initial for the loop to randomly take the initial population and the cost function values for that initial population. Arrange randomly made population. Unearth the best answer among the initial random population. Keep the cost function values for the best and the worst states. Start the main computation loop. Choose some parents at each step. Determine the best genes and offspring at each step. Calculate mutations. Establish the population and mutation unit vector. Order the population in the fundamental loop. Get the worst case (in terms of cost function value). Determine the best answer, which is the terminal solution.

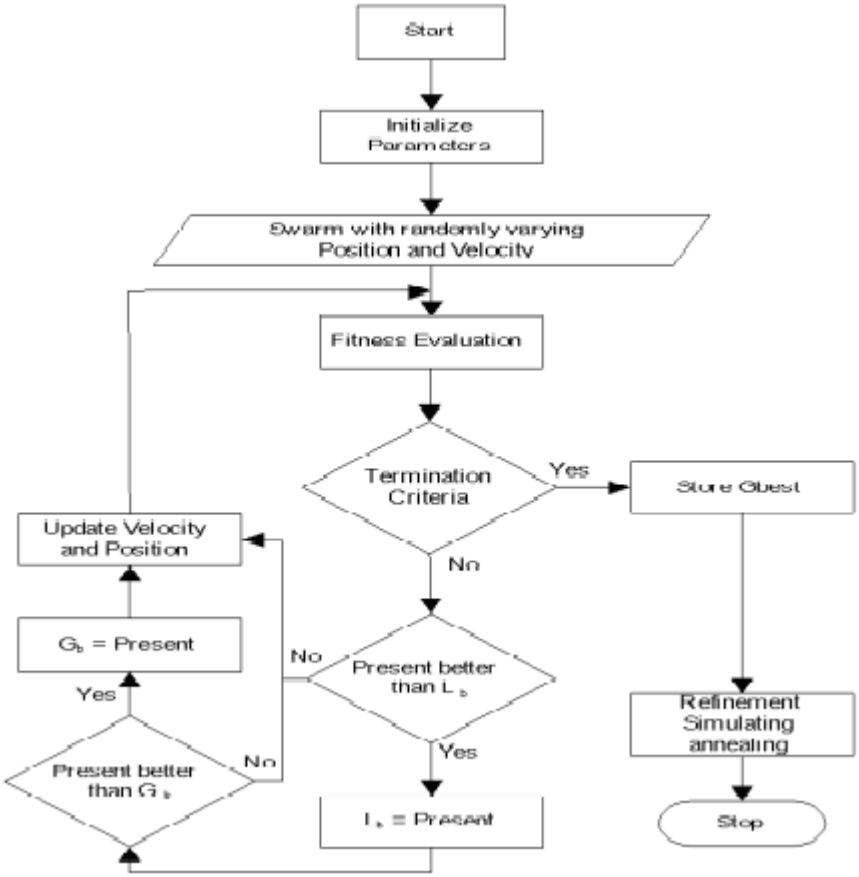

FIGURE 2. Flowchart of particle swarm optimization algorithm.

3.2. **Steps of implementing Particle Swarm Optimization (PSO) algorithm.** Receive necessary parameters and perform initial calculations (and determine the upper and lower particle speed limits). Initialize the values to the optimal variables and the value of the cost function. There is an elementary for loop to a randomly generated population. Sort the values of the optimal variables and the cost function of that initial population. Begin the fundamental loop of the algorithm. Update particle speed. Employ particle speed limits. Update particle position (optimal variables values). Make negative the velocity of those particles outside the allowed speed range to fit in the scale. Restrict the particle's position. Calculate cost function values. Find the best position. In other words, to minimize the value of the cost function, find the values of the optimal variables.

4. Results and simulation

As pointed out, the three variables in the system simulated in this study are the number of healthy cells, the number of diseased cells, and the number of viruses, to regulate HIV to increase the number of healthy cells and lower the number of viruses. Accordingly, the concerned system is both non-linear and multi-objective. To control the illness, an optimal PD controller is utilized, and the algorithm optimizes the coefficients of this controller. This controller's optimization problem is a multi-criteria problem, which means that two and one criteria should be minimized and maximized, respectively. Therefore, the provided cost function is incorporated as a penalty, and if the suggested requirements are not met, the algorithm will repeat the optimization process the suggested controller is constructed in such a way that the controller input is the number of infected cells, and the controller output is added to this variable.

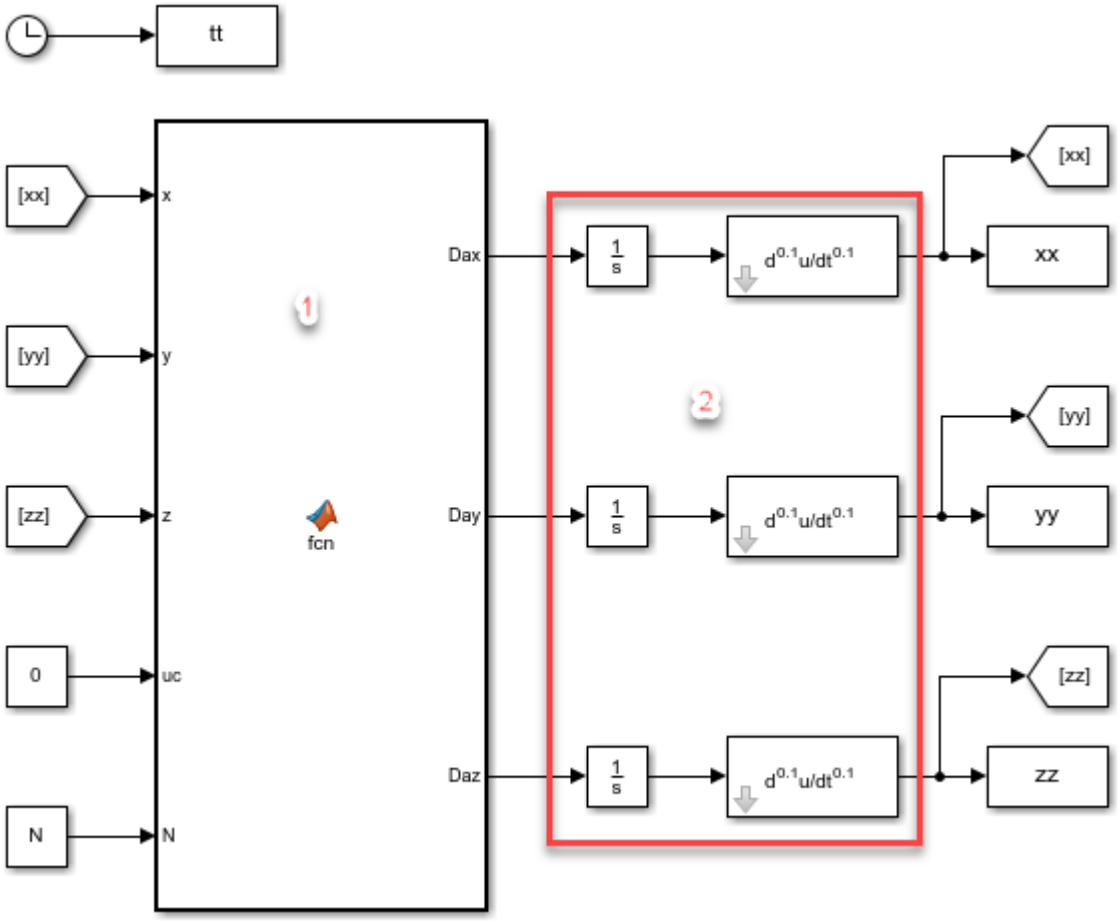

FIGURE 3. The Block diagram of the simulated system.

4.1. **Simulink model (Hiv model without control).** Since the model for the article is a fractional order and not correct, the fractional order derivative block must be used. Since `MATLAB` coding is considered, the main part of Simulink models is `MATLAB` coding (in `MATLAB` function block), but next to it, the fractional derivative block is also used. The main page of this block is as follows in Figure 3.

Part 1: is the same `MATLAB` coding in which the constants of Table 1 and equation (1.2) are written.The block diagram of the simulated system in `MATLAB` is shown in Figure 3. Reference [44] has been used in this model, which according to the Liapunov Function, the system's stability is determined by the number of healthy production cells, therefore if the number of viruses produced by the cell is fewer than 202 ($N < 202$), the system is stable. In other words, the system becomes stable if the number of healthy cells rise while the number of infected cells and viruses decline to zero. That is, the number of healthy cells declines while the number of infected cells and viruses rises. The suggested HIV controller works in such a manner that it remains stable for various quantities of N. According to the description, the control input is added to the y equation, which in this simulation is named uc in Figure 6.

Part 2: is equivalent to the following part of the article in Figure 5:

```
1   function [Dax,Day,Daz] = fcn(x,y,z,uc,N)
2   %   constants
3   gamma = 0.5;
4   eta = 0.5;
5   lambda = 1e6;
6   beta = 1e-8;
7   delta = 0.1;
8   a = 0.5;
9   u = 5;
10
11  %   equation 2
12  Dax = lambda - delta*x - (1 - gamma)*beta*x*z;
13  Day = (1 - gamma)*beta*x*z - a*y + uc;
14  Daz = (1 - eta)*N*a*y - u*z - (1 - gamma)*beta*x*z;
15  end
```

FIGURE 4. The control input is added to the y equation.

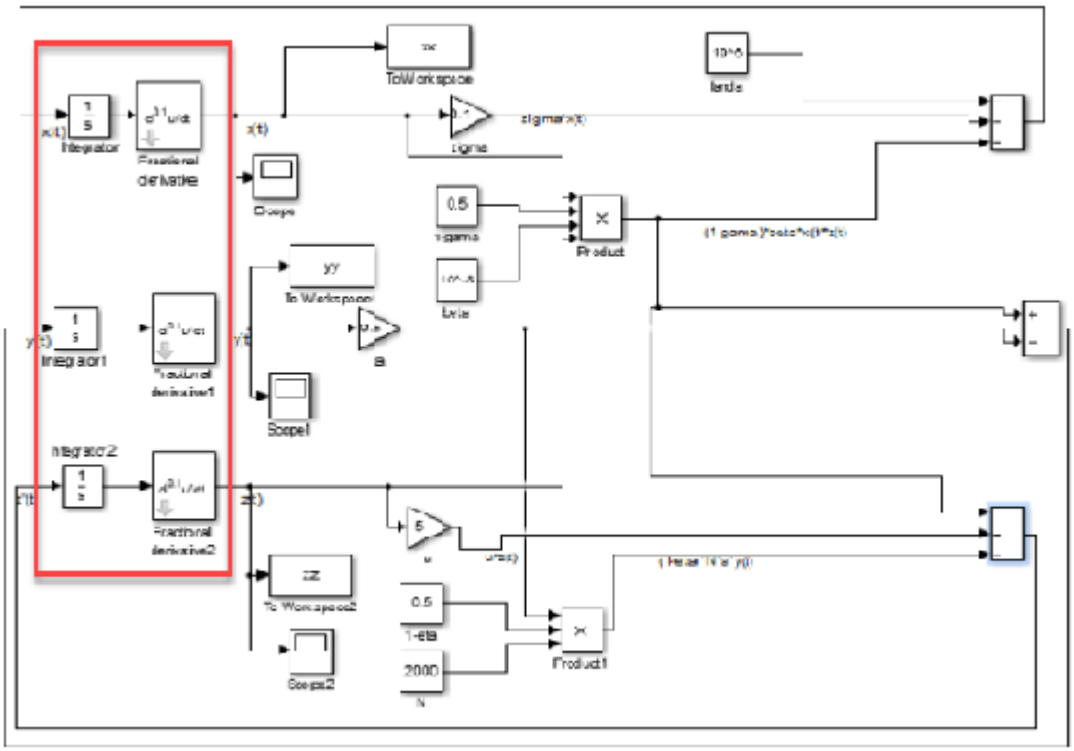

FIGURE 5. The control input is added to the y equation.

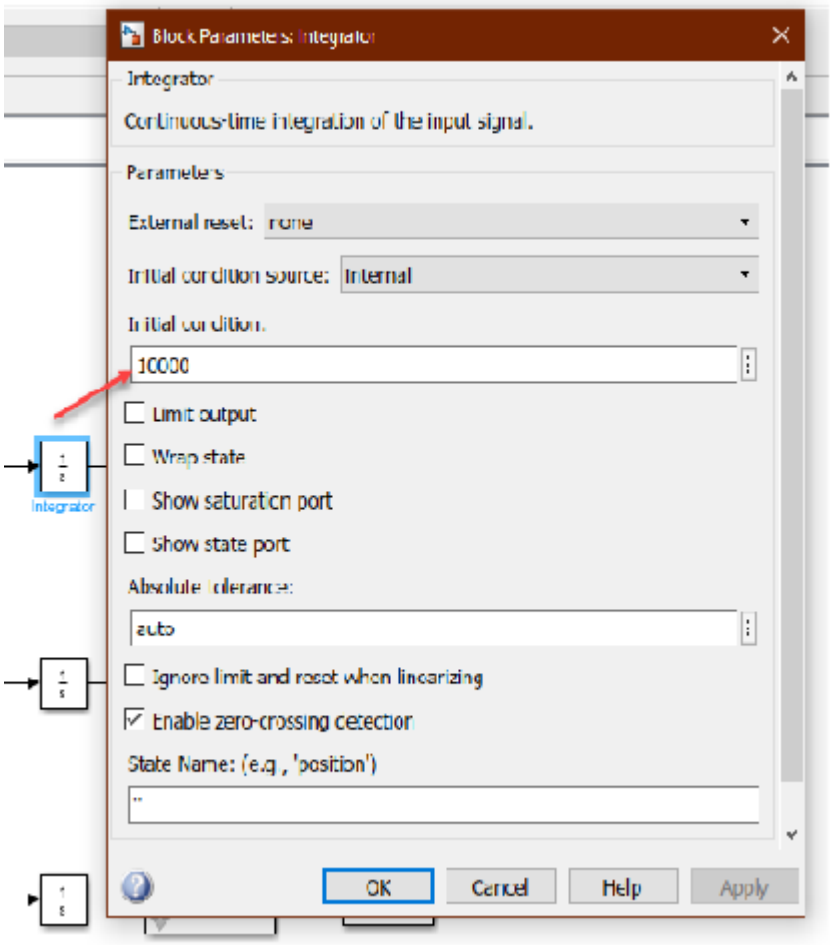

FIGURE 6. The initial values $x(0)$, $y(0)$ and $z(0)$.

In integral blocks, the initial values $x(0), y(0)$, and $z(0)$ are placed according to Table 1. The important point of this block is Part 3, which is the control input, which is equal to 0 (the problem is out of control) in Figure 6.

4.2. **Simulink model (HIV model PID control).** Everything is the same as before. Only in this model, the PID controller block is placed. See Figure 7.

The input of this block is taken from equation y. There are also 3 saturated blocks (purple rectangle). These blocks are placed only to ensure that the state variables do not become negative and the optimization does not diverge, and it is observed that their upper limit is infinite, Figure 8. The PID controller coefficients are also set as unknowns (not fixed numbers) as shown below, and the values of these unknowns will be obtained by solving the optimization problem. See Figure 9. View each of the graphs by selecting the value N and the type of problem (1 to 3). for example:

4.3. **Uncontrolled mode with $N = 100, 400$ and 2000.** There are the results of a simulation for a system with a fractional order of 0.9, which is considered to be $N = 100, 400$ and 2000 in the first simulation (4). As mentioned, because the system is stable for Ns less than 202, no controller is required. See Figures 10, 11, and 12.

4.4. **Control results.** For example, the results for each optimization method are shown with $N = 2000$ in Figure 13. You can easily produce other results. Note that due to the nature of an optimization problem, the controller

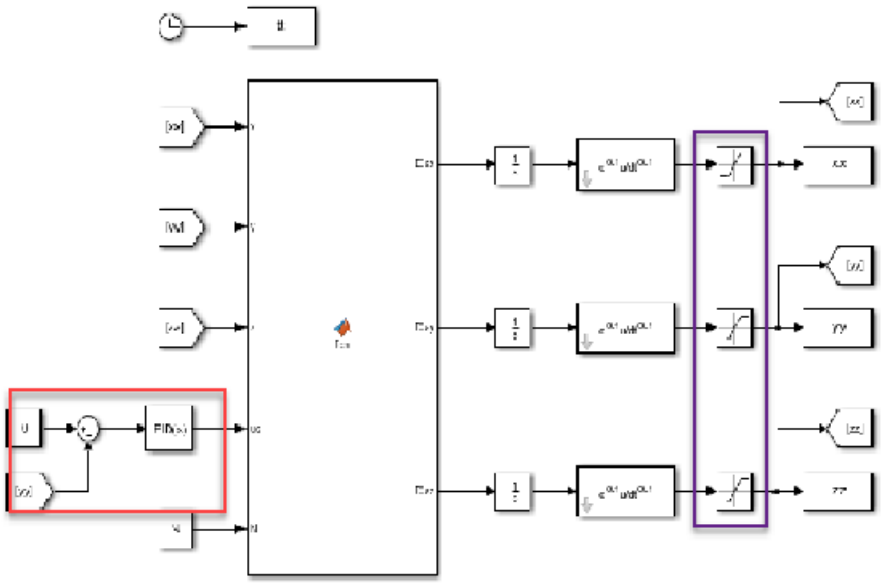

FIGURE 7. The PID controller block.

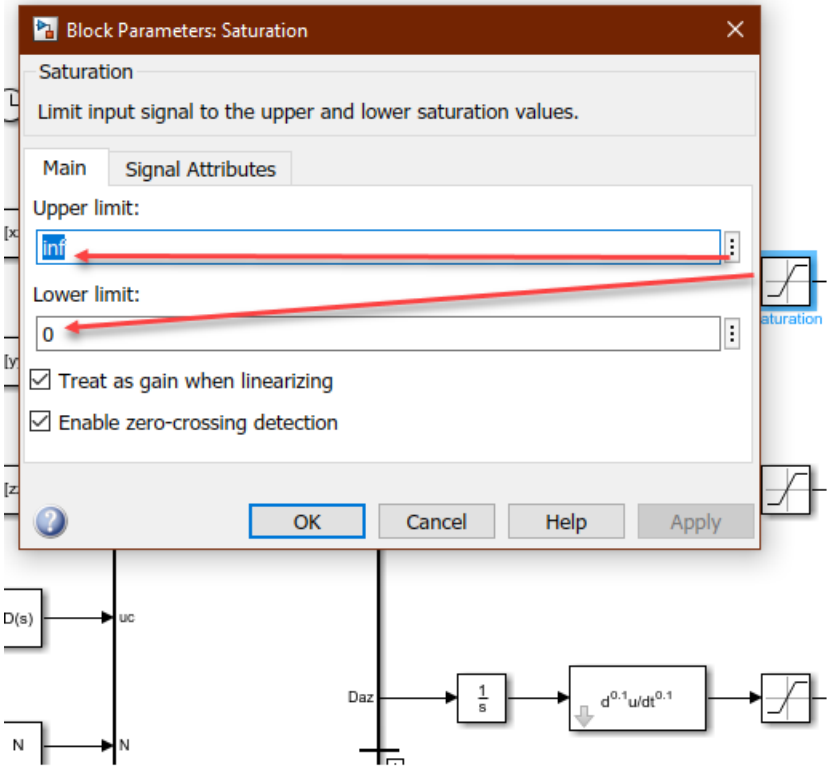

FIGURE 8. Limit input signal to the upper and lower.

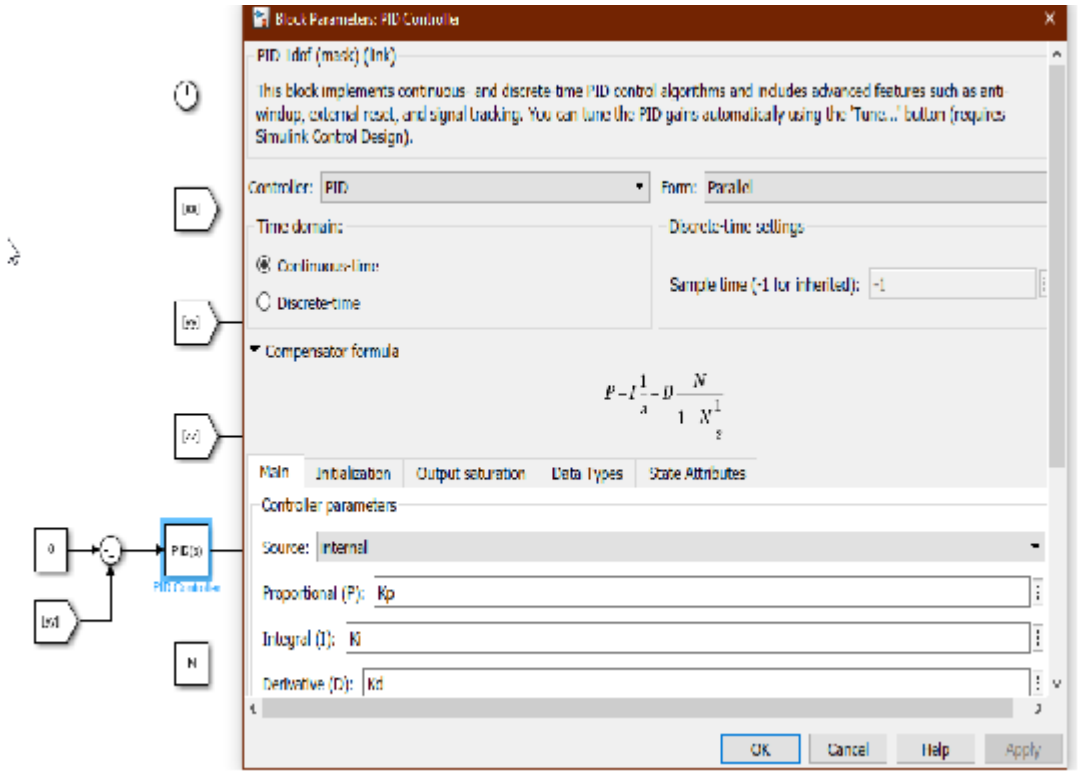

FIGURE 9. The PID controller coefficients.

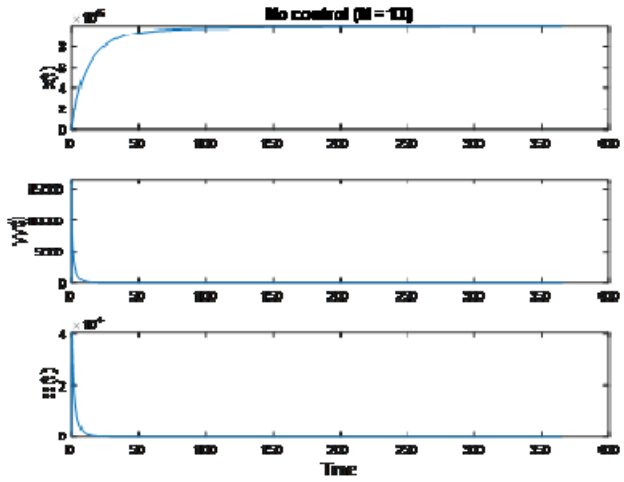

FIGURE 10. The function of the fractional order system without controller per $N = 100$.

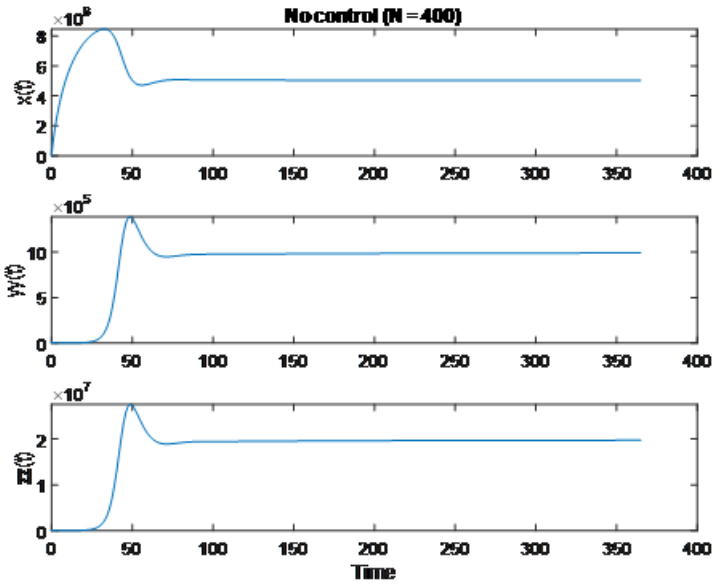

FIGURE 11. The function of the fractional order system without controller per $N = 400$.

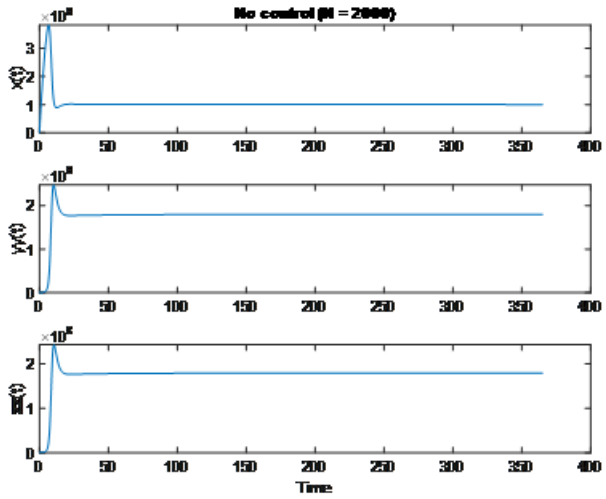

FIGURE 12. The function of the fractional order system without controller per $N = 2000$.

coefficients may be slightly different in each run, and as a result the exact results may not be the same, but the important point is that all the results are the same. Also, the optimal simulation time is so long that you have to wait.

4.4.1. *GA control problem with* $N = 2000$.

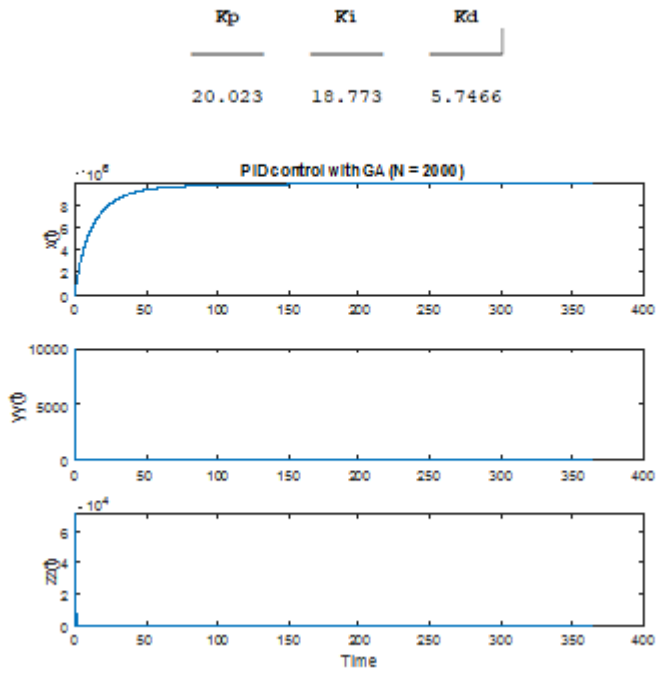

FIGURE 13. PID control with GA ($N = 2000$).

4.4.2. *PSO control problem with* $N = 2000$. Other results can be generated. Due to the nature of an optimization problem, the controller coefficients may be slightly different in each run. The optimal simulation time will be long. Also, The results for PID control with PSO ($N = 2000$) method are shown in figure 14.

In the second simulation, $N = 400$, the system is expected to be unstable in this case, according to the equations expressed in reference [44], so that the two variables of the number of unhealthy cells and the number of infected viruses reach zero, which can be reduced to zero using the proposed controller.The number of healthy cells has reduced from 10×10^6 to 5×10^6, while the number of infected cells has climbed from zero to 1×10^6 and the number of viruses has increased from zero to 2×10^7.The performance of the system with the PD controller can be seen, with the controller being able to raise the number of healthy cells, as well as the number of infected cells and viruses, to zero. In the last simulation, the relevant system is simulated with $N = 2000$, and the results for both modes with and without controller are displayed, as in the prior example. Similarly, as expected, as the N value rises, the controller-free system becomes unstable. Figures depict the system's performance with the controller, demonstrating that the controller developed for this case was likewise capable of controlling the system. The suggested controller, as shown in Figures is meant to reduce the number of viruses and infected cells to zero in the quickest feasible period while also increasing the number of healthy cells. According to the simulations and references utilized, the examined system is dependent on the number of generated cells, therefore if the number of cells is less than 202, the system is stable, and the system's stability may be understood as disease control. Furthermore, if this number of cells exceeds 202, the system becomes unstable, and as this number rises, the illness spreads faster; nevertheless, the constructed controller can re-control the system in diverse scenarios, such as controlling HIV.

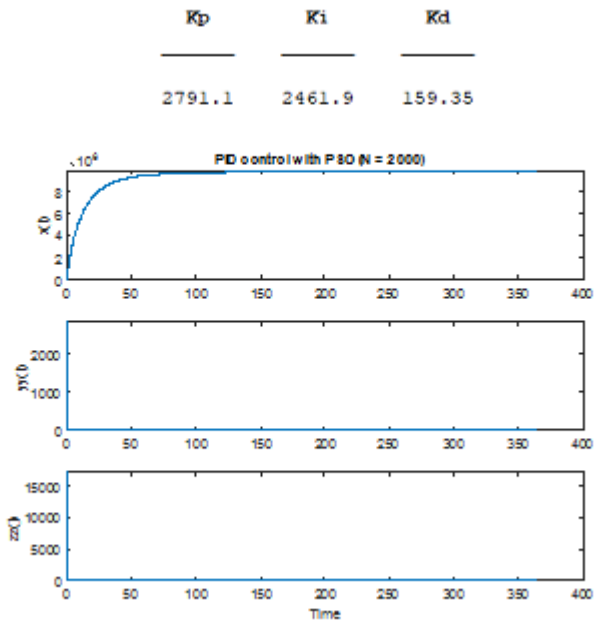

FIGURE 14. PID control with PSO ($N = 2000$).

5. Conclusion

Because most real-world systems are in the form of fractions, the science of employing fractional calculations has gotten a lot of attention. Therefore, modeling and simulation of systems in fractional order have become quite essential. The HIV fractional order model is utilized to simulate in this work. The system in question is nonlinear, and its stability is determined by the system's initial conditions; for example, if the number of healthy cells is less than 202, the system is inherently stable; if it is greater, the system is unstable; thus, the higher this value, the more unstable the system is. An optimal fractional order PD controller was utilized to control HIV. Particle swarm optimization and genetic algorithms were used to optimize the controller parameters. The suggested controller's role is to ensure that the system is stable in all system modes and does not depend on the quantity of healthy generated cells.

References

[1] E. Ahmad, A. M. A. El-Sayed, and H. A. A. El-Saka, *Equilibrium points, stability and numerical solutions of fractional order predator-prey and rabies models*, J. Math. Anal. Appl., *325* (2007), 542–553.

[2] S. Alavi, A. Haghighi, A. Yari, and F. Soltanian, *A numerical method for solving fractional optimal control problems using the operational matrix of Mott polynomials*, Comput. Methods Differ. Equ., *10*(3) (2022), 755–773.

[3] F. A. Alazabi, M. A. Zohdy, and A. A. Ezzabi, *A regulator design for nonlinear HIV-1 infection system*, Int. J. Intel. Cont. Sys., *17*(1) (2012), 1–6.

[4] M. Arablouye Moghaddam, Y. Edrisi Tabriz, and M. Lakestani, *Solving fractional optimal control problems using Genocchi polynomials*, Comput. Methods Differ. Equ., *9*(1) (2021), 79–93.

[5] N. Bellomo and M. Delitala, *The mathematical kinetic, and stochatic game theroy to modelling mutations, progression and immune competition of cancer cells*, Phys. Life Rev. *5* (2008), 183–206.

[6] M. Clerc and J. Kennedy, *The particle swarm–explosion, stability, and convergence in a multidimensional complex space*, IEEE Trans. Evol. Comput., *6*(1) (2000), 58–73.
[7] L. Cai, J. Liu, and Y. Chen, *Dynamics of an age-structured HIV model with super-infection*, Appl. Comput. Math., Int. J., *20*(2) (2021), 257–276.
[8] V. Daftardar-Gejji and S. Bhalekar, *Chaos in fractional ordered Liu system*, Comput. & Math. Appl., *59*(3) (2010), 1117–1127.
[9] W. Deng, C. Li, and J. lU, *Stability analysis of linear fractional differential system with multiple time delays*, Nonlinear Dyn., *48* (2007), 409–416.
[10] Y. Ding, Z. Wang and H. Ye, *Optimal control of a fractional-order HIV-immune system with memory*, IEEE Transactions On Contril Systems Technology, *20*(3) (2012), 763–769.
[11] R. Eberhart and Y. Shi, *Comparing inertia weights and constriction factors in particle swarm optimization*, In: Proceedings of the 2000 Congress on Evolutionary Computation, Washington DC, *1* (2000), 84–88.
[12] G. B. Fogel, E. S. Liu, M. Salemi, S. L. Lamers, and M. S. McGrath, *Evolved Neural Networks for HIV-1 Co-receptor Identification* , 2014 IEEE Congress on Evolutionary Computation (CEC), July 6-11, 2014, Beijing, China.
[13] M. Gachpazan, A. H. Borzabadi, and A. V. Kamyad, *A measure-theoretical approach for solving discrete optimal control problems*, Appl. Math. Comput., *173* (2) (2006), 736–752.
[14] S. Ghasemi, A. Tabesh, and J. Askari, *Application of fractional calculus theory to robust controller design for wind turbine generators*, IEEE Transactions on Energy Conversion, *29*(3) (2014).
[15] A. E. Gohary and I. A. Alwasel, *The chaos and optimal control of cancer model with complete unknown parameters*, Chaos, Solitons and Fractals, *42* (2009), 2865–2874.
[16] A. S. Hegazi, E. Ahmed, and A. E. Matouk, *On chaos control and synchronization of the commensurate fractional order Liu system*, Commun. Nonlinear Sci. Num. Simul., *18*(5) (2013), 1193–1202.
[17] H. Y. Jia, Z. Q. Chen, and G. Y. Qi, *Chaotic characteristics analysis and circuit implementation for a fractional-order system*, IEEE Transactions on circuits and systems–I: Regular paper, *61*(3) (2014).
[18] N. Kanagaraj and V. N. Jha, (2021), *Design of an enhanced fractional order PID controller for a class of second-order system*, COMPEL-The Int. J. Comput. Math. Electrical and Electronic Engin., *40*(3) (2021), 579–592.
[19] J. Kennedy and R. C. Eberhart, (1995)*Particle swarm optimization. In: Proceedings of the 1995 IEEE International Conference on Neural Networks*, Perth, Australia, *4* (1995), 1942–1948. IEEE Service Center, Piscataway.
[20] M. Khanra, J. Pal, and K. Biswas, *Reduced order approximation of MIMO fractional order systems*, IEEE J. on Emerging and selected topics in circuits and systems, *3*(3) (2013).
[21] X. Liu, L. Hong, and L. Yang, *Fractional-order complex T system: bifurcations, chaos control, and synchronization*, Nonlinear Dyn., *75* (2014), 589–602.
[22] X. Liu, L. Hong, L. Yang and D. Tang, *Bifurcations of a new fractional-order system with a one-scroll chaotic attractor*, Discrete Dynamics in Nature and Society, *2019* (2019), Article ID 8341514, 15 pages.
[23] D. Matiognon, *Stability results for fractional differential equations with applications to control processing in: Computational engineering in systems and application multi conference*, IMACS IEEE-SMC Proceedings, *2* (1996), 963–968.
[24] S. Z. Moussavi, M. Alasvandi, S. Javadi, and E. Morad, *PMDC motor speed control optimization by implementing ANFIS and MRAC*, Int. J. Cont. Sci. Engin., *4*(1) (2014), 1–8.
[25] E. L. Mazzaferri and S. M. Jhiang, *Long-term impact of initial surgical and medical therapy on papillary and follicular thyroid cancer*, The American Journal of Medicine, *97*(5) (1994), 418–428.
[26] F. Merrikh-Bayat, *More details on analysis of fractional-order Lotka-Volterra equation*, The 5th IFAC symposium on fractional differentiation and its applications (FDA12), 14-17 May 2012, Hohai University, Nanjing, China, arXiv:1401.0103.
[27] S. Micael, N. M. Couceiro, F. Ferreira, and J. A. Tenreiro Machado, Control optimization of a robotic bird, Computer Science, (2009).
[28] K. Moaddy, A. Freihat, M. Al-Smadi, et al., *Numerical investigation for handling fractional-order Rabinovich–Fabrikant model using the multistep approach*, Soft Comput., *22* (2018), 773–782.

[29] I. Petras, *A note on the fractional-order Volta's system*, Commun. Nonlinear Sci. Num. Simul., *15*(2) (2010), 384–393.
[30] A. Razminia, V. J. Majd, and D. Baleanu, *Chaotic incommensurate fractional order Rössler system: active control and synchronization*, Adv. Differ. Equ., *15* (2011).
[31] S. Sayyad Delshad, M. Mostafa Asheghan, and M. Hamidi Beheshti, *Synchronization of N-coupled incommensurate fractional-order chaotic systems with ring connection*, Commun. Nonlinear Sci. Num. Simul., *16*(9) (2009), 3815–3824.
[32] M. Shahiri, R. Ghaderi, A. Ranjbar N. S. H. Hosseinnia, and S. Momani, *Chaotic fractional-order Coullet system: Synchronization and control approach*, Commun. Nonlinear Sci. Num. Simul., *15*(3) (2010), 665–674.
[33] Y. Shi and R. Eberhart, *A modified particle swarm optimizer. In: Evolutionary Computation Proceedings*, IEEE World Congress on Computational Intelligence, The 1998 IEEE International Conference on, Anchorage, Alaska, (1998), 69–73.
[34] M. S. Tavazoei and M. Haeri, *A necessary condition for double scroll attractor existence in fractional-order systems*, Phys. Let. A, *367*(1–2) (2007), 102–113.
[35] M. S. Tavazoei and M. Haeri (2008), *Chaotic attractors in incommensurate fractional order systems*, Phys. D, *237*(20) (2008), 2628–2637.
[36] M. S. Tavazoei and M. Haeri, *Limitations of frequency domain approximation for detecting chaos in fractional order systems*, Nonlinear Anal.: Theo., Meth. & Appl., *69*(4) (2008),1299–1320.
[37] M. S. Tavazoei, M. Haeri, M. Attari, S. Bolouki, and M. Siami, *More details on analysis of fractional-order Van der Pol oscillator*, J. Vibration Cont., *15*(6) (2009), 803–819.
[38] M. S. Tavazoei, M. Tavakoli-Kakhki, and F. Bizzarri, *Nonlinear fractional-order circuits and systems: motivation, A brief overview, and some future directions*, IEEE Open Journal of Circuits and Systems, *1*, 220–232.
[39] J. Wang Gary, G. Yen Marios, and M. Polycarpou, *9th International symposium on neural networks*, Shenyang, China, July 11-14, 2012. Proceedings, Part I.
[40] Z. Wang, X. Huang, and G. Shi,(2011)*Analysis of nonlinear dynamics and chaos in a fractional order financial system with time delay*, Comput. & Math. Appl., *62*(3) (2011), 1531–1539.
[41] H. Zarei, A. V. Kamyad, and S. Effati, *Multiobjective optimal control ofHIV dynamics*, Math. Problems Engin., *2010* (2010), Article ID 568315, 29 pages.
[42] H. Zarei, A. V. Kamyad and M. H. Farahi, *Optimal control of HIV dynamic using embedding method*, Computational and Mathematical Methods in Medicine, Hindawi Publishing Corporation, 2011.
[43] H. Zarei, A. Vahidian Kamyad, and A. Akbar Heydari, *Fuzzy modeling and control of HIV infection*, Computational and Mathematical Methods in Medicine, *2012* (2012), Article ID 893474, 17 pages.
[44] C. Zeng and Q. Yang, *A fractional order HIV internal viral dynamics model*, CMES, *59*(1) (2010), 65–77.