

Kinematics and Trajectory Planning Analysis Based On Hybrid Optimization Algorithms for an Industrial Robotic Manipulators

Gurjeet Singh D (✉ gurjeetsinghphd63@gmail.com)

Amritsar College of Engineering and Technology

Vijay Kumar Banga

Amritsar College of Engineering and Technology

Research Article

Keywords: Industrial robots, forward kinematics, inverse kinematics, trajectory planning and hybrid optimization algorithms

Posted Date: February 2nd, 2022

DOI: <https://doi.org/10.21203/rs.3.rs-1313895/v1>

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

[Read Full License](#)

Kinematics and Trajectory Planning analysis based On Hybrid Optimization Algorithms for an Industrial Robotic Manipulators

***¹D. Gurjeet Singh & ²Vijay Kumar Banga**

***¹Assistant Professor, Department of Electronics & Communication Engineering,
Amritsar College of Engineering and Technology. Amritsar, Punjab, India.**

**²Professor & Principal, Amritsar College of Engineering and Technology,
Amritsar, Punjab, India.**

***Email: gurjeetsinghphd63@gmail.com**

Abstract: In industrial applications and automation, the robotic manipulators exhibit a significant role. Several complex robotic systems performed a number of industrial works named spray painting, welding, assembly, pick and place action etc. The end-effector's position and the joint angles are plays a vital role since any task is activated inside the pre-defined robotic manipulator's work space. Also, the problem of trajectory planning is a very challenging task in the robotic fields. To solve these problems, this paper proposes a kinematics analysis and trajectory planning of an Industrial robotic manipulators (IRMs) based on the hybrid optimization algorithms. Here, three IRMs such as PUMA 560 (6 DOF), KUKA LBR iiwa 14 R820 (7DOF) and ABB IRB 140 (6DOF) are considered. For each robot, the forward and the inverse kinematics (IK) are analyzed and also the trajectory planning of each robot is discussed using the hybrid optimization algorithms. In this work, 18 optimization algorithms such as PSO (particle swarm optimization), SSO (social spider optimization), DFO (dragonfly optimization), BOA (butterfly optimization), CSA (crow search algorithm), BSA (bird swarm algorithm), SHO (selfish herd optimization), KHO (krill herd optimization), ALO (antlion optimization), ACO (ant colony optimization), GWO (Grey wolf optimization), GOA (grasshopper optimization), SBO (satin bowerbird optimizer), WCO (world cup optimization), COA (cuckoo optimization algorithm), CFA (cuttlefish algorithm), SOA (seagull optimization algorithm), and TSA (tunicate swarm algorithm) are utilized for both forward and inverse kinematic analysis and the trajectory planning problem. For the PUMA 560, KUKA LBR iiwa 14 R820, and ABB IRB 140 IRMs, forward kinematics (FK) are solved by the hybrid combination of PSO-SSO, SHO-KHO, and SBO-WCO individually. Also the IK are solved by the DFO-BOA, ALO-ACO and COA-CFA for each IRMs. The trajectory planning problem is solved by the CSA-BSA, GWO-GOA and SOA-TSA for each robot individually. These optimization techniques gives number of solution for kinematics and trajectory problem but it converses the best solution for the minimum multi-objective function value. The advantages of these hybrid algorithm are shorter computation time, and fewer iterations. The kinematics and trajectory analysis of each IRM is simulated using robotic tool box in MATLAB with GUI interface. For each IRM, optimized position values of the end effector, joint angles and the best optimal path are computed with minimum objective function. Finally, the performance of each robot is compared to the existing robotic works.

Keywords: Industrial robots, forward kinematics, inverse kinematics, trajectory planning and hybrid optimization algorithms

1. Introduction

Nowadays, the usage of Robot manipulators has been increased in the industrial application. Actually, IRMs are suffered by several uncertainties in their dynamics and it is also known as multivariable nonlinear systems. The uncertainties in the dynamics decay the stability and performance, named payload variation, highly time-varying, nonlinear friction, and external disturbance of the system [1]. As a result, accomplishing high performance is a most interesting role in the trajectory planning. In industries, to perform the various tasks industrial robots essentials to the high tractability in which human cannot implement. The device connected to the robot end effector is needed to survey a predefined path to operate objects or accomplish a task in the functioning region [2]. Also it includes the joint angles of the manipulator and the control of the position of every link. By considering the payload and the extreme reach capability, every particular IRM design can be effectively incorporated into a flexible industrial cells from the initial strategy steps based on the specificity of the application [3]. Both of these primary useful characteristics are essentially dependent on the every IRMs link length, an exact beneficial features of every IRMs subassemblies and the obtainable boundaries for every IRMs joint orientation. It is observed that the IRMs work space can denoted as a correlation among the arm's length, joint types and the quantity of degrees of freedom (DOF) [4].

With the help of the representative scheme which labelling the correlation among the two in the motion path, every joint and link, the robot kinematics are generally characterized. By means of the DH (Denavit-Hartenberg) conventions, forming the chain of bodies associated by joints is accomplished [5]. The DH convention depiction given the 4×4 matrix named as the homogeneous transformation matrix. Two types of kinematics such as forward and the inverse kinematics are available in the kinematics of robot. Forward kinematics is used to compute the position and orientation of the end effector for a given joint angles of the RM [6]. Inverse kinematics is used to estimate the joint angles of RM for a given required position and orientation. To define the position and orientation of the tool structure, the similar transformation matrix which obtained from the forward kinematics with corresponding to the robot base structure is employed in the inverse kinematics to solve the joint angles [7]. To enhance their usability and intelligence, different theoretic studies considering the kinematics of the RM have been performed.

In robotics, the problem of trajectory planning plays a significant role. For a particular symmetrical path, it denotes a historical motion law. Therefore for the trajectory, the particular necessities arrangement are satisfied [8]. For the control structure of the manipulator, the input is created by the planning of trajectory to implement a motion. Generally, the objective of the planning of trajectory is to discover the force inputs to change the actuators. Hence the robot trails a trajectory that permits it to drive from the source point to the end point while avoiding the obstacles [9] [10]. This is entitled as the entire motion planning issue. Here, the historical development of motion is eliminated than the path planning issue. To accomplish the robot actuators and the trajectory, the significant portion of accomplishing the effective trajectory plan is utilized which depends on both the interpolation function [11]. At last, the motion of the robot is created by the actuators. For the robot behaviour to be smooth, it is very essential. By the robot trajectory planning, different benefits are offered with the help of the energy measures. It provides smooth trajectories easier to track and decrease the strains of the actuators and of the manipulator framework [12]. In addition, in different applications, saving energy can be essential with an inadequate capacity of the energy source.

For any prearranged issue, optimization is the procedure of determining the good outcome. Using the three significant factors such as variables, objective functions and design variables, the

clarification of a distinctive optimization issue is solved [13]. To the function of the RMs, the optimum motion planning is very significant. Its major objective is the trajectory creation from the initial point to target point. Also this fulfills some objectives for example, minimum time interval or travelling distance, fulfilling the robots dynamic and kinematics, and obstacle avoidance or minimum energy utilization [14]. The robot motions are still enhanced physically in several industrial applications, this being luxurious and influencing to errors. Hence, for spontaneous estimation of optimum trajectories of the robot motions, the investigators are operating to search algorithms in the current decades. Due to the different factors for example, difficult objective function, collision avoidance, an arrangement of suitable positions for the base of robot, redundant kinematics and the uncertain likelihoods for accomplishment the roles, this methodology is verified to be fairly complex [15]. Every optimization algorithm has its own benefit on distinctive task. For the robot motion in various occurrences, the several optimization target have to be chosen based on the tasks of robot motion.

To solve both forward and inverse kinematics issues and the planning of trajectory issues for the several configurations of manipulator, optimization approaches are useful. To solve the nonlinear kinematic issues, the conservative methodologies can be accomplished. But the main disadvantage of these approaches are singularity which contrary to local solutions. Furthermore, these approach becomes unbalanced and does not contrary to optimal solutions if the primary solving is not truthful. Hence to solve these problems, the recently established meta-heuristic algorithms can be employed. In this work, the different hybrid optimization algorithm is used for solving the kinematic analysis and the trajectory planning of the IRs such as PUMA 560 (6DOF), KUKA LBR iiwa 14 R820 (7DOF) and ABB IRB 140 (6DOF).

The major contributions can be summarized as follows:

- Kinematic analysis and trajectory planning problem are presented with the help of the hybrid optimization algorithms for an industrial robotics manipulators such as PUMA 560, KUKA LBR iiwa 14 R820 and the ABB IRB 140.
- In the PUMA 560 robot, the forward and inverse kinematics are analysed by the hybrid optimization algorithms of PSO-SSO and DFO-BOA. The hybrid combination of SHO-KHO and ALO-ACO are utilized for solving both forward and inverse kinematics of the KUKA LBR iiwa 14 R820 robot. The hybrid algorithms of SBO-WCO and COA-CFA methods are employed for solving the forward and inverse kinematics problems of ABB IRB 140.
- The hybrid optimization scheme of CSA-BSA, GWO-GOA, and SOA-TSA are utilized to solve the trajectory planning problem of PUMA 560, KUKA LBR iiwa 14 R820 and the ABB IRB 140. The selection of suitable hybrid optimization method will provides the global optimization solution.
- For each IRM, the optimal position values of the end effector, and the joint angles are estimated with the minimum position error values. Also the optimal path is evaluated with and without obstacles for each IRM with the minimum fitness function value. The multi-objective functions such as travelling time, collision free motion, jerks and accelerations are considered for the trajectory planning evaluation for each IRM.

2. Related works

Recent related works are given below:

An improved Multi-objective Ant Lion Optimizer based Kinematic and Dynamic Optimal Trajectory Planning for Industrial Robot was introduced by Amruta et al. [16]. For such optimal

trajectory planning, the kinematic and dynamic constraints (i.e. jerk (kinematic) and torque (dynamic)) were considered as the essential parameters. But constraining such parameters may increase the total travelling rate of each robots, at last it affects the productivity. Therefore, an enhanced optimization algorithm was used in this method for reducing the jerk-torque rate for 6 axis Kawasaki RS06L industrial robot.

Zhifeng et al. [17] introduced a Screw Theory for planning the target of industrial robots with minimal synthesis error. The proposed trajectory planning obtains a constant movement and reduce the industrial robots end-effector synthesis error. This error was modelled via considering the interpolation effects and flexibilities of entire joints. Whereas, the robot ER3A-C60 was modelled based on screw theory and Kane calculations. The PSO algorithm with dynamic and kinematic constraints was used finally to evaluate the minimal synthesis error.

A smooth and time based optimal trajectory planning using a single polynomial for industrial robots was presented by Nguyen and Huang in [18]. It performs two different function for trajectory planning they are smoothing and optimizing. The trajectories having the constraints acceleration, jerk and velocity of joint space was interpolated using single polynomial and the travelling time reduction was achieved using sequential quadratic programming algorithm.

A kinematic analysis was presented by Ashwani et al. [19] by the way of meta-heuristic algorithms. With the help of the nature-inspired swarm procedures, kinematic analysis for a 4 DOF robotic arm has been described. For the period of joint movement, the presence of error provide growth to imprecision and ineffectiveness. By the way of GWO and ABC (artificial bee colony) algorithms, the absolute error and the position error have been estimated. Experimental outcomes are evaluated for the fitness function and time occupied for the specific iterations. In the suitable stage, both the errors have been reduced. GWO procedure consume minimum time for the evaluation as related to the ABC approach. In 4 DOF RM, the fitness value is enhanced through ABC than GWO. For the evaluation of procedure, the time is enhanced as the quantity of iteration improves.

The inverse kinematic solution by the GA (genetic algorithm) was presented by Srinath et al. [20] for the serial manipulators. This work offers the perception into inverse kinematic design of IRMs of numerous DOF, DH constraints, forward kinematics and the homogeneous transformation matrices. By the way of simple geometry, the inverse kinematics can be accomplished with trigonometry functions for a robot with 2/3/4 DOF. At first, forward kinematic issue can be solved and the concurrent equations are generated for solving the joint angles. The inverse kinematics turn into compound to solve physically for a robot with six joints or more joints. In this situations, the inverse kinematics will be handled as an optimization issue which can be resolved mathematically.

3. System model of PUMA 560 (6DOF)

In the numerous applications such as surgery, welding and laser tracking structure, the open chain mechanism of a PUMA (Programmable Universal Machine for Assembly) industrial robot [21] arm is employed. The six DOF and six revolute joints are involved in the PUMA 560 robot. Here, the first three joints are find the end effector and the remaining three joints are offer the orientation of the end effector. Figure 1 illustrates the PUMA 560 6 DOF IRM.

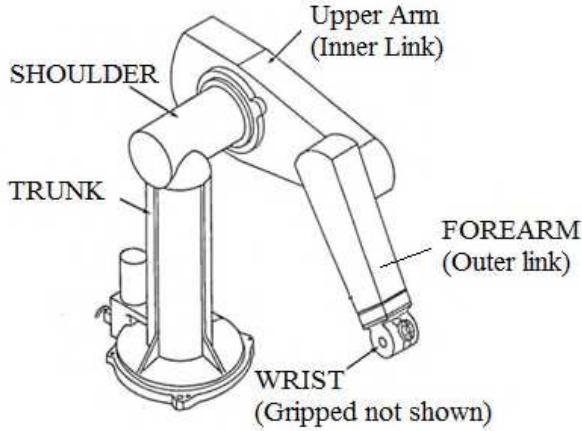


Figure 1: PUMA 560 6DOF

Table 1: DH convention for PUMA 560 (6 DOF)

| Link length (d) | Link offset (a) | Link twist (alpha) | Initial joint value | Final joint value |
|-----------------|-----------------|--------------------|---------------------|-------------------|
| 0 | 0 | -90 | -185 | 185 |
| d2=0.7 | a2=0.5 | 0 | -155 | 35 |
| d3=0.948 | a3=0.0948 | 90 | -130 | 154 |
| 0 | a4=0.68 | -90 | -350 | 350 |
| 0 | 0 | 90 | -130 | 130 |
| 0 | a6=0.853 | 0 | -350 | 350 |

Table 1 provide the DH conventions of PUMA 560 robot. To denote the homogeneous transformation, the product of four fundamental transformation is included in the DH conventions and it is given as:

$$A_i = \begin{bmatrix} C\theta_{n+1} & -S\theta_{n+1}C\alpha_{n+1} & S\theta_{n+1}S\alpha_{n+1} & a_{n+1}C\theta_{n+1} \\ S\theta_{n+1} & C\theta_{n+1}C\alpha_{n+1} & -C\theta_{n+1}S\alpha_{n+1} & a_{n+1}S\theta_{n+1} \\ 0 & S\alpha_{n+1} & C\alpha_{n+1} & d_{n+1} \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (1)$$

Here S= sin, C= cos

3.1 Forward kinematic analysis

For every joint, the transformation matrix can be computed using Eqn (1). This expression contains three fix element such as one variable (theta), one 1*3 type matrix and one 3*3 rotational matrix. The position of the end effector is denoted by the 1*3 type matrix and the orientation of the end effector is represented by the 3*3 rotational matrix.

$$A_1 = \begin{bmatrix} C\theta_1 & 0 & S\theta_1 & a_1C\theta_1 \\ S\theta_1 & 0 & -C\theta_1 & a_1S\theta_1 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2)$$

$$A_2 = \begin{bmatrix} C\theta_2 & -S\theta_2 & 0 & a_2 C\theta_2 \\ S\theta_2 & C\theta_2 & 0 & a_2 S\theta_2 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (3)$$

$$A_3 = \begin{bmatrix} C\theta_3 & 0 & S\theta_3 & 0 \\ S\theta_3 & 0 & -C\theta_3 & 0 \\ 0 & 1 & 0 & d_3 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (4)$$

$$A_4 = \begin{bmatrix} C\theta_4 & 0 & -S\theta_4 & 0 \\ S\theta_4 & 0 & C\theta_4 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (5)$$

$$A_5 = \begin{bmatrix} C\theta_5 & 0 & S\theta_5 & 0 \\ S\theta_5 & 0 & -C\theta_5 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (6)$$

$$A_6 = \begin{bmatrix} C\theta_6 & -S\theta_6 & 0 & 0 \\ S\theta_6 & C\theta_6 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (7)$$

The entire transformation can be given as:

$$T_6 = A_1 A_2 A_3 A_4 A_5 A_6 \quad (8)$$

The final expression can be accomplished in the forward kinematics by multiplying the above-mentioned expressions. Then, to obtain the kinematic equation, the final expression is compared with Eqn (1).

$$T_6 = \begin{bmatrix} n_x & o_x & a_x & p_x \\ n_y & o_y & a_y & p_y \\ n_z & o_z & a_z & p_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (9)$$

$$\begin{aligned} n_x &= C_1 [C_{23}(C_4 C_5 C_6 - S_4 S_6) - S_{23} S_5 C_6] + S_1 (S_4 C_5 C_6 + C_4 S_6) \\ n_y &= S_1 [C_{23}(C_4 C_5 C_6 - S_4 S_6) - S_{23} S_5 C_6] - C_1 (S_4 C_5 C_6 + C_4 S_6) \\ n_z &= S_{23}(C_4 C_5 C_6 - S_4 S_6) + C_{23} S_5 C_6 \\ o_x &= C_1 [-C_{23}(C_4 C_5 C_6 + S_4 C_6) + S_{23} S_5 S_6] + S_1 (-S_4 S_5 S_6 + C_4 C_6) \\ o_y &= S_1 [-C_{23}(C_4 C_5 C_6 + S_4 C_6) + S_{23} S_5 S_6] - C_1 (-S_4 C_5 S_6 + C_4 C_6) \\ o_z &= -S_{23}(C_4 C_5 C_6 + S_4 C_6) - C_{23} S_5 S_6 \end{aligned} \quad (10)$$

$$\begin{aligned}
a_x &= C_1 [C_{23}C_4S_5 + S_{23}C_5] - C_1S_4S_5 \\
a_y &= S_1 [C_{23}C_4S_5 + S_{23}C_5] - C_1S_4S_5 \\
a_z &= S_{23}C_4S_5 - C_{23}C_5 \\
p_x &= C_1(C_2a_2 + a_1) + S_1d_3 \\
p_y &= S_1(C_2a_2 + a_1) - C_1d_3 \\
p_z &= S_2a_2
\end{aligned} \tag{11}$$

3.2 Inverse kinematic analysis

The opposite of the forward kinematic is called as the inverse kinematic model. By multiplying every inverse matrix of T matrices, the inverse kinematic solution is accomplished. Then balancing the equivalent components of the identical matrices of both ends. By the way of the position of end effector, the every joint angle is computed.

$$\begin{aligned}
A_1^{-1} \times \begin{bmatrix} n_x & o_x & a_x & p_x \\ n_y & o_y & a_y & p_y \\ n_z & o_z & a_z & p_z \\ 0 & 0 & 0 & 1 \end{bmatrix} &= A_2A_3A_4A_5A_6 \\
\begin{bmatrix} n_xC_1 + n_yS_1 & o_xC_1 + o_yS_1 & a_xC_1 + a_yS_1 & p_xC_1 + p_yS_1 - a_1(C_1C_1 + S_1S_1) \\ n_z & o_z & a_z & p_z \\ n_xS_1 - n_yC_1 & o_xS_1 - o_yC_1 & a_xS_1 - a_yC_1 & p_xS_1 - p_yC_1 \\ 0 & 0 & 0 & 1 \end{bmatrix} & \\
= \begin{bmatrix} C_{23}(C_4C_5C_6 - S_4C_6) - S_{23}S_5C_6 & C_{23}(-C_4C_5C_6 - S_4C_6) - S_{23}S_5S_6 & C_{23}C_4S_5 + S_{23}C_5 & a_2C_2 \\ S_{23}(C_4C_5C_6 - S_4S_6) - C_{23}S_5C_6 & S_{23}(-C_4C_5C_6 - S_4S_6) - C_{23}S_5S_6 & S_{23}C_4S_5 - C_{23}C_5 & a_2S_2 \\ S_4C_5C_6 + C_4S_6 & -S_4C_5S_6 + C_4C_6 & S_5S_4 & d_5 \\ 0 & 0 & 0 & 1 \end{bmatrix} & \tag{12}
\end{aligned}$$

Then Eqn (12) and Eqn (1) are compared, we get

Where, $p_z = a_2S_2$ and $S_2 = p_z/a_2$

$$\theta_2 = \sin^{-1}(p_z/a_2) \tag{13}$$

Where, $p_xS_1 - p_yC_1 = d_3$, $S_1 = \frac{d_3 + p_yC_1}{p_x}$ and $p_xC_1 + p_yS_1 - a_1 = a_2C_2$

The value of C_1 is estimated by S_1 and a_2C_2 .

Where $C_1 = \frac{p_x(a_2C_2 + a_1) - p_yd_3}{p_x^2 + p_y^2}$

$$\theta_1 = \cos^{-1}\left(\frac{p_x(a_2C_2 + a_1) - p_yd_3}{p_x^2 + p_y^2}\right) \tag{14}$$

Likewise,

$$\begin{aligned}
\theta_3 &= \theta_{23} - \theta_2 \\
\theta_4 &= \tan^{-1} \left[\frac{a_x S_1 - a_y C_1}{a_x C_1 C_{23} + a_y S_1 S_{23} + a_z S_{23}} \right] \\
\theta_5 &= -\tan^{-1} \left[\frac{a_z C_1 C_{23} C_4 + a_y S_1 S_{23} C_4 + a_z S_{23} C_4 + a_z S_1 S_4 - a_y C_1 S_4 + S_5}{-a_z S_{23} - a_y S_1 S_{23} + a_z C_{23}} \right] \\
\theta_6 &= \tan^{-1} \left[\frac{o_z C_{23} - o_x C_1 S_{23} - o_y S_1 S_{23}}{n_x C_1 S_{23} + n_y S_1 S_{23} - n_z C_{23}} \right]
\end{aligned} \tag{15}$$

3.3 Forward kinematic problem using PSO-SSA

In this section, the problem of forward kinematic analysis is solved by the hybrid optimization algorithms of PSO-SSA. For a specified joint angle values, the position of end effector is determined in the forward kinematics. Using the hybrid optimization approaches, the optimized position values of end effector can be accomplished for the given joint angles. This hybrid algorithm combines the exploration stage of the SSA to the exploitation (updation) stage of the PSO [24] and it contains three phases such as initialization, exploration and the exploitation process. The SSA [25] procedure is motivated by the social spiders and it only depends on the foraging scheme of social spiders. To find the prey location, this algorithm employs the vibrations on the spider web. The PSO algorithm is presented to improve the convergence of SSA in the exploitation phase. The hybrid algorithm can be defined as follows.

In the beginning, the quantity of PSO-SSA parameters are initialized. Then the SSA procedure to create the spider population after initialization process. The spider population is treated as the set of joint angles $[\theta_1 \theta_2 \theta_3 \theta_4 \theta_5 \theta_6]$ achieved from table 1. For the period of SSA implementation, the set joint values are unchanged. Therefore, to store their data, the fixed size memory is assigned. In the hunt space, the position of the spiders are arbitrarily created. In the current location, the primary goal vibration of each spider is fixed in the population and the intensity of vibration is zero. Then the number of iterations are executed using this hybrid approach. Based on the fitness value, the numerous position values of the end effector can be estimated for a specified joint angles in each iteration.

The end effector position is treated as the optimization issue. The goal of this optimization issue is to determine the position value of end effector for a given joint angles. Here, the exact estimation of the position values are very significant. This objective issue is solved by the PSO-SSA. Here the fitness function for the forward kinematic is to determine the minimum error which depends on the Euclidean distance among the current location (l, m, n) and desired location (a, b, c) .

$$Fitness = \sqrt{(l-a)^2 + (m-b)^2 + (n-c)^2} \tag{16}$$

After computing the fitness values, the updation process is performed using PSO to get the optimal position values of end effector. According to the behaviour of swarm, the PSO algorithm is performed. This approach motivates the social behaviour of entities for example, bird flocking and fish schooling. In the hunt space, the movement of individual's can accomplished the PSO procedure. Using an operator, the velocity and position of individuals of population are updated.

Therefore, the individuals can be projected to move in the direction of best solution. The process of updation can be given as:

$$\begin{aligned}x_i^d(t+1) &= x_i^d(t) + V_i^d(t+1) \\V_i^d(t+1) &= w(t).V_i^d(t) + c_1.rand().(p_i^d(t) - x_i^d(t)) + c_2.rand().(p_g^d(t) - x_i^d(t))\end{aligned}\quad (17)$$

Here the velocity and the position of i^{th} agent are denoted as $V_i^d(t)$ and $x_i^d(t)$ respectively at time t , the finest earlier location of i^{th} agent is defined as $p_i^d(t)$ and the finest prior location of population is signified as $p_g^d(t)$. The positive constants are denoted as c_1 and c_2 . The random function is defined as $rand()$, dimension is denoted as d and the inertia weight is denoted as $w(t)$. This procedure is repeated until the maximum iteration is reached. Finally, the best optimum position values $[p_x \ p_y \ p_z]$ are obtained using this PSO-SSA.

3.4 Inverse kinematic analysis using DFO-BOA

In this section, the inverse kinematic problem is solved with the help of the hybrid optimization algorithms of DFO-BOA. The inverse kinematic is to discover the optimal joint angles for a specified optimized position values. Here the optimal joint angles are estimated by the combination of DFO and BOA. To enhance the exploitation stage of DFO, the exploration stage of BOA is introduced as a hybrid scheme. The BOA [27] is motivates the food search and the mating behaviour of butterflies. This algorithm only depends on the foraging scheme of butterflies. To find the position of nectar or partner of mating, they use their sense of smell. From the dynamic and static swarming behaviours of dragonflies, the DFO [26] procedure is initiated. By the way of designing the social communication of dragonflies, the two significant stages such as exploration and exploitation are considered in directing, searching for foods and also escaping opponents if swarming energetically. The hybrid algorithm can be defined as follows.

This procedure involve three phases such as initialization, exploration and exploitation. At first, the constraints of DFO-BOA are initialized. After the initialization, the population of butterflies are created by the BOA procedure. The amount of population is treated as the set of optimized position values achieved from the forward kinematics. For the period of BOA implementation, the set of optimized position values are unchanged. Therefore, to store their data, the fixed size memory is assigned. In the hunt space, the position of the butterflies are arbitrarily created. Then the number of iterations are executed using this hybrid approach. Based on the fitness value, the six joint angles can be estimated for a specified optimized position values of end effector in each iteration.

The primary goal of this inverse kinematic is to discover the optimized joint angles for a specified position of the RM. Inverse kinematic is treated as the optimization issue which is resolved by DFO-BOA. The fitness function for the inverse kinematic is to determine the position error which depends on the Euclidean distance among the desired and the predicted points.

$$Fitness = \sqrt{(x - r)^2 + (y - s)^2 + (z - t)^2} \quad (18)$$

After computing the fitness values, the updation process is performed using DFO to get the optimal joint angles. The procedure of updation can be defined as follows:

$$X_{t+1} = X_t + levy(d) \times X_t \quad (19)$$

Here the present iteration is defined as t and the dimension of position vectors is denoted as d .

The levy flight is estimated by the following equation:

$$Levy(x) = 0.01 \times \frac{r_1 \times \sigma}{|r_2|^{\frac{1}{\beta}}} \quad (20)$$

Here, the random numbers are denoted as r_1 and r_2 in the range of [0, 1]. The value of β is equivalent to 1.5. σ is computed by the following expression:

$$\sigma = \left(\frac{\Gamma(1+\beta) \times \sin\left(\frac{\pi\beta}{2}\right)}{\Gamma\left(\frac{1+\beta}{2}\right) \times \beta \times 2^{\frac{(\beta-1)}{2}}} \right)^{\frac{1}{\beta}} \quad (21)$$

Here, $\Gamma(x) = (x - 1)!$. This procedure is repeated until the maximum iteration is reached. Finally, the best optimum joint values $[\theta_1 \theta_2 \theta_3 \theta_4 \theta_5 \theta_6]$ are obtained using this DFO-BOA.

4. System model of KUKA LBR iiwa 14 R820 (7DOF)

The progressive KUKA LBR IIWA 14 R820 cooperative robot contains a characteristics of an 820 mm reach, 14 kg payload and Protection Class IP54 rating. In the robotic development, this robot is latest one and it have the capability to function with humans. In sensors and soft edges, this robot [22] has constructed that create it safe for human cooperation and the capability to sense movement and touch all over. For matching, fastening, palletizing, coating, measuring and other physical management applications, this robot is unlimited. This robot is the redundant manipulator. Figure 2 illustrates the KUKA LBR iiwa 14 R820 IRM.



Figure 2: KUKA LBR iiwa 14 R820 (7 DOF)

Table 2: DH convention for KUKA LBR iiwa 14 R820 (7 DOF)

| Link length | Link offset | Link twist | Initial joint value | Final joint value |
|-------------|-------------|------------|---------------------|-------------------|
| 0 | d1=0.34 | -90 | -170 | 170 |

| | | | | |
|---|----------|-----|------|-----|
| 0 | 0 | -90 | -120 | 120 |
| 0 | d3=0.4 | -90 | -170 | 170 |
| 0 | 0 | -90 | -120 | 120 |
| 0 | d5=0.4 | -90 | -170 | 170 |
| 0 | 0 | -90 | -120 | 120 |
| 0 | d7=0.126 | -90 | -175 | 175 |

Table 2 provide the DH convention of KUKA robot. To denote the homogeneous transformation, the product of four fundamental transformation is included in the DH conventions and it is given as:

$$A_i = \begin{bmatrix} C\theta_i & -S\theta_i C\alpha_i & S\theta_i S\alpha_i & a_i C\theta_i \\ S\theta_i & C\theta_i C\alpha_i & -C\theta_i S\alpha_i & a_i S\theta_i \\ 0 & S\alpha_i & C\alpha_i & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (22)$$

4.1 Forward kinematic analysis

$$T_1^0 = \begin{bmatrix} C_1 & 0 & -S_1 & 0 \\ S_1 & 0 & C_1 & 0 \\ 0 & -1 & 0 & d_1 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (23)$$

$$T_2^1 = \begin{bmatrix} C_2 & 0 & S_2 & 0 \\ S_2 & 0 & -C_2 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (24)$$

$$T_3^2 = \begin{bmatrix} C_3 & 0 & -S_3 & 0 \\ S_3 & 0 & C_3 & 0 \\ 0 & 1 & 0 & d_3 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (25)$$

$$T_4^3 = \begin{bmatrix} C_4 & 0 & -S_4 & 0 \\ S_4 & 0 & -C_4 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (26)$$

$$T_5^4 = \begin{bmatrix} C_5 & 0 & -S_5 & 0 \\ S_5 & 0 & C_5 & 0 \\ 0 & 1 & 0 & d_5 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (27)$$

$$T_6^5 = \begin{bmatrix} C_6 & 0 & S_6 & 0 \\ S_6 & 0 & -C_6 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (28)$$

$$T_7^6 = \begin{bmatrix} C_7 & -S_7 & 0 & 0 \\ S_7 & C_7 & 0 & 0 \\ 0 & 1 & 0 & d_7 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (29)$$

$$T_7^0 = T_1^0 T_2^1 T_3^2 T_4^3 T_5^4 T_6^5 T_7^6 = \begin{bmatrix} R_{11} & R_{12} & R_{13} & P_x \\ R_{21} & R_{22} & R_{23} & P_y \\ R_{31} & R_{32} & R_{33} & P_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (30)$$

$$\begin{aligned} R_{11} = & C_7 S_6 (S_4 (S_1 S_3 - C_1 C_2 C_3) - C_1 C_4 S_2) C_6 (C_5 (C_4 (S_1 S_3 - C_1 C_2 C_3) \\ & + C_1 S_2 S_4) + S_5 (C_3 S_1 + C_1 C_2 S_3)) + S_7 (S_5 (C_4 (S_1 S_3 - C_1 C_2 C_3) \\ & + C_1 S_2 S_4) - C_5 (C_3 S_1 + C_1 C_2 S_3)) \end{aligned}$$

$$\begin{aligned} R_{12} = & C_7 (S_5 (C_4 (S_1 S_3 - C_1 C_2 C_3) + C_1 S_2 S_4) - C_5 (C_3 S_1 + C_1 C_2 S_3)) - \\ & S_7 (S_6 (S_4 (S_1 S_3 - C_1 C_2 C_3) - C_1 C_4 S_2) - C_6 (C_5 (C_4 (S_1 S_3 - C_1 C_2 C_3) \\ & + C_1 S_2 S_4) + S_5 (C_3 S_1 + C_1 C_2 S_3))) \end{aligned}$$

$$\begin{aligned} R_{13} = & -C_6 (S_4 (S_1 S_3 - C_1 C_2 C_3) - C_1 C_4 S_2) - S_6 (C_5 (C_4 (S_1 S_3 - C_1 C_2 C_3) \\ & + C_1 S_2 S_4) + S_5 (C_3 S_1 + C_1 C_2 S_3)) \end{aligned}$$

$$\begin{aligned} R_{21} = & -[C_7 (S_6 (S_4 (C_1 S_3 + C_2 C_3 S_1) + C_4 S_1 S_2) - C_6 (C_5 (C_4 (C_1 S_3 + \\ & C_2 C_3 S_1) - S_1 S_2 S_4) + S_5 (C_1 C_3 - C_2 S_1 S_3))) - S_7 (S_5 (C_4 (C_1 S_3 \\ & + C_2 C_3 S_1) - S_1 S_2 S_4) - C_5 (C_1 C_3 - C_2 S_1 S_3))] \end{aligned}$$

$$\begin{aligned} R_{22} = & -[S_7 (S_6 (S_4 (C_1 S_3 + C_2 C_3 S_1) + C_4 S_1 S_2) - C_6 (C_5 (C_4 (C_1 S_3 + \\ & C_2 C_3 S_1) - S_1 S_2 S_4) + S_5 (C_1 C_3 - C_2 S_1 S_3))) - S_7 (S_5 (C_4 (C_1 S_3 \\ & + C_2 C_3 S_1) - S_1 S_2 S_4) - C_5 (C_1 C_3 - C_2 S_1 S_3))] \end{aligned}$$

$$\begin{aligned} R_{23} = & C_6 (S_4 (C_1 S_3 + C_2 C_3 S_1) + C_4 S_1 S_2) + S_6 (C_5 (C_4 (C_1 S_3 + C_2 C_3 S_1) - \\ & S_1 S_2 S_4) + S_5 (C_1 C_3 - C_2 S_1 S_3)) \end{aligned}$$

$$R_{33} = C_6 (C_2 C_4 - C_3 S_2 S_4) - S_6 (C_5 (C_2 S_4 + C_3 C_4 S_2) - S_2 S_3 S_5) \quad (31)$$

$$\begin{aligned} p_x = & d_3 C_1 S_2 - d_7 (S_4 (S_1 S_3 - C_1 C_2 C_3) - C_1 C_4 S_2) - [d_7 (C_6 (S_4 (S_1 S_3 - C_1 C_2 C_3) \\ & - C_1 C_4 S_2) + S_6 (C_5 (C_4 (S_1 S_3 - C_1 C_2 C_3) + C_1 S_2 S_4) + S_5 (C_3 S_1 + C_1 C_2 S_3)))] \end{aligned}$$

$$p_Y = [d_7(C_6(S_4(C_1S_3 + C_2C_3S_1 + C_4S_1S_2) + S_6(C_5(C_4(C_1S_3 + C_2C_3S_1) - S_1S_2S_4) + S_5(C_1C_3 - C_2S_1S_3))))]d_5(S_4(C_1S_3 + C_2C_3S_1) + C_4S_1S_2) + d_3S_1S_2 \quad (32)$$

$$p_Z = d_1 + d_5(C_2C_4 - C_3S_2S_4) - [(d_7(S_6(C_5(C_2S_4 + C_3C_4S_2) - S_2S_3S_5) - C_6(C_2C_4 - C_3S_2S_4)) + d_3C_2)]$$

4.2 Inverse kinematic analysis

The manipulator model in this study has 7 joint angles while the given position and the orientations of the end-effector can be specified by six parameters meaning that one extra degree of freedom in the manipulator is available. The extra degree of freedom is representing the redundancy. In this case study the joint 3 is fixed so the model has 6 joint angles. So we can solve inverse kinematics problem into two simpler problems and arm link will give inverse position kinematics, and wrist link will give inverse orientation kinematics.

Arm link structure

$$\text{Where, } r_c = \sqrt{x_c^2 + y_c^2}; \quad s = z_c - d_0;$$

$$\theta_1 = a \tan 2\left(\frac{x_c}{y_c}\right) \quad (33)$$

$$\cos(\theta_3) = \frac{r^2 + s^2 - d_1^2 - d_3^2}{2d_1d_3} = D \quad (34)$$

$$\text{Since } r_c = \sqrt{x_c^2 + y_c^2}; \quad s = z_c - d_0$$

$$\begin{aligned} \text{Hence, } \theta_3 &= a \tan 2(D, \pm \sqrt{1 - D^2}) \\ \theta_2 &= a \tan 2(s, r) - a \tan 2(d_3s_3, d_1 + d_3c_3) \end{aligned} \quad (35)$$

Wrist link structure:

$$\text{Where, } R = R_3^0 R_6^3 \text{ and } R_6^3 = R_3^{0T} R = \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix}$$

$$R_6^3 = \text{Rot}(x, \theta_4) \text{Rot}(z, \theta_5) \text{Rot}(x, \theta_6) \quad (36)$$

So, the Euler angle (XZX) solution is applied to this equation.

$$R_6^3 = \begin{bmatrix} c_4c_5c_6 - s_4s_6 & -c_4c_5s_6 - s_4c_6 & c_4s_5 \\ s_4c_5c_6 + c_4s_6 & -s_4c_5s_6 + c_4c_6 & s_4s_5 \\ -s_5c_6 & s_5s_6 & c_5 \end{bmatrix} \quad (37)$$

Hence we can use the last column to solve

$$\begin{aligned} \theta_4 &= a \tan 2(r_{23}, r_{13}) \\ \theta_5 &= a \tan 2\left(\sqrt{r_{13}^2 + r_{23}^2}, r_{33}\right) \\ \theta_6 &= a \tan 2(r_{32}, r_{31}) \end{aligned} \quad (38)$$

4.3 Forward kinematic analysis using SHO-KHO

In this section, the hybrid optimization algorithm of SHO-KHO is utilized to solve the problem of forward kinematics. This hybrid algorithm combines the exploration stage of the SHO to the exploitation (updation) stage of the KHO and it contains three phases such as initialization, exploration and the exploitation process. The SHO [28] is motivates the behaviour of selfish herd and it is established by entities existing in combinations while visible to some kind of predation risk. The KHO [29] is simulates the herding behaviour of krill individuals. The hybrid algorithm can be defined as follows.

At first, the constraints of SHO-KHO are initialized. The amount of populations are created by the SHO after the initialization. The amount of population is treated as the set of joint values $[\theta_1 \theta_2 \theta_3 \theta_4 \theta_5 \theta_6 \theta_7]$ obtained from table 2. With N demonstrating the entire population dimension, the procedure initiated the set A of N individual locations a_i ($A = \{a_1, a_2, \dots, a_N\}$). Such position is denoted as an n-dimensional vector $a_i = [a_{i,1}, a_{i,2}, \dots, a_{i,n}]$. Moreover, for the specified optimization issue, it defines a potential solution. By an arbitrary uniform distribution among a pair of prearranged constraint limits, these locations are modified and it can be given as:

$$a_{i,j}^0 = x_j^{low} + \text{rand}(0,1). (x_j^{high} - x_j^{low}) \quad (39)$$

$$i = 1, 2, \dots, N; \quad j = 1, 2, \dots, n$$

Here, the decision space's upper and lower bounds are denoted as x_j^{high} and x_j^{low} . The individual and constraint indexes are defined as i and j which equivalent to every channel. $\text{rand}(0,1)$ is the arbitrarily created number. Then the number of iterations are executed using this hybrid approach. Based on the fitness value, the numerous position values of the end effector can be estimated for a specified joint angles in each iteration.

The position of end effector is considered as the optimization problem which is solved by SHO-KHO. The fitness for the forward kinematic is to discover the minimum position error which depends on the Euclidean distance among the predicted points and the desired points.

$$\text{Fitness} = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2 + (z_2 - z_1)^2} \quad (40)$$

After computing the fitness values, the updation process is performed using KHO to get the optimal position values of end effector. The process of updation can be given as:

$$X_i(t + \Delta t) = X_i(t) + \Delta t \frac{dX_i}{dt} \quad (41)$$

Here, the utmost significant factor is denoted as Δt . This constraint operate as a scale factor for the speed vector and it only rely upon the hunt space.

$$\Delta t = C_t \sum_{j=1}^{NV} (UB_j - LB_j) \quad (42)$$

Here, the entire amount variable is denoted as NV. The upper and lower limits of j^{th} variables ($j = 1, 2, \dots, NV$) is defined as UB_j and LB_j . C_t is the constant number in the range of [0, 2]. This procedure is repeated until the maximum iteration is reached. Finally, the best optimum position values are obtained using this SHO-KHO.

4.4 Inverse kinematic analysis using ALO-ACO

In this section, the hybrid optimization algorithm of ALO-ACO is utilized to solve the problem of inverse kinematics. This hybrid algorithm combines the exploration stage of the ALO [30] to the exploitation (updation) stage of the ACO [31] and it contains three phases such as initialization, exploration and the exploitation process. ACO is motivates the idea of stigmergy in environment. Stigmergy denotes the management of surrounding through genetic organisms to interconnect with each other. ALO motivates the chasing procedure of antlions. The hybrid algorithm can be defined as follows.

This procedure involve three stages such as initialization, exploration and exploitation. At first, the constraints of ALO-ACO are initialized. Then the initial population is created by the ALO after the initialization. The quantity of population is treated as the set of optimized position values achieved from the forward kinematics. In the surroundings, the ants travel stochastically while searching for food. Therefore, for the demonstrating ants' movement, the arbitrary walk is selected which is given as:

$$X(t) = [0, \text{cumsum}(2r(t_1)-1), \text{cumsum}(2r(t_2)-1), \dots, \text{cumsum}(2r(t_n)-1)] \quad (43)$$

Here, the cumulative sum is denoted as *cumsum*, the extreme number of iteration is denoted as *n*, and the arbitrary walk step is denoted as *t*. The stochastic function *r(t)* can be given as:

$$r(t) = \begin{cases} 1 & \text{if } \text{rand} > 0.5 \\ 0 & \text{if } \text{rand} \leq 0.5 \end{cases} \quad (44)$$

Here, the arbitrary number is defined as *rand* in the range of [0, 1]. Then the number of iterations are executed using this hybrid approach. Based on the fitness value, the joint angles can be estimated for a specified optimized position values of end effector in each iteration.

The inverse kinematic issue is treated as the optimization problem which is solved by ALO-ACO. The fitness for the inverse kinematic is to discover the minimum position error which depends on the Euclidean distance among the desired and the predicted points and it can be given as:

$$\text{Fitness} = \sqrt{(x_d - x_c)^2 + (y_d - y_c)^2 + (z_d - z_c)^2} \quad (45)$$

After computing the fitness values, the updation process is performed using ACO to get the optimal joint angles. The procedure of updation can be defined as follows:

$$\tau_{i,j} = (1-\rho)\tau_{i,j} + \Delta\tau_{i,j} \quad (46)$$

Here the pheromone evaporation rate is denoted as ρ . The entire quantity of pheromone is denoted as $\Delta\tau_{i,j}$ which placed on the *i*, *j* edge can be given as:

$$\Delta\tau_{i,j} = \begin{cases} \frac{1}{L_k} & \text{kth ant travels on edge } i, j \\ 0 & \text{otherwise} \end{cases} \quad (47)$$

Here, the length of path is denoted as L_k . This procedure is repeated until the maximum iteration is reached. Finally, the best optimum joint values are obtained using this ALO-ACO.

5. System model of ABB IRB 140 (6DOF)

In the numerous surroundings and applications, ABB IRB 140 robot [23] is extremely flexible. In the product requirement file, this robot contains different possible characteristics for example, extreme temperature surroundings, clean room categorization for the high degree of decomposition

safety, foundry safety and capability to wash, and several other characteristics for applications like arc welding and gluing. Figure 3 illustrates the ABB IRB 140 (6 DOF) IRM.



Figure 3: ABB IRB 140 (6 DOF)

Table 3: DH convention for ABB IRB 140 (6 DOF)

| Link length (a) | Link offset (d) | Link twist (alpha) | Initial joint value | Final joint value |
|--------------------|--------------------|-----------------------|---------------------|-------------------|
| 0 | d1=352 | 0 | -180 | 180 |
| a1=70 | 0 | -90 | -90 | 110 |
| a2=360 | 0 | 0 | -230 | 50 |
| 0 | d4=380 | -90 | -200 | 200 |
| 0 | 0 | 90 | -115 | 115 |
| 0 | 0 | -90 | -400 | 400 |

Table 2 provide the DH convention of ABB IRB 140 robot. The homogeneous transform can be given as:

$${}^{i-1}T_i = \begin{bmatrix} C\theta_i & -S\theta_i & 0 & a_{i-1} \\ S\theta_i C\alpha_{i-1} & C\theta_i C\alpha_{i-1} & -S\alpha_{i-1} & -S\alpha_{i-1} d_i \\ S\theta_i S\alpha_{i-1} & C\theta_i S\alpha_{i-1} & C\alpha_{i-1} & d_i C\alpha_{i-1} \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (48)$$

5.1 Forward kinematics

By the way of replacing the link components into the common homogeneous transform matrix, the distinct transformation matrix is accomplished for every link.

$${}^0_1 T = \begin{bmatrix} C_1 & -S_1 & 0 & 0 \\ S_1 & C_1 & 0 & 0 \\ 0 & 0 & 1 & d_1 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (49)$$

$${}^1_2 T = \begin{bmatrix} C_2 & -S_2 & 0 & a_1 \\ 0 & 0 & 1 & 0 \\ -S_2 & -C_2 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (50)$$

$${}^2_3 T = \begin{bmatrix} C_3 & -S_3 & 0 & a_2 \\ S_3 & C_3 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (51)$$

$${}^3_4 T = \begin{bmatrix} C_4 & -S_4 & 0 & 0 \\ 0 & 0 & 1 & d_4 \\ -S_4 & -C_4 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (52)$$

$${}^4_5 T = \begin{bmatrix} C_5 & -S_5 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ S_5 & C_5 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (53)$$

$${}^5_6 T = \begin{bmatrix} C_6 & -S_6 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ -S_6 & -C_6 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (54)$$

Then to accomplish the position and orientation of end effector, the forward kinematic chain is employed based on the global reference frame.

$${}^0_2 T = {}^0_1 T \times {}^1_2 T = \begin{pmatrix} c_1 c_2 & -c_1 s_2 & -s_1 & c_1 a_1 \\ s_1 c_2 & -s_1 s_2 & c_1 & s_1 a_1 \\ -s_2 & -c_2 & 0 & d_1 \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad (55)$$

$${}^0_3 T = {}^0_2 T \times {}^2_3 T = \begin{pmatrix} c_1 c_{23} & -c_1 s_{23} & -s_1 & c_1(c_2 a_2 + a_1) \\ s_1 c_{23} & -s_1 s_{23} & c_1 & s_1(c_2 a_2 + a_1) \\ -s_{23} & -c_{23} & 0 & -s_2 a_2 + d_1 \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad (56)$$

$${}^4T = {}^4T \times {}^5T = \begin{pmatrix} c_5c_6 & -c_5c_6 & -s_5 & 0 \\ s_6 & c_6 & 0 & 0 \\ s_5c_6 & -s_5s_6 & c_5 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad (57)$$

$${}^3T = {}^3T \times {}^4T = \begin{pmatrix} c_4c_5c_6 - s_4s_6 & -c_4c_5s_6 - s_4c_6 & -c_4s_5 & 0 \\ s_5c_6 & -s_5s_6 & c_5 & d_4 \\ -s_4c_5c_6 - c_4s_6 & s_4c_5s_6 - c_4c_6 & s_4s_5 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad (58)$$

$${}^0T = {}^0T \times {}^3T = \begin{pmatrix} r_{11} & r_{12} & r_{13} & x \\ r_{21} & r_{22} & r_{23} & y \\ r_{31} & r_{32} & r_{33} & z \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad (59)$$

$$\begin{aligned} r_{11} &= c_1c_{23}(c_4c_5c_6 - s_4s_6) - c_1s_{23}s_5c_6 + s_1(s_4c_5c_6 + c_4s_6) \\ r_{12} &= c_1c_{23}(-c_4c_5s_6 - s_4c_6) + c_1s_{23}s_5s_6 - s_1(s_4c_5s_6 - c_4c_6) \\ r_{13} &= -c_1c_{23}c_4s_5 - c_1s_{23}c_5 - s_1s_4s_5 \end{aligned} \quad (60)$$

$$\begin{aligned} r_{21} &= s_1c_{23}(c_4c_5c_6 - s_4s_6) - s_1s_{23}s_5c_6 - c_1(s_4c_5c_6 + c_4s_6) \\ r_{22} &= s_1c_{23}(-c_4c_5s_6 - s_4c_6) + s_1s_{23}s_5s_6 + c_1(s_4c_5s_6 - c_4c_6) \end{aligned}$$

$$\begin{aligned} r_{23} &= -s_1c_{23}c_4s_5 - s_1s_{23}c_5 + c_1s_4s_5 \\ r_{31} &= -s_{23}(c_4c_5c_6 - s_4s_6) - c_{23}s_5c_6 \\ r_{32} &= -s_{23}(-c_4c_5s_6 - s_4c_6) + c_{23}s_5s_6 \\ r_{33} &= s_{23}c_4s_5 - c_{23}c_5 \\ x &= -d_4c_1s_{23} + c_1(c_2a_2 + a_l) \\ y &= -d_4s_1s_{23} + s_1(c_2a_2 + a_l) \\ z &= -s_2a_2 + d_l - d_4c_{23} \end{aligned} \quad (61)$$

Moreover, it is potential to determine the TCP (tool centre point) based on the robot base. Hence, the final position of end effector based on the global reference frame can be given as:

$$P_{tcp} = {}^0T \times P^6$$

$$P_{tcp} = \begin{pmatrix} r_{11} & r_{12} & r_{13} & x \\ r_{21} & r_{22} & r_{23} & y \\ r_{31} & r_{32} & r_{33} & z \\ 0 & 0 & 0 & 1 \end{pmatrix} \times \begin{pmatrix} 0 \\ 0 \\ d_6 \\ 1 \end{pmatrix} = \begin{pmatrix} d_6r_{13} + x \\ d_6r_{23} + y \\ d_6r_{33} + z \\ 1 \end{pmatrix} \quad (62)$$

5.2 Inverse kinematics

Usually, two methods such as geometrical and analytical are used for the inverse kinematics. To calculate the joint angles of θ_1 , θ_2 and θ_3 , the geometrical scheme is employed which represent the position of end effector in the workspace. Likewise, to compute the joint angles of θ_4 , θ_5 and θ_6 , the analytical method is used which denote the orientation of end effector.

5.2.1 Geometrical Solution

The x and y elements of frame (1) is the equivalent to the frame (0) based on the assignment of frame. For the reason that, a Z-directional offset among the two frames. Hence, the wrist element of x-y plane of frame (0) is the identical elements of frame (1). As both link two and three are planar, the location vector is variations in y direction based on θ_1 . Therefore, two potential solutions can be accomplished for θ_1 by utilizing the arctangent function.

$$\begin{aligned}\theta_1 &= a \tan 2(P_{ytcp}, P_{xtcp}) \\ \dot{\theta}_1 &= \pi + \theta_1\end{aligned}\quad (63)$$

By deliberating the plane designed by the 2nd and 3rd planar links, θ_2 and θ_3 are achieved based on the robot reference frame.

$$\theta_2 = -((\Omega - \lambda) - 90) \quad (64)$$

$$\Omega = a \tan 2(s, r) \quad (65)$$

$$\lambda = a \tan 2((d_4 + d_6) \sin(\zeta), a_2 + (d_4 + d_6) \cos(\zeta)) \quad (66)$$

$$s = (P_{ztcp} - d_1) \quad (67)$$

$$r = \pm \sqrt{(P_{xtcp} - a_1 \cos(\theta_1))^2 + (P_{ytcp} - a_1 \sin(\theta_1))^2} \quad (68)$$

$$\cos(\zeta) = \frac{s^2 + r^2 - a_2^2 - (d_4 + d_6)^2}{2a_2(d_4 + d_6)} \quad (69)$$

$$\theta_3 = -(90 + \zeta) \quad (70)$$

$$\zeta = a \tan 2(\sin(\zeta), \cos(\zeta)) \quad (71)$$

$$\sin(\zeta) = \pm \sqrt{1 - \cos^2(\zeta)} \quad (72)$$

The negative sign in θ_3 denotes that the rotation happened in the reverse direction. It is observed that, with different orientations, any position can be accomplished within the robot environment. Hence, owing to the trigonometric functions, the numerous solutions are accomplished for the joint angles of θ_1 , θ_2 and θ_3 .

5.2.2 Analytical Solution

Here, the inverse kinematic is to determine the joint angles of θ_4 , θ_5 and θ_6 that denote the orientation of end effector. Using the Euler's formula, the last orientation matrix can be obtained by the following expression.

$${}_6^0R = R_z(\alpha)R_y(\beta)R_x(\gamma) = \begin{pmatrix} c_\alpha & -s_\alpha & 0 \\ s_\alpha & c_\alpha & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} c_\beta & 0 & s_\beta \\ 0 & 1 & 0 \\ -s_\beta & 0 & c_\beta \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 \\ 0 & c_\gamma & -s_\gamma \\ 0 & s_\gamma & c_\gamma \end{pmatrix} \quad (73)$$

$${}_6^0R = \begin{pmatrix} c_\alpha c_\beta & c_\alpha s_\beta s_\gamma - s_\alpha c_\gamma & c_\alpha s_\beta c_\gamma + s_\alpha c_\gamma \\ s_\alpha c_\beta & s_\alpha s_\beta s + c_\alpha c_\gamma & s_\alpha s_\beta c_\gamma - c_\alpha s_\gamma \\ -s_\beta & c_\beta s_\gamma & c_\beta c_\gamma \end{pmatrix}$$

$${}^3_6R = ({}^0_3R)^T {}^0_6R = \begin{pmatrix} g_{11} & g_{12} & g_{13} \\ g_{21} & g_{22} & g_{23} \\ g_{31} & g_{32} & g_{33} \end{pmatrix} \quad (74)$$

$$\begin{aligned} \theta_4 &= \alpha = a \tan 2\left(\frac{g_{32}}{s_\beta}, \frac{-g_{31}}{s_\beta}\right) \\ \theta_5 &= \beta = a \tan 2\left(+\sqrt{g_{31}^2 + g_{32}^2}, g_{33}\right) \\ \theta_6 &= \gamma = a \tan 2\left(\frac{g_{23}}{s_\beta}, \frac{g_{13}}{s_\beta}\right) \end{aligned} \quad (75)$$

One more flipped solution of θ_4 , θ_5 and θ_6 can be acquired as:

$$\begin{aligned} \dot{\theta}_4 &= 180 + \theta_4 \\ \dot{\theta}_5 &= -\theta_5 \\ \dot{\theta}_6 &= 180 + \theta_6 \end{aligned} \quad (76)$$

5.3 Forward kinematic analysis using SBO-WCO

In this section, the hybrid optimization algorithm of SBO-WCO is utilized to solve the problem of forward kinematics. This hybrid algorithm combines the exploration stage of the WCO to the exploitation (updation) stage of the SBO and it contains three phases such as initialization, exploration and the exploitation process. In SBO [32] approach, mature males started to construct bower for the period of mating season based on the numerous materials on their land. They employs a choice of materials for example, histrionic motions, twinkling entities, fruits, branches and flowers to attract females. Because of the beauty of bower and histrionic motions of males, females are charming to bower. The male birds employ their normal character and pretend of other males to construct the bower. The WCO [33] procedure mimics the teams stimulating to victory the world cup championship in all around the world. The hybrid algorithm can be defined as follows.

At first, the components of SBO-WCO are initialized. Then the quantity of teams are formed by the WCO procedure. The quantity of teams are treated as the set of joint values $[\theta_1 \theta_2 \theta_3 \theta_4 \theta_5 \theta_6]$ achieved from table 3. For every preliminary continents, approximately of the arbitrarily created teams are hypothetical. For every continent, the mean and standard deviation (SD) can be given as:

$$\begin{aligned}\bar{X} &= \frac{1}{n} \sum_{i=1}^n X_i \\ \sigma &= \sqrt{\frac{1}{n-1} \sum_{i=1}^n (X_i - \bar{X})^2}\end{aligned}\quad (77)$$

Here, the mean of continent X is represented by \bar{X} , the amount of members in X is denoted as n and the SD of the continent X is denoted as σ . The ranking of the continents can be given as:

$$Rank = \frac{(\beta \times \sigma + \bar{X})}{2} \quad (78)$$

Here, a sorting operator is denoted as $Rank$ and the coefficient term is denoted as β in the range of $[0, 1]$. Then the amount of iterations are performed. Based on the fitness value, the numerous position of the end effector can be estimated for a specified joint angles in each iteration. Here the fitness function for the forward kinematic is to determine the minimum error which depends on the Euclidean distance among the desired points and the predicted points.

$$Fitness = \sqrt{(P_x' - P_x)^2 + (P_y' - P_y)^2 + (P_z' - P_z)^2} \quad (79)$$

After computing the fitness values, the updation process is performed using SBO to get the optimal position of end effector. The process of updation can be given as:

$$x_{ik}^{new} = x_{ik}^{old} + \lambda_k \left(\left(\frac{x_{jk} + x_{elite,k}}{2} \right) - x_{ik}^{old} \right) \quad (80)$$

Here, the solution vector is defined as x_i in i^{th} bower. The k^{th} member of this vector is defined as x_{ik} . The goal solution is denoted as x_j in the present iteration. The elite location is denoted as x_{elite} . The attraction power is denoted as λ_k in the target bower. This procedure is repeated until the maximum iteration is reached. Finally, the best optimum position values are obtained using this SBO-WCO.

5.4 Inverse kinematic analysis using COA-CFA

In this section, the hybrid optimization algorithm of COA-CFA is utilized to solve the problem of inverse kinematics. This hybrid algorithm combines the exploration stage of the CFA to the exploitation (updation) stage of the COA and it contains three phases such as initialization, exploration and the exploitation process. The COA [34] is motivated the life of a bird family is named as cuckoo. The CFA [35] motivates the color changing behaviour employed by the cuttlefish. The hybrid algorithm can be defined as follows.

This hybrid procedure contains three steps such as initialization, exploration and exploitation. The components of COA-CFA are initialized in the beginning. Then the amount of preliminary solutions are created by CFA. The amount preliminary solutions are treated as the set of optimized position values $[P_x \ P_y \ P_z]$ achieved from the forward kinematics. The population (P) of N preliminary solutions $P = \text{cells} = \{points_1, \dots, points_N\}$ are spread at arbitrary locations over d-dimensional space can be given as:

$$\begin{aligned}P[i].points[j] &= r * (upperLimit - lowerLimit) + lowerLimit \\ i &= 1, \dots, N; j = 1, \dots, d\end{aligned}\quad (81)$$

Here, the lower and upper boundaries are denoted as *lower Limit* and *upper Limit*. The arbitrary number is denoted as r in the range of [0, 1]. A single cell is denoted by every individual $points_i$ in the population. It accompanying with the two values such as fitness and a vector of d-dimension continuous values. Then the number of iterations are executed using this hybrid approach. Based on the fitness value, the joint angles can be estimated for a specified optimized position values of end effector in each iteration.

The fitness for the inverse kinematic is to discover the minimum position error which depends on the Euclidean distance among the desired and the predicted points and it can be given as:

$$Fitness = \sqrt{(p_d - p_c)^2 + (p_d - p_c)^2 + (p_d - p_c)^2} \quad (82)$$

After computing the fitness values, the updation process is performed using COA to get the optimal joint angles. The procedure of updation can be defined as follows:

$$\begin{aligned} \lambda &\sim U(0,1) \\ \varphi &\sim U(-\omega, \omega) \end{aligned} \quad (83)$$

In all surroundings, the above-mentioned two components of λ and φ to support the cuckoos search much more locations. An arbitrary number is denoted as λ in the range of [0, 1]. The component ω makes the deviation from target habitat. This procedure is repeated until the maximum iteration is reached. Finally, the best optimum joint values are obtained using this COA-CFA.

6. Trajectory Planning of industrial robots using hybrid optimization algorithms

In this section, the problem of trajectory planning is solved using the hybrid optimization algorithms of CSA-BSA, GWO-GOA and SOA-TSA for the industrial robots of PUMA 560 (6 DOF), KUKA LBR IIWA 14 R820 (7 DOF) and ABB IRB 140 (6 DOF) individually. The planning of trajectory undertakes that the symmetrical path of end effector specified by users in advance. The sequence of positions and orientation of the end effector is involved in the path in Cartesian space. By the way of inverse kinematics, every joints can be accomplished.

6.1 Trajectory Planning Problem

In this paper, for the planning of trajectory three industrial robots such as PUMA 560, KUKA LBR IIWA 14 R820 and ABB IRB 140 are considered. Consider the robot executes the task with its end effector passing n points which including initial and target points. Using the inverse kinematics, the N points are transformed into ' n ' equivalent joint variables of the joint space. The robot is travel in the direction of ending point while avoiding the obstacles in the robotic environment with considering the minimum objective function. Most of the autonomous robots are subjected to collision avoidance problem which is considered as the major issue in different industries. Therefore, the major objective of the planning of trajectory is to determine the best optimum collision free path from the starting point to the destination point for the industrial robots. Using the objective function, the ideal solution to avoid the collision.

In this work, the multi-objective functions such as collision free motion, jerk, travelling time and acceleration are considered for the planning of trajectory of the robot manipulator. The fitness function is reduces the multi-objective function. The fitness function problem for the trajectory planning of the industrial robots can be defined as:

$$Fitness = \min \begin{cases} \text{travelling time} \\ \text{collision free motion} \\ \text{joint jerks} \\ \text{joint accelerations} \end{cases} \quad (84)$$

Travelling time can be given as:

$$T = T_1 + T_2 + \dots + T_{i-1} = \sum_{i=1}^{n-1} T_i \quad (85)$$

The measure of obstacle avoidance is represented by the distance among possibly colliding portions. By the way of translation and rotational matrices, the motion is defined. To assure the collision free motion, the penalty function is included in the multi-criterion issue if the obstacles are established in the robotic environment. The concept is to define each obstacle into a particular sphere. The points of trajectory is positioned outside the sphere based on the subsequent equation.

$$r_t = \sqrt{(x - x_0)^2 + (y - y_0)^2 + (z - z_0)^2} > r_0 \quad (86)$$

The penalty function (f_{dis}) is zero when the aforementioned equation is proved the trajectory is external of the sphere. Otherwise the penalty is included to the multi-criterion issue when the trajectory is crossing the sphere.

$$f_{dis} = \begin{cases} 0 & \text{if } r_t > r_0 \\ \sum_{i=1}^{nobs} \frac{1}{(\min r_i)^2} & \text{if } r_t \leq r_0 \end{cases} \quad (87)$$

The integral of squared joint jerks:

$$C = \int_0^T \sum_{i=1}^n \ddot{\alpha}_i^2 dt \quad (88)$$

The integral of squared joint accelerations:

$$D = \int_0^T \sum_{i=1}^n \dot{\alpha}_i^2 dt \quad (89)$$

Subject to the constraints of

$$\text{Displacement } \max |q_{ji}(t)| \leq QC_j$$

$$\text{Velocity } \max |\dot{q}_{ji}(t)| \leq VC_j$$

$$\text{Acceleration } \max |\ddot{q}_{ji}(t)| \leq WC_j$$

$$\text{Jerk } \max |J_{ji}(t)| \leq JC_j$$

$$\text{Obstacle avoidance } d_{lq}(t) > 0 \text{ for } (l, q) \in I_d$$

Here, the robot joint number is denoted as i , maximum displacement is denoted as QC, VC is the maximum velocity, WC is the maximum acceleration and JC is the maximum jerk. The arrangement of probably colliding pairs of parts is denoted as I_d , the displacement of robot joint i at time t is denoted as $q_{ji}(t)$, velocity of robot joint i is denoted as $\dot{q}_{ji}(t)$ at time t and acceleration of robot joint i at time t is denoted as $\ddot{q}_{ji}(t)$. \ddot{q}_{ji} is jerk of robot, the distance among the robot $d_l(t)$ and the obstacle $d_q(t)$ is defined as $d_{lq}(t)$, and the extreme amount of robot joint is represented as

n . The entire amount of obstacles in the workspace is denoted as $nobs$, the distance among the center of obstacle and a trajectory point is denoted as r and the radius of the sphere is defined as r_0 .

6.2 Solving trajectory planning of PUMA 560 (6 DOF) using CSA-BSA

To solve the trajectory planning problem of PUMA 560 robot manipulator, the hybrid optimization algorithms of CSA-BSA is used in this section. This hybrid approach combines the exploration stage of BSA and the exploitation stage of CSA. BSA [37] motivates the social behaviors and interaction in bird swarms. The birds consist of three behaviours such as flight, vigilance and foraging behavior. In the CSA [36], in hiding places each crow store their additional food and recover it while the food is required. The hybrid algorithm can be defined as follows.

This hybrid algorithm contains three stages such as initialization, exploration of BSA and exploitation of CSA. At first, the components of CSA-BSA are initialized. Then the number of individuals are created. The number of individuals are treated as the amount of paths. Hence, numerous paths are made between the initial and target point. Then the fitness values are calculated using the Eqn (84) for each path. The robot move towards the destination point with the minimum objective function. At time t , every N individual bird is represented by their locations $x_i^t (i \in [1, \dots, N])$. They fly in a D-dimensional space and search for food. The bird's behaviours can be given as:

$$x_{i,j}^{t+1} = x_{i,j}^t + (p_{i,j} - x_{i,j}^t) \times c \times rand(0,1) + (g_j - x_{i,j}^t) \times s \times rand(0,1) \quad (90)$$

$$x_{i,j}^{t+1} = x_{i,j}^t + A1(mean_j - x_{i,j}^t) \times rand(0,1) + A2(p_{k,j} - x_{i,j}^t) \times rand(-1,1) \quad (91)$$

$$A1 = a1 \times \exp\left(-\frac{pFit_i}{sumFit + \varepsilon} \times N\right) \quad (92)$$

$$A2 = a2 \times \exp\left(\left(\frac{pFit_i - pFit_k}{|pFit_k - pFit_i| + \varepsilon}\right) \frac{N \times pFit_k}{sumFit + \varepsilon}\right) \quad (93)$$

$$x_{i,j}^{t+1} = x_{i,j}^t + randn(0,1) \times x_{i,j}^t \quad (94)$$

$$x_{i,j}^{t+1} = x_{i,j}^t + (x_{k,j}^t - x_{i,j}^t) \times FL \times randn(0,1) \quad (95)$$

Here, $j \in [1, \dots, D]$, the arbitrary number is denoted as $rand(0,1)$, cognitive factor is denoted as c and social accelerated factor is defined as s . The finest earlier location of i^{th} bird is denoted as $p_{i,j}$, the finest earlier location pooled by the swarm is denoted as g_j , and the positive integer is defined as $k (k \neq i)$ in the range of $[1, N]$. The positive constants are denoted as $a1$ and $a2$ in the range of $[0, 2]$. The finest fitness value of i^{th} bird is represented as $pFit_i$. The entire swarm's finest fitness value is denoted as $sumFit$. To eliminate the zero-division error, the lowest parameter of ε is used. The average location of the entire swarm is denoted as $mean_j$ in j^{th} component. $FL (FL \in [0, 2])$ is the borrower wanted trail the creator to hunt for food.

After that, the new solutions are created. Then to obtain the finest optimal path, the solutions are updated by CSA if the new solutions are superior to their prior solution. The procedure of updation can be given as:

$$m^{i,iter+1} = \begin{cases} x^{i,iter+1} & f(x^{i,iter+1}) \text{ is better than } f(m^{i,iter}) \\ m^{i,iter} & \text{otherwise} \end{cases} \quad (96)$$

Here the objective function is denoted as $f(\cdot)$. This procedure is repeated until the maximum iteration is reached. At last, the finest optimal path is achieved with the minimum fitness value for the PUMA 560 robot. .

6.3 Solving trajectory planning of KUKA LBR IIWA 14 R820 (7 DOF) using GWO-GOA

To solve the trajectory planning problem of KUKA LBR IIWA 14 R820 robot manipulator, the hybrid optimization algorithms of GWO-GOA is used in this section. This hybrid approach combines the exploration stage of GWO and the exploitation stage of GOA. The GWO [38] process motivates the searching procedure of grey wolves and the management hierarchy in the environment. GOA [39] process motivates the grasshopper swarms behaviour in the surroundings. The hybrid algorithm can be defined as follows.

This process consist of three steps such as initialization, exploration and exploitation. At first, the components of GWO-GOA are initialized. Then the amount of populations are created. In this work, the number of population is treated as the amount of paths. Hence, the numerous paths are created from the source to ending point. For every path, the fitness values are evaluated using Eqn (84). The robot move towards the destination point with the minimum objective function. For the period of searching, the grey wolves enclose the prey. The enclosing behaviour can be given by the following expressions.

$$D = |C \cdot X_p(t) - X(t)| \quad (97)$$

$$X(t+1) = X_p(t) - A \cdot D \quad (98)$$

Here, the present iteration is denoted as t , the location vector of prey is denoted as X_p and the location vector of grey wolf is represented as X . The coefficient vectors of A and C can be given as:

$$\begin{aligned} A &= 2a \cdot r_1 - a \\ C &= 2 \cdot r_2 \end{aligned} \quad (99)$$

Here, the parameter a is linearly reduced from 2 to 0. The arbitrary vectors are signified as r_1 and r_2 in the range of $[0, 1]$. Then by the way of GOA process, the updation procedure is performed. The updation process can be given as:

$$X_i^d = c \left(\sum_{\substack{j=1 \\ j \neq i}}^N c \frac{ub_d - lb_d}{2} s(|x_j^d - x_i^d|) \frac{x_j - x_i}{d_{ij}} \right) + T_d \quad (100)$$

Here, the upper and lower limit is denoted as ub_d and lb_d in D^{th} dimension, $s(r) = fe^{-\frac{r}{l}} - e^{-r}$, target value is denoted as T_d and reducing factor is denoted as c which decreases the attraction, comfort and repulsion zone. This procedure is repeated until the maximum iteration is reached. At last, the finest optimal path is achieved with the minimum fitness value for the KUKA LBR IIWA 14 R820 robot.

6.4 Solving trajectory planning of ABB IRB 140 (6 DOF) using SOA-TSA

To solve the trajectory planning problem of ABB IRB 140 robot manipulator, the hybrid optimization algorithms of SOA-TSA is used in this section. This hybrid approach combines the exploration stage of TSA and the exploitation stage of SOA [40]. TSA [41] mimics the jet impulsion and the swarm behavior of tunicates for the period of foraging and navigation procedure. SOA procedure motivates the attacking and the migration behavior of a seagull in the environment. The hybrid algorithm can be defined as follows.

This hybrid procedure consist of three phases such as initialization, exploration and exploitation. At first, the components of SOA-TSA are initialized. Then the amount of tunicate population are created. This population treated as the amount of paths. From the starting to final point, the numerous paths are created. For each path, the fitness values are evaluated using Eqn (84). With the minimum fitness value, the robot travels towards the target point. To eliminate the collision among other tunicates, the parameter of \vec{A} is used to find the new search agent location.

$$\vec{A} = \frac{\rho}{\rho} \vec{G} \quad (101)$$

$$\vec{G} = c_2 + c_3 - \vec{F} \quad (102)$$

$$\vec{F} = 2.c_1 \quad (103)$$

$$\vec{M} = [P_{\min} + c_1.P_{\max} - P_{\min}] \quad (104)$$

Here, the gravity force is denoted as \vec{G} , in Deep Ocean the flow of water advection is denoted as \vec{F} , the arbitrary number is denoted as c_1, c_2 , and c_3 in the range of $[0, 1]$, the social forces among search agents are denoted as \vec{M} , and to create the social communication the primary and secondary speeds are denoted as P_{\min} and P_{\max} . Then the search agents travels in the direction of the finest neighbor after eliminating the collision among neighbors.

$$\vec{PD} = \left| \vec{FS} - r.P_p(x) \right| \quad (105)$$

Here, the distance among the source of food and search agent is denoted as \vec{PD} , the present iteration is defined as x , the location of food source is denoted as \vec{FS} , the location of tunicate is defined as $P_p(x)$ and the arbitrary number is signified as r in the range of $[0, 1]$. Then by the way of SOA, the updation process is executed. The modernized location of search agent can be given as:

$$\vec{P}_s(x) = (\vec{D}_s \times x' \times y' \times z') + \vec{P}_{bs}(x) \quad (106)$$

$$\vec{D}_s = \left| \vec{C}_s + \vec{M}_s \right| \quad (107)$$

$$x' = r \cos(k) \quad (108)$$

$$y' = r \sin(k) \quad (109)$$

$$z' = r \times k \quad (110)$$

$$r = u \times e^{kv} \quad (111)$$

Here, the distance among the search agent and the finest fit search agent is denoted as \vec{D}_s , the radius of every turn of the spiral is defined as r , and the arbitrary number is defined as k in the

range of $[0 \leq k \leq 2\pi]$. The shape of spiral is represented by \mathcal{U} and \mathcal{V} . The base of natural algorithm is denoted as e . The finest solution is stored by $P_{bs}(x)$ and the location of other search agents are updated. This procedure is repeated until the maximum iteration is found. At last, the best optimum path is achieved for the ABB IRB 140 robot manipulator which satisfies the minimum fitness function.

7. Simulation Results and Discussions

The kinematic model and trajectory planning problems for an industrial robot manipulators based on the hybrid optimization algorithms were simulated on robotic tool box in MATLAB with GUI interface. For an evaluation and simulation of arm type robotics in the fields of dynamics, kinematics and creation of trajectory, the toolbox does numerous functions. The toolbox is only depends on the common scheme of demonstrating the kinematics and dynamics of serial-link manipulators. These constraints are involved in MATLAB. GUI provides a set of tools to create GUIs (graphical user interfaces). For the simulation purpose, three industrial robots are considered in this paper. They are PUMA 560, KUKA LBR iiwa 14 R820 and ABB IRB 140 robot manipulators. In this work, 18 meta-heuristic optimization approaches such as PSO, SSO, DFO, BOA, CSA, BSA, SHO, KHO, ALO, ACO, GWO, GOA, SBO, WCO, COA, CFA, SOA and TSA are utilized for the solving both forward and inverse kinematics and the trajectory planning issue for the industrial robot manipulators of PUMA 560, KUKA LBR iiwa 14 R820 and ABB IRB 140.

7.1 Simulation model for PUMA 560

This section given the simulation analysis of PUMA 560 (6 DOF) robot. Here, the forward and the inverse kinematics and the trajectory planning problems are solved using the hybrid optimization algorithm for the PUMA 560 robot. The hybrid approaches of PSO-SSO, DFO-BOA and CSA-BSA are utilized to solve both forward and inverse kinematic model and trajectory planning problem. In PSO, the value of cognitive factor (c_1) is 1.4, social factor (c_2) is 1.4, inertia weight (w) is 0.9~0.4, the number of particle is 20 and the random numbers of r_1 and r_2 in the range of [0, 1]. In SSO, the value of user controlled parameter of r_a and p_m is 1 and 0.1 individually. The user defined attribute (p_c) is 0.7. In DFO, separation weight (s) is 0.1, alignment weight (a) is 0.1, cohesion weight (c) is 0.7, food factor (f) is 1, enemy factor (e) is 1 and the inertia weight (w) is 0.9~0.2. In BOA, size of population is 50, modular modality (c) is 0.01 and power exponent (a) is 0.1 to 0.3. In CSA, the size of population is 20, awareness probability (AP) is 0.1 and flight length (fl) is 2. In BSA, the constant parameters of c , s , a_1 , and a_2 are 1.5, 1.5, 1, and 1 individually. The frequency of bird's flight behavior (FQ) is denoted as 3. For each algorithm, the maximum iteration is 100.

Kinematics of robot is to learn the connection between the space of joint angle and the position and attitude of robot terminal actuator without deliberating the motion force. The positional kinematics only handle the motion geometry without considering the motion time. The speed kinematic issue is significant due to the operator essentials to reach the particular position and also desires to reach these locations at a specified speed. PUMA 560 (6 DOF) industrial robot is designed by robotics toolbox is given in figure 4.

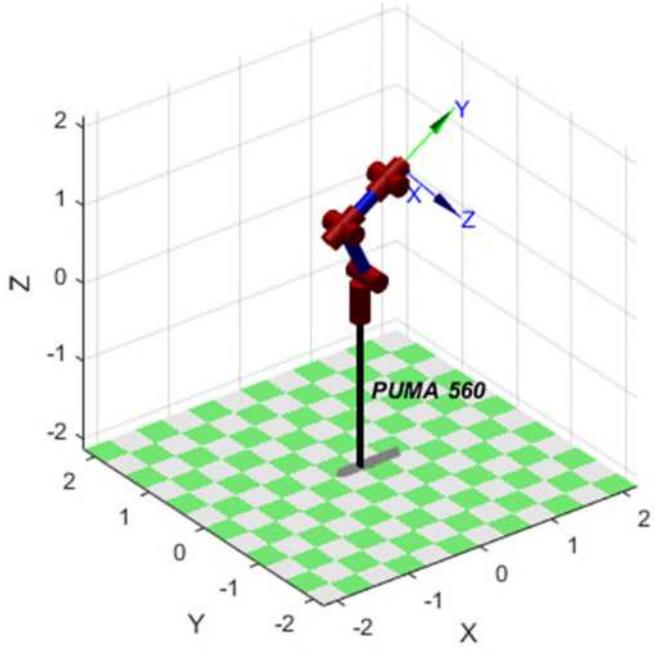


Figure 4: PUMA 560 (6 DOF) structure

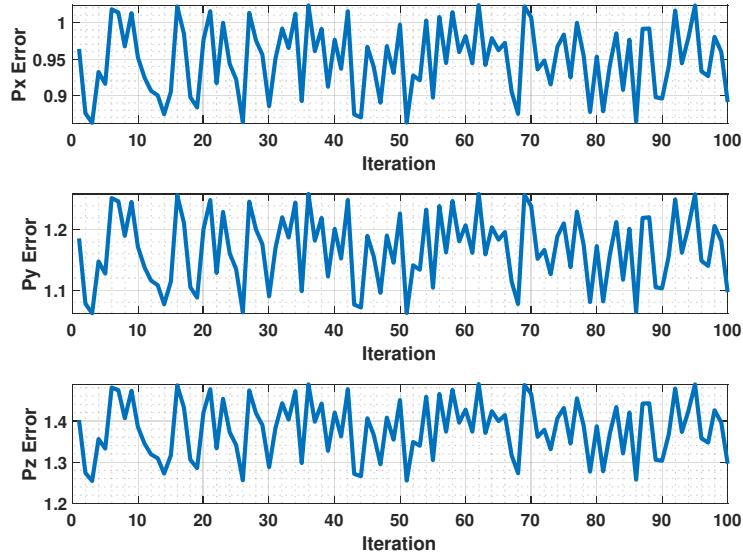


Figure 5: Position error value for the forward kinematics problem

The fitness for the forward kinematic problem is shown in figure 5. The fitness function is to compute the minimum error position value which depends on the Euclidean distance among the predicted and the target points. In this work, the forward kinematic problem is converted as the optimization issue which is resolved with the help of the hybrid algorithms of PSO-SSO. For a particular joint angles, the optimized position values are determined with the minimum position error value in the forward kinematics. The fitness function is reduced in the forward kinematics. For each algorithm, the maximum iteration is taken as 100. For each iteration, the different position of end effector is calculated based on the optimization scheme. The current position value reach the target points with the minimum error value. PUMA 560 is a 6 DOF robot. So the input is 6 joint angles. Finally, the optimized position values are obtained in x, y and z axis.

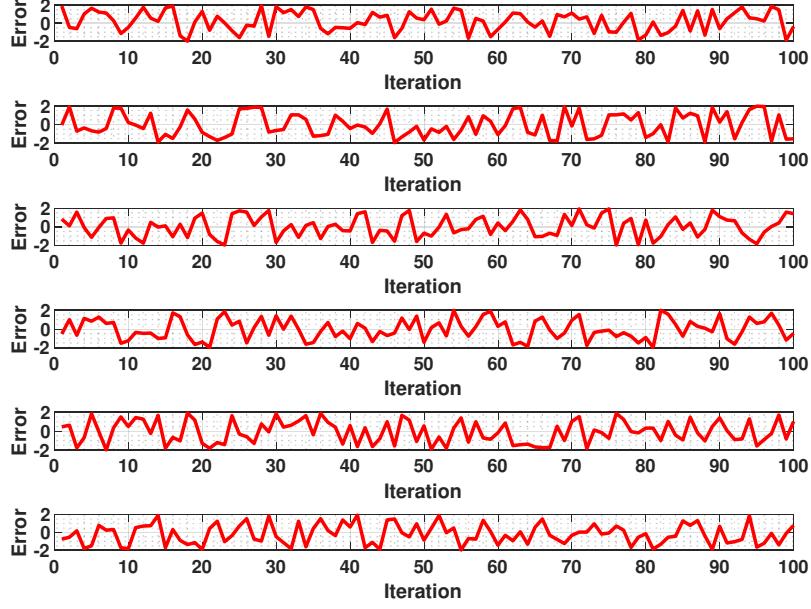


Figure 6: Position error value for the inverse kinematics problem

The inverse kinematic is to find the joint angles for a specified position value of the end effector. The problem of inverse kinematic is converted as the optimization problem in this work which is solved by DFO-BOA. For a given position values, the different joint angles are computed in each iteration. The maximum iteration of each algorithm is 100. Fitness function of the inverse kinematic is Euclidean distance among the predicted and target location which means the minimum joint angles. This minimum value provide the best position error value. Figure 6 shows the position error of inverse kinematic problem. Finally, the optimal joint angles are calculated with the minimum position error values using DFO-BOA approach. The position error value for the 6 joint angles of PUMA 560 robot is given in the figure 6.

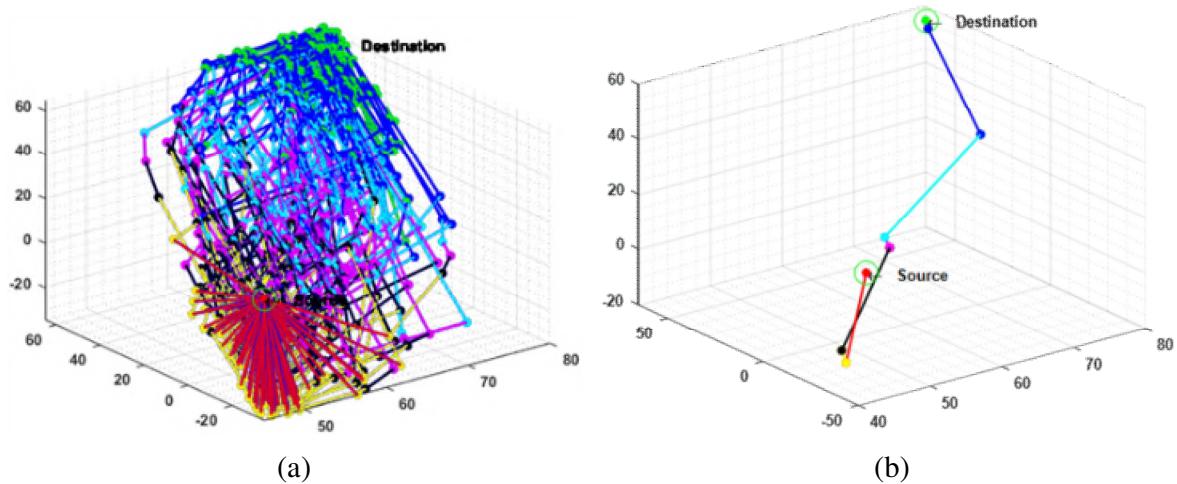


Figure 7: Trajectory planning of PUMA 560 (a) Different paths without obstacles (b) optimized path without obstacles using optimization algorithm

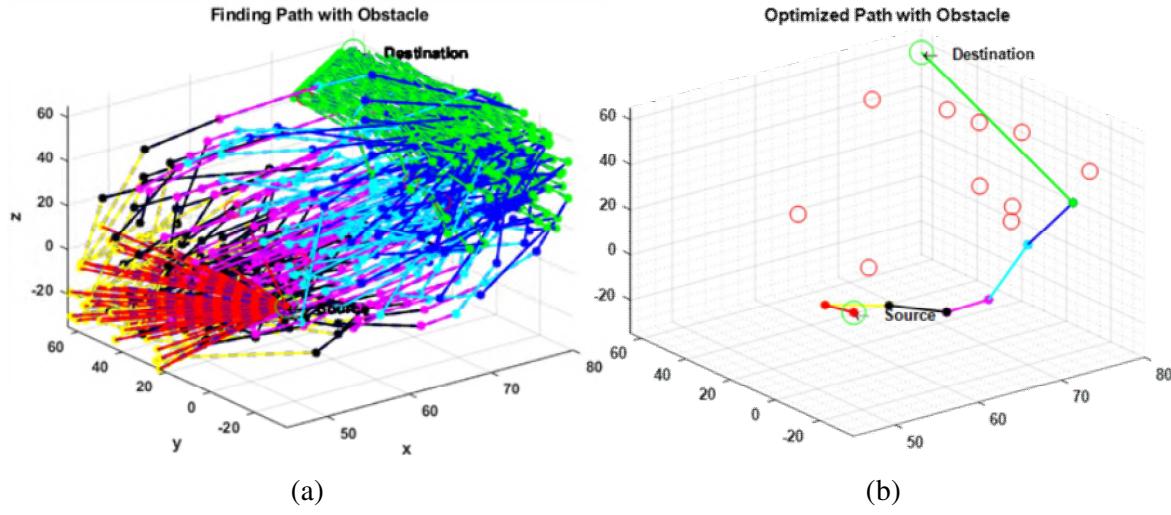


Figure 8: Trajectory planning of PUMA 560 (a) Different paths with obstacles (b) optimized path with obstacles using optimization algorithm

Trajectory planning is the most significant problem in the industrial robot manipulators. The trajectory planning of robot is to design the velocity, acceleration and the displacement of each joint in the motion of robot for a specified path point. Moreover, the related kinematics and dynamic constraints are satisfied. The trajectory problem of PUMA 560 robot is solved by CSA-BSA approach. In the PUMA 560 robot workspace, the source and destination points are randomly generated. From the source points, the robot move towards the target points for the position and orientation of end effector and the joint angles. The proposed CSA-BSA approach reduce fitness function of the trajectory planning the PUMA 560 robot. The best optimal path is determined with obstacles and the without obstacles. The multi-objective function such as travelling time, joint jerks, collision free motion and joint acceleration are used for the trajectory planning of PUMA 560 robot. Figure 7 (a) shows the generation of multiple paths in the robotic workspace. From the initial to final point, the numerous paths are generated without considering any obstacles for the given position value of end effector and orientation and the joint angles. In this work, only 100 paths are generated without any obstacles. In 100 paths, the minimum amount of paths are only reached the destination point. From this, the one best optimized path is determined without any obstacles using CSA-BSA approach in which the path is satisfies the minimum fitness value as shown in figure 7 (b). Figure 8 (a) illustrates the generation of numerous paths with obstacles. In the working area, the different obstacles are located and the 100 paths are generated with obstacles. In this numerous paths, the least amount of paths only reached the destination point. From this, the best optimal path is calculated with the help of the CSA-BSA approach. The optimal path satisfies the fitness function. The fitness function is reduce the travelling time, jerk value, acceleration, and the collision free motion. At last, the robot move towards the destination point in the optimal path.

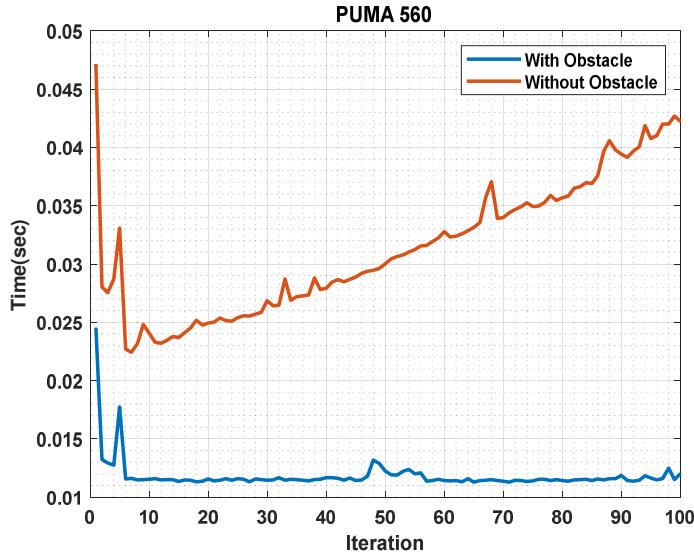


Figure 9: Traveling time performance

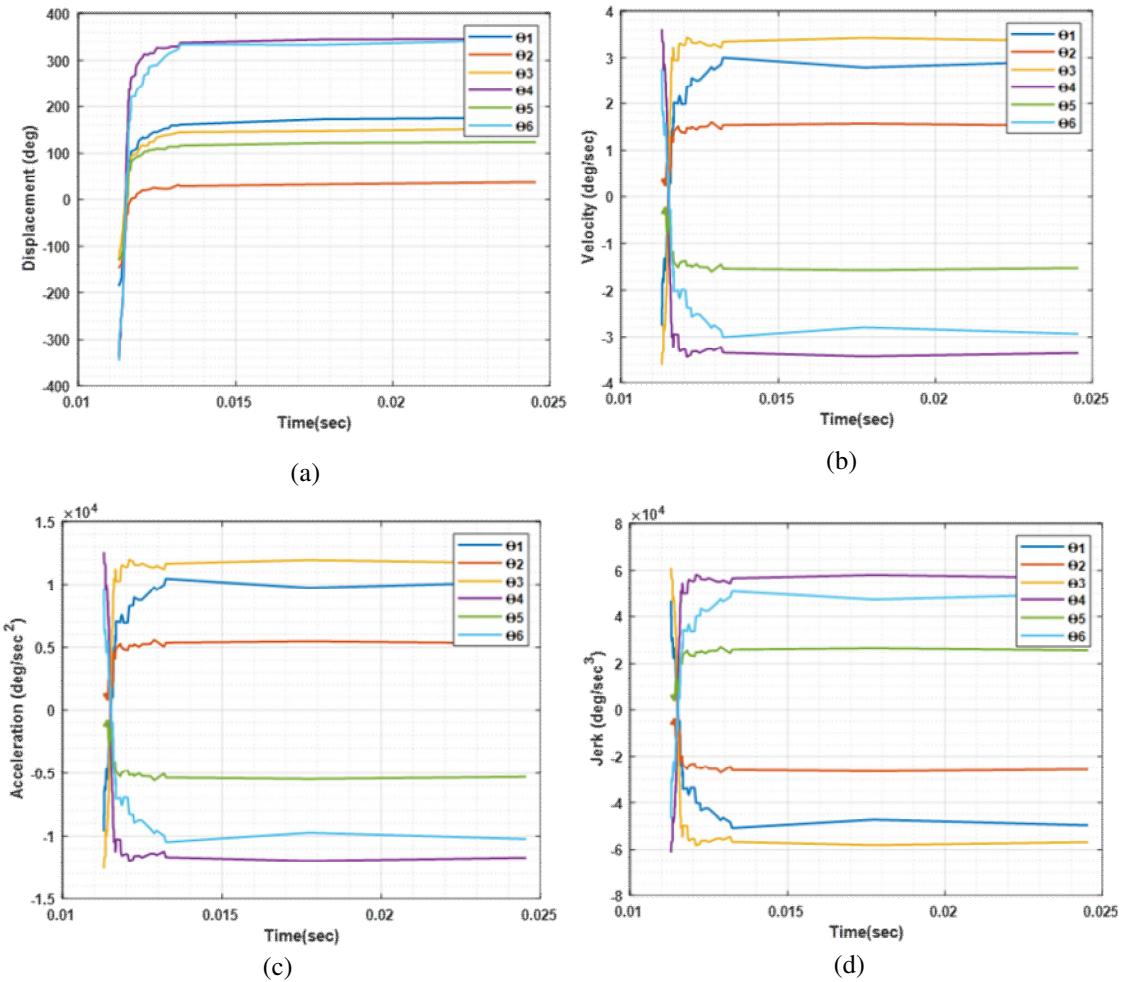


Figure 10: (a) displacement (b) velocity (C) acceleration and (d) jerk of optimized trajectories of PUMA 560 robot without obstacles

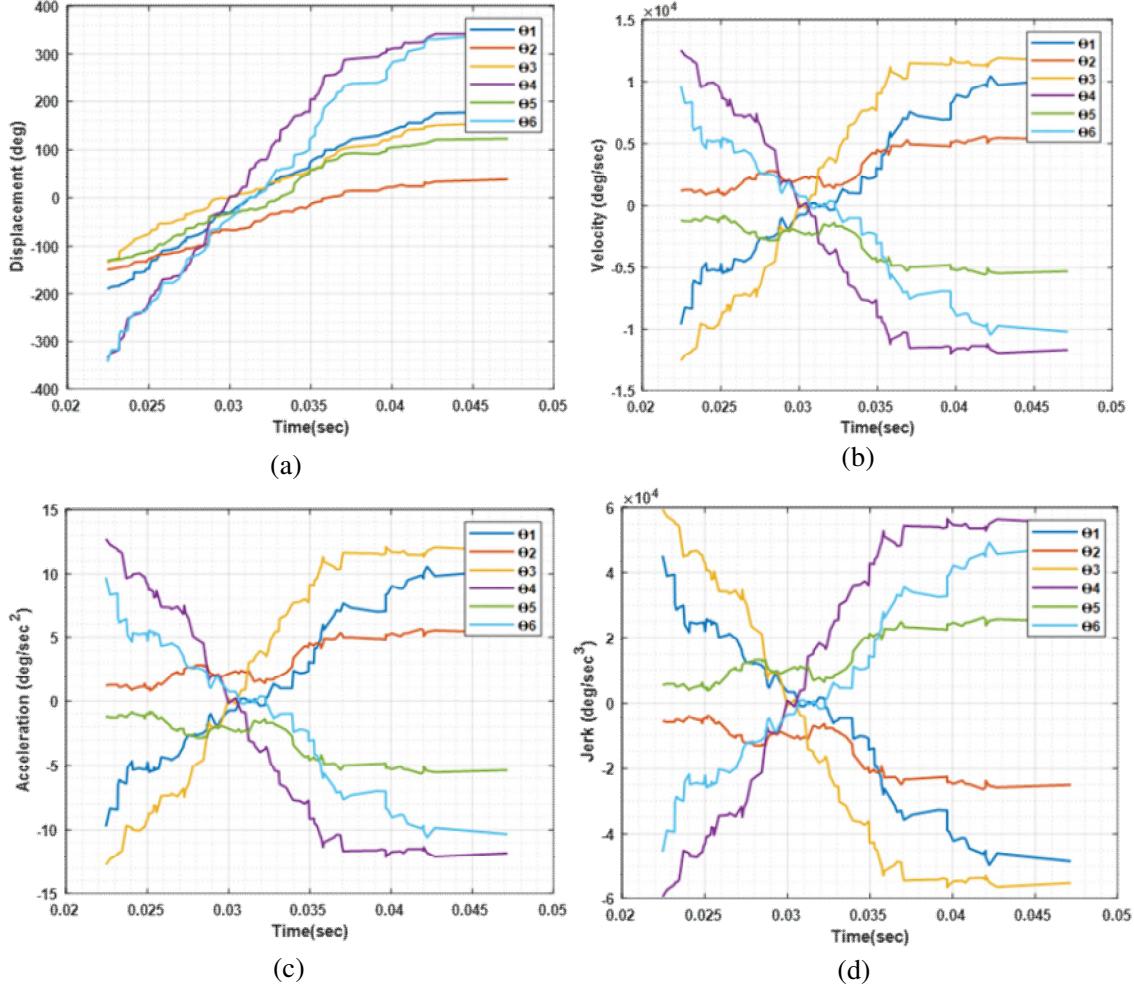


Figure 11: (a) displacement (b) velocity (C) acceleration and (d) jerk of optimized trajectories of PUMA 560 robot with obstacles

Total traveling time for with and without obstacles is evaluated for the PUMA 560 robot as shown in figure 9. For the different iteration, the time is calculated. The travelling time is decreased in the fitness function. The minimum travelling time of path is considered as the best optimal path. Time is one of the objective function. Then other constraints of displacement, velocity, acceleration and jerk of the optimal trajectories of PUMA 560 robot is evaluated for each joint angle. For without and with obstacles, the robot movement is estimated with regards to the displacement, velocity, acceleration and jerk for each joint angles as shown in figure 10 and 11. The displacement, velocity, acceleration and jerk curves are smooth and can reduce shock and impact of the robot arm and ensure smooth operation of the robot. The displacement of the trajectory is to control the end effector that means angles or positions in workspace, maybe involving motion dynamics. The safety is directly connected with the velocity and comfort with the jerk in the human interaction context. In Cartesian space, such constrained movements are soft even for rotations, starts and stops. The time variation is described by the optimal motion. The optimum motion is a motion with acceleration, velocity, and jerk constraints continuously saturated.

7.2 Simulation model for KUKA LBR iiwa 14 R820

This section given the simulation analysis of KUKA LBR iiwa 14 R820 robot. Here, the forward and the inverse kinematics and the trajectory planning problems are solved using the hybrid optimization algorithm for the KUKA LBR iiwa 14 R820 robot. The hybrid approaches of SHO-KHO, ALO-ACO and GWO-GOA are utilized to solve both forward and inverse kinematic model and trajectory planning problem. In SHO, the size of population is 50. In KHO, the inertia weight (w) is 0.9. In ALO, the initial population is 30. In ACO, the number of ants are 40, the impact of pheromone (α) is 1, the impact of desirability (β) is 1 and the pheromone evaporation rate (ρ) is 0.5. In GWO, population size is 200, number of appliances is 12 and coefficient factor is 0 to 2. In GOA, search agent is 30. For each algorithm, the maximum iteration is 100.

For a given joint angles, the end effector position is calculated in the forward kinematics. The input of the joint angles are obtained from DH convention of KUKA LBR iiwa 14 R820 as shown in table 2. With the help of the SHO-KHO, the optimized position of end effector is achieved with the minimum position error value. The optimized position value of end effector is considered as the input value to the inverse kinematics. Finally, the optimized joint angles are evaluated using ALO-ACO with minimum joint angle error. KUKA LBR iiwa 14 R820 (7 DOF) industrial robot is designed by robotics toolbox is given in figure 12.

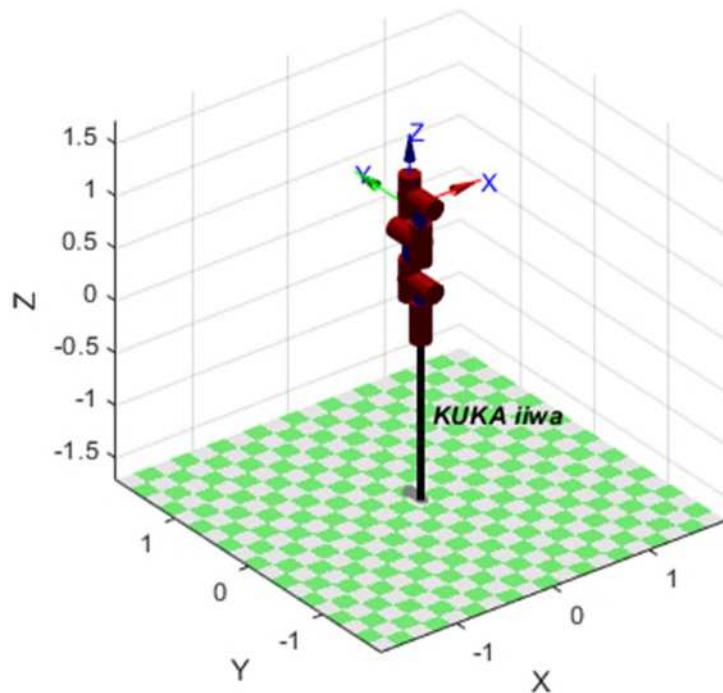


Figure 12: KUKA LBR iiwa 14 R820 structure

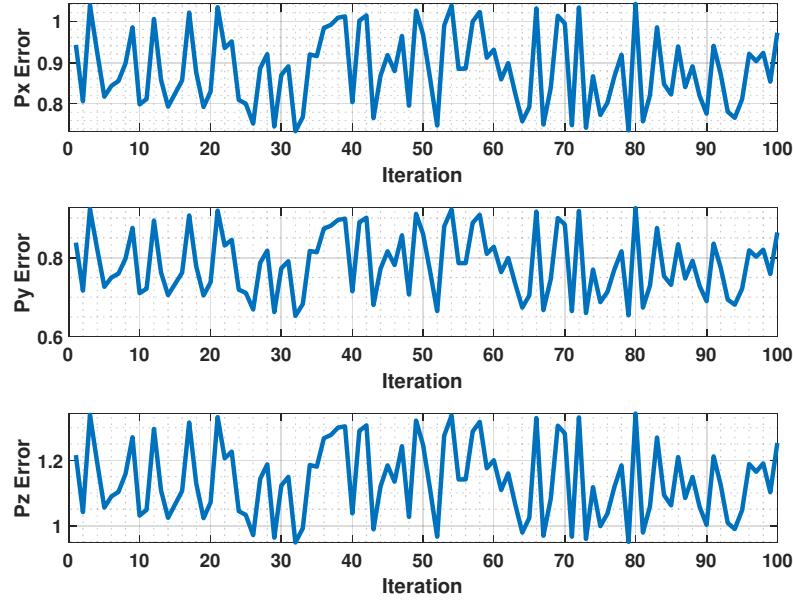


Figure 13: Position error value for the forward kinematics problem

The fitness for the forward kinematic problem is shown in figure 13. The fitness function is to compute the minimum error position value which depends on the Euclidean distance among the desired and the current points. In this work, the forward kinematic problem is converted as the optimization issue which is resolved with the help of the hybrid algorithms of SHO-KHO. For a particular joint angles, the optimized position values are determined with the minimum position error value in the forward kinematics. The fitness function is reduced in the forward kinematics. For each algorithm, the maximum iteration is taken as 100. For each iteration, the different position of end effector is calculated based on the optimization scheme. The current position value reach the target points with the minimum error value. KUKA LBR iiwa 14 R820 is a 7 DOF robot. So the input is 7 joint angles. Finally, the optimized position values are obtained in x, y and z axis.

The inverse kinematic is to find the joint angles for a specified position value of the end effector. The problem of inverse kinematic is converted as the optimization problem in this work which is solved by ALO-ACO. For a given position values, the different joint angles are computed in each iteration. The maximum iteration of each algorithm is 100. Fitness function of the inverse kinematic is Euclidean distance among the predicted and target location which means the minimum joint angles. This minimum value provide the best position error value. Figure 14 shows the position error of inverse kinematic problem. Finally, the optimal joint angles are calculated with the minimum position error values. KUKA robot actually contains 7 joint angles. But in the inverse kinematics only 6 joint angles are determined because one angle is fixed in the KUKA LBR iiwa 14 R820 manipulator. This is a redundant manipulator and one angle is redundancy. Therefore only 6 optimized joint angles are calculated with the minimum fitness value in the inverse kinematics using ALO-ACO.

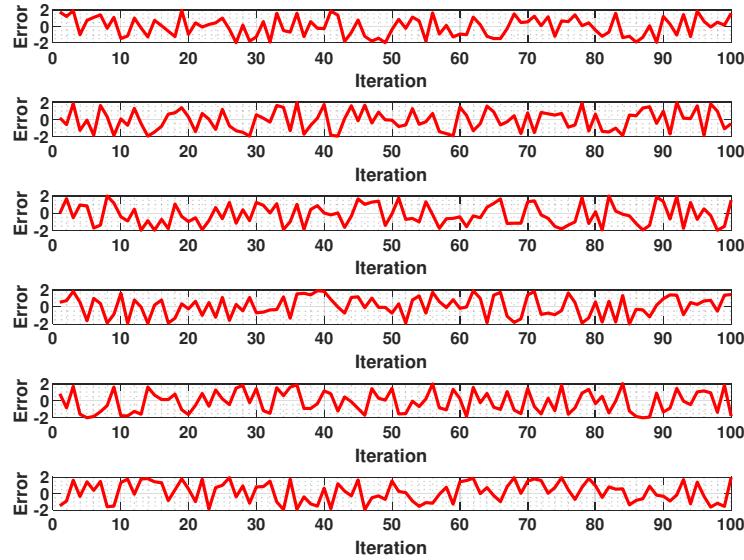


Figure 14: Position error value for the inverse kinematics problem

Figure 15 and 16 shows the trajectory planning of KUKA robot with and without obstacles. The trajectory planning problem is converted as the optimization problem which is solved by the GWO-GOA approach. Figure 15 (a) illustrates the multiple paths without considering any obstacles. In the robotic workspace, the one source and destination point is generated. From the starting and ending points, the 100 paths are created without considering any obstacles. In the different paths, the minimum amount of paths only reached the destination point. From this, the best optimal path is determined with the minimum fitness value using the GWO-GOA approach. The best optimized path satisfies the minimum objective function as shown in figure 15 (b). The fitness function reduce the jerk, acceleration, travelling time and collision free motion. Similarly in the Figure 16 (a) the 100 paths are created between source and target points with considering the different obstacles. The least amount of paths only reached the destination point. From this, the best optimal path is determined using the GWO-GOA which satisfies the minimum fitness value as shown in figure 16 (b). Finally, from the source point, the robot travels towards the destination point in the best optimal path.

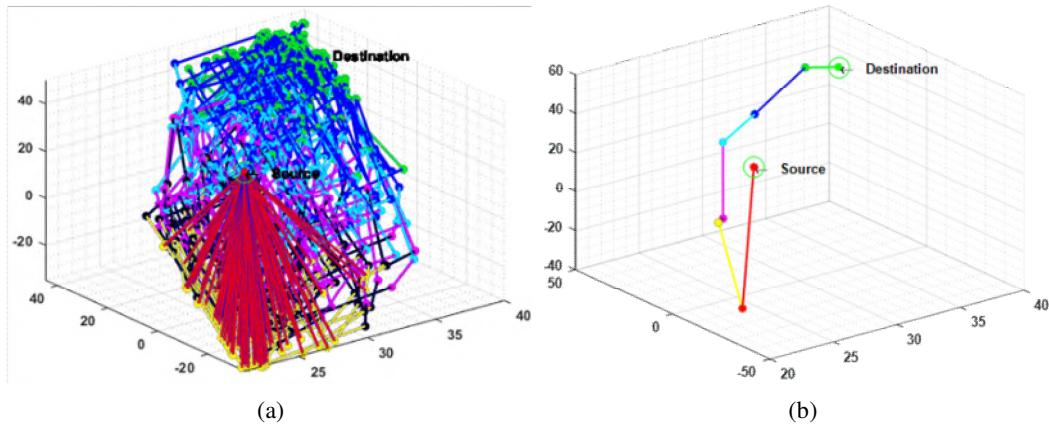


Figure 15: Trajectory planning of KUKA LBR iiwa 14 R820 (a) Different paths without obstacles (b) optimized path without obstacles using optimization algorithm

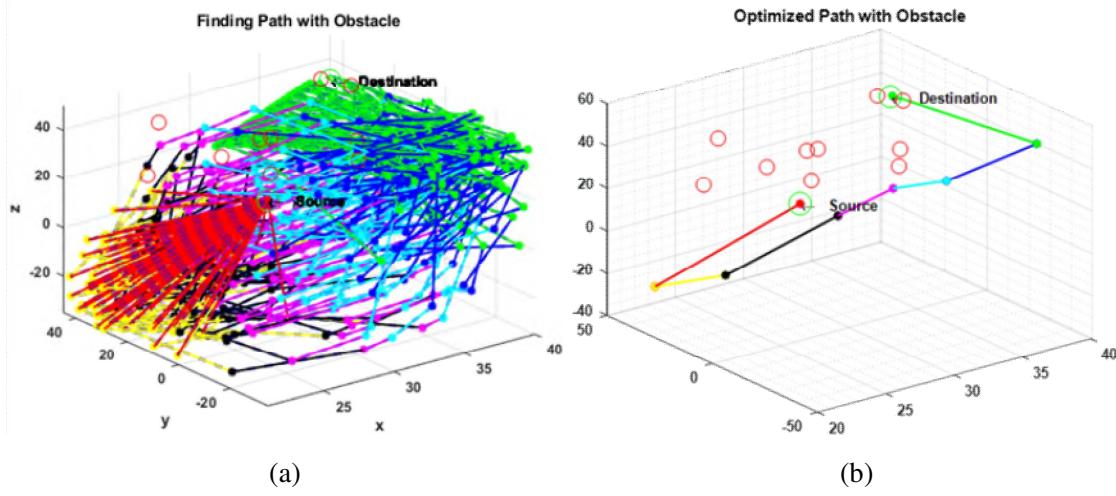


Figure 16: Trajectory planning of KUKA LBR iiwa 14 R820
 (a) Different paths with obstacles
 (b) optimized path with obstacles using optimization algorithm

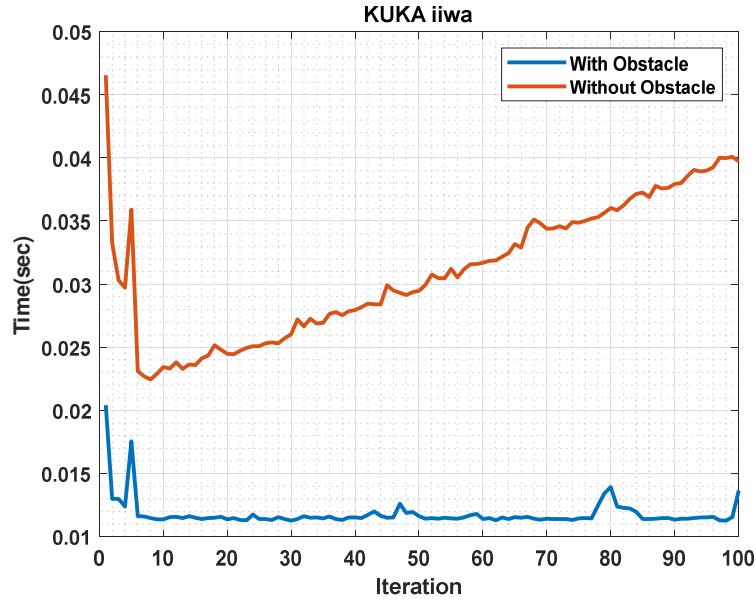


Figure 17: Traveling time performance

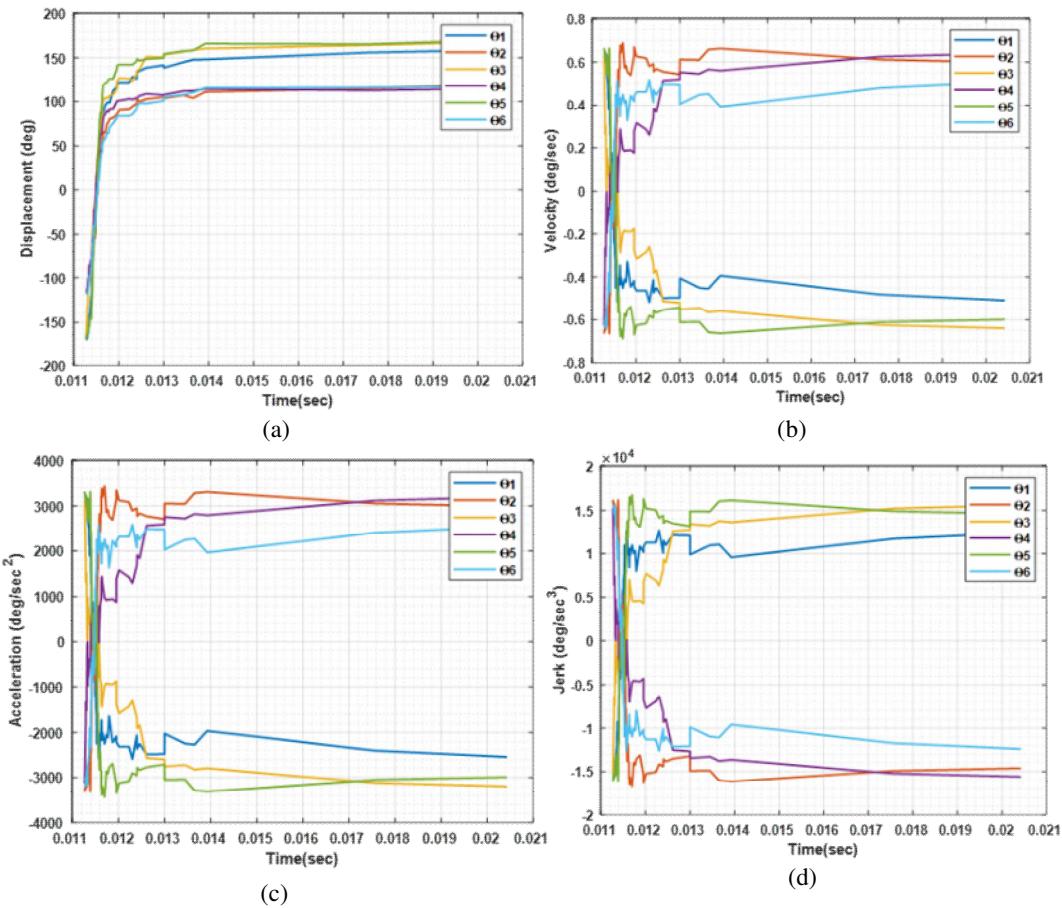


Figure 18: (a) displacement (b) velocity (C) acceleration and (d) jerk of optimized trajectories of KUKA LBR iiwa 14 R820 robot without obstacles

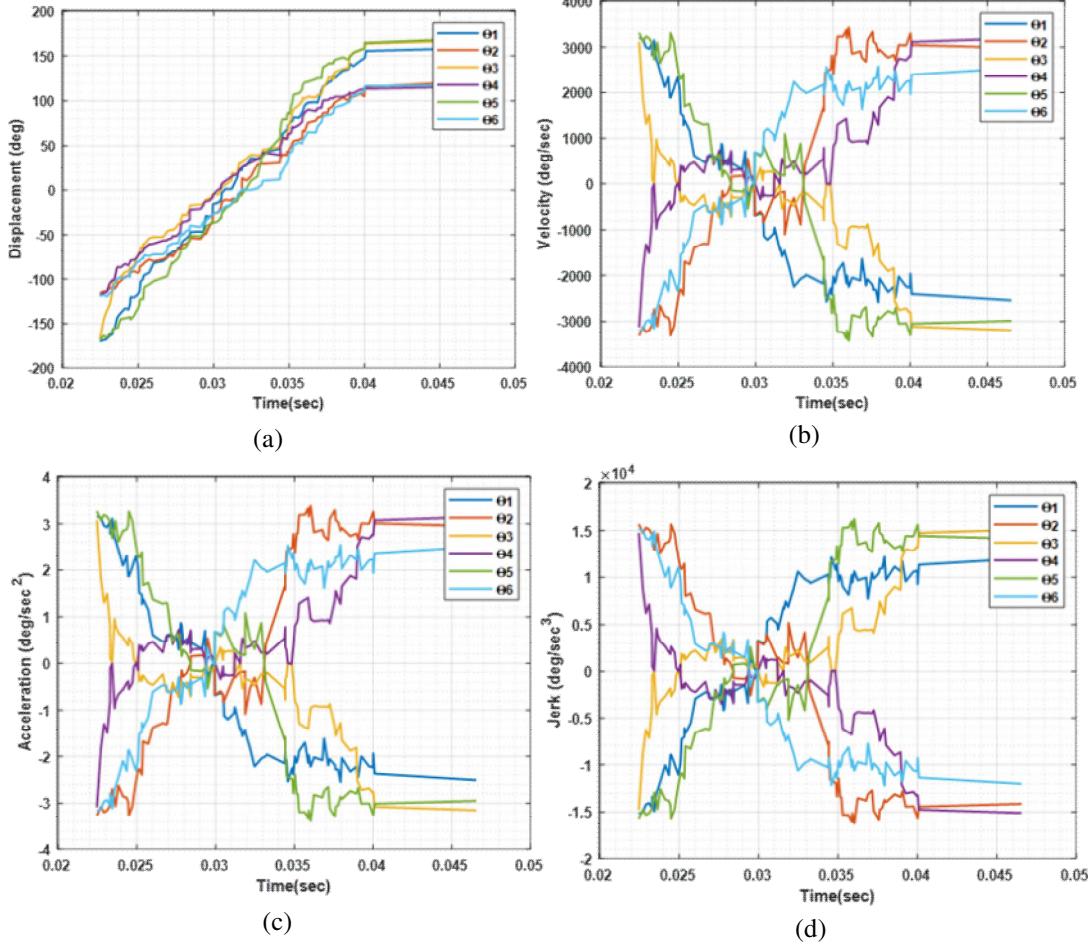


Figure 19: (a) displacement (b) velocity (C) acceleration and (d) jerk of optimized trajectories of KUKA LBR iiwa 14 R820 robot with obstacles

Figure 17 shows the traveling time graph for KUKA robot with and without obstacles. From the source to destination, the robot travels with minimum travelling time in the direction of target point while avoiding the obstacles. The traveling time performance is computed for the different iteration. Then for with and without obstacles, the graphs of displacement, velocity, acceleration and jerk of the optimal trajectories of KUKA robot is evaluated in the figure 18 and 19. For the different joint angles, the movement of robot manipulator is evaluated with respect to the displacement, velocity, acceleration and jerk.

7.3 Simulation model for ABB IRB 140

This section given the simulation analysis of ABB IRB 140 robot. Here, the forward and the inverse kinematics and the trajectory planning problems are solved using the hybrid optimization algorithm for the ABB IRB 140 robot. The hybrid approaches of SBO-WCO, COA-CFA and SOA-TSA are utilized to solve both forward and inverse kinematic model and trajectory planning problem. In SBO, step size (a) is 0.94, difference among upper and lower limit is 0.02, mutation probability is 0.05, and the number of population is 50. In WCO, the number of continents is 5 and the number of team is 20. In COA, number of population is 20. In CFA, the size of population is

60. In SOA, the control parameter (A) is $[2, 0]$ and f_c is 2. In TSA, search agents is 80, parameter P_{\min} is 1 and P_{\max} is 4. For each algorithm, the maximum iteration is 100.

For a given joint angles, the end effector position is calculated in the forward kinematics. The input of the joint angles are obtained from DH convention of ABB IRB 140 as shown in table 3. With the help of the SBO-WCO, the optimized position of end effector is achieved with the minimum fitness value. The optimized position value of end effector is considered as the input value to the inverse kinematics. Finally, the optimized joint angles are evaluated using COA-CFA with minimum joint angle error. ABB IRB 140 (6 DOF) industrial robot is designed by robotics toolbox is given in figure 20.

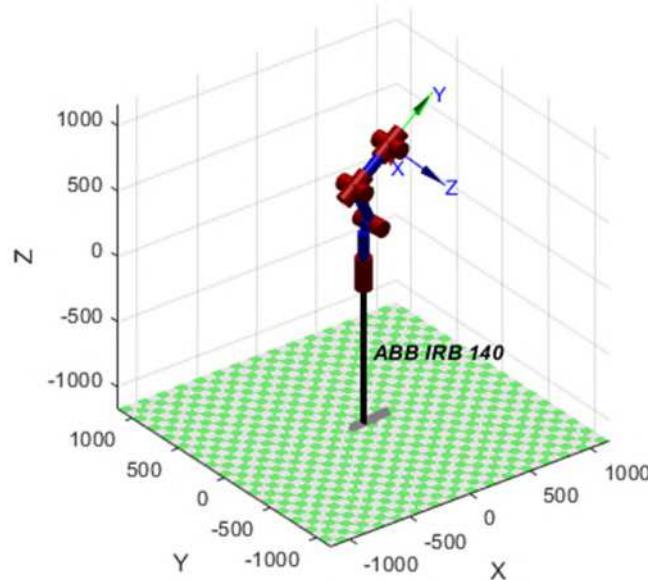


Figure 20: ABB IRB 140 structure

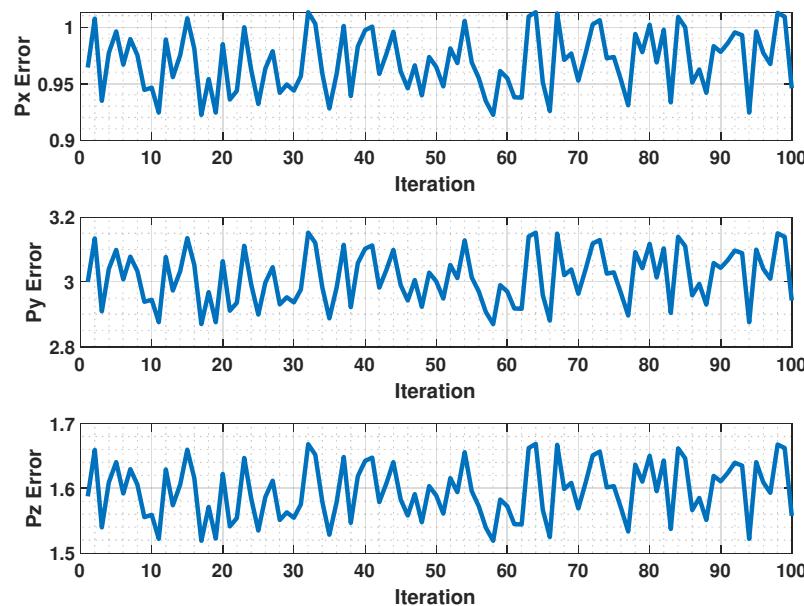


Figure 21: Position error value for the forward kinematics problem

Figure 21 shows the fitness for the forward kinematic problem. The fitness function is to compute the minimum error position value which depends on the Euclidean distance among the desired and the current points. In this work, the forward kinematic problem is converted as the optimization issue which is resolved with the help of the hybrid algorithms of SBO-WCO. For a particular joint angles, the optimized position values are determined with the minimum position error value in the forward kinematics. The fitness function is reduced in the forward kinematics. For each algorithm, the maximum iteration is taken as 100. For each iteration, the different position of end effector is calculated based on the optimization scheme. The current position value reach the target points with the minimum error value. ABB IRB 140 is a 6 DOF robot. So the input is 6 joint angles. Finally, the optimized position values are obtained in x, y and z axis with minimum fitness value.

Figure 22 shows the position error of inverse kinematic problem. The problem of inverse kinematic is converted as the optimization problem in this work which is solved by COA-CFA. For a given position values, the different joint angles are computed in each iteration. The maximum iteration of each algorithm is 100. Fitness function of the inverse kinematic is Euclidean distance among the predicted and target location which means the minimum joint angles. This minimum value provide the best position error value. Finally, the optimal joint angles are calculated with the minimum position error values. ABB IRB 140 robot contains 6 joint angles. In the inverse kinematics, the 6 optimized joint angles are achieved with minimum fitness value using the COA-CFA approach for the specified position value of the end effector and orientation.

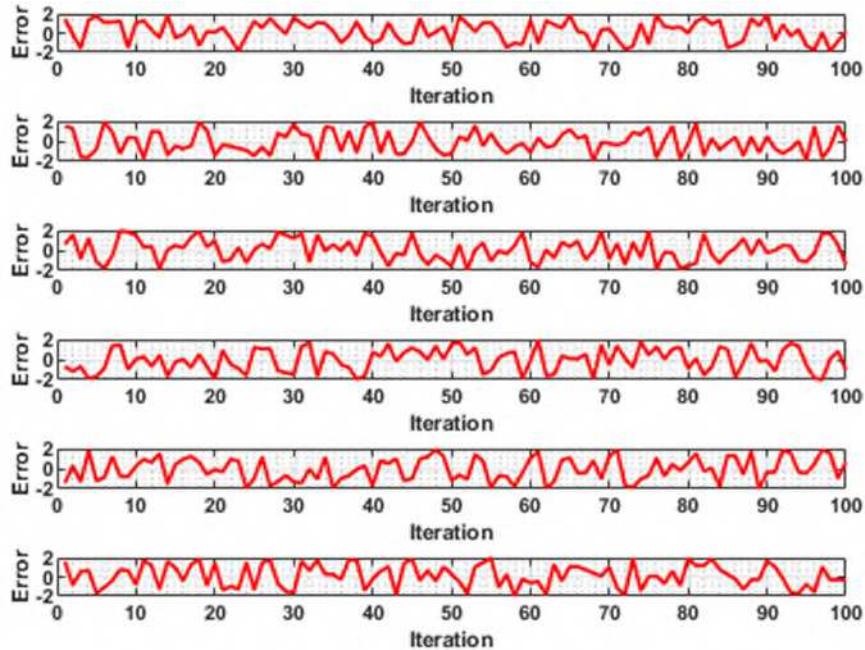


Figure 22: Position error value for the inverse kinematics problem

Figure 23 and 24 displays the trajectory planning of ABB IRB 140 robot. Here, the trajectory planning of ABB IRB 140 robot is solved by the SOA-TSA approach. Figure 23 (a) shows the trajectory generation of ABB IRB 140 robot without considering any obstacles. The source and destination point is created in the robotic workspace. In the workspace, the 100 paths are formed from the starting point to ending point without any obstacles. The least amount of paths are only reached the destination point. From this path, the best optimal path is discovered using the SOA-

TSA approach without any obstacles as shown in figure 23 (b). This optimal path satisfies the minimum fitness value. The fitness function of the trajectory planning of ABB IRB 140 robot is travelling time, jerk, acceleration, and collision free emotion. The overall fitness function is reduced in the trajectory planning. Likewise, in the figure 24 (a) shows the path generation with obstacles for the ABB IRB 140 robot. The source and destination point are created in the workspace. Then different obstacles are located in the workspace. For the given position and orientation of end effector, and joint angles, the 100 paths are created in the workspace with obstacles for each iteration. The few amount of paths are only reached the destination point. From this, the best optimal path is estimated using the SOA-TSA approach with obstacles as shown in figure 24 (b). This optimal path with obstacles satisfies the minimum fitness value. Finally, the robot travels in the optimal path in the direction of target point.

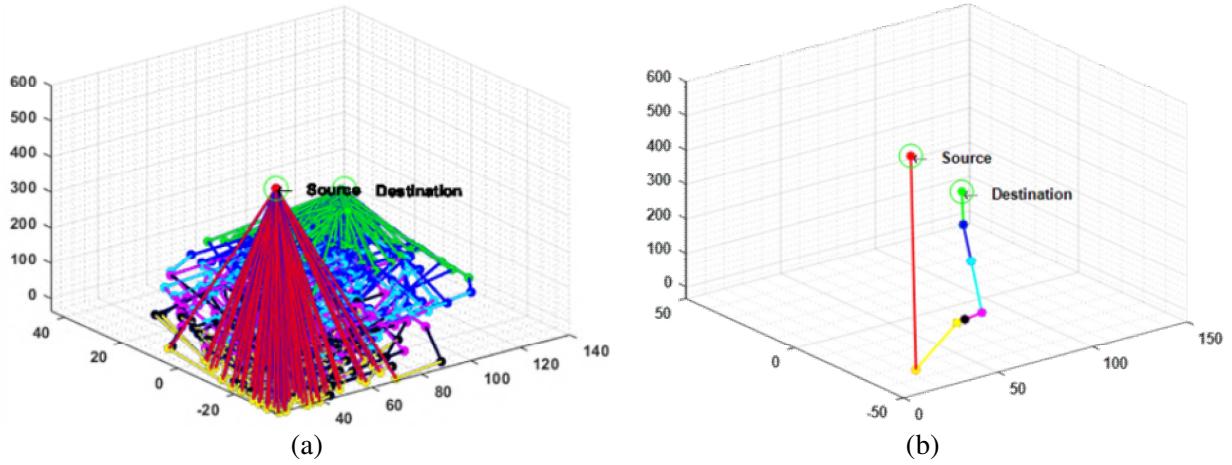


Figure 23: Trajectory planning of ABB IRB 140 (a) Different paths without obstacles (b) optimized path without obstacles using optimization algorithm

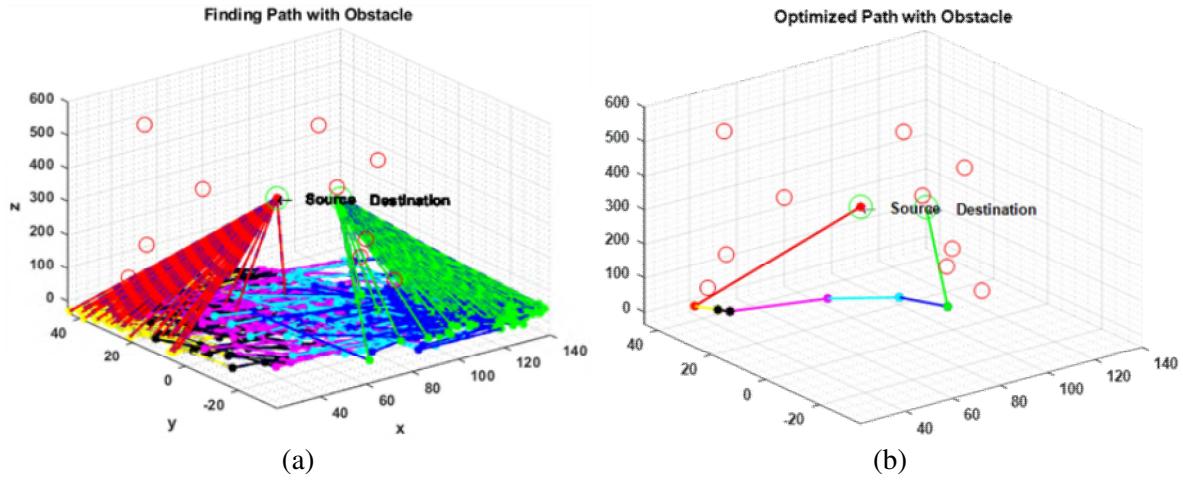


Figure 24: Trajectory planning of ABB IRB 140 (a) Different paths with obstacles (b) optimized path with obstacles using optimization algorithm

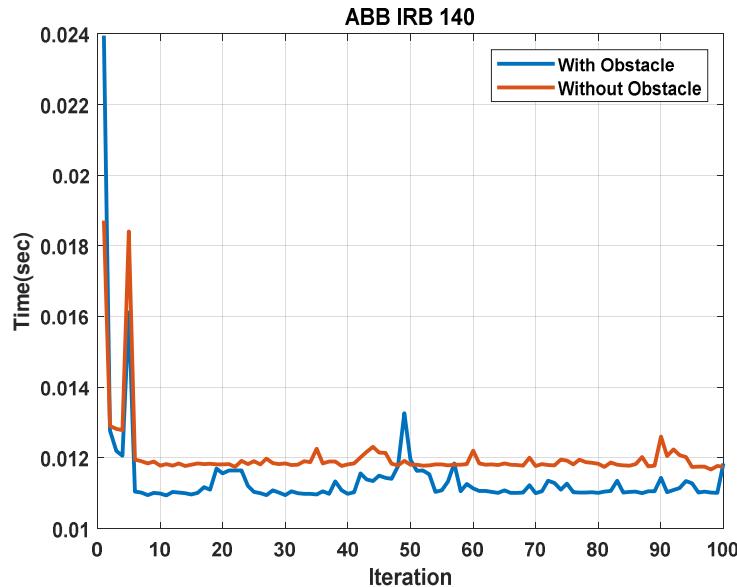


Figure 25: Traveling time performance

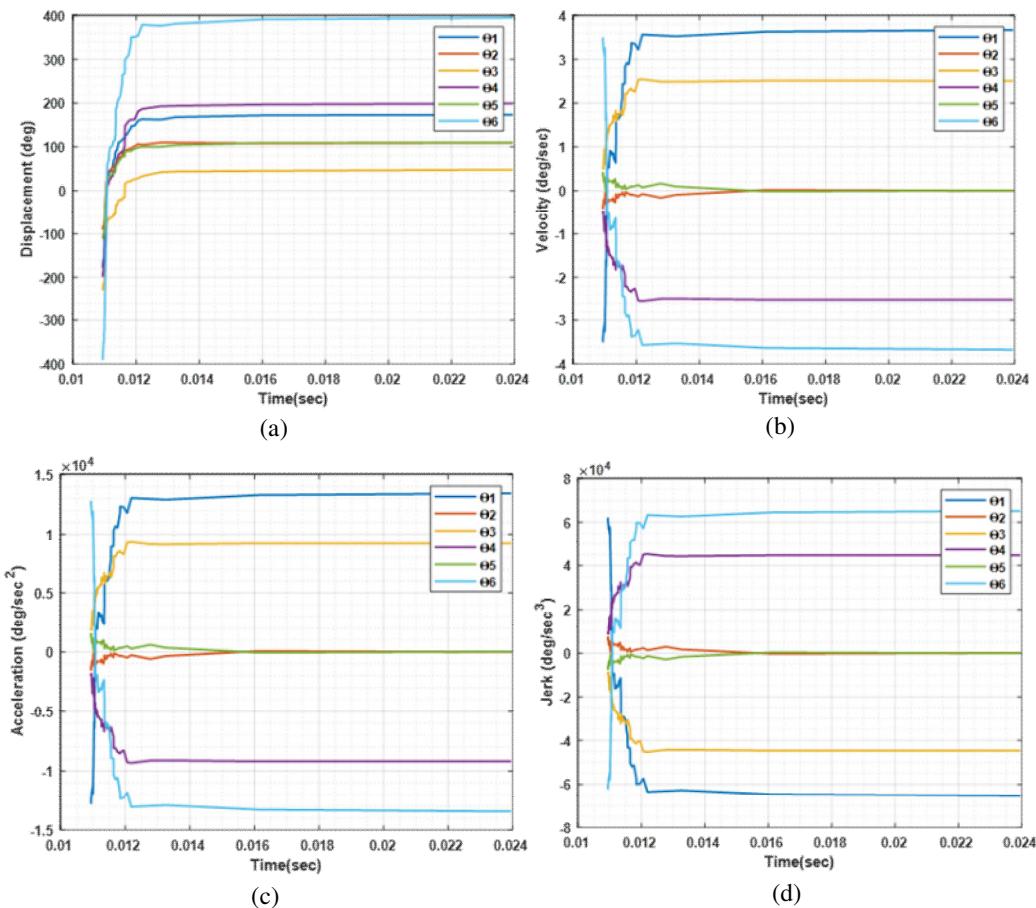


Figure 26: (a) displacement (b) velocity (C) acceleration and (d) jerk of optimized trajectories of ABB IRB 140 robot without obstacles

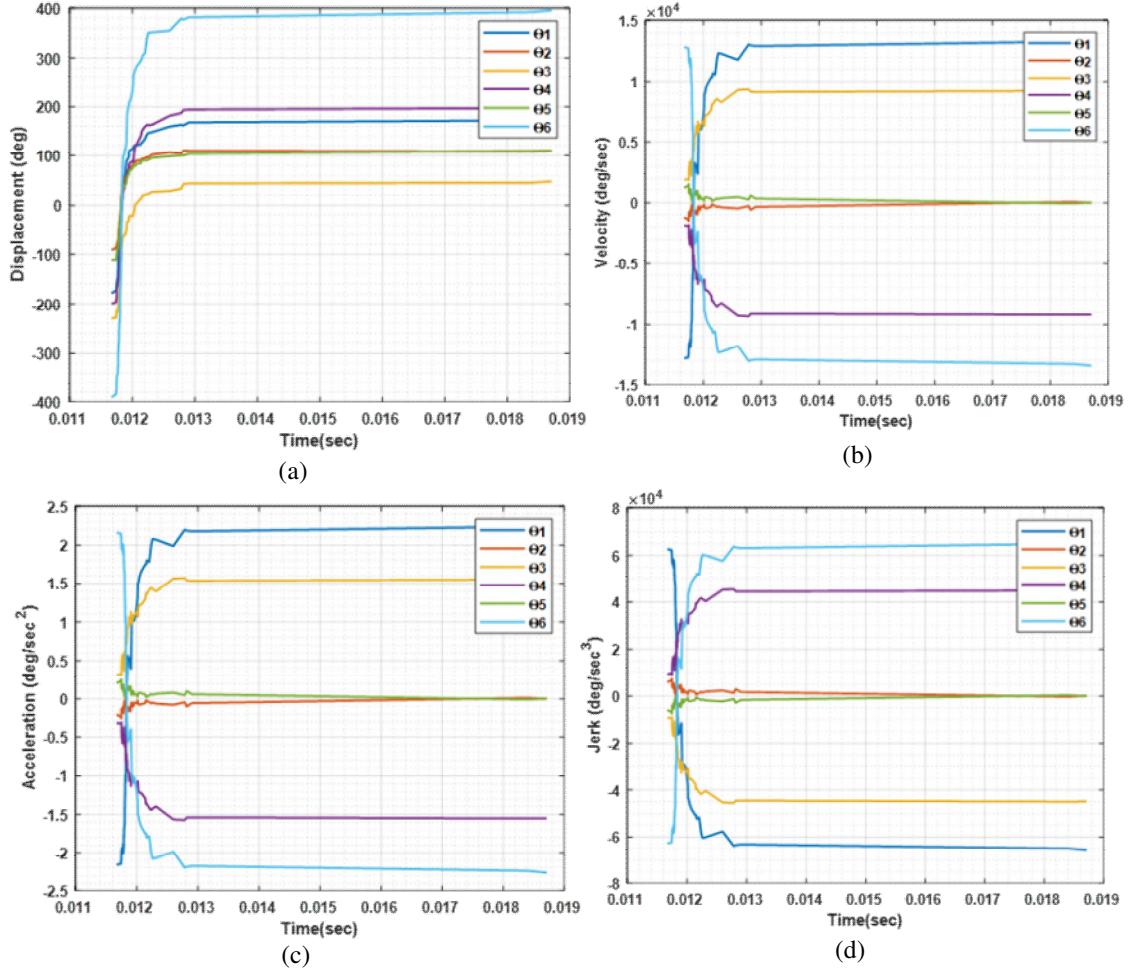


Figure 27: (a) displacement (b) velocity (C) acceleration and (d) jerk of optimized trajectories of ABB IRB 140 robot with obstacles

The travelling time performance of ABB IRB 140 robot is shown in figure 25. Travelling time is one of the objective function for the trajectory planning in this work. The fitness function reduced the travelling time performance for the planning of trajectory. For the different number of iteration, the travelling time performance is computed with and without obstacles. In the planning of trajectory, the optimal best path is identified with minimum travelling time. Therefore, it is most significant factor in the planning of trajectory. The ABB IRB 140 robot travels in the optimal path towards the destination point with minimum travelling time while avoiding the obstacles. Then the robot movement is analysed with respect to the displacement, velocity, acceleration and jerk. For each joints, the graph of displacement, velocity, acceleration and jerk of the optimal trajectories of ABB IRB 140 robot without considering any obstacles are shown in the figure 26. Similarly, in the figure 27, the performance of displacement, velocity, acceleration and jerk of the optimum trajectories of ABB IRB 140 robot with obstacles are analysed.

Table 4: Optimal position values for each robot

| Without obstacles | | | | |
|-------------------|--------|-------|-------|-----------|
| Robots | Px | Py | Pz | iteration |
| PUMA | 80.02 | 65.36 | 55.08 | 65 |
| KUKA | 40.005 | 45.36 | 31.23 | 98 |
| ABB IRB 140 | 140.15 | 45.02 | 85.11 | 11 |
| With obstacles | | | | |
| Robots | Px | Py | Pz | iteration |
| PUMA | 80.02 | 65.36 | 55.08 | 65 |
| KUKA | 40.14 | 45.03 | 31.04 | 8 |
| ABB IRB 140 | 140.04 | 45.11 | 85.07 | 98 |

Table 5: Comparison of proposed with existing robotic works

| Authors | Robots | Methods | Time (s) | Average velocity | Average Acceleration | Average Jerk |
|-----------------------|--------------------------------|----------------|----------|------------------|----------------------|--------------|
| Proposed | PUMA 560 (without obstacle) | CSA-BSA | 0.0118 | 1.62E-17 | 1.89E-14 | -3.03E-13 |
| | PUMA 560 (with obstacle) | CSA-BSA | 0.0313 | -9.25E-17 | 0 | -1.52E-13 |
| | KUKA (without obstacle) | GWO-GOA | 0.0117 | -2.31E-18 | 0 | 3.79E-14 |
| | KUKA (with obstacle) | GWO-GOA | 0.0310 | 9.25E-18 | -9.47E-15 | 3.79E-14 |
| | ABB IRB 140 (without obstacle) | ABB IRB 140 | 0.0114 | 9.25E-18 | 0 | 0 |
| | ABB IRB 140 (with obstacle) | ABB IRB 140 | 0.0120 | 0 | 3.79E-14 | 0 |
| Amruta et al. [16] | Kawasaki RSO6L | Improved MOALO | 28.21 | NA | NA | 55.1 |
| Zhifeng et al. [17] | ER3A-C60 | PSO | 12.445 | 3.85 | 3.74 | 6.8 |
| Nguyen and Huang [18] | PUMA 560 | SQP | 9.097 | 41.9 | 44.63 | 57.63 |

Optimal position values for each robot is shown in table 4. Table 5 provide the comparison table of the proposed industrial robotic works and the existing robotic works. Here, the total travelling time, jerk, acceleration, and velocity are analysed for the proposed works and the existing robotic works. Velocity, acceleration and jerk parameters are reduce the total traveling time in the proposed robotic works. For with and without obstacles, these parameters are evaluated in the proposed works. The zero value indicates that at the initial end target motion time can eliminate the vibrations when the robot starts or stops. For each robotics, the different hybrid optimization approaches are utilized for the trajectory planning. From this analysis, the proposed industrial robotic manipulators are provide the best optimal results for the movement of robots.

8. Conclusion

In this paper, the kinematic model and the trajectory planning problem of an industrial robotic manipulators based on the hybrid optimization algorithms has been proposed. Totally, 18 meta-heuristic algorithms such as PSO, SSO, DFO, BOA, CSA, BSA, SHO, KHO, ALO, ACO, GWO, GOA, SBO, WCO, COA, CFA, SOA and TSA have been used in various field of industrial robotics to solve both forward and inverse kinematic problem and the trajectory planning problem for any configuration of IRMs. In this work, three IRMs manipulators such as PUMA 560, KUKA LBR iiwa 14 R820 and the ABB IRB 140 are considered for the analysis of kinematics and trajectory problems. The hybrid approaches of PSO-SSO, SHO-KHO and SBO-WCO are used to solve the forward kinematics of PUMA 560, KUKA LBR iiwa 14 R820 and the ABB IRB 140. Also the DFO-BOA, ALO-ACO and COA-CFA are used for the inverse kinematics of each robot. In addition, trajectory issues are solved by the CSA-BSA, GWO-GOA and SOA-TSA approaches for each IRM. The main objective of this work is to minimize the position error for both forward and inverse kinematics of each IRM. The multi-objective functions are considered for the trajectory planning of each robot. Finally, the optimized position of the end effector and the joint angles are obtained for each IRM with minimum position error values. Also the best optimal path is achieved with and without obstacles for each IRM with considering the minimum fitness function value. These adopted algorithms has shown the potential of getting faster convergence and yielding global optimum solution for the stated problem. The entire procedure is simulated by robotic tool box in MATLAB with GUI interface. For each IRM, the simulation performances are evaluated it compared to the existing robotic manipulator works. The performances of our proposed work is very fast and efficient scheme than other approaches. In future the tuning of various parameters of optimization algorithms can be considered so as to avoid trapping in local minimum point. Even the deep learning approaches may be proposed for the kinematics and trajectory problems.

Compliance with Ethical Standards

Funding: No funding is provided for the preparation of manuscript.

Conflict of Interest: Authors ^{*}D. Gurjeet Singh & ²Vijay Kumar Banga declares that they have no conflict of interest.

Ethical Approval: This article does not contain any studies with human participants or animals performed by any of the authors.

References

- [1] Raza K, Khan TA and Abbas N (2018) Kinematic analysis and geometrical improvement of an industrial robotic arm. Journal of King Saud University-Engineering Sciences 30(3): 218-223.
- [2] Misra A and Singh G (2019) Kinematic and Dynamic Analysis of an Industrial Six-Axis Robotic Manipulator. In International Conference on Robotics, Automation and Non-Destructive evaluation.
- [3] Vo AT and Kang H-J (2018) An adaptive neural non-singular fast-terminal sliding-mode control for industrial robotic manipulators. Applied Sciences 8(12): 2562.

- [4] Soria PR, Arrue BC and Ollero A (2019) A 3d-printable docking system for aerial robots: Controlling aerial robotic manipulators in outdoor industrial applications. *IEEE Robotics & Automation Magazine* 26(1): 44-53.
- [5] Alebooyeh M and Urbanic RJ (2019) Neural network model for identifying workspace, forward and inverse kinematics of the 7-DOF YuMi 14000 ABB collaborative robot. *IFAC-PapersOnLine* 52(10): 176-181.
- [6] Hwang YL, Cheng JK and Truong VT (2018) Dynamic analysis and control of industrial robotic manipulators. In *Applied mechanics and materials*, Trans Tech Publications Ltd, 883: 30-36.
- [7] Dereli S and Köker R (2018) IW-PSO approach to the inverse kinematics problem solution of a 7-DOF serial robot manipulator. *Sigma Journal of Engineering and Natural Sciences* 36(1): 77-85.
- [8] Huang J, Hu P, Wu K and Zeng M (2018) Optimal time-jerk trajectory planning for industrial robots. *Mechanism and Machine Theory* 121: 530-544.
- [9] Kim J and Croft EA (2019) Online near time-optimal trajectory planning for industrial robots. *Robotics and Computer-Integrated Manufacturing* 58: 158-171.
- [10] Fang Y, Hu J, Liu W, Shao Q, Qi J and Peng Y (2019) Smooth and time-optimal S-curve trajectory planning for automated robots and machines. *Mechanism and Machine Theory* 137: 127-153.
- [11] Wang H, Wang H, Huang J, Zhao B and Quan L (2019) Smooth point-to-point trajectory planning for industrial robots with kinematical constraints based on high-order polynomial curve. *Mechanism and Machine Theory* 139: 284-293.
- [12] Reiter A, Müller A and Gatringer H (2018) On higher order inverse kinematics methods in time-optimal trajectory planning for kinematically redundant manipulators. *IEEE Transactions on Industrial Informatics* 14(4): 1681-1690.
- [13] Liu Y, Guo C and Weng Y (2019) Online time-optimal trajectory planning for robotic manipulators using adaptive elite genetic algorithm with singularity avoidance. *IEEE Access* 7: 146301-146308.
- [14] Gallant A and Gosselin C (2018) Extending the capabilities of robotic manipulators using trajectory optimization. *Mechanism and Machine Theory* 121: 502-514.
- [15] Kobayashi M and Motoi N (2020) Hybrid Control of Trajectory Planning for Desired Trajectory and Collision Avoidance Based on Optimization Problem for a Wheeled Mobile Robot. *IEEJ Journal of Industry Applications* 9(4): 331-340.
- [16] Rout A, Mahanta GB, Bbvl D and Biswal BB (2020) Kinematic and Dynamic Optimal Trajectory Planning of Industrial Robot Using Improved Multi-objective Ant Lion Optimizer. *Journal of the Institution of Engineers (India): Series C* 1-11.
- [17] Liu Z, Xu J, Cheng Q, Zhao Y, Pei Y and Yang C (2018) Trajectory planning with minimum synthesis error for industrial robots using screw theory. *International Journal of Precision Engineering and Manufacturing* 19(2): 183-193.
- [18] Nguyen M-T and Huang J-H (2019) Smooth and Time Optimal Trajectory Planning for Industrial Robot Using a Single Polynomial. In *International Conference on Engineering Research and Applications*, Springer, Cham, 657-663.
- [19] Kumar A, Banga VK, Kumar D and Yingthawornsuk T (2019) Kinematics Solution using Metaheuristic Algorithms. In *2019 15th International Conference on Signal-Image Technology & Internet-Based Systems (SITIS)*, IEEE, 505-510.

- [20] Jaladi S, Rao TE and Srinath A (2020) Inverse Kinematics Analysis of Serial Manipulators Using Genetic Algorithms. In *Soft Computing for Problem Solving*, Springer, Singapore, 519-529.
- [21] Singh TP, Suresh P and Chandan S (2017) Forward and inverse kinematic analysis of robotic manipulators. *International Research Journal of Engineering and Technology (IRJET)* 4(2): 1459-1468.
- [22] Kuhlemann I, Jauer P, Ernst F and Schweikard (2016) A Robots with seven degrees of freedom: Is the additional DoF worth it? In *2016 2nd International Conference on Control, Automation and Robotics (ICCAR)*, IEEE, 80-84.
- [23] Almaged M (2017) Forward and inverse kinematic analysis and validation of the ABB IRB 140 industrial robot. *International Journal of Electronics Mechanical and Mechatronics Engineering* 7(2): 1383-1401.
- [24] Wei C, Ji Z and Cai B (2020) Particle Swarm Optimization for Cooperative Multi-Robot Task Allocation: A Multi-Objective Approach. *IEEE Robotics and Automation Letters* 5(2): 2530-2537.
- [25] James JQ and Li VO (2015) A social spider algorithm for global optimization. *Applied Soft Computing* 30: 614-627.
- [26] Mirjalili S (2016) Dragonfly algorithm: a new meta-heuristic optimization technique for solving single-objective, discrete, and multi-objective problems. *Neural Computing and Applications* 27(4): 1053-1073.
- [27] Arora S and Singh S (2019) Butterfly optimization algorithm: a novel approach for global optimization. *Soft Computing* 23(3): 715-734.
- [28] Cuevas E, Fausto F and González A (2020) The Selfish Herd Optimizer. In *New Advancements in Swarm Algorithms: Operators and Applications*, Springer, Cham, 69-109.
- [29] Gandomi AH and Alavi AH (2012) Krill herd: a new bio-inspired optimization algorithm. *Communications in nonlinear science and numerical simulation* 17(12): 4831-4845.
- [30] Mirjalili S (2015) The ant lion optimizer. *Advances in engineering software* 83: 80-98.
- [31] Mirjalili S (2019) Ant colony optimisation. In *Evolutionary Algorithms and Neural Networks*, Springer, Cham, 33-42.
- [32] Moosavi SHS and Bardsiri VK (2017) Satin bowerbird optimizer: A new optimization algorithm to optimize ANFIS for software development effort estimation. *Engineering Applications of Artificial Intelligence* 60: 1-15.
- [33] Razmjoo N, Khalilpour M and Ramezani M (2016) A new meta-heuristic optimization algorithm inspired by FIFA world cup competitions: theory and its application in PID designing for AVR system. *Journal of Control, Automation and Electrical Systems* 27(4): 419-440.
- [34] Rajabioun R (2011) Cuckoo optimization algorithm. *Applied soft computing* 11(8): 5508-5518.
- [35] Eesa AS, Brifcani AMA and Orman Z (2013) Cuttlefish algorithm-a novel bio-inspired optimization algorithm. *International Journal of Scientific & Engineering Research* 4(9): 1978-1986.
- [36] Askarzadeh A (2016) A novel metaheuristic method for solving constrained engineering optimization problems: crow search algorithm. *Computers & Structures* 169: 1-12.
- [37] Meng X-B, Gao XZ, Lu L, Liu Y and Zhang H (2016) A new bio-inspired optimisation algorithm: Bird Swarm Algorithm. *Journal of Experimental & Theoretical Artificial Intelligence* 28(4): 673-687.

- [38] Mirjalili S, Mirjalili SM and Lewis A (2014) Grey wolf optimizer. *Advances in engineering software* 69: 46-61.
- [39] Saremi S, Mirjalili S and Lewis A (2017) Grasshopper optimisation algorithm: theory and application. *Advances in Engineering Software* 105: 30-47.
- [40] Dhiman G and Kumar V (2019) Seagull optimization algorithm: Theory and its applications for large-scale industrial engineering problems. *Knowledge-Based Systems* 165: 169-196.
- [41] Kaur S, Awasthi LK, Sangal AL and Dhiman G (2020) Tunicate Swarm Algorithm: A new bio-inspired based metaheuristic paradigm for global optimization. *Engineering Applications of Artificial Intelligence* 90: 103541.