

Optimal Trajectory Planning Analysis of Robot Manipulator Using PSO

gurjeet singh (✉ mann_567@yahoo.com)

Amritsar College of Engineering and Technology <https://orcid.org/0000-0003-4066-5675>

V. K. Banga

Amritsar College of Engineering and Technology

Research Article

Keywords: Robot manipulator, optimal trajectory planning, obstacle avoidance, shortest path, PSO

Posted Date: July 7th, 2021

DOI: <https://doi.org/10.21203/rs.3.rs-587955/v1>

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

[Read Full License](#)

Optimal Trajectory Planning Analysis of Robot Manipulator Using PSO

Gurjeet Singh, V. K. Banga

Research Scholar I.KG Punjab Technical University [1], Professor & Principal, ACET, Amritsar [2]

Email: - mann_567@yahoo.com, vijaykumar.banga@gmail.com

Abstract:-

For large- and small-scale industry, the major issue considered as to obtain the orientation and desire position of robot manipulators. The analysis of robotic manipulation requires two kinds of kinematic analysis namely inverse and forward kinematic analysis. This article aims to frame the inverse kinematic model of 5 DOF and 6 DOF robot manipulator. Planning of a movement flow has been designed followed by the evaluation of all the Denavit –Hartenberg (DH) parameter for the estimation of the desired orientation and position of the end effector. For the inverse Kinematics solution, conventional technique like DH notation, iteration and transformation were being utilized. The trajectory inverse kinematics minimization has been the main objective in this study and it has been achieved by Particle Swarm Optimization PSO. A quintic 5th order polynomial with joint space trajectory has been implemented in this study to find the paths for velocity and acceleration evaluation. Cartesian Trajectory is applied to the shortest path and get the transformation matrix for each intermediate point. Finally, the obstacle avoidance has been exhibited by PSO. The distance velocity, acceleration and angular displacement have been evaluated to analyse the shortest path and obstacle path avoidance.

Keywords: - Robot manipulator, optimal trajectory planning, obstacle avoidance, shortest path, PSO

Declaration : NA

Funding : NA

Conflicts of interest/Competing interests: NA

Availability of data and material: NA

Code availability (software application or custom code):NA

Ethics approval (include appropriate approvals or waivers): NA

Consent to participate (include appropriate statements): yes

Consent for publication (include appropriate statements): Yes

I. INTRODUCTION

The Robot trajectory planning is defined as track points provided multiple expectations and target pose. To the end-effector the rotation angle of each joint of robot adjusted in timely manner at approved trajectory followed by every point which reach the target point ultimately. Compared with Cartesian space, the trajectory planning in joint space considered as simpler and convenient [1]. Multiple fixed points situated at the multiple robotic arms end which are usually given. By using inverse kinematics these track points for robot have been computed and from Cartesian space it is converted to joint co-ordinate space [2]. Generally the kinematic transformation is categorised into forward and inverse transformation. The joint coordinates or configuration of robot mapping to the end-effector pose considered as forward kinematics. When the joint angles have been known, the issue in forward transformation lies in end-effector position in Cartesian space whereas the inverse transformation is just the alternative. The inverse kinematic transformation is generally complex to resolve due to its high non-linearity [3].

Further using different spline and polynomial functions and other curves, the track points are used to proceed the interpolation operation. For the robot, each joint variable's time expressions are achieved. Based on robot mechanical characteristics, the acceleration and speed of every joint has to be restricted within the given range. So that the acceleration and speed optimization is essential for every joint arm and other than joint arm smooth trajectories assurance the wear and impact should be reduced to increase the robot working life [4]. The optimal trajectory planning method has been suggested in [5], energy minimum trajectory planning, multi-objective trajectory optimization [6] or impact minimum trajectory planning and optimal trajectory planning has also been presented in [7]. Amidst them, the researchers usually focused on the optimal trajectory planning by means of running time of robot. For example, [8] constructed a function expression for energy dissipation from omnidirectional robot by means of obstacle avoidance performance combination and by the polynomial interpolation method the optimization problem transformed into parameter minimization and it leads to robot obstacle avoidance's effective study. Various optimization techniques as proposed in [9], Particle Swarm optimization based fuzzy technique [10], ant colony optimization [11] and others have been applied to addressed the obstacle avoidance problem and also focus on the collision free concept during trajectories planning movements. To reduce the trajectory optimization complexity of several constraints, the smoothness and angular displacement constraints, angular acceleration and angular velocity of each joint in space have been ensured. In this research paper the enhanced particle swarm optimization has been used to achieve the optimized trajectory planning. In addition to that, the obstacle avoidance has been deal with the particle swarm optimization and forward kinematics. Denavit Hartenberg DH parameters usually have four parameters related to specify convention for reference frames attaching to robot manipulator. For the end-effectors measurements the DH parameters used in this study [13, 14].

The main contribution of this research paper involves,

- i. Shortest distance identified for efficient trajectory planning for robot design.
- ii. The initial and final position theta for each joint will be find by using the Trajectory Inverse Kinematic Minimization by PSO.
- iii. From initial and final theta, we can able to find the paths by joint space trajectory with quintic (5th order) polynomial is used with boundary conditions for velocity and acceleration.
- iv. Cartesian Trajectory is applied to the shortest path and get the transformation matrix for each intermediate point.
- v. Trajectories intended in joint space could be utilized straightaway to regulate motors of low-level with no need of computing inverse kinematics.
- vi. There is no requirement to handle the singularities related to multiple DOF manipulators or redundant joints.
- vii. Dynamic constraints such as high acceleration on joints could be taken into account when trajectories are generated. On the other hand, the constraints have to be tested subsequently for Cartesian space trajectories pertaining to inverse kinematic.

1.1 Paper organization:

Organization of the research paper is as: Basic introduction and concepts regarding Robotic manipulators, forward and inverse kinematics, shortest path, PSO and obstacle avoidance are discussed in Section I. Section II deals with the related works parts where existing studies in the shortest path evaluation and obstacle avoidance in robotic manipulators using various algorithms were listed in an elaborate manner. Section III proposes an effective system of optimal trajectory planning. Section IV analyzes the performance of the proposed system by analyses the robot 1 and robot 2. Section V concludes the proposed work by highlighting the results and effectiveness of the proposed work.

II. RELATED WORKS

Robot Kinematics categorised into forward and inverse kinematics. The forward kinematics analyse the position of end-effector from the joint angles whereas from end-effector position using the inverse kinematics determine the joint angle. For redundant robot arm manipulator the inverse kinematics issue is highly complicated. To solve global optimization problem, a particle swarm optimization PSO algorithm method has been implemented. The three-link redundant manipulator robot arm based PSO proposed without using inverse kinematics equations in this paper. Three joint angles generated at its best and end-effected position generated at its best of three-link robot arm found by enhanced PSO. The square, circle and triangle path trajectories with respect to positions and joints resulted in smoother outcome compared with other trajectories. And also during PSO the three-link robot has been moving fast. In addition to that error rate has possessed too small [15, 16]. This study [17] proposes new technique for robot manipulators i.e. trajectory planning solving namely (TPBSO) trajectory-planning beetle swarm optimization which has been based on heuristic optimization algorithm. The algorithm's practical application has been considered as two specific manipulator trajectory planning issues which are fixed geometric path planning and point to point planning. After evaluating this algorithm results shows better control performance obtained without computational complexity increase and faster computation speed achieved further. For unmanned vehicles, In this study [18] proposed a realistic stochastic trajectory generation technique which offered a trajectories emulation tool in standard flight states. By using quintic B-splines, three scenarios trajectories executed. The Euler angles second order derivatives smoothness and accelerations granted by B-splines. A particle swarm optimization PSO based multi-objectives optimization method the quintic B-spline parameters tuned in search space. The unmanned aerial vehicle UAV configuration imposed the constraints which are satisfied by the proposed method. Some specified constraints has been introduced further like speed limitation, actuator torque limitations and obstacle avoidance. At the end the standard A* algorithm, genetic algorithm GA and standard rapidly-exploring random tree RRT* technique have been simulated for comparing proposed technique with respect to effectiveness in minimum length trajectory identification and implementation time.

In this research [19, 20], proposed powerful particle swarm optimization PASO model based novel divide and conquer optimization context which solves the inverse kinematics issues IK and path planning PP problems for general serial hyper-DOF manipulators. With any DOF number the inverse kinematics IK issue solved by PSOIHK-(generalised PSO application developed). Further, for hyper-DOF manipulators the scalable PP framework generated. Scalability, power and feasibility of proposed technique evaluated and collision free paths planned for simulated mobile robots and the generality has to examine. This [21] paper focused on redundant manipulator's continuous path planning issue where the desired path needs to follows by end-effector. Within the boundary the desired path tracked by redundancy solution and angle domain boundary. The joint angle travelling distance minimized by meta-heuristic optimizations which is selected the best solution by (GWO) Grey Wolf optimizer, Genetic Algorithm GA and PSO. Using sinusoidal function the angle domain trajectories have been designed. From Bezier and algebraic curves difficult geometrical path chosen termed as traced path with hyper redundant manipulator and 3-DOF arm manipulator. Compared with GWO and GA the PSO obtained better results. For serial and parallel manipulators [22, 23] developed (OTGA) Optimal Trajectory generation algorithm to create a reduced-time smooth motion trajectories. The two phases have been described in OTGA. Using cubic spline the reduced time optimal trajectory derivate in first phase and it has been mainly used in acceleration, velocity and robotics. Modifying the interpolation of cubic spline in the optimized trajectory of first and last knots seen in second phase. For optimization performance the PSO algorithm used and for PUMA robot the OTGA has been tested. The results shows that OTGA considered as superior trajectory generation algorithm by compared with others.

This paper[24, 25] explored the inverse kinematics of new 7-revolute joint robot arm calculation using the two variants from PSO. It has been expected that with minimum error the end effector of robot arm has to be move to the desired position in workspace. By global local best inertia weight and random inertia weight the results are obtained and related with standard PSO. This study revealed that for inverse kinematics solution the PSO algorithm has been used efficiently. This research[26] explored the human walking cycle with the help of cubic polynomial equation and quintic polynomial equation. The velocity, acceleration and position have been generated for normal human walking. The knee movement while walking has been copied and duplicated using device controlled by generated signal. From cubic and quartic equations of the gained data during the walking cycle of gait phases have been compared with real human walking data and discussed with respect to the analysed graphs. In this study [27, 28] proposed bionic foot-end trajectory adapted various terrains kinds and gaits depends on trajectory planning idea mixed with joint space and Cartesian space. For joint space trajectories quintic polynomials used and for inverse kinematics solution the trajectory points have been selected. From spatial domain to temporal domain, the interpolation point distributions have been evaluated. Among the original curve and actual trajectory the evaluation function has been recognised and the closeness degree assessed. For precise

trajectory, the PSO and GA algorithms used for selection of points. To deal with the smooth trajectory generation issue in parallel kinematic manipulator PKM, this work[29] proposed minimum jerk trajectory planning strategy. The proverbial virtual work and Jacobian matrices analysed the 3 prismatic universal to universal PKM (3 PUU-PKM). The sequence of joints angular position interpolated by discretised piecewise quintic polynomials in Cartesian space. The whole trajectory's angular jerks maximum joints has to be minimized in this study. The effectiveness and feasibility attained by this minimum jerk trajectory planning approach. For industrial robots, this study[30] proposed smooth point to point trajectory planning technique in joint space. The decelerated part, accelerated part and constant velocity have been considered as three parts from joint motion. By 4th order polynomial with root multiplicity with one co-efficient, the acceleration has been planned. Under the kinematical constraints the constant velocity has been maximized and hence the near time optimal trajectory has been obtained. The velocity, acceleration, jerk and displacement of every end-effector and joint shows all continuous with respect to proposed trajectory planning method. Finally the end-effector smoothly moves and effective proposed method resulted.

This research [31] explored the trajectory planning and shortest path of robot arm with two-link comprised of 2-DOF in 2-D static known environment has been examined. Three issues addressed by this evaluation. The first issue focused on free collision path planning in Cartesian space. The second issue concerns the shortest path using D* technique[32]. Third issue focused on intermediate points selection and using 5th order polynomial equations the smooth trajectory has to be attained. Hence by addressing all these issues the smooth trajectory finally obtained. This work [33] focused on various sub-optimal solutions based on redundancy resolving in joint space considered as initial step. The solution performed at acceleration level employed by inertia related criterion optimization, null-space damping and weighted pseudo-inversion. On various robot systems multiple numerical results achieved with faster motion compared with existing methods. In addition to that the obtained motion time with same Cartesian path has been resulted near to the global time optimal solution. This paper [34] implemented robust non-linear control depends on back stepping method on 7 DOF ETS-MARSE robot to provide the rehabilitation therapy for the stroke patients. This proposed method considered as new solution to inverse kinematics issue. The desired rehabilitation trajectories has been generated and optimal Cartesian solution ensured. The robustness, exactness and stability have been validated from the proposed method and shows the effectiveness. Under Cartesian environment [35] proposed novel collision-free motion planning technique preserve robot body from collision with objects and also robot's original task execution. Inside the robot workspace, various KinectV2 depth cameras operated to track the dynamic obstacles[36]. The closest point from obstacle to robot identified and clustered by K-nearest neighbour KNN algorithm. The velocity and obstacle position estimated by Kalman filter. Based on obstacle observation and task specification, the repulsive and attracted potential generated to avoid collision in collaborative operation for robot end effector. The experimental results shows that the obstacles efficiently avoided to complete original task in constrained environment from 6-DOF robot arm. For industrial manipulators, moreover [37, 38] proposed optimal planning technique based on geodesic to obtain smooth trajectory in accurate way. Geodesic refer as on Riemannian manifold the shortest path connecting any two points along itself. The orientations space and position categorised by end-effector. For both spaces the Riemannian metric assigned. By joint trajectories the Cartesian trajectories have been shown. For trajectory initial conditions the geodesic equations obtained for each space have been evaluated and finally the results plotted. Numerical evaluations performed to evaluate the geodesic effectiveness.

III. METHODOLOGY

3.1 A 6-DOF and 5-DOF robot optimal trajectory planning:

3.1.1 DH parameter initialization:

In order to identify and analyses the 6-DOF and 5-DOF robot arm used in the proposed research are similar to the traditional arm type robot. The kinematic equation formulation is necessary to analyze the arm robot. The proposed system is established by DH parameters initialization for robot modelling.

The four parameters of robotic arm used for linking the reference frames to other frames are,

d_i – from the previous frame distance to next frame through x_i and z_{i-1} perpendicular convergence- link offset/joint distance

a_i - link length/ offset distance among the z_{i-1} and z_i general perpendicular of axes.

θ_i - Joint angle and

α_i - link twist/ offset angle.

3.1.2 Inverse Kinematics:

The homogenous transformation matrix is easily measured by the D-H conversation. To represent the homogenous transformation, the D-H convention used four basic transformation product denoted by A_i . The matrix A_i is defines as homogenous 4x4 transformation matrix explains the orientation of the object and point position on an object in 3-D space. By using DH parameters the homogenous transformation matrix derived from one frame to next frame.

In D-H convention the homogenous transformation matrix has been determined as four basic transformation product,

$$i - 1_{A_i} = \begin{bmatrix} \cos \theta_i & -\sin \theta_i \cos \alpha_{i,i+1} & \sin \theta_i \sin \alpha_{i,i+1} & \alpha_{i,i+1} \cos \theta_i \\ \sin \theta_i & \cos \theta_i \cos \alpha_{i,i+1} & -\cos \theta_i \sin \alpha_{i,i+1} & \alpha_{i,i+1} \sin \theta_i \\ 0 & \sin \alpha_{i,i+1} & \cos \alpha_{i,i+1} & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (1)$$

The inverse kinematic analysis is the alternative to the forward kinematic analysis. Each joint corresponding variables have been found with provided position requirement if the manipulator end in provided references co-ordinates system. Analysis of inverse kinematics done by each inverse matrix multiplication of A_i matrices on left side of equation 4 and the corresponding elements are equalized.

$$i - 1_{A_i}^{-1} = \begin{bmatrix} n_x & o_x & a_x & p_x \\ n_y & o_y & a_y & p_y \\ n_z & o_z & a_z & p_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2)$$

The Procedure for the proposed robot design,

Initialize the DH parameter followed by Angle limitation for each Joint should be provided. Then create the Robot link by Peter Corke tool box and finally Manual angle can be used to verify. The joints are strictly connected by links and joint control variables are used. End effectors or tools are actuators or sensors associated with robotic device last link. The peter corke toolbox has been given various functions which are helpful in the proposed study and for arm type robotics simulations such as dynamics, trajectory generation and kinematics.

3.1.3 Optimal Trajectory Planning:

By the kinematic inverse operation, in the Cartesian space the number of required robot end effector track points are transformed into joint variables and further the joint space track points are interpolated and achieved joint variables of robot. To interpolate trajectory points, a quintic polynomial is used in robot joint space.

If the joint angle is θ_i , assume $\theta_{t_0} = \theta_0$, $\theta_{t_f} = \theta_f$. Various quintic polynomial curves used the described conditions and the shown in figure 1,

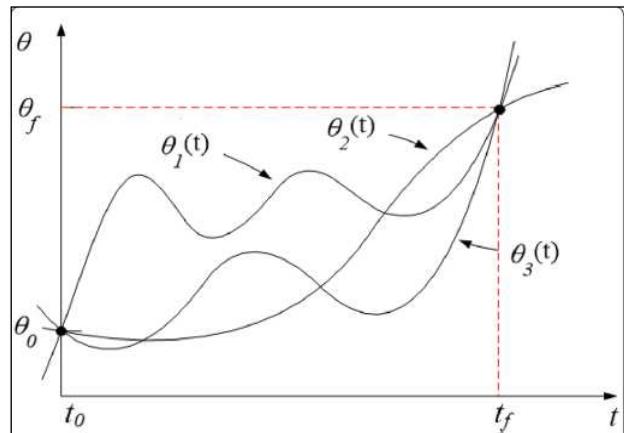


Fig.1. –Same starting and ending point of Quintic polynomial function[39]

The quintic polynomial general expression are,

$$\theta_i = a_0 + a_1 t + a_2 t^2 + a_3 t^3 + a_4 t^4 + a_5 t^5 \quad (3)$$

The trajectory is coarsely defined by the geometric path which describes the sequence of waypoints. The challenges includes shortest path and obstacle avoidance. The optimal trajectory planning shortest distance evaluation described in below figure 2 and similarly for obstacle avoidance the procedure described in the figure4.

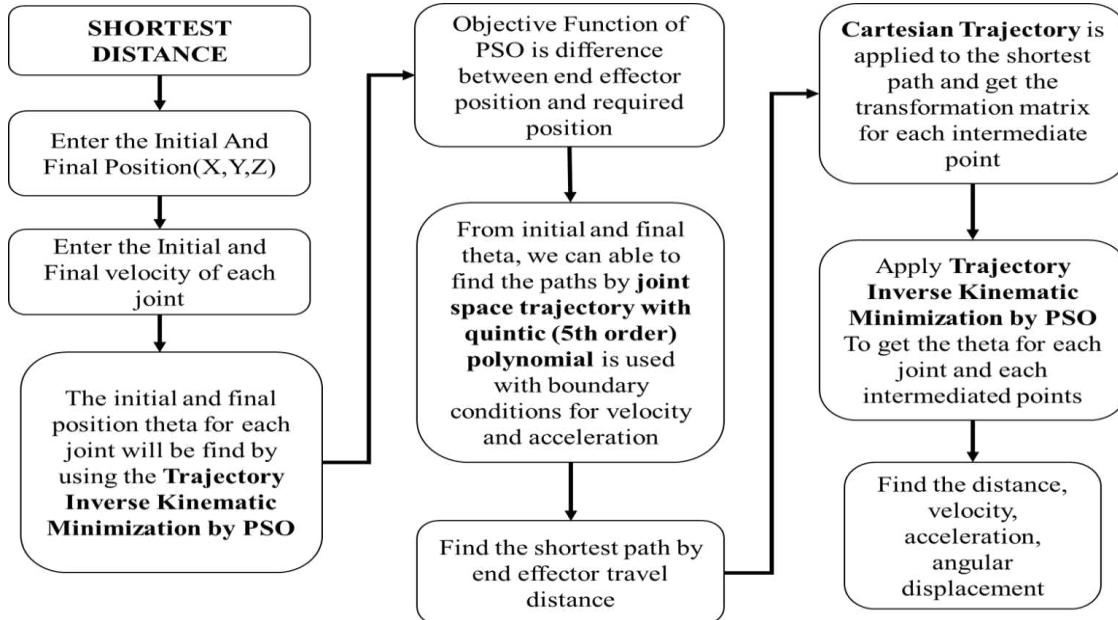


Fig.2. -Optimal Trajectory planning

3.1.3.1 Generating the trajectory:

The path planning provided the path which establishes the trajectory generator input. By polynomial class functions, the trajectory generator estimates waypoints of the preferred or desired path. Time based control sequence generated and moving the mobile platform or manipulator from the initial configuration to destination. The trajectory implicit definition is achieved by solving a problem with particular boundary conditions in given class of functions such as polynomials (quintic, cubic...), clothoids, sinusoids...etc.

Interpolation is defined as passage through points.

Initial, final and intermediate velocity or geometric tangent to the path.

Initial, final and intermediate velocity or geometric curvature.

Joint Space Trajectory

- A quintic (5th order) (from equation 6) polynomial is used boundary conditions for velocity and acceleration.
- Time is assumed to vary from 0 to 1 in intermediate steps.
- From initial point to final point the inverse kinematics solution is calculated.
- Using maximum velocities in joints, the total time is assigned.
- The individual joint trajectories discretized in time.

- Between these points, the continuous function blended.

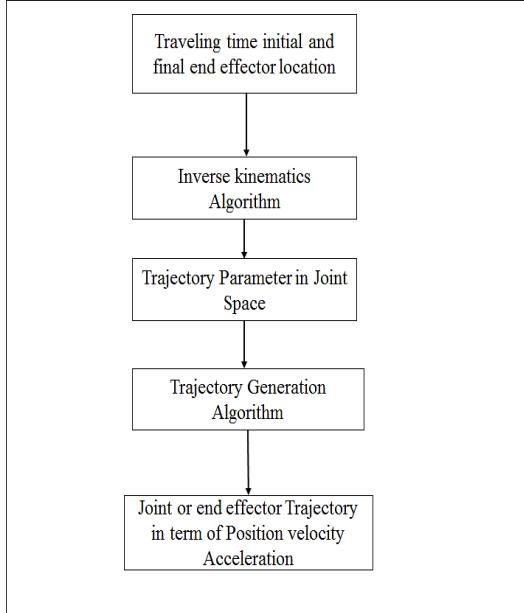


Fig.3. -Joint space Trajectory

The geometric path of joint space from above figure 3 in parametric form is,

$$q(\lambda) = (q_1(\lambda), q_2(\lambda), \dots, q_n(\lambda)) \quad (4)$$

Where, DOF = n and the motion law $S = S(t)$.

Cartesian Trajectory

- The Cartesian trajectory is a homogeneous transform sequence and the last subscript being the point index, that is, $T(:, i)$ is the i^{th} point along the path.
- The Cartesian trajectory is applied on the identified shortest path to find the intermediate points.

3.1.3.2 Particle Swarm Optimization (PSO)

The particle swarm optimization PSO is defines as computational technique which iteratively optimizes a problem described in below algorithm, and trying to improve the candidate solution with respect to quality measure. Hence, the best fitness value selected through the PSO technique. The major merits of PSO algorithm are easy to implement, simple concept, efficient computation, robustness. Due to these merits PSO algorithm is found to be better than the other traditional methods.

Algorithm I:PSO

1. Set parameter $W_{\text{minimum}}, W_{\text{maximum}}, C_1$ and C_2 PSO.
2. Initialize population of particles having positions X and velocities V
3. Set iteration $k = 1$
4. Calculate fitness of particles $F_i^k = f(X_i^k), \forall i$ and find the index of the best particle b
5. Select $P_{best_i}^k, \forall i$ and $Gbest^k = x_b^k$.
6. $\omega = \omega_{\text{maximum}} - k * (\omega_{\text{maximum}} - \omega_{\text{minimum}})/maxite$
7. Update velocity and position of particles

$$V_{i,j}^{k+1} = \omega * V_{i,j}^{k+1} + C_1 * rand() * (p_{best_i^k} - X_{ij}^k) + C_2 * rand() * G_{best_j^k} - X_{ij}^k; \forall i, \text{and } \forall j$$

$$X_{ij}^{k+1} = X_{ij}^{k+1} + V_{i,j}^{k+1}, \forall i, \text{and } \forall j$$

8. Evaluate $F_i^{k+1} = f(X_i^{k+1})$ $\forall i$ find index position b1.

9. Update Pbest of population $\forall i$

10. If $F_i^{k+1} < F_i^k$ then $P_{best_i}^k = X_i^{k+1}$ else $P_{best_i}^{k+1} = P_{best_i}^k$.

11. Update Gbest of population

$F_{b1}^{k+1} < F_i^k$ then $G_{best_i}^{k+1} = P_{best_i}^k$ and set b = b1 else $G_{best_i}^{k+1} = G_{best_i}^k$

12. If $k < maxite$ then $k = k + 1$ and goto step 6 else goto step 12

13. Print optimum solution as $G_{best_i}^k$

14. Inertial weight: 0.9 to 0.4

15. Acceleration factors (c1 and c2): 2 to 2.05

16. Population size: 10 to 100

17. Maximum iteration (Maxite): 500 to 10000

18. Initial velocity: 10 % of position

19. Where $P_{best} = \text{local best}$, $G_{best} = \text{global best}$,

The algorithm I shows the Particle Swarm Optimization where the initial parameters are set. Subsequently population of particles are initialized as position X and velocity V. Iteration is set to k equal to one. Fitness of particles is calculated based on step 4 to find the index of the best particle b. Then the velocity and position of particles are updated. Then the index position b1 is evaluated and updated. The major metrics of PSO algorithm that are used include inertial weight, acceleration factors, population size, maximum iteration and initial velocity. Thus the optimum solution is obtained through step 19.

Trajectory inverse kinematic minimization by PSO,

- Joint limits are considered in this solution.
- The solution returned depends on the initial choice of q0. The q0 is the search agent of PSO
- Works by minimizing the error between the forward kinematics of the joint angle solution and the end-effector frame as an optimization. The objective function (error) is described as:

$$\sum_q \sqrt{(T^{-1} * F_k - I(4)) * \omega} \quad (5)$$

Where omega is gain matrix

In the cartesian space,

$$p(s) = (x(s), y(s), z(s)), \text{ where the motion law is } s = s(t) \quad (6)$$

$$\text{Velocity } \dot{p}(t) = \frac{dp}{ds} s(t), \text{ acceleration } \ddot{p}(t) = \frac{dp}{ds} \ddot{s}(t) + \frac{d^2 p}{ds^2} \dot{s}(t) \quad (7)$$

Angular displacement = $\theta_2 - \theta_1$

(change in angle), where θ_2 is the current angle and θ_1 is the previous angle.

3.1.4 Obstacle avoidance:

The obstacle avoidance is subjected to non collision or non intersection position constraints by some control objectives. Generally the obstacle avoidance is different from path planning which involves reactive control law and whereas other involves the obstacle free path precomputation in which a controller guides the robot. The proposed obstacle avoidance has bee shown in the figure with explanation.

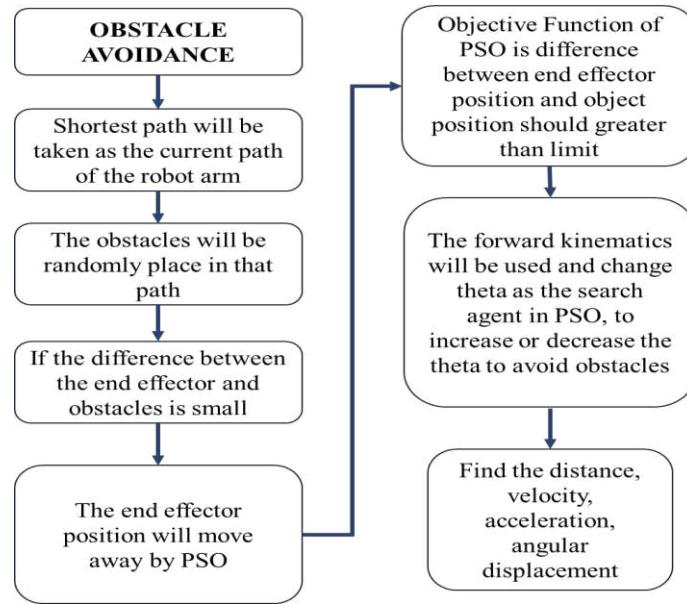
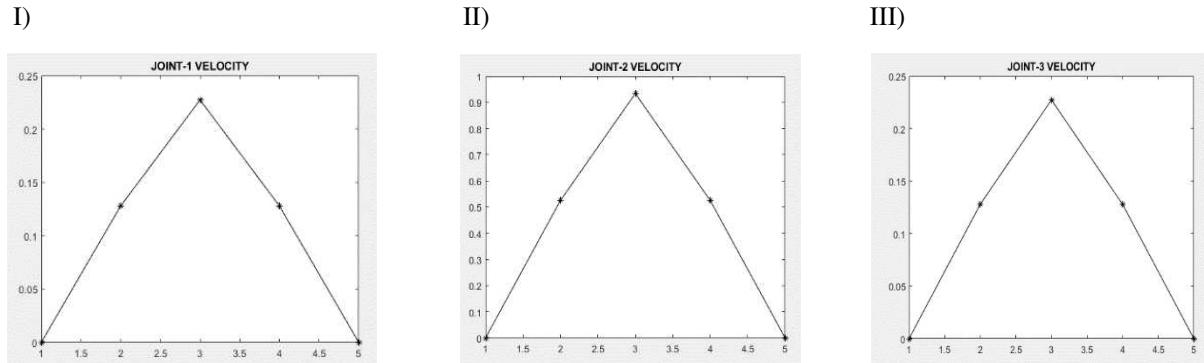


Fig.4. -Obstacle avoidance

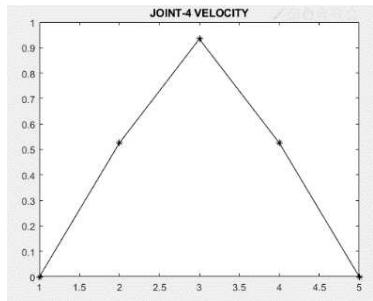
IV. RESULTS AND DISCUSSION

4.1 Evaluation results for ROBOT 1:

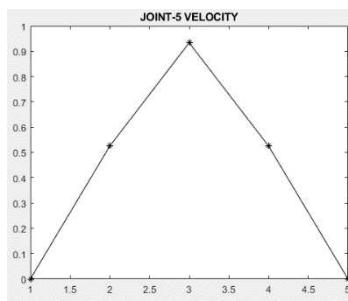
VELOCITY



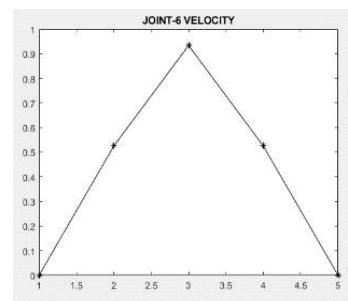
IV)



V)



VI)

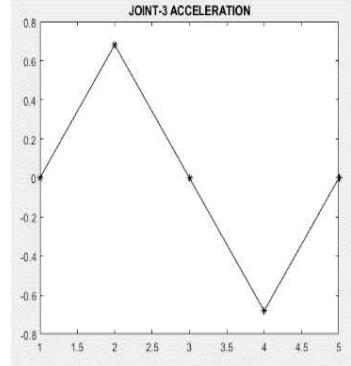
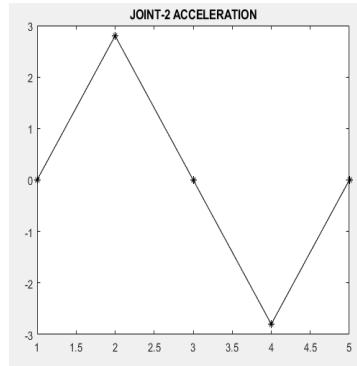
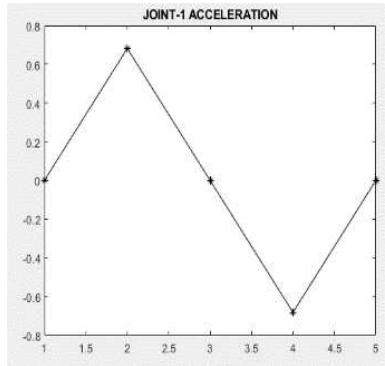


ACCELERATION

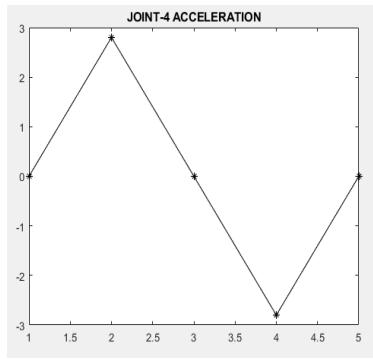
I)

II)

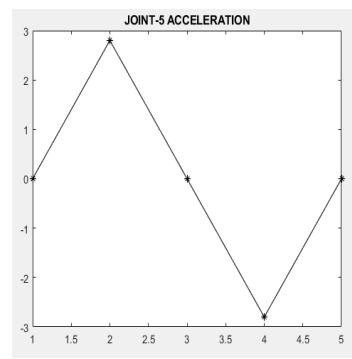
III)



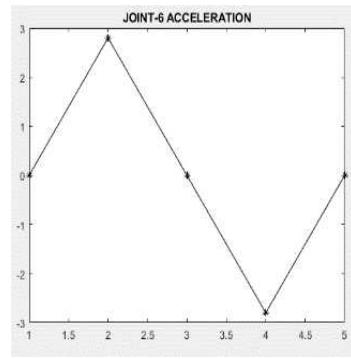
IV)



V)

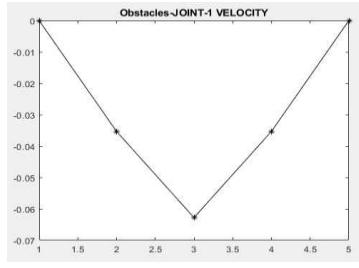


VI)

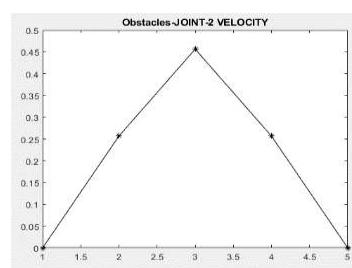


OBSTACLE/VELOCITY

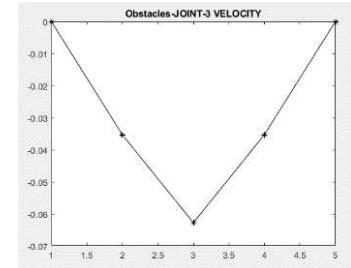
I)



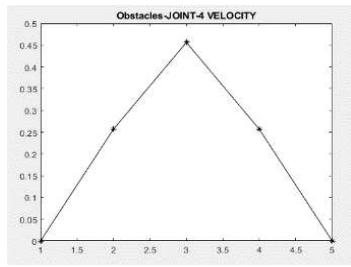
II)



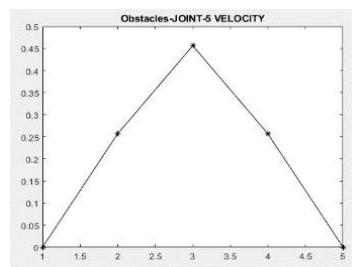
III)



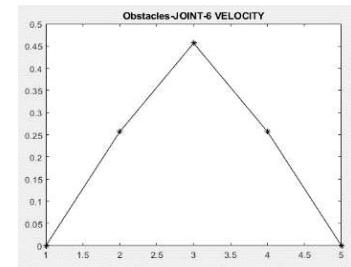
IV)



V)



VI)

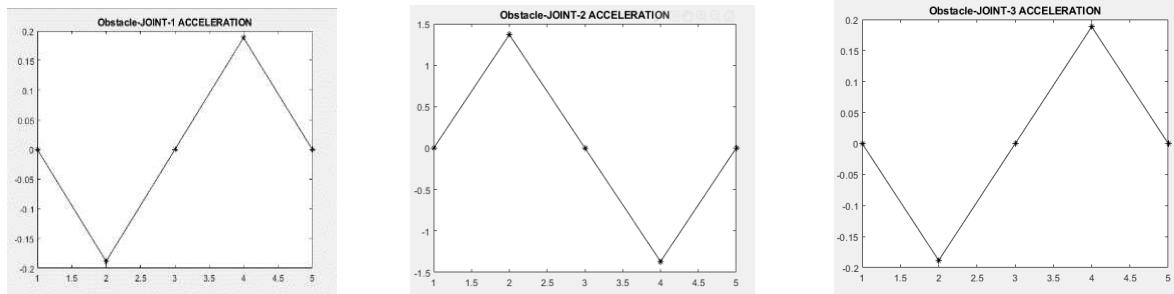


OBSTACLE/ACCELERATION

I)

II)

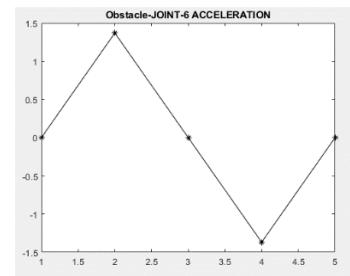
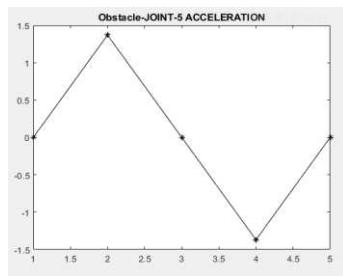
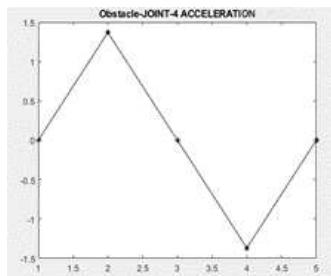
III)



IV)

V)

VI)



From the above results, it clearly depicted in figures about the six joints velocity, six joints acceleration of the optimal trajectory and it shows the respected path diagrammatically. The obstacle velocity and acceleration of the respected joints also depicted above. From Joint 1 to Joint 6 the velocity V is calculated as,

$$V = \frac{\Delta d}{\Delta t} \quad (8)$$

From Eq. (10), Δd is the distance and Δt is the time. From the I position, the starting position is taken as 0 and velocity goes to max. to ending position 0. The values can be of any number. Similarly from Joint 1 to Joint 6 the acceleration is calculated as,

$$A = \frac{\Delta v}{\Delta t} \quad (9)$$

From Eq. (11), Δv is the velocity and Δt is the time. We set the clock at 0 and ending position goes to 0. When the velocity vary the acceleration also varies. Acceleration depends on the velocity value which is set manually.

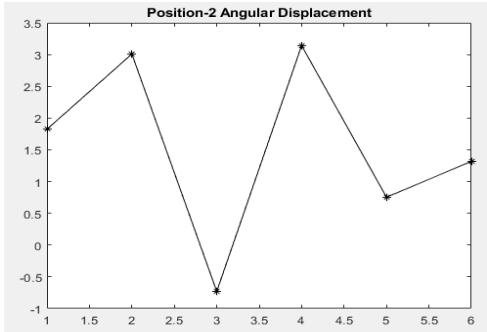
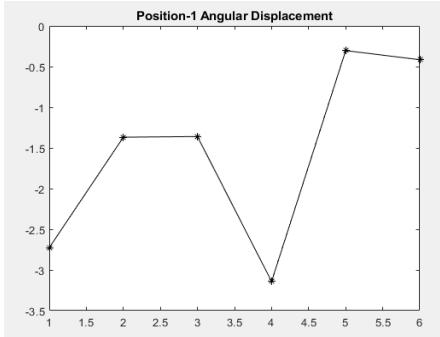
In case of obstacle velocity and obstacle acceleration the same formula in Eq. (8) and Eq. (9) is applied. To avoiding the obstacles the different paths are identified.

ANGULAR DISPLACEMENT

Robot 1 angular displacement

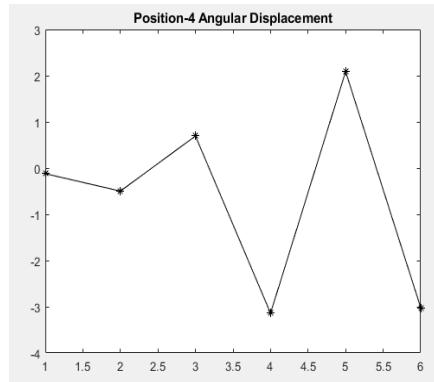
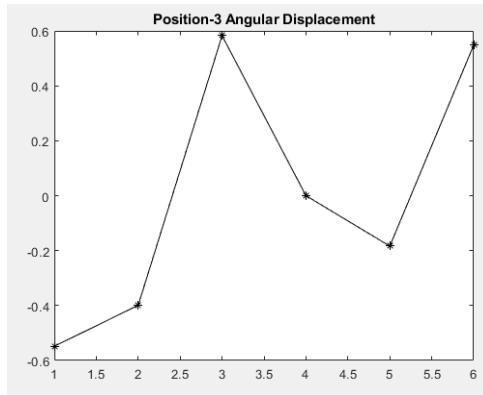
I)

II)



III)

IV)

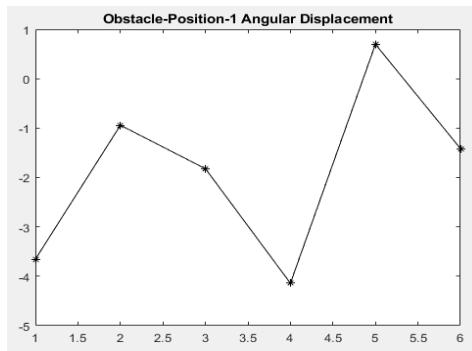


The above figures depicted the angular displacement with respect to the four positions on the trajectory path. Here the x-axis shows the Joint 1, Joint 2 ...Joint 6 whereas the y- axis is the angular displacement. The angular displacement is measured as difference between the previous θ and the present θ for Joint 1... Joint 6 for 6-DOF.

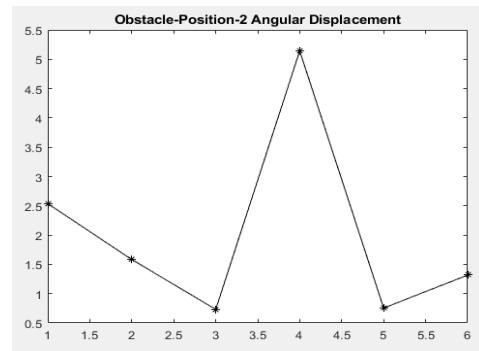
OBSTACLE / ANGULAR DISPLACEMENT

Robot 1 obstacle angular displacement

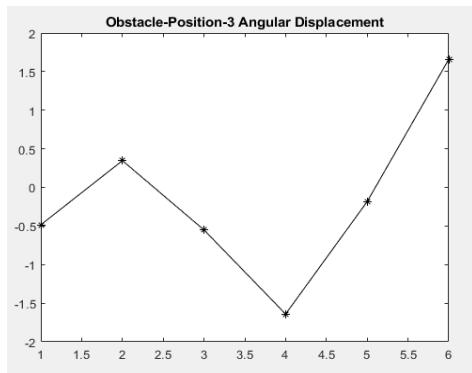
I)



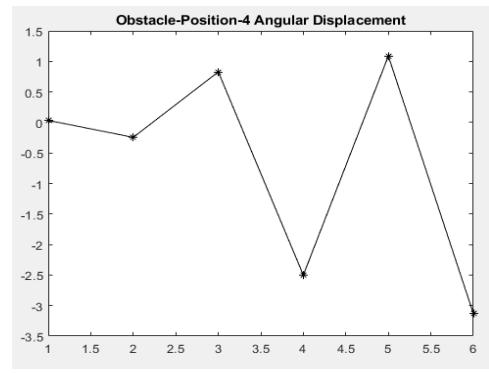
II)



III)



IV)



The above figures depicted the obstacle angular displacement with respect to the four positions on the trajectory path. Similarly for obstacle angular displacement also the same measurement is followed as difference between the previous θ and the present θ for Joint 1... Joint 6 for 6-DOF.

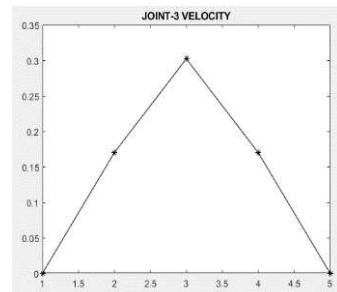
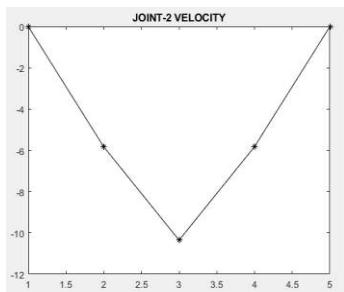
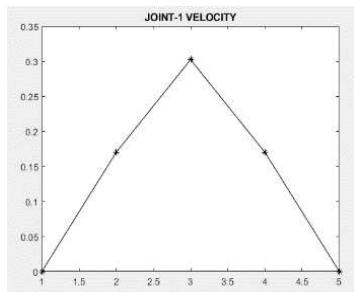
4.2 Evaluation results for Robot 2:

VELOCITY

I)

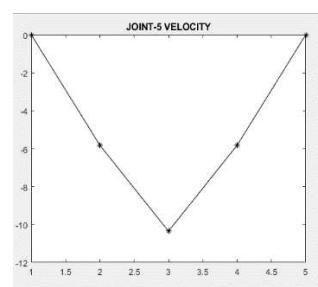
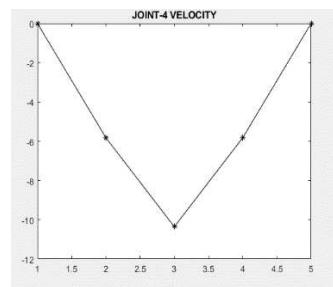
II)

III)



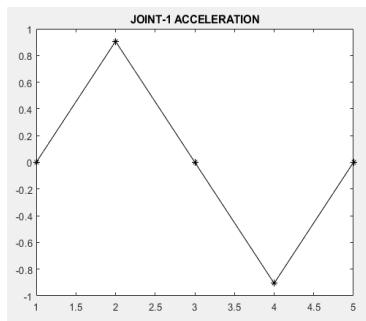
IV)

V)

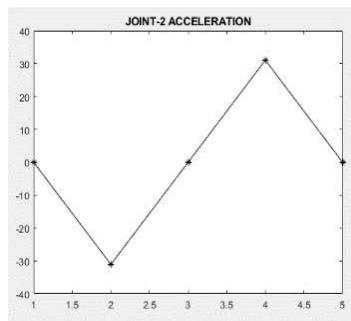


ACCELERATION

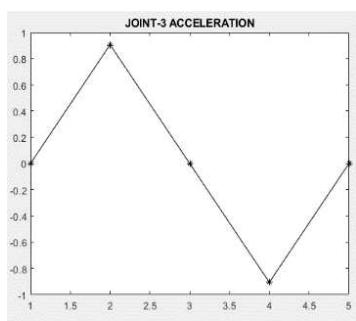
I)



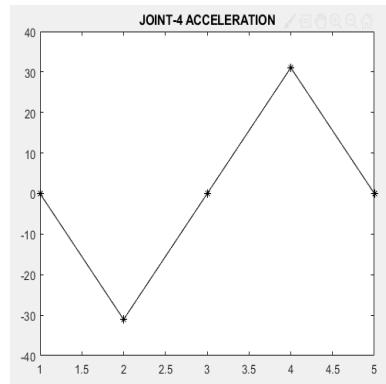
II)



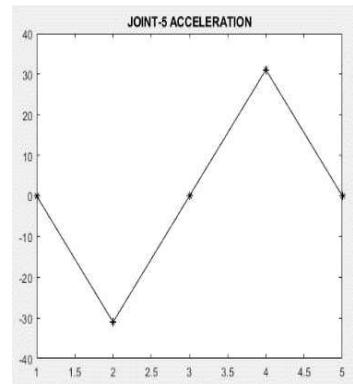
III)



IV)



V)

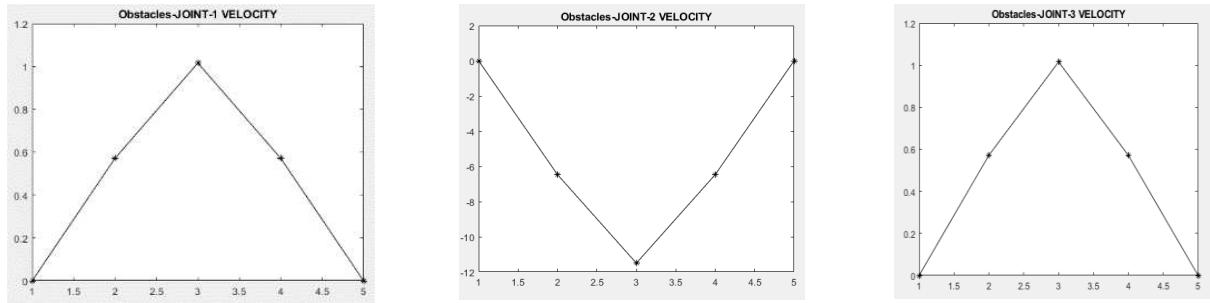


OBSTACLE/VELOCITY

I)

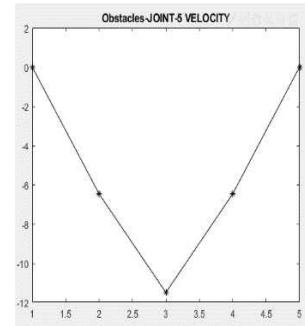
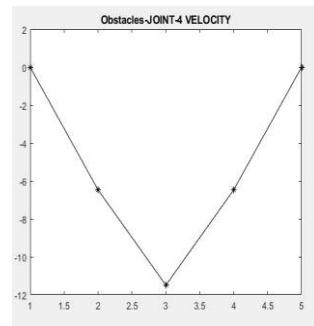
II)

III)



IV)

V)

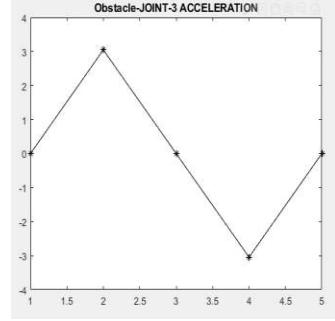
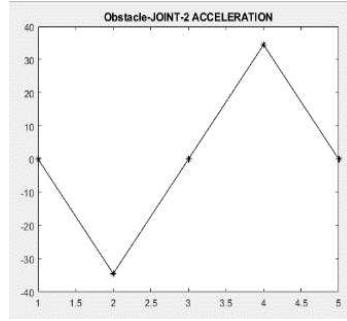
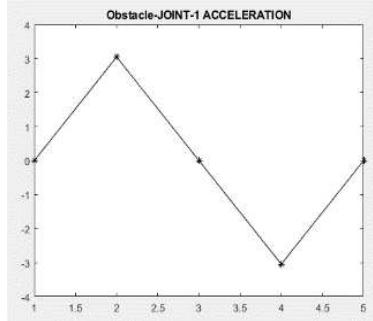


OBSTACLE/ACCELERATION

I)

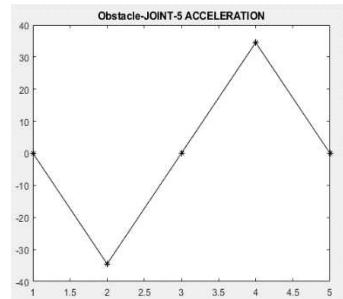
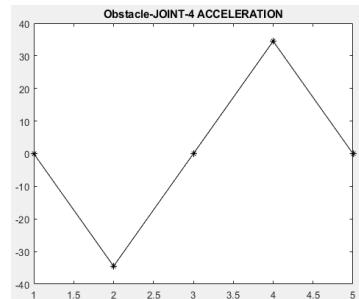
II)

III)



IV)

V)



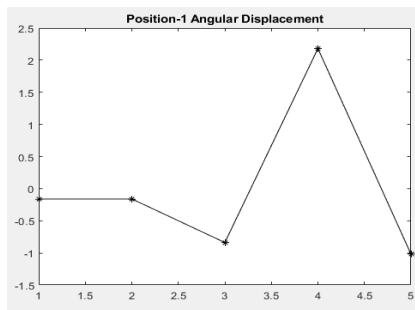
From the above results, it clearly depicted in figures about the six joints velocity, six joints acceleration of the optimal trajectory and it shows the respected path diagrammatically. The obstacle velocity and acceleration of the respected joints also depicted above. From Joint 1 to Joint 6 the velocity V is calculated shown in Eq. (8). From the 1st position- starting position is taken as 0 and velocity goes to max. to ending position 0. The values can be

of any number. Similarly from Joint 1 to Joint 5 the acceleration is calculated by Eq. (9). Similarly, for 5-DOF also we set the clock at 0 and ending position goes to 0. When the velocity vary the acceleration also varies. Acceleration depends on the velocity value which is set manually. In case of obstacle velocity and obstacle acceleration the same formula in Eq. (8) and Eq. (9) is applied. To avoiding the obstacles the different paths are identified.

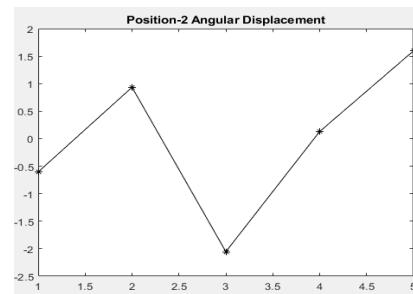
ANGULAR DISPLACEMENT

Robot 2 angular displacement

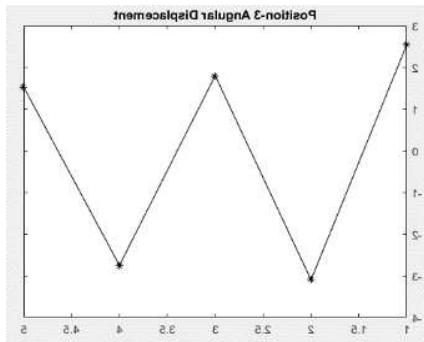
I)



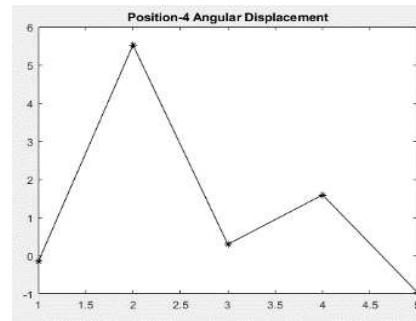
II)



III)



IV)



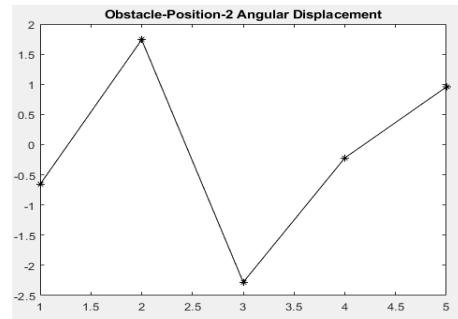
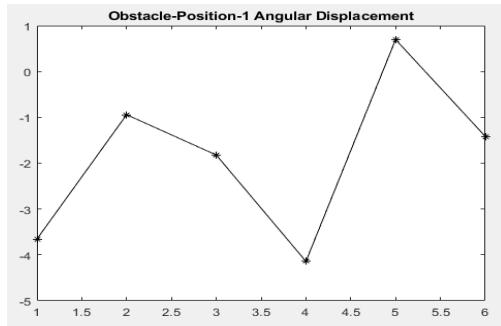
The above figures depicted the angular displacement with respect to the four positions on the trajectory path. Here the x-axis shows the Joint 1, Joint 2 ...Joint 5 whereas the y- axis is the angular displacement. The angular displacement is measured as difference between the previous θ and the present θ for Joint 1 ... Joint 5 for 5-DOF.

OBSTACLE / ANGULAR DISPLACEMENT

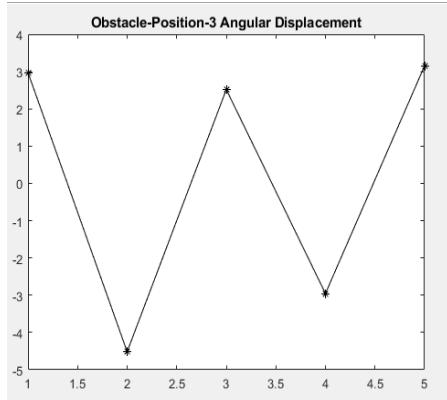
Table 6- Robot 2 obstacle angular displacement

I)

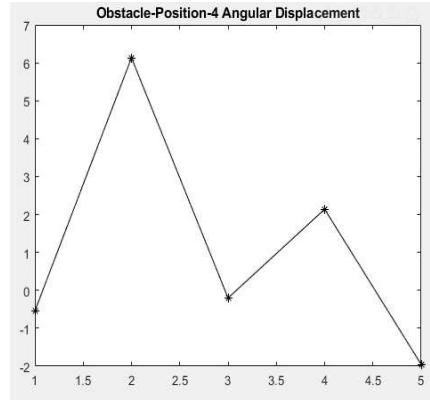
II)



III)



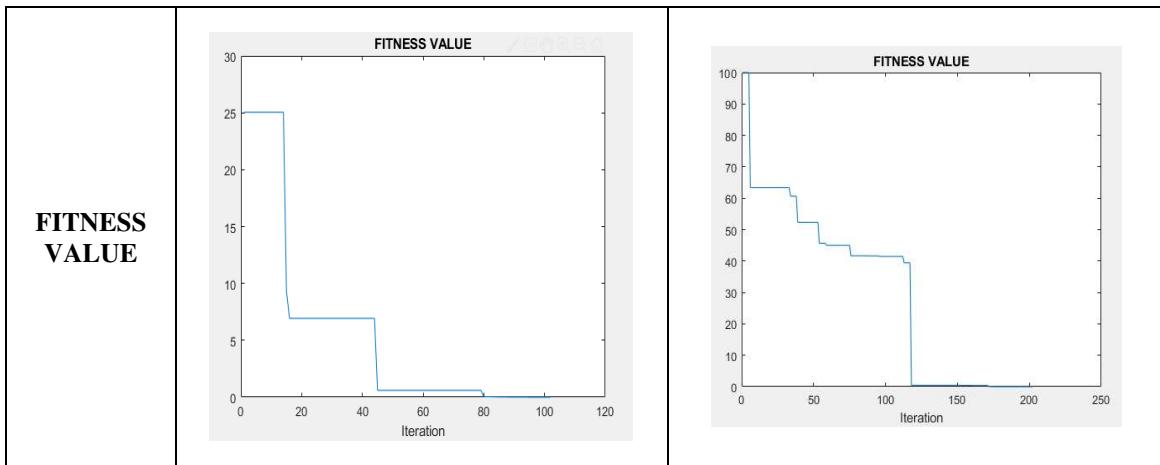
IV)



The above figures depicted the obstacle angular displacement with respect to the four positions on the trajectory path. Similarly for obstacle angular displacement also the same measurement is followed as difference between the previous θ and the present θ for Joint 1 ... Joint 5 for 5-DOF.

Table 1- Position error and Fitness value for Robot 1 and Robot 2

| PARAMETERS | ROBOT 1 | ROBOT 2 |
|----------------|---|--|
| POSITION ERROR | <p>POSITION ERROR</p> <p>This graph shows the position error over 120 iterations. Three lines are plotted: a blue line starting at ~25 and dropping to near 0 by iteration 40; an orange line starting at ~25 and dropping to near 0 by iteration 40; and a yellow line starting at ~25 and dropping to near 0 by iteration 40.</p> | <p>POSITION ERROR</p> <p>This graph shows the position error over 250 iterations. Three lines are plotted: a blue line starting at ~130 and dropping to near 0 by iteration 100; an orange line starting at ~130 and dropping to near 0 by iteration 100; and a yellow line starting at ~40 and dropping to near 0 by iteration 100.</p> |



From the table 1, it clearly depicted the position error and fitness value of the respected Robot 1 and Robot 2. For any 2, 3, 4 position's fitness value of the x, y and z axis are measured. The position error is measured from difference between the Robot's current position and expected position. The position error should be 0 and in this proposed study the position error value 0 attains at 80 – 100 iterations. The average position error is termed as fitness value. Similarly for fitness value it must be 0 and attains at same number of iterations.

Table 2- Shortest path evaluation (Robot 1)

| Path 1 | Path 2 | Path 3 | Path 4 | Shortest time | Collision-time | Shortest-Path | Obstacles path |
|--------|--------|--------|--------|---------------|----------------|---------------|----------------|
| 1.1250 | 0.8750 | 0.8750 | 1.1250 | 4.0000 | 5.4360 | 80.3985 | 109.2624 |

From table 2, for robot 1, the shortest path is from 1 to 5 position and hence there are 4 paths are seen. The time duration for shortest path is 4 secs, the collision time is 5.4 secs. The distance travelled by robot from 1 to 5 the shortest path is 80.3cm whereas the obstacles is 109.26 cm.

For Robot 2,

Table 3- Shortest Path evaluation (Robot 2)

| Path 1 | Path 2 | Path 3 | Path 4 | Shortest time | Collision-time | Shortest-Path | Obstacles path |
|--------|--------|--------|--------|---------------|----------------|---------------|----------------|
| 0.9375 | 0.7292 | 0.7292 | 0.9375 | 3.3333 | 3.5813 | 252.4372 | 271.2138 |

From the table 3, it shows path distance, shortest path distance and obstacles path distance for Robot 1 and Robot 2 which are measured in terms of cm. the shortest path time and collision time evaluated in terms of secs. For robot 2, the shortest path is from 1 to 5 position and hence there are 4 paths are seen. The time duration for shortest path is 3.33 secs, the collision time is 3.5 secs. The distance travelled by robot from 1 to 5 the shortest path is 252.43 cm whereas the obstacles is 271.21 cm.

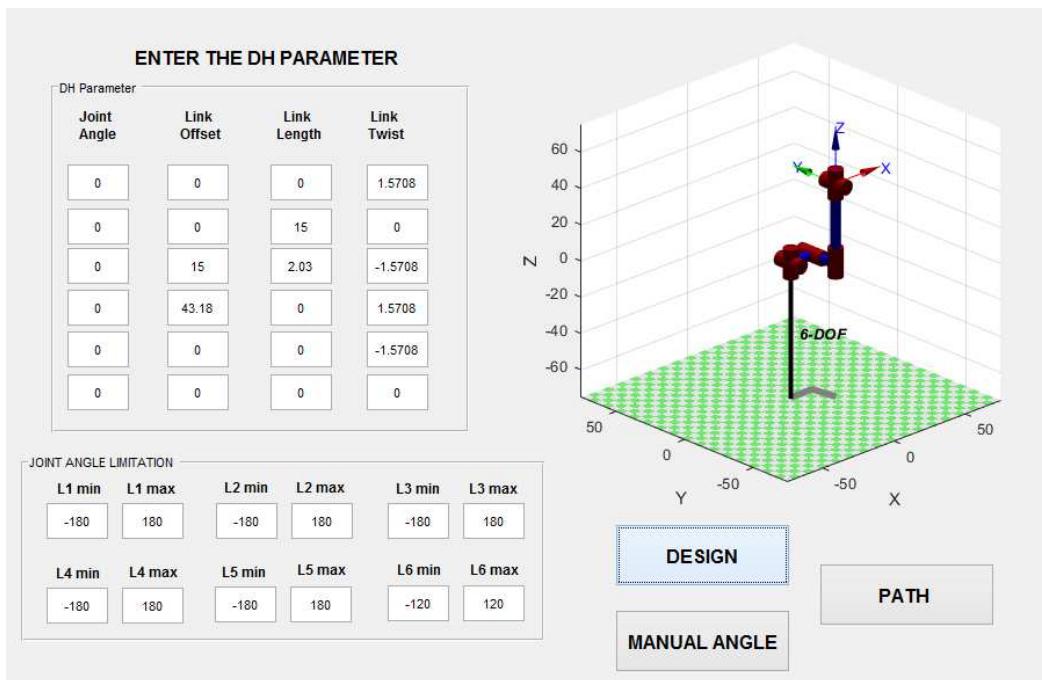


Fig.5. -6-DOF design

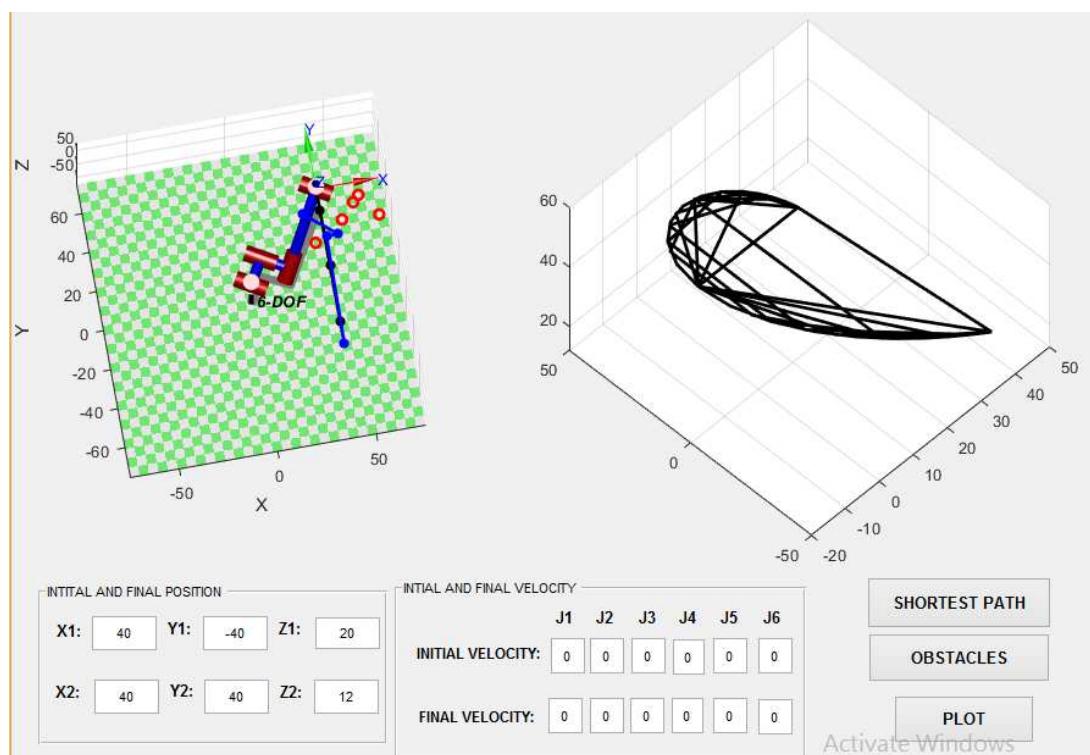


Fig.6. -6-DOF shortest path and obstacles

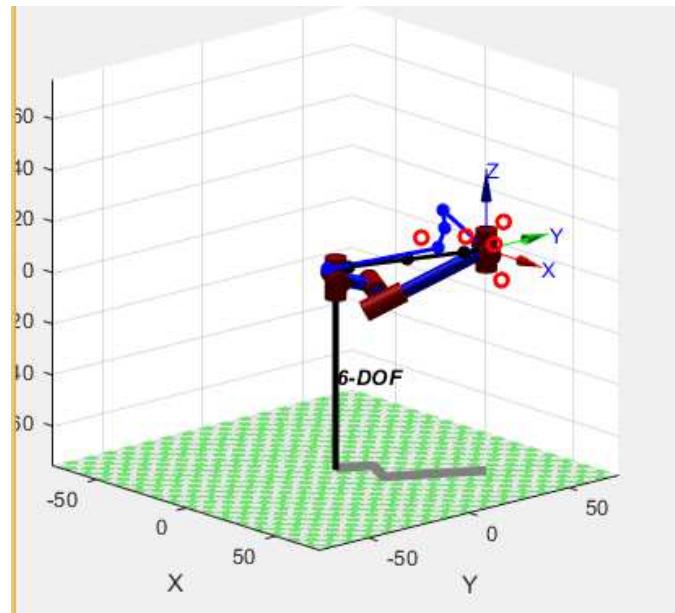


Fig.7. -6-DOF obstacle path

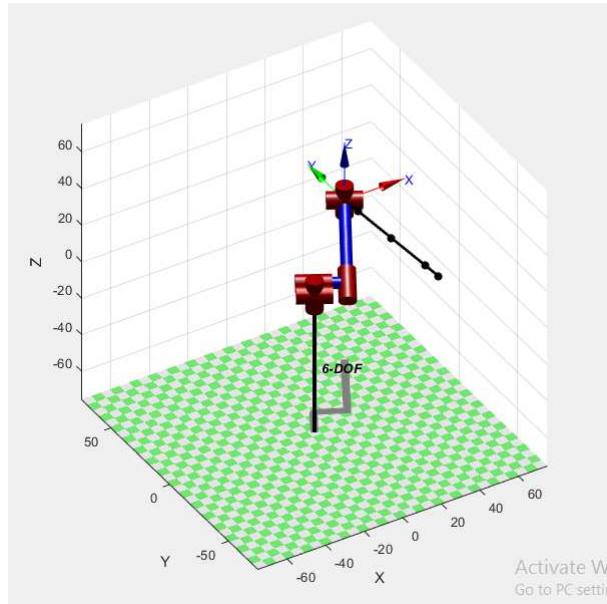


Fig.8. -6-DOF shortest path

From figure 5 to 8, the 6-DOF robotic design and its shortest path and obstacles are depicted. The DH parameters values are shown with x, y and z plane. The initial and final position value of shortest path and shortest path with obstacles are shown in above figures. Any 6-DOF robot design is based on the proposed PSO algorithm and thus it can be run in optimized way. We can set the joint angle limitation for each joint and we can check with manual angle to check its movement. From all the above DOF design figures the black colour shows the shortest path, blue colour shows the obstacle path and red colour shows the obstacles path. The number of paths are identified and from that the shortest path are found and shown in fig 9. Similarly the obstacle path are also shown in response to obstacles avoidance is depicted.

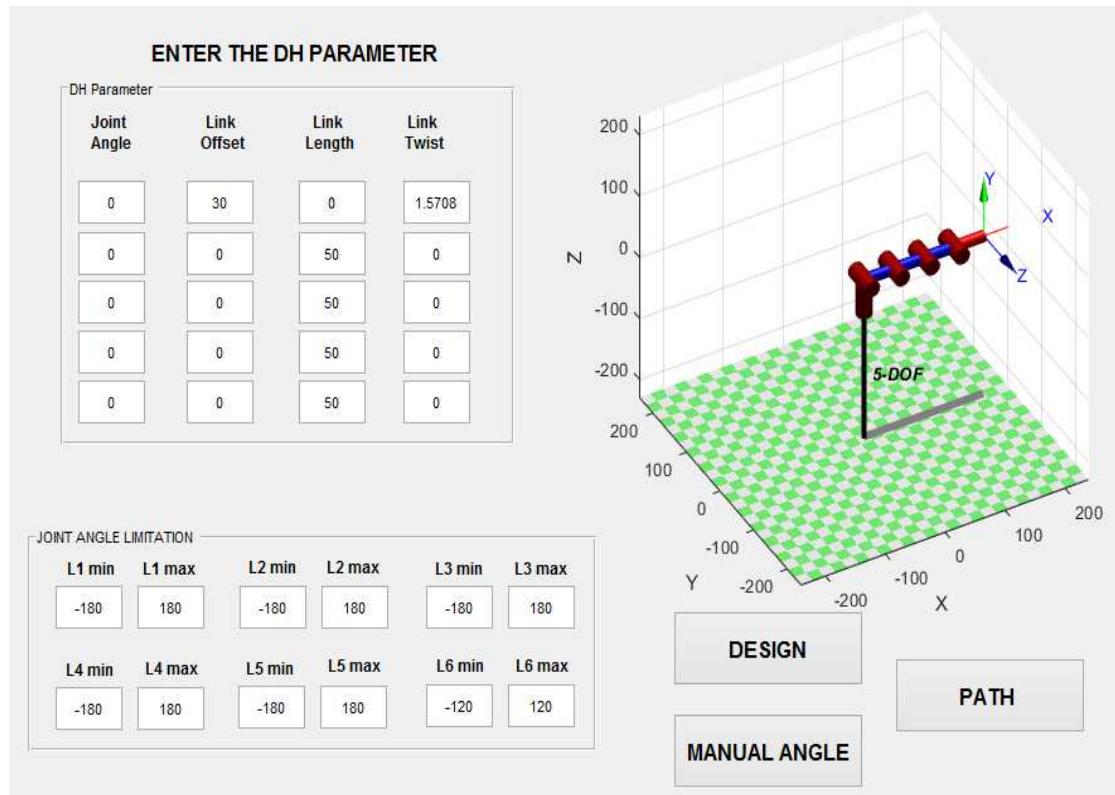


Fig.9. -5-DOF design

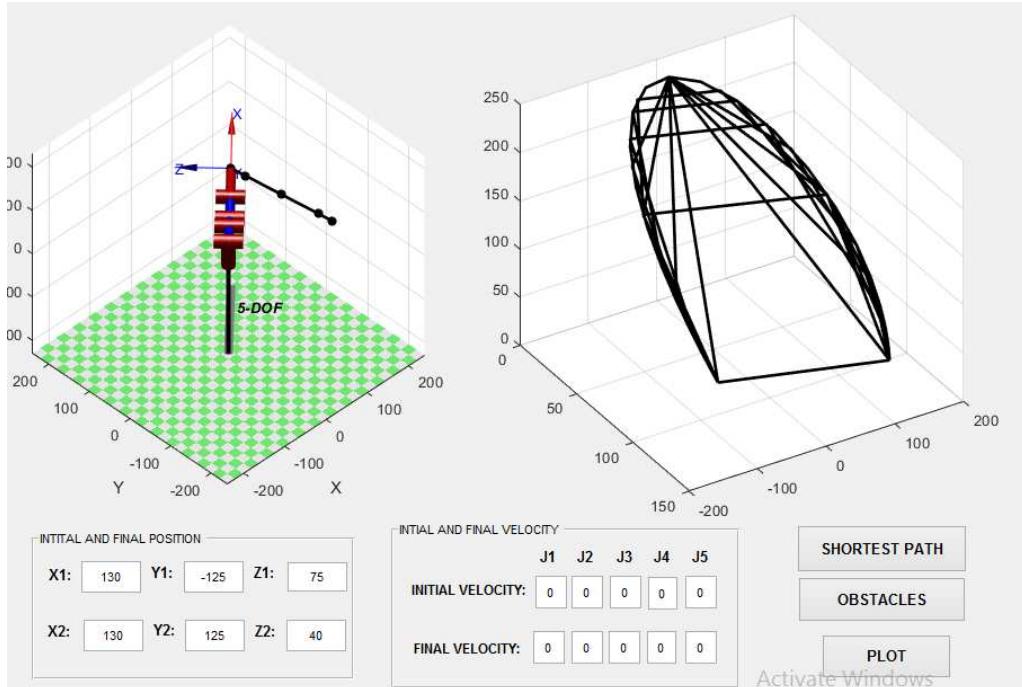


Fig.10. -5-DOF shortest path

From figure 9 to 11, the 5-DOF robotic design and its shortest path and obstacles are depicted. The DH parameters values are shown with x, y and z plane. The initial and final position value of shortest path and shortest path with obstacles are shown in above figure. Any 5-DOF robot design is based on the proposed PSO algorithm and thus it can be run in optimized way. We can set the joint angle limitation for each joint and we can check with manual angle to check its movement. From all the above DOF design figures the black colour shows the shortest path, blue colour shows the obstacle path and red colour shows the obstacles path. The number of paths are identified

and from that the shortest path are found and shown in fig 10. Similarly the obstacle path are also shown in figure 11 which is the response to obstacles avoidance is depicted.

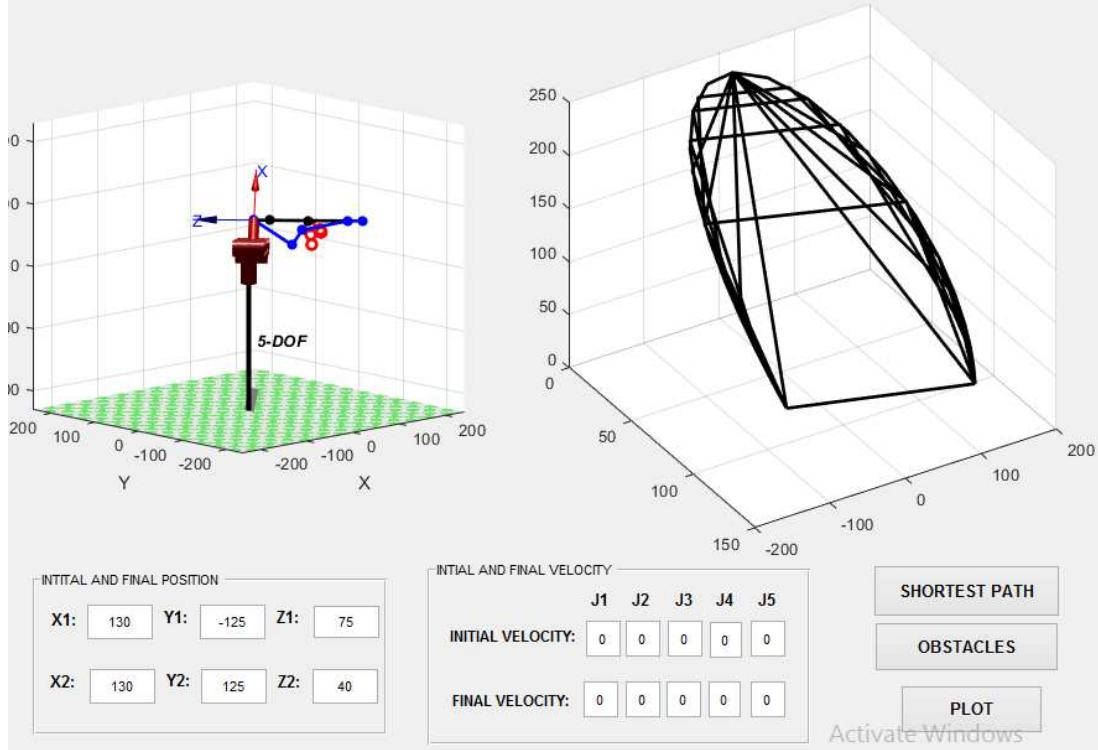


Fig.11. -5-DOF obstacle avoidance path

Hence, by using the optimization algorithm implemented on both joint space trajectory and Cartesian trajectory the highly accurate result is obtained specifically in both shortest path and obstacle avoidance path.

COMPARATIVE ANALYSIS

The fitness value obtained from the proposed Robotic manipulator 5-DOF and 6-DOF is compared with existing robot manipulator fitness value [40]. In the existing study the 3-DOF robot manipulator is utilized and obtained fitness value at 100 iterations. The comparative results shows that the fitness value obtained rapidly in 100-150 iterations in both 5-DOF and 6-DOF compared with existing 3-DOF robot manipulator. Hence the proposed method using the PSO optimisation algorithm shows better performance than the existing method.

Table 4- Fitness value of the existing [40] and proposed system

| Fitness Value(Existing) [40] | Fitness value(Proposed) |
|------------------------------|-------------------------|
| 0.08566 | 0.086 |
| 0.125 | 0.126 |
| 0.0708 | 0.072 |

Table 5-Six joints Velocity {Alam, 2016 #2334}

| Joints | Time | 0 | 1 | 2 | 3 | 4 |
|---------|----------|---|------|------|------|------|
| Joint 1 | Proposed | 0 | 0 | 0.75 | 0.23 | 0.75 |
| | Existing | 0 | -0.2 | 0.4 | 0.4 | 0 |
| Joint 2 | Proposed | 0 | 0 | 0.52 | 0.92 | 0.52 |
| | Existing | 0 | -0.5 | 0.5 | 0.4 | 0 |
| Joint 3 | Proposed | 0 | 0 | 0.75 | 0.23 | 0.75 |

| | | | | | | |
|---------|----------|---|----|------|------|------|
| | Existing | 0 | 0 | 0.2 | 0.2 | 0 |
| Joint 4 | Proposed | 0 | 0 | 0.52 | 0.92 | 0.52 |
| | Existing | 0 | -1 | 0 | -1 | 0 |
| Joint 5 | Proposed | 0 | 0 | 0.52 | 0.92 | 0.52 |
| | Existing | 0 | 1 | 0 | -4 | 0 |
| Joint 6 | Proposed | 0 | 0 | 0.5 | 0.92 | 0.5 |
| | Existing | 0 | 0 | -2.5 | -1 | 0 |

Table 6- Six joints Acceleration {Alam, 2016 #2334}

| Joints | Time | 0 | 1 | 2 | 3 | 4 |
|---------|----------|---|----|-----|------|------|
| Joint 1 | Proposed | 0 | 0 | 0.7 | -0.1 | -0.7 |
| | Existing | 0 | 0 | -2 | -2 | 0 |
| Joint 2 | Proposed | 0 | 0 | 2.8 | -0.2 | -2.8 |
| | Existing | 0 | -1 | 0 | -2 | 0 |
| Joint 3 | Proposed | 0 | 0 | 0.7 | -0.1 | -0.7 |
| | Existing | 0 | -2 | 0 | -2 | 0 |
| Joint 4 | Proposed | 0 | 0 | 2.8 | -0.2 | -2.8 |
| | Existing | 0 | 2 | -2 | -4 | 2 |
| Joint 5 | Proposed | 0 | 0 | 2.8 | -0.2 | -2.8 |
| | Existing | 0 | 0 | 2.5 | -0.2 | 0 |
| Joint 6 | Proposed | 0 | 0 | 0.5 | 0.92 | 0.5 |
| | Existing | 0 | 0 | 2.8 | -0.8 | -2.8 |

From the above table-5 and table-6, it is clearly depicted about the six joints velocity and six joints acceleration of the optimal trajectory and it explores the corresponding path in a table format. Thus through the proposed system, the velocity and acceleration has increased thereby increasing the performance of the system.

V. CONCLUSION

This article developed to frame the inverse and forward kinematic model of 5 DOF and 6 DOF robot manipulator. The trajectory inverse kinematics minimization has been the main objective in this study and it has been achieved by Particle Swarm Optimization PSO. A quintic 5th order polynomial with joint space trajectory used and Cartesian Trajectory is applied to the shortest path and get the transformation matrix for each intermediate point. Finally the obstacle avoidance has been exhibited by PSO. For both robots 1 and 2 the distant velocity, angular displacement and acceleration have been evaluated in terms of shortest paths and obstacle path. The current study has been compared with existing study of robot manipulator and fitness evaluated and shows effective performance. Thus velocity and acceleration increased through the proposed system.

REFERENCES

- [1] Z. Li, G. Li, Y. Sun, G. Jiang, J. Kong, and H. Liu, "Development of articulated robot trajectory planning," *International Journal of Computing Science and Mathematics*, vol. 8, pp. 52-60, 2017.
- [2] A. Reiter, A. Müller, and H. Gatringer, "On higher order inverse kinematics methods in time-optimal trajectory planning for kinematically redundant manipulators," *IEEE Transactions on Industrial Informatics*, vol. 14, pp. 1681-1690, 2018.
- [3] P. Corke, "Robot arm kinematics," in *Robotics, Vision and Control*, ed: Springer, 2017, pp. 193-228.
- [4] K. Mohamed, H. Elgamal, and A. Elsharkawy, "Dynamic analysis with optimum trajectory planning of multiple degree-of-freedom surgical micro-robot," *Alexandria engineering journal*, vol. 57, pp. 4103-4112, 2018.
- [5] X. Wang, W. Sun, E. Li, and X. Song, "Energy-minimum optimization of the intelligent excavating process for large cable shovel through trajectory planning," *Structural and Multidisciplinary Optimization*, vol. 58, pp. 2219-2237, 2018.

- [6] A. Jain, D. Chen, D. Bansal, D. Kent, H. Ravichandar, and S. Chernova, "Anticipatory Human-Robot Collaboration via Multi-Objective Trajectory Optimization," *arXiv preprint arXiv:2006.03614*, 2020.
- [7] X. Gao, Y. Mu, and Y. Gao, "Optimal trajectory planning for robotic manipulators using improved teaching-learning-based optimization algorithm," *Industrial Robot: An International Journal*, 2016.
- [8] F. Tohfeh and A. Fakharian, "Polynomial based optimal trajectory planning and obstacle avoidance for an omni-directional robot," in *2015 AI & Robotics (IRANOPEN)*, 2015, pp. 1-6.
- [9] C.-J. Lin, T.-H. S. Li, P.-H. Kuo, and Y.-H. Wang, "Integrated particle swarm optimization algorithm based obstacle avoidance control design for home service robot," *Computers & Electrical Engineering*, vol. 56, pp. 748-762, 2016.
- [10] Y. Cai and S. X. Yang, "A PSO-based approach with fuzzy obstacle avoidance for cooperative multi-robots in unknown environments," *International Journal of Computational Intelligence and Applications*, vol. 15, p. 1650001, 2016.
- [11] C.-T. Yen and M.-F. Cheng, "A study of fuzzy control with ant colony algorithm used in mobile robot for shortest path planning and obstacle avoidance," *Microsystem Technologies*, vol. 24, pp. 125-135, 2018.
- [12] K. Zheng, Y. Hu, and B. Wu, "Trajectory planning of multi-degree-of-freedom robot with coupling effect," *Journal of Mechanical Science and Technology*, vol. 33, pp. 413-421, 2019.
- [13] C. Klug, D. Schmalstieg, T. Gloor, and C. Arth, "A complete workflow for automatic forward kinematics model extraction of robotic total stations using the Denavit-Hartenberg convention," *Journal of Intelligent & Robotic Systems*, vol. 95, pp. 311-329, 2019.
- [14] T. P. Valayil, V. Selladurai, and N. R. Ramaswamy, "Kinematic modeling of a serial robot using Denavit-Hartenberg method in Matlab," (*TAGA Journal of graphic technology*, vol. 14, pp. 2347-2445, 2018.
- [15] H. Z. Khaleel, "Enhanced Solution of Inverse Kinematics for Redundant Robot Manipulator Using PSO," *Engineering and Technology Journal*, vol. 37, pp. 241-247, 2019.
- [16] M. Wang, J. Luo, J. Yuan, and U. Walter, "Coordinated trajectory planning of dual-arm space robot using constrained particle swarm optimization," *Acta Astronautica*, vol. 146, pp. 259-272, 2018.
- [17] L. Wang, Q. Wu, F. Lin, S. Li, and D. Chen, "A new trajectory-planning beetle swarm optimization algorithm for trajectory planning of robot manipulators," *IEEE access*, vol. 7, pp. 154331-154345, 2019.
- [18] B. Salamat and A. M. Tonello, "Stochastic trajectory generation using particle swarm optimization for quadrotor unmanned aerial vehicles (UAVs)," *Aerospace*, vol. 4, p. 27, 2017.
- [19] T. Collins and W.-M. Shen, "PASO: An Integrated, Scalable PSO-based Optimization Framework for Hyper-Redundant Manipulator Path Planning and Inverse Kinematics.",," *Information Sciences Institute Technical Report*, 2016.
- [20] G. Perumalsamy, P. Visweswaran, J. Jose, S. J. Winston, and S. Murugan, "Quintic Interpolation Joint Trajectory for the Path Planning of a Serial Two-Axis Robotic Arm for PFBR Steam Generator Inspection," in *Machines, Mechanism and Robotics*, ed: Springer, 2019, pp. 637-648.
- [21] A. Machmudah, S. Parman, and M. Baharom, "Continuous path planning of kinematically redundant manipulator using particle swarm optimization," *International Journal of Advanced Computer Science and Applications*, vol. 9, pp. 207-217, 2018.
- [22] S. Kucuk, "Optimal trajectory generation algorithm for serial and parallel manipulators," *Robotics and Computer-Integrated Manufacturing*, vol. 48, pp. 219-232, 2017.
- [23] Z. Liu, J. Xu, C. Yang, Y. Zhao, and T. Zhang, "A TE-E optimal planning technique based on screw theory for robot trajectory in workspace," *Journal of Intelligent & Robotic Systems*, vol. 91, pp. 363-375, 2018.
- [24] S. Dereli and R. Köker, "IW-PSO approach to the inverse kinematics problem solution of a 7-DOF serial robot manipulator," *Sigma Journal of Engineering and Natural Sciences*, vol. 36, pp. 77-85, 2018.
- [25] S. Macfarlane and E. A. Croft, "Jerk-bounded manipulator trajectory planning: design for real-time applications," *IEEE Transactions on robotics and automation*, vol. 19, pp. 42-52, 2003.
- [26] S. A. Ali, K. A. M. Annuar, and M. F. Miskon, "Trajectory planning for exoskeleton robot by using cubic and quintic polynomial equation," *International Journal of Applied Engineering Research*, vol. 11, pp. 7943-7946, 2016.
- [27] M. Chen, K. Zhang, S. Wang, F. Liu, J. Liu, and Y. Zhang, "Analysis and Optimization of Interpolation Points for Quadruped Robots Joint Trajectory," *Complexity*, vol. 2020, 2020.
- [28] R. L. Andersson, "Aggressive trajectory generator for a robot ping-pong player," *IEEE Control Systems Magazine*, vol. 9, pp. 15-21, 1989.
- [29] S. Lu, B. Ding, and Y. Li, "Minimum-jerk trajectory planning pertaining to a translational 3-degree-of-freedom parallel manipulator through piecewise quintic polynomials interpolation," *Advances in Mechanical Engineering*, vol. 12, p. 1687814020913667, 2020.

- [30] H. Wang, H. Wang, J. Huang, B. Zhao, and L. Quan, "Smooth point-to-point trajectory planning for industrial robots with kinematical constraints based on high-order polynomial curve," *Mechanism and Machine Theory*, vol. 139, pp. 284-293, 2019.
- [31] F. A. Raheem, A. T. Sadiq, and N. A. F. Abbas, "Robot Arm Free Cartesian Space Analysis for Heuristic Path Planning Enhancement," *Int. J. Mech. Mechatron. Eng.*, vol. 19, pp. 29-42, 2019.
- [32] A. T. Sadiq, F. A. Raheem, and N. A. F. Abbas, "Optimal Trajectory Planning of 2-DOF Robot Arm Using the Integration of PSO Based on D* Algorithm and Cubic Polynomial Equation," in *First Int. Conf. Eng. Res., no. March*, 2017, pp. 458-467.
- [33] K. Al Khudir and A. De Luca, "Faster motion on cartesian paths exploiting robot redundancy at the acceleration level," *IEEE Robotics and Automation Letters*, vol. 3, pp. 3553-3560, 2018.
- [34] B. Brahmi, M. Saad, M. H. Rahman, and C. Ochoa-Luna, "Cartesian trajectory tracking of a 7-DOF exoskeleton robot based on human inverse kinematics," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 49, pp. 600-611, 2017.
- [35] J.-H. Chen and K.-T. Song, "Collision-free motion planning for human-robot collaborative safety under cartesian constraint," in *2018 IEEE International Conference on Robotics and Automation (ICRA)*, 2018, pp. 1-7.
- [36] A. Seddaoui and C. Saaj, "Optimised collision-free trajectory and controller design for robotic manipulators," in *Proceedings: 14th Symposium on Advanced Space Technologies in Robotics and Automation*, 2017.
- [37] A. Jena, P. K. Sahu, S. C. Bharat, and B. Biswal, "Optimal trajectory planning of a 3R SCARA manipulator using geodesic," in *2016 IEEE 1st International Conference on Power Electronics, Intelligent Control and Energy Systems (ICPEICES)*, 2016, pp. 1-6.
- [38] P. K. Sahu, G. Balamurali, G. B. Mahanta, and B. B. Biswal, "A heuristic comparison of optimization algorithms for the trajectory planning of a 4-axis SCARA robot manipulator," in *Computational Intelligence in Data Mining*, ed: Springer, 2019, pp. 569-582.
- [39] J. Zhang, Q. Meng, X. Feng, and H. Shen, "A 6-DOF robot-time optimal trajectory planning based on an improved genetic algorithm," *Robotics and Biomimetics*, vol. 5, pp. 1-7, 2018.
- [40] A. Machmudah, S. Parman, A. Zainuddin, and S. Chacko, "Polynomial joint angle arm robot motion planning in complex geometrical obstacles," *Applied Soft Computing*, vol. 13, pp. 1099-1109, 2013.