RESEARCH ARTICLE

Design of Cardiac Pacemaker Controller Based on Reinforcement Learning

Kağan Orbay [a], **Mehmet Sağbaş** [a†], **Murat Demir** [a]

[a] Department of Electrical and Electronics Engineering, İzmir Bakırçay University, İzmir, Türkiye
[†] sagbasm@gmail.com, corresponding author

RECEIVED DECEMBER 23, 2024
ACCEPTED APRIL 9, 2025

CITATION Orbay, K., Sağbaş, M. & Demir, M. (2025). Design of cardiac pacemaker controller based on reinforcement learning. *Artificial Intelligence Theory and Applications, 5(1)*, 29-41.

Abstract

This study investigates the derivation of PID controller parameters, commonly used for pacemaker control, using both genetic algorithm (GA) and reinforcement learning (RL) methods. We compare the PID parameters obtained by RL with those obtained by GA, a well-known and often preferred method in literature. The aim of the study is to analyze the performance of the control parameters obtained by both methods and to determine which approach is more effective in pacemaker applications. In particular, comparisons on important control criteria such as settling time, rise time and overshoot of the system will reveal the advantages and disadvantages of these methods.

Keywords: heart rhythm regulation, pacemaker control system, PID controller optimization, reinforcement learning

1. Introduction

Cardiovascular diseases, including heart attacks and arrhythmias, are among the leading causes of death worldwide [1-2]. Arrhythmias disrupt the normal electrical activity of the heart and often require medical intervention. One of the most effective solutions for regulating heart rhythm is the cardiac pacemaker, which delivers controlled electrical impulses to the heart [3]. Pacemakers continuously monitor cardiac activity and correct irregularities by providing appropriate electrical stimulation [4]. This regulation is crucial for preventing complications that can arise from untreated arrhythmias, such as stroke or heart failure. Additionally, advancements in technology have led to the development of more sophisticated pacemakers that can adapt to a patient's activity level, further improving overall cardiac health.

To improve the efficiency of pacemakers, researchers have developed advanced control strategies to optimize their performance. A key aspect of this optimization is the use of Proportional-Integral-Derivative (PID) controllers, which provide precise regulation of the heart rhythm. The PID controller has three components. The proportional (P) component provides a correction proportional to the current error, allowing a fast and accurate response to instantaneous changes in heart rate. This accuracy ensures that the heart rate is maintained at the desired level. The integral (I) component accounts for the

Artificial Intelligence Theory and Applications, ISSN: 2757-9778. ISBN: 978-605-69730-2-4 © 2025 İzmir Bakırçay University

accumulation of error over time and provides long-term corrections. This prevents the accumulation of continuous errors and helps the pacemaker maintain a more stable heart rhythm over time. Provides long-term performance improvements. The derivative (D) component analyzes the rate of change of the error and reacts quickly to instantaneous changes. This quickly compensates for sudden changes in heart rate and prevents the system from overreacting. It improves overall system performance by adapting to dynamic changes. Proper tuning of these parameters is critical to achive the desired control system behavior. However, determining optimal PID parameters remains a challenge, leading researchers to explore advanced optimization techniques such as Genetic Algorithms (GA) and Reinforcement Learning (RL) [5-6].

Several mathematical models have been proposed to describe cardiac dynamics, which are crucial for developing pacemaker control strategies. Biswas et al. (2006) modeled the cardiovascular system using a closed-loop negative unit feedback system based on transfer functions [7]. Additionally, mathematical models such as the Noble model for Purkinje fibers and the Beeler-Reuter model for ventricular myocardial cells have been widely used to simulate cardiac activity [8-9].

This study investigates the effectiveness of GA and RL in optimizing pacemaker PID controller parameters. By comparing these approaches, we aim to identify the most efficient method based on performance criteria such as settling time, rise time, and overshoot. The results provide insights into the advantages and limitations of AI-driven optimization strategies in biomedical control applications.

Traditional PID tuning methods, such as Ziegler-Nichols and Cohen-Coon, are widely used but often struggle with adaptability in dynamic physiological conditions [10-11]. To overcome this limitation, evolutionary algorithms and machine learning-based optimization techniques have been explored.

Various control techniques have been explored to improve the performance of pacemakers. Apart from classical methods, the following approaches have been utilized:

- Studies using optimization algorithms [2], [12-13]
- Embedded designs using microcontrollers and FPGAs [14-15]
- Machine learning based designs using various machine learning algorithms [16-17]
- Studies using analogue circuits [18-19]

Several studies have demonstrated the effectiveness of Genetic Algorithms (GA) in optimizing PID controllers for pacemakers. Bajpai et al. (2017) showed that GA-based tuning minimizes overshoot and improves transient response [2]. Similarly, Momani et al. (2019) examined fractional-order PID controllers tuned via GA and found improved accuracy in heart rate regulation [4]. These findings highlight GA’s ability to efficiently explore solution spaces and optimize control parameters.

However, GA has limitations [20]:

- It relies on heuristic search mechanisms that may converge to local optima.
- Its performance is highly dependent on mutation and crossover rates.
- It does not adapt well to real-time physiological variations.

In contrast, Reinforcement Learning (RL) has gained attention for adaptive control systems [21-22]. In contrast to GA, RL continuously learns from the environment, thereby

improving decision-making over time [5]. Lima et al. (2023) applied RL to cardiac rhythm regulation, demonstrating its ability to dynamically adjust pacing parameters with high accuracy [16].

Despite its advantages, RL also has challenges [23-24]:

- It requires extensive training episodes to achieve convergence.
- Traditional RL methods struggle in high-dimensional continuous spaces.
- Computational complexity can be high, requiring deep RL techniques for scalability.

Although GA and RL have been studied separately, a direct comparative analysis of these methods in pacemaker control is still lacking. This study aims to bridge this gap by evaluating GA and RL in optimizing PID parameters for pacemaker applications. The key contributions are:

1. A comparative analysis of GA and RL for PID tuning in pacemakers, assessing their effectiveness in optimizing heart rhythm control.

2. A structured performance evaluation based on key control metrics (settling time, rise time, overshoot, and peak response).

3. An RL-based adaptive tuning framework, demonstrating its potential advantages over GA in reducing overshoot and improving stability.

4. A scalable optimization methodology that can be extended to other AI techniques such as Particle Swarm Optimization (PSO) and Model Predictive Control (MPC).

By integrating modern AI-driven techniques with traditional evolutionary algorithms, this study provides a novel perspective on cardiac pacemaker controller design. The findings of this study suggest that GA is more effective in achieving rapid responses, while RL offers superior long-term adaptability, making it a promising solution for real-world applications.

2. Modelling of Pacemaker

Mathematical models of the heart have been developed to facilitate understanding of cardiac function. Noble described the Purkinje fiber cell action potential in 1962 with the Noble model [8]. Beeler and Reuter introduced an electrical activity model of the ventricular myocardial cell in 1977 [9]. The mathematics of cardiac dynamics helped to design pacemaker control systems for artificial and implanted devices. Biswas et al. proposed a transfer function-based cardiovascular system mathematical model [7]. The cardiovascular system is depicted as closed loop negative unit feedback with filter and controller. Figure 1 depicts the cardiovascular closed-loop control system block diagram. Equations. (1) and (2) provide the pacemaker and heart transfer functions $G_{Pacemaker}(s)$ and $G_{heart}(s)$, for the configuration depicted in Figure 1. The closed loop system receives the real heart rate $R(s)$ and produces the target heart rate $Y(s)$. The function of $G_K(s)$ is to serve as the controller.

$$\boldsymbol{G_{Pacemaker}(s) = \frac{\omega_{lpf}}{s+\omega_{lpf}}} \quad (1)$$

$$\boldsymbol{G_{Heart}(s) = \frac{1}{Ms^2+Bs+K}} \quad (2)$$

The cut-off frequency of the low-pass filter representing the pacemaker is ω_{lpf}, while the mass of the heart muscle is M, the viscous drag of the heart myocardial cell is B, and the torsional drag is K.

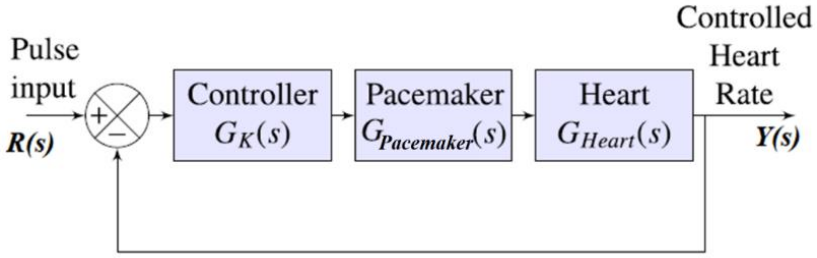

Figure 1. The block diagram of the cardiovascular system.

If the numerical values of the parameters given in Eqs. (1) and (2) are substituted in the studies in the literature, $G_P(s)$ is obtained as follows [7].

$$G_P(s) = \frac{1352}{s(s+8)(s+20.8)} \quad (3)$$

Given the closed loop system shown in Figure 1, its closed loop transfer function is as follows:

$$(s) = \frac{Y(s)}{R(s)} = \frac{G_K(s)G_P(s)}{1+G_K(s)G_P(s)} \quad (4)$$

3. PID Controllers

PID controllers are used to improve pacemaker efficiency. The PID controller improves pacemaker performance and cardiac rhythm management. The PID controller can adjust the pacemaker's output to target heart rate for accurate cardiac rhythm regulation. To respond quickly and accurately to immediate heart rate changes, the proportional (P) component corrects the present mistake. This accuracy keeps the heart rate at the correct level. Long-term error fixes were provided via the integral (I) component. Avoiding ongoing mistakes helps the pacemaker maintain a steady cardiac rhythm over time. Improves long-term performance. The derivative (D) component evaluates error rate and reacts swiftly to sudden changes. This swiftly adjusts for unexpected heart rate variations and minimizes overreaction. It adapts to dynamic changes to boost system performance. PID controller settings can adjust to patient activity and physiological changes. This adjustment allows the pacemaker to automatically modify heart rate based on stress or physical activity, enhancing performance.

The PID controller has three parameters: The error signal's current value determines K_p. Control action rises proportionately with mistake. The error signal's historical values determine the integral term (K_i). Long-term errors activate the integral term, boosting control action. The derivative term (K_d) predicts fault signal value. The derivative term increases control action if the mistake is rising fast. These three parameters are crucial to PID controller performance. To accomplish control system behavior, these parameters must be tuned properly depending on the application.

Equations (5) and (6) respectively provide the time-based response of the output u(t) and transfer function of the PID-controller.

$$u(t) = K_p e(t) + T_i \int_0^t e(\tau)d\tau + T_d \frac{d}{dt} e(t), \quad (5)$$

$$G_K(s) = \frac{U(s)}{E(s)} = K_p + \frac{T_i}{s} + T_d s \quad (6)$$

where E(s) and U(s) denote the Laplace transforms of the error and control signals, respectively.

4. Genetic Algorithm

Evolutionary search and optimization methods like genetic algorithms address difficult optimization issues. These algorithms replicate natural selection, crossover, and mutation to efficiently explore the solution space. Genetic algorithms analyse and choose the best potential solutions from a population. Each cycle selects the best people and assesses them using a fitness function [6, 25]. Crossover and mutation procedures expand solution space while retaining population genetic diversity. Traditional approaches fail to solve difficult and large-scale optimization issues, but this method can. Genetic algorithms' success depends on parameter values and problem-specific design.

$$\mathrm{C} = \{C_1, C_2, \dots, C_n\} \quad (7)$$

In Eq. (8), *C* represents a chromosome, and C_i represents the *i*-th gene of the chromosome.

$$f(C) = \text{Fitness Function } (C) \quad (8)$$

The fitness function in Eq. (8) determines chromosomal (solution) quality. This function represents the optimization goal. The objective is usually to maximize fitness. Selection ensures that the following generation inherits the finest population members.

$$P_i = \frac{f(C_i)}{\sum_{j=1}^{N} f(C_i)} \quad (9)$$

P_i is the probability of selection of the *i*-th individual, and $f(C_i)$ is its fitness value. Crossover creates a new person from two parental chromosomes. Single-point crossover is the most frequent crossover mechanism. Single-point crossover is a typical way to make new people from two parental chromosomes. This approach switches genes on the two chromosomes from a point. Mutations affect the value of a randomly selected gene to preserve genetic variation

5. Reinforcement Learning

Machine learning refers to algorithms that acquire knowledge from data. The domains of machine learning encompass supervised, unsupervised, and reinforcement learning. RL is extensively utilized in supervised issues because of its reward-based learning framework.

RL simply works to develop a function that produces an output based on feedback obtained from the environment utilizing data. Upon structural analysis, it is evident that it has three fundamental components: Agent, State, and Environment [26]. The learned function is referred to as policy. The policy permits the choice of the action (at) that will provide the maximum reward over time, based on the observed state [5]. The strategy

may be stochastic or deterministic, contingent upon the chosen RL approach. Figure 2 depicts the overarching framework for training in RL.

This research utilizes the Q-learning algorithm, a traditional approach in RL. Q-learning is a fundamental RL method designed for systems with a discrete solution space. The Q-learning approach is especially appropriate for systems characterized by discrete action and state spaces, and in this research, it is employed to optimize the PID parameters.

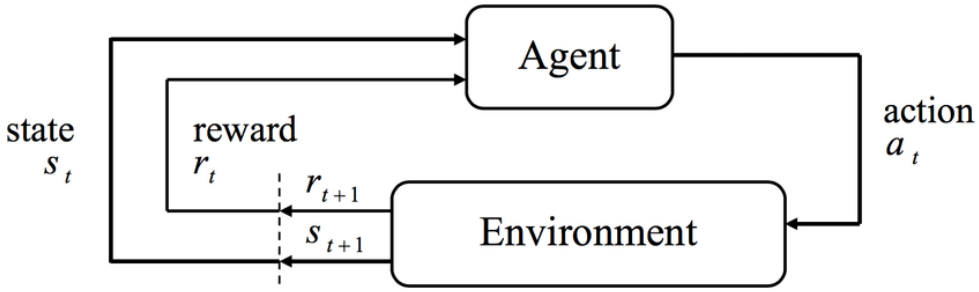

Figure 2. The reinforcement learning training schematic.

Q-learning is a model-free technique designed for RL tasks characterized by discrete state and action spaces. The agent determines the appropriate action to select in each environmental state. At each stage, the agent selects an action, obtains a reward for that action, and transitions to the subsequent state. The agent aims to select activities that optimize the cumulative reward over time. Q-learning creates a table that allocates a Q-value to each state-action combination and subsequently modifies this table to enhance learning. Acts are selected using the Epsilon-Greedy technique, whereby the agent occasionally engages in random acts for exploration and at other times opts for the action deemed optimal based on the existing Q-values. The updates consider the prospective benefits of each action, as dictated by the Bellman Equation. Consequently, the agent discerns the action that yields the maximum reward in each scenario.

6. Simulation Results and Discussion

Genetic Algorithm Optimization was first used to select the parameters of the PID controller used to control the pacemaker. The transfer functions in Equation (4) are used for the heart and pacemaker. These transfer functions model the dynamic characteristics of the heart and pacemaker systems.

The objective function of the GA is to measure the performance of the unit step response of the system, representing a heart rate of 72 bpm, by determining the parameters (K_p, K_i, K_d) of the PID controller. 72 bpm represents the healthy heart rate of an average person, and the system aims to approach this reference value with the fastest and least oscillation. By punishing changes in system response, rise time, settling time, overshoot, and peaks, the objective function tries to minimize the total error for all possible combinations of controller parameters.

The GA optimization process involves defining lower and upper bounds for the parameters of the PID controller and configuring the algorithm's operating parameters. Table 1. shows the selected parameters. We also chose a Gaussian mutation function for mutation and enabled parallel processing to speed up the calculations.

Table 1. The parameter used in simulations for GA

Parameter	Genetic Algorithm (GA)
Population Size	100
Number of Generations	200
Mutation	Gaussian Mutation
Crossover Rate	90%

To properly tune controllers and evaluate their performance, one can consider several performance criteria. The performance criteria used in this study are Integral Square Error (ISE), Integral Time Absolute Error (ITAE), Integral Time Square Error (ITSE), Integral Absolute Error (IAE), and the Discrete Time Integral Sample Based Double Square Error (dTISDSE). The following equations illustrate how these performance criteria are calculated.

$$ISE(e) = \int_0^\infty e^2(t)\,dt \tag{10a}$$

$$ITAE(e) = \int_0^\infty t|e(t)|\,dt \tag{10b}$$

$$ITSE(e) = \int_0^\infty te^2(t)\,dt \tag{10c}$$

$$IAE(e) = \int_0^\infty |e(t)|\,dt \tag{10d}$$

$$dTISDSE(e) = \sum_{k=1}^{n} k(e_k^2)^2 \tag{10e}$$

Figure 3 displays the step response of the closed-loop system resulting from GA optimization for various performance criteria. Table 2 provides the PID parameter values obtained for all performance criteria.

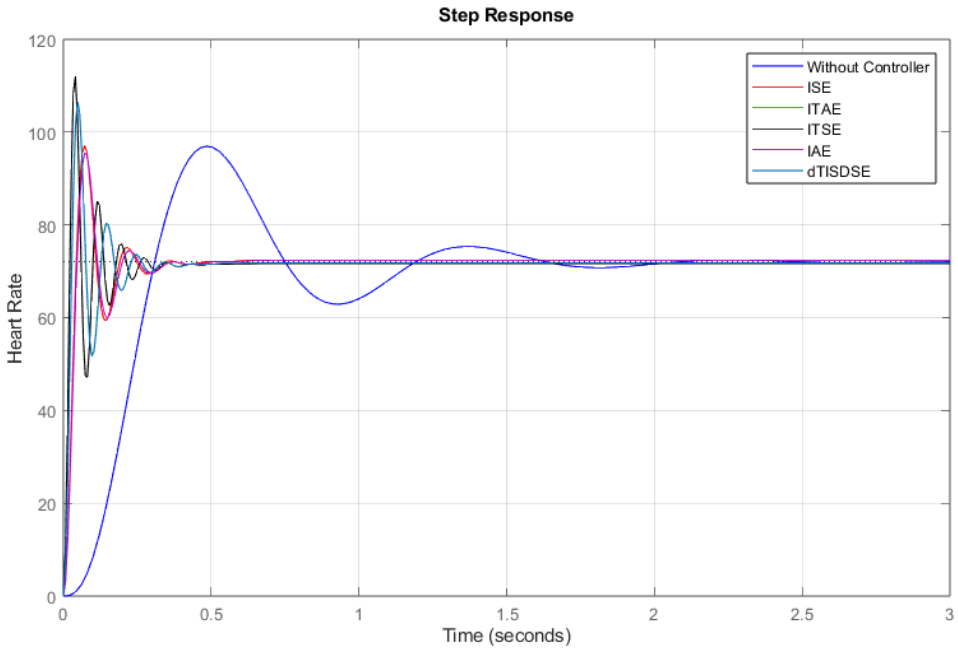

Figure 3. PID-controller responses for various performance criteria

Table 2. PID parameter values obtained using different error functions

Error Functions	K_p	K_i	K_d
ISE	4.779060	3.148517	1.487819
ITAE	6.437423	3.612167	3.090703
ITSE	7.601399	2.917295	4.871950
IAE	4.266956	2.623989	1.387273
dTISDSE	6.419923	3.476024	3.076429

Figure 4 displays the step response for ISE, yielding the best result among the used performance criteria. Table 3 shows the performance metrics for the step response obtained for ISE. Figures 3 and 4 demonstrate that the controlled system, utilizing PID parameters from the genetic algorithm, achieved the target heart rate of 72 bpm more quickly and with fewer oscillations than the uncontrolled system. As can be seen from Table 3, significant improvements are observed, especially in performance criteria such as response time, settling time, overshoot, and peak values.

Table 3. GA Optimization results for ISE performance criteria

Performance Metrics	Controlled System	Uncontrolled Closed-loop System Step Response
Rise Time (s)	0.035707	0.1908
Settling Time (s)	0.342283	1.5414
Overshoot (%)	28.820775	34.6568
Peak	92.750958	96.9529

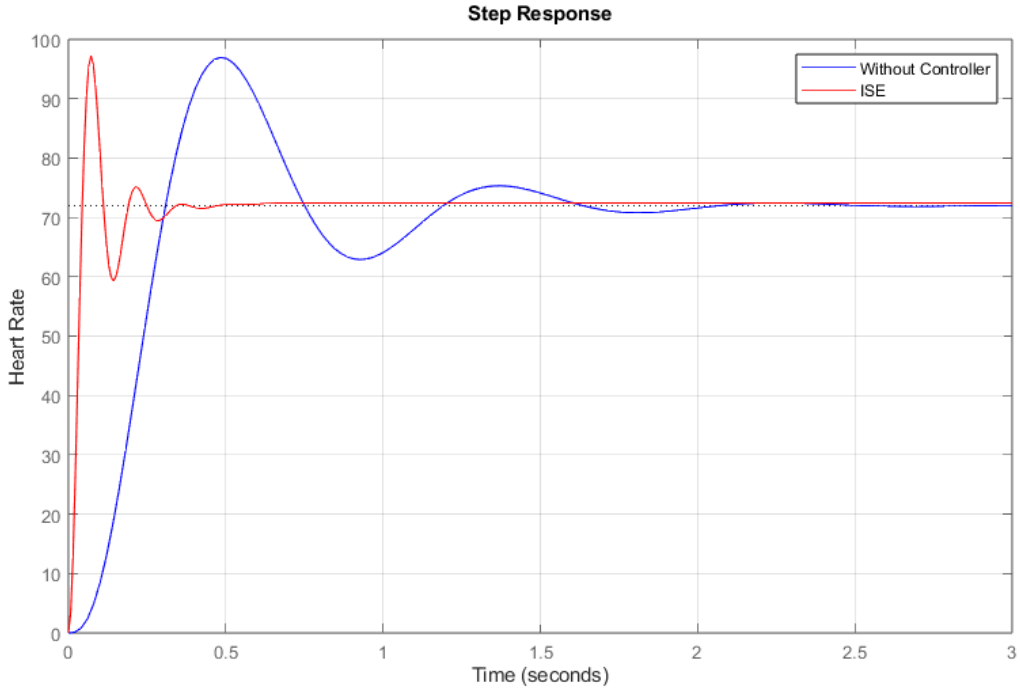

Figure 4. The step responses of the controlled system for ISE

Secondly, the PID controller parameters determined through Q-learning-based reinforcement learning (RL) are as follows: Kp = 1.3819, Ki = 0.12864, Kd = 0.26231. We use the transfer functions in Eq. (4) for heart and pacemaker dynamics. In this study, a Q-learning based PID tuning algorithm is used to find the PID parameters of the system through reinforcement learning as follows.

Algorithm: Q-learning based PID parameter optimization:
Step 1: Initialization:

- **Initialize PID Parameter Ranges:** Define the ranges for K_p, K_i, and K_d and initialize the Q-table with small random values.
- **Set Learning Parameters:** Define the learning rate (α), decay factor (γ), exploration rate (ϵ), and number of episodes.
- **Set Random Seed:** Use the rng() function to set a fixed seed value for reproducibility of simulations.

Step 2: Start the Loop (for each episode):

- **Initial State:** Choose a random combination of K_p, K_i, K_d .

For each step:

1. **Select Action:**
   - Use the Epsilon-Greedy strategy to select an action:
     - If a random number is less than ϵ\epsilonϵ, choose a random action (exploration).
     - Otherwise, select the best action based on the current Q-values (exploitation).
2. **Action Implementation:**
   - Apply the selected PID parameters (K_p, K_i, K_d) to the system.
3. **Simulate the System:**
   - Create the feedback loop according to the system's transfer function and obtain the system response (e.g., step response).
4. **Calculate Reward:**
   - Calculate the reward based on the system's performance metrics such as error, overshoot, and settling time. The reward can include penalties for these metrics.

$$reward = -error - 0.1 * overshoot - 0.01 * settling\_time$$

5. **Update Q-Table:**
   - Update the Q-value using the Bellman equation:

$$Q(state, action) = Q(state, action) + \alpha(reward + \gamma . \max\left(Q(next_{state}, next_{action})\right) - Q(state, action))$$

6. **Transition to New State:**
   - Determine the new state based on the selected PID parameters and continue the loop for the next step.
7. **Epsilon Decay:**
   - Decrease the exploration rate (ϵ) at the end of each episode according to the decay factor (γ): $\epsilon = \max(0.1, \epsilon . decay\_factor)$
8. **Complete the Loop:**
   - End the episode if the error, overshoot, and settling time are within specified limits. $error < 0.005$ && $overshoot < 0.05$ && $settling\_time < 3$

Step 3: Result: Select the optimal K_p, K_i, and K_d parameters from the Q-value with the best reward.
Step 4: End of Loop:

- Terminate the algorithm when the desired performance criteria are met.

The Q-learning parameters were set in shown in Table 4. The ranges for the PID parameters utilized in the simulations were established as follows: Kp spans from 1 to 20, while both Ki and Kd range from 0.1 to 2. Each parameter was divided into 200 linearly spaced values for optimization. Additionally, a fixed seed value was used to ensure reproducibility, which was implemented using MATLAB's RNG function.

Table 4. The parameter used in simulations for RL

Parameter	Reinforcement Learning (RL)
Learning Rate	0.1
Decay factor	0.9
Exploration Rate	0.1
Number of Episodes	1000

The step response of the closed-loop system optimized with RL is illustrated in Figure 5. The optimum PID parameters obtained by reinforcement learning using the above parameters and 100 and 123 as fixed seeds are $K_p = 1.3819$, $K_i = 0.12864$, $K_d = 0.26231$ and $K_p = 1.0955$, $K_i = 0.10955$, $K_d = 0.45327$, respectively. In this case, the step response of the system is given in Figure 5. In Figure 5, Simulation I is obtained when the fixed seed is 100 and Simulation II is obtained when the fixed seed is 123.

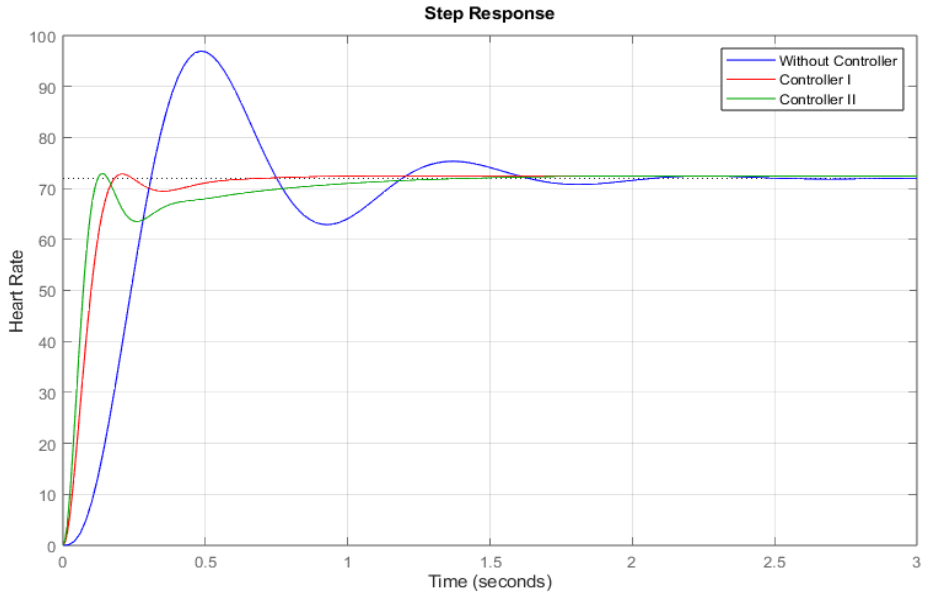

Figure 5. The simulation results of the pacemaker control system using RL.

To better understand the difference between Reinforcement Learning and Genetic Algorithm, the step response of the closed-loop system generated using the PID parameters obtained with both algorithms is shown in Figure 6. Table 5 presents a comparison of the performance metrics of GA, RL, and the uncontrolled system. As can be seen from Table 5, the step response of the pacemaker controlled with the PID controller obtained with RL.

The simulation results of the pacemaker control system in Table 5 show that the step response obtained with the control parameters optimized by both RL and GA have lower overshoot values compared to the system without controller. The overshoot value and the maximum peak value of the step response of the system controlled with PID parameters obtained by RL (1.14% and 72.83) are significantly lower than those of the system controlled with PID parameters obtained by GA (28.82% and 92.75). However, the step response of the system controlled with PID parameters generated by genetic algorithms shows much better performance for both rise time and settling time.

Tablo 5. Comparison of the performance metrics

Performance Metrics	PID Controlled System with RL	PID Controlled System with GA	Uncontrolled System
Rise Time (s)	0.1101	**0.035707**	0.1908
Settling Time (s)	0.4555	**0.342283**	1.5414
Overshoot (%)	**1.1413**	28.820775	34.6568
Peak (bpm)	**72.8217**	92.750958	96.9529

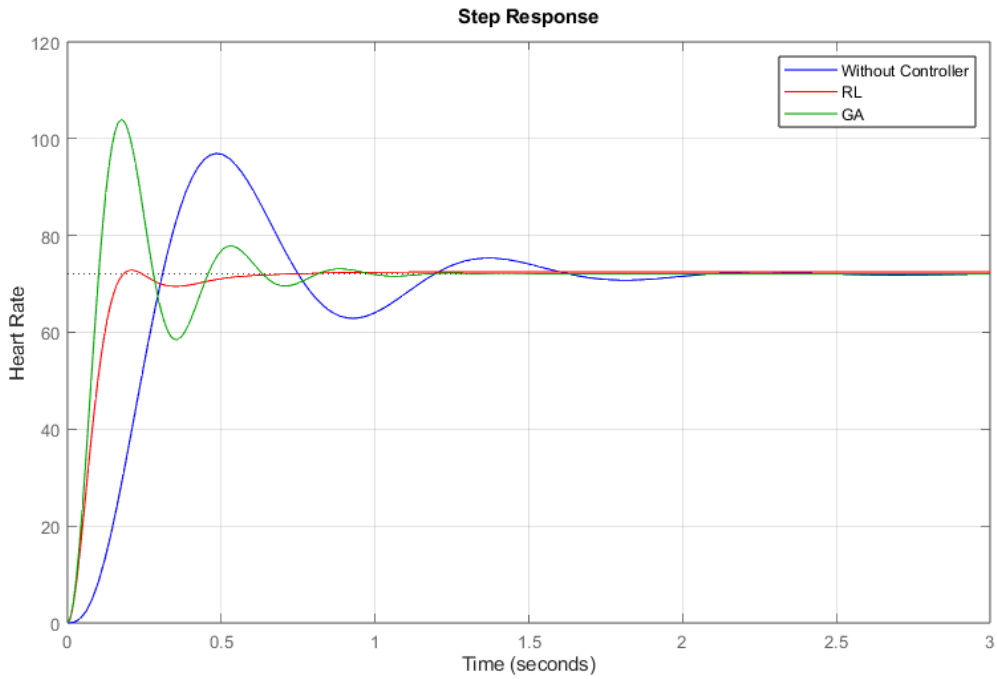

Figure 6. Comparison of the step responses

7. Conclusion

In this study, the effectiveness of Genetic Algorithm (GA) and Reinforcement Learning (RL) in optimizing PID controller parameters for pacemaker applications was investigated. The simulation results demonstrate that both methods improved the step response of the system compared to the uncontrolled closed-loop system. However, their advantages and limitations vary significantly.

Table 5 presents the performance comparison of the optimized PID parameters using both methods. The results indicate that RL-based tuning yielded significantly lower overshoot (1.14%) and peak value (72.82 bpm) compared to GA-based tuning (28.82% overshoot and 92.75 bpm peak value). This suggests that RL provides a more stable and accurate response, minimizing unwanted oscillations in heart rate regulation.

However, GA-based tuning outperformed RL in terms of rise time and settling time. The rise time for GA (0.0357 s) was significantly lower than RL (0.1101 s), and the settling time was also shorter (0.342 s for GA vs. 0.455 s for RL). This implies that GA is more effective for achieving a rapid response, which may be beneficial in scenarios requiring immediate stabilization of heart rate.

To further validate the effectiveness of these approaches, future studies could compare GA and RL with additional optimization techniques such as Particle Swarm Optimization (PSO) or Model Predictive Control (MPC). Additionally, real-time implementation and hardware validation on an actual pacemaker system would provide deeper insights into the practical feasibility of these methods.

The parameter settings for both GA and RL were carefully selected to ensure optimal performance in PID controller tuning. The following table summarizes the key parameters used in the simulations:

Acknowledgement

This research has been presented in IV. International Congress on Artificial Intelligence in Health.

This work has been supported by İzmir Bakırçay University Scientific Research Projects Coordination Unit, under grant number BBAP.2024.011. This work has been supported by İzmir Bakırçay University Scientific Research Projects Coordination Unit, under grant number BBAP.2024.011.

References

[1] Ponikowski, P., Anker, S. D., AlHabib, K. F., Cowie, M. R., Force, T. L., Hu, S., et al. (2014). Heart failure: preventing disease and death worldwide. ESC Heart Failure, 1(1), 4–25.

[2] Bajpai, S., Alam, S., Ali, M.A. (2017). Intelligent Heart Rate Controller using Fractional Order PID Controller Tuned by Genetic Algorithm for Pacemaker. International Journal of Engineering Research & Technology. 6(05), 715-720.

[3] Arunachalam, S. P., Kapa, S., Mulpuru, S.K., Friedman P.A., Tolkacheva, E.G. (2017). Intelligent Fractional-Order PID (FOPID) Heart Rate Controller for Cardiac Pacemaker, 2016 IEEE Healthcare Innovation Point-Of-Care Technologies Conference (HI-POCT), Cancun, Meksika, 2016, s. 105-108

[4] Bikki, P., Dhiraj, Y., & Kumar, R. N. (2023). Implementation of a Dual-Chamber Pacemaker for Low-Power Applications. https://doi.org/10.1109/icecct56650.2023.10179677

[5] Sutton R.S. and Barto, A.G. Reinforcement Learning: An Introduction. The MIT Press, London, 2018.

[6] Holland, J.H., Adaptation in Natural and Artificial Systems. Ann Arbor, MI: University of Michigan Press, 1975.

[7] Biswas, S.C., Das, A., Guha, P. (2006). Mathematical Model of Cardiovascular System by Transfer function Method, Calcutta Medical Journal, 4, 15-17.

[8] Noble, D., A Modification of the Hodgkin-Huxley Equations Applicable to Purkinje Fibre Action and Pace-maker Potentials, 1962 Journal of Physiology , 160, 317-352. PubMed ID: 14480151

[9] Beeler, G.W. and Reuter, H. (1977) Reconstruction of the action potential of ventricular myocardial fibres. The Journal of physiology, 268, 177-210.

[10] Ziegler, J. G., & Nichols, N. B. (1942). Optimum Settings for Automatic Controllers. Trans. ASME, 64(11), 759–768.

[11] Cohen, G. H., & Coon, G. A. (1953). Theoretical Consideration of Retarded Control. Trans. ASME, 75, 827–834.

[12] Momani, S., Batiha I.M., El-Khazali, R. (2019). Design of PIλDδ-Heart Rate Controllers for Cardiac Pacemaker, 2019 IEEE International Symposium on Signal Processing and Information Technology (ISSPIT), Ajman, United Arab Emirates, pp. 1-5. https://doi.org/10.1109/ISSPIT47144.2019.9001785

[13] Govind K.R.A., Sekhar, R.A.: Design of a novel PID controller for cardiac pacemaker. 2014 International Conference on Advances in Green Energy (ICAGE), Thiruvananthapuram, India, pp. 82-87 (2014). https://doi.org/10.1109/ICAGE.2014.7050147

[14] Alfarhan, K.A., Mashor, M.Y., Mohd Saad, A.R., Omar, M.I. (2018). Wireless heart abnormality monitoring kit based on raspberry pi. Journal of Biomimetics, Biomaterials and Biomedical Engineering. 35, 96–108. https://doi.org/10.4028/www.scientific.net/JBBBE.35.96

[15] Srivastava, R., Kumar, B. (2022). Design of Anfis based pacemaker controller having improved transient response and its FPGA implementation. Biomedical Signal Processing and Control. https://doi.org/10.1016/j.bspc.2021.103186

[16] Lima, G.S., Savi, M.A., Bessa, W.M. (2023). Intelligent control of cardiac rhythms using artificial neural networks. Nonlinear Dynamics. 111(12), 11543–11557. https://doi.org/10.1007/s11071-023-08447-1

[17] Chen, E.Z., Wang, P., Chen, X., Chen, T., Sun, S. (2022). Pyramid Convolutional RNN for MRI Image Reconstruction. in IEEE Transactions on Medical Imaging. 41(8), 2033-2047. https://doi.org/10.1109/TMI.2022.3153849

[18] Nako, J., Psychalinos, C., & Elwakil, A. S. (2023). Minimum Active Component Count Design of a PIλDμ Controller and Its Application in a Cardiac Pacemaker System. J. Low Power Electron. Appl. https://doi.org/10.3390/jlpea13010013 (7)

[19] Yürdem, B., Aksu, M. F., & Sağbaş, M. (2024). Design of Fractional/Integer Order PID Controller Using Single DVCC and Its Cardiac Pacemaker Application. Circuits Syst Signal Process. https://doi.org/10.1007/s00034-024-02810-2 (10)

[20] Beg, A. H., & Islam, M. Z. (2016, June). Advantages and limitations of genetic algorithms for clustering records. In 2016 IEEE 11th Conference on Industrial Electronics and Applications (ICIEA) (pp. 2478-2483). IEEE.

[21] Chen, X., Qu, G., Tang, Y., Low, S., & Li, N. (2022). Reinforcement learning for selective key applications in power systems: Recent advances and future challenges. IEEE Transactions on Smart Grid, 13(4), 2935-2958.

[22] Zhang, T., & Mo, H. (2021). Reinforcement learning for robot research: A comprehensive review and open issues. International Journal of Advanced Robotic Systems, 18(3), 17298814211007305.

[23] Arulkumaran, K., Deisenroth, M. P., Brundage, M., & Bharath, A. A. (2017). Deep reinforcement learning: A brief survey. IEEE Signal Processing Magazine, 34(6), 26-38.

[24] Hernandez-Leal, P., Kartal, B., & Taylor, M. E. (2019). A survey and critique of multiagent deep reinforcement learning. Autonomous Agents and Multi-Agent Systems, 33(6), 750-797.

[25] Köse, B., Işıklı, İ., Sagbas, M. (2024). Estimation of Weibull Probability Distribution Parameters with Optimization Algorithms and Foça Wind Data Application. Gazi University Journal of Science, 37(3), 1236-1254. https://doi.org/10.35378/gujs.1311992

[26] Armoogum S., Li, X., (2019). Big Data Analytics and Deep Learning in Bioinformatics with Hadoop. Deep Learning and Parallel Computing Environment for Bioengineering Systems, Elsevier. pp. 17–36. https://doi.org/10.1016/B978-0-12-816718-2.00009-9