Konya Mühendislik Bilimleri Dergisi, c. 12, s. 2, 494-510, 2024
Konya Journal of Engineering Sciences, v. 12, n. 2, 494-510, 2024
ISSN: 2667-8055 (Electronic)
DOI: 10.36306/konjes.1291710

PARAMETER OPTIMIZATION OF LQR CONTROLLER APPLIED TO THREE DEGREES OF FREEDOM SYSTEM WITH HYBRID APPROACH

[1,*] **Yasin BÜYÜKER**, [2] **İlhan İLHAN**

[1] *KTO Karatay University, Vocational School of Trade and Industry, Electronics and Automation Department, Konya, TÜRKİYE*
[2] *Necmettin Erbakan University, Engineering Faculty, Mechatronic Engineering Department, Konya, TÜRKİYE*
[1] yasin.buyuker@karatay.edu.tr, [2] ilhan@erbakan.edu.tr

Highlights

- Hybrid Algorithm Development: Introduction of a novel hybrid algorithm (hSA-GWO) combining Simulated Annealing (SA) and Gray Wolf Optimization (GWO).
- Optimization Efficiency: Enhanced optimization of complex control parameters for the Linear-Quadratic Regulator (LQR) using advanced algorithms.
- Algorithm Performance Comparison: Comprehensive comparison with traditional optimization algorithms, demonstrating the effectiveness of the proposed method.
- Quadcopter Control Improvement: Significant improvement in controlling quadcopter flight behaviors with the proposed hSA-GWO algorithm.
- Simulation-Based Analysis: Effective application of optimization techniques to the 3-DOF Hover system in simulation for detailed flight behavior analysis.

***Corresponding Author**: Yasin BÜYÜKER, yasin.buyuker@karatay.edu.tr

Konya Mühendislik Bilimleri Dergisi, c. 12, s. 2, 494-510, 2024
Konya Journal of Engineering Sciences, v. 12, n. 2, 494-510, 2024
ISSN: 2667-8055 (Electronic)
DOI: 10.36306/konjes.1291710

PARAMETER OPTIMIZATION OF LQR CONTROLLER APPLIED TO THREE DEGREES OF FREEDOM SYSTEM WITH HYBRID APPROACH

[1,*] Yasin BÜYÜKER, [2] İlhan İLHAN

[1] KTO Karatay University, Vocational School of Trade and Industry, Electronics and Automation Department, Konya, TÜRKİYE
[2] Necmettin Erbakan University, Engineering Faculty, Mechatronic Engineering Department, Konya, TÜRKİYE
[1] yasin.buyuker@karatay.edu.tr, [2] ilhan@erbakan.edu.tr

(Received: 03.05.2023; Accepted in Revised Form: 06.04.2024)

ABSTRACT: There have been numerous studies on the control of quadcopters. These studies mainly aim to control the flight behavior of quadcopters. To achieve this, researchers have been developing new tools and testing new methods. One of the developed tools is the 3-DOF Hover system, which enables researchers to analyze the flight behaviors of quadcopters, such as roll, pitch, and yaw, even in a physically limited area or only in a computer environment. The control method applied in the control of the 3-DOF Hover system has been determined by the manufacturer as Linear-Quadratic Regulator (LQR). LQR has control parameters that are complex to calculate. This complex calculation process creates an optimization problem. Beyond controlling the 3-DOF Hover system using LQR, this study focuses on calculating the complex control parameters of LQR using optimization algorithms when controlling a dynamic system with LQR.

This study includes well-known algorithms such as Genetic Algorithm (GA), Particle Swarm Optimization (PSO), and Simulated Annealing (SA), as well as an innovative approach known Gray Wolf Optimization (GWO). These algorithms were selected due to their proven effectiveness in various studies. Based on the results obtained from these algorithms, a hybrid algorithm incorporating SA and GWO is proposed. The aim of this hybrid algorithm is to combine the advantages of different methods and achieve a more effective and efficient optimization process. The mentioned hybrid algorithm, obtained by combining SA and GWO, is named hSA-GWO. This hSA-GWO is compared with traditional algorithms, and the comparison results show that the proposed hybrid algorithm can be used as an alternative and competitive method for controlling the flight behaviors of quadcopters.

Keywords: *Hybrid Algorithm, LQR Control, Optimization Algorithms, Quadcopter*

1. INTRODUCTION

Unmanned aerial vehicles with four rotors are called quadcopters. Unlike fixed-wing aircraft, they are difficult to control. However, they offer advantages such as maneuverability, portability, runway-free takeoff, and landing, to their users. Despite the difficulties in controlling them, quadcopters are used in many fields due to the advantages they offer. Control methods are used to maximize the benefits of quadcopters and minimize their disadvantages [1]–[4].

Linear and nonlinear control systems exist, and both approaches are evaluated for quadcopters. However, quadcopters are nonlinear systems, unlike fixed-wing aircraft, making linear control methods inadequate for controlling them. Therefore, the handicap caused by using linear control methods in nonlinear systems needs to be overcome. Researchers have turned to nonlinear control methods for this purpose [1]–[13]. One of the control methods considered in this context is LQR control. LQR control is a control method that can be applied to both linear and nonlinear systems. Due to difficulties in calculating controller parameters, LQR control has not been preferred until recently [1], [8].

The optimization of controller parameters in LQR control is a complex optimization process. Initially, this optimization process was conducted using numerical methods. However, difficulties are encountered in reaching the ideal solution using numerical methods. When it is realized that the solution

***Corresponding Author**: Yasin BÜYÜKER, yasin.buyuker@karatay.edu.tr

obtained by the numerical method is not suitable for the system, minor changes are made to the obtained solution using a trial-and-error method to try to reach the ideal solution. In other words, the fate of the developed system depended on both the large computational cost caused by numerical methods and the luck of the researcher. With the development of high-capacity computers, the computational cost of numerical methods is transferred to optimization algorithms. With the help of these processes, complex optimization problems can be solved in an applicable time and in a way that is close to being correct [14]–[17]. As it can be understood, optimizing the controller parameters of LQR is also a complex optimization process. Therefore, it is a reasonable solution to use optimization algorithms in optimizing the controller parameters of LQR. In addition, many researchers today are optimizing LQR parameters using optimization algorithms [1], [8], [9].

In this study, a simulation model of the 3-DOF Hover test environment, produced by Quanser which is a company developing test systems in various fields, was used [2]. Currently, LQR is applied to this system. LQR parameters were optimized using the hSA-GWO algorithm, which is a hybridization of SA and GWO, to obtain a better system response. The results obtained were compared with traditional methods.

2. DESCRIPTION OF THE PROBLEM

In Figure 1, a block diagram representing the optimization processes to be performed in this study is given. It can be seen that control will be applied to the 3-DOF Hover system in this diagram. The control method used for the system is LQR, which is represented by the K block in the diagram. The K block is not only a block but also a matrix that is the root of the Ricatti Equation. The Q and R matrices, which are the control matrices of LQR and are optimized using optimization algorithms, are used in the Ricatti Equation to obtain the K matrix [18].

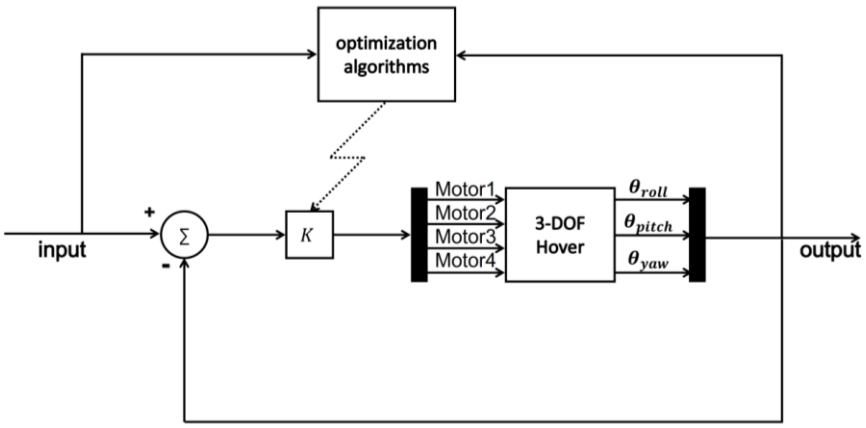

Figure 1. Block diagram of the optimization process.

2.1. 3-DOF Hover

Figure 2 shows the 3-DOF Hover environment developed by the Quanser company. From the image, it can be understood that 3-DOF Hover is a system that can be fixed to the ground and does not have the ability to fly. Because it cannot fly, unlike quadcopters, it has three degrees of freedom instead of six. With this system, the roll, pitch, and yaw movements of quadcopters can be examined in a narrow area. There are two different uses of the system. In the first use, the 3-DOF Hover system and a computer are required. In this way of using the system, the data generated from the behavior of the real-world system is transferred to the computer through a software called Quarc. The second method, which is also preferred in this study, focuses only on the model of the system designed in the computer environment. The data

obtained in the second method is based on the physical laws and system dynamics defined in the model. [1], [2].

The preference for the second method in this study leads to some disadvantages due to the absence of real-world data. For instance, real-world conditions such as wind disturbances are present, which are not considered in the simulation environment. Additionally, even though the motors used in the system are assumed to be identical, there may be slight differences. These mentioned disadvantages result in a greater emphasis on the theoretical aspect of the study. However, simulations allow for a more controlled examination of the system's behavior and enable work on various scenarios. In situations where obtaining difficult or costly real-world data is challenging, simulations offer a valuable alternative. This study aims to focus on model-based analysis of the system, contributing to a general understanding and interpretation of theoretical results.

Figure 2. 3-DOF Hover [2].

The model of the 3-DOF Hover system has been designed in MATLAB/Simulink, which is an accepted program by the researchers. Additionally, the 3-DOF Hover system has been the subject of many studies [1], [2], [8]. The state-space matrices used in the modeling of the system in MATLAB/Simulink program are given in Eqs. (1) – (4) as [1]. The meanings and values of symbols used in these equations are presented in Table 1.

$$A = \begin{bmatrix} 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \tag{1}$$

$$B = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ -\frac{K_t}{J_y} & -\frac{K_t}{J_y} & \frac{K_t}{J_y} & \frac{K_t}{J_y} \\ \frac{L \times K_f}{J_p} & \frac{L \times K_f}{J_p} & 0 & 0 \\ 0 & 0 & \frac{L \times K_f}{J_r} & \frac{L \times K_f}{J_r} \end{bmatrix} \tag{2}$$

$$C = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \end{bmatrix} \tag{3}$$

$$D = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \tag{4}$$

Table 1. Parameters of the system [1].

Symbol	Description	Value	Unit
$K_{t,n}$	Counter rotation propeller torque-thrust constant	0.0036	Nm/V
$K_{t,c}$	Normal rotation propeller torque-thrust constant	0.0036	Nm/V
K_f	Propeller force-thrust constant	0.1188	N/V
l	Distance between pivot to each motor	0.197	m
J_y	Equivalent moment of inertia about the yaw axis	0.110	kgm^2
J_p	Equivalent moment of inertia about the pitch axis	0.0552	kgm^2
J_r	Equivalent moment of inertia about the roll axis	0.0552	kgm^2

2.2. Linear-Quadratic Regulator (LQR)

LQR is used in the control of 3-DOF Hover. Besides being a control method that can be applied to both linear and nonlinear systems, LQR stands out as a state feedback control method. In contrast to output feedback control methods, all state space variables such as speed, acceleration, and angle can be controlled instead of a single output variable in LQR. For example, a linear control method such as PID control receives feedback from the output signal, but this is not the case in LQR. With LQR, it is possible to access and control all designated state variables of the system. A block diagram of the operation of LQR control is given in Figure 3. It is clearly seen here that LQR is a state feedback control method.

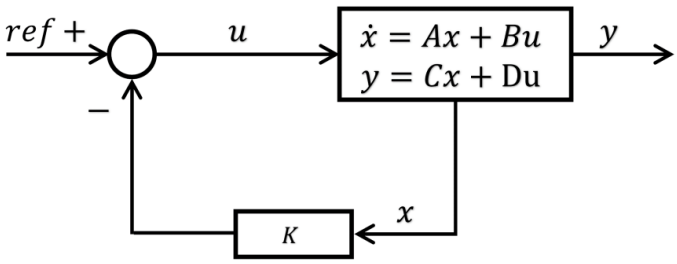

Figure 3. LQR diagram.

To apply LQR, it is necessary to obtain the mathematical model of the system. The state-space matrices formed in Eqs. (1) – (4) represent this mathematical model, and the model created is used in Eq. (5). The goal here is to express the time derivative of the system's state variables in terms of state variables, so that the state variables of the system can be controlled separately with LQR [18].

$$\dot{x} = Ax + Bu \tag{5}$$
$$u = -Kx \tag{6}$$

The cost function of LQR control is as shown in Eqs. (7) – (9). Here, x represents the time-varying state variable of the system, u represents the time-varying control signal, and Q and R represent the LQR controller parameters. LQR control aims to achieve maximum gain with minimum cost using these equations.

$$J = \int_0^\infty (x^T Q x + u^T R u) dt \tag{7}$$
$$x[k+1] = Ax[k] + Bu[k] \tag{8}$$
$$J = \sum_{k=0}^{\infty} (x^T Q x + u^T R u) \tag{9}$$

To minimize this cost function, the Ricatti Equation obtained in Eqs. (12) is used. Here, K represents the root of the equation. It should be noted that the control signal is generated using K and state variables in Eq. (6). Although the operations may seem complex, the LQR control can easily be applied to the system using the "lqr" function developed by MATLAB company.

$$0 = A^T P + PA - PBR^{-1}B^T P + Q \tag{10}$$
$$0 = A^T PA - PA^T PB(B^T PB + R)^{-1}B^T PA + Q \tag{11}$$
$$K = (B^T PB + R)^{-1}B^T PA \tag{12}$$

2.3. Preparing The System for Optimization

In this study, the hSA-GWO method is used to optimize the Q and R parameters of LQR. The Q and R parameters already provided by the manufacturer for the device are given in Eqs. (13) and (14) [3].

$$Q = \begin{bmatrix} 500 & 0 & 0 & 0 & 0 & 0 \\ 0 & 350 & 0 & 0 & 0 & 0 \\ 0 & 0 & 350 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 20 & 0 \\ 0 & 0 & 0 & 0 & 0 & 20 \end{bmatrix} \tag{13}$$

$$R = \begin{bmatrix} 0{,}01 & 0 & 0 & 0 \\ 0 & 0{,}01 & 0 & 0 \\ 0 & 0 & 0{,}01 & 0 \\ 0 & 0 & 0 & 0{,}01 \end{bmatrix} \tag{14}$$

Q and R parameters are matrices with positive real numbers in their diagonal elements (except for the fourth row and fourth column of the Q matrix, which can be zero). The other elements of the matrices have zero values and do not need to be optimized. In summary, there are ten parameters that need to be calculated by optimization algorithms. The parameters to be optimized are named in Eqs. (15) and (16).

$$Q = \begin{bmatrix} q_{11} & 0 & 0 & 0 & 0 & 0 \\ 0 & q_{22} & 0 & 0 & 0 & 0 \\ 0 & 0 & q_{33} & 0 & 0 & 0 \\ 0 & 0 & 0 & q_{44} & 0 & 0 \\ 0 & 0 & 0 & 0 & q_{55} & 0 \\ 0 & 0 & 0 & 0 & 0 & q_{66} \end{bmatrix} \tag{15}$$

$$R = \begin{bmatrix} r_{11} & 0 & 0 & 0 \\ 0 & r_{22} & 0 & 0 \\ 0 & 0 & r_{33} & 0 \\ 0 & 0 & 0 & r_{44} \end{bmatrix} \tag{16}$$

2.3.1. Formation of Search Space

Optimization algorithms perform a search operation on a search space. The search space to be used in this study was determined based on the values provided by the Quanser company for the device in Eqs. (13) and (14). The search space is given in Table 2. Default Quanser parameters and the created search space.. System parameters are very important in any control system. If the system parameters are incorrectly determined in control systems, the systems can easily fail. To cope with this situation, control system designers use saturation blocks in the control system they design. In the "3-DOF Hover" system whose simulation files are used in this study, saturation blocks are also used by the system designer. Therefore, in this study, the search space was kept large and parameters that would cause the system to exhibit undefined behavior, such as division by zero error, were excluded from the search space.

Table 2. Default Quanser parameters and the created search space.

	q_{11}	q_{22}	q_{33}	q_{44}	q_{55}	q_{66}	r_{11}	r_{22}	r_{33}	r_{44}
Qua [2]	500	350	350	0	20	20	0.01	0.01	0.01	0.01
Max	700	500	500	20	50	50	2	2	2	2
Min	1	1	1	0	0.01	0.01	0.01	0.01	0.01	0.01

The lower and upper limits are sufficient due to the parametric proportionality of the LQR controller. The proportionality mentioned here can be explained with a simple example as follows. The current control parameters of the system are [500 350 350 0 20 20 20 0.01 0.01 0.01 0.01 0.01]. If these parameters were applied to the system by multiplying them by a coefficient - for example, if the coefficient were 2, then the control parameters would be [1000 700 700 0 40 40 0.02 0.02 0.02 0.02] - there would be no change in the control response of the system.

Finally, the algorithms will be run for 200 iterations and a population size of 50, as described in the following pages. Therefore, in order to determine the final parameter, 10000 solution sets will be applied to the system and simulated. Here, very bad parameters can also be generated for the control of the system. However, in this study, such parameters are also needed to remove the system from local minima. For example, this provides genetic diversity in the genetic algorithm.

Optimization algorithms try to find the most suitable solution for the system in the created search space by means of metaheuristics. To be able to perform this process, it is necessary to express the problem appropriately. The expressed problem must be in a form that the algorithm can understand. In short, a problem descriptor is required. This problem descriptor is called the objective function.

2.3.2. Formation of Objective Function

A properly constructed objective function will aid in the metaheuristics of the algorithm, enabling the evaluation of better solutions instead of poor solutions in the search space. The objective function created for this problem is given in Eq.(17). The objective function created includes some parameters that are important for dynamic systems [19]. The meanings of these parameters used in the equation are explained in Table 3.

$$Je = (\theta_{Rrt} + \theta_{Prt} + \theta_{Yrt} + \theta_{Rst} + \theta_{Pst} + \theta_{Yst} + \theta_{Rpt} + \theta_{Ppt} + \theta_{Ypt} + \theta_{Ros} + \theta_{Pos} + \theta_{Yos} + \theta_{Rp} + \theta_{Pp} + \theta_{Yp} + \theta_{Rn} + \theta_{Pn} + \theta_{Yn}) \quad (17)$$

Table 3. The meanings of the expressions in the objective function.

Roll			**Pitch**			**Yaw**		
Rise time (sec)	Settling time (sec)	Peak time (sec)	Rise time (sec)	Settling time (sec)	Peak time (sec)	Rise time (sec)	Settling time (sec)	Peak time (sec)
θ_{Rrt}	θ_{Rst}	θ_{Rpt}	θ_{Prt}	θ_{Pst}	θ_{Ppt}	θ_{Yrt}	θ_{Yst}	θ_{Ypt}
Over shot (degree)	Peak (degree)	Norm	Over shot (degree)	Peak (degree)	Norm	Over shot (degree)	Peak (degree)	Norm
θ_{Ros}	θ_{Rp}	θ_{Rn}	θ_{Pos}	θ_{Pp}	θ_{Pn}	θ_{Yos}	θ_{Yp}	θ_{Yn}

The fitness value is calculated using the objective function. In this study, the aim is to achieve the fastest rise time of the system response without maximum overshoot and steady-state error. This can be achieved by minimizing the fitness value. The term 'minimization' is used in optimization processes of this kind.

System response obtained when the default Q and R parameters are applied to the system by the manufacturer are shown in Figure 4. These system responses are obtained without any optimization process. The objective function created will try to make these system responses more optimum.

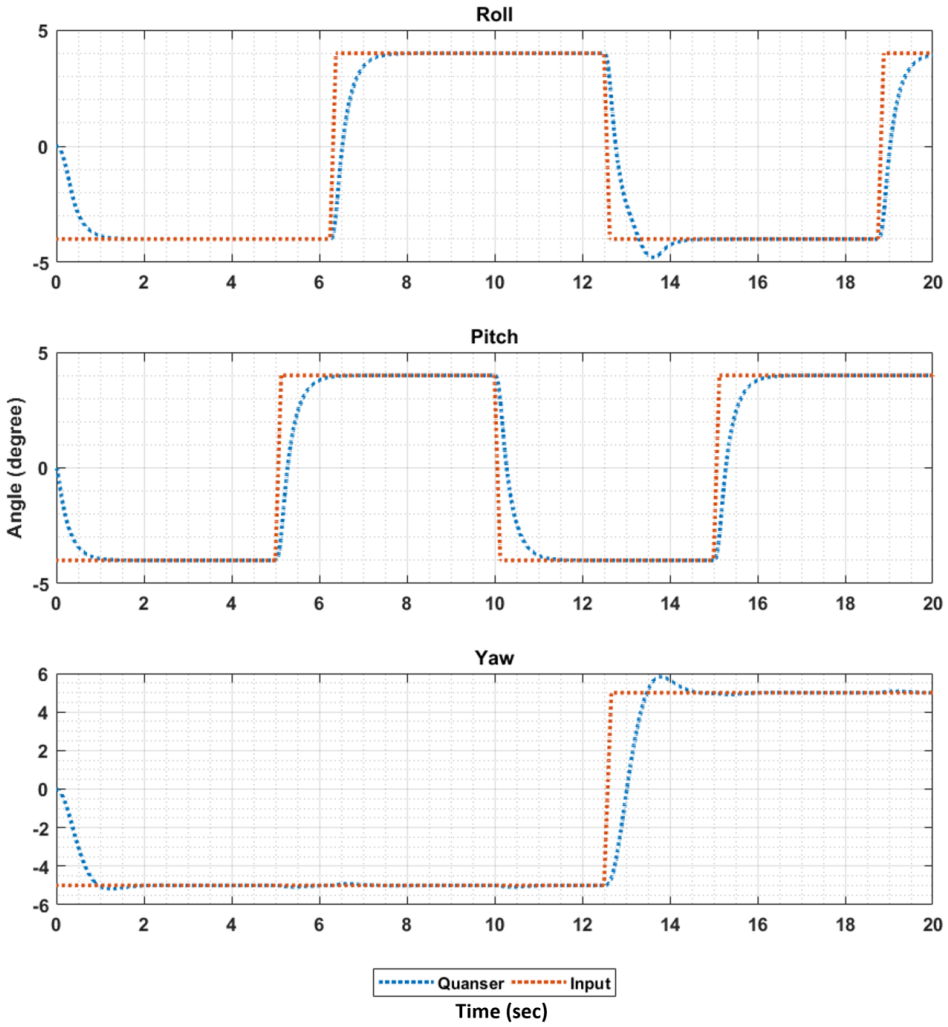

Figure 4. System responses obtained using default Quanser parameters [2].

A dynamically optimum system will try to produce the most ideal output signal against the input signal given to it. For optimum conditions, the output signal must closely follow the input signal. In other words, the better the input signal covers the output signal, the more controlled the system is operating.

3. THE PROPOSED HYBRID ALGORITHM

In this study, a hybrid algorithm (hSA-GWO) containing SA and GWO has been developed for solving the problem under investigation. Hybrid algorithms arise from the combination of two algorithms that solve the same problem. The aim of this process is to simultaneously benefit from the unique advantages of both algorithms. With the performed hybridization process, both SA's ability to overcome local optimal solutions and GWO's early generation of satisfactory solution sets are aimed to be utilized at the same time [20], [21].

The pseudocode of the developed hSA-GWO algorithm along with some abbreviations is provided in Algorithm 1. The developed hSA-GWO works as follows: The population size and iteration number are determined for the GWO section of the hSA-GWO. Starting temperature, ending temperature, and the number of trials parameters are determined for the SA section of hSA-GWO. Similar to GWO, hSA-GWO

also starts with a randomly generated population and the fitness value of the solutions in the population is calculated. Additionally, the positions of the alpha, beta, and delta wolves are determined as in GWO.

Algorithm 1. HSA-GWO algorithm pseudocode.

```
/*
Abbreviations:
dimention of problem = d, number of iteration = gn, number of population = pn,
number of trial = tn, start temperature = t_s, end temprature = t_e,
alpha candidate wolf = acw, beta candidate wolf = bcw,
delta candidate wolf = dcw, annealed alpha wolf = aaw,
annealed beta wolf = abw, annealed delta wolf = adw,
real alpha wolf = raw, real beta wolf = rbw,
real delta wolf = rdw, alpha wolf fitness = awf,
beta wolf fitness = bwf, delta wolf fitness = dwf
*/
Input: The parameters d, a, gn, pn, tn, ts, te
Output: The best solution set, The best fitness value
Begin
  initialize_parameters ()
  define (search_space)
  t_GWO = t_s
  frac_GWO = ( t_e/ t_s) ^ (1 / (gn – 1))
  p = generate_initial_wolf_position ()
  p = limitation_search_space (p)
  f (p_i) = calculate_fitness_value (p_i)
  determination (awc, bwc, dwc)
  [aaw, abw, adw] = apply_sa (awc, bwc, dwc)
  [raw, rbw, rdw] = sort (aaw, abw, adw)
  while (ts < te)
    p = generate_gwo_population (raw, rbw, rdw)
    p = limitation_search_space (p)
    f (p_i) = calculate_fitness_value (pi)
    determination (awc, bwc, dwc)
    [aaw, abw, adw] = apply_sa (awc, bwc, dwc)
    t_GWO = t_GWO * frac_GWO
    [raw, rbw, rdw] = sort (aaw, abw, adw)
  end while
Finish
```

The determined wolf positions here, however, are not directly used. Therefore, the determined wolf positions are expressed as "candidate wolf positions" in this study. For example, the alpha wolf position obtained as a result of this process is referred to as the "alpha candidate wolf position – (acw)". The best position set is determined as the acw, the second-best position set as the beta candidate wolf's position (bcw) and the third-best position set as the delta candidate wolf's position (dcw). First, SA is applied separately for each of the determined candidate wolf positions. To apply SA to the candidate wolf positions, a variable named t is created to represent the current temperature value. The start temperature is assigned to this t variable. The cooling coefficient created in Eq. (18) is used with Eq. (19) to reduce the value of t temperature during the defined number of trials. Here, $frac_{SA}$ represents the cooling coefficient

used for the SA section of the hSA-GWO, t_e represents the ending temperature, t represents the current temperature, and tn represents the number of trials.

$$frac_{SA} = \left(\frac{t_e}{t}\right)^{\frac{1}{tn-1}} \tag{18}$$

$$t = t \times frac_{SA} \tag{19}$$

A loop is created for a predetermined number of trials. Within this loop, the SA procedure, respectively, are applied to candidate wolf positions. The loop will end when the number of trials is reached. Additionally, the equations used ensure that the current temperature is lower than the finishing temperature when the specified number of trials is reached. In other words, the loop can be terminated when the calculated current temperature is less than the finishing temperature. Following the operation, alpha, beta, and delta wolves with SA applied are obtained. These SA-applied positions obtained are named as "annealed wolf" in this study. For example, the SA-applied alpha candidate wolf position is named "annealed alpha wolf – (aaw)". At the end of the SA section, aaw is obtained by applying SA to acw. The start temperature is assigned to the variable t again. The same process is applied for bcw and dcw, as a result, "annealed beta wolf – (abw)" and "annealed delta wolf – (adw)" are obtained.

The fitness values of the wolves that have undergone SA may have changed. Therefore, their fitness values should be reevaluated. Thus, the fitness values of the annealed wolves are sorted in ascending order. As a result of the sorting process, the best solution is determined, and it is named as "real alpha wolf – (raw)" in this study. Similarly, the second-best solution is determined as "real beta wolf – (rbw)" and the third best solution as "real delta wolf – (rdw)". The starting temperature is reduced using Eq. (20) by using the cooling coefficient given in Eq. (21). Here, $frac_{GWO}$ represents the cooling coefficient used for the GWO section of the hSA-GWO, $t_e$represents the ending temperature, t_s represents the starting temperature, and gn represents the number of iterations.

$$frac_{GWO} = \left(\frac{t_e}{t_s}\right)^{\frac{1}{gn-1}} \tag{20}$$

$$t_s = t_s \times frac_{GWO} \tag{21}$$

The wolf population for the new iteration is obtained using raw, rbw, and rdw. The hybridization process described above is applied to the newly generated wolf population, and the hybridization process is repeated for the number of iterations. In summary, there is one outer loop and three inner loops in the hSA-GWO. GWO operates in the outer loop, and SA operates in the inner loops, provided that the initial temperatures are equal.

4. EXPERIMENTAL STUDIES

4.1. Algorithms Used in the Comparison

The performance of the hSA-GWO algorithm was compared with the performances of the GA, PSO, SA, and GWO algorithms. The operators and parameter values used in these algorithms were explained below.

GA was created by modeling the natural selection, crossover, and mutation processes occurring in evolutionary processes [14]. In this implementation, the real-coded version of GA was used. A special multi-point crossover method was used for GA, different from its usage in the literature. The crossover method used is given in Eqs. (22) and (23).

$$chromosome1_{new} = chromosome_1 \times \alpha + (1-\alpha) \times chromosome_2 \tag{22}$$

$$chromosome2_{new} = chromosome_2 \times \alpha + (1-\alpha) \times chromosome_1 \tag{23}$$

As the problem involves ten elements, single-point and two-point crossover operations commonly used in the literature are insufficient. To overcome this limitation, A multi-point crossover method similar to the one described in the literature was utilized. The value denoted by α represents a randomly generated number between [-0.05, 0.05].

The crossover rate (cr) and mutation rate (mr) in GA are crucial parameters significantly affecting the quality of the search process. Both parameters theoretically can be selected from the range [0, 1]. Empirical studies indicate better results when the crossover rate exceeds 0.75 and the mutation rate is below 0.10 [22]. It is critical to select appropriate parameter values to ensure the effective functioning of GA for a given problem. Hence, A test scenario was established with the crossover rates set to 0.75, 0.85, and 0.95, and the mutation rates set to 0.03, 0.05, and 0.07, respectively. Nine scenarios were generated. Each scenario was run three times with 100 iterations and 50 populations. Table 4 shows the obtained results, with 'Best' and 'Worst' indicating the highest and lowest fitness values in all runs, respectively. 'Avg.' represents the average fitness value of all runs. As can be seen from the table, better results were obtained when the crossover rate was 0.85 and the mutation rate was 0.07.

Table 4. The results obtained for different cr and mr values of GA.

	scen.1	**scen.2**	**scen.3**	**scen.4**	**scen.5**	**scen.6**	**scen.7**	**scen.8**	**scen.9**
	cr=0.75 mr=0.03	cr=0.75 mr=0.05	cr=0.75 mr=0.07	cr=0.85 mr=0.03	cr=0.85 mr=0.05	cr=0.85 mr=0.07	cr=0.95 mr=0.03	cr=0.95 mr=0.05	cr=0.95 mr=0.07
Avg.	43.1685	43.1460	40.5917	42.4631	40.7681	40.4742	41.5950	41.0700	41.3856
Best	40.0128	40.7031	40.3187	41.1961	38.2624	40.1983	38.8629	38.2745	39.4478
Worst	45.1583	44.4829	40.8501	44.1879	43.2103	40.9879	43.8623	44.3430	44.2344

PSO, a search method developed by taking inspiration from the social behaviors of living creatures that live in flocks. This algorithm uses a swarm intelligence-based approach [15]. In this study, PSO was applied as described in the literature. However, unlike the literature, only a limitation was imposed on the velocity value. According to this limitation, a particle's velocity will be between -20% and +20% of the maximum position value that the particle can take. For example, the maximum position value that the q_{11} element of the solution set can take is 700. Therefore, the maximum velocity that a q_{11} particle can reach is +140 units, and the minimum velocity value is -140 units. This means that, for example, a q_{11} particle with a current position of 250 can have a next position between the minimum value of 110 and the maximum value of 390.

In PSO, like in GA, there are parameters that directly affect the algorithm's performance. These parameters are the cognitive component (c_1), social component (c_2), and inertia weight (w). In the literature, various methods were used to optimize these parameters. In this study, c_1 and c_2 were chosen to be equal to each other, and their values are 1.50, 1.75, and 2.00, respectively. The w value was created geometrically to decrease from 1 to 0.5, from 0.9 to 0.4, and from 0.8 to 0.3, respectively. Nine different scenarios have been created in total. Each scenario was run three times with 100 iterations and 50 populations. The results are presented in Table 5. According to this table, scenario 1 works well for PSO.

Table 5. The results obtained for different c and w values of PSO.

	scen.1	**scen.2**	**scen.3**	**scen.4**	**scen.5**	**scen.6**	**scen.7**	**scen.8**	**scen.9**
	$c_{1,2}$ = 1.50 w = 0.8 to 0.3	$c_{1,2}$ = 1.50 w = 0.9 to 0.4	$c_{1,2}$ = 1.50 w = 1 to 0.5	$c_{1,2}$ = 1.75 w = 0.8 to 0.3	$c_{1,2}$ = 1.75 w = 0.9 to 0.4	$c_{1,2}$ = 1.75 w = 1 to 0.5	$c_{1,2}$ = 2.00 w = 0.8 to 0.3	$c_{1,2}$ = 2.00 w = 0.9 to 0.4	$c_{1,2}$ = 2.00 w = 1 to 0.5
Avg.	27.8970	36.3531	34.1600	28.1666	32.9110	30.4122	28.3296	32.5397	30.3248
Best	27.7259	34.1887	28.4640	27.9749	28.6407	27.9180	27.8047	29.9864	27.8553
Worst	27.9910	37.5576	41.7170	28.3460	36.6938	34.8586	29.2942	34.1673	34.7977

The annealing process of metals has been mimicked to develop SA and solve nonlinear problems. SA's most significant advantage lies in its structure, which prevents it from becoming trapped in local minima [16]. Various methods have been proposed to adjust the temperature reduction in SA [23]. In this study, a geometric cooling method is employed. Two cooling coefficients ($frac_1$ and $frac_2$) are utilized for SA. $frac_1$ is used in the main loop to decrease the initial temperature at the end of each iteration, while $frac_2$ is used in inner loop, known as the testing process, to reduce current temperature. Eqs. (24) - (27) describe how the cooling coefficients are calculated and utilized, where t_e represents the ending temperature, t_s represents the starting temperature, gn represents the iteration number, and tn represents the testing number.

$$frac_1 = \left(\frac{t_e}{t_s}\right)^{\frac{1}{gn-1}} \tag{24}$$

$$t_s = t_s \times frac_1 \tag{25}$$

$$frac_2 = \left(\frac{t_e}{t}\right)^{\frac{1}{tn-1}} \tag{26}$$

$$t = t \times frac_2 \tag{27}$$

In this study, trials were conducted using different values for the starting and ending temperatures, which are the parameters of SA, in order to optimize them. For example, various studies were conducted using values between 1000 and 100 for the starting temperature and between 0.01 and 0.001 for the ending temperature. However, no significant difference was observed in the results of the trials. Therefore, the starting temperature was set to 100 and the ending temperature to 0.01.

GWO is an optimization algorithm based on the hunting behavior of gray wolves in nature. This algorithm has been applied in this study as described in the literature [17]. All of the described algorithms were run ten times under equal conditions with the parameters outlined in Table 6.

Table 6. The parameters used in the algorithms.

Algorithms	Search Agents	Number of Iterations	Parameters
GA	50	200	crossover rate: 0.85, mutation rate: 0.07, crossover method: multi-point crossover, α: range of randomly generated numbers for multi-point crossover [-0.05 - 0.05]
PSO	50	200	cognitive component (c_1): 1.5, social component (c_2): 1.5, inertia weight: geometrically decreasing from 0.8 to 0.3
GWO	50	200	distance control parameter a: linearly decreasing from 2 to 0
SA	50	200	start temperature: 100, end temperature: 0.01, cooling factor: geometric reduction method (it was described)
hSA-GWO	50	200	distance control parameter a: linearly decreasing from 2 to 0, start temperature: 100, end temperature: 0.01, cooling factor: geometric reduction method (it was described), trial number: 10

4.2. Experimental Results

All tests conducted within the scope of this study were carried out on a computer equipped with an Intel Core i7 4510u processor and DDR3 8 GB RAM. Each algorithm was executed ten times using the parameters described earlier and listed in Table 6. As a result of these experiments, ten fitness values were obtained for each algorithm. The arithmetic mean, standard deviation, best fitness value, worst fitness value, and average elapsed time data associated with these fitness values are presented in Table 7. These data allowed for a superficial comparison of the performance of the algorithms.

For example, when GA was executed ten times, the best fitness value achieved was 33.8631, and no better fitness value was obtained. It is observed that the average fitness value obtained by GA is comparatively lower than that of the other algorithms, and its standard deviation is also lower. Therefore, it can be inferred that GA yielded poorer results in comparison to the other fitness values. Another example pertains to SA, where the best fitness value obtained was 27.7106, and the worst fitness value was 40.9759. Upon examining the average fitness values of SA, it achieved a value of 31.2066, and its standard deviation is relatively high. From this, it can be deduced that SA's performance is closer to the worst fitness value for only a few fitness values, while it is closer to the best for the others.

Table 7. The results obtained by algorithms.

	GA	PSO	SA	GWO	hSA-GWO
Avg. fitness	39.3906	33.2920	31.2066	29.5936	28.5446
Std. deviation.	2.1138	3.9054	4.9310	3.6637	2.2507
Best fitness	33.8631	27.6717	27.7106	27.5239	27.5006
Worst fitness	41.8453	38.0265	40.9759	37.9892	35.2547
Avg. elapsed time (sec.)	1682.0768	1651.0739	1699.7117	1583.6175	2098.8970

When evaluated based on their best results, the performance of the algorithms is in the following order from high to low: hSA-GWO, GWO, PSO, SA, and GA. Comparing the algorithms based on the average fitness value, the standard deviation value, and the best fitness values, it is observed that hSA-GWO yields the best results. On the other hand, when the average time taken is examined, it is seen that hSA-GWO gives a worse result than the others. The reason for hSA-GWO appearing to be poor in terms of time is due to simulating an additional thirty solutions compared to other algorithms in each iteration.

A superficial analysis of the fitness values achieved by the algorithms has been conducted up to this point. Now a more in-depth analysis will be performed using the optimal values obtained by the

algorithms. The third row of Table 7 provides the best fitness values achieved by the algorithms during these studies. The convergence curve for the iterations that produced these fitness values are shown in Figure 5. When comparing the convergence curves of the algorithms, it can be seen that hSA-GWO leads the race for the most part and wins the competition. The solution sets that resulted in the best fitness values reached by the algorithms during these studies are given in Table 8.

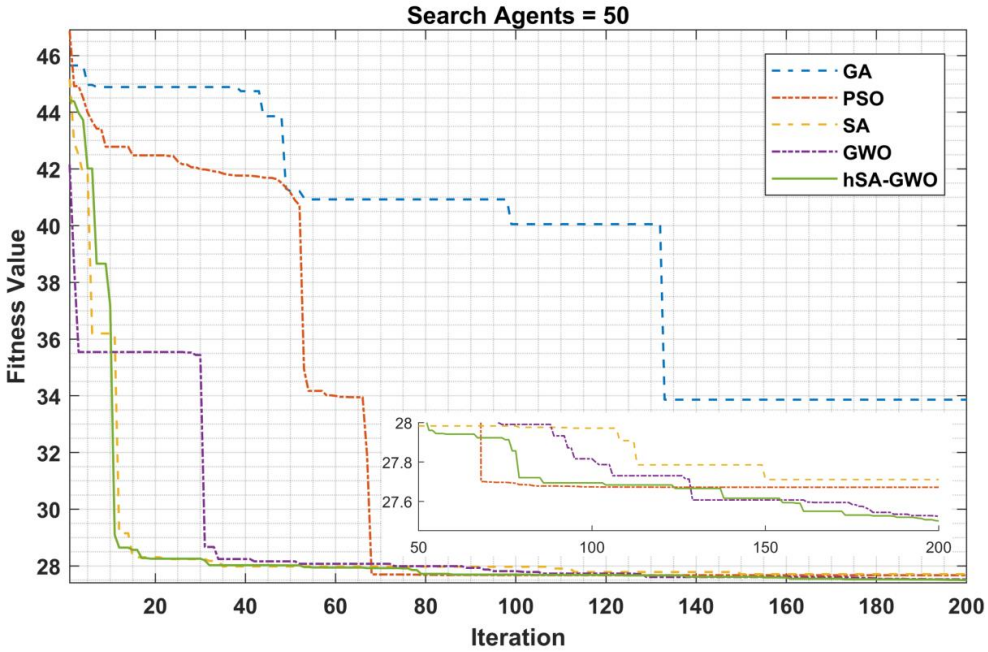

Figure 5. The convergence curves belong to the iterations where the best fitness values were achieved.

Table 8. The best solution sets reached by the algorithms.

	Quanser	GA	PSO	SA	GWO	hSA-GWO
q_{11}	500.0000	549.3473	677.3150	674.8986	696.1097	698.5897
q_{22}	350.0000	362.2833	472.9308	500.0000	498.2206	499.1072
q_{33}	350.0000	136.0409	371.8367	55.1683	166.0198	90.8695
q_{44}	0.0000	1.6330	3.7883	0.0000	3.0516	0.5257
q_{55}	20.0000	8.1317	10.7165	7.8515	8.9430	7.3262
q_{66}	20.0000	25.3702	4.6246	0.0475	2.0045	0.7627
r_{11}	0.0100	0.0100	0.0100	0.0102	0.0100	0.0100
r_{22}	0.0100	0.0196	0.0202	0.0131	0.0152	0.0128
r_{33}	0.0100	1.1162	0.0355	0.0100	0.0168	0.0109
r_{44}	0.0100	0.4970	0.1035	0.0569	0.0610	0.0476

The system response when applying these solution sets is shown in Figure 6. It can be seen that the system response obtained with hSA-GWO reaches the input value given for roll, pitch, and yaw angles earlier than the system responses attained with other algorithms, as shown in Figure 6. However, when the upper overshoots in roll and yaw angles are examined, it is observed that hSA-GWO exhibits a similar upper overshoot as other algorithms. Note that hSA-GWO only performs worse than the other algorithms only in the top overshoot. Overall, it is evident from Figure 6 that hSA-GWO demonstrates superiority over the others.

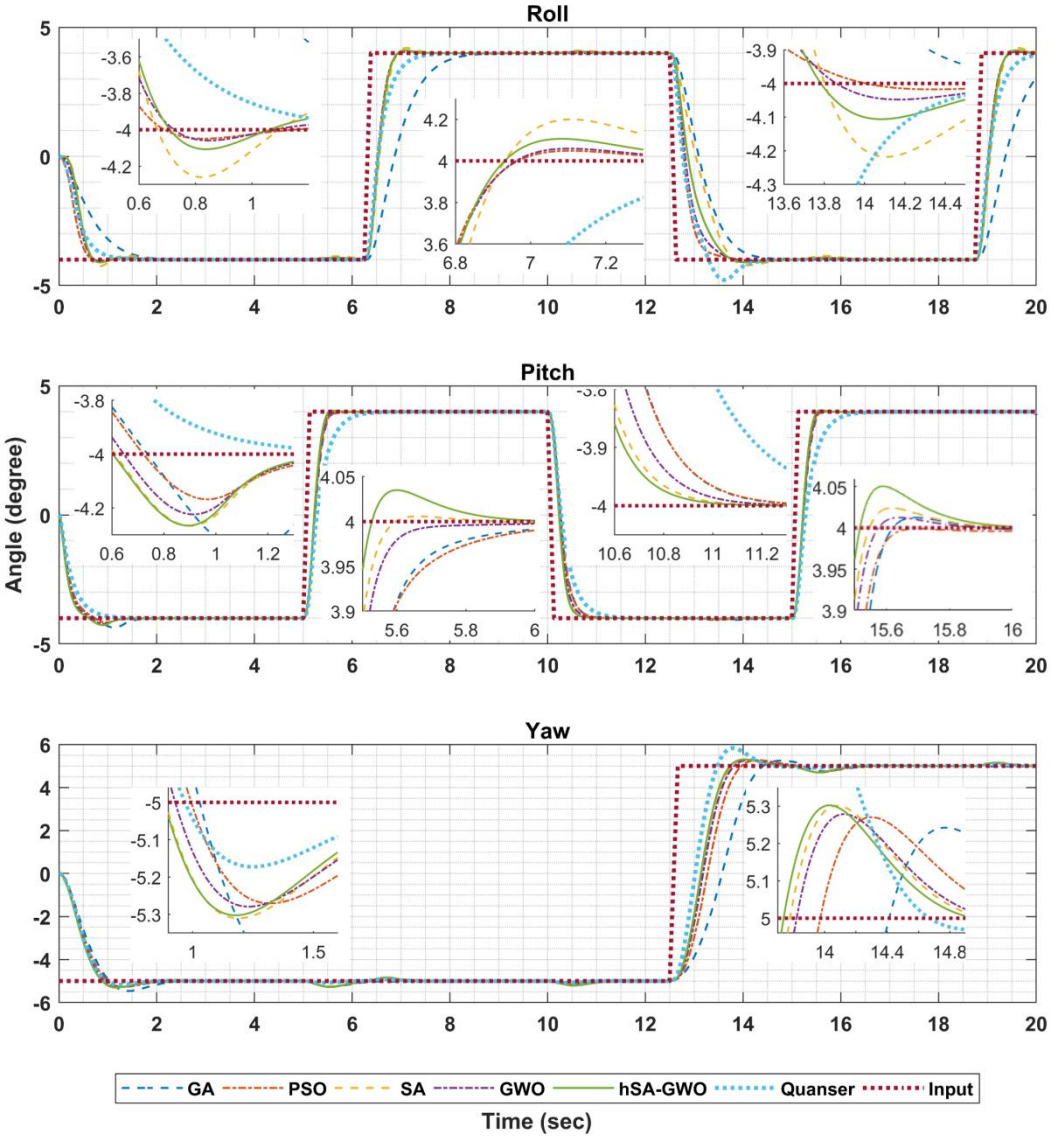

Figure 6. The response of the system to the application of the best obtained solutions.

Table 9 presents the numerical data supporting hSA-GWO superiority, explained in Figure 6 in the previous paragraph. Table 9 provides the settling time and overshoot values for Figure 6. These data are obtained using the stepinfo function in the MATLAB program, which evaluates the system response to a pulse input signal after the last pulse entry. The stepinfo function is a powerful tool used to assess and analyze the behavior of control systems, offering valuable insights into various performance metrics such as settling time, overshoot, rise time, and peak time. The settling time refers to the duration when the system output remains stable within a 2% range, indicating the stabilization duration of system. A low settling time means that the system rapidly approaches the desired setpoint. The stepinfo function also provides a significant measure, the overshoot value, representing the percentage by which the system response exceeds the desired setpoint. Overshoot occurs when the system output temporarily surpasses the target value. This information provides essential guidance for control engineers and system designers in optimizing the system and achieving the desired behavior [19].

For example, the frequency of the roll input signal is 0.08 hertz. One period consists of positive and negative pulses and these pulses last for 6.25 seconds. For the system response obtained with the GWO, the roll output signal reaches the settling time at the 0.6289th second after the last pulse (6.25 * 3 + 0.6289).

As mentioned in the previous sections, early settling time and low overshoot are targeted in this system. Therefore, the data in Table 9 should be evaluated from this perspective.

Table 9. System response data.

	Settling time (sec.)			Overshoot (deg)		
Algorithms	**Roll**	**Pitch**	**Yaw**	**Roll**	**Pitch**	**Yaw**
GA	1.1952	0.5264	3.3231	0.0085	0.3597	0.4649
PSO	0.6245	0.5145	3.3587	0.0487	0.1678	0.2702
SA	0.6427	0.4780	3.2918	0.2602	0.2664	0.3100
GWO	0.6289	0.4894	3.3073	0.0592	0.2238	0.2789
hSA-GWO	0.6228	0.4660	3.2838	0.1070	0.2654	0.3021

When considering settling time, hSA-GWO outperforms other algorithms shown in Table 9. In terms of overshoot, hSA-GWO never ranks last. This phenomenon stems from the nature of dynamic systems. When a dynamic system aims to reach the settling time as rapidly as possible, the overshoot of the system is likely to be higher. Multi-objective optimization problems arise when the desired data for attaining the optimum point conflict with each other. Therefore, it is crucial to consider all objectives and find a balance between them. The main aim is to identify the best parameters to improve the system's performance without compromising its current parameters. In other words, the aim is to strike a balance between different objectives within the system and determine the most suitable parameters.

5. CONCLUSION

This study addresses the problems that arise with the application of LQR to control a complex system with three degrees of freedom. The first and most important of these problems is the formation of ten different controller parameters arising from three degrees of freedom and the need to calculate them. The second difficulty is the impossibility of calculating ten-element optimum solution matrix containing control parameters using mathematical methods. For the system to work optimally, it needs to be tested in real-world conditions; This requires multiple iterations with different sets of ten solutions each, which can be a time-consuming process. This study offers an innovative solution to overcome the mentioned problems.

To make this process possible in a reasonable time, a computer model of the system was used. Within the scope of this study, the system was simulated using randomly generated solution sets within a certain range in the computer environment using optimization algorithms. Weak solution sets were intelligently filtered out and new solution sets were automatically regenerated under certain conditions. As a result, a comprehensive search was made within the specified range and the new solution sets created were repeatedly applied to the system. As a result of the experiments conducted in the simulation environment, it was concluded that the optimization algorithms were superior to each other in different subjects. These findings are shared in detail in the Experimental Results section.

This study, which focuses on the computability of controller parameters with optimization algorithms when LQR is applied to a complex system for control purposes, has shown that this is possible. In addition, considering the results obtained and the work of the algorithms, it was predicted that better results could be obtained by handling the SA and GWO algorithms with a hybrid approach. To test the accuracy of this prediction, the SA and GWO algorithms were hybridized. The resulting hSA-GWO algorithm was applied to the system under equal conditions with other algorithms and the results were presented impartially. As a result, it appears that hSA-GWO has superior performance compared to other algorithms in some aspects.

Declaration of Ethical Standards

This study was conducted in strict accordance with ethical standards and guidelines.

Credit Authorship Contribution Statement

Y.B.: Conceptualization, Investigation, Methodology, Software, Writing – review, Original draft & editing, Design. İ.İ.: Conceptualization, Investigation, Methodology, Software, Writing – review, Supervision, Validation.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Funding / Acknowledgements

This research received no specific grant from any funding agency in the public, commercial, or not-for-profit sectors.

Data Availability

The data that support the findings of this study are available from the corresponding author upon reasonable request.

REFERENCES

[1] S. Mohanty and A. Misra, "3 DOF Autonomous Control Analysis of an Quadcopter Using Artificial Neural Network," *Studies in Computational Intelligence*, vol. 885, pp. 39–57, 2020, doi: 10.1007/978-3-030-38445-6_4/COVER.

[2] "3 DOF Hover- Quanser." https://www.quanser.com/products/3-dof-hover/ (accessed Apr. 17, 2023).

[3] M. K. Bayrakceken and A. Arisoy, "An Educational Setup for Nonlinear Control Systems: Enhancing the Motivation and Learning in a Targeted Curriculum by Experimental Practices [Focus on Education]," *IEEE Control Syst*, vol. 33, no. 2, pp. 64–81, Mar. 2013, doi: 10.1109/MCS.2012.2234971.

[4] Ö. Bayraktar and A. Güldaş, "Quadrotor İtme ve Tork Katsayılarının Optimizasyonu ve Matlab/Simulink ile Simülasyonu," *Politeknik Dergisi*, vol. 23, no. 4, pp. 1197–1204, Dec. 2020, doi: 10.2339/POLITEKNIK.636950.

[5] B. Alagoz, A. Ates, and C. Yeroglu, "Auto-tuning of PID Controller According to Fractional-order Reference Model Approximation for DC Rotor Control," *Mechatronics*, vol. 23, no. 7, pp. 789–797, Oct. 2013, doi: 10.1016/J.MECHATRONICS.2013.05.001.

[6] R. Beard, "*Quadcopter Dynamics and Control Rev*, no. 1, p. 1325, 2008, Accessed: Apr. 17, 2023. [Online]. Available: https://scholarsarchive.byu.edu/facpubhttps://scholarsarchive.byu.edu/facpub/1325

[7] T. Oktay and O. Köse, "Farklı Uçuş Durumları için Quadcopter Dinamik Modeli ve Simulasyonu," *European Journal of Science and Technology*, pp. 132–142, Mar. 2019, doi: 10.31590/EJOSAT.507222.

[8] M. İçen, A. Ateş, and C. Yeroğlu, "Optimization of LQR Weight Matrix to Control Three Degree of Freedom Quadcopter," *IDAP 2017- International Artificial Intelligence and Data Processing Symposium*, Oct. 2017, doi: 10.1109/IDAP.2017.8090164.

[9] V. E. Ömürlü, U. Büyükşahin, R. Artar, A. Kirli, and M. N. Turgut, "An Experimental Stationary Quadrotor with Variable DOF," *Sadhana- Academy Proceedings in Engineering Sciences*, vol. 38, no. 2, pp. 247–264, Apr. 2013, doi: 10.1007/S12046-013-0132-6.

[10] H. K. Tran and T. N. Nguyen, "Flight Motion Controller Design Using Genetic Algorithm for a Quadcopter," *Measurement and Control (United Kingdom)*, vol. 51, no. 3–4, pp. 59–64, Apr. 2018, doi: 10.1177/0020294018768744/ASSET/IMAGES/LARGE/10.1177_0020294018768744-FIG7.JPEG.

[11] A. Reizenstein, "Position and Trajectory Control of a Quadcopter Using PID and LQ Controllers," 2017, Accessed: Apr. 17, 2023. [Online]. Available: http://urn.kb.se/resolve?urn=urn:nbn:se:liu:diva-139498

[12] Demiryurek A, "Modelıng and Control of a Quadrotor," Hacettepe University, Ankara, 2018.

[13] M. Karakoyun and A. Özkış, "Transfer Fonksiyonları Kullanarak İkili Güve-Alev Optimizasyonu Algoritmalarının Geliştirilmesi ve Performanslarının Karşılaştırılması," *Necmettin Erbakan Üniversitesi Fen ve Mühendislik Bilimleri Dergisi*, vol. 3, no. 2, pp. 1–10, 2021.

[14] D. E. Goldberg and J. H. Holland, "Genetic Algorithms and Machine Learning," *Mach Learn*, vol. 3, no. 2, pp. 95–99, 1988, doi: 10.1023/A:1022602019183/METRICS.

[15] J. Kennedy and R. Eberhart, "Particle Swarm Optimization," *Proceedings of ICNN'95- International Conference on Neural Networks*, vol. 4, pp. 1942–1948, doi: 10.1109/ICNN.1995.488968.

[16] S. Kirkpatrick, C. D. Gelatt, and M. P. Vecchi, "Optimization by Simulated Annealing," *Science (1979)*, vol. 220, no. 4598, pp. 671–680, May 1983, doi: 10.1126/SCIENCE.220.4598.671.

[17] S. Mirjalili, S. M. Mirjalili, and A. Lewis, "Grey Wolf Optimizer," *Advances in Engineering Software*, vol. 69, pp. 46–61, Mar. 2014, doi: 10.1016/J.ADVENGSOFT.2013.12.007.

[18] "Linear-Quadratic Regulator (LQR) design- MATLAB lqr." https://www.mathworks.com/help/control/ref/lti.lqr.html (accessed Apr. 18, 2023).

[19] "Rise time, settling time, and other step-response characteristics- MATLAB stepinfo." https://www.mathworks.com/help/control/ref/dynamicsystem.stepinfo.html (accessed Apr. 17, 2023).

[20] İ. İlhan, "An Improved Simulated Annealing Algorithm with Crossover Operator for Capacitated Vehicle Routing Problem," *Swarm Evol Comput*, vol. 64, p. 100911, Jul. 2021, doi: 10.1016/J.SWEVO.2021.100911.

[21] A. Tabak and İ. İlhan, "An Effective Method Based on Simulated Annealing for Automatic Generation Control of Power Systems," *Appl Soft Comput*, vol. 126, p. 109277, Sep. 2022, doi: 10.1016/J.ASOC.2022.109277.

[22] A. Hassanat, K. Almohammadi, E. Alkafaween, E. Abunawas, A. Hammouri, and V. B. S. Prasath, "Choosing Mutation and Crossover Ratios for Genetic Algorithms—A Review with a New Dynamic Approach," *Information 2019, Vol. 10, Page 390*, vol. 10, no. 12, p. 390, Dec. 2019, doi: 10.3390/INFO10120390.

[23] A. K. Peprah, S. K. Appiah, and S. K. Amponsah, "An Optimal Cooling Schedule Using a Simulated Annealing Based Approach," *Appl Math (Irvine)*, vol. 08, no. 08, pp. 1195–1210, Aug. 2017, doi: 10.4236/AM.2017.88090.