

Optimal design of tricept parallel manipulator with particle swarm optimization using performance parameters

Syed Saad Farooq*, Aamer Ahmed Baqai and Muhammad Faizan Shah*

**Mechanical Engineering, Khwaja Fareed University of Engineering and Information Technology Rahim Yar Khan
Mechanical Engineering, National University of Sciences and Technology Rawalpindi, 46000, Pakistan*

**Corresponding Author: syedsaad34@gmail.com*

Submitted: 20/11/2019

Revised: 04/05/2020

Accepted: 14/07/2020

ABSTRACT

The parallel manipulators are skilled for their precision manufacturing but need optimized design to get maximum dexterity that will lead towards better industrial production rates. The 3-DOF tricept is chosen to utilize its maximum capabilities for its functionality. Three performance parameters conditioning index, workspace volume, and global conditioning index are used to obtain optimum design variables of tricept mechanism. With a view to compare them in terms of processing effort, particle swarm optimization (PSO) is applied here. Finally, multiobjective optimization with two strategies weighted and epsilon constraint is performed to control the different parameters simultaneously and also to give validation of previously obtained GA based optimum design values of tricept mechanism.

Keywords: Inverse Kinematics; Dexterity; Evolutionary Algorithms; Multi Objective Particle Swarm Optimization (MOPSO); Epsilon Constraint Strategy; MATLAB.

1. INTRODUCTION

Manipulators are expressed as subpart of robot (Spong, 1995), which are controlled by the motors and drives. They have programming based numerical control (He et al., 2007). Parallel manipulators are well known for their rapid acceleration and immediate precise movements (Shah, Kausar, and Farooq, 2018). They are more flexible than serial manipulators and also have the feature of bearing high loads and provide high stiffness (Zhang, 2010) (Y D Patel and George, 2012). On the other hand, end effector, usually known as platform in parallel manipulators, is limited to a certain workspace resulting in limitation to work in far-off places. These kinds have complex inputs and outputs solutions. It is difficult to find high number of singularities in parallel manipulators, although these singularities are static (YJ Lou, Liu, and Li, 2005). These closed loop parallel manipulators possess less inertial effects.

Classification of parallel manipulators includes symmetric, planar, spherical, and spatial (Y D Patel and George, 2012). An example like Gough Stewart Platform is a 6-DOF basic architecture and has spherical prismatic spherical architecture (SPS) explained by Gupta et al. (Mehta and Dasgupta, 2011). 3-DOF revolute (RRR) architecture has been well explained by Gosselin and Guillot (1991) and YJ Lou et al. (2005). This architecture has all joints revolute and does not possess translations. Orthoglide mechanism has been illustrated by Chablat and Wenger (2003). This structure moves in the x, y, z directions having fixed orientation and is heavily used for the machining purposes. Tricept manipulators, which are the center of discussion in this work, have three legs with prismatic actuated design and a center leg that has a UPS architecture, which is connected from base to the moving platform above. They have

prismatic actuators. These types of structures have been well explained by Wang et al. (2003) and Hosseini, Daniali, and Taghirad (2011).

1.1 Performance Parameters

Performance or dexterity can be evaluated in terms of ease of mobility and reachability (Shao et al., 2017). Dexterity is actually the measure of sensitivity between the end effector and the actuator movement (YJ Lou et al., 2005) (Pond and Carretero, 2008). Conditioning index provides the sensitivity ratio for dexterity (Rezania and Ebrahimi, 2017). The conditioning index can also use the singular values of Jacobian matrix, so it better explains the singularities and links nearness to singularities. Furthermore, it also explains the error in the design and stiffness (YJ Lou et al., 2005).

Global Conditioning Index is based on the requirement whether the user needs the local conditioning or the global conditioning. If the user wants the results to be with respect to global conditioning, they should use the global conditioning index for its simulation results (Clement; Gosselin, 1988).

The workspace of a parallel manipulator is another useful parameter in determining its trajectories. The workspace is defined as the volume of the region end effectors that can occupy all throughout its maximum reach (Arrouk, Bouzgarrou, and Gogu, 2010). Reachable workspace is the volume of space in which the end effector can reach all its points through at least one orientation, whereas the most important term of the dexterous workspace is the volume of the space in which the end effector can reach its all points from all possible orientations. Basically, the dexterous workspace is the subset of the reachable workspace. There are two main types of determining the workspace. The first one is finding the design of the manipulators with the prescribed workspace (Merlet, 1997) (Gosselin and Guillot, 1991), and the second is to maximize the workspace by changing the geometry of a workspace (YJ Lou et al., 2005); both will be discussed here.

1.2 Optimization

Optimization is being used extensively in automation. To achieve user demands, the manufacturer needs a design that will enhance its production rate. Smaller changes in design can bring significant results. In short, the selection of design must ensure its maximum capabilities. For that purpose, different approaches are being used for the optimization. Evolutionary algorithm (EA) will be the prime focus to get optimized design variables. Conventional single objective deals with the optimization of parameters independently (Mei et al., 2018). Here are some points for better understanding of how to use the evolutionary algorithms for this task (Carretero et al., 2000) (Deb n.d.). EAs are used here because of the uncertainty in the solutions. Secondly, multiple design variables are involved. Thirdly, there are complex constraints for calculations, and there are more numbers of local and global optimum points, so the use of evolutionary algorithms is inevitable here.

2. LITERATURE

Parallel manipulators caught attention in the mid-nineties. James E Gwinnett made a spherical parallel robot platform, and this was indeed very interesting invention for the entertainment industry. In 1947, Dr. Eric Gough presented his research on parallel robots, and it was functional in 1954, which was variable six-strut octahedral hexapod. It paved the way for the scientists to have that strong existing base for the parallel robots. The universal tire testing machine was built by Dr. Eric Gough (Gough 1962), and it has the property of inspecting the tire characteristics under combined loading effects. Multiaxis simulation table was of that kind, and it was built by Dr. Hubert. Later Stewart in 1965 made a flight simulator model that was likely to be a octahedral hexapod. Meanwhile, Klaus Cappel made a motion simulator. It was actually the same octahedral hexapod.

Tsai made a valuable contribution in describing the difference between the serial and parallel manipulators (Tsai, 1999). Designing parallel robots depends entirely on the optimum performance parameters. Clément Gosselin (1988) presented the general idea for the calculation of conditioning and global indexes. They had also used searching technique for the design optimization. Some used sequential quadratic and some had used the genetic algorithms for the tasks. In the mid-nineties, Richard et al. optimized the workspace based on dexterity and GCI for 3-DOF

translational platform (Stamper, Lung-Wen Tsai, and Walsh n.d.). In the start of 20th century, hybrid serial and parallel manipulator optimization was explained by Tanev (2000). After that, Siciliano (1999) proposed the tricept robot kinematics and did its workspace optimization using different parameters. later, Hui Cheng worked on the dynamics and control of parallel manipulators (Cheng, 2001). YJ Lou et al. (2005) compared different manipulators and their optimization based on dexterity. In 2007, Carretero et al. expressed the quantitative analysis of manipulators based on the dexterity and dexterous workspace. On the basis of dexterity, Huang (2011) designed the planar manipulator. Binbin et al. (2011) tried to develop 3-DOF UPU kinematic structure and worked on the optimization and singularities. In 2012, Zhang and Fang (2012) proposed a 3-DOF PRS optimization based on interval analysis.

Recent advancements include multiobjective optimization for the special applications of industry, as well as medical sector. A study compared the different optimization methods for Process Parameters of Machining (Yusup, Zain, and Hashim, 2012). In 2015, multiobjective optimization of a parallel ankle rehabilitation robot has been done using modified differential evolution algorithm (Wang, Fang, and Guo, 2015). Later workspace optimization for parallel has been done for drilling operation using genetic algorithm (Sudheer and Kumar, 2017). In multiobjective optimization, parallel manipulator is used to design a prosthetic arm using evolutionary algorithm (GA) (Leal-Naranjo et al., 2018). Industrial Advances in Machining Parameters particle swarm Optimization for materials were done using Response Surface Methodology (Lmalghan et al., 2018). Further, multiobjective optimization is used for 6-DOF Parallel Manipulator for setting total Workspace (Qiang et al., 2019). Recently, parametric performances of parallel structures with 3 or higher degrees of freedom are being controlled to get maximum dexterity (Xie et al., 2019).

Tricept parallel structures are also helpful where extreme attention is required for reactive material handling. It is used for sensitive medical equipment like prosthesis and cardio pulmonary resuscitation operation (CPR) (Y. D. Patel and George, 2012) and others. Now, this paper will progress with problem formulation and the methodology adopted for the work and will conclude with results and discussions.

3. TRICEPT

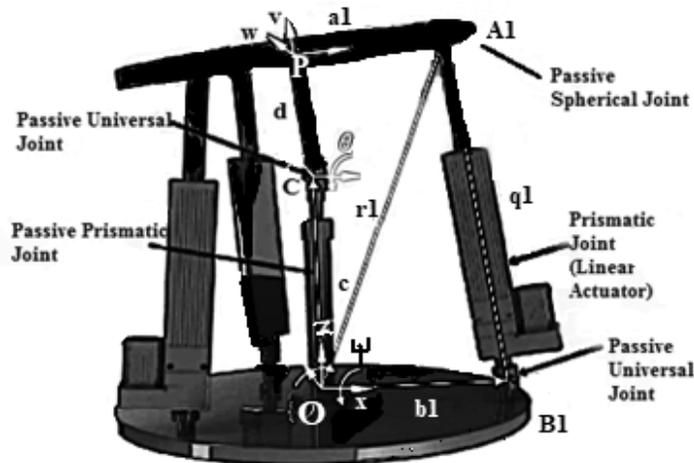


Figure 1. Tricept mechanism.

This mechanism has 3 DOF, and the combination of joints includes two rotations and one translation (Hosseini, 2011). The actuated joint is prismatic, and it has SPS configuration, but later, one spherical has been replaced with the Universal joint, so it becomes the UPS structure. The center link connects the base to the moving platform. When the structure is static, the line passing through the universal joint of the moving platform is parallel with the x and y axis of base. When the prismatic joint is activated, other universal and spherical joints are passive with that prismatic joint movement (Hosseini et al., 2011) (Hosseini, 2011). Limitations for its geometry are illustrated in Table 1.

Table 1. Geometric Constraints.

Actuator lengths (mm)	Angle (rad)	d (mm)	b(mm)	a(mm)
400-750	-1 to +1	20-200	300-500	200-300

In Table 1, ‘d’ is the length of the joint from point C to point P, and also ‘b’ is the length of the static platform from point O to B₁ whereas ‘a’ is the length from point P to A₁ of the moving platform.

4. PROBLEM FORMULATION

The limited workspace and dexterity are mainly the issues of the parallel manipulator (Glozman and Shoham, 2009) (Chen et al., 2014) (Furqan, Suhaib, and Ahmad, 2017) (Yunjiang Lou et al., 2005) As a result, it becomes vital to design a set of geometric parameters of the parallel manipulator with ideal workspace and good performance. Also, it is known that workspace volume and conditioning index are inversely proportional to each other. So, we ought to make sure of the optimal performance parameters, which will give optimal design. PSO is famous for its fast tracking of optimum value. Fast tracking will save certain amount of computer processing effort. It will bring here the optimum values of the performance index using PSO. The task of controlling performance parameter simultaneously at the same time, MOPSO, is an additional ingredient to the work.

5. RESEARCH METHODOLOGY

The schematic methodology is starting from left top of figure 2, and inverse kinematic solution is obtained by close loop method for evaluating geometry (Wang, Li, and Zhao, 2010). It will help in finding the performance parameters, namely, workspace volume, conditioning index, and global indexes. Then, single objected particle swarm optimization is executed on their performance indexes. For the maximum individual performance parameter of the manipulator, corresponding design variables are achieved. Then, apply multiperformance parameter (MOPSO) with weighted and epsilon strategies to find the optimum performance points simultaneously. Results will be analyzed and compared, and validation was done by Hosseini et al. (2011). Conclusion will be drawn at the end with future suggestions.

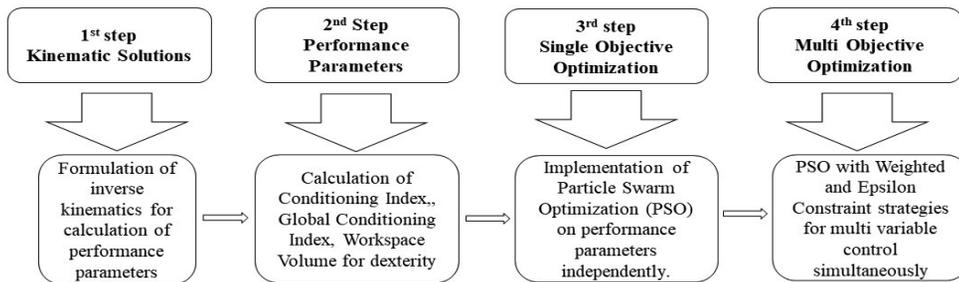


Figure 2. Methodology.

5.1 Kinematic Solutions

In order to find the performance parameters like conditioning index and global index, first, find the inverse kinematics of the whole structure. Here are some steps to calculate the inverse kinematics:

- Formulate position vectors of limbs with respect to frame ‘O’, which is the base frame, i.e., OB₁, OB₂, and OB₃.
- Formulate position vectors of limbs with respect to ‘P’ frame, which is the moving frame, i.e., PA₁, PA₂, and PA₃.
- Consider the center link and make the rotation matrices ‘R’ and translational matrices ‘a’:

$$R_x = \begin{pmatrix} 1 & 0 & 0 \\ 0 & C\Psi & S\Psi \\ 0 & -S\Psi & C\Psi \end{pmatrix}, R_y = \begin{pmatrix} C\theta & 0 & -S\theta \\ 0 & 1 & 0 \\ S\theta & 0 & C\theta \end{pmatrix} \quad (1)$$

$$a_1 = \begin{pmatrix} a/\sqrt{3} \\ 0 \\ d \end{pmatrix}, a_2 = \begin{pmatrix} -a/2\sqrt{3} \\ a/2 \\ d \end{pmatrix}, a_3 = \begin{pmatrix} -a/2\sqrt{3} \\ -a/2 \\ d \end{pmatrix} \quad (2)$$

where the 1st, 2nd, and 3rd rows of a_1 are the position values in x, y and z directions from the 1st leg to the center point of the moving platform, while a_2 is from the 2nd leg and a_3 is from the 3rd leg.

- Then, from the closed loop procedure, the position vector is indicated from base to moving platform.

$$A_i = Q_0^P * PA_i + OP \quad (3)$$

where A_i = Transformation from base point 'O' to the moving 'P', and i ranges from 1 to 3

Q_0^P = Rotation matrix from point of base to moving platform.

OP = Position vector from base to moving platform.

By putting Eq. (1) (2) in (3), Eq. (4) takes the form

$$A = \begin{pmatrix} C\theta & S\Psi S\theta & C\Psi S\theta \\ 0 & C\Psi & -S\Psi \\ -S\theta & C\theta S\Psi & C\theta C\Psi \end{pmatrix} \begin{pmatrix} a/\sqrt{3} & -a/2\sqrt{3} & -a/2\sqrt{3} \\ 0 & a/2 & -a/2 \\ d & d & d \end{pmatrix} + \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ c & c & c \end{pmatrix} \quad (4)$$

Similarly, B_i = Transformation from base point 'O' to the point 'B', and i ranges from 1 to 3:

$$B = \begin{pmatrix} b/\sqrt{3} & -b/2\sqrt{3} & -b/2\sqrt{3} \\ 0 & b/2 & -b/2 \\ 0 & 0 & 0 \end{pmatrix} \quad (5)$$

Then, from the constraint equations, proceed towards the inverse kinematics:

$$\| (A_i - B_i) \| = q_i \quad (6)$$

where i increases from 1 to 3, $\{q_1, q_2, q_3\}$ denote the actuated lengths of joints configuration, and $\{\varphi, \theta, c\}$ are the Cartesian coordinates, where θ denotes the rotation angle along x axis and Ψ denotes the rotation angle along the y axis, whereas c is the translation along z axis. The Jacobian matrix has been formulated.

$$q_1^2 = \frac{a^2}{3} + \frac{b^2}{3} + c^2 + d^2 - \frac{2}{3}abC\theta + 2cdC\theta C\Psi - \frac{2bd}{\sqrt{3}}C\Psi S\theta$$

$$q_2^2 = \frac{a^2}{3} + \frac{b^2}{3} + c^2 + d^2 - \frac{1}{2}ab\left(\frac{1}{3}C\theta - \frac{1}{\sqrt{3}}S\Psi S\theta + C\Psi\right) + bd\left(\frac{C\Psi S\theta}{\sqrt{3}} + S\Psi\right) + 2cdC\theta C\Psi + ac\left(\frac{S\theta}{\sqrt{3}} + C\theta S\Psi\right) \quad (7)$$

$$q_3^2 = \frac{a^2}{3} + \frac{b^2}{3} + c^2 + d^2 - \frac{1}{2}ab\left(\frac{1}{3}C\theta + \frac{1}{\sqrt{3}}S\Psi S\theta + C\Psi\right) + bd\left(\frac{C\Psi S\theta}{\sqrt{3}} - S\Psi\right) + 2cdC\theta C\Psi + ac\left(\frac{S\theta}{\sqrt{3}} - C\theta S\Psi\right)$$

$$Jx \dot{x} = Jq \dot{q} \quad (8)$$

After taking differentials, Eq. (7) can be rearranged to form Eq. (8):

$$P = \begin{bmatrix} 1 + dC\theta C\psi \\ 1 + dC\theta C\psi + \frac{S\theta}{2\sqrt{3}} + C\theta S\psi \\ 1 + dC\theta C\psi + \frac{S\theta}{2\sqrt{3}} - C\theta S\psi \end{bmatrix}, Q = \begin{bmatrix} \frac{abS\theta}{3} - cdS\theta C\psi - \frac{2bdC\psi C\theta}{\sqrt{3}} \\ \frac{abS\theta}{12} + \frac{S\psi C\theta}{4\sqrt{3}} + \frac{bdC\psi C\theta}{2\sqrt{3}} - cdS\theta C\psi + \frac{acC\theta}{2\sqrt{3}} - \frac{acS\theta S\psi}{2} \\ \frac{abS\theta}{12} - \frac{S\psi C\theta}{4\sqrt{3}} + \frac{bdC\psi C\theta}{2\sqrt{3}} - cdS\theta C\psi + \frac{acC\theta}{2\sqrt{3}} + \frac{acS\theta S\psi}{2} \end{bmatrix}, \tag{9}$$

$$R = \begin{bmatrix} -cdS\psi C\theta - \frac{2bdS\psi S\theta}{\sqrt{3}} \\ \frac{-C\psi S\theta}{2\sqrt{3}} + \frac{S\psi}{2\sqrt{3}} - \frac{bdS\psi S\theta}{2\sqrt{3}} + \frac{bdC\psi}{2} - cdS\psi C\theta + \frac{acC\theta C\psi}{2} \\ \frac{-C\psi S\theta}{2\sqrt{3}} + \frac{S\psi}{2\sqrt{3}} - \frac{bdS\psi S\theta}{2\sqrt{3}} - \frac{bdC\psi}{2} - cdS\psi C\theta - \frac{acC\theta C\psi}{2} \end{bmatrix},$$

Finally, the Jacobian matrix of order 3x3 is shown as

$$J = [P \ Q \ R] \tag{10}$$

where P, Q, and R are the 3x1 order matrix as shown in Eq. (9). The conditioning index ‘k’ may then be found using Eq. (12).

5.2 Performance Analysis

Performance or dexterity is the standard that deals with manipulator’s capability to obtain the given task (Shao et al., 2017). For that matter, optimal design has to be found for better performance of the manipulator. As stated previously, performance can be in terms of reachability and mobility. The calculation measure is the performance controlling parameters, which will be explained next.

5.2.1 Conditioning index

Dexterity index “K”, also known as the condition number, is the ability of the mechanism to change its poses randomly. Its value will range from 1 to infinity, while the conditioning index is the reciprocal of dexterity (Wu, 2019):

$$K = \|J\| * \|J^{-1}\| \tag{11}$$

$$k = 1/K \tag{12}$$

where ‘J’ is the Jacobian matrix.

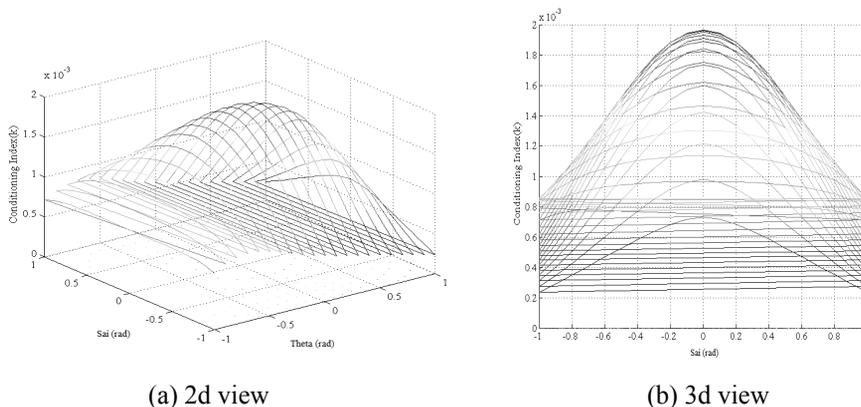


Figure 3. Conditioning index versus different orientation theta ‘θ’ and sai ‘Ψ’ at elevation ‘z’ of ‘500’mm.

Iterations have been taken with a step size of 0.1 between -1 and 1 in MATLAB. It is shown from the graph that the peak point of the curve is at 1.9e-3. The step size taken between the set of constraints is 20. The lesser the conditioning index, the higher the performance of a manipulator.

5.2.2 Global Conditioning Index

Further, check the results globally by using the global indexing performance index ‘GCI’. In other words, the global index is the mean of the conditioning index in a prescribed volume around its workspace (Angeles, 1991). The higher the global index, the higher the performance in terms of mobility of a manipulator.

5.2.3 Workspace Volume

There are many methods adopted by many researchers for the calculation of workspace volumes. Analytical and numerical approaches have been introduced previously in Deb (n.d.), and the same will be practiced here. Firstly, it takes the whole of the workspace as a cube, which has three axes x, y and z, respectively; then, it takes the subspace, a cylinder in particular for the workspace calculation. It restricts the legs and the platforms of the manipulator around a cylinder and from the inverse kinematic solutions of the parallel manipulator. By keeping the constraints in view, this searches each q’s in that subspace, which forms the closed cylinder. After each z increasing, this will try to find out the solutions that are trapped inside or onto the surface of that subspace. (Deb n.d.). MATLAB code takes the following necessary steps for workspace.

Step 1: For a certain z, find the inverse kinematics solutions for a prescribed set of parameters and their Design and Geometric constraints.

Step 2: Do the necessary procedure to make a sampled hollow cylinder as a subspace.

Step 3: Start a check for the point of solutions to be in that cylinder and discard the remaining set of points as it is beyond our boundary conditions.

Step 4: Repeat the procedure from step 1 to step 3 for z=z+1.

Step 5: Get a set for all z and save it in a column matrix form to be used later for analysis. Plot it to get a desired dexterous workspace around a subspace.

In case 1, the step size taken between the set of geometric constraints is 20. In this algorithm, it will search out the points that lie within the taken cylindrical subspace and discard the remaining set of points. 1123.2 mm³ is the dexterous workspace volume when taking only the values of inverse kinematic solutions q’s under the subspace it has taken along z ranging from 0 to 1000mm. Step size of angles is 0.1 rad. Step size for geometric constraints is 10mm.

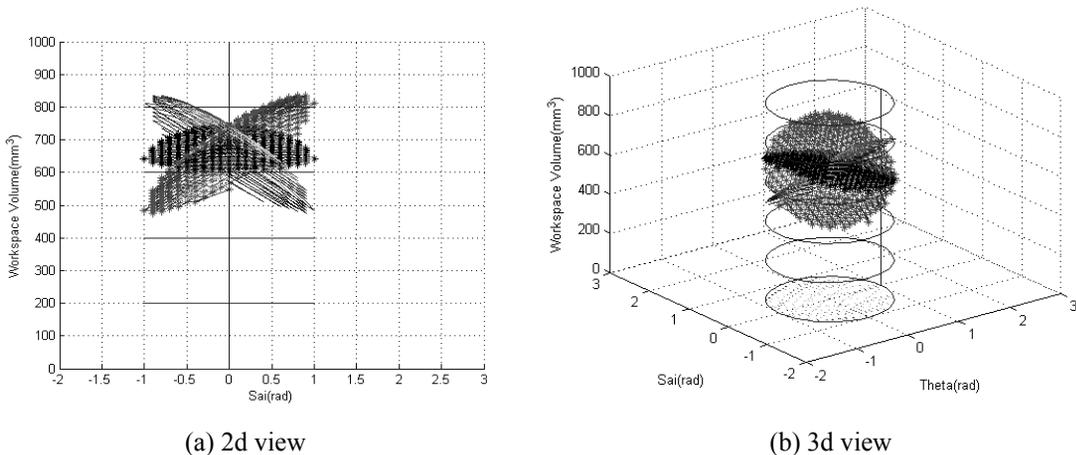


Figure 4. View of workspace volume without actuator limits in the subspace for ‘θ’ and ‘Ψ’.

It is obvious now that the region will be eliminated in the workspace calculation when keeping the actuator lengths ranging from 400 to 750 mm in view and in the subspace. After considering the actuator constraints, the volume is found to be 845.2571 mm.

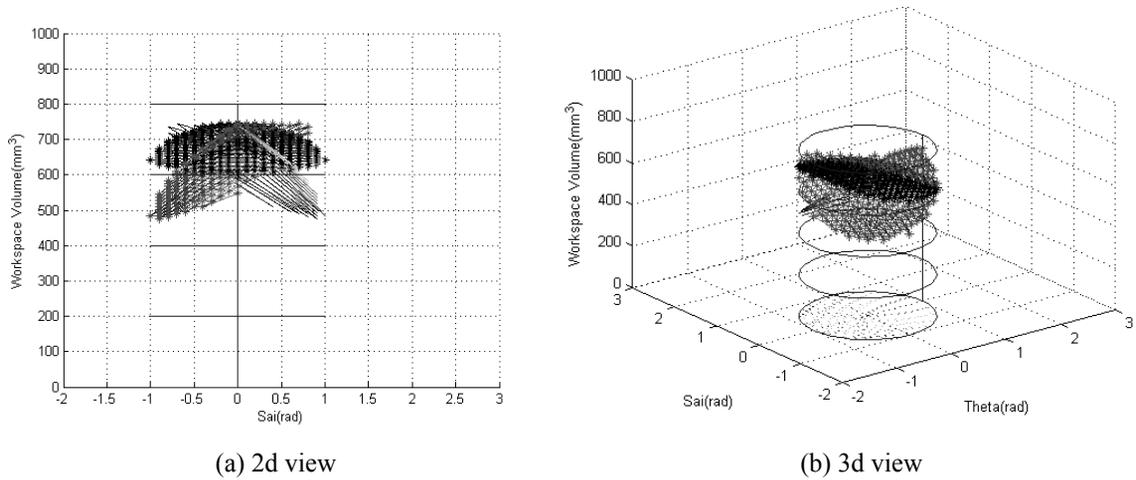


Figure 5. View of workspace volume with actuator limits of 400 to 750mm in the subspace for ‘ θ ’ and ‘ Ψ ’.

5.3 Conventional Single Objective Optimization

Optimization process for one parameter irrespective of other performance parameters will be of major concern in this section. These evolutionary algorithms perform swiftly for the findings of local and global minimal and maximal points. Other traditional methods like bracketing and elimination optimization techniques do not guarantee findings of optimum points. They can skip their local and global points. So, in order to have that EA in our parallel manipulator calculations, ant colony optimization, genetic algorithms optimization (GA), and particle swarm optimization (PSO) have recently paved the way to optimization of design variables.

5.4 Particle Swarm Optimization

PSO usually takes the advantage of lesser iterations and its higher convergence rate in the start than genetic algorithm (Rajendra and Pratihar, 2011). PSO has few setting parameters and simple and easy implementation; hence, it can be used to solve nonlinear, nondifferentiable, and multipeak optimization problems in the fields of science and engineering (Harrison, Engelbrecht, and Ombuki-Berman, 2018) (El-Shorbagy and Hassanien, 2018).

This algorithm follows the social behavior of birds, when the birds move in search for food and all do not know the exact location of food. Finally, the food is located by one bird, and it is found to be nearest so now all the birds will follow that food, which has been found by one of them (Rini and Shamsuddin, 2011). That bird can be named as a leader. PSO is a searching algorithm. PSO starts with the same process of initialization (Kennedy, 1999). Steps for the PSO algorithm are given below. All further readings and calculations are run through MATLAB.

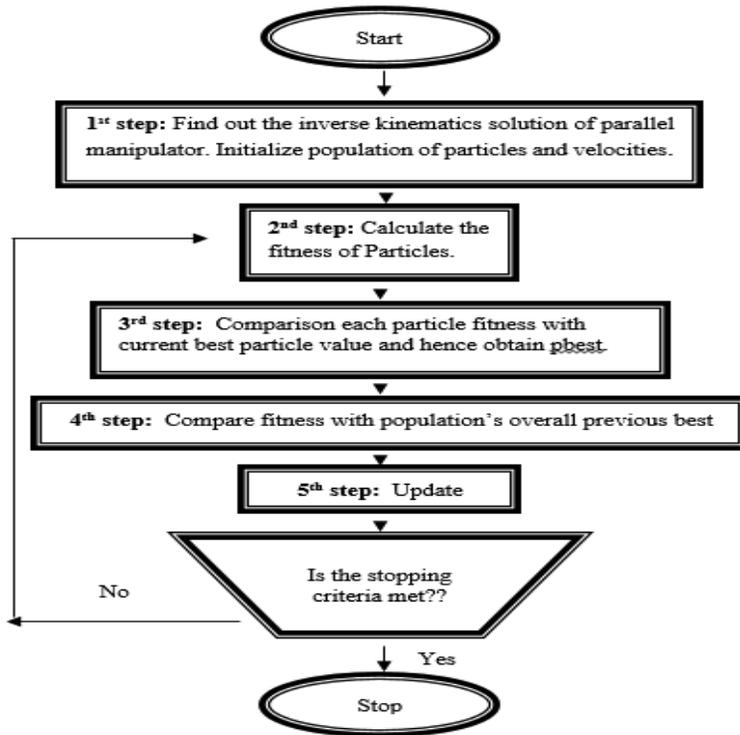


Figure 6. Flow diagram for the Particle Swarm Optimization.

The equation below is an updated velocity function for step 5 in Fig.6.

$$v_{new} = v_g(j) + c_1 * r_1 * (p_{best} - x(i,j)) + c_2 * r_2 * (g_{best}(j) - x(i,j)) \tag{13}$$

v_{new} = New velocity after update

v_g = Global velocity of the particle

p_{best} = Particle best, same at start as $x(i,j)$.

$x(i,j)$ = the value of particle taken from i^{th} row and j^{th} column from the start to size of the swarm ‘n’

g_{best} = global best is the global best of the swarm corresponding to the fitness value of the objective function.

c_1, c_2 are the first and second constants, and r_1, r_2 are the first and second random values.

Usually, these constants should both sum up to 4 in simulations, whereas r_1 and r_2 both are random values taken from 0 to 1. Similarly, the position update of the particle takes place in accordance with the velocity update equation, which is represented in this form normally (del Valle et al. 2008).

$$x_{new} = x(i,j) + v_{new} \tag{14}$$

where x_{new} is the new position of the particle. The maximum swarm values will optimize and finally declare the final constant value as maxima optimum point. The conditioning index is being optimized for the set of design variables $a, b,$ and $d,$ and PSO algorithm is launched. The aim to find a minimum point for this performance index was accomplished, and the corresponding design variables were saved against that best minimum point. The execution of the MATLAB code reveals the following results.

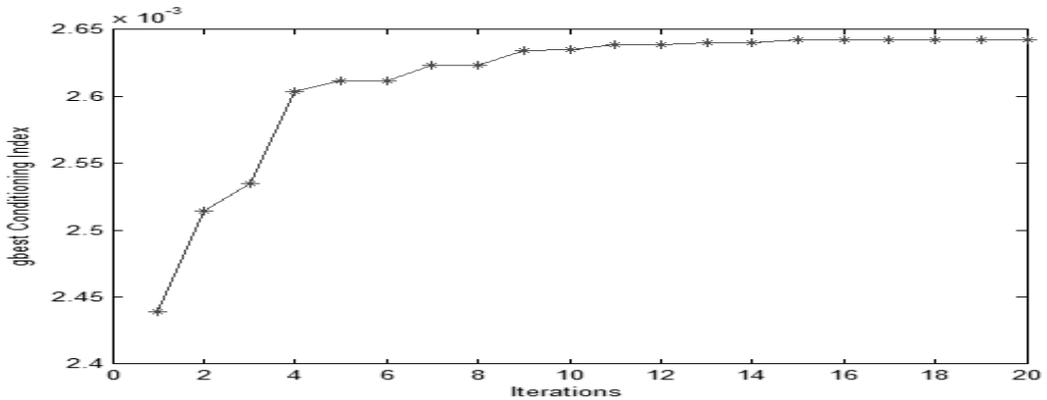


Figure 7. Conditioning index 'k' versus the iterations.

Figure 7 represents twenty iterations with a step size of 0.1 rad of angle. These values are taken for the one elevation of 'c' for 500 mm elongation. For 20 intervals between the design variables, the iteration started until a smooth constant line comes, as it is a sign that the algorithm has most probably found its optimum point. Iterations are being used here as a stopping criterion. As shown in the graph, the nearby optimum conditioning index 'k' point is .002642. Table 2 shows the corresponding optimum design variables.

Table 2. gbest parameters versus design variable.

Sol no	a(mm)	b(mm)	d(mm)	θ	Ψ	K	k
1423	203.0038	302.9585	102.369	0.2	0	532.0505	0.00188
1424	204.2438	300.8429	93.57697	0.2	0	577.5259	0.001732
1425	204.2327	302.8744	164.8218	0.2	0	378.4926	0.002642
1426	203.2896	303.1866	161.7431	0.2	0	381.9439	0.002618
1427	204.2203	300.8822	25.35778	0.2	0	2409.055	0.000415

Now, check the maximum workspace values. Firstly, run the algorithm, and then check the maximum values of the workspace until it computes.

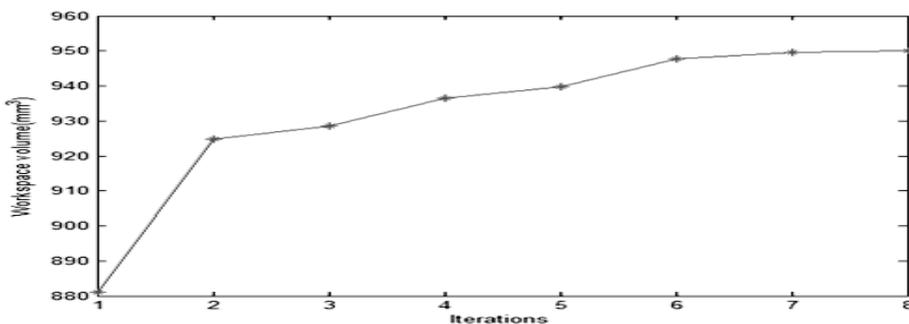


Figure 8. Maximum workspace 'Vol' values versus the iterations.

Twenty iterations for the 20 intervals have been taken between the design variables with a step size of 0.1 rad of angle. The curve in figure 8 has made its own threshold at 8 iterations. This is because the curve has reached its maximum height in 8 iterations, and PSO fast convergence is claimed and shown to be constant, and this optimum point is regarded as global maxima. gbest value for the maximum optimized volume is found to be 950.0733mm³ as shown in figure 8, and table 3 shows its corresponding optimum design variables within their geometric constraints.

Table 3. Design variables versus maximum volume for orientation ‘ θ ’ and ‘ Ψ ’.

Sol no	a(mm)	b(mm)	d(mm)	θ	Ψ	gbest Vol
252	298.6565	497.9428	163.9254	-0.2	-1	934.6555
253	298.4786	493.1514	195.2298	-0.2	-1	870.2555
254	299.9863	497.5456	154.6851	-0.2	-1	950.07333
255	298.9249	497.5168	162.3638	-0.2	-1	937.0999
256	295.3666	497.4281	170.0234	-0.2	-1	917.0779

In both the cases, for the optimization of workspace volume ‘Vol’ and conditioning index ‘k’, previously, the optimum performance points were achieved in 45 to 50 iterations by using GA in Hosseini et al. (2011)

5.5 Multiobjective Particle Swarm Optimization (MOPSO)

In rapid manufacturing, it needs an optimum design to get maximum performance by triggering parametric values at the same time. In the multiobjective one, it is desirable to produce a function comprised of performance parameters to be treated as variables that can address variations of all the parameters at the same time. New function will be treated as objective function for the job and the performance parameters act as function variables (Asadollahi-Yazdi, Gardan, and Lafon, 2018). After that, the process for the optimization remains the same (Guohua et al., 2013). There are numerous ways of forming that new function. See weighted sum strategy for further calculations (Radovanović, 2019).

5.5.1 Weighted sum method

In this multiobjective technique, the function is formed by assigning the weights. Each variable is assigned weight, which can be utilized according to the user demands. And the equation will be created (Goel and Stander, 2007). Maximize ‘y’ now, and the fitness value is examined using this function. Normalization is also performed for the weights.

$$y=w1*z(i,1)+w2*z(i,2)+w3*z(i,3) \quad (15)$$

w1, w2, and w3 are the three weights assigned to the conditioning index ‘k’, workspace volume ‘Vol’, and global Conditioning index (GCI). This method is good for continuous and convex problems; however, local optima usually achieve discontinuous functions as well (Goel and Stander, 2007).

Preference is set in start for the desirable performance parameter. Figure 9 relates the three performance parameters evaluated through the weighted sum strategy. 20 iterations with 0.1 step size of angles have been made, and the objective function shows a constant behavior. Maximization PSO runs, and it has given preference to the conditioning index ‘k’.

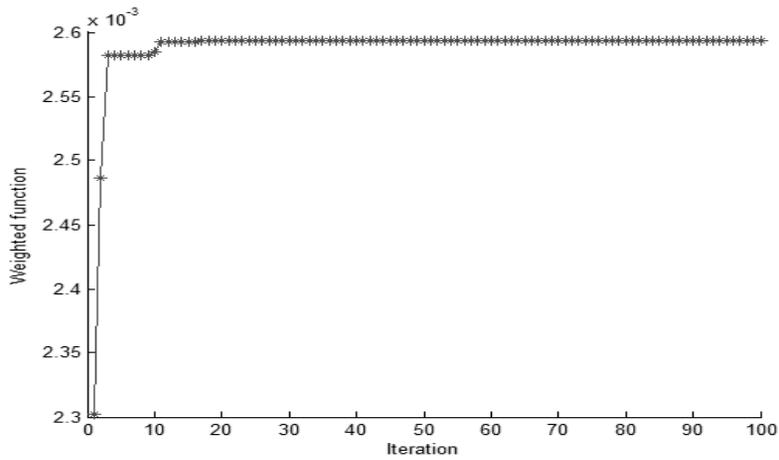


Figure 9. gbest conditioning Index ‘k’ increasing of the weighted objective function.

Table 4. Conditioning index ‘k’ against the set of GCI, workspace volume ‘Vol’, and the design variables for ‘ θ ’ and ‘ Ψ ’ of the multiobjective maximum optimization.

Sol no	a(mm)	b(mm)	d(mm)	θ	Ψ	k	Vol	GCI
4822	200.0313	305.1959	98.97398	0.1	0	0.001807	28.21148	0.000984
4823	200.0645	304.7332	88.68052	0.1	0	0.001636	27.23871	0.000985
4824	200.0367	305.2536	155.9036	0.1	0	0.002593	32.97995	0.000985
4825	200.0068	305.2877	155.0147	0.1	0	0.002583	32.91182	0.000985
4826	200.0301	305.0755	53.63707	0.1	0	0.000985	23.72593	0.000985

Set 1 for workspace volume, and 0 for the other two parameters. The MATLAB code gives the following result.

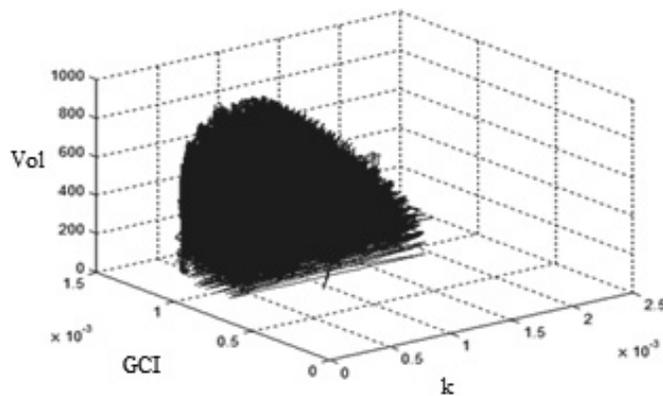


Figure 10. Multiobjective maxima with workspace volume given preference of 1.

Table 5. Workspace volume against the set of GCI, Conditioning Index 'k', and the design variables for 'θ' and 'Ψ' of the multiobjective maximum optimization.

Sol no	a(mm)	b(mm)	d(mm)	θ	Ψ	k	Vol	GCI
2529	286.0273	471.2437	173.9444	-0.4	-1	0.000735	888.6863	0.00106
2530	298.1691	495.2726	195.2288	-0.4	-1	0.000757	873.7818	0.00106
2531	296.1315	489.4961	189.9752	-0.4	-1	0.000751	935.0132	0.0001059
2532	299.8161	497.553	198.7088	-0.4	-1	0.000759	927.8974	0.0001059
2533	296.6987	490.0646	191.722	-0.4	-1	0.000752	933.3976	0.0001059

The result concludes the relationship that workspace volume 'Vol' is inversely proportional to the conditioning index 'k', and conditioning index is directly proportional to global index 'GCI'. Hence, results have been compared with those of single objective optimization and were validated.

5.5.2 Epsilon Constraint Method

This method uses one single function and restricts other functions through some constraints. Hence, it gives overall optimum desired results under that set of constraints. The user has to play through these set of constraints very carefully and needs expertise about the boundaries of the solution (Goel and Stander, 2007). Maximize y with respect to the other constraints, i.e., Function 'Y' is

$$Y = F_i(X) \text{ where } i=1, 2, \dots, I \quad (16)$$

subject to constraint 'C' and 'L'

$$C_j(X) \quad j=1, 2, \dots, J$$

$$L_k(X) \quad \text{where } k=1, 2, \dots, K \quad (17)$$

Workspace volume is being treated as a constraint, and the conditioning index 'k' is shown to be maximized under the 700 mm³ volume restriction. Maximum 'k' at this point is .002593, which will be regarded as local maxima. This result is also shown from 100 iterations. It has got its optimum in 20 iterations.

Table 6. Compensated workspace volume 'Vol', conditioning Index 'k' combined against the set of GCI, and the design variables for 'θ' and 'Ψ' of the multiobjective maximum constraint optimization.

Sol no	a(mm)	b(mm)	d(mm)	θ	Ψ	Vol	k	GCI
2529	200.1062	306.0383	88.68052	0.1	0	27.28973	0.001635	0.001012
2530	200.1541	306.1213	156.2243	0.1	0	33.06519	0.002593	0.001012
2531	200.141	306.069	156.2009	0.1	0	33.05932	0.002593	0.001012
2532	200.1247	306.0919	53.63707	0.1	0	23.75598	0.000985	0.001012
2533	200.1803	306.0872	108.1576	0.1	0	29.08988	0.00195	0.001012

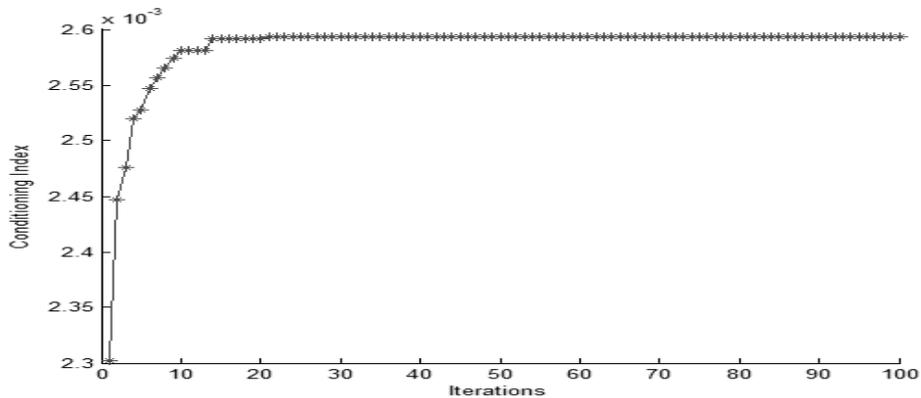


Figure 11. gbest compensation value increasing graph of Conditioning Index ‘k’ of the constraint volume up to 700mm³.

6. DISCUSSION

In our proposed methodology, multiobjective optimization of tricept manipulator is an addition to the previous research work. This work has given validation to the results shown in Hosseini et al. (2011) for single objective optimization for volume and conditioning indexes optimization, respectively, and it has been evaluated with lesser iterations, which ensures fast result with less processing effort. The corresponding design variables achieved are considered to be the best optimum values with correspondence to the best performance indexes.

MATLAB algorithm states that, after initialization of the random variables, it will calculate the performance parameters. Single objective is performed to get the results, in which performance parameters will be treated independently. If it achieves the constant line, it has found the optimum solutions, and the algorithm stops with the corresponding optimum design variables. The same process is repeated for the multiobjective technique, and the performance parameters will be served simultaneously. If it achieves the constant line, it has found the optimum solutions, and the algorithm stops with the corresponding optimum design variables. If not found, the process will be repeated iteratively for both cases until optimum solutions are achieved.

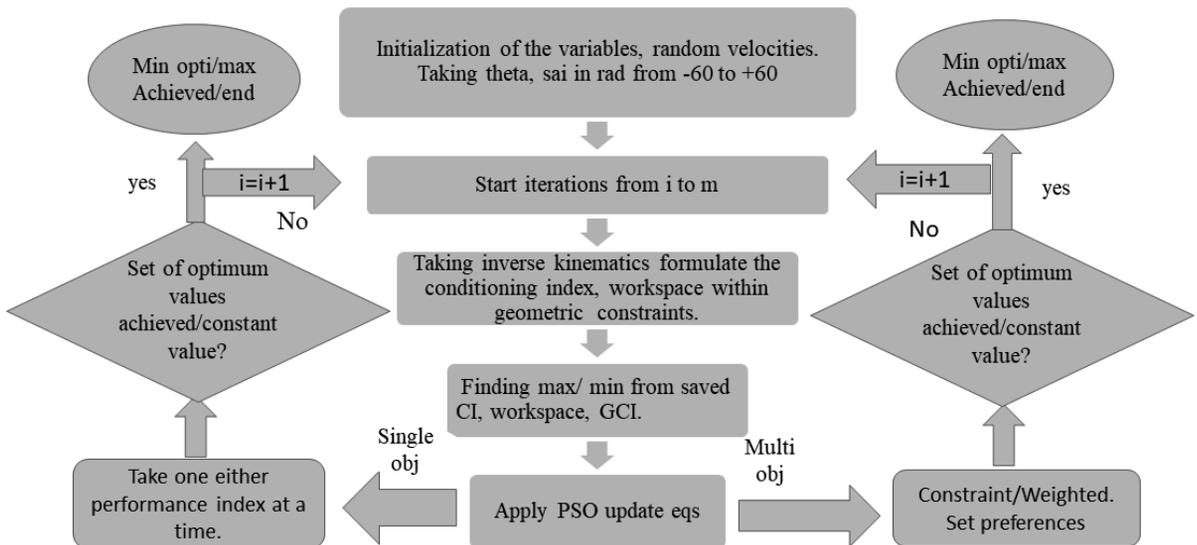


Figure 12. MATLAB algorithm.

7. CONCLUSION

This work concludes that PSO usually takes lesser iterations than previously used genetic algorithm (GA) in Hosseini et al. (2011) and is declared as faster than GA for this case. GA exerted more computation on the processor than PSO. PSO has a higher convergence rate than GA for this task. But sometimes PSO can treat the local maxima or minima as global ones so more efforts are needed to be exerted to declare the point as nearby global optimum (del Valle et al., 2008) (Qiu et al., 2018). It is random. It can try different iterations at start to see the variance of optimum points. The optimum points achieved are being claimed for maximum performance that can be extracted from this tricept manipulator with less computational effort; hence, optimum and maximum dexterity have been achieved.

Further, it can go for more performance parameters like stiffness index. More constraints can be added. Shape singularity analysis can be done completely for future works. The purpose to get the optimized design variables with less computation has been achieved for this tricept mechanism.

DECLARATION

We certify that this research work titled “Multiobjective Optimization of a Tricept Parallel Manipulator using Evolutionary Algorithm” is our own work. Work has been acknowledged through proper references. It is clearly that the work of others is attributed whenever consultation is required, and none of the work written here is presented to anyone else.

➤ Availability of data and materials

Higher simulations were done in the university research lab having i7 core PCs equipped with registered MATLAB. No material requirement was there for the task.

➤ Competing Interests

Not applicable

➤ Funding

Not applicable

➤ Authors' Contributions

All authors have worked with dedication equally to achieve the desired results.

ACKNOWLEDGMENT

Moral support of Research mates and guidance by the supervisor and research environment of KFUEIT and NUST are acknowledged.

REFERENCES

- Angeles, J. 1991. A Global Performance Index for the Kinematic Optimization of Robotic Manipulators. *Journal of Mechanical Design, Transactions of the ASME*, **113**(3): 220-26.
- Arrouk, K.A., B.C. Bouzgarrou, and G. Gogu. 2010. Workspace Determination and Representation of Planar Parallel Manipulators in a CAD Environment. Pp. 605-12 in *Mechanisms and Machine Science*. Vol. 5. Kluwer Academic Publishers.
- Asadollahi-Yazdi, Elnaz, Julien Gardan, and Pascal Lafon. 2018. Multi-Objective Optimization of Additive Manufacturing Process, *IFAC-PapersOnLine*, **51**(11): 152-57.
- Binbin, Peng, Li Zengming, Wu Kai, and Sun Yu. 2011. Kinematic Characteristics of 3-UPU Parallel Manipulator in Singularity and Its Application, *International Journal of Advanced Robotic Systems*, **8**(4): 34.
- Carretero, J.A., R.P. Podhorodeski, M.A. Nahon, and C.M. Gosselin. 2000. Kinematic Analysis and Optimization of a New

Three Degree-of-Freedom Spatial Parallel Manipulator, *Journal of Mechanical Design, Transactions of the ASME*, **122**(1): 17-24.

- Chablat, Damien and Philippe Wenger. 2003.** Architecture Optimization of a 3-DOF Translational Parallel Mechanism for Machining Applications, the Orthoglide, *IEEE Transactions on Robotics and Automation*, **19**(3): 403-10.
- Chen, Xiang, Xin Jun Liu, Fu Gui Xie, and Tao Sun. 2014.** A Comparison Study on Motion/Force Transmissibility of Two Typical 3-DOF Parallel Manipulators: The Sprint Z3 and A3 Tool Heads, *International Journal of Advanced Robotic Systems*, **11**(1).
- Cheng, Hui. 2001.** Dynamics and Control of Parallel Manipulators with Actuation Redundancy, The Hong Kong University of Science and Technology, Clear Water Bay, Kowloon, Hong Kong.
- Deb, Kalyanmoy. n.d.** Single and Multi Objective Optimization Using Evolutionary Computation.
- El-Shorbagy, M.A. and Aboul Ella Hassanien. 2018.** Particle Swarm Optimization from Theory to Applications, *International Journal of Rough Sets and Data Analysis*, **5**(2):1-24.
- Furqan, Mohd, Mohd Suhaib, and Nazeer Ahmad. 2017.** Studies on Stewart Platform Manipulator: A Review, *Journal of Mechanical Science and Technology*, **31**(9): 4459-70.
- Glözman, Daniel and Moshe Shoham. 2009.** Novel 6-DOF Parallel Manipulator with Large Workspace, *Robotica*, **27**(6): 891-95.
- Goel, Tushar and Nielen Stander. 2007.** Multi-Objective Optimization Using LS-OPT.
- Gosselin, C.M. and M. Guillot. 1991.** The Synthesis of Manipulators with Prescribed Workspace, *Journal of Mechanical Design*, **113**(4): 451.
- Gosselin, Clement; 1988.** Kinematic Analysis Optimization and Programming of Parallel Robotic Manipulators.Pdf. 252.
- Gosselin, Clément. 1988.** Kinematic Analysis, Optimization and Programming of Parallel Robotic Manipulators.
- Gough, V. 1962.** Universal Tyre Test Machine.
- Guohua, Cui, Wei Bin, Wang Nan, and Zhang Yanwei. 2013.** Stiffness, Workspace Analysis and Optimization for 3UPU Parallel Robot Mechanism, **11**(9).
- Harrison, Kyle Robert, Andries P. Engelbrecht, and Beatrice M. Ombuki-Berman. 2018.** Self-Adaptive Particle Swarm Optimization: A Review and Analysis of Convergence, *Swarm Intelligence*, **12**(3): 187-226.
- He, J.F., H.Z. Jiang, D.C. Cong, Zheng Mao Ye, and Jun Wei Han. 2007.** A Survey on Control of Parallel Manipulator, *Key Engineering Materials*, **339**: 307-13.
- Hosseini, Mir Amin; Mohammadai Hamid-Reza. 2011.** Kinematic Analysis of Tricept Parallel Manipulator, *IJUM Engineering Journal*, **12**(5): 7-16.
- Hosseini, Mir Amin, Hamid Reza M. Daniali, and Hamid D. Taghirad. 2011.** Dexterous Workspace Optimization of a Tricept Parallel Manipulator, *Advanced Robotics*, **25**(13-14): 1697-1712.
- Huang, Ming Z. 2011.** Design of a Planar Parallel Robot for Optimal Workspace and Dexterity, *International Journal of Advanced Robotic Systems*, **8**(4):176-83.
- Kennedy, James. 1999.** Minds and Cultures: Particle Swarm Implications For Beings in Sociocognitive Space, *Adaptive Behavior*, **7**(4): 269-87.
- Leal-Naranjo, José Alfredo, Marco Ceccarelli, Christopher René Torres-San-Miguel, Luis Antonio Aguilar-Perez, Guillermo Urriolagoitia-Sosa, and Guillermo Urriolagoitia-Calderón. 2018.** Multi-Objective Optimization of a Parallel Manipulator for the Design of a Prosthetic Arm Using Genetic Algorithms, *Latin American Journal of Solids and Structures*, **15**(3).
- Lmalghan, Rashmi, Karthik Rao, S. ArunKumar, Shrikantha S. Rao, and Mervin A. Herbert. 2018.** Machining Parameters Optimization of AA6061 Using Response Surface Methodology and Particle Swarm Optimization, *International Journal of Precision Engineering and Manufacturing*, **19**(5): 695-704.
- Lou, YJ, GF Liu, and ZX Li. 2005.** A General Approach for Optimal Design of Parallel Manipulators, *IEEE Transactions on Automation Science ... X(X)*: 1-16.

- Lou, Yunjiang, Guanfeng Liu, Ni Chen, and Zexiang Li. 2005.** Optimal Design of Parallel Manipulators for Maximum Effective Regular Workspace, Pp. 1208-13 in 2005 IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS.
- Mehta, Vivek Kumar and Bhaskar Dasgupta. 2011.** A General Approach for Optimal Kinematic Design of 6-DOF Parallel Manipulators, *Sadhana - Academy Proceedings in Engineering Sciences*, **36(6)**:977-94.
- Mei, Jiangping, Xu Zhang, Jiawei Zang, and Fan Zhang. 2018.** Optimization Design Using a Global and Comprehensive Performance Index and Angular Constraints in a Type of Parallel Manipulator, *Advances in Mechanical Engineering*, **10(7)**.
- Merlet, J.P. 1997.** Designing a Parallel Manipulator for a Specific Workspace, *The International Journal of Robotics Research*, **16(4)**: 545-56.
- Patel, Y.D. and P.M. George. 2012.** Parallel Manipulators Applications—A Survey, *Modern Mechanical Engineering*, **02(03)**: 57-64.
- Pond, Geoff T. and Juan A. Carretero. 2008.** Quantitative Dexterous Workspace Comparison Of Serial And Parallel Planar Mechanisms | *InTechOpen* (April).
- Qiang, Hongbin, Lihang Wang, Jisong Ding, and Lijie Zhang. 2019.** Multiobjective Optimization of 6-DOF Parallel Manipulator for Desired Total Orientation Workspace, *Mathematical Problems in Engineering* 2019.
- Qiu, Na, Yunkai Gao, Jianguang Fang, Guangyong Sun, and Nam H. Kim. 2018.** Topological Design of Multi-Cell Hexagonal Tubes under Axial and Lateral Loading Cases Using a Modified Particle Swarm Algorithm, *Applied Mathematical Modelling*, **53**:567-83.
- Radovanović, Miroslav. 2019.** Multi-Objective Optimization of Multi-Pass Turning AISI 1064 Steel, *International Journal of Advanced Manufacturing Technology*, **100(1-4)**: 87-100.
- Rajendra, Rega and Dilip K. Pratihari. 2011.** Particle Swarm Optimization Algorithm vs Genetic Algorithm to Develop Integrated Scheme for Obtaining Optimal Mechanical Structure and Adaptive Controller of a Robot, *Intelligent Control and Automation*, **02(04)**: 430-49.
- Rezania, Vahid and Saeed Ebrahimi. 2017.** Dexterity Characterization of the RPR Parallel Manipulator Based on the Local and Global Condition Indices, *Journal of Mechanical Science and Technology*, **31(1)**: 335-44.
- Rini, Dian Palupi and Siti Mariyam Shamsuddin. 2011.** Particle Swarm Optimization: Technique, System and Challenges, *International Journal of Applied Information Systems*, **1(1)**:33-45.
- Shah, M.F., Z. Kausar, and S.S. Farooq. 2018.** Workspace Design for 6 Degree of Freedom Machining Bed, *Journal of Physics: Conference Series* **1016(1)**: 012018.
- Shao, Zhu Feng, Jiao Mo, Xiao Qiang Tang, and Li Ping Wang. 2017.** Transmission Index Research of Parallel Manipulators Based on Matrix Orthogonal Degree, *Chinese Journal of Mechanical Engineering (English Edition)* **30(6)**: 1396-1405.
- Siciliano, Bruno. 1999.** Tricept Robot: Inverse Kinematics, Manipulability Analysis and Closed-Loop Direct Kinematics Algorithm, *Robotica*, **17(4)**: 437-45.
- Spong, Mark W. 1995.** Adaptive Control of Flexible Joint Manipulators: Comments on Two Papers, *Automatica*, **31(4)**: 585-90.
- Stamper, R.E., Lung-Wen Tsai, and G.C. Walsh. n.d.** Optimization of a Three DOF Translational Platform for Well-Conditioned Workspace, Pp. 3250-55 in *Proceedings of International Conference on Robotics and Automation*. Vol. 4. IEEE.
- Sudheer, A.P. and Prabhat Kumar. 2017.** Workspace Optimization of 3PRR Parallel Manipulator for Drilling Operation Using Genetic Algorithm, in *ACM International Conference Proceeding Series*. Vol. Part F1320. Association for Computing Machinery.
- Tanev, Tanio K. 2000.** Kinematics of a Hybrid (Parallel-Serial) Robot Manipulator, *Mechanism and Machine Theory*, **35(9)**: 1183-96.
- Tsai, LW. 1999.** Robot Analysis: The Mechanics of Serial and Parallel Manipulators, *The Mechanics of Serial and Parallel Manipulators* 520.
- del Valle, Yamille, Ganesh Kumar Venayagamoorthy, Salman Mohagheghi, Jean Carlos Hernandez, and Ronald G. Harley. 2008.** Particle Swarm Optimization: Basic Concepts, Variants and Applications in Power Systems, *IEEE Transactions on Evolutionary Computation*, **12(2)**: 171-95.

- Wang, Congzhe, Yuefa Fang, and Sheng Guo. 2015.** Multi-Objective Optimization of a Parallel Ankle Rehabilitation Robot Using Modified Differential Evolution Algorithm, *Chinese Journal of Mechanical Engineering (English Edition)*, **28**(4): 702-15.
- Wang, J.S., Michael Stock, Karol Miller, Bruno Siciliano, M.A. Laribi, L. Romdhane, S. Zegloul, Daniela Craita Ciocardia, and Stefan Staicu. 2003.** The Tricept Robot: Inverse Kinematics, Manipulability Analysis and Closed-Loop Direct Kinematics Algorithm, *Journal of Mechanical Design*, **125**(February): 870-75.
- Wang, Jingguo, Yangmin Li, and Xinhua Zhao. 2010.** Inverse Kinematics and Control of a 7-DOF Redundant Manipulator Based on the Closed-Loop Algorithm, *International Journal of Advanced Robotic Systems*, **7**(4): 37.
- Wu, Xiaoyong. 2019.** Optimal Design and Singularity Analysis of a Spatial Parallel Manipulator, *Symmetry*, **11**(4).
- Xie, Zenghui, Fugui Xie, Xin Jun Liu, Jinsong Wang, and Xu Shen. 2019.** Parameter Optimization for the Driving System of a 5 Degrees-of-Freedom Parallel Machining Robot with Planar Kinematic Chains, *Journal of Mechanisms and Robotics*, **11**(4).
- Yusup, Norfadzlan, Azlan Mohd Zain, and Siti Zaiton Mohd Hashim. 2012.** Overview of PSO for Optimizing Process Parameters of Machining. Pp. 914-23 in *Procedia Engineering*. Vol. 29.
- Zhang, Dan. 2010.** Parallel Robotic Machine Tools.
- Zhang, Xu and Hao Fang. 2012.** Optimization of a 3-PRS Parallel Manipulator Based on Interval Analysis, Pp. 2452-56 in *Proceedings of the World Congress on Intelligent Control and Automation (WCICA)*.