STATISTICS, OPTIMIZATION AND INFORMATION COMPUTING
Stat., Optim. Inf. Comput., Vol. **16**, December 2026, pp 5730–5760.
Published online in International Academic Press (www.IAPress.org)

Metaheuristic-Optimized Direct Parameter Control: A Flexible Framework for Chaos Control and Target-Oriented Chaos Shaping

Moh Hasan*, Alfian Futuhul Hadi, Faizal Rifky Fahreza

Department of Mathematics, University of Jember, Jember, 68121, Indonesia

Abstract Many dynamical systems exhibit chaotic behavior, making it difficult to steer their dynamics toward predefined dynamical behaviors, particularly when system-specific analytical analysis becomes complex. Metaheuristic-Optimized Direct Parameter Control (MODPC) is proposed as a flexible framework that uses metaheuristic optimization to determine control variables that modify a prescribed system parameter through a transformation function. The framework separates target selection from control operation: global dynamical information obtained from a bifurcation diagram is used in the offline design stage to identify a desired dynamical behavior, while metaheuristic optimization is performed in the online control stage to drive the system toward the corresponding target trajectory. This formulation enables different algebraic and nonlinear transformation functions, including multidimensional control variables, to be incorporated into the control process. Numerical experiments on discrete and continuous dynamical systems demonstrate the feasibility of the proposed framework under the tested conditions, including the steering of chaotic dynamics toward stable fixed points and periodic regimes. The theoretical analysis clarifies that target reachability depends on the reachable set induced by the transformation function and the prescribed control domain. A Mackey–Glass case study further illustrates the ability of the framework to steer chaotic dynamics toward a selected period-3 regime. These results indicate that MODPC provides a flexible approach for chaos control and targeted dynamical behavior, while its convergence and general applicability remain dependent on the transformation function, search domain, and system conditions.

Keywords Bifurcation Diagrams, Chaotic Systems, Direct Parameter Control, Chaos Shaping

AMS 2010 subject classifications 37D45, 37C75, 90C59, 90C26

DOI: 10.19139/soic-2310-5070-4761

1. Introduction

Dynamical systems sometimes demonstrate chaotic behavior, which can be characterized by extreme sensitivity to the initial conditions and is commonly referred to as the butterfly effect [1]. This means that even a small perturbation in the initial values can lead to an exponential divergence in the system's trajectory, which can make its behavior appear random and incredibly difficult to predict over the long-term [2]. However, chaos is not mere randomness and is instead deterministic behavior tied to a chaotic attractor. This chaotic behavior is not only of theoretical interest but is also relevant across various scientific fields. Examples of chaotic systems can be found in ecology (e.g., predator–prey population dynamics with disease [3]), biology (e.g., irregular heartbeat rhythms [4]), and epidemiology (e.g., the spread of diseases in systems with reinfection [5]). Given the unpredictable nature of chaotic systems, strategies for chaos control have become a crucial aspect of various practical applications. Chaos control aims to change a system from a chaotic state to a desired and stable state (e.g., a fixed point or periodic orbit) by using relatively small interventions or perturbations to the system's parameters or variables [6]. One of the pioneering and most influential strategies is the Ott–Grebogi–Yorke (OGY) method [6, 7], which

*Correspondence to: Moh Hasan (Email: hasan.fmipa@unej.ac.id). Department of Mathematics, University of Jember, Jember, 68121, Indonesia.

ISSN 2310-5070 (online) ISSN 2311-004X (print)
Copyright © 2026 International Academic Press

works by leveraging the presence of unstable fixed points or periodic orbits hidden within the chaotic attractor. One of these periodic orbits is selected as a target, and small and adaptive perturbations are applied to the system's parameters whenever the system's trajectory crosses a control window near the target orbit. This approach has been successfully applied to various systems such as the Hassel-type [8] and Ricker population models [9] to stabilize populations to a fixed point or a period-2 cycle as well as to discrete predator–prey systems [10].

However, applying the OGY method to high-order dynamical systems presents considerable difficulties. Its success is highly dependent on the system's linearization around the target point, which involves constructing and evaluating a Jacobian matrix. For an N-order system, the Jacobian matrix is $N \times N$. As N increases, determining the eigenvalues of this high-dimensional matrix becomes computationally complex and prone to numerical inaccuracies. Furthermore, determining appropriate control inputs that ensure all eigenvalue moduli are less than one, which is a crucial condition for stability in discrete systems, involves a vast and intricate search space. These difficulties not only limit theoretical analysis but also hinder the practical application of the OGY method to real-world phenomena with high dimensions and complex dynamics, such as in neurology, climate, or economics. Therefore, a more efficient and adaptive approach is needed for chaos control of high-order dynamical systems.

The OGY method is also limited in terms of operational flexibility, particularly when it targets various dynamic scenarios. The OGY method is designed to steer a chaotic system to a specific periodic orbit, which then becomes a stable attractor [7]. While it can successfully stabilize systems to a period-2 orbit (e.g., in Hassel-type and Ricker population models [8, 9]) or even period-4 orbit (e.g., in the logistic map system [7]), this capability does not come without consequences. Whenever there is a change in the control scenario, such as a change in the system's parameter values that shifts the fixed point or a desire to target a different periodic orbit (e.g., from period-1 to period-2 or period-4), the complex computational analysis must be repeated. This involves relinearizing the system around the new target and determining appropriate control inputs to achieve the desired dynamical behavior. This repetitive complexity is a major obstacle to exploring multiple control scenarios or adapting to dynamic changes in external conditions. Thus, a more flexible approach to chaos control is needed that does not require the repetition of complex mathematical analysis to handle changes in the control scenario.

Optimization-based approaches have been applied to nonlinear mathematical models and parameter-selection problems in related fields. For example, Rosa and Torres [11] fitted a periodic SEIRS model to real Human Respiratory Syncytial Virus (HRSV) surveillance data and subsequently formulated an optimal control problem in which treatment was used as the control variable. Related work by Rosa and Torres [12] extended this perspective to a fractional-order tuberculosis model, where sensitivity analysis was used to identify important parameters before formulating a fractional optimal control problem with treatment rate as the control variable. These studies illustrate how model parameters and intervention variables can be linked through optimization-based control formulations in epidemiological systems. Beyond dynamical-system control, Abubakar et al. [13] compared simulated annealing, differential evolution, and genetic algorithms with conventional numerical methods for estimating Weibull distribution parameters. Their results demonstrate the potential of population-based metaheuristics for parameter optimization when conventional numerical approaches provide less favorable solutions. Together, these studies provide related evidence for the use of optimization and metaheuristic methods in parameter-dependent mathematical problems, although their formulations and objectives differ from the chaos-control problem considered in this study.

In the context of chaos control, optimization-based methods have also been investigated as alternatives to conventional analytical approaches. Particle swarm optimization (PSO), for example, has been used to transform chaos-control tasks into optimization problems for tracing unstable periodic orbits and directing chaotic systems toward desired behaviors [14]. Other studies have employed PSO to adjust multiple feedback gains concurrently in minimum-entropy control, allowing several control parameters or control inputs to be optimized simultaneously [15]. PSO has also been applied within optimal-control formulations to track unstable periodic orbits, where the optimization determines control actions that keep the system trajectory close to a prescribed orbit [16]. More recently, population-based metaheuristics such as Teaching-Learning-Based Optimization have been used to formulate the targeting of chaotic systems as constrained multimodal optimization problems and to direct

chaotic trajectories toward desired target regions [17]. Metaheuristic optimization has additionally been used to tune feedback controllers for chaotic systems under parameter uncertainty and measurement noise [18].

These studies establish that metaheuristic optimization can effectively support chaos stabilization, trajectory tracking, and target-oriented control. However, the role of the metaheuristic in these approaches is generally associated with optimizing controller gains, feedback parameters, control actions, or trajectory-tracking variables within a predefined control structure. MODPC adopts a different formulation by explicitly linking an intervention variable to the system parameter responsible for the observed dynamics through a prescribed transformation mechanism. The optimization therefore operates directly on the intervention variable, while the transformation mechanism determines how the intervention modifies the system parameter. This formulation separates the numerical optimization from the mathematical representation of the intervention mechanism and allows the same control framework to accommodate different algebraic and nonlinear parameterizations.

Metaheuristic-Optimized Direct Parameter Control is proposed as a flexible approach for targeted chaos control. In this study, the term *chaos shaping* refers specifically to the process of deliberately steering a chaotic system toward a predefined dynamical behavior rather than merely suppressing chaos. Conventional chaos control and stabilization primarily aim to eliminate chaotic behavior by stabilizing the system around a selected fixed point or periodic orbit. In contrast, chaos shaping emphasizes the selection of the desired dynamical regime itself. The target may therefore be a stable fixed point, a period-2 or period-4 cycle, or another predefined periodic trajectory. Under this definition, stabilization can be regarded as a particular case of chaos shaping in which the desired target is a stable state or orbit.

Target-oriented chaos control itself is not a new concept, as previous optimization-based approaches have demonstrated the use of desired trajectories or target regions to direct chaotic systems [14, 16, 17]. Therefore, the term chaos shaping in this study is not intended to claim a fundamentally new class of chaos-control methods. Instead, it is used to describe the target-oriented capability of MODPC, in which the desired dynamical behavior is selected explicitly through a target parameter and its corresponding trajectory. In this formulation, stabilization to a fixed point or periodic orbit is treated as a special case of chaos shaping, while different stable regimes can be selected according to the desired system behavior.

The proposed MODPC framework separates target selection from the subsequent control operation into offline and online stages. In the offline design stage, a bifurcation diagram is generated to characterize the global dynamical structure of the system and to identify a suitable target regime. The corresponding target trajectory is then generated from the selected parameter value and provided to the objective function. The bifurcation analysis is therefore used only for target selection and is not repeatedly performed during control operation. In the online control stage, the system is driven from its current chaotic state using the selected target trajectory. A metaheuristic optimizer searches for an appropriate control input by minimizing the trajectory discrepancy between the controlled system and the predefined target. The control process terminates when the trajectory error satisfies the prescribed convergence criterion. Thus, the offline stage determines *what dynamical behavior is desired*, whereas the online stage determines *how the system is driven toward that behavior*.

This study evaluates the proposed framework across dynamical systems with different dimensions and characteristics, including one-dimensional maps, a multidimensional discrete map, a continuous chaotic system, and a higher-dimensional epidemic model. The evaluation considers control accuracy, convergence behavior, computational cost, robustness to measurement noise, and the resulting dynamical behavior. In addition, the framework is compared conceptually and numerically with established chaos control approaches to clarify its applicability, advantages, and limitations.

2. Methods

2.1. Procedure

MODPC controls a chaotic system by directly modifying a prescribed system parameter through a transformation function whose control variables are optimized using a metaheuristic algorithm. The overall procedure consists of the following steps:

1. **System initialization and identification of the initial dynamical regime:** The dynamical system is first simulated using a fixed set of model parameters ($Param$), initial conditions (IC_{init}), and an initial value of the parameter to be controlled (P_{init}). The resulting trajectory is used to identify the initial dynamical regime, such as chaotic behavior. The system can generally be represented as

$$X_{\text{init}} = S(Param, IC_{\text{init}}, P_{\text{init}}).$$

For the Logistic map, P_{init} corresponds to the parameter r that determines the dynamical regime, while $Param$ is not required. For the epidemic model, P_{init} corresponds to the infection-rate parameter, while the remaining model parameters are included in $Param$.

2. **Determination of the target dynamics:**
A desired target behavior is then selected based on the objective of the control process. The target may be a stable fixed point, a periodic orbit, or another predefined dynamical trajectory. When global dynamical information is available, such as a bifurcation diagram, a target parameter P_T can be selected from a parameter region associated with the desired behavior. The corresponding target trajectory is generated as

$$X_{\text{Target}} = S(Param, IC_{\text{init}}, P_T).$$

The target trajectory is subsequently used as the reference for the objective function. Thus, the target-selection stage determines the dynamical behavior that the control process is intended to reproduce.

3. **Metaheuristic initialization:** The control parameter K is defined as the optimization variable. It may consist of a single scalar or a vector,

$$K = [k_1, k_2, \ldots, k_n],$$

depending on the selected transformation function and the number of intervention variables. A metaheuristic optimization algorithm is then initialized with a prescribed population size ($PopSize$), maximum number of iterations ($MaxIter$), and search boundaries ($K_{\min}$ and $K_{\max}$). In this study, the Driving Training-Based Optimization (DTBO) algorithm is used as the metaheuristic optimizer [19].

4. **Parameter transformation and system simulation:** During the optimization process, each candidate value of K is used to modify the initial system parameter through a prescribed transformation function,

$$P_{\text{new}} = g(P_{\text{init}}, K).$$

The transformation function g defines how the control variables affect the system parameter and may take a simple algebraic or nonlinear form. The system is then simulated using the transformed parameter to obtain the corresponding trajectory,

$$X_{\text{Sim}} = S(Param, IC_{\text{init}}, P_{\text{new}}).$$

The transformation function is therefore specified by the control designer and is not itself optimized by the metaheuristic.

5. **Objective function evaluation:** Each candidate control parameter is evaluated by comparing the resulting trajectory X_{Sim} with the target trajectory X_{Target}. In this study, the Mean Absolute Percentage Error (MAPE) is used as the objective function:

$$J(K) = \text{MAPE}\left(X_{\text{Target}}, X_{\text{Sim}}(K)\right).$$

The metaheuristic retains candidate solutions that provide lower objective values and iteratively updates the population toward better-performing control parameters.

6. **Termination and selection of the control parameter:**
The optimization process is terminated when the best objective value reaches the prescribed stopping threshold, which is set to MAPE $\leq 5\%$ in this study, or when the maximum number of iterations is reached. The best control parameter identified during the optimization is denoted by K^*, and the corresponding transformed parameter and system trajectory are used as the final controlled solution. A solution satisfying the stopping criterion is considered a feasible control solution for the selected target under the prescribed transformation function and search domain.

2.2. MODPC Algorithm

The procedure described above is summarized in Algorithm 1. The pseudocode follows the implementation used in this study, including target-trajectory generation, parameter transformation, population initialization, objective-function evaluation, and the stopping criterion.

Algorithm 1 Metaheuristic-Optimized Direct Parameter Control (MODPC) Framework

Require: Initial parameter P_{init}, model parameters $Param$, initial condition IC_{init}, target parameter P_T, parameter mapping function $g(\cdot)$, search domain $[K_{\min}, K_{\max}]$, population size N, iteration limit $T_{\max}$, error tolerance $\epsilon = 5\%$.

Ensure: Optimal control factor K^*, stabilized parameter P_{new}, controlled trajectory X^*, minimum objective value J^*.

Generate reference target trajectory:

$$X_{\text{Target}} = S(Param, IC_{\text{init}}, P_T).$$

Define parameter transformation rule:

$$P_{\text{new}}(K) = g(P_{\text{init}}, K).$$

Define objective evaluation function:

$$J(K) = \text{MAPE}\left(X_{\text{Target}},\, S\big(Param, IC_{\text{init}}, P_{\text{new}}(K)\big)\right).$$

Randomly initialize population: $K_i \sim \mathcal{U}(K_{\min}, K_{\max})$ for $i = 1, \ldots, N$.

Evaluate fitness $J(K_i)$ for all individuals $i \in \{1, \ldots, N\}$.

Identify initial global best:

$$K^* = \arg\min_{K_i} J(K_i), \qquad J^* = \min_i J(K_i).$$

for $t = 1$ **to** $T_{\max}$ **do**

  Update candidate population $\{K_i\}_{i=1}^N$ via metaheuristic search operators.

  Enforce bound constraints: $K_i = \max\big(K_{\min}, \min(K_i, K_{\max})\big)$.

  Evaluate objective value $J(K_i)$ for each updated candidate.

  Apply greedy selection: retain K_i if $J(K_i^{\text{new}}) \le J(K_i^{\text{old}})$.

  Update global best solution (K^*, J^*).

  if $J^* \le \epsilon$ **then**

    break ▷ Target convergence threshold satisfied

  end if

end for

Compute final modulated parameter:

$$P_{\text{new}} = g(P_{\text{init}}, K^*).$$

Generate final controlled trajectory:

$$X^* = S(Param, IC_{\text{init}}, P_{\text{new}}).$$

return K^*, P_{new}, X^*, J^*.

Algorithm 1 distinguishes the target-design stage from the subsequent optimization stage. The target trajectory is generated before the optimization begins, using the selected target parameter and the same model structure and initial conditions. This stage constitutes the offline design component of MODPC when the target is selected using global dynamical information such as a bifurcation diagram. The subsequent metaheuristic search constitutes the

online control component, in which candidate values of K are evaluated through the transformation function g and the resulting system trajectory. Once a feasible K^* is obtained, the corresponding transformed parameter is applied to generate the controlled trajectory.

For reproducibility, the complete MATLAB implementation of MODPC, including the optimization algorithm, objective functions, dynamical-system models, parameter settings, initial conditions, and simulation scripts, is provided in the accompanying code repository [20]. The repository also contains the configurations required to reproduce the numerical experiments reported in this study. The pseudocode presented above summarizes the general MODPC procedure, while the repository provides the complete implementation for each dynamical system and experimental scenario.

2.3. Simulations

Numerical simulations were conducted to evaluate the performance and flexibility of MODPC across chaotic systems with different dimensions and dynamical characteristics. Four systems were considered: the Logistic map, Hénon map, Lorenz system, and discrete epidemic model. The simulations examined the ability of MODPC to control chaotic behavior and, where applicable, to shape the system toward predefined target dynamics. The experiments also evaluated the effect of different transformation functions, control-parameter dimensions, and target behaviors. All simulations were performed using MATLAB R2023b on a computer equipped with an Intel Core i7-8700 CPU (6 cores; 12 threads; 3.2 GHz) and 16 GB of RAM. The reported computational times therefore reflect the performance under this specific hardware and software configuration.

3. Results and discussion

3.1. Metaheuristic Independence and Hyperparameter Setup

The proposed MODPC is designed as a universal, algorithm-agnostic framework. It does not rely on one specific metaheuristic algorithm to function. To demonstrate this flexibility, the framework was implemented using three different metaheuristic algorithms: DTBO, PSO, and Differential Evolution (DE). The performance of each algorithm was evaluated across the four chaotic models over 30 independent runs, with a uniform population size of 30 agents.

As shown in Table 1, all three algorithms successfully stabilized the chaotic models and achieved the target MAPE of $\leq 5\%$. This confirms that MODPC can be seamlessly integrated with various optimization algorithms. However, DTBO was selected as the primary algorithm for this study due to its superior convergence efficiency. While PSO and DE successfully found the control parameters, they required significantly more iterations. For example, in the highly complex discrete epidemic model, DTBO required an average of only 18.67 iterations to converge, whereas PSO and DE required 65.10 and 58.43 iterations, respectively. DTBO consistently found the optimal control parameters with the lowest number of iterations across the tested systems.

Following the selection of DTBO, a hyperparameter tuning analysis was conducted to determine its optimal population size (Search Agents, N_{pop}). The algorithm was tested using four different population sizes ($N_{pop} \in \{10, 30, 50, 100\}$) over 30 independent runs under clean conditions. Table 2 summarizes the clear mathematical trade-off between exploration capability and computational cost. Using a small population size ($N_{pop} = 10$) limits the algorithm's ability to explore the search space, resulting in a significantly higher number of iterations to successfully find the control parameter. Conversely, increasing the population size to 50 or 100 agents improves exploration and reduces iterations, but it severely increases the computational workload. This time penalty is highly evident in continuous systems that require discretization. In the Lorenz system, the computational time jumped from 14.67 seconds for 10 agents to 56.46 seconds for 100 agents.

Based on these empirical results, a population size of 30 ($N_{pop} = 30$) was selected. It provides a practical balance between convergence speed and computational cost across the tested systems. Furthermore, this analysis confirms that performing sensitivity testing on the maximum iteration limit ($MaxIter = 1000$) is unnecessary. The MODPC framework employs an objective-based early stopping criterion. Once the search finds a control parameter that results in a MAPE of $\leq 5\%$, the simulation immediately stops. Because the algorithm consistently achieves this

Table 1. Performance comparison of MODPC using different metaheuristic algorithms (30 runs, 30 population size)

Model	Algorithm	MAPE (%)	Iterations (Mean ± SD)	Time (s) (Mean ± SD)
Logistic Map	DTBO	2.57 ± 1.58	10.43 ± 3.24	0.0105 ± 0.0091
	PSO	2.19 ± 1.52	47.33 ± 42.03	0.0113 ± 0.0116
	DE	2.71 ± 1.26	28.30 ± 9.64	0.0173 ± 0.0150
Hénon Map	DTBO	2.46 ± 1.63	8.77 ± 4.02	0.0195 ± 0.0283
	PSO	2.64 ± 1.41	33.97 ± 25.06	0.0123 ± 0.0088
	DE	2.44 ± 1.39	26.90 ± 14.43	0.0193 ± 0.0113
Lorenz System	DTBO	1.86 ± 1.32	9.83 ± 3.77	30.09 ± 11.21
	PSO	2.30 ± 1.42	39.37 ± 30.69	39.41 ± 30.63
	DE	2.39 ± 1.49	15.37 ± 6.69	19.32 ± 8.52
Epidemic Model	DTBO	3.00 ± 1.30	18.67 ± 15.00	1.88 ± 1.37
	PSO	2.62 ± 1.33	65.10 ± 48.15	2.61 ± 1.86
	DE	2.16 ± 1.21	58.43 ± 27.34	2.12 ± 1.15

target well below 50 iterations, the maximum iteration parameter only acts as a theoretical safety limit to prevent infinite loops, and the maximum iteration limit primarily acts as a safety bound in the tested cases. With the optimal hyperparameter empirically established, the following subsections discuss the detailed performance of the MODPC framework across each specific chaotic dynamical system.

Table 2. Hyperparameter sensitivity analysis of population size (Search Agents) for DTBO over 30 independent runs

Model	Search Agents (N_{pop})	MAPE (%)	Iterations (Mean ± SD)	Time (s) (Mean ± SD)
Logistic Map	10	2.26 ± 1.63	19.10 ± 12.21	0.0086 ± 0.0102
	30	2.57 ± 1.58	10.43 ± 3.24	0.0105 ± 0.0091
	50	1.87 ± 1.50	9.33 ± 4.73	0.0117 ± 0.0069
	100	2.21 ± 1.37	6.90 ± 3.41	0.0119 ± 0.0054
Hénon Map	10	1.79 ± 1.59	17.33 ± 11.39	0.0085 ± 0.0090
	30	2.46 ± 1.63	8.77 ± 4.02	0.0195 ± 0.0283
	50	2.07 ± 1.38	9.73 ± 5.48	0.0110 ± 0.0068
	100	2.15 ± 1.48	5.80 ± 3.96	0.0121 ± 0.0074
Lorenz System	10	2.16 ± 1.25	15.80 ± 9.32	14.67 ± 8.45
	30	1.86 ± 1.32	9.83 ± 3.77	30.09 ± 11.21
	50	1.95 ± 1.41	7.03 ± 3.25	33.35 ± 14.70
	100	1.99 ± 1.39	5.83 ± 2.60	56.46 ± 23.71
Epidemic Model	10	1.83 ± 1.45	47.07 ± 38.94	1.26 ± 1.04
	30	3.00 ± 1.30	18.67 ± 15.00	1.88 ± 1.37
	50	2.61 ± 1.39	15.57 ± 11.67	2.08 ± 1.51
	100	2.35 ± 1.20	11.37 ± 6.09	3.04 ± 1.59

3.2. Logistic Map

The logistic map illustrates how a simple mathematical model can generate complex and even chaotic behavior [21], and it is often used to describe population growth dynamics. Mathematically, the logistic map is a first-order system expressed as

$$x_{n+1} = r x_n (1 - x_n), \tag{1}$$

where r is a parameter representing the population growth rate and x_n is the population proportion at time n. Varying the value of r leads to diverse dynamics ranging from stability around a fixed point to the emergence

of chaos when $r > 3.57$. Figure 1 shows a simulation of the periodic behavior of system (1) with a given value of $r = 3.8$ and initial perturbation of 10^{-3}. Two trajectories starting from very similar initial conditions rapidly diverge exponentially, which is characteristic of chaotic behavior: extreme sensitivity to initial values and impossible long-term prediction.

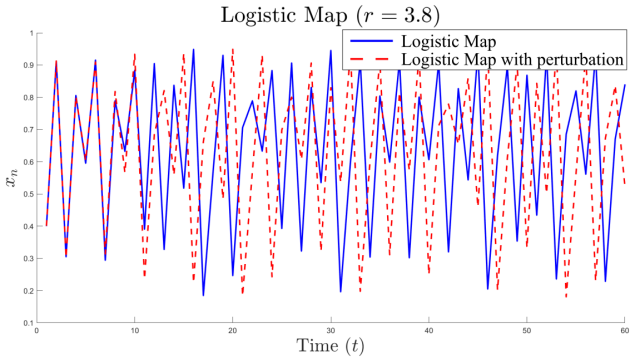


Figure 1. Simulation of system (1) at $r = 3.8$.

Figure 2 shows the results of applying MODPC to the system. For the metaheuristic optimization, the control parameter bounds were set between -100 and 100. The population size was set to 30, and the maximum number of iterations was 1,000. The target stopping condition was set to a MAPE of $\leq 5\%$. As shown in Figure 2a, the system was initially chaotic at $r = 3.8$ but was successfully controlled into a single stable fixed point. To ensure statistical reliability, the MODPC algorithm was executed for 30 independent runs. The statistical results demonstrate that MODPC is highly efficient. It required an average of only 11.77 ± 6.30 iterations to successfully reach the target, achieving a final average MAPE of $1.87 \pm 1.64\%$. The average computational time was remarkably fast at 0.0077 ± 0.0061 s. This rapid performance is highly beneficial for controlling dynamic systems in real-time scenarios.

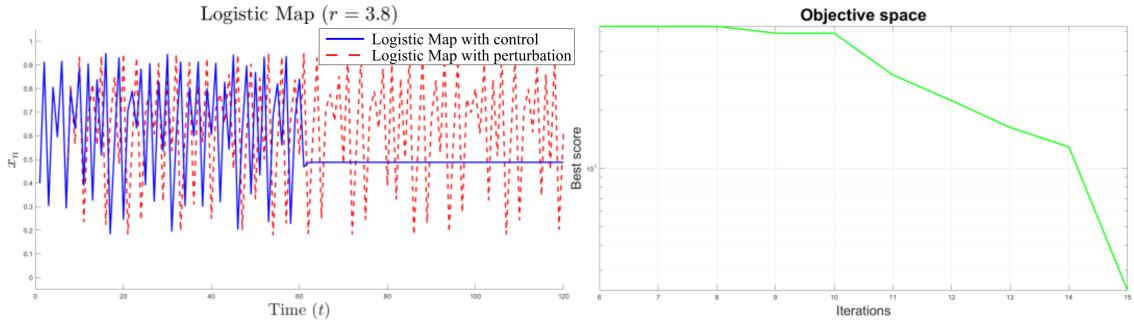


Figure 2. Application of MODPC to system (1) at $r = 3.8$: (a) simulation results and (b) MAPE in each iteration.

Table 3 summarizes the statistical results of applying MODPC across various r values over 30 independent runs. These scenarios cover different dynamic targets, including period-2, period-4, and chaotic behaviors. In every scenario, MODPC consistently guided the system toward the desired target with an average MAPE well below the 5% threshold and minimal computational time (averaging around 0.01 seconds per run). This consistency and speed were achieved without the need for relinearization analysis or eigenvalue calculations for every change in r. These results indicate that MODPC can adaptively control the system across its entire dynamic spectrum with very low computational overhead, directly solving one of the major limitations of conventional chaos control methods.

Furthermore, to objectively verify the system's behavioral transition, the Largest Lyapunov Exponent (LLE) was estimated using a data-driven time-series analysis based on the Rosenstein algorithm [27]. To ensure methodological transparency, the simulation was evaluated over 6,000 data points, with the control mechanism

Table 3. Statistical simulation results (Mean ± Standard Deviation) from applying MODPC to system (1) over 30 independent runs.

r	Behavior	MAPE (%)	Iterations	Time (s)
3.3	Period-2 cycle	1.88 ± 1.55	11.07 ± 6.36	0.0129 ± 0.0109
3.48	Period-4 cycle	2.59 ± 1.41	12.43 ± 6.74	0.0139 ± 0.0158
3.68	Chaotic	2.27 ± 1.51	11.73 ± 5.88	0.0154 ± 0.0149
3.7	Chaotic	2.32 ± 1.62	10.60 ± 5.83	0.0117 ± 0.0146
3.8	Chaotic (Fixed Target)	1.87 ± 1.64	11.77 ± 6.30	0.0077 ± 0.0061
4.0	Chaotic	2.57 ± 1.58	10.43 ± 3.24	0.0105 ± 0.0091

activated at iteration 3,001. The pre-control LLE was extracted using the first 3,000 unforced chaotic data points. For the post-control evaluation, data points from iteration 3,501 to 6,000 were utilized, deliberately discarding the initial 500 points immediately following the control activation to eliminate the transient phase. The phase-space reconstruction was conducted utilizing the fundamental parameters: a time delay of $\tau = 1$, an embedding dimension of $m = 3$, and a Theiler window of 10 to strictly exclude temporally correlated nearest neighbors.

Figure 3 illustrates the average logarithmic divergence of nearest neighbors over the evolution time steps, where the slope of the linear fit determines the LLE. The pre-control trajectory exhibits a clear linear scaling region, yielding a positive LLE of approximately 0.676. This quantitatively confirms the presence of chaotic divergence and strongly agrees with the theoretical behavior of the unforced map. Conversely, the identical time-series analysis of the post-control trajectory shows no observable exponential divergence, with the fitted slope approaching zero. Because the controlled trajectory converged to a stable fixed point, the nearest-neighbor divergence rapidly reached the numerical precision limit, preventing the identification of a meaningful linear divergence region. This result provides strong evidence that the proposed MODPC framework successfully eliminated the observable chaotic behavior from the controlled trajectory.

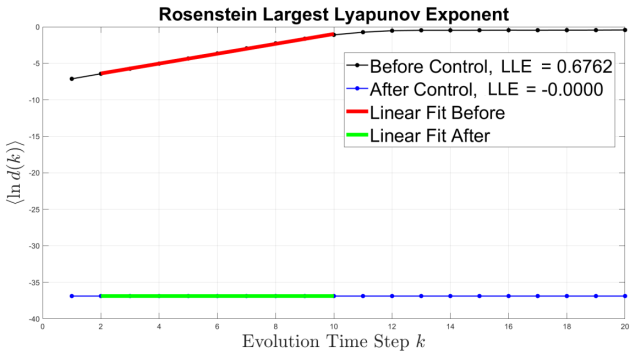


Figure 3. Data-driven estimation of the LLE using the Rosenstein algorithm for system (1) before and after the application of MODPC.

3.3. Hénon Map

The Hénon map exhibits chaos in two dimensions and is often considered a simplification of the Lorenz attractor as it effectively captures the stretch-and-fold dynamics characteristic of chaotic systems [22]. Mathematically, the Hénon map is a second-order system expressed as

$$\begin{aligned} x_{n+1} &= 1 - ax_n^2 + y_n, \\ y_{n+1} &= bx_n, \end{aligned} \tag{2}$$

where a and b are parameters. Chaotic behavior occurs at $a = 1.4$ and $b = 0.3$; this is known as the Hénon attractor, which has fractal characteristics and demonstrates extreme sensitivity to the initial values. Figure 4 shows a

simulation of system (2) demonstrating how two trajectories starting from very similar initial conditions rapidly diverge exponentially.

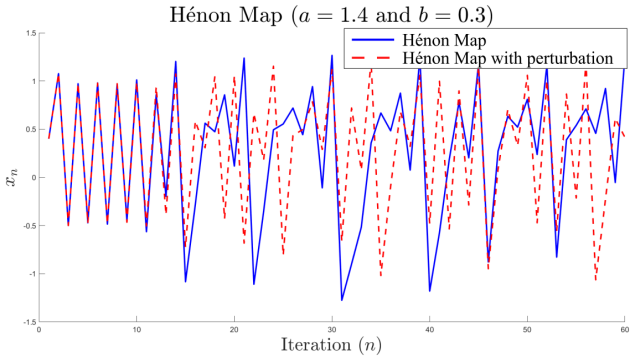


Figure 4. Simulation of system (2) at $a = 1.4$ and $b = 0.3$.

Figure 5 shows the results of applying MODPC to control the chaotic behavior of the system. The metaheuristic optimization used the same settings as the first-order system: the control parameter bounds were set between -100 and 100, the population size was 30, and the maximum number of iterations was 1,000. For the scenario where $a = 1.4$ and $b = 0.3$, the parameter being modified by the controller was a. As shown in Figure 5a, MODPC successfully controlled the system into a stable fixed point. To ensure the reliability of these results, the algorithm was evaluated over 30 independent runs with a target MAPE of $\leq 5\%$. The statistical data shows that MODPC achieved convergence in an average of only 8.77 ± 4.02 iterations, resulting in a final average MAPE of $2.46 \pm 1.63\%$. The required computational time was very fast, averaging 0.0195 ± 0.0283 s. These results confirm that MODPC effectively and efficiently controls both first- and second-order chaotic systems without requiring any fundamental changes to its algorithmic structure.

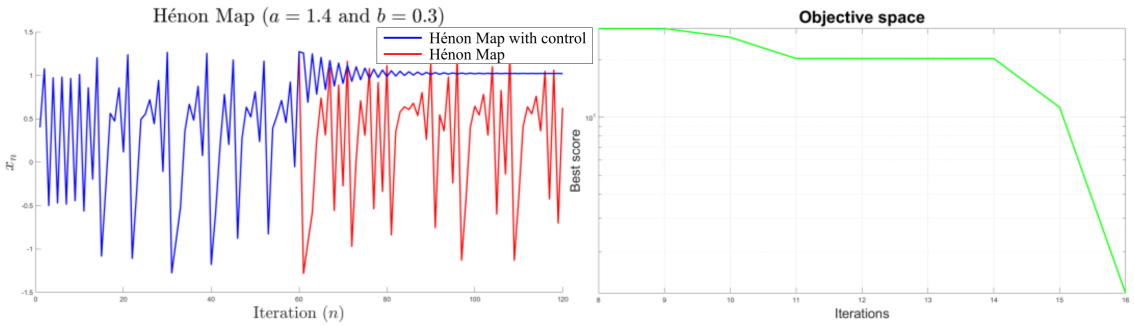


Figure 5. Application of MODPC to system (2) at $a = 1.4$ and $b = 0.3$: (a) simulation results and (b) MAPE in each iteration.

Table 4 summarizes the statistical results of applying MODPC across various values of a over 30 independent runs. In every dynamic scenario, including period 2, period 4, and complex higher period cycles, MODPC successfully guided the system toward a single fixed point. The average MAPE consistently remained below the 5% threshold, and the computational time was minimal across all tested behaviors. This high level of efficiency was achieved without the need for in-depth analytical modifications, Jacobian matrices, or recalculations for each change in the parameter a.

The same data-driven LLE analysis was applied to the Hénon system using the x-coordinate time series. Based on 3,000 unforced data points, the pre-control trajectory yielded an LLE of 0.4244 with an R^2 of 0.999936, confirming the exponential divergence associated with its chaotic dynamics. After MODPC control, the first 500 points following control activation were discarded, and the subsequent steady-state trajectory was analyzed. The resulting divergence curve reached the numerical precision limit and exhibited no meaningful linear growth, with

Table 4. Statistical simulation results (Mean $\pm$ Standard Deviation) from applying MODPC to system (2) over 30 independent runs.

a	Behavior	MAPE (%)	Iterations	Time (s)
0.73	Period-2 cycle	2.30 ± 1.62	9.53 ± 4.95	0.0122 ± 0.0111
0.96	Period-4 cycle	2.19 ± 1.69	9.77 ± 6.04	0.0139 ± 0.0191
1.04	Period-8 cycle	2.39 ± 1.36	10.83 ± 5.59	0.0133 ± 0.0116
1.18	Chaotic	1.88 ± 1.54	11.20 ± 5.82	0.0136 ± 0.0123
1.23	Period-6 cycle	1.80 ± 1.26	10.83 ± 8.26	0.0138 ± 0.0133
1.40	Chaotic (Fixed Target)	2.46 ± 1.63	8.77 ± 4.02	0.0195 ± 0.0283

the estimated slope approaching zero. This indicates that observable exponential divergence was suppressed after control, providing additional evidence that MODPC successfully stabilized the Hénon system.

3.4. Lorenz System

The Lorenz system was a pioneer in chaos theory and provided the fundamental explanation of extreme sensitivity to initial values, which is widely known as the butterfly effect. The Lorenz system describes atmospheric convection or the movement of a fluid heated from below and cooled from above, which is similar to Earth's atmosphere [23]. Mathematically, the Lorenz system is a third-order system expressed as

$$\begin{aligned} \frac{dx}{dt} &= \sigma(y - x), \\ \frac{dy}{dt} &= x(\rho - z) - y, \\ \frac{dz}{dt} &= xy - \beta z, \end{aligned} \tag{3}$$

where σ, ρ, and β are positive system parameters. System (3) exhibits chaotic behavior when these parameters are set to their standard values of $\sigma = 10$, $\rho = 28$, and $\beta = \frac{8}{3}$. Figure 6a shows a phase portrait of system (3), and Figure 6b clearly illustrates how sensitivity to initial values causes an exponential divergence in the system's trajectories with very small differences.

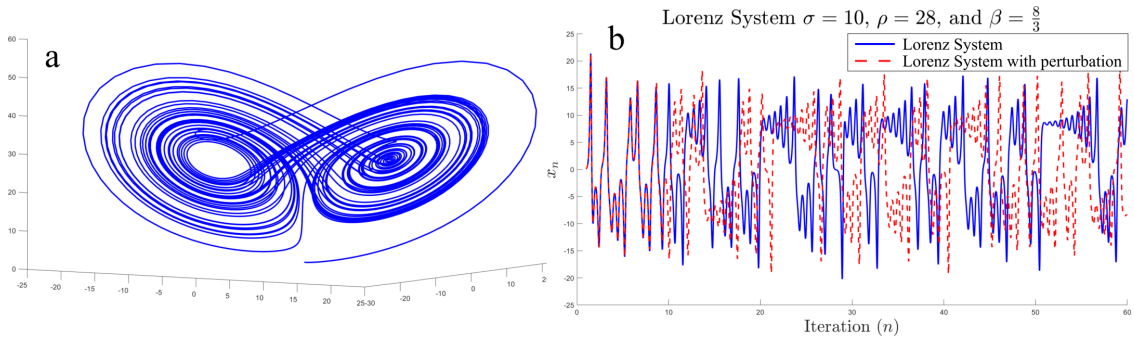


Figure 6. System (3): (a) phase portrait and (b) simulation at $\sigma = 10$, $\rho = 28$, and $\beta = \frac{8}{3}$.

Figure 7 shows the results of applying MODPC to control the chaotic behavior of the system. The metaheuristic optimization was conducted using the standard settings: the control parameter bounds were set between -100 and 100, the population size was 30, and the maximum number of iterations was 1,000. In this scenario, the control parameter being modified was ρ. As shown in Figure 7a, the system was successfully controlled into a stable fixed point. To ensure statistical reliability, the algorithm was evaluated over 30 independent runs with a target MAPE of $\leq 5\%$. The statistical results show that the target was achieved in an average of only 9.83 ± 3.77

iterations, with a final average MAPE of $1.86 \pm 1.32\%$. The average computational time was 30.09 ± 11.21 s. This computational time is longer compared to the previous two discrete systems. The increase in time occurs because the continuous system must first be discretized using the Euler method with a time step of $\Delta t = 0.01$. Consequently, each evaluation within the MODPC framework requires numerous time-point computations, which substantially increases the computational load.

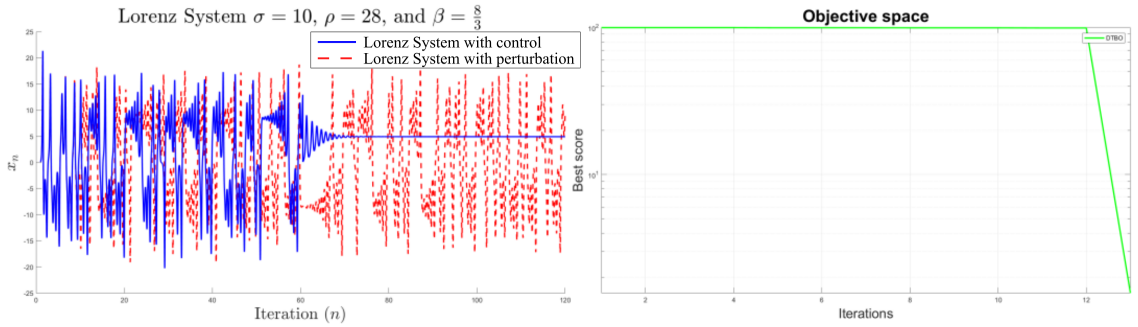


Figure 7. Application of MODPC to system (3) at $\sigma = 10$, $\rho = 28$, and $\beta = \frac{8}{3}$: (a) simulation results and (b) MAPE in each iteration.

Table 5 summarizes the statistical results of applying MODPC across six different initial chaotic ρ values over 30 independent runs. In all scenarios, MODPC successfully transitioned the systems from chaotic behavior to the desired stable fixed point. Although the computational time averaged around 20 to 30 seconds due to the discretization process of the continuous system, this performance remains highly feasible and efficient for controlling complex dynamics. More importantly, these results demonstrate that MODPC can seamlessly handle more complex, higher-order continuous systems without requiring intricate analytical procedures, such as Jacobian matrix derivations or eigenvalue evaluations for every new scenario. This highlights a fundamental advantage of MODPC over conventional chaos control methods.

Table 5. Statistical simulation results (Mean $\pm$ Standard Deviation) from applying MODPC to system (3) over 30 independent runs.

ρ	Behavior	MAPE (%)	Iterations	Time (s)
15.9	Chaotic	1.86 ± 1.42	6.73 ± 2.45	20.48 ± 6.93
18.57	Chaotic	2.06 ± 1.50	7.67 ± 3.49	23.94 ± 10.39
20.45	Chaotic	2.47 ± 1.56	7.40 ± 3.41	22.92 ± 10.77
24.2	Chaotic	1.82 ± 1.42	8.70 ± 4.60	27.87 ± 13.94
28.0	Chaotic	1.86 ± 1.32	9.83 ± 3.77	30.09 ± 11.21
29.8	Chaotic	2.51 ± 1.60	9.17 ± 3.66	27.75 ± 11.05

Regarding the continuous nature of system (3), the choice of the discretization time-step size (Δt) significantly dictates the total computational time. Empirical experiments revealed that the time cost originates entirely from the numerical integration process, rather than the metaheuristic optimization loop. For instance, decreasing the time-step to $\Delta t = 0.005$ and further to $\Delta t = 0.001$ slightly refines the final control parameter K (stabilizing around the $0.34 - 0.36$ range) and improves the final MAPE to 1.04% and 0.79%, respectively. The algorithm's convergence capability remains highly robust, still requiring fewer than 10 iterations in both cases. However, this finer discretization forces the numerical integrator to evaluate a substantially larger number of trajectory points per iteration. Consequently, the computational time inflates exponentially from an average of 37.04 seconds (at $\Delta t = 0.01$) to 88.59 seconds (at $\Delta t = 0.005$), and drastically to 2,370.50 seconds, or nearly 40 minutes (at $\Delta t = 0.001$). Therefore, $\Delta t = 0.01$ was established as the strict optimal threshold, providing the best trade-off between control parameter accuracy and feasible computational efficiency.

To further evaluate the suppression of chaotic dynamics in a continuous-time system, the LLE of the Lorenz system was estimated directly from the simulated time series using the data-driven Rosenstein algorithm [27]. The Rosenstein analysis was performed using a time delay of $\tau = 10$, an embedding dimension of $m = 3$, a Theiler window of 50, and a maximum evolution time of $k = 150$. These parameters were selected to provide sufficient phase-space reconstruction while excluding temporally correlated neighboring points from the nearest-neighbor search.

The Lorenz system was numerically integrated using a time step of $\Delta t = 0.01$. Consequently, a simulation covering the considered time interval generated approximately 600,000 data points. Directly processing the entire time series would substantially increase the computational cost of the nearest-neighbor search. Therefore, the time series was downsampled by retaining every 10th data point. This procedure reduced the number of points used in the Rosenstein analysis while preserving the relevant dynamical structure of the trajectory. Because of this downsampling, the effective sampling interval became $\Delta t_{\text{eff}} = 10\Delta t = 10(0.01) = 0.1$. Thus, one evolution step in the downsampled time series corresponds to 0.1 time units in the original Lorenz system.

For the pre-control analysis, the data range from iterations 1 to 31,000 was selected with an increment of 10, corresponding to the unforced chaotic regime. For the post-control analysis, the data range from iterations 350,000 to 380,000 was selected using the same increment. This latter range was chosen sufficiently far from the control activation point to ensure that the transient response had disappeared and that the analysis represented the steady-state controlled behavior. After constructing the logarithmic divergence curve, the linear scaling region was determined from the observed behavior of the curve. The evolution steps $k = 15$ to $k = 30$ were selected because this interval exhibited the most consistent linear growth before the onset of saturation.

The linear regression over this scaling region produced a slope of approximately 0.0611 for the uncontrolled Lorenz trajectory. Since the regression was performed with respect to the evolution steps of the downsampled time series, the slope was normalized using the effective sampling interval of $\Delta t_{\text{eff}} = 0.1$. Therefore, the LLE was calculated as

$$\lambda_{\max} = \frac{\text{slope}}{\Delta t_{\text{eff}}} = \frac{0.0611}{0.1} \approx 0.611.$$

The resulting positive LLE confirms the exponential divergence of nearby trajectories and provides quantitative evidence of the chaotic behavior of the uncontrolled Lorenz system.

After the application of MODPC, the same Rosenstein analysis was applied to the post-control steady-state trajectory. The logarithmic divergence exhibited no observable growth and rapidly approached the numerical precision limit, resulting in an estimated slope approaching zero. Consequently, the post-control LLE was approximately zero. This result indicates that the controlled trajectory no longer exhibits measurable exponential divergence. Therefore, the LLE analysis provides additional empirical evidence that MODPC successfully suppresses the chaotic behavior of the Lorenz system and stabilizes its trajectory around the desired fixed point.

3.5. Discrete Epidemic Model

The discrete epidemic model describes the dynamics of disease spread while accounting for vaccination and quarantine interventions as well as limitations in medical resources [24]. Mathematically, the discrete epidemic model is a seventh-order system expressed as

$$\begin{aligned}
S_{t+1} &= S_t + \Lambda - \lambda S_t I_t - \lambda q S_t Q_t - \varphi_1 S_t + \rho R_t - \mu S_t,\\
V_{1_{t+1}} &= V_{1_t} + \varphi_1 S_t - \lambda p V_{1_t} I_t - \lambda pq V_{1_t} Q_t - \varphi_2 V_{1_t} - \mu V_{1_t},\\
V_{2_{t+1}} &= V_{2_t} + \varphi_2 V_{1_t} - \lambda r V_{2_t} I_t - \lambda rq V_{2_t} Q_t - \mu V_{2_t},\\
I_{t+1} &= I_t + \lambda S_t I_t + \lambda p V_{1_t} I_t + \lambda r V_{2_t} I_t + \lambda q S_t Q_t + \lambda pq V_{1_t} Q_t + \lambda rq V_{2_t} Q_t\\
&\quad - \tau_1 I_t + \tau_2 Q_t - \mu I_t - \zeta I_t - f(I_t),\\
Q_{t+1} &= Q_t + \tau_1 I_t - \tau_2 Q_t - \gamma_n Q_t - \zeta d Q_t - \mu Q_t,\\
R_{t+1} &= R_t + \gamma_n Q_t - \rho R_t - \mu R_t + f(I_t),\\
D_{t+1} &= D_t + \zeta I_t + \zeta d Q_t,
\end{aligned} \tag{4}$$

where,

$$f(I_t) = \begin{cases} (\gamma_n + \gamma_m) I_t & I_t \leq I_b \\ (\gamma_n + \gamma_m) I_b + \gamma_n (I_t - I_b) & I_t > I_b. \end{cases}$$

The function $f(I_t)$ represents the recovery rate, which depends on the availability of medical resources. $f(I_t)$ decreases when the number of infected cases (I_t) exceeds the capacity of medical resources (I_b). The infection rate (λ) can trigger chaotic behavior as its value increases, which reflects real-world observations: as the infection rate of a disease rises, its spread becomes very difficult to control and unpredictable. Figure 8 shows a simulation of system (4) demonstrating how two trajectories starting from initial points very close together have considerably different paths, which is characteristic of chaotic behavior.

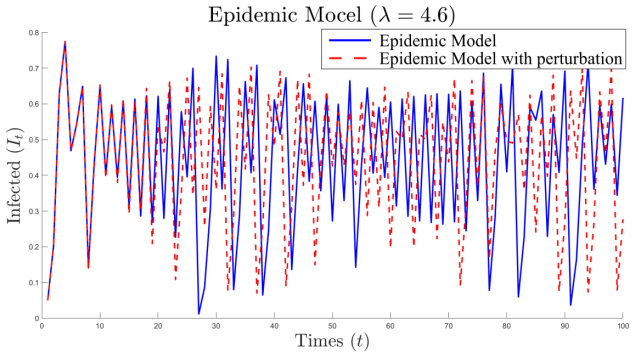


Figure 8. Simulation of system (4) at $\lambda = 4.6$.

Figure 9 shows the results of applying MODPC to control the system. The objective was to stabilize the population dynamics into a single fixed point, which represents a stable endemic or disease-free condition. The metaheuristic optimization used the standard settings: the control parameter bounds were set between -100 and 100, the population size was 30, and the maximum number of iterations was 1,000. The parameter modified by the controller was the infection rate (λ). As shown in Figure 9a, the system was successfully controlled into a stable fixed point. To ensure statistical reliability, the algorithm was evaluated over 30 independent runs with a target MAPE of $\leq 5\%$. The statistical results show that MODPC reached the target in an average of 18.67 ± 15.00 iterations with a final average MAPE of $3.00 \pm 1.30\%$. The average computational time was remarkably fast at 1.88 ± 1.37 s. This indicates that the method can maintain moderate computational cost even for the seventh-order nonlinear system.

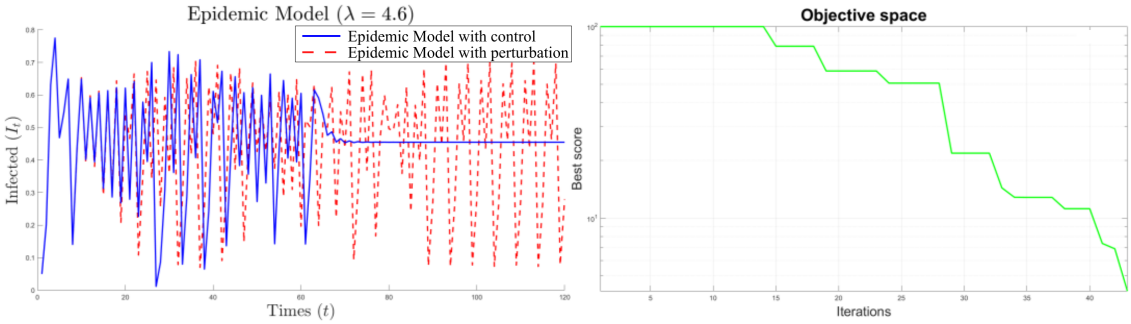


Figure 9. Application of MODPC to system (4) at $\lambda = 4.6$: (a) simulation results and (b) MAPE in each iteration.

Table 6 summarizes the statistical results of applying MODPC to the system across various λ values over 30 independent runs. In every scenario, ranging from periodic cycles to chaotic dynamics, MODPC successfully stabilized the system. The computational time remained consistently low, averaging between 1.6 and 2.1 seconds

per run. These results indicate that MODPC can maintain low computational cost when applied to the high-dimensional discrete system. In addition, MODPC does not require explicit evaluation of the 7×7 Jacobian or its eigenvalues during the control process.

Table 6. Statistical simulation results (Mean $\pm$ Standard Deviation) from applying MODPC to system (4) over 30 independent runs.

λ	Behavior	MAPE (%)	Iterations	Time (s)
3.9	Period-2 cycle	2.34 ± 1.35	18.93 ± 10.15	1.84 ± 1.06
4.153	Period-4 cycle	2.44 ± 1.61	22.97 ± 17.16	2.09 ± 1.54
4.2852	Period-8 cycle	2.50 ± 1.40	17.73 ± 13.71	1.73 ± 1.33
4.4	Chaotic	2.69 ± 1.64	20.47 ± 13.43	1.63 ± 1.08
4.6	Chaotic	3.00 ± 1.30	18.67 ± 15.00	1.88 ± 1.37

To further evaluate the suppression of chaotic dynamics in the higher-dimensional setting, the LLE of the 7-D epidemic model was estimated directly from the simulated time series using the data-driven Rosenstein algorithm [27]. The analysis was performed using a time delay of $\tau = 1$, an embedding dimension of $m = 8$, a Theiler window of 10, and a maximum evolution time of $k = 100$. Since the epidemic model is formulated in discrete time, no additional time-step normalization was required, and each evolution step directly corresponds to one iteration of the model.

The simulation consisted of 6,000 data points, with the first 3,000 points representing the uncontrolled regime and the remaining points corresponding to the controlled regime. For the pre-control analysis, data points from iterations 1 to 3,000 were used. For the post-control analysis, the first 500 points following control activation, corresponding to iterations 3,001 to 3,500, were discarded to eliminate the transient response. The steady-state controlled trajectory from iterations 3,501 to 6,000 was then used for the LLE estimation.

For the uncontrolled trajectory, the logarithmic divergence curve exhibited a clear linear scaling region, indicating exponential separation of initially nearby trajectories. Linear regression over the selected scaling region yielded a high coefficient of determination of approximately $R^2 = 0.999$ and a positive LLE of approximately 0.279. The positive exponent confirms the presence of exponential divergence and provides quantitative evidence of chaotic behavior in the uncontrolled epidemic dynamics.

Following the application of MODPC, the same Rosenstein analysis was applied to the post-control steady-state trajectory. The controlled trajectory exhibited no observable growth in logarithmic divergence, with the estimated slope approaching zero. Consequently, the post-control LLE was approximately zero. This result indicates that the controlled trajectory no longer exhibits measurable exponential divergence and provides further empirical evidence that MODPC successfully suppresses the chaotic behavior of the 7-D epidemic model and stabilizes the system around the desired equilibrium.

3.6. Robustness Against Measurement Noise

In practical real-world applications, observed data and sensor measurements are rarely perfect. They usually contain measurement noise caused by environmental disturbances or instrument inaccuracies. Therefore, an effective chaos control method must be robust against these uncertainties. To evaluate this, a zero-mean Gaussian distribution noise was injected into the state variables of the four models during the simulation. Following the noise modeling approach by [28], the noisy simulated state variables, X_t^*, fed into the objective function are modeled as follows:

$$X_t^* = X_t(1 + n_t), \quad \text{with } n_t \sim \mathcal{N}(0, \sigma^2)$$

where X_t is the clean simulated state at iteration t, and n_t is a realization of a zero-mean Gaussian distribution $\mathcal{N}(0, \sigma^2)$ where σ^2 denotes the noise distribution variance. In this study, the noise level was set to 5%, corresponding to a standard deviation of $\sigma = 0.05$. The systems were tested at their most chaotic parameter values: $r = 4$ for the Logistic map, $a = 1.4$ for the Hénon map, $\rho = 28$ for the Lorenz system, and $\lambda = 4.6$ for the discrete epidemic model. The MODPC framework was executed for 30 independent runs for each system to ensure statistical validity.

Table 7 compares the performance of MODPC under ideal (clean) conditions and noisy conditions. The results show that MODPC successfully maintained stability and achieved the target MAPE of $\leq 5\%$ across all models, from the first-order discrete map to the seventh-order discrete epidemic system.

Table 7. Comparison of MODPC statistical performance (Mean $\pm$ Standard Deviation) under clean and 5% noise conditions over 30 independent runs.

Model	Chaotic Param.	Condition	MAPE (%)	Iterations	Time (s)
Logistic Map (1D)	$r = 4.0$	Clean	2.57 ± 1.58	10.43 ± 3.24	0.0105 ± 0.0091
		Noisy (5%)	4.22 ± 0.43	14.80 ± 6.33	0.0185 ± 0.0086
Hénon Map (2D)	$a = 1.4$	Clean	2.46 ± 1.63	8.77 ± 4.02	0.0195 ± 0.0283
		Noisy (5%)	4.13 ± 0.39	13.37 ± 7.37	0.0187 ± 0.0111
Lorenz System (3D)	$\rho = 28$	Clean	1.86 ± 1.32	9.83 ± 3.77	30.09 ± 11.21
		Noisy (5%)	4.21 ± 0.25	10.33 ± 4.36	39.23 ± 16.14
Epidemic Model (7D)	$\lambda = 4.6$	Clean	3.00 ± 1.30	18.67 ± 15.00	1.88 ± 1.37
		Noisy (5%)	4.31 ± 0.34	17.03 ± 15.44	1.42 ± 1.24

When measurement noise is introduced, there are two main impacts on the algorithm's performance. First, the final MAPE naturally increases to approximately 4.1% to 4.3%. This is an expected mathematical consequence; the 5% injected noise creates a minimum error boundary (noise floor) that prevents the evaluation function from reaching near-zero values. However, the algorithm effectively handles this and keeps the error below the 5% threshold, with a very low standard deviation. Second, the noise makes the objective function landscape slightly irregular. As a result, the metaheuristic algorithm generally requires a slightly higher number of iterations and computational time to find and confirm the optimal control parameter. Despite this minimal increase in computational effort, no failed runs were observed across the 30 independent trials for each model. These results indicate that MODPC maintains its control performance under the tested 5% measurement noise.

3.7. Comparison with Established Chaos Control Methods

To further assess the proposed MODPC framework, a comparison was conducted with two classical chaos control approaches, namely the OGY (Ott-Grebogi-Yorke) method and the Pyragas delayed-feedback control (DFC) method. The comparison focuses on the controller design requirements when stabilizing the chaotic Logistic map (1) at $r = 3.8$ toward its non-trivial fixed point, $x^* = 1 - 1/r \approx 0.7368$.

3.7.1. OGY Method Analysis The OGY method stabilizes an Unstable Periodic Orbit (UPO) by applying small time-dependent perturbations to an accessible system parameter. The method exploits the local dynamics around the selected UPO rather than modifying the global structure of the chaotic attractor [6]. For the Logistic map in (1), the non-trivial fixed point at $r = 3.8$ is

$$x^* = 1 - \frac{1}{r} \approx 0.7368.$$

Let $\delta x_n = x_n - x^*$ denote the deviation from the target fixed point and let $\delta r_n = r_n - r$ denote a small perturbation of the control parameter. Linearizing the map around (x^*, r) gives

$$\delta x_{n+1} \approx \lambda_0 \delta x_n + g \delta r_n,$$

where

$$\lambda_0 = \left.\frac{\partial f}{\partial x}\right|_{x^*,r} = 2 - r$$

and

$$g = \left.\frac{\partial f}{\partial r}\right|_{x^*,r} = x^*(1 - x^*).$$

For $r = 3.8$, these quantities are $\lambda_0 = -1.8$ and $g \approx 0.193906$. Therefore, the uncontrolled fixed point is unstable because $|\lambda_0| > 1$.

To obtain locally stable closed-loop dynamics, a desired multiplier λ_c with $|\lambda_c| < 1$ can be specified. The required parameter perturbation is then obtained from

$$\delta r_n = \frac{\lambda_c - \lambda_0}{g} \delta x_n.$$

For example, choosing $\lambda_c = 0$ gives

$$\delta r_n \approx 9.283(x_n - x^*).$$

The perturbed parameter is therefore applied as $r_n = r + \delta r_n$ when the trajectory enters the prescribed local neighborhood of the target fixed point. Outside this neighborhood, the nominal parameter $r = 3.8$ is retained. This illustrates the local controller-design procedure used in the analytical OGY formulation: the target orbit, local derivative, parameter sensitivity, desired local multiplier, and activation neighborhood must first be specified before the control perturbation can be applied.

3.7.2. Pyragas Method (DFC) Analysis The Pyragas delayed-feedback control (DFC) method stabilizes an UPO by feeding back the difference between the current state and a delayed state. For the period-1 fixed point of the Logistic map, the controlled system can be written as

$$x_{n+1} = r x_n(1 - x_n) + K(x_{n-1} - x_n),$$

where K denotes the feedback gain. The control is non-invasive at the target fixed point because $x_{n-1} = x_n = x^*$, causing the delayed-feedback term to vanish.

To determine a stabilizing gain without relying on numerical trial and error, the local stability of the controlled map is first analyzed. Let $\delta x_n = x_n - x^*$. Linearization around the fixed point gives

$$\delta x_{n+1} = (\lambda_0 - K)\delta x_n + K\delta x_{n-1},$$

where $\lambda_0 = 2 - r$ is the multiplier of the uncontrolled fixed point. Assuming a perturbation of the form $\delta x_n = \lambda^n$ yields the characteristic equation

$$\lambda^2 - (\lambda_0 - K)\lambda - K = 0.$$

For $r = 3.8$, $\lambda_0 = -1.8$, resulting in

$$\lambda^2 + (1.8 + K)\lambda - K = 0.$$

The Jury stability criterion is then applied to determine the conditions under which both roots lie inside the unit circle. This analysis gives the stabilizing interval

$$-1 < K < -0.4.$$

Therefore, the stabilizing gain is not selected arbitrarily. For the subsequent numerical comparison, a representative value of $K = -0.7$ is selected from within the analytically derived stability interval. The controlled trajectory is then simulated using the same initial condition and simulation settings as those used for the OGY and MODPC comparisons.

3.7.3. Numerical Comparison To complement the analytical comparison, the three control approaches were numerically applied to the Logistic map at $r = 3.8$ under the same initial condition. Each method was evaluated according to its respective control formulation. The OGY and Pyragas DFC methods were configured to stabilize the non-trivial unstable fixed point of the original system, $x^* = 1 - 1/r \approx 0.7368$. In contrast, MODPC used a target trajectory selected from the bifurcation structure of the Logistic map. In this experiment, $r_{\text{target}} = 2$ was selected as the desired dynamical regime, corresponding to the stable fixed point $x_{\text{target}} = 1 - 1/r_{\text{target}} = 0.5$.

Figure 10 presents the resulting trajectories. The uncontrolled trajectory exhibits sustained chaotic fluctuations before control activation. Following activation, all three methods successfully suppress the chaotic behavior and

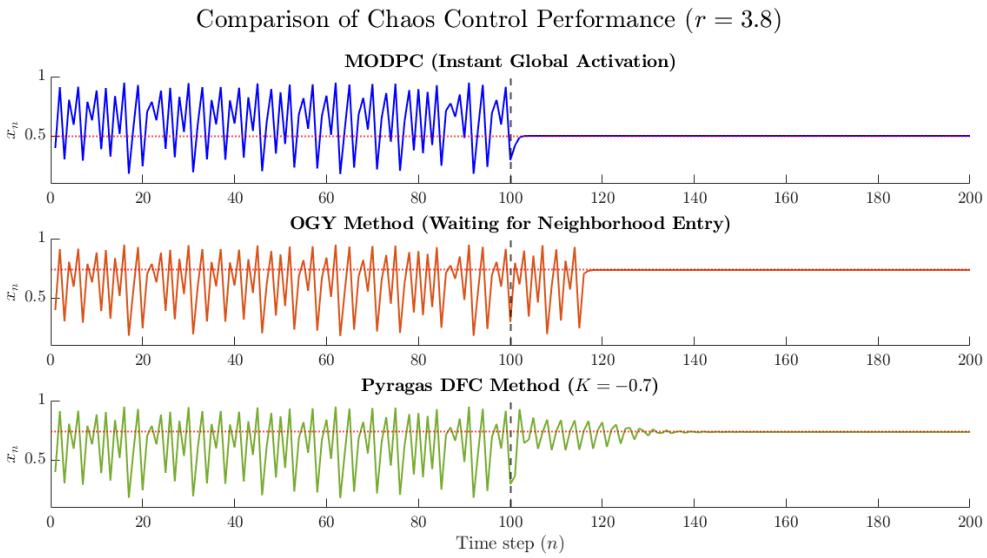


Figure 10. Numerical comparison of OGY, Pyragas DFC, and MODPC for controlling the chaotic Logistic map at $r = 3.8$.

produce stable trajectories. The OGY method stabilizes the trajectory after the chaotic orbit enters the prescribed neighborhood of the target fixed point. The Pyragas DFC method produces a gradual damped convergence toward the same fixed point. In contrast, MODPC becomes active directly through the optimization process and drives the trajectory toward the predefined target at $x = 0.5$.

Although all three approaches successfully suppress the chaotic behavior, the resulting equilibrium points are not identical because the control objectives are defined differently. OGY and Pyragas stabilize the non-trivial fixed point associated with the original system at $r = 3.8$, whereas MODPC is configured to follow a target selected from a different dynamical regime. This distinction illustrates that the three approaches differ not only in their control mechanisms but also in how the desired controlled state is specified.

3.7.4. UPO-Based Target Test of MODPC To further examine the applicability of MODPC to the type of target used by classical UPO-based control methods, an additional test was conducted by modifying the objective function to directly target the non-trivial period-1 orbit of the Logistic map at $r = 3.8$. The target was obtained analytically from the fixed-point condition

$$x^* = f(x^*) = rx^*(1 - x^*).$$

Solving this equation gives the two fixed points

$$x_1^* = 0, \qquad x_2^* = 1 - \frac{1}{r}.$$

For $r = 3.8$, the non-trivial unstable fixed point is therefore

$$x^* = 1 - \frac{1}{3.8} \approx 0.7368.$$

The objective function was then modified so that the reference trajectory consisted of this constant UPO value. The optimization settings and stopping criterion were otherwise retained. This configuration was intended to provide a closer comparison with OGY and Pyragas, both of which target the unstable period-1 orbit of the original system.

Figure 11 shows the resulting controlled trajectory. Unlike the bifurcation-based target experiment, MODPC did not converge to the desired period-1 fixed point. The chaotic fluctuations were nevertheless suppressed, and the controlled trajectory evolved into a period-2 oscillation around the target value. The resulting MAPE remained

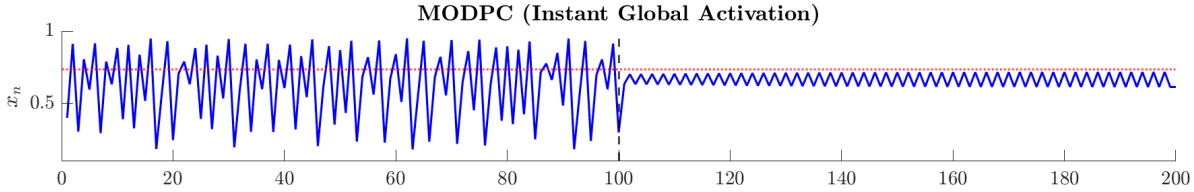


Figure 11. MODPC response when the unstable period-1 orbit of the Logistic map at $r = 3.8$ is directly specified as the target trajectory.

at approximately 10% and did not reach the prescribed stopping threshold of 5% within the maximum 1,000 optimization iterations.

This result indicates that the current MODPC formulation can effectively suppress chaotic behavior but does not directly guarantee stabilization of an arbitrary UPO when that orbit is specified as the target while the original system parameter remains in the chaotic regime. The observed period-2 response suggests that the optimizer identified a more accessible periodic regime that reduced the trajectory error but did not fully satisfy the point-to-point UPO target. Therefore, the UPO-based experiment identifies a limitation of the current objective-control formulation rather than a failure of chaos suppression itself.

3.7.5. Discussion of Methodological Differences The numerical results highlight several methodological differences among the three approaches. OGY uses local dynamical information to construct a parameter perturbation around the selected unstable orbit, while Pyragas DFC uses delayed state information and a feedback gain determined from its stability condition. MODPC instead formulates controller selection as a trajectory-based optimization problem and evaluates candidate parameters through the prescribed objective function.

An important distinction is therefore the definition of the control target. In the present implementation, MODPC can use a trajectory generated from a selected dynamical regime as its reference. This allows the desired controlled behavior to be specified independently from the original chaotic trajectory. However, the additional UPO-based experiment shows that this flexibility does not automatically guarantee stabilization of an UPO. When the non-trivial fixed point at $r = 3.8$ was directly used as the target, the current formulation suppressed the chaotic behavior but converged to a period-2 response and did not reach the prescribed MAPE threshold.

These results indicate that MODPC and the classical methods provide different control mechanisms and should not be considered interchangeable. The present implementation is effective for trajectory-based selection of a desired dynamical regime, while direct stabilization of unstable periodic orbits represents a remaining area for further development.

3.7.6. Computational Comparison on the Seventh-Order Epidemic Model To further illustrate the methodological differences between MODPC and a conventional chaos control method, both approaches were applied to the same seventh-order epidemic model under the chaotic condition $\lambda = 4.6$. The comparison does not focus on the number of iterations or MAPE values, since the two methods obtain their control solutions through fundamentally different procedures. Instead, the comparison focuses on the computational steps required to construct and apply the respective control solutions.

For the OGY implementation, the control procedure requires a local characterization of the unstable dynamics before the feedback controller can be constructed. The epidemic model considered here consists of seven state variables, namely S, V_1, V_2, I, Q, R, and D. The variable D represents a cumulative quantity and does not feed back into the evolution of the other state variables. Therefore, although the original model is seventh-order, the local Jacobian required for the OGY construction can be evaluated on the corresponding six-dimensional dynamical subsystem (S, V_1, V_2, I, Q, R), while D is treated as a dependent cumulative state.

The corresponding fixed point of the six-dimensional subsystem was first obtained by solving the fixed-point equations at the chaotic parameter $\lambda = 4.6$. The resulting state was

$$X^* = [0.1058,\ 0.0119,\ 0.0016,\ 0.4993,\ 0.2131,\ 0.1666]^{\mathrm{T}},$$

Table 8. Conceptual and operational comparison between OGY, Pyragas DFC, and MODPC

Aspect	**OGY**	**Pyragas (DFC)**	**MODPC**
Control principle	Local perturbation	Delayed feedback	Trajectory optimization
Local dynamical information	Used for analytical design	Useful for analytical gain selection	Not required
Jacobian / eigenvalue analysis	Used in analytical formulation	Optional	Not required
Target information	Selected unstable orbit	Target period/orbit	User-defined reference trajectory
Controller parameter	Local perturbation law	Feedback gain K	Optimized automatically
Optimization	No	Optional	Core mechanism
Control objective	Local orbit stabilization	Delayed synchronization	Point-to-point trajectory matching
Activation mechanism	Local neighborhood	Delayed feedback	Optimization-based activation
Controller design	Local analytical construction	Gain and delay selection	Metaheuristic search

with a residual norm of 1.22×10^{-16}. The infectious-state component was $I^* = 0.49933441$. This fixed point was then used to construct the analytical 6×6 Jacobian matrix of the dynamical subsystem.

The eigenvalue analysis of the Jacobian yielded

$$\mathrm{eig}(J) = \{-1.4821, -0.5889, 0.3087, 0.4277, 0.6998, 0.6918\},$$

with a spectral radius of 1.4821. The eigenvalue

$$\mu_u = -1.4821, \qquad |\mu_u| = 1.4821 > 1,$$

was therefore identified as the unstable eigenvalue. The parameter sensitivity vector with respect to the infection-rate parameter was subsequently evaluated as

$$B = \frac{\partial F}{\partial \lambda} = [-0.0540, -0.0036, -0.0002, 0.0577, 0, 0]^{\mathrm{T}},$$

with the projection onto the corresponding unstable left eigenvector given by

$$v_u B = -3.8797 \times 10^{-2}.$$

Using the unstable eigenvalue and its corresponding left eigenvector, the OGY feedback gain was then obtained as

$$K_{\mathrm{OGY}} = [-35.3447,\ 12.8168,\ 1.5792,\ -6.5749,\ -0.1595,\ 0.1056].$$

Thus, even for this seventh-order epidemic model, the OGY implementation requires several system-specific analytical steps before the feedback controller can be applied. These steps include fixed-point determination, analytical Jacobian construction, eigenvalue and eigenvector analysis, parameter sensitivity calculation, and feedback-gain construction. Although the cumulative variable D does not increase the dimension of the local feedback calculation, the remaining six state variables must still be analytically characterized to construct the OGY controller.

Figure 12 shows the resulting OGY-controlled trajectory. Starting from the chaotic condition at $\lambda = 4.6$, the feedback controller successfully suppresses the unstable behavior and drives the system toward the local target dynamics around the identified fixed point.

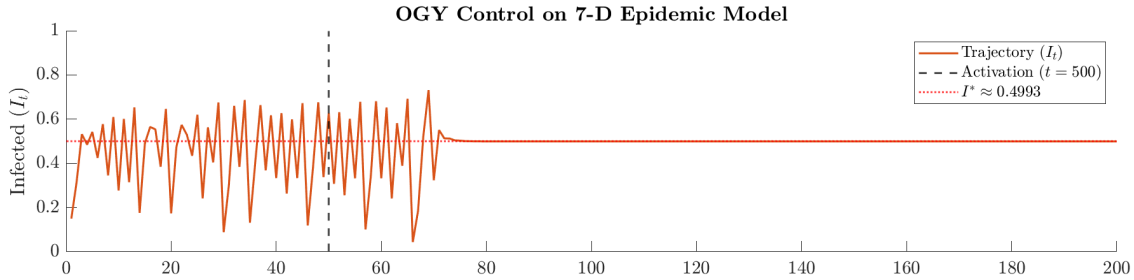


Figure 12. Control of the seventh-order epidemic model using the OGY method from the chaotic condition $\lambda = 4.6$.

In contrast, MODPC does not require the analytical construction of the Jacobian, eigenvalues, eigenvectors, or parameter sensitivity vector. For the same seventh-order epidemic model, the target parameter can be selected directly from the known global dynamical behavior. For example, $\lambda = 2$ can be selected as a target parameter associated with a stable fixed-point regime. The corresponding target trajectory is generated offline, after which the metaheuristic optimization searches for K through the prescribed transformation function g and minimizes the trajectory error.

The resulting MODPC-controlled trajectory is shown in Figure 9a. Starting from the same chaotic condition at $\lambda = 4.6$, MODPC successfully drives the system toward the stable dynamics associated with the selected target parameter. Therefore, both OGY and MODPC are able to control the chaotic epidemic system, but they obtain the control solution through substantially different computational pathways.

The comparison highlights a methodological difference rather than claiming a direct superiority in runtime or iteration count. OGY constructs a local feedback controller from system-specific analytical information around an unstable state, whereas MODPC formulates the intervention as a numerical optimization problem. Consequently, the OGY procedure requires analytical characterization of the local system dynamics before the controller can be constructed, while MODPC primarily requires a target behavior, a prescribed control domain, a transformation function, and repeated numerical evaluations of the system.

3.8. Flexibility of the control parameter

3.8.1. Existence of Control Solutions Before conducting the empirical evaluation of MODPC flexibility through simulations, it is useful to characterize the conditions under which a feasible control parameter can exist. Unlike OGY, which relies on local linearization around a selected UPO, MODPC searches for a control parameter through the prescribed transformation function and objective function. Therefore, the feasibility of the control parameter depends on whether the transformation can map an admissible control value into a parameter region associated with the desired dynamical behavior.

Consider a discrete dynamical system defined by

$$x_{n+1} = F(x_n, p), \qquad x \in \mathbb{R}^m, p \in \mathbb{R}.$$

Let $\mathcal{P}_{\text{chaos}}$ denote the set of parameter values associated with chaotic dynamics, and let $\mathcal{P}_{\text{stable}}$ denote the set of parameter values associated with the desired stable dynamics, such as stable fixed points or periodic orbits.

In this study, membership in $\mathcal{P}_{\text{stable}}$ is defined with respect to the specified dynamical system, model parameters, and stability criterion used in the numerical analysis. Thus, a parameter value is classified as stable only when the corresponding system dynamics satisfy the prescribed stability condition, such as convergence to a fixed point or a stable periodic orbit. This classification does not imply stability for arbitrary initial conditions or under arbitrary perturbations.

The existence of stable parameter regions can be identified from the bifurcation diagrams presented in Section 3.9 (Figs. 13 and 15). Parameter variation to move a system between different dynamical regimes is consistent with the general idea of open-loop bifurcation control [25]. However, the presence of a stable region in the original

parameter space does not by itself guarantee that the region is reachable through a particular transformation function.

Within the MODPC framework, a chaotic parameter $p_c \in \mathcal{P}_{\text{chaos}}$ is modified according to

$$p_{\text{new}} = g(p_c, K),$$

where $K \in \Omega_K$ denotes the control parameter within the prescribed search domain. A feasible stabilizing solution exists when there is at least one $K^* \in \Omega_K$ such that

$$g(p_c, K^*) \in \mathcal{P}_{\text{stable}}.$$

The reachability of a specified target parameter can be examined using the continuity of the transformation function. In particular, the Intermediate Value Theorem provides a sufficient condition for reaching a target value that lies between two attainable values of a continuous mapping [26]. This condition does not imply that every stable region is reachable, since the image of g may not cover the desired parameter region.

The following theorem provides a sufficient condition for a specified target parameter to be reachable through the transformation g.

Theorem 1
(Sufficient Condition for Target Parameter Reachability) Let $g(p_c, \cdot) : \Omega_K \to \mathbb{R}$ be continuous, where $\Omega_K \subseteq \mathbb{R}$ is a connected set. If there exist $K_1, K_2 \in \Omega_K$ such that

$$\min\{g(p_c, K_1), g(p_c, K_2)\} \leq p_T \leq \max\{g(p_c, K_1), g(p_c, K_2)\},$$

then there exists at least one $K^* \in [K_1, K_2]$ satisfying

$$g(p_c, K^*) = p_T.$$

Proof
Since $g(p_c, \cdot)$ is continuous on the connected set Ω_K, the Intermediate Value Theorem applies to the interval between K_1 and K_2. Because p_T lies between $g(p_c, K_1)$ and $g(p_c, K_2)$, there exists at least one $K^* \in [K_1, K_2]$ such that

$$g(p_c, K^*) = p_T.$$

Therefore, p_T is reachable within the admissible control-parameter domain. □

The theorem separates the feasibility of the control parameterization from the numerical optimization used to search for a feasible solution. The continuity condition provides a sufficient local condition for reaching a specified parameter value when that value lies between two attainable values of the transformation. If no such attainable values bracket the target, the theorem does not establish reachability. In particular, a metaheuristic optimizer cannot produce an exact target parameter that lies outside the image of the transformation over the prescribed search domain. Therefore, the practical success of MODPC depends not only on the optimization procedure but also on the reachable set induced by g and Ω_K.

The reachability condition also provides a simple criterion for comparing different transformation functions. Specifically, two transformation functions can both admit feasible control solutions if their respective reachable images intersect the desired stability region.

Corollary 1
(Feasibility Across Different Transformation Functions) Let g_1 and g_2 be two transformation functions defined on the same control-parameter domain. If

$$\text{Im}_{g_1} \cap \mathcal{P}_{\text{stable}} \neq \varnothing$$

and

$$\text{Im}_{g_2} \cap \mathcal{P}_{\text{stable}} \neq \varnothing,$$

then both parameterizations admit at least one feasible stabilizing control parameter.

Proof
For each transformation function, the nonempty intersection with $\mathcal{P}_{\text{stable}}$ means that at least one parameter value in the image corresponds to the desired stable regime. By the definition of the image, each such parameter value has at least one corresponding control parameter. Therefore, both g_1 and g_2 admit at least one feasible control parameter. □

This result indicates that different algebraic forms of the transformation function can admit feasible control solutions when their reachable images intersect the desired stability region. However, the corollary does not imply equivalent convergence speed, optimization difficulty, or control performance across different transformation functions.

Proposition 1
(Robustness to Small Perturbations of the Transformation Function) Let $g(p_c, \cdot)$ be continuous on Ω_K, and suppose that there exist $K^* \in \Omega_K$ and $p^* \in \text{int}(\mathcal{P}_{\text{stable}})$ such that

$$g(p_c, K^*) = p^*$$

and

$$\text{dist}\,(p^*, \partial\mathcal{P}_{\text{stable}}) > \delta > 0.$$

For a perturbed transformation

$$\tilde{g}(p_c, K) = g(p_c, K) + \varepsilon(K),$$

if ε is continuous and

$$\|\varepsilon\|_\infty < \delta,$$

then

$$\tilde{g}(p_c, K^*) \in \mathcal{P}_{\text{stable}}.$$

Hence, the same control parameter K^* remains feasible under the perturbed transformation.

Proof
Since p^* lies in the interior of $\mathcal{P}_{\text{stable}}$ and its distance from the boundary exceeds δ, the open interval

$$(p^* - \delta, p^* + \delta)$$

is contained in $\mathcal{P}_{\text{stable}}$. Because $\|\varepsilon\|_\infty < \delta$,

$$|\tilde{g}(p_c, K^*) - p^*| = |\varepsilon(K^*)| < \delta.$$

Therefore,

$$\tilde{g}(p_c, K^*) \in \mathcal{P}_{\text{stable}}.$$

Hence, K^* remains a feasible control parameter under the specified perturbation. □

This proposition provides a local robustness condition for the transformation mapping. It shows that a feasible solution can remain feasible under sufficiently small perturbations when the corresponding mapped parameter lies inside the stability region with a nonzero margin. The result does not imply robustness for arbitrary perturbations or for targets located at the boundary of a stability region.

The flexibility of the control parameter also makes the MODPC formulation an inverse problem. Rather than specifying the control parameter directly, the desired system behavior is specified through a target parameter or target trajectory, while the corresponding control parameter K must be determined through the transformation function g. For a selected chaotic parameter P_{init} and target parameter P_T, the inverse problem can be expressed as finding

$$K^* \in \Omega_K$$

such that

$$g(P_{\text{init}}, K^*) = P_T,$$

or, for a target stability region, such that

$$g(P_{\text{init}}, K^*) \in \mathcal{P}_{\text{stable}}.$$

The metaheuristic optimizer is therefore used to search for a feasible solution to this inverse problem rather than to establish the existence of such a solution.

The distinction between feasibility and numerical discovery is important. If the target parameter lies outside the reachable image of the transformation over the prescribed domain, i.e.,

$$P_T \notin \text{Im}\left(g(P_{\text{init}}, \Omega_K)\right),$$

the target is outside the set of values that can be reached through the prescribed transformation and control domain. Conversely, when the target lies within the reachable image, a feasible control solution exists in principle within the prescribed domain. In this case, the performance of the metaheuristic reflects its ability to identify such a solution within the available computational budget. Therefore, feasibility is determined by the reachable set induced by the transformation and control domain, whereas numerical discovery depends on the optimization process and its computational budget.

Accordingly, a control result is considered feasible only when the obtained solution satisfies the predefined objective criterion, such as $\text{MAPE} \leq 5\%$. When stochastic metaheuristic optimization is used, the reliability of the numerical procedure can be assessed empirically through repeated independent runs. The success rate, distribution of the final MAPE values, convergence iterations, and computational time provide additional information on the repeatability of the obtained solution. However, repeated numerical success should be interpreted as empirical evidence of reliability under the tested settings rather than as a formal guarantee of global convergence.

It should be emphasized that the reachability results above do not provide a convergence guarantee for the DTBO optimizer. In particular, no general global convergence guarantee is claimed for arbitrary nonlinear or highly complex transformation functions g. The role of the theoretical analysis is to establish when a feasible control solution can exist, whereas the ability of DTBO to discover such a solution within a finite computational budget is evaluated empirically. Consequently, increasing the complexity of g may alter the optimization landscape and may affect the probability and speed with which a feasible solution is discovered.

3.8.2. Numerical Validation The theoretical analysis above indicates that the feasibility of MODPC depends on the reachable set induced by the transformation function g. To examine this flexibility numerically, additional simulations were conducted using several transformation functions with different algebraic structures. The purpose of these experiments was not to establish universality for arbitrary functions g, but to examine whether feasible control solutions could be obtained under different parameterizations of the intervention.

The discrete epidemic model described in Section 3.4 (system (4)) was selected because its parameters have an interpretable physical meaning, allowing the transformation function to be related to potential intervention mechanisms. The system was evaluated at $\lambda = 4.6$, which corresponds to a chaotic regime. The metaheuristic optimization used the same settings as in the previous experiments, with the control parameters bounded between -100 and 100, a population size of 30, and a maximum of 1,000 iterations. For the nonlinear transformations, the search range was subsequently adjusted for the four- and five-variable cases based on the convergence behavior observed during the preliminary runs.

First, four basic algebraic transformations were examined using a single control variable K. These included addition, subtraction, multiplication, and division. For example, the additive transformation was defined as

$$g(P_{\text{init}}, K) = P_{\text{init}} + K.$$

The results from 30 independent runs are summarized in Table 9. All four transformation functions achieved mean MAPE values below the prescribed 5% stopping threshold. The division transformation produced the lowest mean MAPE of $1.6733 \pm 1.5793\%$ and required the fewest iterations, with an average of 3.80 ± 1.13 iterations. The multiplication transformation required the highest average number of iterations (22.87 ± 11.39) and computational

time (1.8965 ± 0.9815 s), while still achieving a mean MAPE of $3.1357 \pm 1.4978\%$. These results indicate that the tested basic algebraic forms can provide feasible control parameterizations, although their convergence behavior and computational cost vary with the form of the transformation function. All 30 independent runs for each transformation reached the prescribed MAPE threshold, corresponding to a success rate of 100% for all four tested cases.

Table 9. Statistical results of MODPC using basic algebraic transformation functions g over 30 independent runs.

Operation	MAPE (%)	Iterations	Time (s)	Success Rate (%)
$+$	2.2949 ± 1.8270	9.23 ± 6.97	0.8097 ± 0.6442	100
$-$	1.9885 ± 1.7200	7.97 ± 4.28	0.7911 ± 0.4362	100
$\times$	3.1357 ± 1.4978	22.87 ± 11.39	1.8965 ± 0.9815	100
$\div$	1.6733 ± 1.5793	3.80 ± 1.13	0.3761 ± 0.1143	100

The transformation functions were then extended to nonlinear forms with multiple control variables. The elements of K were introduced into combinations of trigonometric, exponential, logarithmic, square-root, and hyperbolic functions to create increasingly complex mappings between the control variables and P_{init}. The resulting functions were constructed as numerical test cases rather than calibrated physical intervention models. Each configuration was evaluated over 30 independent runs, and the results are summarized in Table 10.

Table 10. Statistical results of MODPC using nonlinear transformation functions g over 30 independent runs.

n	g	MAPE (%)	Iterations	Time (s)	Search Range
2	$P_{\text{init}} \cdot k_1 + \sin(k_2 \cdot P_{\text{init}})$	2.0678 ± 1.3088	19.73 ± 15.82	1.7327 ± 1.4866	$[-100, 100]$
3	$P_{\text{init}} + k_1 \cdot \exp(-k_2^2) - k_3 \cdot \cos(P_{\text{init}})$	2.3488 ± 1.4178	1.90 ± 1.03	0.2008 ± 0.0953	$[-100, 100]$
4	$P_{\text{init}} + k_1 \cdot \sqrt{\lvert k_2 \rvert} - k_3^2 \cdot \log(k_4^2 + 1)$	2.4739 ± 1.6530	4.40 ± 2.82	0.5774 ± 0.3784	$[-10, 10]$
5	$P_{\text{init}} \cdot (1 + k_1 \cdot \tanh(k_2)) + k_3 \cdot \log(\lvert k_4 \rvert) - k_5 \cdot P_{\text{init}}^2$	2.5657 ± 1.1674	11.00 ± 8.20	0.9379 ± 0.6782	$[-10, 10]$

All four nonlinear transformation functions achieved mean MAPE values below the prescribed 5% threshold under their respective search settings. The two-variable transformation required 19.73 ± 15.82 iterations on average, whereas the three-variable case converged in only 1.90 ± 1.03 iterations. The four- and five-variable transformations required 4.40 ± 2.82 and 11.00 ± 8.20 iterations, respectively. These results indicate that the number of control variables alone does not determine the convergence effort; the algebraic structure and scaling of the transformation function also affect the optimization landscape.

An additional observation was obtained for the four- and five-variable transformations. When the original search domain of $[-100, 100]$ was used, some independent runs failed to reach the prescribed stopping criterion within the maximum iteration limit. The search domain was therefore reduced to $[-10, 10]$ for these two configurations, after which the reported 30-run experiments were conducted. This behavior suggests that the scale and nonlinear structure of g can affect the optimization landscape and that the choice of search domain may influence convergence. The adjustment should therefore be regarded as a numerical design choice for these specific transformation functions rather than evidence that larger search domains are intrinsically unsuitable. All 30 independent runs for each configuration reached the prescribed MAPE threshold, corresponding to a success rate of 100%.

Beyond these numerical test cases, the transformation function can also be formulated to represent a specific physical intervention. For example, suppose $c \in [0, 1]$ represents the level of mask compliance, where $c = 0$ denotes no compliance and $c = 1$ denotes full compliance. An illustrative nonlinear relationship between mask compliance

and the infection-rate parameter can be written as

$$\lambda_{\text{eff}} = \lambda_0 \left(1 - \eta c^{\gamma}\right), \qquad 0 \leq \eta \leq 1, \quad \gamma > 0,$$

where λ_0 is the baseline infection-rate parameter, η represents the maximum fractional reduction associated with the intervention, and γ controls the nonlinear response to compliance. In the MODPC framework, this relationship can be expressed as

$$g(\lambda_0, K) = \lambda_0 \left(1 - \eta K^{\gamma}\right), \qquad K \in [0, 1].$$

The control variable K therefore has a direct interpretation as the intervention level rather than being an abstract numerical perturbation. This formulation is illustrative rather than an empirically calibrated epidemiological relationship; in an application, the parameters η and γ would need to be determined from domain-specific or empirical data.

These results provide numerical evidence that the tested MODPC formulation can operate with different algebraic structures and multidimensional control variables. More importantly, the formulation allows the transformation function to incorporate domain-specific knowledge about how an external intervention may affect a system parameter. In an epidemic application, for example, the elements of K may represent intervention variables such as mask compliance, contact tracing intensity, or mobility restrictions, provided that an appropriate mapping g is defined. The proposed framework therefore separates the definition of the intervention mechanism from the numerical search for its effective control level. The validity of a particular physical mapping, however, remains dependent on the underlying application and would require empirical or domain-specific calibration.

3.9. Chaos shaping

3.9.1. Mathematical Formulation of Target Reachability The concept of chaos shaping in MODPC can be formulated as a target reachability problem over the parameter space characterized by the bifurcation diagram. While traditional chaos targeting methods often rely on exploiting the sensitive dependence on initial conditions to direct system trajectories toward a desired state [1], MODPC approaches targeting directly through the parametric domain. Let $\mathcal{M}$ denote the forward mapping that associates the system parameter p with its corresponding behavioral attractor A, namely $A = \mathcal{M}(p)$. The bifurcation diagram provides a graphical representation of this mapping $\mathcal{M}$.

The objective of chaos shaping is to determine a control parameter K such that the modified system produces a prescribed target attractor A_{target} (for instance, a period-3 orbit).

Theorem 2
(Target Reachability Under a Surjective Transformation) Let a desired dynamical behavior A_{target} correspond to a target parameter p_T such that

$$\mathcal{M}(p_T) = A_{\text{target}}.$$

If the transformation $g(p_c, \cdot)$ is surjective onto a set $\mathcal{D} \subseteq \mathbb{R}$ containing p_T, then there exists at least one $K^* \in \Omega_K$ such that

$$g(p_c, K^*) = p_T,$$

and the target parameter is reachable through the MODPC control parameterization.

Proof
Since $p_T \in \mathcal{D}$ and $g(p_c, \cdot)$ is surjective onto $\mathcal{D}$, there exists at least one $K^* \in \Omega_K$ satisfying

$$g(p_c, K^*) = p_T.$$

By construction, the system evaluated at p_T generates A_{target}. Therefore, the target behavior is reachable through the MODPC control parameterization. □

The surjectivity condition is a sufficient reachability condition rather than a general property guaranteed by MODPC. In practice, the transformation g may not be surjective over the desired parameter range, and the image Im_g may fail to contain p_T. In such cases, no control parameter within the prescribed domain can exactly reach the target parameter, regardless of the optimization algorithm used. The practical feasibility of a target therefore depends on both the control-parameter domain and the range of the transformation g.

3.9.2. Numerical Simulation The previous results demonstrated that MODPC can control chaotic behavior and help the system reach a stable fixed point. However, MODPC can not only eliminate chaotic behavior but also shape it into various desired target behaviors. The determination of the target trajectory X_{Target} from a selected target parameter P_T is an important step in the chaos shaping process. The selection of a target can be systematically guided when information on the global dynamical behavior of the system is available.This can be represented by a bifurcation diagram, which provides a visual roadmap of how a system's dynamics change as a chaotic parameter varies. For example, Figure 13 shows the bifurcation diagram of the discrete epidemic model (i.e., system (4)) with the infection rate (λ) as the bifurcation parameter. At $\lambda = 0.5$, system (4) stabilizes to a disease-free condition. At $\lambda = 2$, system (4) stabilized to a single fixed point. At $\lambda = 3.82$ and $\lambda = 4.15796$, system (4) exhibited period-2 and period-4 cyclic behavior, respectively. Obtaining this global information allows for the selection of specific target behaviors.

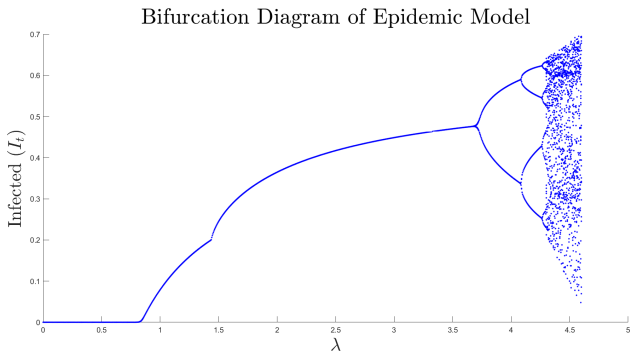


Figure 13. Bifurcation diagram of the discrete epidemic model (i.e., system (4)).

Simulations were conducted to demonstrate the chaos shaping capability of MODPC, where system (4) was controlled from a chaotic state (i.e., $\lambda = 4.6$) toward several different stable conditions. The metaheuristic optimization was conducted using the same settings as before: the control parameter was set to a minimum value of -100 and maximum value of 100, the population size was set to 30, and the maximum iterations were set to 1,000. This time, however, the optimization was directed toward four different P_T values, each of which represented a specific target behavior in the bifurcation diagram. Figure 14 shows the simulation results. From the initial chaotic condition ($\lambda = 4.6$), MODPC was able to transition system (4) not only to different fixed points (Figure 14a, Figure 14b) but also to specific periodic orbits (Figure 14c for a period-2 cycle and Figure 14d for a period-4 cycle). These results demonstrate the effectiveness of using global system information from the bifurcation diagram to improve the effectiveness of the targeting process. These results illustrate that, when the selected target parameter is reachable through the prescribed transformation function, MODPC can shape the chaotic trajectory toward a predefined dynamical regime rather than merely suppressing chaos. The target may therefore be selected according to the desired behavior identified from the global bifurcation structure.

In this process, the bifurcation diagram serves as an offline design tool, where the global dynamical structure is examined and the target parameter P_T is selected before the control optimization is performed. Once the target trajectory X_{Target} has been generated from P_T, the bifurcation diagram is no longer required during the online control phase. The metaheuristic optimizer then operates on the current system trajectory and searches for the control parameter K through the prescribed transformation function g to minimize the objective function. Thus,

the bifurcation diagram determines *what behavior is targeted*, whereas the online optimization determines *how the system is driven toward that target*.

When a complete bifurcation diagram is not available a priori, the target parameter does not need to be selected from a precomputed global diagram. Instead, candidate target parameters can be identified through preliminary parameter sweeps or continuation-based simulations over the admissible parameter range. The resulting trajectories can then be evaluated to identify parameter regions associated with the desired dynamical behavior, such as a stable fixed point or a periodic orbit. A suitable parameter within such a region can subsequently be selected as P_T and used to generate X_{Target}. In this setting, the bifurcation analysis serves as a target-selection procedure performed during the offline phase, rather than as information that must be available analytically in advance. The online MODPC optimization remains unchanged once P_T and X_{Target} have been determined.

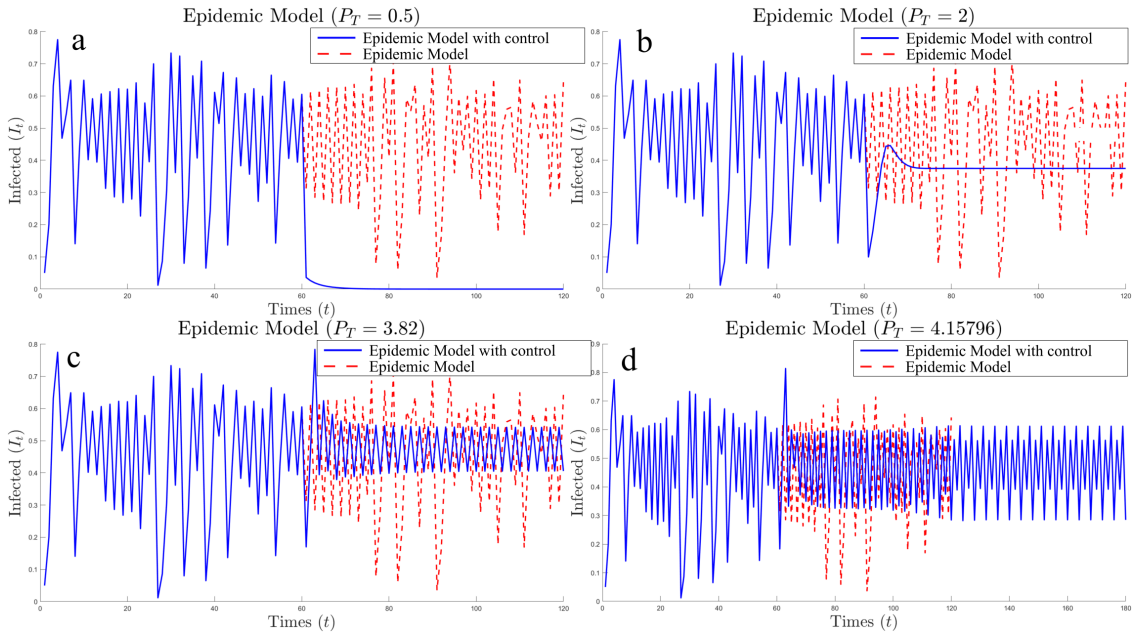


Figure 14. Simulation results from applying MODPC to system (4) with different target behaviors: (a) $P_T = 0.5$, (b) $P_T = 2$, (c) $P_T = 3.82$, and (d) $P_T = 4.15796$.

Furthermore, a critical aspect of controlling chaotic behavior into periodic cycles, such as period 2 and period 4, is handling potential phase lag or time alignment issues. In the proposed MODPC framework, the objective function uses a direct point to point MAPE evaluation without applying dynamic time shifting compensation. This formulation implicitly imposes a phase alignment requirement on the control solution. If a candidate control parameter successfully drives the system toward the desired periodic orbit but introduces a phase lag, the point to point error remains relatively high. As a result, such asynchronous solutions are penalized by the objective function and are unlikely to satisfy the stopping criterion of MAPE $\leq 5\%$. This property can be interpreted as an inherent phase locking behavior of the proposed framework.

The implications of chaos shaping were tested on the Mackey–Glass equation, which is a classic delay differential equation that is widely used to model physiological systems with delayed feedback such as the heartbeat rhythm [29]. The heartbeat rhythm was chosen as an example because it normally forms stable periodic oscillations. However, if the heart experiences arrhythmia (i.e., a disorder that can vary from an irregular heartbeat rhythm to chaotic heartbeat behavior), immediate action is needed to restore the heartbeat rhythm to a normal or near-normal condition. The Mackey–Glass equation describes the rate of change in the variable $x(t)$ (e.g., beat-to-beat interval) at time t:

$$\frac{dx(t)}{dt} = \frac{\beta x(t-\tau)}{1+[x(t-\tau)]^n} - \gamma x(t) \tag{5}$$

This rate of change depends on $x(t-\tau)$, which is the value of x in the past. The parameter τ represents the delay in the biological feedback mechanism, such as the time it takes for a physiological response to affect the next heartbeat. β is the production or input rate of activity, γ is the decay or recovery rate of activity, and n is a nonlinear parameter that shapes the saturation curve in the production term. Changes in n can trigger bifurcations and transitions to chaos, which affect the system's response to feedback. The Mackey–Glass equation can capture the complex dynamics observed in many physiological systems including transitions from stable (i.e., healthy) behavior to complex periodic oscillations (e.g., cardiac alternans) and chaotic behavior (i.e., serious arrhythmia). This makes it a tool capable of providing a rough picture of how a normal heartbeat rhythm can transition to an arrhythmic condition and how an intervention might restore it.

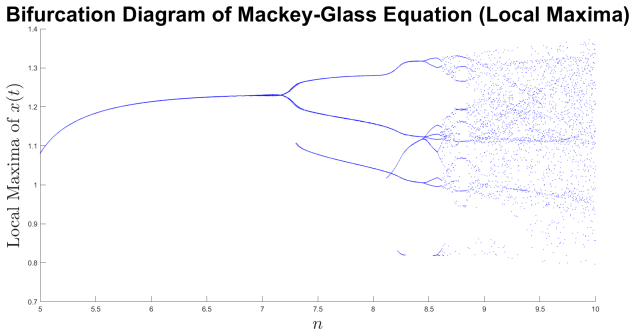


Figure 15. Bifurcation diagram of equation (5).

Figure 15 presents global information on the dynamics of equation (5). This bifurcation diagram is a crucial guide for determining the target condition. This diagram shows how the system's dynamics change from a stable fixed point to a periodic cycle and finally enters a chaotic region as parameter n is increased.

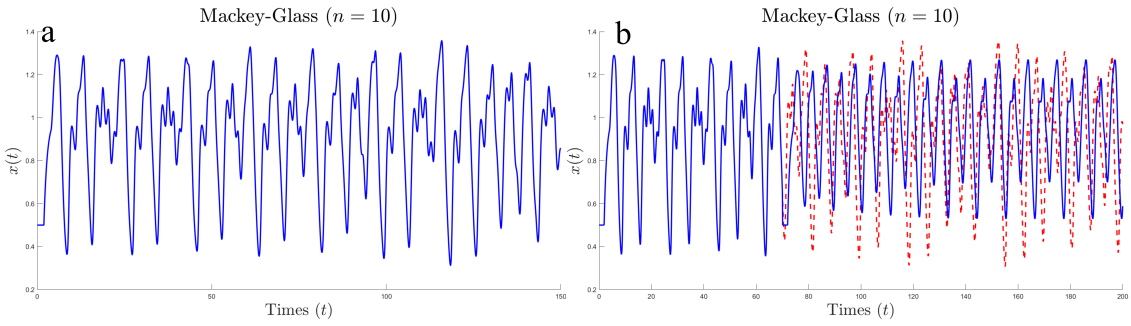


Figure 16. Simulation of equation (5) at $n = 10$: (a) before and (b) after application of MODPC to achieve $n = 7.5$.

Figure 16a shows the chaotic trajectory of equation (5) at $n = 10$. The other parameters were set to $\beta = 2$, $\gamma = 1$, and $\tau = 2$. MODPC was then applied with a target value of $n = 7.5$, which, according to the bifurcation diagram, corresponds to a stable period-3 regime. Figure 16b shows that the controlled trajectory converges to the selected period-3 behavior. Because the Mackey–Glass equation is a simplified mathematical model, this result should be interpreted as an illustration of chaos shaping in a delay system rather than as a direct model of cardiac treatment or clinical intervention.

These results illustrate the potential of the chaos shaping framework for applications in which a chaotic system needs to be guided toward a predefined dynamical behavior. The flexibility of MODPC in this context lies in its ability to incorporate application-specific intervention variables through the transformation function g.

The flexibility of the control parameter K and transformation function g also allows potential application-specific interpretations. While the nonlinear parameter n in the Mackey–Glass equation is not necessarily a directly controllable physical quantity, an application-specific transformation function could be constructed to relate measurable intervention variables to the effective value of n. For example, such variables could represent treatment intensity, stimulation parameters, or other physiological factors, depending on the underlying application. The function g would then encode the assumed relationship between these intervention variables and the model parameter. Such a formulation would, however, require appropriate physiological or empirical calibration before being used for a real-world application.

3.10. Failure Case and Current Limitations

To examine the limitations of the proposed framework, an additional failure-case experiment was conducted by assigning the unstable period-1 orbit of the Logistic map at $r = 3.8$ as the target trajectory. This case was intentionally selected because the target corresponds to an unstable orbit of the original chaotic system and therefore represents a more restrictive control objective than the bifurcation-based target used in the main experiment.

The optimization was performed using the same control settings and a maximum of 1,000 iterations, with a prescribed stopping criterion of $\mathrm{MAPE} \leq 5\%$. As shown in Figure 11, MODPC did not reach the prescribed MAPE threshold within the maximum number of iterations. Instead, the controlled trajectory evolved from the chaotic state into a period-2 oscillation around the target value, with the final MAPE remaining at approximately 10%. Thus, the chaotic behavior was suppressed, but the intended period-1 target was not successfully stabilized.

This behavior indicates that the current formulation can converge toward a suboptimal periodic solution when the specified target is an unstable orbit that is not readily attainable through the available control parameter. The objective function evaluates the point-to-point trajectory error, while the control parameter affects the global dynamical behavior of the system. Consequently, reducing the trajectory error does not necessarily guarantee that the optimizer will reach the specified UPO. The period-2 solution observed in this experiment represents an undesired local solution relative to the prescribed target. This case therefore identifies a practical limitation of the current MODPC formulation and highlights the importance of target feasibility and control-parameter sensitivity in the optimization process.

The failure case also illustrates an important limitation concerning convergence. No formal global convergence guarantee is claimed for DTBO within the MODPC framework, particularly for arbitrary nonlinear transformation functions and complex optimization landscapes. The theoretical results establish conditions under which a feasible control solution can exist, but they do not guarantee that the stochastic metaheuristic will discover that solution within a finite number of iterations. If the target is reachable but the optimizer fails to satisfy the prescribed objective criterion within $MaxIter$, the result should therefore be interpreted as a failure of numerical discovery under the specified computational budget rather than evidence that the target is mathematically unreachable. Conversely, when repeated independent runs consistently satisfy the stopping criterion, the results provide empirical evidence of numerical reliability under the tested settings, rather than a formal proof of global convergence.

4. Conclusion

This study demonstrates that Metaheuristic-Optimized Direct Parameter Control (MODPC) can be used to control and shape chaotic dynamical systems by optimizing control variables that modify a prescribed system parameter through a transformation function. The numerical experiments across discrete and continuous systems show that MODPC can obtain feasible control solutions for the tested transformation functions, including basic algebraic and nonlinear multidimensional forms, without requiring system-specific Jacobian or eigenvalue analysis. The results also demonstrate that MODPC can shape chaotic behavior toward predefined dynamical targets, including stable fixed points and periodic orbits, by using target trajectories obtained from the global dynamical structure of the system. The theoretical analysis further shows that the feasibility of control depends on the reachable set induced by the transformation function and the prescribed control domain, while the numerical experiments

indicate that convergence behavior can vary with the structure and scaling of the transformation. Therefore, the results support MODPC as a flexible framework for chaos control and shaping under the tested conditions, but do not establish universal convergence, global optimality, or robustness for arbitrary transformation functions, search domains, initial conditions, or disturbances. Future work should investigate failure and false-optimum cases more systematically and evaluate MODPC under measurement noise, changing system conditions, and online or real-time control settings.

REFERENCES

1. T. Shinbrot, W. Ditto, C. Grebogi, E. Ott, M. Spano, and J. A. Yorke, *Using the sensitive dependence of chaos (the "butterfly effect") to direct trajectories in an experimental chaotic system*, Physical Review Letters, vol. 68, no. 19, pp. 2863–2866, 1992.
2. G. Grassi, *Special Issue Editorial "Chaotic Systems and Nonlinear Dynamics"*, Symmetry, vol. 14, no. 6, p. 1137, 2022.
3. A. Eilersen, M. H. Jensen, and K. Sneppen, *Chaos in disease outbreaks among prey*, Scientific Reports, vol. 10, no. 1, p. 3907, 2020.
4. B. B. Ferreira, M. A. Savi, and A. S. de Paula, *Chaos control applied to cardiac rhythms represented by ECG signals*, Physica Scripta, vol. 89, no. 10, p. 105203, 2014.
5. E. L. Brugnago, E. C. Gabrick, K. C. Iarosz, J. D. Szezech, R. L. Viana, A. M. Batista, and I. L. Caldas, *Multistability and chaos in SEIRS epidemic model with a periodic time-dependent transmission rate*, Chaos: An Interdisciplinary Journal of Nonlinear Science, vol. 33, no. 12, 2023.
6. E. Ott, C. Grebogi, and J. A. Yorke, *Controlling chaos*, Physical Review Letters, vol. 64, no. 11, pp. 1196–1199, 1990.
7. S. Lynch, *Dynamical Systems with Applications using Python*, Springer International Publishing, Cham, 2018.
8. G. Feng, *Chaotic Dynamics and Chaos Control of Hassell-Type Recruitment Population Model*, Discrete Dynamics in Nature and Society, vol. 2020, pp. 1–9, 2020.
9. G. Feng, *Controlling Chaos of the Ricker Population Model by OGY Method*, American Journal of Biochemistry and Biotechnology, vol. 16, no. 3, pp. 424–431, 2020.
10. F. Kangalgil and S. Işık, *Controlling chaos and Neimark-Sacker bifurcation in a discrete-time predator-prey system*, Hacettepe Journal of Mathematics and Statistics, vol. 49, no. 5, pp. 1761–1776, 2020.
11. S. Rosa and D. F. M. Torres, *Parameter Estimation, Sensitivity Analysis and Optimal Control of a Periodic Epidemic Model with Application to HRSV in Florida*, Statistics, Optimization & Information Computing, vol. 6, no. 1, pp. 139–149, 2018.
12. S. Rosa and D. F. M. Torres, *Optimal Control and Sensitivity Analysis of a Fractional Order TB Model*, Statistics, Optimization & Information Computing, vol. 7, no. 3, pp. 617–625, 2019.
13. H. Abubakar, M. Misiran, A. A. I. Sayed, and A. B. Karaye, *Optimization of Weibull Distribution Parameters with Application to Short-Term Risk Assessment and Strategic Investment Decision-Making*, Statistics, Optimization & Information Computing, vol. 12, no. 6, pp. 1684–1709, 2024.
14. F. Gao and H. Tong, *Particle Swarm Optimization: An Efficient Method for Tracing Periodic Orbits and Controlling Chaos*, Dynamics of Continuous, Discrete and Impulsive Systems Series B: Applications and Algorithms, vol. 13, pp. 1–7, 2006.
15. M. Sadeghpour, H. Salarieh, G. Vossoughi, and A. Alasty, *Multi-variable Control of Chaos Using PSO-Based Minimum Entropy Control*, Communications in Nonlinear Science and Numerical Simulation, vol. 16, no. 6, pp. 2397–2404, 2011.
16. M. Abedini, R. Vatankhah, and N. Assadian, *Stabilizing Chaotic System on Periodic Orbits Using Multi-interval and Modern Optimal Control Strategies*, Communications in Nonlinear Science and Numerical Simulation, vol. 17, no. 10, pp. 3832–3842, 2012.
17. Y. Wang, X. Feng, X. Lyu, Z. Li, and B. Liu, *Optimal Targeting of Nonlinear Chaotic Systems Using a Novel Evolutionary Computing Strategy*, Knowledge-Based Systems, vol. 107, pp. 261–270, 2016.
18. A. Dali, S. Abdelmalek, A. Bakdi, and M. Bettayeb, *A Class of PSO-Tuned Controllers in Lorenz Chaotic System*, Mathematics and Computers in Simulation, vol. 204, pp. 430–449, 2023.
19. M. Dehghani, E. Trojovská, and P. Trojovský, *A new human-based metaheuristic algorithm for solving optimization problems on the base of simulation of driving training process*, Scientific Reports, vol. 12, no. 1, p. 9924, 2022.
20. F. R. Fahreza, *MODPC*, GitHub repository, 2026. Available at: https://github.com/BianLaoZhi/MODPC.
21. R. M. May, *Simple mathematical models with very complicated dynamics*, Nature, vol. 261, no. 5560, pp. 459–467, 1976.
22. M. Hénon, *A two-dimensional mapping with a strange attractor*, Communications in Mathematical Physics, vol. 50, no. 1, pp. 69–77, 1976.
23. E. N. Lorenz, *Deterministic Nonperiodic Flow*, Journal of the Atmospheric Sciences, vol. 20, no. 2, pp. 130–141, 1963.
24. F. R. Fahreza, M. Hasan, and K. A. Santoso, *Chaotic outbreak in discrete epidemic model with vaccination and quarantine interventions and limited medical resources*, Majalah Ilmiah Matematika dan Statistika, vol. 25, no. 1, pp. 22–35, 2025.
25. G. Chen and J. L. Moiola, *An overview of bifurcation, chaos and nonlinear dynamics in control systems*, Journal of the Franklin Institute, vol. 331, no. 6, pp. 819–858, 1994.
26. K. T. Alligood, T. D. Sauer, and J. A. Yorke, *Chaos: An Introduction to Dynamical Systems*, Springer, New York, 1996.
27. M. T. Rosenstein, J. J. Collins, and C. J. De Luca, *A practical method for calculating largest Lyapunov exponents from small data sets*, Physica D: Nonlinear Phenomena, vol. 65, no. 1–2, pp. 117–134, 1993. doi: 10.1016/0167-2789(93)90009-P.
28. J. Diaz-Escobar, P. Díaz-Montiel, S. Venkataraman, and A. Díaz-Ramírez, *Classification and Characterization of Damage in Composite Laminates Using Electrical Resistance Tomography and Supervised Machine Learning*, *Structural Control and Health Monitoring*, vol. 2023, pp. 1–19, 2023, doi: 10.1155/2023/1675867.
29. M. C. Mackey and L. Glass, *Oscillation and Chaos in Physiological Control Systems*, Science, vol. 197, no. 4300, pp. 287–289, 1977.