J. Math. Sci. Model., 8(2) (2025) 42-55 https://doi.org/10.33187/jmsm.1617412

Journal of Mathematical Sciences and Modelling

ISSN (Online) 2636-8692
https://dergipark.org.tr/en/pub/jmsm

Research Article

Modelling and Chaotic Based Parameter Optimization of Sliding Mode Controller

Muhammed Salih Sarıkaya[1*], Onur Demirel[1], Sezgin Kaçar[2] and Adnan Derdiyok[1]

[1]*Department of Mechatronics Engineering, Faculty of Technology, Sakarya University of Applied Sciences, Sakarya, Türkiye*
[2]*Department of Electrical and Electronics Engineering, Faculty of Technology, Sakarya University of Applied Sciences, Sakarya, Türkiye*
[*]*Corresponding author*

Article Info

Keywords: *Chaotic, Henry gas solubility optimization, Hybrid optimization, Sliding mode control*
2020 AMS: *37N35, 90C31, 95B52*
Received: *12 January 2025*
Accepted: *15 April 2025*
Available online: *27 May 2025*

Abstract

This study presents a sliding mode controller design for DC motor speed control using optimization algorithms. The design of sliding mode controllers typically requires expert input during the parameter determination phase. Traditionally, these parameters are set through trial-and-error methods based on the experience of specialists. However, this approach can be both time-consuming and costly. The application of optimization methods automates the parameter-tuning process, reducing human intervention and, in turn, minimizing both design time and costs. The goal of this study is to enhance the performance of optimization methods by hybridizing them with chaotic systems. The random structures of chaotic systems allow optimization algorithms to explore a broader solution space, thereby improving their performance. The analyses conducted in this study reveal that hybrid chaotic algorithms outperform their original ones. The data indicate that the use of hybrid algorithms generally leads to a decrease in Steady-State Error. Additionally, it is observed that when all hybrid algorithms are employed, the sliding mode controller does not exhibit any overshoot. The results demonstrate that the sliding mode controller performs effectively, achieving low settling time, rise time, and steady-state error, while also preventing chattering. Among the methods examined, the sliding mode controller optimized with the Chaotic Henry Gas Solubility Optimization algorithm delivers the best performance, ensuring optimal system stability.

1. Introduction

Sliding Mode Controllers (SMCs) are recognized as an important controller in nonlinear dynamic system control, known for their robustness and efficiency. One of the most important advantages of these controllers is their robustness to system uncertainties and disturbances. SMCs are widely applied in fields such as robotics, aerospace, electric vehicles, energy systems, and automotive industries. However, to maximize the performance of a sliding mode controller, precise tuning of specific controller parameters is essential. The accurate selection of these parameters is critical for system stability. Traditionally, determining controller parameters has been done through trial and error, which is a time-consuming and complex process. This problem can be overcome by optimization methods. Optimization methods allow the sliding mode controller parameters to be tuned more efficiently. These methods simplify controller design, reduce human intervention, and enhance system performance.

Optimization methods help to identify the most suitable parameters or strategies for a given system or process, thereby reducing costs, increasing efficiency and improving processes. In the context of control systems, optimization methods serve as a crucial tool for enhancing controller performance and making the system more robust. Moreover, in recent years, hybrid approaches have been developed to improve the performance of optimization methods. One such approach involves hybridizing optimization methods with chaotic systems. Due to their high sensitivity and randomness, chaotic systems facilitate a broader exploration of the solution space in optimization processes, leading to better results.

Email addresses: **salihsarikaya@subu.edu.tr, onurdemirel@subu.edu.tr, skacar@subu.edu.tr, derdiyok@subu.edu.tr**

Cite as: M. S. Sarıkaya, O. Demirel, S. Kaçar, A. Derdiyok, *Modelling and chaotic based parameter optimization of sliding mode controller*, J. Math. Sci. Model., 8(2) (2025), 42-55.

One of the commonly used optimization methods in the literature is the Henry Gas Solubility Optimization (HGSO) algorithm. This algorithm simulates gas solubility behavior, providing a robust foundation for optimization tasks in various fields such as machine learning and control systems [1,2]. Furthermore, it can be further enhanced by integrating with other metaheuristic algorithms, such as Simulated Annealing (SA), Harris Hawk Optimization (HHO), and the Aquila Optimizer (AQO) [3–6]. The hybridization of HGSO with other metaheuristic algorithms has been shown to enhance the algorithm's performance in terms of solution accuracy and convergence speed. Additionally, there are applications of hybridizing the HGSO algorithm with opposition-based learning (OBL) strategies. Studies such as OBL/HGSO for PID parameter optimization and CNN-OBL/HGSO for autonomous vehicle systems have been conducted [7,8]. To overcome certain limitations in engineering designs, researchers are increasingly using chaotic systems, characterized by sensitivity to initial conditions and complex, unpredictable behavior. The integration of chaotic dynamics into optimization algorithms has been shown to enhance the algorithms' capabilities to escape local optima and improve convergence rates [9,10]. For instance, chaotic maps provide a more structured level of randomness compared to traditional random processes, allowing optimization algorithms to explore the solution space more effectively [10]. This is particularly important in the context of the HGSO algorithm, as chaotic hybridization can improve the search capabilities of the HGSO algorithm by diversifying the exploration strategies used during optimization.

The random behaviors of chaotic systems within a structured framework have led to their hybridization with various metaheuristic optimization methods. The studies in the literature related to this are as follows; Chaotic PID-controlled Particle Swarm Optimization (PSO), logistic and tent map PSO for complex functions, chaotic quantum particle swarm optimization with support vector regression (SVRCQPSO), hybrid multi-stage probabilistic selection particle swarm optimization supported by sine chaotic inertial weight and symmetric tangent chaotic acceleration coefficients (MPSPSO-ST), chaotic PSO for fuzzy system parameter optimization, singer chaotic map hybrid PSO for laser cutting problems (LPSPSO), quantum evolutionary algorithm for chaotic search strategy (QEA), evolutionary algorithms hybridized with chaotic mapping-based Aquila optimizer (CMAOE), chaotic multi-objective evolutionary algorithms, chaotic simulated annealing for the traveling salesman problem, wavelet chaotic simulated annealing neural network for the traveling salesman problem (WCSANN), chaotic simulated annealing for multi-task optimization problems, chaotic whale optimization algorithm for production scheduling problems, chaotic whale optimization algorithm tested with benchmark functions, and hybrid whale optimization algorithm with fractional chaotic map for parameter estimation in wind-diesel power systems [11–26].

In this study, a sliding mode controller has been designed for the speed control of a DC motor. The parameters of the sliding mode controller have been tuned by using optimization methods. Henry Gas Solubility Optimization (HGSO), Particle Swarm Optimization (PSO), Whale Optimization Algorithm (WOA), Simulated Annealing (SA), and Evolutionary Algorithms (EA) optimization methods have been employed. To examine the effects of chaotic systems on optimization methods, these algorithms have been hybridized with the Rössler, Duffing-Van Der Pol, and Sprott-A chaotic equations [27]. The results of the Chaotic Henry Gas Solubility Optimization (CHGSO), Chaotic Particle Swarm Optimization (CPSO), Chaotic Whale Optimization Algorithm (CWOA), Chaotic Simulated Annealing (CSA), Chaotic Evolutionary Algorithms (CEA), HGSO, PSO, WOA, SA, and EA algorithms have been compared.

The paper consists of five sections. The first section provides a brief introduction and an evaluation of recent studies. The second section presents information about the HGSO and CHGSO methods, as well as the chaotic system used. The third section deals with the optimization of sliding mode controller (SMC) parameters for DC motor speed control. In the fourth section, the performances of the optimization methods are compared. The fifth section presents an evaluation of the findings and the conclusions.

2. Materials and Methods

2.1. Henry gas solubility optimization algorithm

The Henry Gas Solubility Optimization Algorithm is a physics-based optimization method. This method is based on Henry's Law, which defines the relationship between the solubility of gases in liquids and the pressure of the gas on the liquid. According to Henry's Law, at a constant temperature, the solubility of gases in a liquid is directly proportional to the partial pressure of the gas. This algorithm considers various factors that affect the solubility of gases in a liquid and aims to increase the solubility level of the gas within the system [28].

The HGSO algorithm takes place in 8 steps.

Step 1. Initialization process: The number of gases (population size N) and their positions are initialized according to the following equation:

$$X_i(t+1) = X_{min} + r * (X_{max} - X_{min}) \tag{2.1}$$

The initial positions of the gas particles are established using equation (2.1). Here, X_{min} and X_{max} define the boundaries of the problem, r is a random number between 0 and 1, and X_i denotes the position of the i^{th} gas particle within the gas particle population. The initial values for Henry's constant, the partial gas pressure, and the constant (enthalpy) are assigned according to the following equation.

$$\begin{aligned} H_j(t) &= l_1 * r \\ P_{ij}(t) &= l_2 * r \\ C_j(t) &= l_3 * r \end{aligned} \tag{2.2}$$

According to equation (2.2), H_j denotes the Henry constant of the cluster j, P_{ij} represents the partial gas pressure of the i^{th} gas particle within cluster j, C_j is the enthalpy constant of cluster j and t indicates the number of iterations. It is important to note that l_1, l_2 and l_3 are constant values, with $l_1 = 0.05$, $l_2 = 100$ and $l_3 = 0.01$ [28].

Step 2. Clustering: The gas particle population is segmented into clusters that each contain an equal number of gas particles. Because the clusters consist of similar gases, the Henry and enthalpy constants are uniform across the clusters.

Step 3. Evaluation: Each gas particles within the clusters are assessed using the objective function. Following this assessment, the gas particles are ranked from best to worst results.

Step 4. Updating the Henry constant: The Henry constant is updated according to equation (2.3).

$$\begin{gathered} H_j(t+1) = H_j(t) * e^{-C*[\frac{4}{T(t)} - \frac{1}{T^0}]} \\ T(t) = e^{-\frac{t}{iter}} \end{gathered} \tag{2.3}$$

In equation (2.3), H_j represents Henry's constant of cluster j, T denotes temperature, $T^0 = 298.15K$ is the reference temperature [28], t refers to the current iteration, and $iter$ is the total number of iterations.

Step 5. Updating the solubility: Gas solubility is updated according to equation (2.4).

$$S_{ij}(t) = K * H_j(t+1) * P_{ij}(t) \tag{2.4}$$

According to equation (2.4), S_{ij} represents the solubility of i^{th} gas particle in cluster j, P_{ij} denotes the partial gas pressure of i^{th} gas particle in cluster j, and K is a constant value.

Step 6. Updating the positions: The positions of gas particles are updated according to equation (2.5).

$$\begin{gathered} X_{ij}(t+1) = X_{ij}(t) + F * r * \Upsilon * (X_{ibest}(t) - X_{ij}(t)) + F * r * a * (S_{ij}(t) * X_{best}(t) - X_{ij}(t)) \\ \Upsilon = \beta * e^{-\frac{F_{best}(t)+\varepsilon}{F_{ij}(t)+\varepsilon}} \\ \varepsilon = 0.05 \end{gathered} \tag{2.5}$$

In equation (2.5), X_{ij} is the position of i^{th} gas particle in cluster j, while r is a random number ranging from 0 to 1. X_{ibest} refers to the position of optimal gas particle, and X_{best} denotes the position of optimal gas particle within the entire population. The parameter Υ indicates the capability of the i^{th} gas particle in cluster j to interact with other gases in its cluster. Parameter a defines the influence of gas particles on the i^th gas particle in cluster j (a=1), while β is a constant value. F_{ij} represents the value of the objective function for the i^{th} gas particle in cluster j, whereas F_{best} refers to the value of the objective function for the optimal gas particle across the entire population. Additionally, F is a flag (F=±1) that alters the direction of movement for the gas particle's position.

Step 7. Escape from local optima: To escape the local optimum, the gas particles are organized, and the less effective gas particles are identified. The selection process is carried out according to equation (2.6).

$$\begin{gathered} N_w(t) = N * (r * (M_2 - M_1) + M_1) \\ M_1 = 0.1 \\ M_2 = 0.2 \end{gathered} \tag{2.6}$$

Here, N is the gas particle number, N_w is the inferior gas particle number to be selected.

Step 8. Updating the positions of the worst agents: The positions of the gas particles chosen in step 7 are randomly updated within the global boundaries of the problem.

$$G_{ij}(t+1) = G_{min} + r * (G_{max} - G_{min}) \tag{2.7}$$

According to equation (2.7), G_{ij} represents the position of i^{th} gas particle in cluster j and G_{min}, G_{max} denotes the global boundaries of the problem. For clearer comprehension of the HGSO method, the pseudocode for the optimization algorithm is provided in Algorithm 2.1.

```
Initialization $X_i$ (i = 1, 2,... N), number of gas types $i$, $H_j$, $P_{ij}$, $C_j$, $l_1$, $l_2$ and $l_3$. equations (2.1) and (2.2)
Divide the population agents into the number of gas types (cluster) with the same Henry's constant value $H_j$.
Evaluate each cluster $j$.
Get the best gas $X_{ibest}$ in each cluster, and the best search agent $X_{best}$.
while $t <$ maximum number of iterations do
    for each search agent do
        Update the positions of all search agents using equation (2.5)
    end for
    Update Henry's constant of each gas type using equation (2.3)
    Update solubility of each gas using equation (2.4)
    Rank and select number of worst agents using equation (2.6)
    Update the position of the worst agents using equation (2.7)
    Update the best gas $X_{ibest}$, and the best search agent $X_{best}$.
end while
$t = t + 1$
return $X_{best}$
```

Algorithm 2.1: Pseudo-code of HGSO algorithm [28]

2.2. Chaotic Henry gas solubility optimization algorithm

Chaotic systems are dynamic systems that exhibit unpredictable behaviors due to their sensitivity to initial conditions, despite having a deterministic structure. Among the most prominent characteristics of these systems are complexity, extreme sensitivity, and cyclic structures. These features of chaotic systems present significant opportunities in optimization processes. In particular, chaotic structures enhance the exploration capabilities of optimization algorithms, enabling them to avoid local optima and closer to global solutions.
The hybridization of optimization methods with chaotic systems utilizes the complex nature of these systems as an advantage, thereby enhancing algorithm performance. This hybridization is typically implemented by integrating chaotic maps into optimization algorithms. For example, the use of chaotic behavior in the exploration phases of traditional optimization algorithms expands the search area and increases the probability of finding an optimal solution.
Hybridization is particularly successful in reducing the time to solution and improving the quality of the solutions obtained. The literature indicates that hybrid optimization methods provide higher success rates. For these reasons, the Rössler chaotic system (2.8), the Duffing-Van Der Pol Chaotic System (2.9), and the Sprott-A chaotic system (2.10) have been selected for hybridizing the HGSO algorithm in this study [27, 29]. The equation and initial conditions for the Rössler chaotic system are provided below.

$$\begin{aligned} \dot{x} &= -y - z \\ \dot{y} &= x + a * y \\ \dot{z} &= b + z * (x - c) \end{aligned} \tag{2.8}$$

The parameters and initial conditions are $a = 0.2$, $b = 0.2$, $c = 5.7$; $x(0) = -9, y(0) = 0, z(0) = 0$, respectively [30].

$$\begin{aligned} \dot{x} &= y \\ \dot{y} &= a * (1 - x^2) * y - x^3 + b * cos(c * z) \\ \dot{z} &= 1 \end{aligned} \tag{2.9}$$

The parameters and initial conditions are $a = 0.2$, $b = 5.8$, $c = 3$; $x(0) = 0, y(0) = 0, z(0) = 0$, respectively [27].

$$\begin{aligned} \dot{x} &= y \\ \dot{y} &= y * z - x \\ \dot{z} &= 1 - y^2 \end{aligned} \tag{2.10}$$

The initial conditions are $x(0) = 0, y(0) = 1, z(0) = 0$ [29].

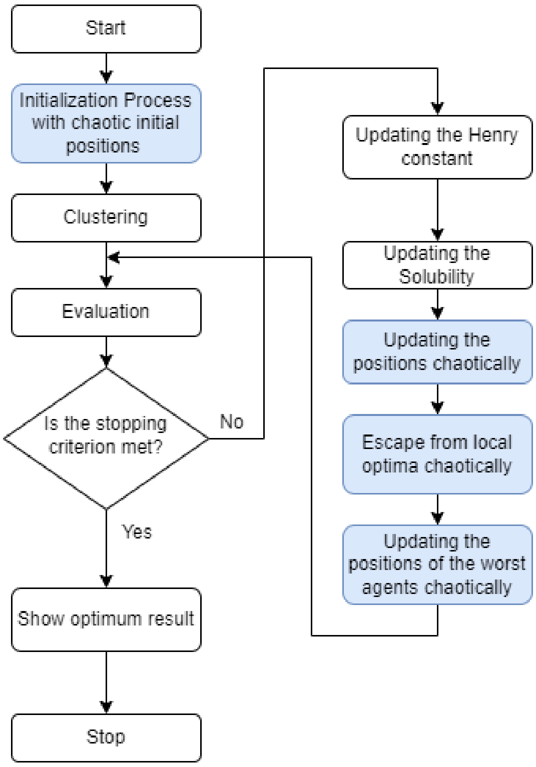

Start
Initialization Process with chaotic initial positions
Clustering
Evaluation
Is the stopping criterion met?
No
Yes
Show optimum result
Stop
Updating the Henry constant
Updating the Solubility
Updating the positions chaotically
Escape from local optima chaotically
Updating the positions of the worst agents chaotically

Figure 2.1: CHGSO Flowchart diagram [27]

The main objective of this study is to boost the performance of the HGSO algorithm by utilizing the advantages and features offered by chaotic systems. Unpredictability, which is the main feature of these systems, plays an important role in increasing the effectiveness of optimization algorithms. Chaotic systems, due to their inherently complex structures, contribute to the creation of a richer and more effective search space when integrated into algorithms. The non-repetitive and ergodic properties of chaotic systems enable the development of a broader and efficient search strategy in stochastic searches. This property suggests that randomness can be used in optimization processes. Specifically, the randomness in chaotic systems becomes an effective tool for improving the overall performance of the algorithm when combined with randomly generated numbers (r) and the equations (2.1),(2.2),(2.5),(2.6),(2.7) [27]. When the HGSO algorithm is hybridized with the Duffing–Van der Pol, Rössler, and Sprott-A chaotic systems, the resulting methods are denoted as $C_{d-v}HGSO$, C_rHGSO, and C_sHGSO, respectively. The flow diagram of the CHGSO algorithm is presented in Figure 2.1 to provide a clearer understanding of the chaotic updates integrated into the HGSO algorithm.

3. Modeling of Sliding Mode Controller for a DC Motor Application

The block diagram of the conventional SMC is presented in Figure 3.1.

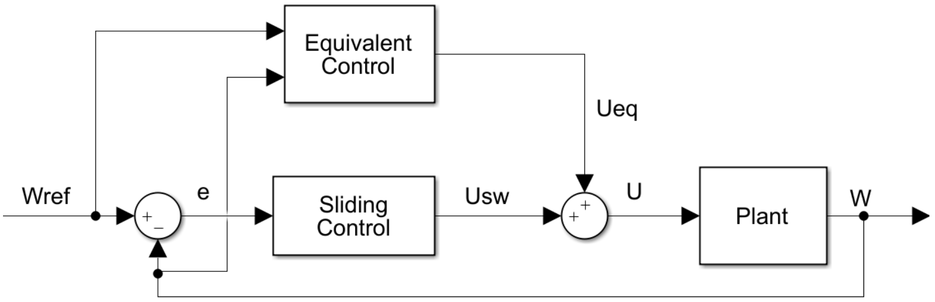


Figure 3.1: Block diagram of the conventional SMC

The control signal applied to the U motor, the equivalent control signal (U_{eq}), and the switching control signal (U_{sw}) are presented in Figure 3.1. The speed control of a DC motor has been implemented using SMC. The transfer function obtained for the speed control of the DC motor is as follows;

$$\frac{\omega(s)}{V_a(s)} = \frac{\frac{K_t}{JL_a}}{\left(\left(\frac{s+R_a}{L_a}\right)\left(\frac{s+B}{J}\right)+\frac{K_bK_t}{JL_a}\right)}. \tag{3.1}$$

The mathematical model of the DC motor is given in equation (3.1). In equation (3.1), V_a is the applied voltage to the motor, R_a is the resistance of the motor windings, L_a is the inductance of the motor windings, K_b is the electrical constant, J is the moment of inertia, B is the damping constant, K_t is the mechanical constant, and ω is the angular velocity of the motor. In this study, the parameters of DC motor [27], which were simulated for speed control, are presented in Table 3.1.

Parameter	DC motor
$R_a(\Omega)$	0.517
$L_a(H)$	0.0573
B	0.000244
$J(kgm^2/s^2)$	0.00000145
$K_b(Vs/rad)$	0.0112
$K_t(Nm/A)$	0.0112
reduction ratio	1/52

Table 3.1: The parameters of the DC motor

For simplicity, the following variables have been defined in equation (3.2).

$$\begin{aligned} A &= \frac{K_t}{JL_a} \\ D &= \frac{R_a}{L_a}+\frac{B}{J} \\ E &= \frac{R_a}{L_a}*\frac{B}{J}+\frac{K_bK_t}{JL_a} \end{aligned} \tag{3.2}$$

When the variable definitions are substituted into equation (3.1), the following equations (3.3)-(3.7) are obtained:

$$\frac{\omega(s)}{V_a(s)} = \frac{A}{s^2 + Ds + E} \tag{3.3}$$

$$\ddot{\omega}(t) + D\dot{\omega}(t) + E\omega(t) = AV_a \tag{3.4}$$

$$\begin{aligned} x_1 &= \omega(t) \\ x_2 = \dot{x}_1 &= \dot{\omega}(t) \\ \dot{x}_2 &= \ddot{\omega}(t) \end{aligned} \tag{3.5}$$

$$\begin{aligned} X &= \begin{bmatrix} x_1 & x_2 \end{bmatrix}^T \\ y = x_1 &= \omega(t) \\ u &= V_a(t) \end{aligned} \tag{3.6}$$

$$\dot{x}_2 = \ddot{\omega}(t) = -D\dot{\omega}(t) - E\omega(t) + AV_a. \tag{3.7}$$

In the design of SMC, the first step should be the design of the sliding surface. Typically, the switching function is chosen as a linear form of the state variables, as shown in equation (3.8) [31].

$$\begin{aligned} s(X) &= \begin{bmatrix} s_1(X) & s_2(X) & \cdots & s_m(X) \end{bmatrix}^T \\ C &= \begin{bmatrix} c_1^T & c_2^T & c_3^T & \cdots & c_m^T \end{bmatrix}^T \\ s(X) &= CX \end{aligned} \tag{3.8}$$

In traditional SMC, the expression commonly used for designing the sliding surface is in the form given by equation (3.9). Here, $\lambda \in \mathbb{R}^+$Present a positive number denoting the gradient of the sliding surface, and n denotes the system order [31].

$$s(X,t) = \left(\frac{d}{dt} + \lambda\right)^{n-1} e(t) \tag{3.9}$$

For example, in the case of a second-order system, the surface equation is given in equation (3.10).

$$s(X,t) = \left(\frac{d}{dt} + \lambda\right)^{2-1} e(t) = \lambda e(t) + \dot{e}(t) \tag{3.10}$$

In SMC, during the design of the surface, where $s(X,t) = \dot{s}(X,t) = 0$ the error approaches zero. Therefore, the sliding mode is defined as given in equation (3.11).

$$\begin{aligned} s(X,t) &= \lambda e(t) + \dot{e}(t) = 0 \\ \dot{s}(X,t) &= \lambda \dot{e}(t) + \ddot{e}(t) = 0 \end{aligned} \tag{3.11}$$

The condition that ensures the system states reach or move towards the sliding surface is called the reaching condition. When the system is under the reaching condition, its trajectory is in the reaching phase [31]. There are various approaches to the reaching condition in sliding mode control, and the Lyapunov function is widely used. The Lyapunov function is represented by the following equation (3.12).

$$V(s) = \frac{1}{2}s^2(t) \tag{3.12}$$

Here, for stability, the conditions $V > 0$ and $\dot{V} < 0$ must be satisfied.

$$\dot{V}(s) = \frac{1}{2}\frac{d}{dt}s^2(t) \leq -\eta|s(t)| \tag{3.13}$$

In equation (3.13), $\eta \in \mathbb{R}^+$ denotes a positive real number. When the derivative of $V(s)$ is taken, the stability condition described above transforms into the form shown in equation (3.14).

$$\dot{s}(t) \leq -\eta\, \mathrm{sign}(s(t)) \tag{3.14}$$

According to equation (3.14), since there is a condition for reaching $s(t) = 0$ in finite time, it is referred to as the reaching condition. For $\eta > 0$, the system state trajectories will reach the sliding surface $s(t) = 0$ and remain on that surface.
In SMC, for the design of the sliding surface, equation (3.15) can be used, and by utilizing equation (3.16) for the derivative of the sliding surface function, if its derivative is set equal to zero. By substituting into the equation and simplifying, the following equations (3.17)–(3.21) are obtained.

$$s(t) = Ce(t) + \dot{e}(t) = C(\omega_r(t) - \omega(t)) + (\dot{\omega}_r(t) - \dot{\omega}(t)) \tag{3.15}$$

$$\begin{gathered} \dot{s}(t) = 0 = -K\,\mathrm{sign}(s(t)) \\ \dot{s}(t) = C(\dot{\omega}_r(t) - \dot{\omega}(t)) + (\ddot{\omega}_r(t) - \ddot{\omega}(t)) = 0 = -K\,\mathrm{sign}(s(t)) \end{gathered} \tag{3.16}$$

$$\dot{s}(t) = C(\dot{\omega}_r(t) - \dot{\omega}(t)) + \ddot{\omega}_r(t) - (-D\dot{\omega}(t) - E\omega(t) + uA) = -K\,\mathrm{sign}(s(t)) \tag{3.17}$$

$$\dot{s}(t) = (C\dot{\omega}_r(t) - C\dot{\omega}(t)) + \ddot{\omega}_r(t) + D\dot{\omega}(t) + E\omega(t) - uA = -K\,\mathrm{sign}(s(t)) \tag{3.18}$$

$$\dot{s}(t) = (C\dot{\omega}_r(t) - C\dot{\omega}(t)) + \ddot{\omega}_r(t) + D\dot{\omega}(t) + E\omega(t) + K\,\mathrm{sign}(s(t)) = -u(t)A \tag{3.19}$$

$$u(t) = \frac{1}{A}\left[C\dot{\omega}_r(t) - C\dot{\omega}(t) + \ddot{\omega}_r(t) + D\dot{\omega}(t) + E\omega(t) + K\,\mathrm{sign}(s(t))\right] \tag{3.20}$$

$$u(t) = \frac{1}{A}\left[(D - C)\dot{\omega}(t) + E\omega(t) + C\dot{\omega}_r(t) + \ddot{\omega}_r(t) + K\,\mathrm{sign}(s(t))\right] \tag{3.21}$$

In order to reduce chattering in SMC, the control signal is modified using a smooth sigmoid function. The resulting control signal is given in equation (3.22).

$$\begin{gathered} u(t) = \frac{1}{A}\left[(D - C)\dot{\omega}(t) + E\omega(t) + C\dot{\omega}_r(t) + \ddot{\omega}_r(t) + K\left(\frac{s}{|s| + \delta}\right)\right] \\ 0 < \delta < 1 \end{gathered} \tag{3.22}$$

The parameters K, C, and δ in equation (3.22) are values that need to be determined. The block diagram of the designed system is shown in Figure 3.2.

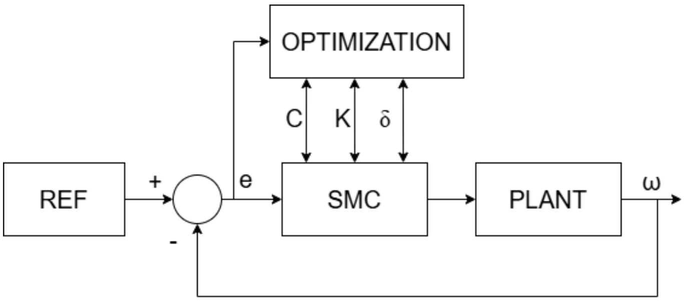


Figure 3.2: Block diagram of the designed system

4. Results

In this section, simulation results are presented to demonstrate the performance of the optimized controllers. Table 4.1 presents the parameters used in optimization methods compared in the study.

Method	Parameters
	Number of iterations: 50
$C_{d-v}HGSO$	Number of gas particles: 50
C_rHGSO	Number of clusters: 5
C_sHGSO	M1 = 0.1, M2 = 0.2
$HGSO$(2019) [28]	L1 = 0.005, L2 = 100, L3 = 0.01
	a, b, k = 1
	e = 0.05
$C_{d-v}PSO$	Number of iterations: 50
C_rPSO	Number of swarm: 50
C_sPSO	C1 = 2.1(Individual learning coefficient)
PSO(1998) [32]	C2 = 2.1(Social learning coefficient)
$C_{d-v}WOA$	
C_rWOA	Number of iterations: 50
C_sWOA	Number of whales: 50
WOA(2016) [33]	
$C_{d-v}sA$	Number of iterations: 50
C_rSA	Number of materials: 50
C_SSA	Cooling rate: 0.98
SA(1987) [34]	
$C_{d-v}EA$	Number of iterations: 50
C_rEA	Number of parents: 20
C_sEA	Number of children: 4
EA(2002) [35]	

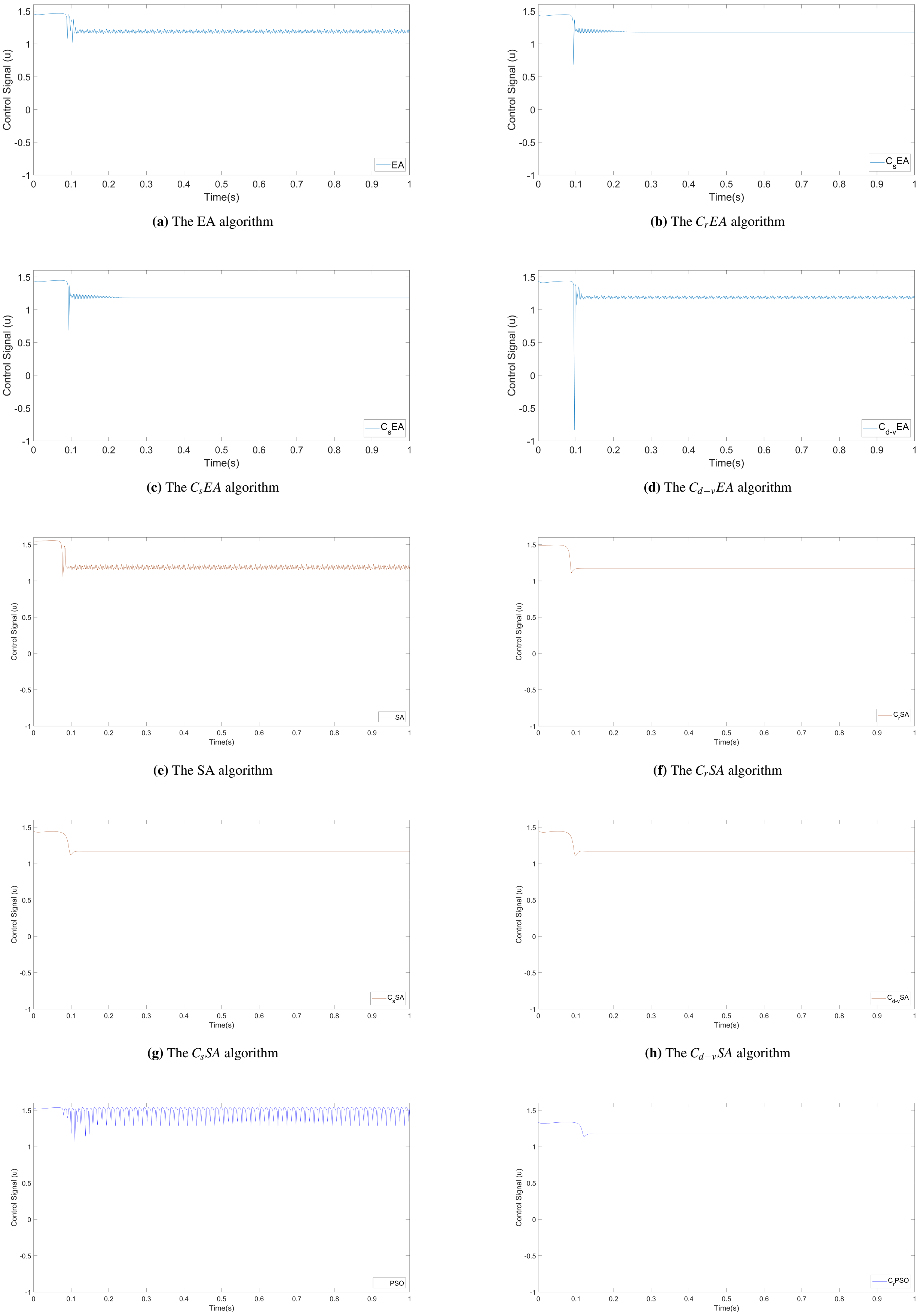
Table 4.1: Algorithm parameters

The parameters K, C, and δ of the sliding mode controller, as specified in equation (3.22), have been optimized using the HGSO [28], PSO [32], WOA [33], SA [34], EA [35], algorithms and chaotic variants. The parameters adjusted by these algorithms are presented in Table 4.2.

Method	C	K	δ	Fitness
$C_{d-v}HGSO$	398.062587	209874.34347	0.329845	0.000384
C_sHGSO	362.340149	208738.325916	0.306580	0.000385
$C_{d-v}WOA$	334.291252	209715.546287	0.273285	0.000386
C_rHGSO	375.658937	208968.3768195	0.286351	0.000387
C_rWOA	350.691352	209380.016280	0.268539	0.000388
$HGSO$	350.595683	209657.567603	0.192955	0.000390
C_sWOA	310.583040	208472.372945	0.237463	0.000391
WOA	278.291252	208010.546287	0.207646	0.000392
SA	219.907082	209039.747574	0.18824	0.000398
C_rSA	256.357642	201728.395761	0.452480	0.000420
$C_{d-v}EA$	282.926727	196335.511718	0.121435	0.000448
C_sEA	320.415718	194239.978390	0.148088	0.000450
C_rEA	294.671428	197376.736512	0.153561	0.000463
C_sSA	307.340209	195382.379581	0.728305	0.000471
EA	367.261204	193033.390611	0.113622	0.000474
$C_{d-v}SA$	374.13673	196431.024497	0.926444	0.000501
PSO	340.435828	206990.671095	0.141625	0.000557
C_rPSO	370.617110	180237.177519	0.356741	0.000642
C_sPSO	385.144228	175253.264785	0.654917	0.000674
$C_{d-v}PSO$	387.398806	177413.126137	0.75771	0.000693

Table 4.2: The parameters of the SMC optimized by the algorithms for the DC motor

The parameters of the SMC optimized according to the error-based ITSE objective function are presented in Table 4.2. It is observed that the best objective function result is provided by the $C_{d-v}HGSO$ algorithm. Additionally, when the HGSO, WOA, and EA algorithms are hybridized with the chaotic equation, a convergence toward the minimum of the objective function is observed.



(a) The EA algorithm

(b) The C_rEA algorithm

(c) The C_sEA algorithm

(d) The $C_{d-v}EA$ algorithm

(e) The SA algorithm

(f) The C_rSA algorithm

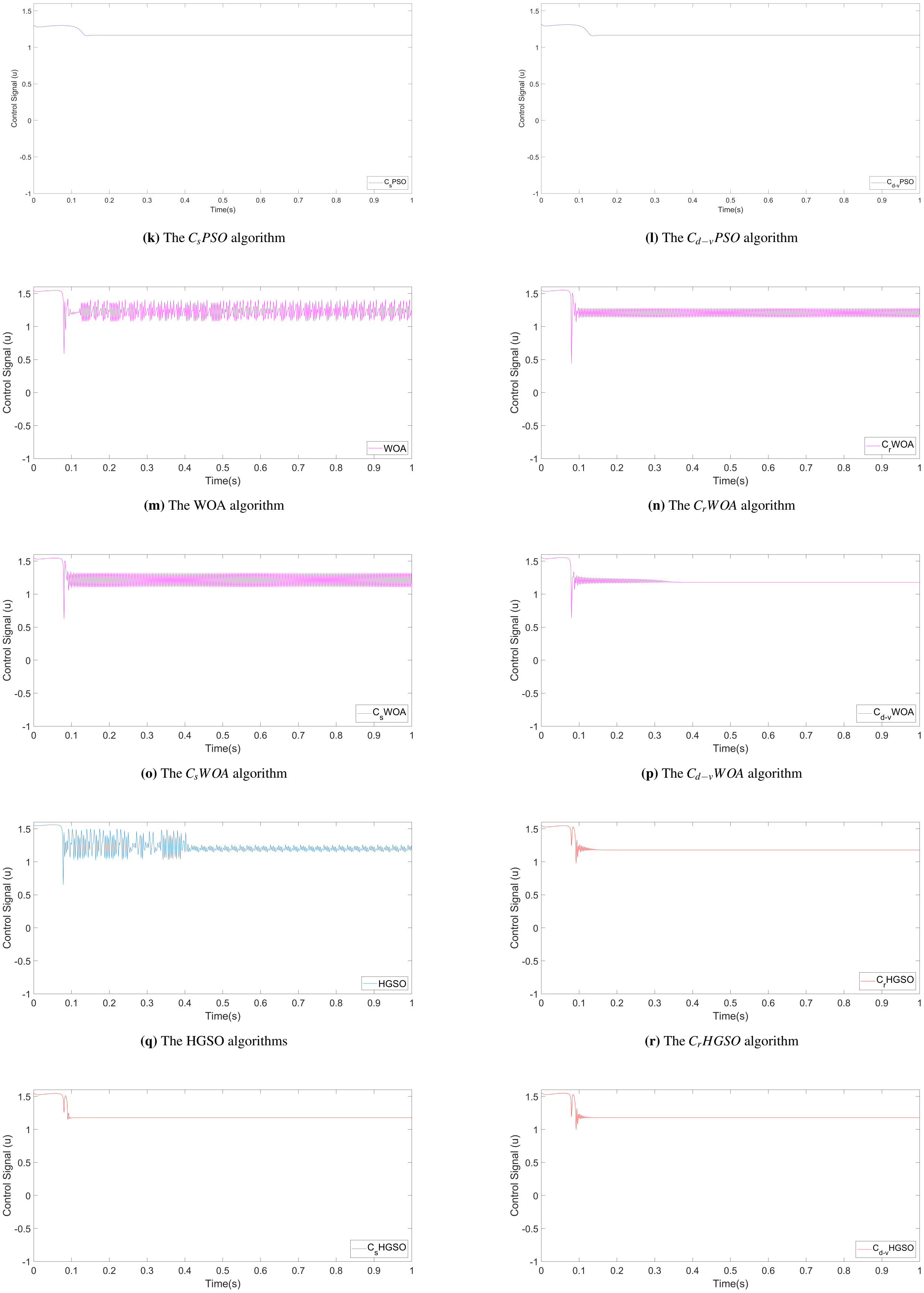
(g) The C_sSA algorithm

(h) The $C_{d-v}SA$ algorithm

(i) The PSO algorithm

(j) The C_rPSO algorithm

Figure 4.1: The control signals generated by the algorithms for DC motor (Part 1)



(k) The C_sPSO algorithm

(l) The $C_{d-v}PSO$ algorithm

(m) The WOA algorithm

(n) The C_rWOA algorithm

(o) The C_sWOA algorithm

(p) The $C_{d-v}WOA$ algorithm

(q) The HGSO algorithms

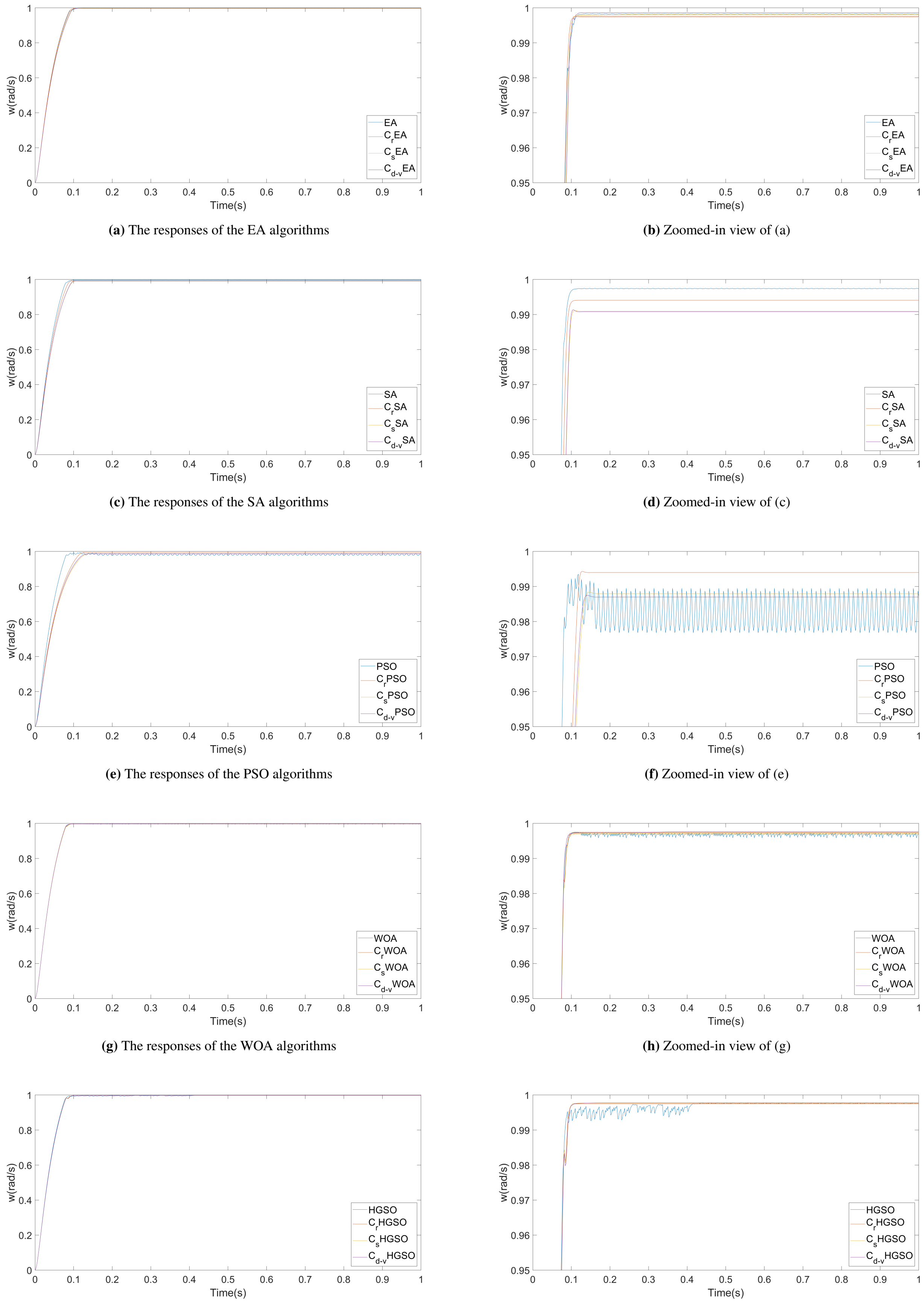
(r) The C_rHGSO algorithm

(s) The $C - sHGSO$ algorithm

(t) The $C_{d-v}HGSO$ algorithm

Figure 4.1: The control signals generated by the algorithms for DC motor (Part 2)

Figure 4.1 presents the control signals generated by the optimized SMC for the DC motor. It can be observed that the chattering is reduced in the optimization methods hybridized with chaotic equations. As the parameter δ in the smooth sigmoid function approaches one, the reduction in chattering is expected. When reviewing Table 4.2, it is evident that the δ values in the hybrid algorithms are closer to 1 compared to the original algorithms. For this reason, it has been observed that hybrid algorithms are more effective in optimizing δ.



(a) The responses of the EA algorithms

(b) Zoomed-in view of (a)

(c) The responses of the SA algorithms

(d) Zoomed-in view of (c)

(e) The responses of the PSO algorithms

(f) Zoomed-in view of (e)

(g) The responses of the WOA algorithms

(h) Zoomed-in view of (g)

(i) The responses of the HGSO algorithms

(j) Zoomed-in view of (i)

Figure 4.2: The step response of the optimized SMC for the DC motor.

Figure 4.2, the speed responses of the DC motor controlled by the sliding mode controller to a unit step reference signal are presented. In Figure 4.2d, it is observed that, except for the SA algorithm, the other algorithms demonstrate lower tracking errors when hybridized with chaotic systems. An analysis of the control signals reveals that the EA and chaotic EA algorithms exhibit similar responses, and no significant reduction in chattering is observed in $C_{d-v}EA$ after hybridization. However, a noticeable decrease in chattering is observed in the Rössler and Sprott-A systems. Although the rise time (Tr) and settling time (Ts) exhibit some degradation in the chaotic PSO algorithm compared to the PSO, a notable reduction in chattering is achieved, indicating improved control smoothness. For the SA and chaotic SA algorithms, although a slight improvement in chattering is noted after hybridization, there are deteriorations in rise time, settling time, and steady-state error.In the evaluation of the WOA and chaotic WOA algorithms, it is noted that the chaotic WOA algorithms do not exhibit any change in settling time and rise time. However, a significant improvement is observed in chattering and steady-state error. The analysis of the results produced by the HGSO and chaotic HGSO algorithms reveals that hybridizing the HGSO algorithm with chaotic systems leads to a reduction in steady-state error and an improvement in settling time. Additionally, while the rise time improves in $C_{d-v}HGSO$, it remains unchanged in the other variants. An examination of the control signals indicates a significant reduction in chattering, where a notable improvement in chattering is observed within the first 0.45 seconds, during which the system remains within the settling band in the speed response.

Based on Figure 4.2, a detailed analysis of the effects of the algorithms on performance metrics is presented in Table 4.3 by examining the responses of the DC motor to the unit step reference signal.

Algorithm	**Tr**	**Ts(%1)**	**Steady-State Error**
$C_{d-v}HGSO$	0.068	0.084	0.0022
C_rHGSO	0.070	0.088	0.0023
C_sHGSO	0.070	0.086	0.0024
$HGSO$	0.070	0.090	0.0025
$C_{d-v}WOA$	0.070	0.088	0.0024
C_rWOA	0.072	0.090	0.0026
C_sWOA	0.072	0.090	0.0028
WOA	0.072	0.090	0.0030
$C_{d-v}EA$	0.082	0.096	0.0014
C_rEA	0.082	0.098	0.0014
C_sEA	0.080	0.094	0.0020
EA	0.078	0.102	0.0018
$C_{d-v}SA$	0.080	0.100	0.0094
C_rSA	0.080	0.102	0.0092
C_sSA	0.076	0.094	0.0058
SA	0.070	0.090	0.0026
$C_{d-v}PSO$	0.096	-	0.0140
C_rPSO	0.098	-	0.0130
C_sPSO	0.094	0.124	0.052
PSO	0.074	-	0.0175

Table 4.3: Comparison of the Step Response of the SMC for the DC motor

When examining the results presented in Table 4.3, the Friedman test was conducted to determine whether at least one of the methods could be considered statistically acceptable. In accordance with common practice in the literature, the significance level was set at $a = 0.05$ [36]. When this test was applied to Table 4.3, the obtained p-value was 0.002727. Since $p < a$, H_0 hypothesis (stating that the methods have similar effects) was rejected, and the alternative hypothesis (H_1, stating that there is a statistically significant difference among the methods) was accepted.

Optimization algorithms were applied to a different DC motor whose parameters are provided in Table 4.4, and the performance was evaluated based on rise time, settling time, and steady-state error criteria, as well as through the Friedman statistical test. Among the compared methods, the CHGSO algorithm yielded the best overall performance across all evaluation metrics.

Parameter	**DC motor2**
$R_a(\Omega)$	0.6
$L_a(H)$	0.012
B	0.0167
$J(kgm^2/s^2)$	0.0167
$K_b(Vs/rad)$	0.8
$K_t(Nm/A)$ 2	0.8

Table 4.4: The parameters of the DC motors2

5. Conclusion

The results of this study demonstrate that hybridizing the HGSO, WOA, EA, and PSO algorithms with chaotic equations significantly enhances control performance. The use of hybrid algorithms generally exhibits a tendency to decrease Steady-State Error. A lower steady-state error means that the system remains in a position closer to the reference value in the long term, and this is an indicator of an improvement in control performance. In addition, it has been observed that the SMC does not show any overshoot. While a significant improvement was achieved in the settling time (Ts) HGSO and EA algorithms for the 1% error band, it was found that the rise time (Tr) remained constant in the HGSO and WOA algorithms. These findings offer an important advantage, especially in control applications that require high precision.

In addition, chattering caused by high frequency switching in physical systems can negatively affect the life of systems. Chattering prevention is of critical importance in terms of preventing damage to system components. The findings obtained in this study reveal that the SMC with chaotic-based optimized parameters not only provides low settling time, rise time, and steady state error, but also offers an effective performance in terms of chattering prevention.

In this context, the application of optimization algorithms in the parameter setting process of control methods reduces time and labor costs by minimizing human intervention. According to the Friedman test, at least one of the applied methods exhibits a statistically significant difference. The results of the study show that their performance can be improved by hybridizing optimization methods with chaotic systems. SMC optimized by the Chaotic Henry Gas Solubility Optimization algorithm has provided optimal stability for the system by exhibiting the best performance compared to HGSO, chaotic WOA, WOA, chaotic EA, EA, chaotic SA, SA, chaotic PSO, and PSO. When comparing the chaotic systems among themselves, it is observed that optimizing the SMC with the Duffing–Van der Pol chaotic system yields better controller responses compared to the Rössler and Sprott-A systems. These obtained results show that optimization methods integrated with chaotic systems offer significant advantages in controller design and can provide valuable contributions to future studies in this field.

Article Information

Acknowledgements: The authors would like to express their sincere thanks to the editor and the anonymous reviewers for their helpful comments and suggestions.

Author's contributions: All authors read and approved the final manuscript. **M.S.S.**: Research design and implementation, literature review, data analysis, and manuscript writing. **O.D.**: Literature review, manuscript revision, and editing. **S.K.**: Editing and supervision. **A.D.**: Editing and supervision.

Artificial Intelligence Statement: No AI tools or technologies were used in the preparation of this manuscript. All content was created solely by the authors.

Conflict of Interest Disclosure: No potential conflict of interest was declared by the authors.

Plagiarism Statement: This article was scanned by the plagiarism program.

References

[1] W. Cao, X. Liu, J. Ni, *Parameter optimization of support vector regression using Henry gas solubility optimization algorithm*, IEEE Access, **8** (2020), 88633-88642. https://doi.org/10.1109/ACCESS.2020.2993267

[2] M. Liu, T. Liu, M. Zhu, et al., *A PI control method with HGSO parameter regulator for trajectory planning of 9-DOF redundant manipulator*, Sensors, **22**(18) (2022), Article ID 6860. https://doi.org/10.3390/s22186860

[3] H. Abdel-Mawgoud, S. Kamel, M. Khasanov, et al., *A strategy for PV and BESS allocation considering uncertainty based on a modified Henry gas solubility optimizer*, Electr. Power. Syst. Res., **191** (2021), Article ID 106886. https://doi.org/10.1016/j.epsr.2020.106886

[4] D.T. Mosa, A. Mahmoud, J. Zaki, et al., *Henry gas solubility optimization double machine learning classifier for neurosurgical patients*, PLoS One, **18**(5) (2023), Article ID e0285455. https://doi.org/10.1371/journal.pone.0285455

[5] W. Xie, C. Xing, J. Wang, et al., *Hybrid henry gas solubility optimization algorithm based on the harris hawk optimization*, IEEE Access, **8** (2020), 144665-144692. https://doi.org/10.1109/ACCESS.2020.3014309

[6] M. Abd Elaziz, D. Yousri, *Automatic selection of heavy-tailed distributions-based synergy Henry gas solubility and Harris hawk optimizer for feature selection: Case study drug design and discovery*, Artif. Intell. Rev., **54**(6) (2021), 4685-4730. https://doi.org/10.1007/s10462-021-10009-z

[7] S. Ekinci, B. Hekimoğlu, D. Izci, *Opposition based Henry gas solubility optimization as a novel algorithm for PID control of DC motor*, Eng. Sci. Technol., Int. J., **24**(2) (2021), 331-342. https://doi.org/10.1016/j.jestch.2020.08.011

[8] S. Ravikumar, D. Kavitha, *CNN-OHGS: CNN-oppositional-based Henry gas solubility optimization model for autonomous vehicle control system*, J. Field Robot., **38**(7) (2021), 967-979.https://doi.org/10.1002/rob.22020

[9] B.M. Alshammari, A. Farah, K. Alqunun, et al., *Robust design of dual-input power system stabilizer using chaotic Jaya algorithm*, Energies, **14**(17) (2021), Article ID 5294. https://doi.org/10.3390/en14175294

[10] E. Emary, H.M. Zawbaa, *Impact of chaos functions on modern swarm optimizers*, PLoS One, **11**(7) (2016), Article ID e0158738. https://doi.org/10.1371/journal.pone.0158738

[11] D. Yan, Y. Lu, M. Zhou, et al., *Empirically characteristic analysis of chaotic PID controlling particle swarm optimization*, PLoS One, **12**(5) (2017), Article ID e0176359. https://doi.org/10.1371/journal.pone.0176359

[12] D. Tian, *Particle swarm optimization with chaotic maps and Gaussian mutation for function optimization*, Int. J. Grid Distrib. Comput., **8**(4) (2015), 123-134. http://dx.doi.org/10.14257/ijgdc.2015.8.4.12

[13] M.L. Huang, *Hybridization of chaotic quantum particle swarm optimization with SVR in electric demand forecasting*, Energies, **9**(6) (2016), Article ID 426. https://doi.org/10.3390/en9060426

[14] Y. Du, F. Xu, *A hybrid multi-step probability selection particle swarm optimization with dynamic chaotic inertial weight and acceleration coefficients for numerical function optimization*, Symmetry, **12**(6) (2020), Article ID 922. https://doi.org/10.3390/sym12060922

[15] I.A. Hodashinsky, M.B. Bardamova, *Tuning fuzzy systems parameters with chaotic particle swarm optimization*, J. Phys. Conf. Ser., **803** (2017), Article ID 012053. https://doi.org/10.1088/1742-6596/803/1/012053

[16] P. Qu, F. Du, *Improved particle swarm optimization for laser cutting path planning*, IEEE Access, **11** (2023), 4574-4588. https://doi.org/10.1109/ACCESS.2023.3236006

[17] C. Yanguang, M. Zhang, C. Hao, *A hybrid chaotic quantum evolutionary algorithm*, In 2010 IEEE International Conference on Intelligent Computing and Intelligent Systems (ICIS), (2010), pp. 771–776. https://doi.org/10.1109/ICICISYS.2010.5658622

[18] M. Verma, M. Sreejeth, M. Singh, et al., *Chaotic mapping based advanced Aquila Optimizer with single stage evolutionary algorithm*, IEEE Access, **10** (2022), 89153-89169. https://doi.org/10.1109/ACCESS.2022.3200386

[19] H. Lu, L. Yin, X. Wang, et al., *Chaotic multiobjective evolutionary algorithm based on decomposition for test task scheduling problem*, Math. Probl. Eng., **2014**(1) (2014), Article ID 640764.https://doi.org/10.1155/2014/640764

[20] H. Lu, X. Wang, Z. Fei, et al., *The effects of using chaotic map on improving the performance of multiobjective evolutionary algorithms*, Math. Probl. Eng., **2014**(1) (2014), Article ID 924652. https://doi.org/10.1155/2014/924652

[21] L. Wang, S. Li, F. Tian, et al., *A noisy chaotic neural network for solving combinatorial optimization problems: Stochastic chaotic simulated annealing*, IEEE Trans. Syst. Man Cybern. Part B: Cybern., **34**(5) (2004), 2119-2125. https://doi.org/10.1109/TSMCB.2004.829778

[22] Y. Jiang, Y. Lei, Z. Zhong, et al., *A wavelet chaotic simulated annealing neural network and its application to optimization problems*, In 2011 International Conference on Network Computing and Information Security (NCIS), (2011), 347-351. https://doi.org/10.1109/NCIS.2011.166

[23] K. Ferens, D. Cook, W. Kinsner, *Chaotic simulated annealing for task allocation in a multiprocessing system*, In 12th IEEE International Conference on Cognitive Informatics and Cognitive Computing (ICCI*CC), (2013), 26-35. https://doi.org/10.1109/ICCI-CC.2013.6622222

[24] J. Li, L. Guo, Y. Li, et al., *Enhancing whale optimization algorithm with chaotic theory for permutation flow shop scheduling problem*, Int. J. Comput. Intell. Syst., **14**(1) (2021), 651–675. https://doi.org/10.2991/ijcis.d.210112.002

[25] H. Ding, Z. Wu, L. Zhao, *Whale optimization algorithm based on nonlinear convergence factor and chaotic inertial weight*, Concurr. Comput.: Pract. Exper., **32**(24) (2020), Article ID e5949. https://doi.org/10.1002/cpe.5949

[26] Y. Mousavi, A. Alfi, I.B. Kucukdemiral, *Enhanced fractional chaotic whale optimization algorithm for parameter identification of isolated wind-diesel power systems*, IEEE Access, **8** (2020), 140862-140875. https://doi.org/10.1109/ACCESS.2020.3012686

[27] M.S. Sarıkaya, Y. H. El Naser, S. Kaçar, et al., *Chaotic-Based Improved Henry Gas Solubility Optimization Algorithm: Application to Electric Motor Control*, Symmetry, **16**(11) (2024), Article ID 1435. https://doi.org/10.3390/sym16111435

[28] F.A. Hashim, E.H. Houssein, M.S. Mabrouk, et al., *Henry gas solubility optimization: A novel physics-based algorithm*, Future Gener. Comput Syst., **101** (2019), 646-667. https://doi.org/10.1016/j.future.2019.07.015

[29] Y. Hamida El Naser, D. Karayel, *Modeling the effects of external oscillations on mucus clearance in obstructed airways*, Biomech. Model. Mechanobiol. **23**(1), (2024), 335-348. https://doi.org/10.1007/s10237-023-01778-3

[30] M. Rafikov, J.M. Balthazar, *On an optimal control design for Rössler system*, Phys. Lett. A, **333**(3-4), (2004), 241-245. https://doi.org/10.1016/j.physleta.2004.10.032

[31] J.Y. Hung, W. Gao, J.C. Hung, *Variable structure control: A survey*, IEEE Trans. Ind. Electron., **40**(1) (1993), 2-22. https://doi.org/10.1109/41.184817

[32] Y. Shi, R.C. Eberhart, *A modified particle swarm optimizer*, In Proceedings of the IEEE International Conference on Evolutionary Computation, (1998), 69–73. https://doi.org/10.1109/ICEC.1998.699146

[33] S. Mirjalili, A. Lewis, *The whale optimization algorithm*, Adv. Eng. Softw., **95** (2016), 51–67. https://doi.org/10.1016/j.advengsoft.2016.01.008

[34] P.J. Van Laarhoven, E.H. Aarts, *Simulated Annealing*, Springer, Berlin, 1987. https://doi.org/10.1007/978-94-015-7744-1_2

[35] H.G. Beyer, H.P. Schwefel, *Evolution strategies–a comprehensive introduction*, Nat. Comput., **1** (2002), 3–52. https://doi.org/10.1023/A:1015059928466

[36] L. Xie, T. Han, H. Zhou, et al., *Tuna swarm optimization: A novel swarm-based metaheuristic algorithm for global optimization*, Comput. Intell. Neurosci., **2021**(1) (2021), Article ID 9210050. https://doi.org/10.1155/2021/9210050