Optimized LQR-PID Controller Using Squirrel Search Algorithm for Trajectory Tracking of a 6-DoF Parallel Manipulator

Chandan Choubey (✉ er.choubey.chandan@gmail.com)

National Institute of Technology Kurukshetra <https://orcid.org/0000-0002-2034-2054>

Jyoti Ohri

National Institute of Technology Kurukshetra

Research Article

Keywords: 6-DOF parallel manipulator, LQR-PID controller, squirrel search algorithm, optimal torque

Posted Date: December 21st, 2021

DOI: <https://doi.org/10.21203/rs.3.rs-1114813/v1>

License: This work is licensed under a Creative Commons Attribution 4.0 International License.

[Read Full License](#)

Optimized LQR-PID Controller Using Squirrel Search Algorithm for Trajectory Tracking of a 6-DoF Parallel Manipulator

*Chandan Choubey

*Electrical Department, Dronacharya Group of Institutions, Greater Noida, UP, India
E-mail: er.choubey.chandan@gmail.com

Jyoti Ohri

Electrical Department, NIT Kurukshetra, Haryana, India
E-mail: ohrijyoti@rediffmail.com

Abstract

In 6 Degree of Freedom (DOF) parallel manipulator, trajectory tracking is one of the main challenges. To obtain the desired trajectory, the DC motor needs to generate optimal torque. So to obtain optimal torque, an optimized Linear Quadratic Regulator-Proportional-Integral-Derivative (LQR-PID) controller is presented in this paper. For optimizing the Q, R and gain parameters of LQR-PID controller, Squirrel Search Algorithm (SSA) is presented. In this algorithm, minimal cost function of LQR-PID controller is considered as objective function. The SSA based LQR-PID controller leads the motor to generate optimal torque that helps to attain the desired trajectory of 6-DOF parallel manipulator. Results of the work depicts that the SSA based LQR-PID controller achieves the best mean velocity, sum square error (SSE), integral square error (ISE) and integral absolute error (IAE).

Keywords: 6-DOF parallel manipulator, LQR-PID controller, squirrel search algorithm and optimal torque.

1. Introduction

Parallel manipulators comprise of at least one closed loop mechanisms and their fixed base is connected to moving platform through at least two serial kinematic chains. Contrasted with serial manipulators, they show better powerful execution, higher accuracy, speed increase, minimization, payload capacity and furthermore have lower producing cost. Appropriately, these manipulators have been utilized in many fields, for example, flight simulations, in fast machine instrument applications, pick-and-spot activity, delicate cutting, bundling and clinical tasks. Be that as it may, the disadvantages of little work area and singularity confine the usage of these manipulators. Luckily, redundancy idea can adequately adapt to these difficulties since, for a given position, an exceptional solution can be picked among a limitless arrangement of inverse kinematic solutions relating to all substantial manipulator configurations [1-3].

Trajectory tracking control is a main point of contention in the field of 6-DOF parallel manipulator. It means to empower the joints or connections of the robot manipulator for tracking the target trajectory with ideal unique quality or to balance out them in the predetermined position. A sensible learning gain matrix can further develop the speed of trajectory tracking. To get the ideal learning gain, many explores have been finished including variable exponential gain technique, fuzzy PID strategy, and the strategy joining neural network controller and compensation controller. Be that as it may, the learning law and the complex control law will bring a huge measure of calculation time and influence the speed of trajectory tracking [4-6].

Besides, over the recent decades, many researchers had presented novel algorithms for controlling and tracking of ideal path, but none of them is comparable with PID controller because of its simple performance and intricacy. Zeigler Nichols introduced an empirical scheme for adjusting the PID gains, which is used in spite of some drawbacks and restrictions. The major drawbacks of the PID controller are error in computation, noise degradation, over-simplification and complications due to inefficient tuning. This prompted many researchers to evolve a classical optimal control theory to formulate a Linear Quadratic Regulator (LQR) which reduces the excursion in state trajectories of a system with minimum controller effort. LQR controller is popular in handling control [7] algorithm for stable and precious trajectory tracking issues.

Contributions of the paper are described as follows,

- a. To enhance the performance of trajectory tracking of 6-DOF parallel manipulator, an optimized LQR-PID controller is presented in this paper.
- b. Efficiency of LQR-PID controller is enhanced by optimizing the parameters Q, R and gain parameters. For optimization, SSA algorithm is presented.
- c. The performance of the approach is analyzed with the mean velocity, SSE, ISE, IAE and cost function.

Remaining sections of the paper are organized as pursues. Recent related literatures are reviewed in section 2. Section 3 presents the optimized LQR-PID controller for trajectory tracking of 6-DOF parallel manipulator. Results of the proposed scheme are analyzed in section 4. The conclusion of the work is described in section 5.

2. Related works

As the trajectory tracking leads to support the performance of parallel manipulator, many researchers had focused on that. For example, Haiqiang Zhang *et al* [8] had proposed an effective trajectory

tracking control method by combining position/force hybrid with the integration of inertia feed-forward control and back propagation (BP) neural network PID control for parallel mechanism with redundant actuation. The task space and joint space of the dynamic models were formulated using d'Alembert and virtual work formulations. Because of the position/force hybrid control method, the authors achieved good performance of trajectory tracking.

Atila Bayram [9] had proposed a kind of planar parallel manipulator with 3-DOF. The proposed model was structured as a variable geometry truss which is also called as planar Stewart platform. The workspaces with reachable and orientation were attained for the manipulator. The author had solved the inverse kinematic analysis depend on the avoidance of joint and redundancy limit. Then, virtual work technique was used to establish the manipulator's dynamic model. The simulations were executed to pursue the input planar trajectories with the dynamic equations of the computed force control scheme and the variable geometry truss manipulator. In the method, gain parameters of PD controller were optimized using genetic algorithm.

Xi Wang and Baolin Hou [10] had proposed trajectory tracking control of 2-DOF manipulator utilizing an implicit Lyapunov function technique and computed torque control (CTC) method. The manipulator was performed under the constraints of payload uncertainty and random base vibration. In both vertical and pitching direction, base vibration performed on the manipulator. The CTC method was used to linearize the non-linear coupling manipulator framework as well as it was used to decouple the framework. Implicit Lyapunov function control method assured the control forces bounded in norm via the process of control. By presenting this combined scheme, the authors had achieved high accuracy of tracking.

Jitendra Kumar, Vineet Kumar and K. P. S. Rana [11] had presented a fractional-order self-tuned fuzzy PID controller for controlling MIMO, non-linear and three link robotic manipulator framework. The authors aimed to reduce the complexity of manipulator system. So, the proposed controller was proposed for model uncertainty, disturbance avoidance, trajectory tracking and noise suppression. The gain parameter of the PID controller is tuned using cuckoo search algorithm. The execution of the proposed controller was compared with that of the integer-order and fractional-order fuzzy PID controllers. Due to the proposed controller, integral of absolute error was minimized.

L. Angel and J. Viola [12] had presented fractional order PID controllers for tracking control of a robotic manipulator kind delta. The fractional order PID controller is quite unique from integer order PID controller. Also, it was combined with the CTC method. The proposed model was included following three phases. Identification phase was executed with the least squares algorithm. The CTC

was utilized to linearize the robotic system model. Then, integer order and fractional order PID controllers were designed from the linearized model. Dynamical behaviour was evaluated in terms of evaluation trajectories. Because of the proposed scheme, the authors attained better spatial error, joint error and applied torque.

Pedro Pedrosa Reboucas Filho *et al* [13] had presented two schemes for dealing singularities and to enhance the control of trajectory tracking. One scheme was performed depend on a local genetic algorithm and the second scheme was performed depend on a global genetic algorithm. Each scheme's performance was evaluated using Polynomial trajectories up to 3rd degree in terms of computation cost, trajectory error and count of singularities. The authors had concluded that the global genetic algorithm achieved better computation cost and trajectory error.

3. Optimized LQR-PID Controller for Trajectory Tracking of 6-DOF Parallel Manipulator

3.1. Overview

Figure 1 illustrates the overall structure of the proposed scheme. At first, the objective function is defined using error function $e(t)$. In this work, cost function is considered as the objective function. Depend on the objective function; the SSA algorithm optimizes the Q, R and gain parameters of LQR-PID controller. Using the optimized LQR-PID controller, optimal torque is generated. With the optimal torque, the 6-DOF parallel manipulator outputs the actual trajectory $y(t)$. Then the error function $e(t)$ is attained by calculating the difference between the actual trajectory and reference trajectory. The process is continued until getting the desired trajectory.

Figure 1: The overall structure of the proposed scheme

3.2. Description of 6-DoF Parallel Manipulator

Figure 2 shows the structure of 6-DOF parallel manipulator. As shown in the structure, the manipulator has universal joints, base platform, legs and moving platform. At the two end of the manipulator, base and moving platforms are connected. These platforms are connected with the support of universal joints and the legs. Universal joints connect the base platform and the legs and the centre point of it is denoted as B_i . As well as, the centre point of the universal joints which connect the moving platform and legs is denoted as A_i . With the point B_i , a symmetrical hexagon is formed. Besides, symmetrical hexagon centre is denoted as C_b and it is the root point of the reference coordinate frame which is denoted as $\{C_b\}$. 'z' axes direction is perpendicular to the points of base platform upwards. 'x' axes direction is parallel to the points of base platform to the centre point between B_1 and B_6 . 'y' axes direction is estimated using the right-hand rule. With the point A_i , a symmetrical hexagon is formed. Besides, symmetrical hexagon centre is denoted as C_a and it is the root point of the reference coordinate frame which is denoted as $\{C_a\}$. 'z' axes direction is perpendicular to the points of moving platform upwards. 'x' axes direction is parallel to the points of moving platform to the centre point between A_1 and A_6 . 'y' axes direction is estimated using the right-hand rule. The vectors a_i and b_i denote the vectors of A_i and B_i in $\{C_a\}$ and $\{C_b\}$ respectively. These vectors are denoted as follows,

$$a_i = [x_{ai}, y_{ai}, 0] \quad (1)$$

$$b_i = [x_{bi}, y_{bi}, 0] \quad (2)$$

In the first state, the z axis of $\{C_b\}$ is joined together with the z axis of $\{C_a\}$. The x and y axes of $\{C_b\}$ are parallel to the x and y axes of $\{C_a\}$ and the point C_a vector in $\{C_b\}$ is 'h' and is denoted as follows,

$$h = [0, 0, H] \quad (3)$$

3.2.1. Inverse kinematics

The overall orientation of the 6-DOF parallel manipulator is represented as follows,

$$\Psi = \{x, y, z, \chi, \delta, \eta\} \quad (4)$$

Where, position (Q) of the manipulator translation and the posture (Z) of the manipulator orientation are represented as follows,

$$Q = \{x, y, z\} \quad (5)$$

$$Z = \{\chi, \delta, \eta\} \quad (6)$$

The orientation transformation is described using Roll Pitch-Yaw (RPY) angles. Initially, the coordinate frame $\{C_a\}$ is rotated around the x-axis of η angle which is known as Yaw, and then the occurring coordinate frame is rotated around the y-axis of δ which is known as Pitch. At final, the coordinate frame is rotated around the z-axis of χ angle which is known as Roll. The transformation matrix between the moving and reference coordinate frame is defined using (7).

$${}^B_A M = M(z, \chi) M(y, \delta) M(z, \eta) \quad (7)$$

Equation (7) also can be represented as follows,

$${}^B_A M = \begin{bmatrix} c\chi c\delta & c\chi s\delta s\eta - s\chi c\eta & c\chi s\delta c\eta + s\chi s\eta \\ s\chi c\delta & s\chi s\delta c\eta + c\chi c\eta & s\chi s\delta c\eta - c\chi s\eta \\ -s\delta & c\delta s\eta & c\delta c\eta \end{bmatrix} \quad (8)$$

Here, c and s denote the cos and sin functions respectively.

Using (5), (6) and (7), length of the leg of manipulator is determined. Figure 3 shows the i^{th} actuator's vector diagram. In the diagram, the dotted line represents the initial state $\{C_a\}$ and solid state represents the arbitrary position state in the overall orientation. The vector of i^{th} leg is attained by calculating the difference between a_i and b_i in $\{C_b\}$. The expression of i^{th} actuator is defined as follows,

$$u_i = ({}^B_A M \bullet a_i + {}^B P_A + h) - b_i = [u_{ix}, u_{iy}, u_{iz}]^T \quad (9)$$

Equation (9) can be written as follows,

$$u_i = \begin{bmatrix} c\chi c\delta x_{ai} + (c\chi s\delta s\eta - s\chi c\eta) y_{ai} + x - x_{bi} \\ s\chi c\delta x_{ai} + (s\chi s\delta c\eta + c\chi c\eta) y_{ai} + y - y_{bi} \\ -s\delta x_{ai} + c\delta s\eta y_{ai} + z + H \end{bmatrix} \quad (10)$$

Where, ${}^B P_A$ represents the position.

The i^{th} leg length is calculated using (11) and the length square of i^{th} leg is attained using (12).

$$k_i(x, y, z, \chi, \delta, \eta) = \sqrt{u_{ix}^2 + u_{iy}^2 + u_{iz}^2} \quad (11)$$

$$l_i(x, y, z, \chi, \delta, \eta) = u_{ix}^2 + u_{iy}^2 + u_{iz}^2 \quad (12)$$

Figure 2: Structure of 6-DOF parallel manipulator

Figure 3: i^{th} actuator's vector diagram

3.3. LQR-PID controller

To optimize the linear system, LQR design is an efficient method utilized in state-space feedback. Figure 4 shows the conceptual diagram of the LQR-PID of the manipulator. From the figure, state variables of the manipulator are described in the following equation (13).

$$x_1(t) = \int e(t) dt \quad x_2(t) = e(t) \quad \text{and} \quad x_3(t) = \frac{de(t)}{dt} \quad (13)$$

$$\frac{Y(s)}{U(s)} = \frac{K}{s^2 + 2\zeta^0 \omega_n^0 s + (\omega_n^0)^2} = -\frac{E(s)}{U(s)} \quad (14)$$

The input reference point $r(t)$ is set as '0' as the exterior set point is not influence the design of controller.

Equation (14) can be revised as follows if there is no change in the $r(t)$.

$$-kU(s) = E(s) \left[s^2 + 2\zeta^0 \omega_n^0 s + (\omega_n^0)^2 \right] \quad (15)$$

The Inverse Laplace transform of equation (15) is described as follows,

$$-Ku = \ddot{e} + 2\zeta^0 \omega_n^0 \dot{e} + (\omega_n^0)^2 e \quad (16)$$

The formulation of state space model is described as follows,

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & (\omega_n^0)^2 & 2\zeta^0 \omega_n^0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ -K \end{bmatrix} u \quad (17)$$

From equation (17),

$$-Ku = \ddot{x} + 2\zeta^0 \omega_n^0 \dot{x}_3 + (\omega_n^0)^2 x_2 \quad (18)$$

From equation (17), the following matrices are defined.

$$M = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & (\omega_n^0)^2 & 2\zeta^0 \omega_n^0 \end{bmatrix} \quad N = \begin{bmatrix} 0 \\ 0 \\ -K \end{bmatrix} \quad (19)$$

To solve the problem of LQR, the cost function is to be minimized. The cost function is defined using (20).

$$J(u) = \int_0^\infty [Qx(t)x^T + Ru(t)U^T(t)]dt \quad (20)$$

The state feedback control law minimizes the quadratic cost function. It is defined as follows,

$$u(t) = -V B^T R^{-1} x(t) = -F x(t) \quad (21)$$

Where V denotes the symmetric positive matrix of Continuous Algebraic Riccati function defined as follows,

$$V M^T + M V - N V R^{-1} N^T V + Q = 0 \quad (22)$$

Where, Q denotes the positive semi definite matrix and R denotes the positive constant. The appropriate ranges of Q and R are considered as $Q = \text{diag}[0,0,0]$ to $\text{diag}[500,500,500]$ and $R = 0.01$.

The related state feedback gain matrix is defined as follows,

$$F = R^{-1} N^T V = R^{-1} \begin{bmatrix} 0 & 0 & K \end{bmatrix} \begin{bmatrix} V_{11} & V_{12} & V_{13} \\ V_{12} & V_{22} & V_{23} \\ V_{13} & V_{23} & V_{33} \end{bmatrix} \quad (23)$$

$$= K R^{-1} \begin{bmatrix} V_{13} & V_{23} & V_{33} \end{bmatrix} = - \begin{bmatrix} K_i & K_p & K_d \end{bmatrix} \quad (24)$$

The gain parameters of PID can be estimated from equation (24) and are defined as follows,

$$\begin{aligned} K_i &= -K R^{-1} V_{13} \\ K_p &= -K R^{-1} V_{23} \\ K_d &= -K R^{-1} V_{33} \end{aligned} \quad (25)$$

To enhance the performance of LQR-PID controller for attaining the optimal control signal, the Q, R and gain parameters (K_i , K_p and K_d) of the controller is optimized using SSA algorithm, the following section describes the optimization of LQR-PID controller using SSA algorithm.

Figure 4: Structure of LQR-PID controller

3.4. SSA Algorithm Based LQR-PID Controller

Squirrel search algorithm is developed from the strategy of foraging behaviour followed by squirrels. The food search process isn't simpler even during warm climate (autumn), since the squirrels have to investigate various areas of forest and just glide from one tree to the next for looking through food resources. Anyway they get food from the plentifully available acorns. The squirrels consume acorns directly and begin looking for the optimal food source (hickory nuts) intended for to be stored for

winter season. This procedure of squirrels will assist them with satisfying their energy necessity during exceptionally harsh weather and accordingly the probability of survival is expanded. As the winter seasons bring about loss of leaf cover, the danger of predation is high. In this manner, the squirrels are less active (goes to hibernate mode) and decrease their foraging trips during such situations. During the ending of winter season, they are back to ordinary and become active in order to search food for their survival. The above process is repeated all through their lifecycle and this cyclic process prompts the establishment of SSA.

In the numerical model of SSA, a portion of the assumptions are made as of:

- ✓ Total count of squirrels is ‘ M ’ and everyone is thought to be on one tree.
- ✓ Food search of each squirrel is individual and applies dynamic foraging behaviour for choosing the ideal food source
- ✓ Three sorts of trees like normal tree, oak tree with acorn nuts food source and hickory tree with hickory nuts food source are only available in forest.
- ✓ Within search space, one hickory tree and three oak trees are available.

For instance, if the number of squirrels be, (say, $M=50$), their food resources are gotten from 3 acorn nut trees and 1 hickory nut tree, while the remaining 46 trees have no food source. From the entire population of squirrels, around 92% of them are on normal trees. Fittest squirrels discover optimal food sources and increase their survival rate. The algorithm is applied in this work to find the fittest solution (i.e. Q , R and gain parameters of LQR-PID) based on the foraging behaviour of squirrels. Steps involved in the proposed SSA based LQR-PID is given beneath,

Initialization: In this approach, the Q , R , K_i , K_P and K_d of LSTM are considered as solutions. The initialization of the candidate solutions or M number of flying squirrels S in different locations is randomly initialized as,

$$S_M = (S_{i1}, S_{i2}, \dots, S_{iD}) \quad (26)$$

Where, Y_{iD} can be defined as follows,

$$S_{iD} = (w_F, w_I, w_V, w_O)_{iD} \quad (27)$$

In above equation, each value $S_{i,D}$ represents the location of i^{th} squirrel in D^{th} dimension.

Also, the random location of squirrels is allocated based on uniform distribution function applied within the minimum $S_{\min}$ and maximum $S_{\max}$ range of l^{th} squirrel in k^{th} dimension is given as,

$$S_l = S_{\min} + v(0,1) \times (S_{\max} - S_{\min}) \quad (28)$$

Where, $v(0,1)$ is the uniformly distributed arbitrary number within range $[0,1]$.

Fitness evaluation: During this stage, the fitness of each and every solution is evaluated. In this work, two objective functions are defined to attain the optimal Q, R, K_i, K_d and K_p. The fitness calculation is given as,

$$Fit = Min(\alpha J(u) + \beta J(k)) \quad (29)$$

Where, α and β denote the control factors within the range $[0, 1]$. $J(u)$ denotes the cost function Q and R parameters and is defined in equation (). $J(k)$ denotes the cost function of gain parameters K_i, K_p and K_d and defined as follows,

$$J(k) = (ISE * w_1) + (SSE * w_2) + (IAE * w_3) \quad (30)$$

Where, w_1 , w_2 and w_3 are weight factors and set as 1. Besides,

$$IAE = \int_0^{\infty} e(t) dt \quad (31)$$

$$SSE = \frac{1}{M} \sum_{i=1}^M [e_i]^2 \quad (32)$$

$$ISE = \int_0^{\infty} (e(t))^2 dt \quad (33)$$

In the above equation, the fitness of every solution is calculated with the help of eqn (20). By this way, the optimal parameters Q, R, K_i, K_p and K_d are found.

Update the solutions: Depend on the following phases; the solutions are updated until attaining the optimal solution.

Sort, declare and random select process: Once after finding the fitness values of every solution (i.e. squirrel's location), the values are sorted in ascending order. It is considered that the fittest position is recognized as to be on hickory nut tree. Also, the subsequent three best solutions are assumed to be on

acorn nuts trees and the squirrels tries to go in the direction of the hickory nut tree. Left over squirrels is assumed to be on normal trees. Once the declaration step is done, a random selection process is done to choose some squirrels to travel towards the hickory nut tree. However, all the squirrels are affected by predators and thus the algorithm is modelled based on the predator presence probability (p_{prob}).

Generate new locations: Once the assumptions about the flying squirrels are made at the declaration stage, the dynamic foraging behaviour based on the predator presence probability (p_{prob}) is mathematically formulated as follows,

Norm 1: Flying squirrels on acorn nut trees (S_A^i) tends to move towards hickory nut tree.

$$S_A^{i+1} = \begin{cases} S_A^i + g_{dist} \times g_{const} \times (S_H^i - S_A^i) & ; r_1 \geq p_{prob} \\ Rand & ; otherwise \end{cases} \quad (34)$$

Norm 2: Flying squirrels resting on normal trees (S_N^i) be likely to move towards acorn nut trees.

$$S_N^{i+1} = \begin{cases} S_N^i + g_{dist} \times g_{const} \times (S_A^i - S_N^i) & ; r_2 \geq p_{prob} \\ Rand & ; otherwise \end{cases} \quad (35)$$

Norm 3: Flying squirrels on top of normal trees (S_N^i) trying to move towards hickory nut trees.

$$S_N^{i+1} = \begin{cases} S_N^i + g_{dist} \times g_{const} \times (S_H^i - S_N^i) & ; r_3 \geq p_{prob} \\ Rand & ; otherwise \end{cases} \quad (36)$$

In the above equations (34), (35) and (36), g_{dist} represents the random gliding distance; ($g_{const} = 1.9$) denotes the gliding constant that helps controlling between exploration and exploitation phases of squirrels; i and $i + 1$ defines the current and next iteration; S_A^i , S_H^i and S_N^i represents the position of the flying squirrel that arrived acorn nut tree, hickory nut tree and normal trees respectively. Also, r_1 , r_2 and r_3 are random numbers within range [0, 1].

Aerodynamics of gliding: Gliding mechanism is illustrated by the resultant force produced by the lift (E) and drag (d) force of the flying squirrels. Thus, the lift-to-drag ratio (also termed as, glide ratio) of the flying squirrel gliding at steady speed descending at horizontal angle is defined as,

$$G = \frac{E}{d} = \frac{1}{\tan \psi} \quad (37)$$

In above equation, ψ represents the glide angle, and can be given as,

$$\psi = \arctan\left(\frac{d}{E}\right) \quad (38)$$

Here, the lower value of ψ helps in increasing the glide path length of the squirrels. Moreover, the lift force generated from the downward deflection of air flowing through the wings is given as,

$$E = \frac{1}{2\phi E_{coef} T^2 L} \quad (39)$$

Further, the frictional drag is given as,

$$d = \frac{1}{2\phi d_{coef} T^2 L} \quad (40)$$

Where, $\phi = (1.204 \text{ kg m}^{-3})$ represents the density of air; $T = 5.25 \text{ ms}^{-1}$ denotes the speed; $L = 154 \text{ cm}^2$ specifies the surface area of the body; E_{coef} and d_{coef} represents the lift coefficient and drag coefficients respectively.

Seasonal monitoring condition: As the seasonal change performs an significant role in the foraging activity of flying squirrels, it is necessary to add a seasonal monitoring condition to make the algorithm more realistic. To mathematically represent the seasonal monitoring condition, few steps are followed as,

evaluate seasonal constant ($c_s^{(i)}$) by,

$$c_s^{(i)} = \sqrt{\sum_{j=1}^z (S_{A,j}^i - S_{H,j}^i)^2}, \text{ where } i = 1, 2, 3 \text{ represents iteration count}$$

Find minimum seasonal constant using,

$$c_0 = \frac{10e^{-6}}{(365)^{\frac{2.5i}{i_{\max}}}}, \text{ where } i_{\max} \text{ represents maximum iteration}$$

// Analyze seasonal monitoring condition

if ($c_s^i \leq c_0$)

 Perform random relocation

Else ($c_s^i > c_0$)

Random relocation during winter season: Based on the seasonal monitoring condition, the active squirrels that are able to explore their optimal food source are found. Moreover, the squirrels which are still survived but cannot explore are also found and made to explore their optimal food source from the random relocation strategy. Here, the random relocation is based on the levy flight distribution method that can be formulated as

$$S_N^* = S_{\min} + \text{levy}(m) \times (S_{\max} - S_{\min}) \quad (41)$$

Where, $\text{levy}(y)$ represents the levy distribution as:

$$\text{levy}(y) = 0.01 \times \frac{u_p \times \delta}{|u_q|^{\frac{1}{\chi}}} \quad (42)$$

In the above levy distribution representation, the parameter $\chi = 1.5$ is a constant and u_p and u_q denotes the random numbers distributed normally within $[0,1]$. Also, δ is found as,

$$\delta = \left(\frac{\Gamma(1+\chi) \times \sin\left(\frac{\pi\chi}{2}\right)}{\Gamma\left(\frac{1+\chi}{2}\right) \times \chi \times 2^{\left(\frac{\chi-1}{2}\right)}} \right)^{\frac{1}{\chi}}, \text{ where: } \Gamma(y) = (y-1)! \quad (43)$$

Stopping criterion: Stopping criterion is made when the optimal solution is found or the maximum count of iterations is attained. At the end of SSA, the optimal parameters are found, which are then applied to the LQR-PID for estimating optimal control signal.

Algorithm 1: The optimal parameters of LQR-PID selection using SSA algorithm

Input: Random Q, R, K_i , K_P and K_d and max iterations

Output: Optimal parameters of LQR-PID

1. Predefine the search range for the values of parameters of LQR-PID.
2. Set the population size ‘ M ’ and make ‘ M ’ random initial solutions as like eq. (26).
3. Now find the fitness function for each solution

4. Sort every solution based on fitness
 5. Store best solution (i.e. controller gain parametric values)
 6. Construct new set of solutions based on predator presence probability (p_{prob})
 7. ***Evaluate seasonal monitoring condition***
 8. If $(c_s^i \leq c_0)$, Perform random relocation using eq. (41)
 9. Again find fitness function for the newer solutions
 10. Compare with previous best solution
 11. Replace and store if current solution is better than the previous solution
 12. Repeat
 13. If termination criteria is met
 14. Get final optimal parameters and apply in LQR-PID
 15. End
-

4. Results and Discussions

The proposed SSA based LQR-PID controller method for trajectory tracking is simulated in the platform of MATLAB. The execution of the SSA-LQR-PID controller is contrasted with that of the GWO-LQR-PD, PSO-LQR-PID and GA-LQR-PID. The values of the parameters of SSA algorithm as well as that of GWO (Grey Wolf Optimization) algorithm, PSO (Particle Swarm Optimization) algorithm, and GA (Genetic Algorithm) are tabulated in table 1.

Table 1: Values of algorithm parameters

Algorithm	Parameters	values
SSA	Maximum iteration	100
	Population size	50
	P_{prob}	0.1
	g_{cons}	1.9
GWO	Maximum iteration	100
	Search extent	2

	Total search extent	20
PSO	Maximum iteration	100
	Number of particles	20
	Inertia weight	0.2-0.9
	Best value of individual	2
	Velocity fastening parameters	6
GA	Maximum iteration	100
	Number of elements	4
	variation	0.3
	Critical	0.7
	Dimension of inhabitant	30

Table shows the optimized values of LQR-PID controller obtained by using SSA, GWO, PSO and GA. Contrasted to GWO, PSO and GA, SSA has the best convergence speed so that it outputs the best optimized values of LQR-PID controller i.e., $Q_1=412.33$, $Q_2=254.22$, $Q_3=271.13$, $R=.05$, $K_i= 81.34$, $K_p=98.71$ and $K_d= 78.22$. Although SSA algorithm takes 54 iterations to attain the optimized values, PSO algorithm takes only 48 iterations than the SSA algorithm. Table shows the error analysis of the different algorithms based LQR-PID controller for input link of the manipulator. As detailed in the table, IAE of the SSA based LQR-PID controller is reduced to 41%, 78%, 88% and 98% than that of the LQR-PID controller with GWO, PSO, GA and without optimization. Compared to LQR-PID controller with GWO, PSO, GA and without optimization, ISE of the SSA based LQR-PID controller is reduced to 33%, 70%, 80% and 98% respectively. The LQR-PID controller without optimization attains the SSE to 15.713 but SSA based LQR-PID controller obtains the least SSE i.e., 0.478 of SSE than the remaining. Besides, cost function of the SSA based LQR-PID controller is reduced to 2%, 46%, 67% and 97% than that of the LQR-PID controller based on GWO, PSO, GA and without optimization.

Table shows the analysis of torque error for different algorithms based LQR-PID controller. As the SSA algorithm optimizes the parameter of LQR-PID controller, optimal torque also attained to obtain the target trajectory. Besides, as depicted in the table, IAE of the SSA based LQR-PID controller is

reduced to 6%, 70%, 82% and 98% than that of the LQR-PID controller with GWO, PSO, GA and without optimization. Compared to LQR-PID controller with GWO, PSO, GA and without optimization, ISE of the SSA based LQR-PID controller is reduced to 14%, 57%, 98.23% and 98.74% respectively. The LQR-PID controller without optimization attains the SSE to 14.732 but SSA based LQR-PID controller obtains the least SSE i.e., 0.3265 of SSE than the remaining. Besides, cost function of the SSA based LQR-PID controller is reduced to 17%, 47%, 69% and 98% than that of the LQR-PID controller based on GWO, PSO, GA and without optimization. Table 5 depicts the power consumption of the 6-DOF manipulator with different algorithms based LQR-PID controller. As illustrated in the table, the proposed SSA based LQR-PID controller consumes the power 1.783W while remaining LQR-PID controller based on GWO, PSO, GA and without optimization consumes 2.962W, 8.342W, 12.48W and 23.576W respectively.

Table 2: Optimized values of LQR-PID controller parameters

		SSA	GWO	PSO	GA
LQR	Q_1	412.23	382.12	253.71	178.92
	Q_2	254.22	201.12	193.72	258.64
	Q_3	271.13	213.37	232.48	162.81
	R	0.05	0.07	0.09	0.28
PID	K_i	81.34	71.33	51.21	32.47
	K_p	98.71	97.32	67.37	41.83
	K_d	78.22	52.28	56.53	24.21
Iteration required		54	57	48	61

Table 3: The error analysis of the different algorithms based LQR-PID controller for input link of the manipulator

Metrics	SSA	GWO	PSO	GA	Without optimization
IAE	0.0123	0.0209	0.0562	0.108	1.032
ISE	0.0018	0.0027	0.0061	0.009	0.152
SSE	0.478	0.501	0.815	1.312	15.713

Cost function	0.4933	0.5012	0.913	1.496	17.01
----------------------	--------	--------	-------	-------	-------

Table 4: The analysis of torque error for different algorithms based LQR-PID controller

Metrics	SSA	GWO	PSO	GA	Without optimization
IAE	0.0178	0.0189	0.0598	0.0997	0.9879
ISE	0.0018	0.0021	0.0042	0.0987	0.1437
SSE	0.3265	0.4323	0.6925	1.1783	14.732
Cost function	0.3978	0.4789	0.7590	1.3148	15.215

Table 5: Power consumption of manipulator with different algorithms based LQR-PID controller

Methods	Power consumption (Watts)
SSA	1.783
GWO	2.962
PSO	8.342
GA	12.48
Without optimization	23.576

4.1. Performance analysis of optimized LQR-PID controller for six joint angles

As per the joint angles of 6-DOF manipulator, average joint velocity of different algorithms based LQR-PID controller is analyzed in figures 1-10. Figure 1 shows the mean velocity of different methods for joint angle 1. As illustrated in the figure, the SSA based LQR-PID controller obtained -0.0298 while GA, PSO and GWO attain -0.1528, -0.2103 and -0.0412 respectively. The mean velocity of different methods for joint angle 2 is depicted in figure 2. In the figure, GA, PSO and GWO based LQR-PID controller obtains mean velocity 0.0423, 0.0652 and 0.0141 respectively. But the proposed SSA based LQR-PID controller attains the 0.0103 of mean velocity. Besides, as shown in figures 3-6, for angle 3 SSA based LQR-PID controller attains -0.0423 of mean velocity, for angle 4 it attains -0.9872, for angle 5 0.1112 and for angle 6 it attains 0.00418.

Figure 5: The mean velocity of different methods for joint angle 1

Figure 6: The mean velocity of different methods for joint angle 2

Figure 7: The mean velocity of different methods for joint angle 3

Figure 8: The mean velocity of different methods for joint angle 4

Figure 9: The mean velocity of different methods for joint angle 5

Figure 10: The mean velocity of different methods for joint angle 6

5. Conclusion

To attain the desired trajectory for 6-DOF parallel manipulator, an optimized LQR-PID controller has been presented in this paper. The parameters Q , R , K_i , K_d and K_p of LQR-PID controller are optimally chosen using SSA algorithm. Because of the optimized LQR-PID controller, the motor has generated optimal torque. The optimal torque has helped to obtain the desired trajectory for 6-DOF parallel manipulator. The execution of the SSA based LQR-PID controller has been analyzed in terms of mean velocity, SSE, ISE and IAE. Besides, the performance of SSA based LQR-PID controller is compared with that of LQR-PID controller based on GWO, PSO and GA.

References

1. Shanzeng, Liu, Zhu Zhencai, Zi Bin, Liu Chusheng, and Yu Yueqing. "Dynamics of 3-DOF spatial parallel manipulator with flexible links." In 2010 International Conference on Mechanic Automation and Control Engineering, pp. 2327-2331. IEEE, 2010.
2. Fan, Caixia, Hongzhao Liu, Gexia Yuan, and Yanbin Zhang. "A novel 2T2R 4-DOF parallel manipulator." In 2011 Fourth International Symposium on Knowledge Acquisition and Modeling, pp. 5-8. IEEE, 2011.
3. Dachang, Zhu, Feng Yanping, Xiao Guifang, and Cai Jinbao. "A Novel Structure Design Method for Parallel Manipulators." In 2010 Third International Conference on Knowledge Discovery and Data Mining, pp. 359-362. IEEE, 2010.
4. Zihao, Song, Wu Bin, and Zhou Ting. "Trajectory tracking control of a spherical robot based on adaptive PID algorithm." In 2019 Chinese Control And Decision Conference (CCDC), pp. 5171-5175. IEEE, 2019.
5. Wang, Lin, and Chongquan Zhong. "Trajectory tracking of robot based on fractional order fuzzy pi controller." In 2018 International Conference on Sensing, Diagnostics, Prognostics, and Control (SDPC), pp. 767-771. IEEE, 2018.
6. Jouini, Marwa, Mohamed Sassi, Neji Amara, and Anis Sellami. "Modeling and control for a 6-DOF platform manipulator." In 2013 International Conference on Electrical Engineering and Software Applications, pp. 1-5. IEEE, 2013.
7. Kumar, E. Vinodh, and Jovitha Jerome. "LQR based optimal tuning of PID controller for trajectory tracking of magnetic levitation system." *Procedia Engineering* 64 (2013): 254-264.
8. H. Zhang, H. Fang, D. Zhang, Q. Zou and X. Luo, "Trajectory Tracking Control Study of a New Parallel Mechanism with Redundant Actuation", *International Journal of Aerospace Engineering*, vol. 2020, pp. 1-14, 2020.

9. A. BAYRAM, "Trajectory Tracking of a Planer Parallel Manipulator by Using Computed Force Control Method", Chinese Journal of Mechanical Engineering, vol. 30, no. 2, pp. 449-458, 2017.
10. X. Wang and B. Hou, "Trajectory tracking control of a 2-DOF manipulator using computed torque control combined with an implicit lyapunov function method", Journal of Mechanical Science and Technology, vol. 32, no. 6, pp. 2803-2816, 2018.
11. J. Kumar, V. Kumar and K. Rana, "Fractional-order self-tuned fuzzy PID controller for three-link robotic manipulator system", Neural Computing and Applications, vol. 32, no. 11, pp. 7235-7257, 2019.
12. L. Angel and J. Viola, "Fractional order PID for tracking control of a parallel robotic manipulator type delta", ISA Transactions, vol. 79, pp. 172-188, 2018.
13. P. Rebouças Filho, S. P. da Silva, V. Praxedes, J. Hemanth and V. de Albuquerque, "Control of singularity trajectory tracking for robotic manipulator by genetic algorithms", Journal of Computational Science, vol. 30, pp. 55-64, 2019.