Chapter 6

TRAJECTORY PLANNING OF A ROBOTIC MANIPULATOR

The layout design of single and multi-robot workcell has been explained in previous chapters, and the present chapter explains the trajectory planning of robotic manipulator. This chapter presents the useful technique to plan the robot trajectory by using point cloud simulation approach. The proposed technique takes into consideration the kinematic as well as the dynamic parameters of the robot during motion. In this technique, three different profiles has been taken and finally recommends the best profile under given conditions. This technique has applied to an industrial robotic workcell used for pick and place operation.

The introduction of the trajectory planning and the previously developed approaches are briefed in section 6.1. Section 6.2 demonstrates the collision detection problem during motion, and a trajectory optimization problem is formulated in section 6.3. The results obtained by solving the optimization problem are presented and discussed in section 6.4. Finally, section 6.5 presents some conclusions drawn from the research work presented in this chapter.

6.1. Introduction

The objective of an industrial workcell is to produce desired product at minimum possible efforts in terms of time, cost, energy, and jerk. Most of the robotic workcells handle repetitive tasks, thus, the trajectory of the robot end-effector becomes the crucial factor in optimizing their overall performance. For simulating trajectory motion, the offline trajectory planning methods often utilize informal ways for modeling robotic manipulators and machines for checking collision between them (Zha and Du, 2001; Chettibi *et al.*, 2004; Valero, Mata, and Besa, 2006). The line model for robot and spherical model for machines were used. The problem formulation also lacks constraints considering the machine geometry which affects the authenticity of the solution. Therefore, the approach lacks reliability and thus required several manipulations at the production planning level.

However, the advent of CAD-based trajectory planning methods such as IRoSim (Baizid *et al.*, 2016) and by Neto and Mendes (2013), considerably changes the trajectory planning scenario. The application programming interface of drafting software (SolidWorks, AutoCAD) allows the transfer of geometric data of the drafted models to the coding platform based on Visual Basic, Visual C++, Matlab. This approach can solve various planning problems, due to their open programming interface.

Several commercial robotic software (Delmia and RobotStudio) provide extensive graphical details of the existing robotic systems used in the industries and produces quality results with graphical images for profound understanding and feedback. The planning functions and visual feedback of the generated results through GUI helps in generating the results according to real workcell scenario. However, high licensing cost and controlled research environment, have decreased their use to a lesser number of applications.

The virtual simulator was a significant development in realistic collision detection involving geometric modeling and simulation of robot. The user interface with simulator helps in analysing the collision and providing visual image to the user. However, the modeling of machines, and trajectory optimization has been neglected in the approach (Fawaz *et al.*, 2009). Saravanan *et al.* (2009) has formulated multi-objective optimization problem and added constraints for obstacle in the configuration space during trajectory motion. Heuristic algorithms specifically search for global minima, which can handle geometric constraints in Cartesian space and omits requirement of initial feasible solution. However, the trajectory planning approach lacks the efficient collision simulation and suffers from weak modeling of robots and machines.

Further, the adaptive learning algorithms for trajectory interpolation, sophisticated obstacle avoidance, and trajectory planning approaches using augmented reality ae advantageous for precision trajectory planning problems. These algorithms work in a dynamic environment with motion modulation by the potential fields. They are capable of searching the shortest traveling path under close collision tolerance with obstacles and machines (Wang, Wang, and Rao, 2010; Phung *et al.*, 2011; Nee *et al.*, 2012). However, the augmented reality and the adaptive learning approaches require auxiliary devices and complex simulation environments to evaluate trajectory planning problems.

The latest approaches in trajectory path optimization also works on dual step constrained problem formulation which generates more realistic configuration of robot (Menasri *et al.*, 2015). The position of end-effector center point at each trajectory step has been evaluated and simultaneously get the advantage of avoiding singularities and obstacle in Cartesian space. However, the machine modeling and simulation has been the common drawback of this approach. The combination of spheres forms the model of obstacle and robot links that covers large volume in comparison to the original machine in the workspace and can restricts several optimal solutions. Moreover, the collision has been simulated by generating the control points on each sphere to measure point to point distance.

The evolutionary algorithms in conjunction with the point cloud modeling can be advantageous for automation of manufacturing processes. (Mineo *et al.*, 2016) presents an interdisciplinary application regarding robotics, CAD, and mesh modeling for developing the path of robotic manipulator for NDT inspection of metallic sheets. Therefore, by grasping the broad perspective of the point cloud modeling, it has utilized for task-based optimization of trajectory of an industrial manipulator.

Nowadays, in several manufacturing processes, the point cloud modeling and CAD modeling has been preferred to perform the precision requiring tasks such as mass balancing of automobile crankshaft (Guarato *et al.*, 2017) and 3D quality inspection of the machined surface (Yu and Wang, 2014). The real working object is scanned to generate point cloud image, and its CAD model has been used for referral for point cloud image. Then, the point cloud data has been fed to subsidiary package or programming interface to accomplish the desired objective.

The goal of the present chapter is to demonstrate the procedure of optimizing the trajectory of robotic manipulator using point cloud simulation approach for an industrial workcell. The optimality criteria have been formulated to best suit the different trajectories taken. Three distinct predefined trajectory profiles were selected for best fit under point cloud constrained environment. The multi-criteria function has been used which is a weighted combination of total task time and the sum of the squared joint jerk. This procedure was successfully applied and demonstrated on the data collected from the industrial workcell (refer section 3.5, Chapter 3, for details). The point cloud simulation approach uses Axis Aligned Bounding Box based collision detection for point clouds demonstrates the potential of searching for close proximity solutions in workspace of the robot.

6.2. Collision Detection

Trajectory planning of a robotic manipulator is defined as searching for optimal flight points in the configuration space between two fixed points, traced by the center point of end-effector. The difference occurs when the trajectory passes through a fixed via-point, which divides the trajectory into two parts. This method optimizes the location of the via-point within *C*-space, subjected to kinodynamic constraints. Trajectory optimization with via-point location search adds the benefit of optimum global search with redundancy. Moreover, the via-point enable to manage trajectory by satisfying the kinematic parameters for complex problems. The primary purpose of introducing the via-point in the trajectory planning has been explained in Fig. 6.1.

Fig. 6.1 shows the point cloud maps of the industrial workcell, simulated by using point cloud simulation approach using actual dimensional data. The robotic manipulator is following a predefined trajectory as shown in close view in Fig. 6.1b. According to the workcell configuration, at these endpoints, two machines are placed with given orientation and geometry. During motion, tool attached to the manipulator collides with the machine M2 as shown in Fig. 6.1b (encircled) and the given trajectory, thus becomes invalid. The trajectory planning without using the actual dimension of the robot and machines causes such problems during implementation. The collision between tool and machine is close, which signifies the use of actual dimension of each object for simulation. Hence to avoid collision, end-effector displaces its path and passes through a new location away from the collision area.

Several approaches have been discussed earlier in previous section to solve such problem. However, to utilize the precision of point cloud modeling and robustness of the via-point search, a new trajectory planning approach has been presented in this chapter. A premium feature of the proposed technique is that it generates practically valid solutions and the point cloud modeling supports it and also helps in a close tolerant iteration of the optimum solution.

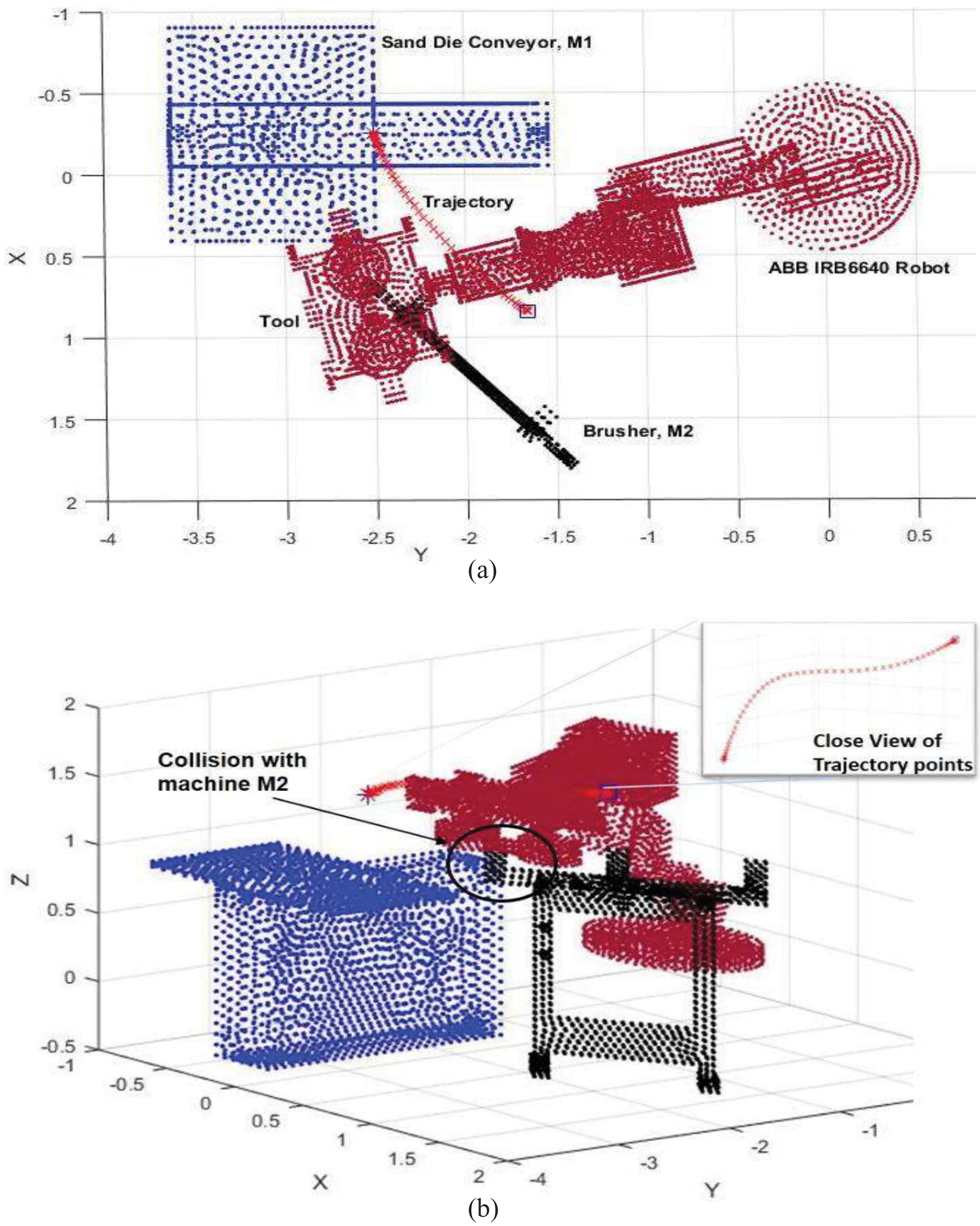

Figure 6.1. Point cloud maps of the industrial workcell having two machines and a robot with tool (a) top view and (b) isometric view

The collision detection is a vital part of the trajectory planning problem therefore much attention should be given to enhancing its ability and applicability specifically. The authors have developed such algorithm, to plan the layout of an industrial robot workcell.

The planning algorithm has utilized axis-aligned bounding box technique and combined it with point cloud modeling to design a workcell layout. Furthermore, this method has been extended for multi-robot workcell layout design in which a single robot workcell was transformed into a multi-robot workcell. This striking multirobot workcell design has been possible by using the precise planning approach which can adjust Cartesian space constraints (Sharma and Jha, 2017; Sharma *et al.*, 2017).

6.3. Problem Formulation

Point cloud simulation approach-based trajectory optimization and collision detection technique has been developed in Matlab. The trajectory planning problem is formulated with dual criteria objective function considering the weighted sum of time and jerk for specific time intervals. The objective function is subjected to three types of constraints as kinematic, dynamic and geometric. The dynamic constraint is imposed by the torque limits of the manipulator while the velocity and acceleration limits impose the kinematic constraints. Geometric constraints during manipulator trajectory motion has been imposed by the geometry of the manipulator links, tool, and machines in the workcell. These constraints has been generated by point cloud modeling algorithm (Sharma *et al.*, 2017) using real dimensions of workcell objects.

Several trajectory planning problems consider time-based objective function as optimal criterion because the time factor directly affects the length of the trajectory traced by the end-effector and overall production time (Rubio *et al.*, 2009; Liu *et al.*, 2013; Baizid *et al.*, 2016). However, the hybrid objective function considering time with jerk or power or penalty function in some cases have been used readily (Gasparetto and Zanotto, 2007; Gasparetto Lanzutti *et al.*, 2012; Yang *et al.*, 2013). The present method formulates the objective function by taking jerk term from Gasparetto and Zanotto (2007) and execution time from Chettibi *et al.* (2004) taking into account both dynamic and kinematic constraints together. The variation in actuator torque directly modulates the trajectory motion, as it affects the actuator power and related kinematic parameters (angular velocity and position). Several authors have used the dynamic behavior of the robot for trajectory planning with obstacle avoidance (Zha and Chen, 2004; Valero *et al.*, 2006; Al-Dois *et al.*, 2013).

Considering, n d.o.f. serial robotic manipulator fixed to the origin of the Cartesian coordinate frame. The dynamic equation of motion for this robot can be derived by using Lagrange's equations (Angeles, 2003) assuming no external force applied during motion, is given by:

$$M(q)\ddot{q} + Q(q, \dot{q}) + G(q) = \tau \qquad \ldots (6.1)$$

Where M is the inertia matrix, Q is a vector for centrifugal and *Coriolis* forces, G is the gravitational forces vector, and τ is the vector of actuator effort. The robot travel in several sections from start to end configuration and is passing through few via-points. This path is governed by a predefined mathematical profile for each section. The task is to search the optimal position of the via-points satisfying velocity, acceleration and torque bounds to evaluate the formulated objective function. Especially under geometric constrains where configuration space is highly restricted and manipulator may take sharp turn to avoid collisions. Moreover, the deficiency of jerk bounds can be overcome by the combine effort of continuous acceleration profile and its bounds and minimization of the jerk by the objective function.

Objective function

$$min \quad \alpha N \sum_{i=1}^{p-1} w_i + (1-\alpha) \sum_{j=1}^{N} \int_0^{t_f} \left(\dddot{\boldsymbol{q}}_{\boldsymbol{j}}(t)\right)^2 \, \mathrm{d}t \qquad \ldots (6.2)$$

subjected to,

$$\left|\dot{\boldsymbol{q}}_{\boldsymbol{j}}(t)\right| \leq \dot{q}_{max},\ j = 1,\ldots,N \qquad \ldots (6.3)$$

$$\left|\ddot{\boldsymbol{q}}_{\boldsymbol{j}}(t)\right| \leq \ddot{q}_{max},\ j = 1,\ldots,N \qquad \ldots (6.4)$$

$$\left|\boldsymbol{\tau}_{\boldsymbol{j}}(t)\right| \leq \tau_{max},\ j = 1,\ldots,N \qquad \ldots (6.5)$$

$$\boldsymbol{T_{jk}} = [\ \{ \left({}^{j}_{x}M_{min} \leq {}^{k}_{x}R_{max}\right) \oplus \left({}^{j}_{x}M_{max} \geq {}^{k}_{x}R_{min}\right) \}\ldots$$
$$\vee \{ \left({}^{j}_{y}M_{min} \leq {}^{k}_{y}R_{max}\right) \oplus \left({}^{j}_{y}M_{max} \geq {}^{k}_{y}R_{min}\right) \}\ldots$$
$$\vee \{ \left({}^{j}_{z}M_{min} \leq {}^{k}_{z}R_{max}\right) \oplus \left({}^{j}_{z}M_{max} \geq {}^{k}_{z}R_{min}\right) \}\ ],\ \text{j = 1,.., m k = 1,..,n} \qquad \ldots (6.6)$$

Where the symbols appearing in Equations 6.2 to 6.6 are as follows: α is the weighting factor, w_i is the time interval between two via-points, $\dot{\boldsymbol{q}}_{\boldsymbol{j}}(\boldsymbol{t})$ is the velocity of *j*th joint, $\ddot{\boldsymbol{q}}_{\boldsymbol{j}}(\boldsymbol{t})$ is the acceleration of *j*th joint, $\dddot{\boldsymbol{q}}_{\boldsymbol{j}}(\boldsymbol{t})$ is the jerk of *j*th joint, $\boldsymbol{\tau}_{\boldsymbol{j}}(\boldsymbol{t})$ is the torque of *j*th joint, $\dot{q}_{max}$ is the velocity limit, $\ddot{q}_{max}$ is the acceleration limit, τ_{max} is the torque limit, N is the number of robot joints, p is the number of via-points, t_f is the total execution time of trajectory. $\oplus$ is the logical symbol for the *XOR* gate and $\vee$ is the logical symbol for *OR* gate.

${}^{j}_{x}M_{min}$ and ${}^{j}_{x}M_{max}$ are the minimum and maximum value of minimum bounding box of *j*th machine point cloud data matrix *M* along x-axis respectively. m is the number of machines in a workcell. ${}^{k}_{x}R_{min}$ and ${}^{k}_{x}R_{max}$ are the minimum and maximum value of the minimum bounding box along the x-axis of *k*th link of robot *R,* respectively. Similarly, other values in $\boldsymbol{T_{jk}}$ along y-axis and z-axis for *j*th machine and link of robot can be obtained. $\boldsymbol{T_{jk}}$ calculates the intersection between point cloud data of machine and robot link along all three axes. If the intersection at any axis is less than zero then collision is detected for that configuration of robot.

Table 6.1. D-H & kinodynamic parameters of ABB IRB 6640 robot

Joints	***D-H Parameters***			***Position bounds, rad***	***Velocity max., rad/s***	***Acceleration max., rad/s²***	***Torque max., N/m***
	d, mm	***a, mm***	***α, rad***				
1	780	320	-π/2	[-2.96 2.96]	1.91	5.44	6700
2	0	1075	0	[-1.13 1.48]	1.57	5.09	6300
3	0	200	-π/2	[-3.14 1.22]	1.57	7.29	6500
4	1142	0	-π/2	[-5.23 5.23]	3.31	42.01	990
5	0	0	π/2	[-2.09 2.09]	2.44	27.00	1300
6	200	0	0	[-6.28 6.28]	4.09	59.34	500

In this work, three different polynomial trajectories were evaluated under point cloud constrained problem. Initially, the point cloud models of machines were fixed at the given location, which are the endpoints of the trajectory. Inverse kinematics can determine the joint coordinates of the given end and via-points. The base of point cloud model of robot is fixed to ground and end-effector traverse from one machine to another along a predefined trajectory generated by trajectory interpolation method. Among the trajectory planning approaches mentioned in section 1, the B-spline profile is often used where smoothness is required. However, the limitation is that the degree of B-spline should be more than four for achieving thrice continuously differentiable profile. Due to this, the number of via points required increases to four. Thus, the trajectory planning is divided into two cases: (1) single via-point and (2) multi-via-point trajectory planning.

The first profile is a polynomial interpolation function having a fifth-degree time-dependent relationship with joint angles (Craig, 2005) and can be written as

$$q(t) = a_{i,5}t^5 + a_{i,4}t^4 + a_{i,3}t^3 + a_{i,2}t^2 + a_{i,1}t + a_{i,0} \qquad i = 1, \ldots, n \qquad \ldots (6.7)$$

Where q is the joint angle as a function of time and $a_{i,j}$, j= 0, .., 5 are constant coefficient to be determined, n is the degree of freedom of manipulator. In the single via-point trajectory optimization two fifth degree polynomial have been used. However, the trajectory planning through the above approach results in discontinuous acceleration at the via-point and leads to infinite peaks in jerk profile. Therefore, the solutions proposed by R.L. William (Williams, 2013) has been adopted and extended for multi degree of freedom robot for the single via-point problem. The single sixth order trajectory has been designed to pass through a via-point with continuous velocity, acceleration and jerk. The single sixth order trajectory for *n d.o.f* robot can be expressed as

$$q(t) = a_{i,6}t^6 + a_{i,5}t^5 + a_{i,4}t^4 + a_{i,3}t^3 + a_{i,2}t^2 + a_{i,1}t + a_{i,0} \quad i = 1,\ldots, n \qquad \ldots (6.8)$$

Where $a_{i,j}$ are constant coefficients to be evaluated by boundary conditions. For continuity at the via-point, it is treated as a trajectory point q_v at some time t_v. For solving the $7n$ coefficients, we need *7n* equations to be solved and similar number of conditions to generate them. The seven constraints applied to the problem for time interval 0 to T for i = *1,…, n* are,

$$q_i(0) = q_s \qquad \dot{q}_i(0) = 0 \qquad \ddot{q}_i(0) = 0 \qquad \ldots (6.9a)$$

$$q_i(t_v) = q_v \qquad \ldots (6.9b)$$

$$q_i(T) = q_f \qquad \dot{q}_i(T) = 0 \qquad \ddot{q}_i(T) = 0 \qquad \ldots (6.9c)$$

The constraints applied keeps zero velocity and acceleration at start and end. Also, the time-based continuity at via-point ensure twice continuously differentiable in the interval $[t_{j,}\ t_{j+1}], j = 1, \ldots, T$. Among 7n coefficients, the 3n coefficients can be found from constraints given in eq. 6.9a for the initial time value which directly evaluate coefficients as $a_{i,0} = q_s$, $a_{i,1} = 0$ and $a_{i,2} = 0$. From Equations 6.9b & 6.9c, 4n third order equations are obtained as:

$$\begin{bmatrix} 1 & t_v & t_v^2 & t_v^3 \\ 1 & T & T^2 & T^3 \\ 3 & 4T & 5T^2 & 6T^3 \\ 6 & 12T & 20T^2 & 30T^3 \end{bmatrix} \begin{bmatrix} a_{1,3} & \cdots & \cdots & a_{n,3} \\ a_{1,4} & \cdots & \cdots & a_{n,4} \\ a_{1,5} & \cdots & \cdots & a_{n,5} \\ a_{1,6} & \cdots & \cdots & a_{n,6} \end{bmatrix} = \begin{bmatrix} (q_v - q_s)/t_v^3 \\ (q_f - q_s)/T^3 \\ 0 \\ 0 \end{bmatrix} \quad \text{. . . (6.10)}$$

From equation 6.10 remaining coefficients can be calculated and by putting the via-point time as T/2, the equation will be further simplified. For multi-via-point trajectory planning using a fifth degree or higher profile will be costly as the number of discontinuity will increase, and the jerk spikes exaggerate, resulting in colossal vibration and chatter. In this case, splines are used which have multi-polynomials and continuous differentiability at each trajectory point. The B-spline trajectory planning proposed by Gasparetto and Zanotto (2007) has been adopted for this problem to ensure smooth trajectory interpolation. The present method evaluates the B-spline trajectory under point cloud constraints. B-splines uses fifth-degree polynomial which is thrice continuously differentiable. The fifth degree B-spline trajectory equation can be expressed as:

$$q_5(t) = \sum_{j=i}^{n+1} S_j . B_{j,5}(t) \quad \text{. . . (6.11)}$$

Where q is the number of trajectory points in joint space and S_j(j=0, 1…, n) are the number of control points to be solved. The basis function $B_{j,5}(t)$ for knot vector l_j can be obtained from the De Boor recursion formula (Carl de Boor, 1978) as:

$$\begin{cases} B_{j,5}(t) = \frac{t - l_j}{l_{j+5} - l_j} B_{j,4}(t) + \frac{l_{j+6} - t}{l_{j+6} - l_{j+1}} B_{j+1,4}(t) \\ B_{j,0} = \begin{cases} 1, & for, \quad l_j \le t < l_{j+1} \\ 0, & elsewhere \end{cases} \end{cases} \quad \text{. . . (6.12)}$$

6.4. Results & Discussion

The proposed methodology based on point cloud simulation approach is implemented on a robotic workcell having four machines and one robot at the center is illustrated in section 3.5. The data required regarding trajectory planning between any two adjacent machines of the workcell has taken from chapter 3 and the problem formulated in previous section has been analysed. The trajectory planning is a minimization problem, which is explored for various cases by varying the number of via-points and using different predefined trajectory profiles. Two cases discussed in this section are (1) Single via-point solution and (2) Multi-via-point solution.

6.4.1. Single via-point solution

In this case, the trajectory of the end-effector is passing through a via-point in the workspace of the robot between two machines. The via point trajectory planning facilitates to iterate the vast number of solutions for the series of trajectory points in workspace. The optimal location of via-point has been determined by optimizing the kinodynamic parameters at joints. The trajectory planning has been performed in joint space using inverse kinematics formulation. In the present solution, the trajectory points were traced by last joint of the robot while tool interacts with machines and reflect changes directly to robot joint configuration.

It is due to the fact that the tool is used to pick and place workpiece on machines at certain point which may be outside the robot workspace. Thus, the machines are placed close to robot and layout become congested near robot base. It is important to note here that the last joint of robot is set as lead point that follows the entire trajectory. Therefore, the location of machines has been offset (according to tool dimensions: x, y, z coordinate values) to avoid the unnecessary collision between tool and machines at the endpoints of trajectory where the manipulator is at rest. The results obtained from several test run of the algorithm (refer. Fig. 3.8) in Matlab has been shown in Fig. 6.2.

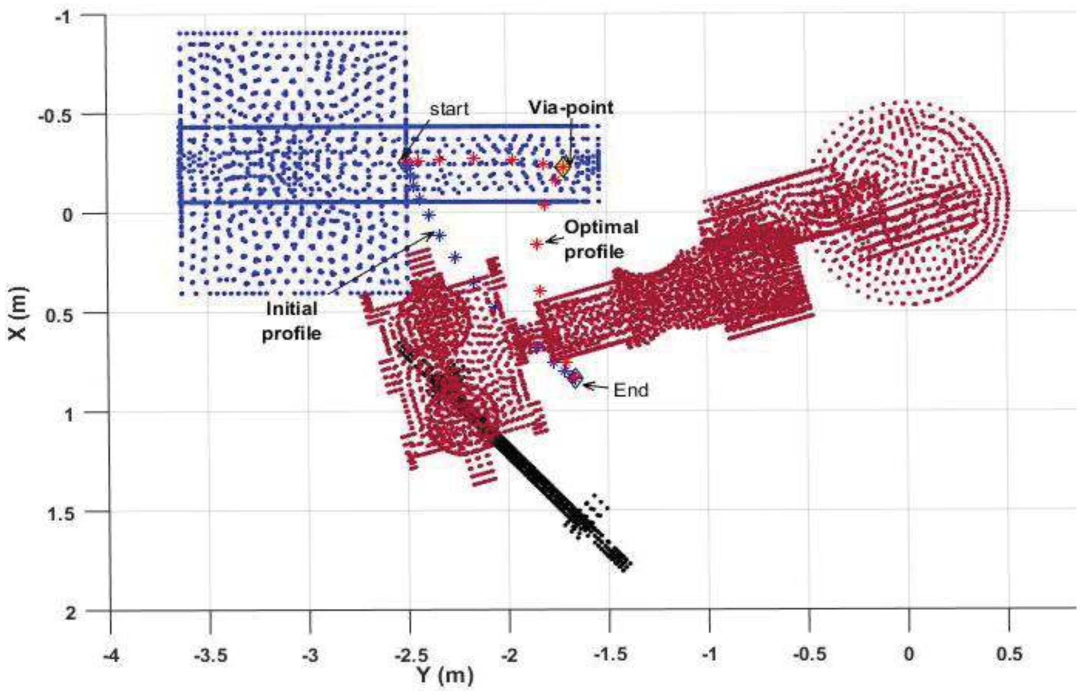

Figure 6.2. Point cloud maps of robotic manipulator with two machines in a workcell with optimal and original trajectories

Fig. 6.2 shows the optimal trajectory of the fifth-degree polynomial profile with a via-point and the original trajectory profile followed by manipulator in the industrial workcell between point cloud of two machines. To understand the working principle of point cloud simulation approach, fifth degree polynomial interpolation has been taken initially. From fig. 6.2, it is clear that the trajectory points are followed by the lead point of robot from start to end while the tool deals with machines during motion. The optimal trajectory is notably displaced in comparison to the original trajectory according to the location of via-point. This location of via point (as shown in Fig. 6.3) has been due to avoiding the collision between the tool and machine M2, as explained in section 6.2 and shown in Fig. 6.1b. The collision between the point cloud model of machine and tool changes the location of the via-point in the Cartesian space, thus becomes a crucial factor for searching its optimal location.

Along with it, the precise collision avoidance algorithm and highly constrained problem formulation agglomerate to cause massive displacement of via point. Another noticeable point is that the optimal via-point is located equidistant from each endpoint and the deviation occurs at the open space between two machines. Therefore, the location and orientation of machines also affect the trajectory of the manipulator.

Fig. 6.3 shows the isometric view of the workcell having point clouds of two machines and a robotic manipulator approaching machine M2. Both point cloud machines are located at the endpoints of the trajectory. The trajectory is divided into two sections by a via-point whose optimal location has been determined. The total execution time of the trajectory is also optimized, with a fixed time step of 0.1 s. The point cloud map indicates remarkable change in the trajectory plan of the manipulator, due to the introduction of point cloud models of robot, tool and machines. The tool also possesses sand cores, and their dimensions are also considered during optimization.

Fig. 6.4 shows the jerk profiles or third order derivative of the fifth and sixth order polynomial trajectories with respect to time. The profiles indicate the jerk values at each joint of the robot during motion. Fig. 6.4a shows the jerk profile of the original trajectory of the robot, showing continuous but high jerk values at the end. Indeed, the profile shown in Fig. 6.4b have lower jerk at ends but the motion has been disrupted at the via-point which affects the continuity. This abrupt change in jerk profile at via-point decrease smoothness and increase the overall jerk values during motion.

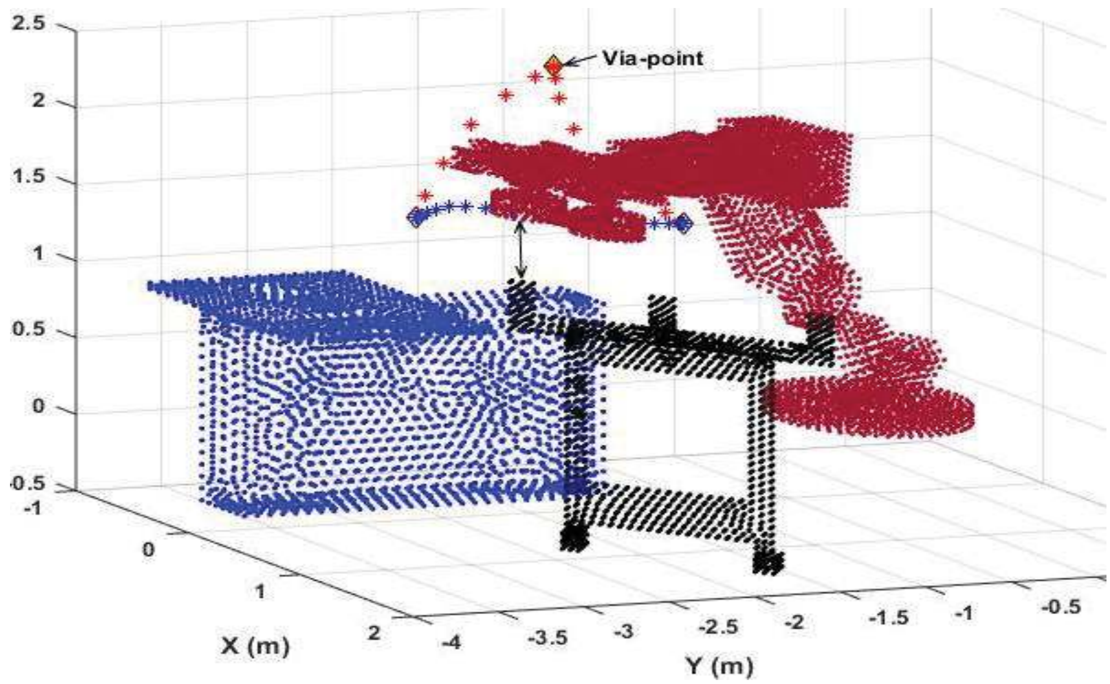

Figure 6.3. Isometric view of the point cloud map of robotic workcell illustrating the fifth order optimal and original trajectory profile

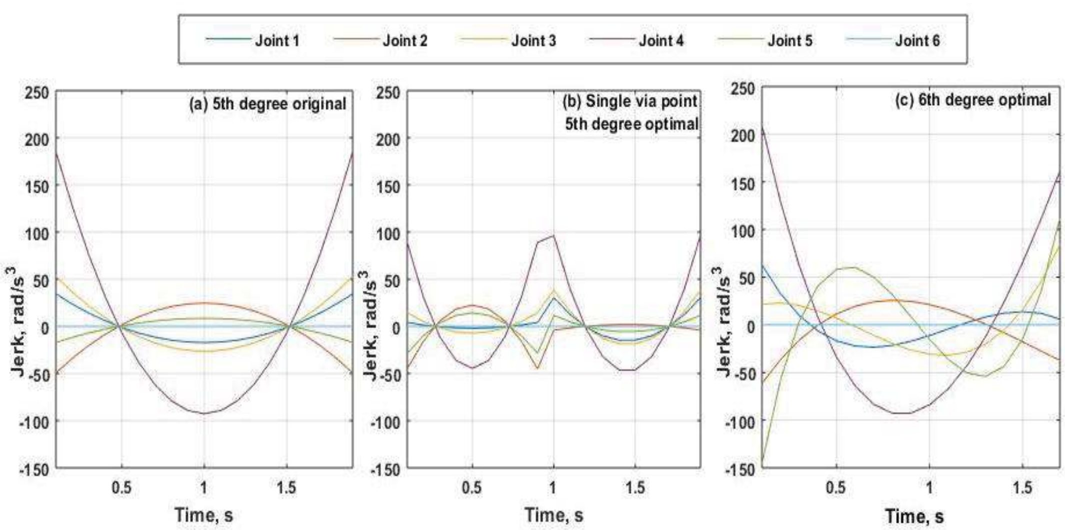

Figure 6.4. Third order derivative of the interpolated curve in joint space for (a) original fifth-degree polynomial, (b) fifth-degree polynomial single via-point optimal and (c) optimal sixth-degree polynomial

Also, from the jerk values given in Table 6.2, it has been found that the overall jerk of optimal fifth degree trajectory is reduced and it is 39.78 % lesser than the original trajectory. The optimal via-point not only optimizes the jerk but also reduces the length of the trajectory traveled by the robot by 36%. Moreover, to remove the sharp discontinuity at the via-point, the single sixth degree polynomial profile is used in place of two fifth-degree polynomials for trajectory interpolation with a via-point.

Table 6.2. Optimal results

Cases		*α*	*Jerk Value, x 10^3 (rad/s^3)*	*Execution time (s)*	*Optimal via-points*
Single via-point	5^{th} degree polynomial	0.25	1.786	2.00	(0.122, -1.751, 2.188)
		0.50	1.587	1.80	(-0.226, -1.712, 2.336)
		0.75	3.0424	3.90	(0.394, -2.113, 1.705)
	6^{th} degree polynomial	0.25	5.436	1.85	(0.220, -2.059, 1.868)
		0.50	3.211	1.6	(0.391, -2.131, 1.691)
		0.75	7.374	3.32	(0.210, -1.644, 2.358)

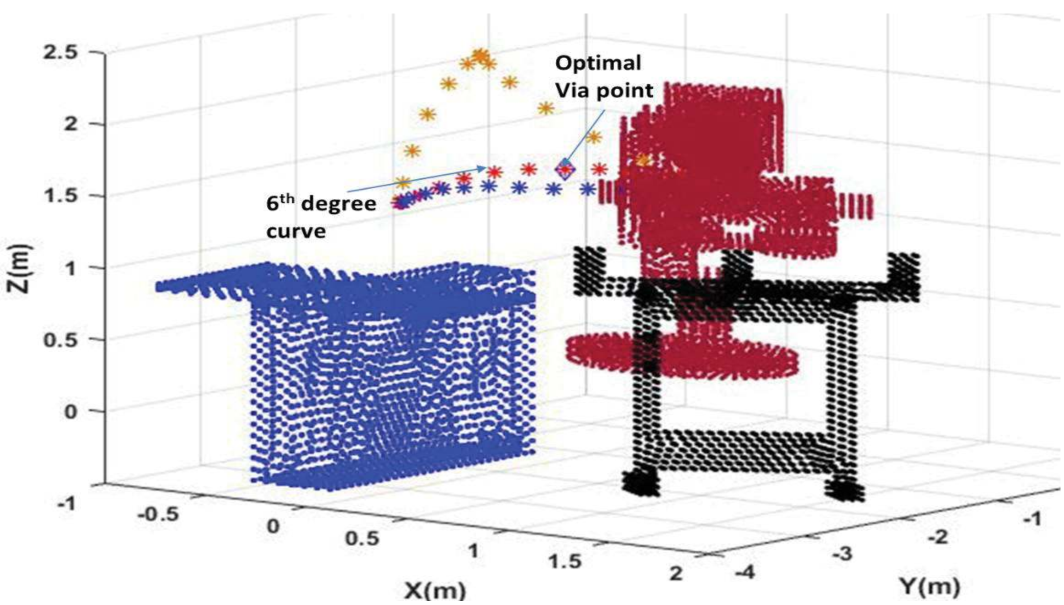

Figure 6.5. Point cloud maps of the sixth-degree single via-point trajectory interpolation with via-point

The trajectory profile of the sixth-degree polynomial interpolation is shown in Fig. 6.5 along with two previously generated trajectories. The new sixth-degree profile maintains the continuity at via-point which gives an advantage of saving the loss of time at via-point due to zero velocity at the intersection of two fifth degree trajectory profiles. Nearly, 11.1 % saving in execution time has been achieved, however, the overall jerk value becomes double. The possible reason for jerk rise is the use of one-degree higher profile with comparatively higher velocity. Also, the 24.48% increment in the length of the path traveled is observed. Therefore, at the expense of jerk and path traveled, continuous motion at the via-point is secured.

The trajectory planning by present methodology using the fifth-degree polynomial profile with single via-point reduces jerk, but the distance traveled by the manipulator has also increased. Nonetheless, using sixth-degree trajectory interpolation, the increment in the jerk values and the length of trajectory traveled by the robot was limiting its advantage of having continuous motion at via-point. Thus, the drawbacks occurring during trajectory planning using single via-point approach are discontinuity and higher jerk during motion. In this situation, it has been proposed to increase the number of via-points between machines and use splines for trajectory planning. Thus, next sub-section discusses the multi-via-point trajectory interpolation and searches for continuous and minimum jerk profile.

6.4.2. Multi via-point solution

In this section, the effect of increasing the number of via-points in trajectory planning problem has been studied. The fifth degree B-spline profile has been used for obtaining smooth and continuous trajectory. Table 6.3 shows the results obtained by solving the trajectory planning problem for three different weightage values. Total four via points have been taken because the B-Spline interpolating has fifth degree polynomial thus minimum number of control points necessary for interpolation is four. These results have been generated by running several tests of genetic algorithm in Matlab using a personal computer with problem formulated in the previous section.

The optimization algorithm has been run on an Intel Xenon E7-8870 server several times, and the result obtained are shown in the Table 6.3. Three cases are evaluated individually, and computational load is high due to large number of variables. The weightage factor α is selected according to weightage factors used in various literature (Gasparetto and Zanotto, 2007). The table shows the optimal locations of the via points, execution time and net jerk value of all joints during motion. The data given in table is used to plot point cloud maps of the trajectory planning by multi via-points.

Table 6.3. Multi via-point optimal solutions

Case	*Weightage factor, α*	*Jerk Value, $x10^3$ (rad/s^3)*	*Execution time (s)*	*Optimal via-points*
Multi via-point 5^{th} degree B-spline	0.25	4.781	2.77	[-0.231, -1.843, 2.372; 0.211, -1.843, 2.317; 0.572, -1.757, 2.259;0.555, -1.524, 2.376]
	0.5	3.192	2.46	[-0.132, -2.236, 1.936; 0.136, -1.985, 2.212; 0.569, -1.594, 2.38; 0.628, -1.704, 2.307]
	0.75	1.632	2.11	[-0.236, -1.947, 2.279; 0.044, -2.056, 2.151; 0.398, -1.81, 2.288; 0.597, -1.636, 2.346]

The optimal fifth degree B-Spline curve is continuous and reduces the overall jerk by 69.9% with respect to the continuous sixth-degree polynomial trajectory. The sixth-degree curve was also smooth, but the jerk is very high, while the B-spline curve has lower jerk and continuous motion both.

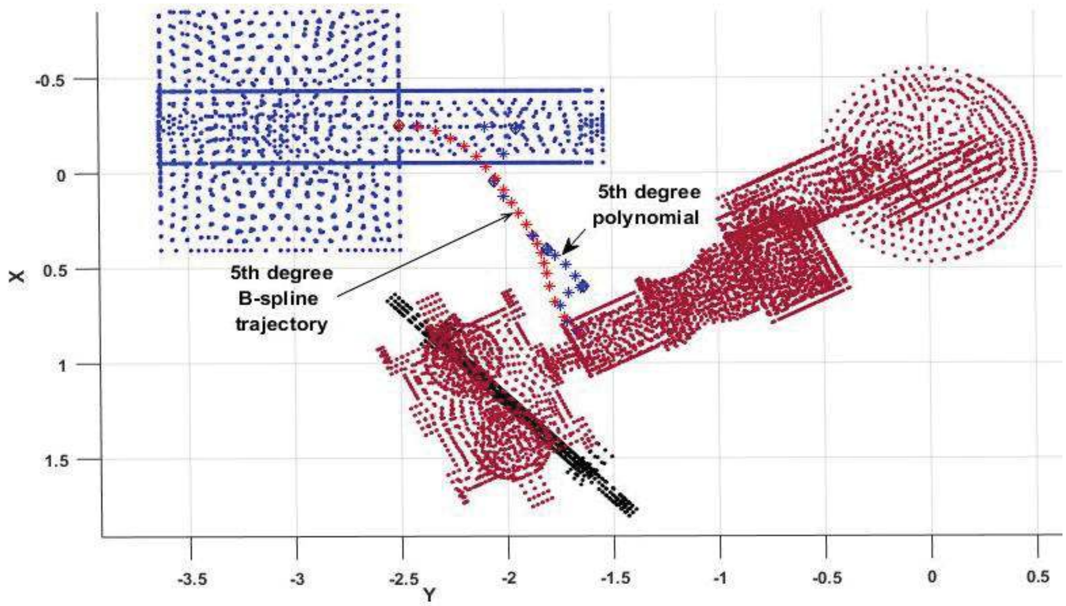

Figure 6.6. Point cloud map of the fifth degree B-Spline trajectory profile passing through four via point

Fig. 6.6 shows the point cloud maps of the solutions corresponding to the weightage factor of 0.75. The B-Spline trajectory indicated in the figure as red points and the robot lead point is tracing these points without colliding with any machine in the workcell. For the comparative purpose, the figure also shows the fifth-degree polynomial profile as blue points passing through all the optimal via points. Fig. 6.7 shows the jerk profile of the fifth-degree polynomial trajectory and the fifth degree B-Spline trajectory for all six joints of the robot. As it is evident from Fig. 6.7a the curve contains large number of hills and valley which have high amount of jerk at joint 4. While in fig. 6.7b all joint has continuous jerk profile with considerable lower jerk values than the polynomial trajectory. B-Spline trajectory profile is governed by the knot vector which controls the curve according to the control points' location in the coordinate space. The locations of control points are indicated in Fig. 6.8 and among them are the optimal via points.

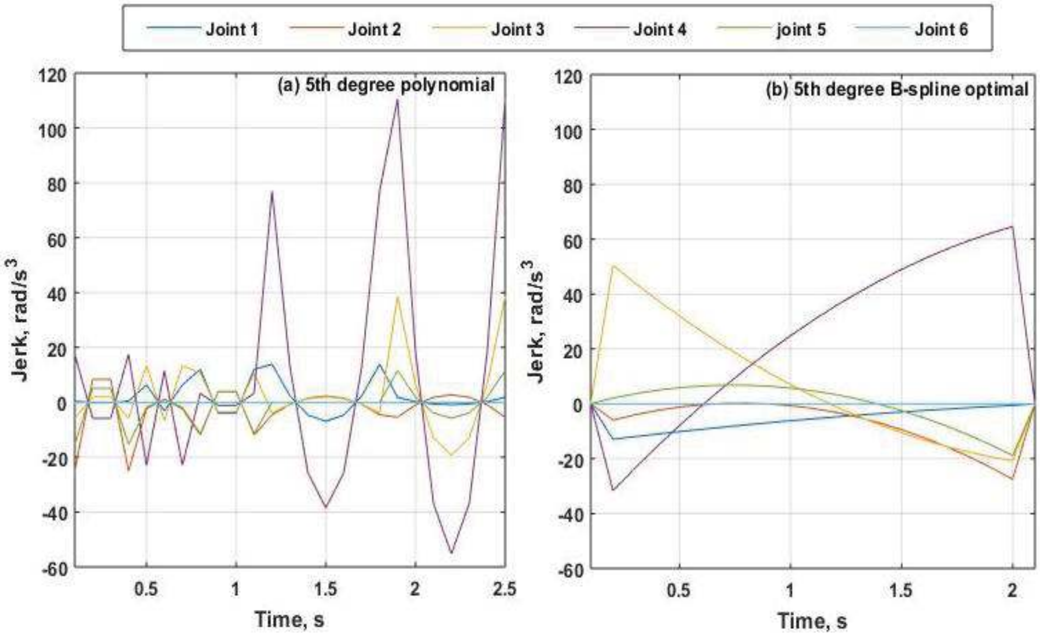

Figure 6.7. Third order derivative of (a) fifth-degree polynomial, (b) optimal fifth degree B-spline interpolated curves in joint space

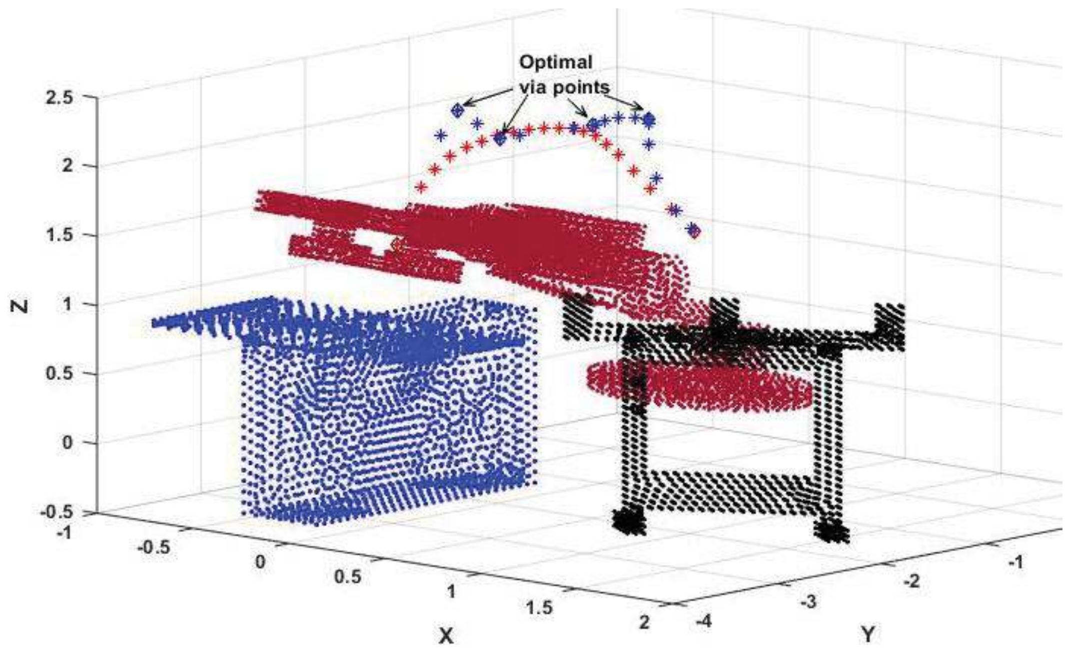

Figure 6.8. Point cloud map showing the isometric view of the fifth degree B-spline profile and fifth degree polynomial profile passing through optimal via-points

Fig. 6.8 shows four optimal via-points, located at a distance from B-spline interpolated trajectory points (red), while the fifth-degree polynomial interpolated curve passes through them. Therefore, for the same degree of interpolation, the jerk profile has been analyzed. Also, the B-spline curve successfully avoids the point of collision, and the tool is considerably distant from the machine, M2. From Figs. 6.6 & 6.8 it is understood that the curve bend at the collision area and further bend near the end to minimize the path traveled. Therefore, 2.9 % decrement in the path length from the continuous sixth-degree trajectory has obtained.

Thus, it has observed from the data given in Table 6.3 and by analyzing all point cloud maps generated from the trajectory interpolation, the best possible trajectory for the point cloud constrained problem is fifth degree B-spline at $\alpha = 0.75$, with execution time of 2.11 sec. The point cloud simulation approach thus succeeds in simulating the realistic trajectory planning situations and improved the solution by observing the point cloud maps.

6.5 Concluding Remarks

This chapter proposes a trajectory planning methodology based on point cloud simulation approach which imposes constraints for inducing the real environment collision checking and producing implementable solutions. This approach also introduces the trajectory planning by optimizing the position of via-points between given endpoints. Three different interpolation curves have been used with different number of via-points. For single via-point, quintic profiles were used, having a discontinuity at the via-point. The results were improved by using 6th-degree polynomial profile continuous at via-points. However, the increment in jerk has been observed by using higher degree polynomial.

Therefore, to achieve lower jerk values and continuous motion throughout the trajectory quintic B-spline profile has been adopted for which multi via-points have been taken. Genetic algorithm has been used for searching the optimal via-points within the configuration space of the robot. The proposed algorithm validates through an industrial case study, using minimum jerk and execution time as the optimality criteria. The total jerk in the B-spline trajectory is much lesser in comparison with the other profiles. Future work will be the application of point cloud approach to trajectory planning of the multi-robot workcell. The research work described in this chapter adds to the vast knowledge of trajectory planning approaches developed previously by various researchers.

The next chapter presents some conclusion drawn on the basis of the complete research work done in the field of robotics regarding layout design and trajectory planning of robotic workcell.