

Chapter 26

Tuning of PID Controller Coefficient by Genetic Algorithm Using Runge-Kutta Method



Soumyendu Banerjee , Arnab Ghosh , Sanjay Bhadra ,
Rahul Kumar , Rahul Yadav , and Sarfaroze Ansari

Abstract This paper introduces an innovative approach for adjusting the coefficients of a Proportional-Integral-Derivative (PID) controller within a second-order control system. Although PID controllers are extensively utilized in engineering and industrial settings, conventional tuning techniques frequently face challenges in attaining the optimal balance between stability, rapid response, and error reduction. To address these challenges, a genetic algorithm (GA) was utilized to enhance the proportional, integral, and derivative gains by minimizing the total output error of the system. The system was represented in state space form and examined using the fourth-order Runge-Kutta numerical method, which guarantees precise dynamic assessment. Throughout the optimization phase, the state variable matrices were modified to keep the Mean Absolute Error (MAE) below 1%. The refined controller underwent testing with a unit step input, and the system's performance was evaluated based on settling time, steady state error, and transient response. Simulation findings show that the PID controller optimized by GA markedly improves system performance in comparison to traditional tuning methods. The suggested approach results in decreased settling time, minimized error, and enhanced stability, showcasing its strength and flexibility for second-order systems. These findings validate that evolutionary algorithms present an effective and dependable strategy for sophisticated PID tuning, delivering superior control performance for real-world applications.

S. Banerjee (✉)

Centre of Excellence for Indian Knowledge Systems, Institute of Engineering and Management,
University of Engineering and Management, Kolkata 700160, India

A. Ghosh

Department of Robotics & AI and Office of the Sustainability, Institute of Engineering and
Management, University of Engineering and Management, Kolkata 700160, India

S. Bhadra · R. Kumar · R. Yadav · S. Ansari

Department of Electrical Engineering, Institute of Engineering and Management, University of
Engineering and Management, Kolkata 700160, India

Keywords PID controller · Runge-Kutta method · Genetic algorithm · Nonlinear equation

26.1 Introduction

The PID controller is mostly used controller for controlling various systems in industry purposes, because optimal selection of controller parameters makes the system to operate more accurately by achieving the desired output response. Hence, tuning of these parameter values has become a very important research area, to achieve more simplicity on system desired response. Now manual control and tuning of these parameters may not provide desired output response; therefore, various optimization techniques have been implemented to get optimized value.

The offline method of tuning of PID controller like Ziegler-Nicholas method [1] is the first heuristic method, but its main disadvantage is that sometimes it fails to achieve optimal values of PID controller coefficients. Therefore, Artificial Neural Network (ANN) is utilizing for optimizing these values with a great advantages. The Particle Swarm Optimization (PSO) is a optimization tool by searching for best solution by reducing or maximizing the cost function; hence, this method was implemented [2] on PID controller to achieve perfectly tuned parameters by reducing overall output error of the system. Also, Fuzzy supervised learning, in addition to Ant Colony Optimization (ACO) [3] has provided robustness to the system responses by reducing the overall output error of the system.

The Genetic algorithm (GA) [4] is an optimization technique based on genetic evaluation process that can locate optimal position of controlling parameter depending upon a fitness function. Now, implementation of genetic algorithm on PID controller [5–8] has provided great result by tuning the optimized value of PID controller coefficients. Also, controller values with fraction order [9–12] have provided better result compared to integer-order values of controller coefficient, based on GA. With the advancement of optimization techniques, Neuro-GA-based adaptive PID controller has also been proposed to achieve optimized value in different conditions of same system [13].

Now, while dealing with any control system, solving of its state space equation becomes a vital task to find out optimal operating condition. Now, solution of this nonlinear state space equation becomes quite difficult job while dealing with its optimal solution. Runge-Kutta method [14] is a iterative approach to find out the solution of a nonlinear equation using a set of linear equation, and by computing the result step by step [15] iterative sequence of equation, the actual solution of that nonlinear equation can be achieved within a specified tolerance value [16, 17].

Therefore in this work, the solution of first order nonlinear differential equation of a control system was computed using fourth order Runge-Kutta method and in addition to it GA was implemented to search for optimal value of PID controller coefficient. At first, a control system was designed with arbitrary state space matrixes. Then this system was realized using fourth order Runge-Kutta method with a PID

controller. Finally, genetic algorithm was implemented on this system to tune the PID controller coefficient to achieve optimized vales to limit overall output error within 1%.

26.2 Methodology

26.2.1 State Variable Representation of Control System

A PID controller is a widely used controller that provides a controlled and desired output for industrial purpose. The PID stands for proportional-derivative-integral controller and the equation and corresponding transfer function of it is shown below,

$$\begin{aligned} y(t) &= k_p e(t) + \frac{k_i}{T_i} \int_0^t e(t) dt + k_d T_d \frac{de(t)}{dt}, \\ \frac{Y(s)}{E(s)} &= C(s) = k_p + \frac{k_i}{s} + s.k_d, \end{aligned} \quad (26.1)$$

where k_p , k_d , k_i are the P , I , D gain values, respectively, T_d , T_i derivative time and integral reset time, respectively, $m(t)$, $e(t)$ and $C(s)$ are output, error of controller in time domain and transfer function in frequency domain, respectively. Now manual control of these three coefficient is very difficult because uncontrolled values of these parameters can causes undesirable response to the system depending upon the system characteristics, e.g., high value of P -controller makes the system output faltering, excess I -controller creates high overshoot, and system becomes oscillatory with high derivative action. Therefore, optimized value of these three coefficients is very necessary to achieve minimum error. In this work, therefore, genetic algorithm is proposed using Runge-Kutta method to optimize these values.

26.2.2 Runge-Kutta Method

The Runge-Kutta method is generally used to solve any first order ordinary differential equation. The first order control system can be represented as its state space model and from that state equation the using arbitrary values of state variable matrixes, the output values of that equation was derived by Runge-Kutta method. This method can approximate the value of “ p ” for a given value of “ q ” for any first order ordinary differential equation as shown below,

$$\frac{dq}{dp} = f(p, q). \quad (26.2)$$

Now, the fourth order Runge-Kutta method can be expressed as,

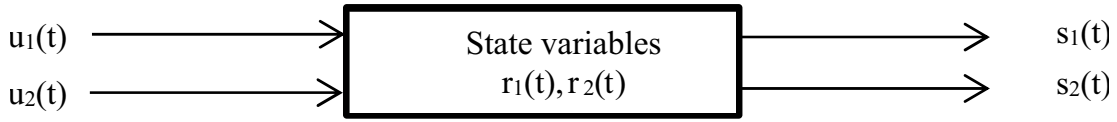


Fig. 26.1 2-input-2-output control system

$$\begin{aligned}
 K_1 &= hf(p_n, q_n), \\
 K_2 &= hf\left(p_n + \frac{h}{2}, q_n + \frac{K_1}{2}\right), \\
 K_3 &= hf\left(p_n + \frac{h}{2}, q_n + \frac{K_2}{2}\right), \\
 K_4 &= hf(p_n + h, q_n + K_3), \\
 q_{n+1} &= [q_n + K_1/6 + K_2/3 + K_3/3 + K_3/6], \quad (26.3)
 \end{aligned}$$

which provides the value of q_{n+1} with the help of q_n and p_n {where $n = 0, 1, 2, 3, \dots (x_{n+1} - x_n)/h$ }, with step input size h and p_0 is the initial value of p .

Let us consider a 2-input-2-output linear time invariant control system with 2 state variables as shown in Fig. 26.1, so the general equation can be represented by state variables as,

$$\begin{aligned}
 \dot{r}_1 &= f_1(r_1, r_2, u_1, u_2) \\
 \dot{r}_2 &= f_2(r_1, r_2, u_1, u_2) \quad (26.4)
 \end{aligned}$$

Now as it is a time invariant system, the term $\dot{r}$ in Eq. 26.4 is a linear combination of the inputs and state variables. Thus.

$$\begin{aligned}
 \dot{r}_1 &= a_{11}r_1(t) + a_{12}r_2(t) + b_{11}u_1(t) + b_{12}u_2(t), \\
 \dot{r}_2 &= a_{21}r_1(t) + a_{22}r_2(t) + b_{21}u_1(t) + b_{22}u_2(t). \quad (26.5)
 \end{aligned}$$

And this equations can be expressed as a matrix form as shown below,

$$\begin{bmatrix} \dot{r}_1 \\ \dot{r}_2 \end{bmatrix} = \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix} \times \begin{bmatrix} r_1 \\ r_2 \end{bmatrix} + \begin{bmatrix} b_{11} & b_{12} \\ b_{21} & b_{22} \end{bmatrix} \times \begin{bmatrix} u_1 \\ u_2 \end{bmatrix}, \quad (26.6)$$

where a_{ij} and b_{ij} are constant and the vector matrix can be written as,

$$\dot{R} = AR + BU \quad (26.7)$$

Similarly, the output matrix can be constructed in the same manner and be represented as.

$$\dot{S} = CS + DU \quad (26.8)$$

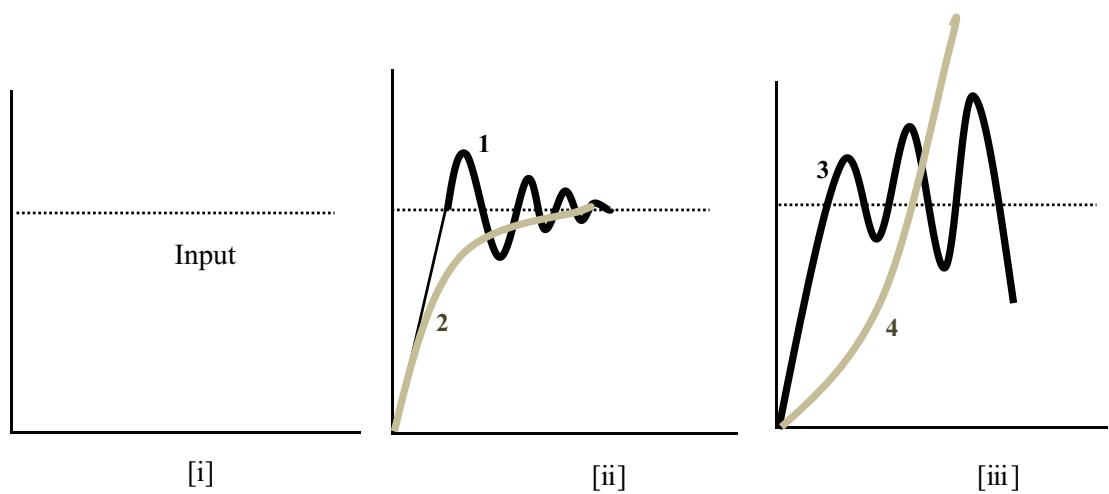


Fig. 26.2 Various types of system response, with input signal (a), stable output (b), unstable output (c)

The equation is called “state equation” and “output equation,” respectively.

Now for a control system, the values of A , B , C , and D are fixed and the system stability is dependent upon the output response with respect to input. In Fig. 26.2a with a step response, if the output is bounded within the same input as shown in Fig. 26.2b. On the other hand, an unstable output is shown in Fig. 26.2c.

Now in Fig. 26.2a, the output response 1, 2 is damped oscillatory in nature. The main drawback of this response is that the system takes a finite time to achieve its final stable condition after provision of input signal. Also, there present a overshoot (response 1) which sometimes causes malfunction of the whole system. Therefore, PID controller is used to reduce the oscillation so that system can acquire the final stable condition as early as possible with less maximum overshoot. Now manual provision of PID controller coefficient is quiet difficult; therefore, in this work for an arbitrary system with specific system matrixes, a PID controller is so designed to reduce the output error using genetic algorithm.

26.2.3 Genetic Algorithm Procedure

The genetic algorithm is an adaptive optimization technique to find out the best solution among a set of chromosome. This algorithm mainly possesses three rules. These are,

Selection: to select individual chromosome called parents that contributes to population set for the next generation.

Crossover: combines two parent chromosomes and exchanges their characteristics with each other.

Mutation: the parent chromosome change its characteristics randomly.

The whole process was implemented in MATLAB [18–20] simulation software, and the ranges of PID controller coefficients were chosen arbitrarily. Now, the parameters of GA were selected as followed,

- Population of Gene: 20.
- Number of generation: 30.
- Rate of cross over: 0.5.
- Crossover type: single point crossover.
- Rate of mutation: 0.05
- Mutation type: single point mutation.
- Reproduction method: Roulette Wheel.

26.2.4 *Implementation of Runge-Kutta Method in a Control System*

Choice of System: A stable system has a bounded output for a bounded input. In Fig. 26.2 for an step input signal when the output is bounded, it is stable system Fig. 26.2a, with output signal 1, 2, but the system is unstable, when it is unbounded, for signal type, 3,4, in Fig. 26.2c. Now, in Fig. 26.2a, the output is underdamped (response 2) and overdamped (response 1) type. The main aim of this work is to reduce peak overshoot as well as the settling time by decreasing the overall error (deflection from the desired output).

Now, a system can be realized using Runge-Kutta method as follows:

System realization process

1. $A = (3 \times 3 \text{ matrix})$, arbitrarily chosen
2. $B = (1 \times 3 \text{ matrix})$, arbitrarily chosen
3. $C = (3 \times 1 \text{ matrix})$, arbitrarily chosen
4. $D = (1 \times 1 \text{ matrix})$, arbitrarily chosen
5. $a_0, a_1, a_2 = (1 \times 1 \text{ matrix})$ empty matrix

PID controller realization process

1. $k_p = P\text{-controller}$ (lies between 0 and 255)
2. $k_i = I\text{-controller}$ (lies between 0 and 31)
3. $k_d = D\text{-controller}$ (lies between 0 and 31)

Runge-Kutta method implantation

1. $h = \text{step size}$
2. $r = \text{desired output}$
3. $a_0 = (k_p + h * k_i + (k_d/h));$
4. $a_1 = -(2 * (k_d/h) + k_p);$
5. $a_2 = (k_d/h);$
6. $u = a_0 * e + a_1 * f + a_2 * g + u;$
7. $k_1 = h * (A * X + B * u);$

```

8. k2=h*(A*(X+0.5*h*k1)+B*u);
9. k3=h*(A*(X+0.5*h*k2)+B*u);
10. k4=h*(A*(X+k3)+B*u);
11. X=X+(1/6)*(k1+(2*k2)+(2*k3)+k4);
12. y=C*X+D*u;          ## system output
13. e=r-y;               ## system error

```

Now using this above algorithm searching for optimal values of PID controller coefficients is described in next section.

26.3 Results and Discussion

In this work, an arbitrary system was chosen and its normal output response with step input was obtained. Then, a PID controller was designed with optimal values of coefficients and the whole procedure was implemented using Runge-Kutta method in MATLAB [21–23] simulating software. The result in both cases (with and without PID controller) is tabulated [24, 25].

Example 26.1 Transfer function of system:

$$\frac{1}{S^3 + 10S^2 + 5S + 1}.$$

The steady state variables of this system can be calculated as.

$$A = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ -1 & -5 & -10 \end{bmatrix} \quad B = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}$$

$$C = [1 \ 0 \ 0] \quad D = [0]$$

Now with a unit step input signal, the output response without any controller is shown in Fig. 26.3b. Now using a PID controller, after optimizing its coefficients, with values of 18, 31, 4 of PID controller, respectively, the output response is shown in Fig. 26.3a. Now, Table 26.1 shows the comparative result of these two output responses.

Example 26.2 Transfer function of system:

$$\frac{3}{S^3 + 20S^2 + 5S + 9}$$

The steady state variables of this system can be calculated as

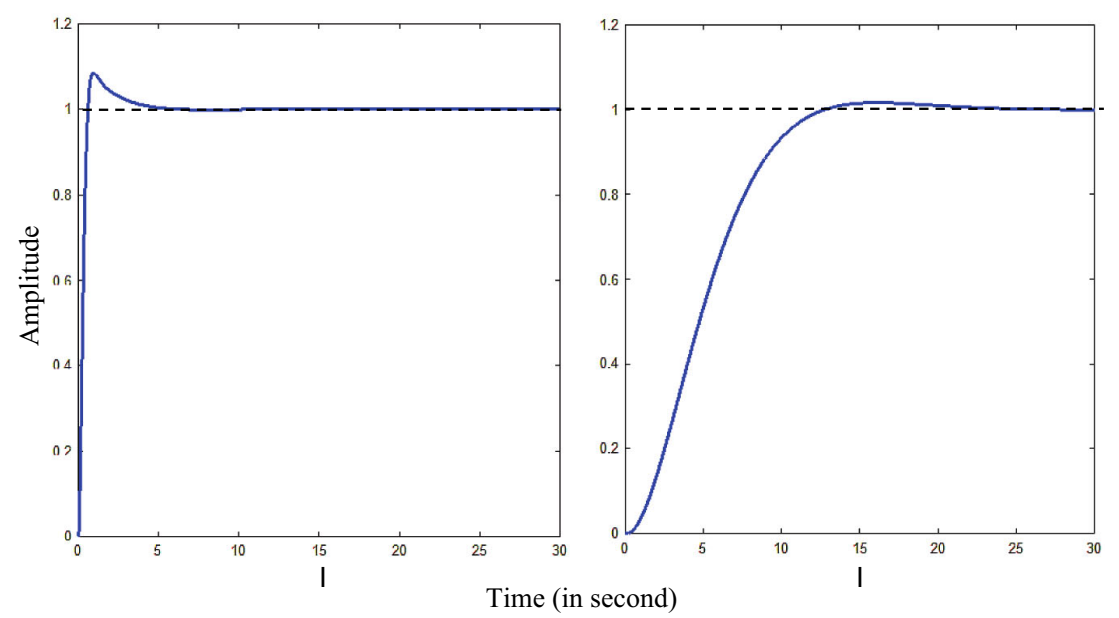


Fig. 26.3 Output response of system (Example 26.1) with optimized PID controller (a), without any PID controller (b)

Table 26.1 Achieved output result for Example 26.1

Characteristics	With PID controller	Without PID controller
Rise time	0.4155	7.68
Settling time	0.65	11.6
Settling max	1.03	1.02
Settling min	0.9996	0.9027
Overshoot (in %)	3	2
Peak time	0.95	16.06

$$\begin{aligned} A &= \begin{bmatrix} 0 & 3 & 0 \\ 0 & 0 & 1 \\ -3 & -5 & -20 \end{bmatrix}, \quad B = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}, \\ C &= [1 \ 0 \ 0], \quad D = [0]. \end{aligned}$$

Now with a unit step input signal, the output response without any controller is shown in Fig. 26.4b. Now using a PID controller, after optimizing its coefficients, with values of 7, 28, 4 of PID controller, respectively, the output response is shown in Fig. 26.4a. Now, Table 26.2 shows the comparative result of these two output responses.

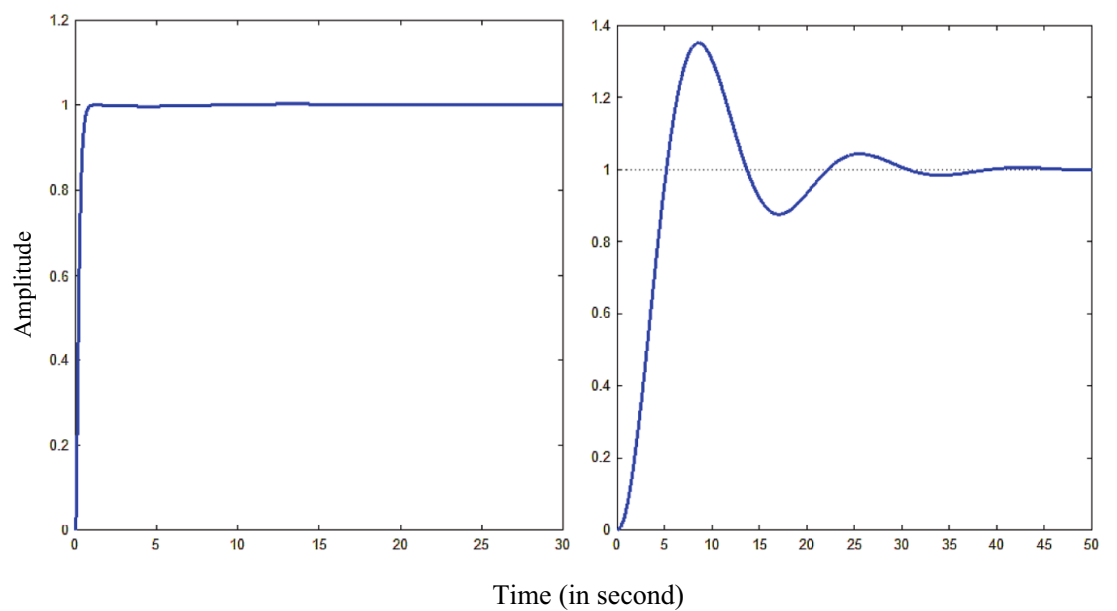


Fig. 26.4 Output response of system (Example 26.2) with optimized PID controller (a), without any PID controller (b)

Table 26.2 Achieved output result for Example 26.2

Characteristics	With PID controller	Without PID controller
Rise time	0.37	3.45
Settling time	0.65	28.75
Settling max	0.9998	1.35
Settling min	0.99959	0.8759
Overshoot (in %)	0.02	35
Peak time	1.26	8.63

26.4 Conclusion

In this work, an adaptive Genetic algorithm (GA)-based tuning approach for Proportional-Integral-Derivative (PID) controller coefficients has been successfully proposed and analyzed. The control system under study was modeled using its state space representation, and the nonlinear dynamics were addressed through linearization using the fourth order Runge-Kutta method to ensure accurate simulation. The primary objective of this research was to enhance the performance of the system by minimizing the MAE to within 1% and by significantly reducing the settling time, thereby achieving faster and more precise responses. The adaptive GA provided a systematic way of optimizing the PID parameters, overcoming the limitations of conventional trial-and-error tuning methods. Despite the absence of experimental validation owing to constraints related to equipment and scope, the simulation outcomes derived from accurate state space modeling and Runge-Kutta

techniques convincingly illustrate the efficacy of the GA-optimized PID controller. Future endeavors will concentrate on the experimental application to substantiate these results further. The results, as tabulated in the study, clearly demonstrate the effectiveness of the proposed approach in improving stability and transient response characteristics, highlighting its potential applicability for a wide range of control system applications.

References

1. Krohling RA, Rey JP (2001) Design of optimal disturbance rejection PID controllers using genetic algorithm. *IEEE Trans Evol Comput* 5:78–82
2. Gaing Z-L (2004) A particle swarm optimization approach for optimum design of PID controller in AVR system. *IEEE Trans Energy Convers* 19(2):384–391
3. Samarth Singh RM (2014) Comparative analysis of robustness of optimally offline tuned PID controller and fuzzy supervised PID controller. In: *Proceedings of the of IEEE RAECS UIET Panjab University*, Mar 2014, India
4. Srinivas M, Patnaik LM (1994) Adaptive probabilities of crossover and mutation. *IEEE Trans Sys Man Cybe* 24(4):656–667
5. Fan L, Joo EM (2009) Design for Auto-tuning PID controller based on genetic algorithms. In: *Proceedings of the 4th IEEE conference on industrial electronics and applications*, China, pp 1924–1928
6. Jones AH, Oliveira PBD (1995) Genetic auto-tuning of PID controllers. In: *Proceedings of the institution of electrical engineers conference on genetic algorithm engineering systems: innovations and applications*, pp 141–145
7. Lin G, Liu G (2010) Tuning PID controller using adaptive genetic algorithms. In: *Proceedings of the 5th international conference on computer science and education*, China, pp 519–523
8. Mitsukura Y, Yamamoto T, Kaneda M (1999) A design of self-tuning PID controllers using a genetic algorithm. In: *Proceedings of the American control conference*, California, pp 439–443
9. Wang L, Wang C, Li Y, Liu Y, Sun J (2017) Design of fractional—order PID controller based on genetic algorithm. In: *Proceedings of IEEE 29th Chinese control and decision conference (CCDC)*, China, pp 808–813
10. Maurya R, Bhandari M (2016) Design of optimal PID [FOPID] controller for linear system. In: *Proceedings of IEEE international conference on micro-electronics and telecommunication engineering*, India, pp 439–443
11. Park J-H (2025) Optimal controller design for a mobile robot using genetic algorithm and adaptive PID controller. *IEEE Access* 13:86167–86184. <https://doi.org/10.1109/ACCESS.2025.3570472>
12. Zhang Y et al (2023) Research on stellar spectrum simulation method based on genetic algorithm and fuzzy PID composite control. *IEEE Access* 11:134318–134330
13. Vinodhini R, Ganesh C, Patnaik SK (2012) Genetic algorithm optimized on-line neurotuned robust position control of BLDC motor. In: *Proceedings of IEEE students' conference on electrical, electronics and computer science*, India
14. Butcher JC, Mahooti M (2016) *Numerical methods for ordinary differential equations*. Wiley, United Kingdom
15. Lin R, Wang C, Liu F, Y X (2010) A new numerical method of nonlinear equations by four order Runge-Kutta method. In: *Proceedings of IEEE international conference on industrial engineering and engineering management*, China, pp 1295–1299
16. Li XY, Chen CM (2008) Solving nonlinear equations by Runge-Kutta method. *Math Theor Appl* 28(2):62–65

17. Suzuki C (2009) Two-stage sand-Runge-Kutta methods powerful for non-linear equations with multiple solutions. In: Proceedings of the computation world: future computing, service computation, cognitive, adaptive, content, and patterns, China, pp 575–579
18. Patra N, Banerjee S, Bhadra S (2025) On-device compression of multilead electrocardiogram using tunable-Q wavelet transform and MLPNN trained using multi-optima optimization based PSO. *Signal Image Video Process* 19(8)
19. Banerjee S, Singh GK (2021) A new approach of ECG steganography and prediction using deep learning. *Biomed. Sig. Proc. Con.* 64:102151
20. Banerjee S, Singh GK (2022) A Robust Bio-signal Steganography with Lost-data Recovery Architecture using Deep Learning. *IEEE T. Inst. Meas.*:1–1
21. Banerjee S, Singh GK (2023) A new real-time lossless data compression algorithm for ECG and PPG signals. *Biomed. Sig. Proc. Con.* 79:104127
22. Banerjee S, Singh GK (2022) Agent-based beat-by-beat compression of 12-lead electrocardiogram signal using adaptive Fourier decomposition. *Biomed Signal Proc Control* 75:103628
23. Banerjee S, Singh GK A new moving horizon estimation based real-time motion artifact removal from wavelet subbands of ECG signal using particle filter. *J Signal Process Sys* 95(8):1021–1035
24. Banerjee S, Singh GK. AUDSER: Auto-detect and self-recovery reversible steganography algorithm for biological signals. *Biomed Signal Process Control* 100, article no. 106974
25. Banerjee S, Singh GK. Quality aware compression of multilead electrocardiogram signal using 2-mode tucker decomposition and steganography. *Biomed Signal Process Control* 64, article no. 102230