A COMPARISON OF GENETIC ALGORITHMS IN OPTIMIZING CONTROLLERS FOR INVERTED PENDULUM

Thanh-Tri-Dai Le*, Thien-An Do, Quoc-Tuan Le, Viet-Cuong Tran, Van-Thuan Le, Tan-Dung Nguyen, Hong-Duc Tran, Quoc-Tuan Tran

Ho Chi Minh city University of Technology and Education (HCMUTE)
Vo Van Ngan St., No. 01, Thu Duc City, Ho Chi Minh City, 700000, Vietnam
*Corresponding author. E-mail: 19151051@student.hcmute.edu.vn

Abstract: In our study, we compared classical genetic algorithm (GA) and non-dominated sorting GA (NSGA) II to optimize LQR and sliding mode control (SMC) for an inverted pendulum. We aimed to identify the algorithm that best improves stability and peak performance. Our results clarified each algorithm's unique strengths. The provement is shown in Matlab simulations.

Keywords: inverted pendulum, LQR control, sliding control, genetic algorithm.

1. Introduction

In the field of automatic control, optimizing the controller plays a vital role in ensuring the performance and stability of the system. In this process, the Genetic Algorithm (GA) has been proven to be a powerful tool for finding optimal parameters [1]. Not long after, Srinivas and Deb (1995) introduced NSGA II, a multi-objective genetic algorithm, considered to be more suitable for multi-objective optimization problems [2] .This method later underwent significant improvements, as evidenced by Deb et al. (2002) who refined NSGA II, enhancing its speed and elitist strategies, further cementing its reputation in the realm of evolutionary computation [3].

Prior to delving into our research, it's important to note that previous studies have already been conducted on the design of controllers [4], performance comparison of controllers [5], and the application of genetic algorithms for optimizing controller parameters [6] , [7]. However, a direct comparison between control algorithms employing genetic techniques in the context of the inverted pendulum model remains a relatively unexplored area. Therefore, we have chosen to delve deeper into this subject in order to provide a comparative analysis of the performance between two genetic optimization algorithms, GA and NSGA II, when applied to optimize controllers for the inverted pendulum swing-up model.

The inverted pendulum swing-up model represents a complex control system, necessitating refined designs to maintain stability and optimal performance. Smith et al. (2000) posed this control problem as a significant challenge, particularly in maintaining stability under all conditions [8].

The objective of this study is to investigate how GA and NSGA II can be employed to optimize LQR (Linear Quadratic Regulator) and SMC (Sliding Mode Control) controllers for the inverted pendulum swing-up model, while ensuring stability as previously mentioned [8]. Through this comparison, we aim to derive deeper insights into the effectiveness of each algorithm and their applicability in real-world control problems.

2. Mathematical Model

According to document [9], the mathematical model used in this research is expressed through the formula below:

$$\mathfrak{A}\begin{bmatrix}\ddot{\alpha}\\ \ddot{\beta}\end{bmatrix}+\mathcal{B}\begin{bmatrix}\dot{\alpha}\\ \dot{\beta}\end{bmatrix}+\mathfrak{D}=\begin{bmatrix}\tau\\ 0\end{bmatrix} \quad (1)$$

We have:

$$\mathfrak{A}=\begin{bmatrix}J_0+m_1L_0^2+m_1l_1^2 sin^2\beta & -m_1L_0l_1\cos\beta\\ -m_1L_0l_1\cos\beta & J_1+m_1l_1^2\end{bmatrix}$$

$$\mathcal{B}=\begin{bmatrix}C_0+\frac{1}{2}m_1l_1^2\dot{\beta}\sin 2\beta & m_1L_0l_1\dot{\beta}\sin\beta+\frac{1}{2}m_1l_1^2\dot{\alpha}\sin 2\beta\\ -\frac{1}{2}m_1l_1^2\dot{\alpha}\sin 2\beta & C_1\end{bmatrix}$$

$$;\ \mathfrak{D}=\begin{bmatrix}0\\ -m_1gl_1\sin\beta\end{bmatrix};$$

From the system of equations (1), we transform it to the form:

$$\begin{cases}\ddot{\alpha}=f_1(\alpha.\dot{\alpha},\beta,\dot{\beta},\tau)\\ \ddot{\beta}=f_2(\alpha.\dot{\alpha},\beta,\dot{\beta},\tau)\end{cases}$$

$$\begin{cases}x_1=\alpha\\ x_2=\dot{\alpha}\\ x_3=\beta\\ x_4=\dot{\beta}\end{cases}\rightarrow(1)=\begin{cases}\dot{x_1}=x_2\\ \dot{x_2}=f_1(x_1,x_2,x_3,x_4,\tau)\\ \dot{x_3}=x_4\\ \dot{x_4}=f_2(x_1,x_2,x_3,x_4,\tau)\end{cases} \quad (2)$$

From the system of equations (2), the linearization around the operating point is: x = x_0= [0 0 0 0]^T. Specifically, it is : x1 = 0 ; x2 =0 ; x3 = 0 ; x4 = 0 ;$t=0$;

$$\Rightarrow\begin{bmatrix}\dot{x_1}\\ \dot{x_2}\\ \dot{x_3}\\ \dot{x_4}\end{bmatrix}=A\begin{bmatrix}x_1\\ x_2\\ x_3\\ x_4\end{bmatrix}+B*t$$

⇒The linear system of equations operating around the equilibrium point is:

$$\dot{x}=Ax+B\tau \quad (3)$$

where:

$$A=\begin{bmatrix} 0 & 1 & 0 & 0 \\ \frac{\partial f_1}{\partial x_1}_{\tau=0;x=x_0} & \frac{\partial f_1}{\partial x_2}_{\tau=0;x=x_0} & \frac{\partial f_1}{\partial x_3}_{\tau=0;x=x_0} & \frac{\partial f_1}{\partial x_4}_{\tau=0;x=x_0} \\ 0 & 0 & 0 & 1 \\ \frac{\partial f_2}{\partial x_1}_{\tau=0;x=x_0} & \frac{\partial f_2}{\partial x_2}_{\tau=0;x=x_0} & \frac{\partial f_2}{\partial x_3}_{\tau=0;x=x_0} & \frac{\partial f_2}{\partial x_4}_{\tau=0;x=x_0} \end{bmatrix}$$

$$; B = \begin{bmatrix} 0 \\ \frac{\partial f_1}{\partial \tau}_{\tau=0;x=x_0} \\ 0 \\ \frac{\partial f_2}{\partial \tau}_{\tau=0;x=x_0} \end{bmatrix};$$

Parameters can be found in the table below:

Table 1: Model Parameter

Symbols	Value	Meaning	Unit
m_1	0.0319	Mass of pendulum	Kg
l_1	0.1572	Length of pendulum	m
L_0	0.137	Distance from arm's pivot point to pendulum's pivot point	m
J_0	0.008591	Moment of inertia of arm	kgm^2
J_1	0.000217	Moment of inertia of pendulum	kgm^2
g	9.81	gravitational acceleration	m/s^2
C_0	0.006408	Friction coefficient of arm's pivot	kgm^2/s
C_1	0.000158	Friction coefficient of pendulum's pivot	kgm^2/s

We compute the matrices A and B using calculations (3) and the model parameters from Table 1

$$A=\begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & -0.7348 & 3.8552 & -0.0124 \\ 0 & 0 & 0 & 1 \\ 0 & -0.5022 & 51.5689 & -0.1656 \end{bmatrix}; \quad (4)$$

$$B=\begin{bmatrix} 0 \\ 114.6758 \\ 0 \\ 78.3676 \end{bmatrix} \quad (5)$$

3. Design of Controller and Application of Gas

3.1. Design of LQR cCntroller

With a system that has a clear mathematical equation, complete system parameters, and a specific, fixed working point, the LQR control algorithm is a common method. With its simple structure, easy computation (thanks to Matlab tools), and simple correction capabilities based on the weight matrix, LQR controller is often suggested for balance robot control. This is also the solution for the system in this paper.

$$J = \int_0^\infty (x^T(t)Qx(t) + u^T(t)Ru(t)) \quad (6)$$

where:

Q: Positive definite matrix (or semi-positive definite)

R: Positive definite matrix

Matrix K is optimized from the Riccati equation in the form:

$$K = R^{-1}B^T P \quad (7)$$

The control law u(t) is computed as:

$$u(t) = R^{-1}B^T Px(t) = Kx(t) \quad (8)$$

where P is the semi-positive definite solution of the Ricatti algebraic equation:

$$PA + A^T P + Q + PBR^{-1}B^T P = \dot{P} \quad (9)$$

The Q matrix represents the control object, the R matrix represents the control signal.

The control law is computed using the function in Matlab as follows:

K = lqr(A,B,Q,R) (10)

3.2. Design of SMC

SMC is a nonlinear control technique widely used due to its stability and excellent noise resistance.

The basic principle of sliding mode control is to bring the system from an initial state to an estimated sliding surface and then keep the system on that surface. When the system is on the sliding surface, it will track a desired trajectory unaffected by disturbance factors.

The steps of designing SMC controller:

➢ **Step 1:** Select sliding mode variable**:**

$$\sigma = \lambda e + \dot{e} \quad (11)$$

where $e = x - x_d$; $\dot{e} = \dot{x} - \dot{x}_d$; λ is a positive constant

Differentiate the sliding mode variable with respect to time:

$$\dot{\sigma} = \lambda\dot{e} + \ddot{e} = \lambda\dot{e} + f(x) + g(x)u + d(t) - \ddot{x}_d \quad (12)$$

➢ **Step 2:** Select sliding mode control law: SMC consists of two terms, balance condition and robustness:

$$u = u_{eq} + u_R \quad (13)$$

The balance term, u_{eq} ,is chosen when we set

d(t)= 0, $\dot{\sigma} = -K\sigma$;

$$u_{eq} = g^{-1}(x)(\ddot{x}_d - \lambda\dot{e} - f(x) - K\sigma) \quad (14)$$

$$u_R = -g^{-1}(x)\eta sign(\sigma)$$

where: $\eta \geq \| d(t) \|_\infty$

➢ **Step 3:** Prove system stability when controlled: Replace sliding mode control (12) with (13) we get:

$$\dot{\sigma} = -K\sigma - \eta sign(\sigma) + d(t) \quad (15)$$

To prove the stability of the controlled system, we choose the Lyapunov function as:

$$V = \frac{1}{2}\sigma^2 \quad (16)$$

Differentiate the Lyapunov function with respect to time:

$$\dot{V} = \sigma\dot{\sigma} = -K\sigma^2 - \sigma(\eta sign(\sigma) - d(t)) \leq -K\sigma^2 \leq 0 \quad (17)$$

3.3. Genetic Algorithm

GA is an optimization approach based on the principle of natural evolution. GA is often applied to solve optimization and search problems. When applying GA to control, the goal is usually to optimize the parameters or design of the controller.

This paper does not aim to present GA. It is only used as a tool to solve the optimization problem, with the aim of finding the optimal parameter set for the proposed controller function.

In closed-loop control of the inverted pendulum with e(t) being the error between the desired signal r(t) and the response signal y(t), then e(t) = r(t) - y(t).

The objective function of the controller tuning process in the article is defined as:

$$J = \sum_{i=1}^{N} e_{1i}(t) + \sum_{i=1}^{N} e_{2i}(t) \quad (18)$$

Thus, the fitness function is expressed as:

Fitness $=\frac{1}{J}$

Trong đó :

e1: Error between the pendulum's angular position and the desired position

e2 : Error between the pendulum's angular velocity and the desired velocity

Parameters of the GA in the article are chosen as follows:

✧ The process goes through evolution over 100 generations (generations=50)

✧ Population size of 100 (populationsize=100)

✧ Crossover frequency 0.8 (CrossoverFraction=0.8)

✧ Mutation probability (MutationFraction=0.8)

Process of finding the optimal value of the GA algorithm is shown in Fig. 1.

3.4. NSGA II algorithm

When GA reaches its limit in multi-objective optimization, NSGA II emerges, opening up a new opportunity for more flexible and efficient controller design. In the intricate world of control engineering, where technical requirements and constraints often conflict with each other, NSGA II not only offers a solution for optimizing individual objectives but also assists engineers in finding the optimal balance between multiple objectives simultaneously.

❖ Compared to the Genetic Algorithm (GA), NSGA II has the following notable advantages:

❖ Multi-objective optimization: The ability to handle multiple objectives simultaneously, while traditional GA often focuses only on a single objective.

❖ Non-relative classification: NSGA II employs a hierarchical system to identify the best solutions based on priority, avoiding the determination of a single optimal solution and offering a range of solutions for users to choose from.

❖ Diversity preservation: Thanks to the crowding distance mechanism, NSGA II maintains diversity within the population, helping the algorithm avoid getting stuck at local minima and enhancing its ability to search for global minima.

❖ High computational performance: Faster and more efficient optimization in complex problems.

In the following section, we will focus on constructing objective functions to evaluate performance.

$$J1= \sum_{i=1}^{N} e_{1i}(t) + \sum_{i=1}^{N} e_{2i}(t) \quad (19)$$

$$J2= \sum_{i=1}^{N} e_{1i}(t) * e_{3i}(t) + \sum_{i=1}^{N} e_{2i}(t) * e_{4i}(t) \quad (20)$$

Specifically, the values e1, e2, e3, e4 represent the errors between the desired values corresponding to each value (pendulum angle position, pendulum speed, arm angle position, wing speed).

The parameters in the NSGA II algorithm are chosen similarly to those in GA (20), to ensure that the comparison between them is carried out in the most optimal way.

Process of finding the optimal value of the NSGA II algorithm in Fig. 2.

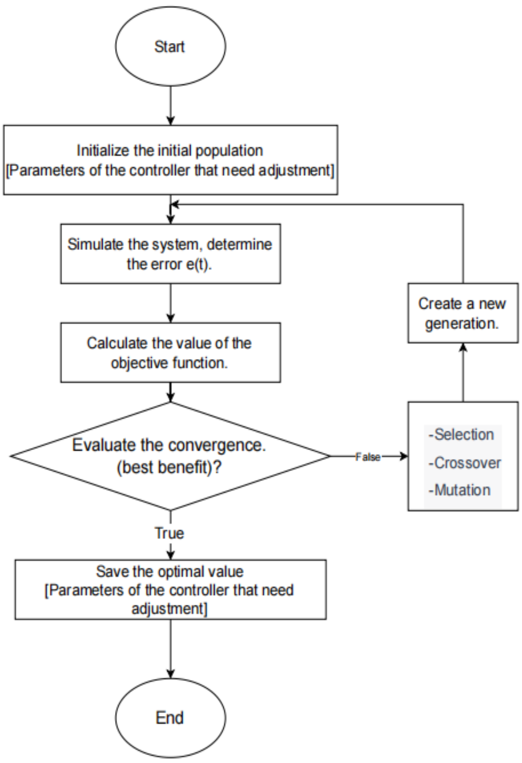

Fig. 1. The operational process of GA

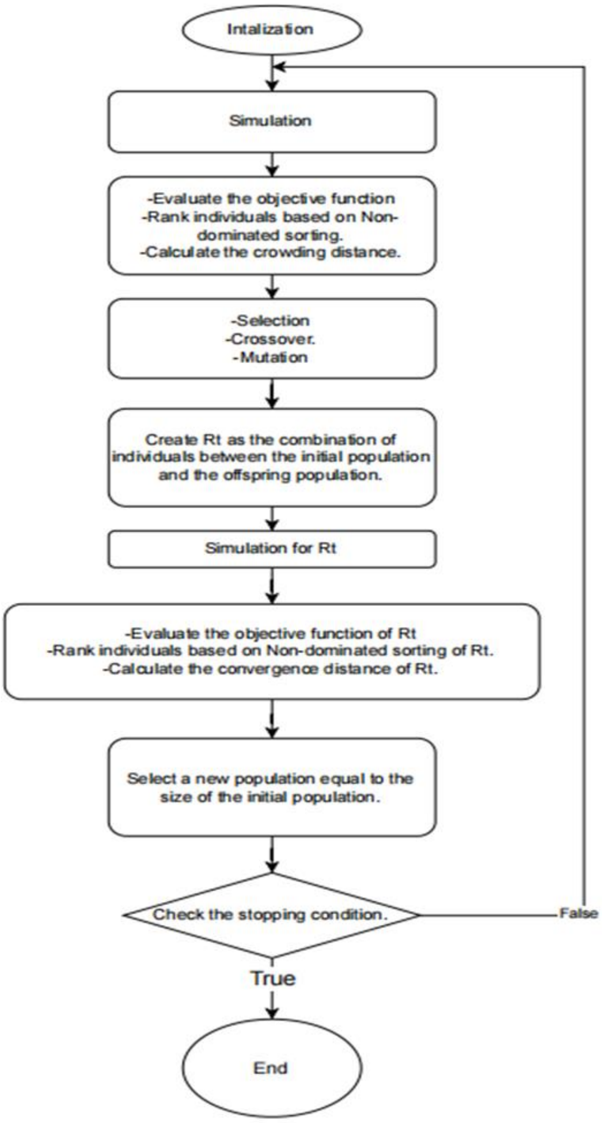

Fig. 2. The operational process of NSGA II

4. Results and Simulation

4.1. Selection and Value Determination of GA and NSGA II for LQR and SMC

Implemented through custom coding in MATLAB, we carried out the selection and search process autonomously, without depending on pre-existing tools:

4.1.1 Results for the GA Search and Selection Pertaining to LQR and SMC:

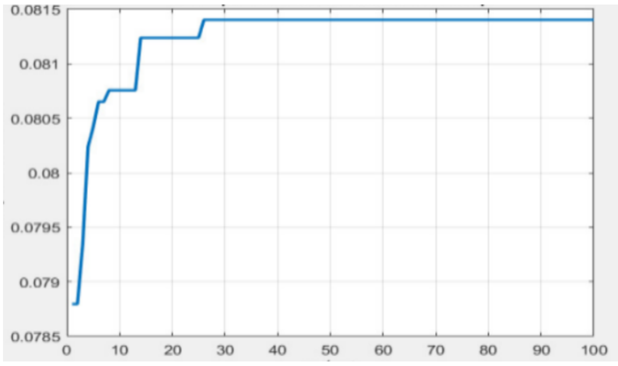

Fig. 3. Fitness value after 100 generations for GA of the LQR controller

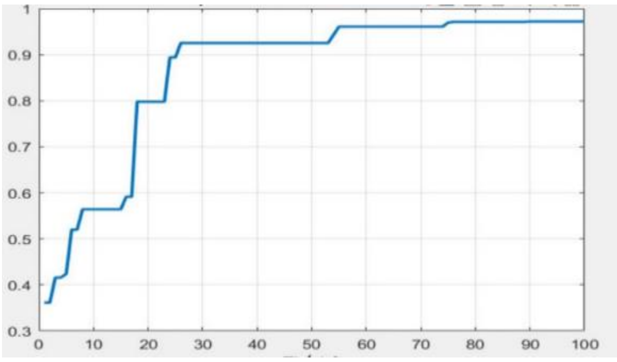

Fig. 4. Fitness value after 100 generations for the GA of the SMC controller

The GA value of matrix Q at the 100th generation was identified and selected based on **Fig. 3**.

Here: the x-axis represents the generation number, and the y-axis represents the Fitness value

$$Q=\begin{pmatrix} 11.6611 & 0 & 0 & 0 \\ 0 & 0.0808 & 0 & 0 \\ 0 & 0 & 0.6575 & 0 \\ 0 & 0 & 0 & 0.4195 \end{pmatrix}$$

The values of the parameter set (K, lamda, eta) from the 100th generation were identified and selected based on **Fig. 4**, with values: K = 63.0146; lamda = 226.4352; eta = 48.4612.

4.1.2 Results of NSGA II Search for LQR and SMC

4.1.2.1 Search and Selection of Values from the Pareto Set for SMC

The results of searching for values in the Pareto front of the NSGA II algorithm over 100 generations for the SMC controller applied to the inverted pendulum:

The value sets (K, lamda, eta) with the same rank 1 that were identified are:

1) K= 70.7617 ; lamda= 51.8097 ; eta = 50.7727;
2) K1 =70.4424; lamda1= 51.7532 ; eta1= 50.7683;
3) K2 =70.8452; lamda2= 51.8112; eta2 = 50.7727;

Observation: Since the search process yielded values that are quite similar, we chose one value from the three to carry out the comparison and evaluation.

The selected value is: K = 70.7617; lamda = 51.8097; eta = 50.7727

4.1.2.2 Search and selection of values from the Pareto set for NSGA II

Table 2: Table of Q Values with the Same Top Rank

Order number	Value	F1 value	F2 value	Crowding Distance
1	$\begin{bmatrix} 26.0389 & 0 & 0 & 0 \\ 0 & 465.6007 & 0 & 0 \\ 0 & 0 & 364.3308 & 0 \\ 0 & 0 & 0 & 368.9208 \end{bmatrix}$	80.8399	2.337	0.6889
2	$\begin{bmatrix} 31.703 & 0 & 0 & 0 \\ 0 & 430.2203 & 0 & 0 \\ 0 & 0 & 467.2026 & 0 \\ 0 & 0 & 0 & 492.1992 \end{bmatrix}$	80.0483	2.4351	0.2985
3	$\begin{bmatrix} 6.6416 & 0 & 0 & 0 \\ 0 & 448.5957 & 0 & 0 \\ 0 & 0 & 98.3291 & 0 \\ 0 & 0 & 0 & 46.6853 \end{bmatrix}$	85.5113	1.7491	Inf
4	$\begin{bmatrix} 23.1756 & 0 & 0 & 0 \\ 0 & 252.7141 & 0 & 0 \\ 0 & 0 & 380.7129 & 0 \\ 0 & 0 & 0 & 315.5350 \end{bmatrix}$	77.8714	2.542	0.8176
5	$\begin{bmatrix} 44.9458 & 0 & 0 & 0 \\ 0 & 40.4312 & 0 & 0 \\ 0 & 0 & 388.6203 & 0 \\ 0 & 0 & 0 & 452.5674 \end{bmatrix}$	71.0937	3.9621	Inf
6	$\begin{bmatrix} 63.944 & 0 & 0 & 0 \\ 0 & 274.77 & 0 & 0 \\ 0 & 0 & 242.61 & 0 \\ 0 & 0 & 0 & 445.237 \end{bmatrix}$	77.4517	3.8459	1.1118

In this case, the selected individual is the individual with the order number 6, as individual 6 has the highest crowding distance value based on the table 2 below.

4.2 Comparison and Simulation

In this simulation section: we will select initial values for the simulation process as follows:

$$x_0 = [0.7 \;\; 0.3 \;\; 0.01 \; 0]^T$$

The simulation model parameters are presented in Table 1.

The control parameter values selected for simulation are:

R =1

$$\text{Q-GA} = \begin{pmatrix} 11.6611 & 0 & 0 & 0 \\ 0 & 0.0808 & 0 & 0 \\ 0 & 0 & 0.6575 & 0 \\ 0 & 0 & 0 & 0.4195 \end{pmatrix}$$

Q - NSGA II

$$= \begin{pmatrix} 63.9442 & 0 & 0 & 0 \\ 0 & 274.7701 & 0 & 0 \\ 0 & 0 & 242.6147 & 0 \\ 0 & 0 & 0 & 445.2378 \end{pmatrix}$$

The GA parameter set for SMC is: K = 63.0146; lamda = 226.4352; eta = 48.4612.

The NSGA II algorithm parameter set for SMC is: K = 70.7617; lamda = 51.8097; eta = 50.7727.

4.2.1 Comparison between GA and NSGA II for LQR.

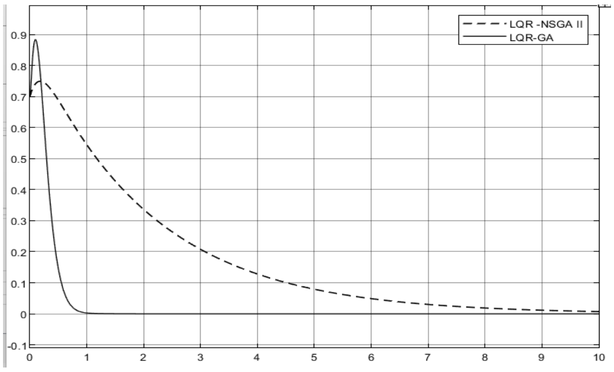

Fig. 5. Position of the Pendulum (rad) through time (sec)

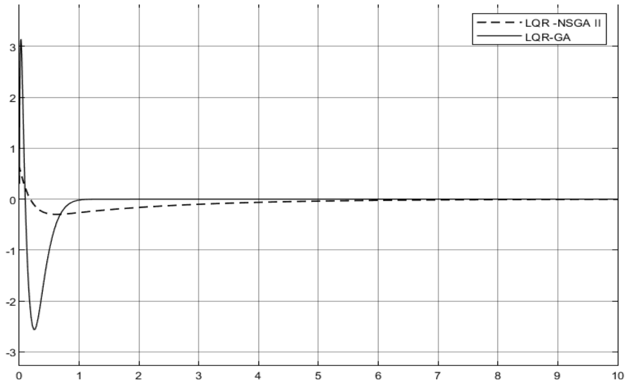

Fig. 6. Velocity of the Pendulum (rad/s) through time (sec).

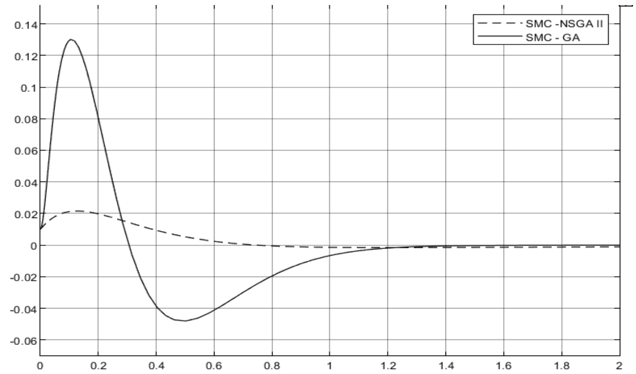

Fig. 7. Position of the Arm (rad) through time (sec)

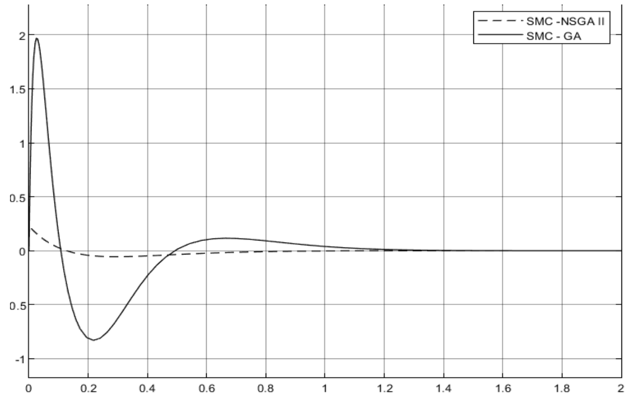

Fig. 8. Velocity of the Arm (rad/s) through time (sec)

Observations:

- ❖ Pendulum's settling time: The settling time for the LQR NSGA II is 9s, which is 1s slower than the LQR-GA, as observed.
- ❖ Pendulum's overshoot angle: In Fig. 5, we detected that pendulum's overshoot angle using the GA algorithm reaches 0.87, higher than 0.75 of NSGA II.
- ❖ Oscillations and amplitude: As illustrated in Fig. 6, Fig. 7, and Fig. 8, GA produces more oscillations and with higher amplitudes compared to NSGA II. These oscillations might cause issues such as Force Overload or wear and tear on the hardware when the speed is too high.

4.2.2 Comparison between GA and NSGA II for SMC

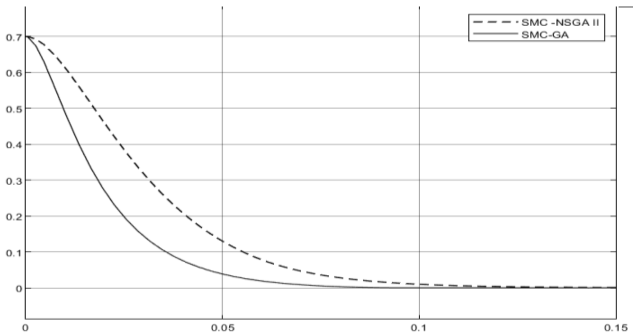

Fig. 9. Position of the Pendulum (rad) through time (sec)

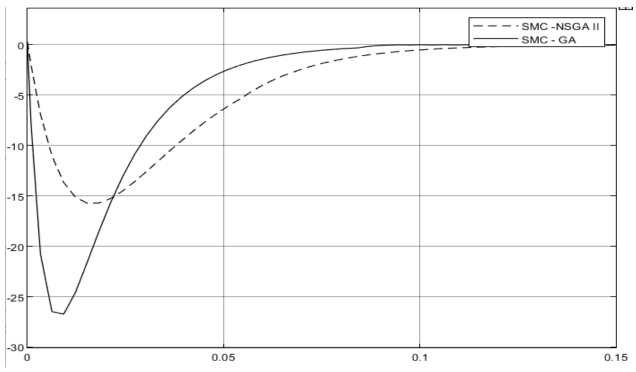

Fig. 10. Velocity of Pendulum (rad/s) through time (sec)

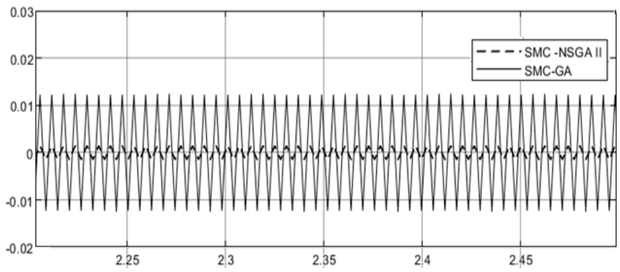

Fig. 11. Chattering phenomenon occurs in the velocity behavior of pendulum (rad/s) through time (sec)

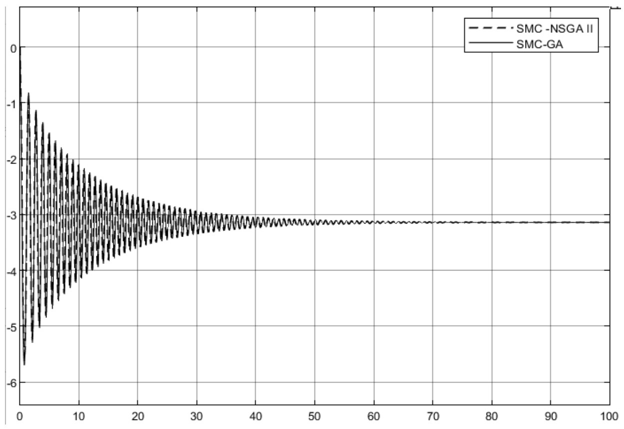

Fig. 12. Position of the arm (rad) through time (sec)

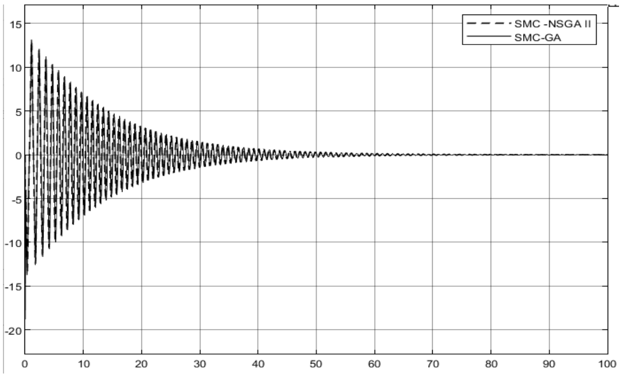

Fig. 13.Velocity of arm(rad/s) through time (sec)

Observations:

- Convergence Speed: Both algorithms demonstrate rapid convergence towards the position of the pendulum, as clearly illustrated in Fig. 9.
- Chattering Phenomenon: Regarding GA, we have detected the chattering phenomenon in Fig. 11 , which has a higher oscillation amplitude compared to NSGA II, indicating a significant difference in the operation of the two algorithms.
- Angular Position and Arm Velocity: In general, the angular position of the arm for SMC-GA and SMC-NSGA II closely follows each other, as depicted in Fig. 12 and Fig. 13. However, when considering arm velocity, we observed that the initial value for SMC-GA fluctuates more compared to NSGA II.

5. Conclusion

In this paper, we conducted a comparative study between two optimization algorithms, GA and NSGA II, applied for optimizing the control parameters of two LQR and SMC controllers in a simple SISO model.

The results have clarified that, in the context of single-objective optimization, GA outperforms NSGA II. However, NSGA II, with its focus on multi-objective optimization, presents a significant challenge. It needs to balance between objectives, sometimes meaning certain trade-offs are required to achieve comprehensive results.

Although NSGA II hasn't achieved as high performance as GA in single-objective optimization, it introduces an exciting and critical area of research, namely stability and flexibility in handling multiple objectives. This challenge not only enriches optimization theory but also poses intriguing questions about the design and implementation of effective multi-objective optimization algorithms.

The paper provides an in-depth look at both algorithms, highlighting the challenges and opportunities in the field of optimization and automatic control. These findings will become a valuable inspiration for subsequent studies, aiming to better harness the strengths of both algorithms in practical applications.

Acknowledgement

We want to give thanks to PhD. Van-Dong-Hai Nguyen (HCMUTE) due to his support in giving ideas for this study.

Reference

[1] Goldberg D.E.: "Genetic Algorithms in Search, Optimization, and Machine Learning”, Addison-Wesley, 1989.
[2] Srinivas N., Deb K.: "Multi objective optimization using non-dominated sorting in genetic algorithms”, Evolutionary computation, 2(3), 221-248, 1995.
[3] Deb K., Pratap A., Agarwal S., Meyarivan T.: "A fast and elitist multiobjective genetic algorithm: NSGA-II", IEEE Transactions on Evolutionary Computation, 6(2), pp. 182–197, 2002.
[4] Setiawan N., Pratama G.N.P.: "Application of LQR Full-State Feedback Controller for Rotational Inverted Pendulum”, Journal of Physics: Conference Series, 2111(1), 012006, pp. 3-7, 2021.
[5] Villacres J. et al.: "Controllers comparison to stabilize a two-wheeled inverted pendulum: PID, LQR and sliding mode control”, International Journal of Control Systems and Robotics, 1, p31-p35, 2016.
[6] Wongsathan C., Sirima C.: "Application of GA to design LQR controller for an inverted pendulum system", IEEE International Conference on Robotics and Biomimetics. 951-954, 2009.
[7] Ou C., Lin W.: "Comparison between PSO and GA for Parameters Optimization of PID Controller", The Faculty of Information Science and Technology, University of NingBo, pp. 2471-2475, 2006.
[8] Smith R., Jones D., Lewis F.: "Properties and performance of inverted pendulum models. In Proceedings of the American Control Conference (pp. 500-504), 2000.
[9] Huynh T.H.: "Intelligent Control Systems (He thong dieu khien thong minh in Vietnamese)", Ho Chi Minh City, Vietnam, pp. 227-228, 2006.