

Chapter 3

POINT CLOUD SIMULATION APPROACH

The proposed approach for modeling and simulation of the robotic workcell developed by using computer-aided design based elemental discretization method has been presented in this chapter. In this approach, the real world moving and nonmoving machines/equipment were transformed into the virtual point cloud models, whose coordinate data has been utilized for planning and collision avoidance purposes. The point cloud-based design and planning approach has been used to generate practical solution which has been validated by the industrial case study on a robotic workcell.

The introduction of the proposed approach has been presented in section 3.1, the detailed modeling of moving objects robots and nonmoving objects has been discussed in section 3.2. The working algorithms and flowchart have been presented in section 3.3, and the kinematic calibration of the moving object has been evaluated in section 3.4. The industrial case study performed to validate the proposed approach was briefed in section 3.5, and finally, the chapter is summarized in section 3.6.

3.1. Introduction

The simulation of workcell is a tedious task involving various aspects to be taken care such as collision avoidance, robot motion sequence, and optimal location searching. These problems require precise planning and programming that demands realistic modeling of workcell objects for simulation. The computer-aided simulation is the suitable solution for efficient modeling by which any changes in system can be implemented conveniently.

The point cloud simulation approach developed for design and planning of the robotic workcell utilizes the point cloud data of machines and robots. This approach searches the optimal location of machines in the workcell under actual working environment in which point cloud data acts as constraint. The point cloud data of machines and robots have been developed by programming in Matlab, using PDE toolbox and the given discretization format. The discretization or meshing of the given object requires stereolithographic data generated from the CAD drawing files developed in AutoCAD.

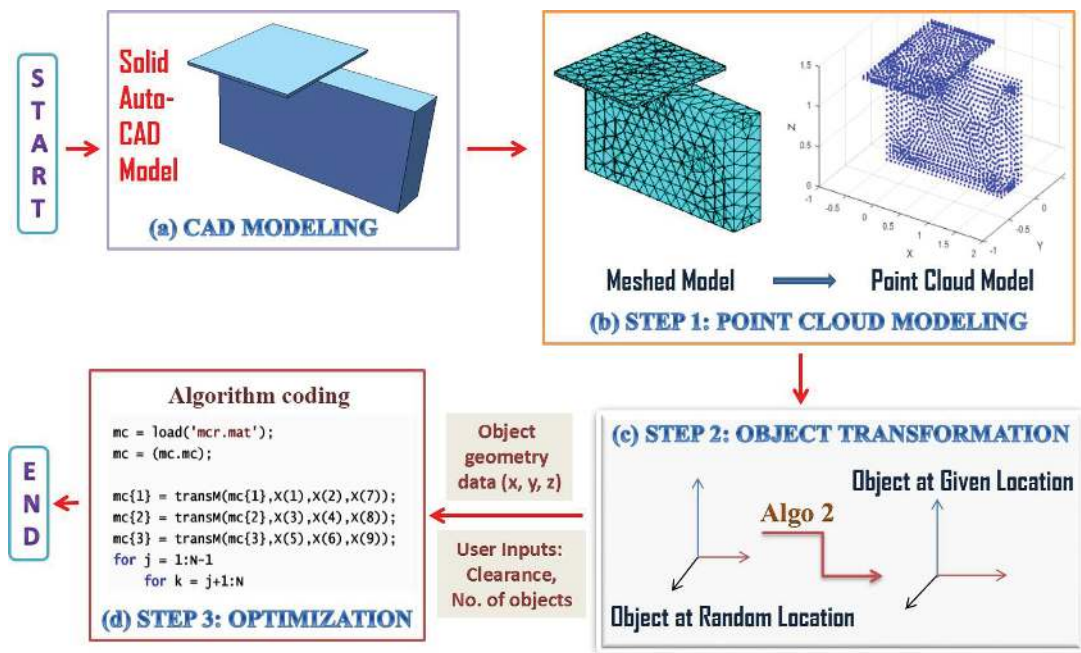


Figure 3.1. A schematic diagram of the proposed approach for design and planning of robotic workcell

Fig. 3.1 presents an overview of the proposed approach and it illustrates as (a) drafting of CAD models of robotic workcell objects (b) generating point cloud modeling from CAD model (c) transformation of randomly located point cloud model at the desired location and (d) finally, searching the optimal locations of point cloud models according to desired objective under imposed constraints for the entire workcell. The point cloud

simulation of robotic workcell assumes that all objects have been located about a fixed point in the coordinate frame of reference generally assigned as the base of the robot. It ensures relative measurement system within the workcell and gives the robot centered workcell design. Similarly, for more than one robot in the workcell, one robot is fixed while other robots and machines act like variables.

Thus, the proposed approach has following three major steps:

- ⇒ Step 1 - Generation of point cloud data from CAD model of robots and machines.
- ⇒ Step 2 - Transformation of point cloud model of machines and robots to the given point in the coordinate frame.
- ⇒ Step 3 - Searching for the optimal configuration of the robots and machines satisfying the formulated objectives and constraints.

In step 1, the simplified models of the real world robotic workcell objects have prepared by measuring their dimensions manually, thereby retaining actual volume occupied by them in the workcell. After that, CAD model has been exported in the stereolithographic file format, i.e., the working format to generate the point cloud model in Matlab. It is evident from the Fig. 3.1 (a) & (b) that the drafted model is a simplified version of the complex working object from which the external parts are either merged into the body or removed without changing the overall geometry.

The next step is the transformation of point clouds to the desired location in the coordinate space to construct the rough model of the complete robotic workcell. For the transformation process, several algorithms have been developed considering the motion and kinodynamic parameters of the objects. In the last step, the locations of the transformed point cloud models are optimized, and the entire robotic workcell is redesigned according to the formulated objectives.

The robotic workcell contains various kinds of machines which differ in motion and topology, due to which the modeling of machines also varies accordingly. The machines having relative moving parts in their topology are disintegrating and each moving part is drafted separately, while the machines having single topological structure are drafted in single file. For instance, the robot has relative motion between its links, therefore, they are drafted in separate files which are later combined together and calibrated during point cloud modeling in Matlab taking information from the D-H parameters.

3.2. Point Cloud Modeling

The first step of the approach is to develop the point cloud model of objects in the robotic workcell. The point cloud model of an object is a discrete mesh model containing the locations of nodal points generated by Matlab algorithms. The purpose of point cloud modeling is to give a realistic environment for the simulation of robotic workcell and generate practical solutions. The partial differential equation toolbox of Matlab provides several functions that can be used for creating the meshed model from the CAD file. Quadratic order 3D finite elements of tetrahedral shape have been used for the discretization. The mesh model is a combination of nodes stored in a $3 \times p$ matrix, where p is the number of nodes. The number of nodes can be controlled by altering the edge length of the 3D element. Small element increases the number of nodes in the mesh which is more accurate but simultaneously increases the computational load. The mesh model is then transformed to the point cloud model by extracting the coordinates of the nodes in the Cartesian coordinate frame.

The point cloud modeling of objects can be classified into two categories as shown in Fig. 3.2. First, the nonmoving objects such as machines, job tables, stands, holders, fluid tanks, and conveyors. These objects have their base fixed to the ground, walls or ceiling of

the workcell. The second type is the moving objects such as robots, AGVs, etc., which have moving parts that are involved in the transportation of workpiece in the workcell from one place to another. The modeling procedure for each type of object is different. However, the basis of both modeling procedures remains the same.

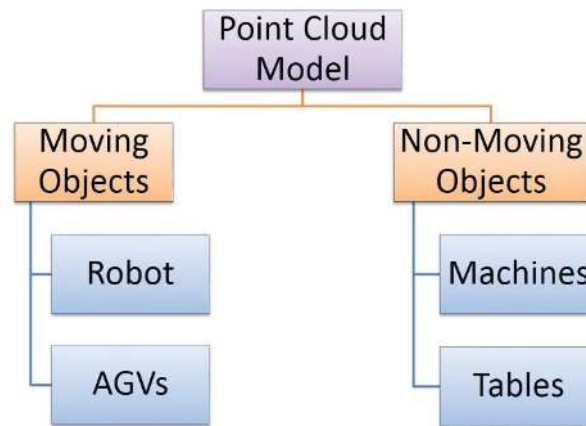


Figure 3.2. The classification of the point cloud modeling of workcell objects

3.2.1. Modeling of Non-Moving Objects

The modeling procedure for the non-moving objects assume them as a single solid unit specifically without any relative moving parts. The point cloud model of these objects has been prepared in single file, and the kinematic relation are not required for simulation. This type of object is located and traversed by its job point. The job point of an object is a point in the point cloud data matrix of an object.

This point is selected while generating the point cloud data and is chosen manually through point cloud maps of the object in the Cartesian coordinate frame in Matlab. The point selected from the data matrix has set to (0,0,0) and all the point in matrix thus measured from it. A point cloud model of an object traverse from one frame to another according to the change in location of its job point. Each point in the point cloud data change its location relative to change in job point coordinates.

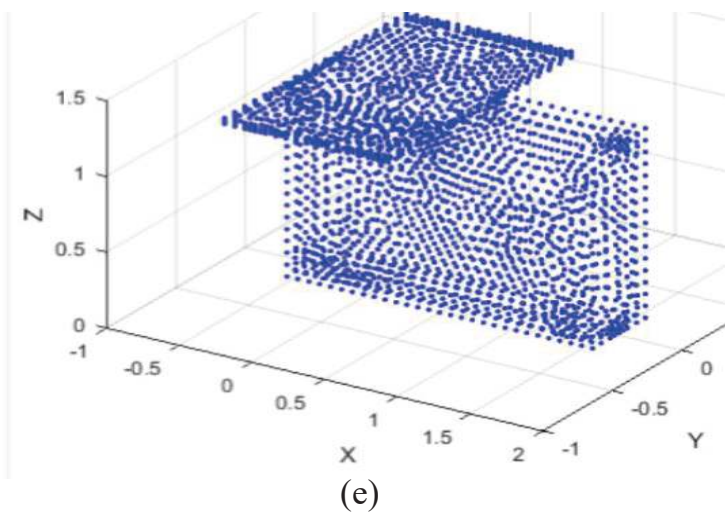
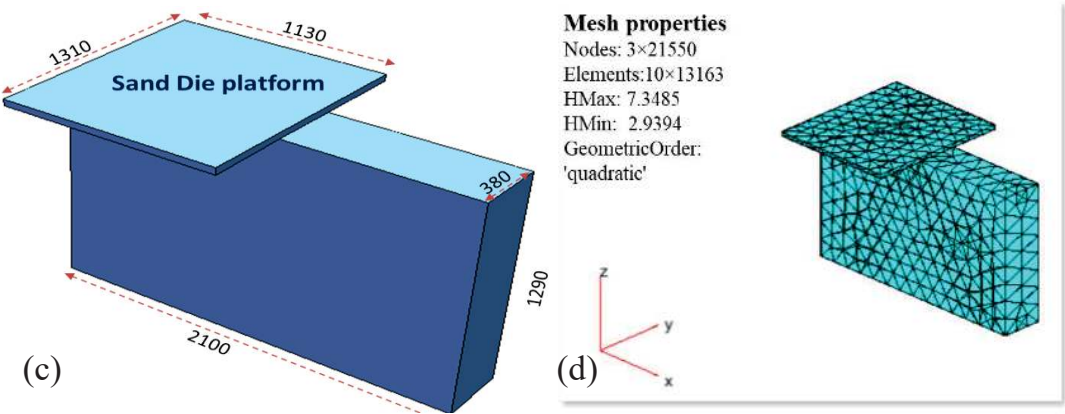
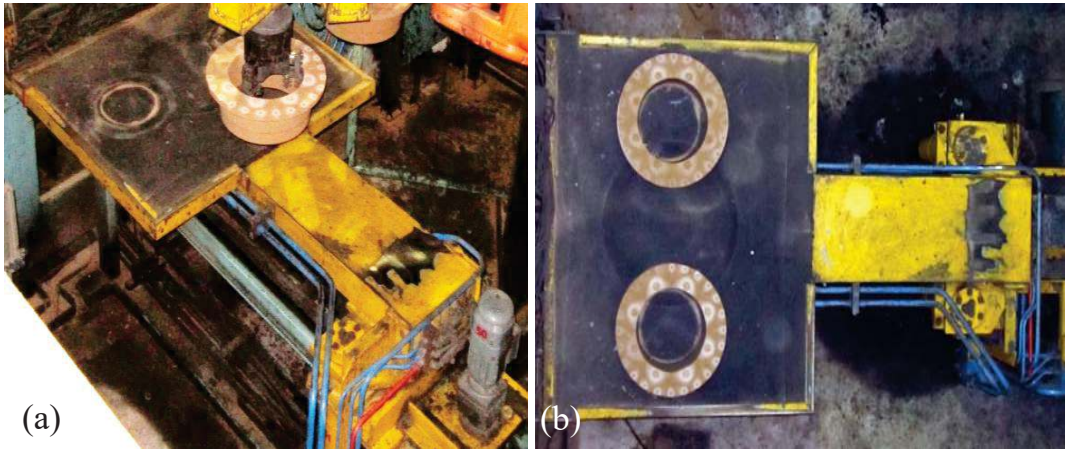


Figure 3.3. Illustrative diagram for the point cloud modeling of non-moving object (a) the side view of an object (b) top view of same object (c) CAD model (d) meshed model (e) point cloud model

Fig. 3.3 shows the diagrams illustrating the modeling of the real world non-moving object into the simulated point cloud model in Cartesian coordinate frame. The step by step approach for the conversion is as follows. First the image of real world object is captured as shown in Fig. 3.3 (a) & (b) and the dimension of the object such as height, breath, length are also measured through instrument. Second step is to draft the object into suitable CAD modeling software using actual measured dimensions as shown in Fig. 3.3 (c). The height of each CAD model was drafted along the positive z-axis of the coordinate system, measuring from the base plane.

Further, the next step is to convert the CAD model into the meshed structure by using 3D triangulation method in Matlab as shown in Fig. 3.3 (d). The CAD file in *stl* format acts as an input for creating the meshed geometry made up of n number of elements of suitable type and size. The discretization process uses a 3D element of tetrahedral shape which is of two kinds - linear and quadratic based on the geometric order. The linear 3D element has nodal points at the corners of edges only, while the quadratic element has an extra point at the center of each edge too. The maximum edge length (H_{max}) and the minimum edge length (H_{min}) determine the size of an element. The values of H_{max} and H_{min} plays a significant role in manipulating the size of point cloud data.

Finally, the information from the meshed structure is extracted in the data matrix and plotted in the form of point wise structure as shown in Fig. 3.3 (e). The required PC model can either be surface, generated from the shell CAD model or a solid generated from the solid CAD model, depending on the computational capability of the system. Note, the point cloud modeling of nonmoving object as explained in this section has been utilized to model the moving object as explained in next section.

3.2.2. Modeling of Moving Objects

The second type of modeling procedure is for objects having relative moving parts such as a robot having moving links, which is similar to the procedure developed for nonmoving objects without any relative moving parts. All moving parts of object have been modeled separately and finally combined according to the kinematic relationship between them. Each point cloud model of different parts is then combined and finally complete model has been calibrated according to the D-H representation. The final model should work like the original working model of the object.

For a robot, links that form its topology and having relative motion among themselves have been modeled in separate files. Fig. 3.4 (a) shows the actual onsite image of a moving object, and Fig. 3.4 (b) shows the snapshot of the AutoCAD drawing window in which the CAD model of robot has been dismantled into three distinct sections. These sections have been modeled in different CAD files which has then converted to its corresponding point cloud file as shown in Fig. 3.4 (c). The reason behind using separate modeling is to correctly simulate the relative motion between moving parts of the object. Thus, the point cloud model, approximately sustains same features as that of the original object without losing its overall geometrical aspect.

In present thesis work, ABB IRB 6640 robot is taken for demonstrating the modeling of moving object. It is a six degree-of-freedom arm type manipulator and its modeling can be divided into two halves. The first half contains three larger links, which cover the entire robot workspace, while the second half is the wrist portion with another three links, which helps in the precise location of tool centre point (Mata and Tubaileh, 1998). The above process shows the modeling of its three links and rest three links do not contribute much in the topography of the robotic arm.

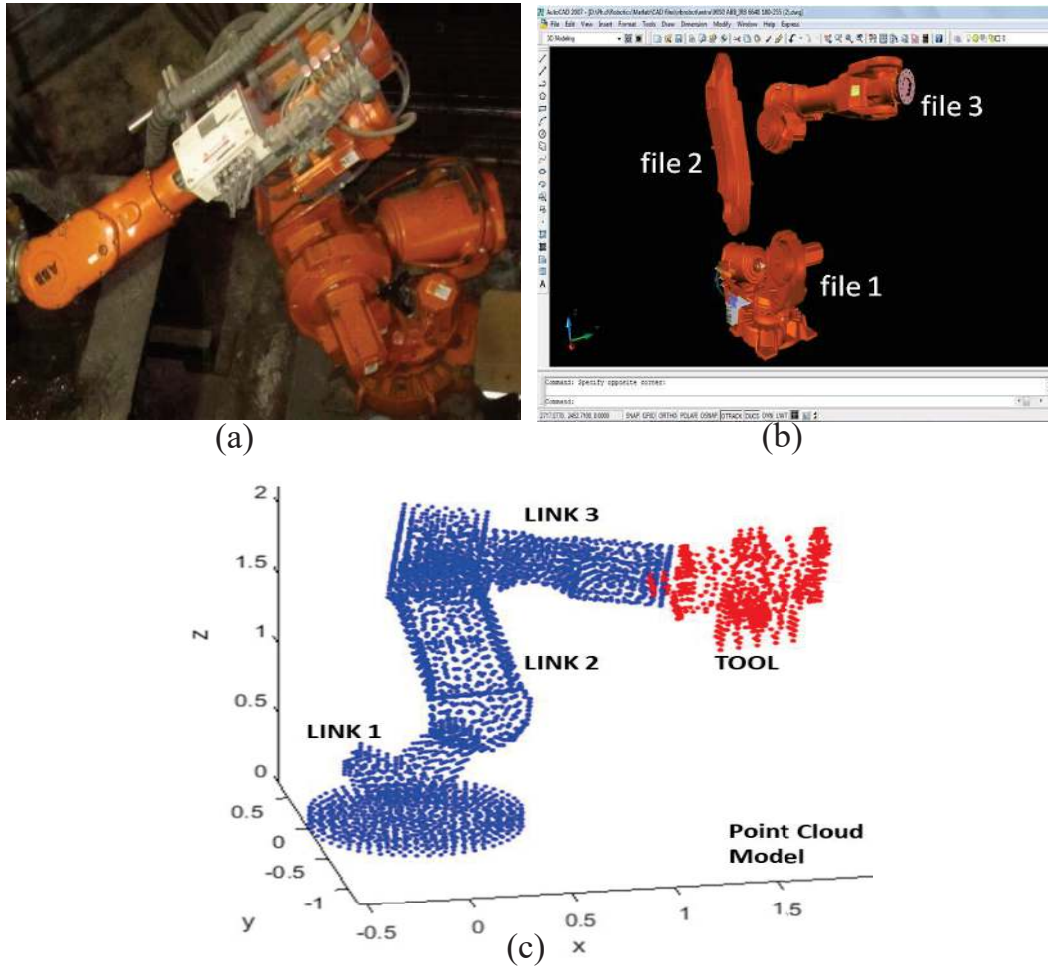


Figure 3.4. (a) Image of the real robot installed in a workcell, (b) snapshot of the AutoCAD file disintegrating robot model and (c) the point cloud model of the robot

However, the modeling procedure for wrist is same, having three degrees of freedom in roll-pitch-yaw motion. The tool attached with the robot is also modeled and it is a considered here as a non-moving without any moving parts. Since, there is no kinematic relation for the tool, thus for simplicity, its motion is simulated with the third link by gathering information from D-H convention. Finally, there will be three moving links; link 1 formed by file 1, rotates around the z-axis perpendicular to the ground, link 2 by file 2, rotates around y-axis parallel to ground and link 3 by file 3 contains wrist and tool that rotates similar to link 2.

3.3. Working Algorithms

During an operation, the robot approaches a machine at some point for loading and unloading the job, avoiding collision with another machine in the workcell. This point is referred as job point as described in section 3.2.1 and used for transformation of machine in coordinate system. It is selected while modeling a machine knowing the appropriate location of working site at which the robot end-effector approaches machine comfortably. Fig. 3.5 (a) shows a CAD model of machine in AutoCAD along with the job point which is at a certain distance from the origin C .

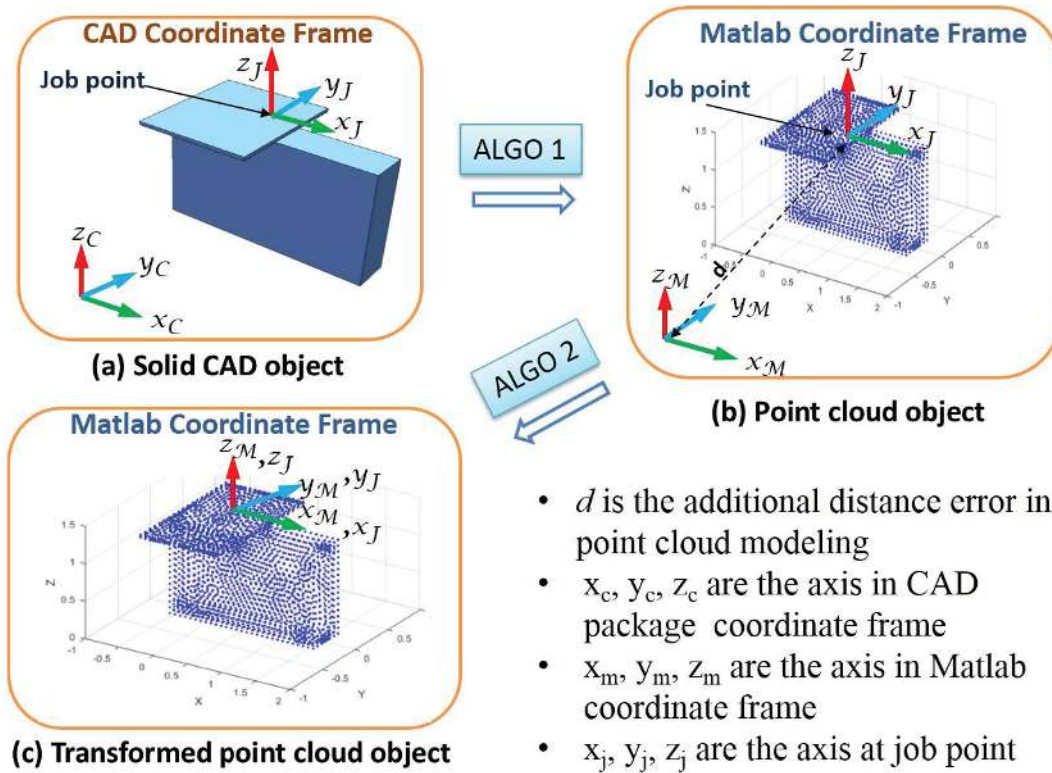


Figure 3.5. Schematic diagram for modeling and transformation of an object in various coordinate frames

This model is then discretized into point cloud model using algorithm 1. Fig. 3.6 shows the samples codes in Matlab to extract the coordinate values from the discrete mesh

Algorithm 3: ROBOT WORKCELL OPTIMISATION

Input: Point cloud data of machines and robots

Output: Optimal configuration of robot workcell

Begin

Given N machines, M robots & initial feasible solution X

Get transformed machines & robots matrices, **Algorithm 1 & 2**

Given job point & constraints for machines

Initialize *optimisation* function

For N machines **Do**

Check for workspace limits of robots

If (joint angle \neq Nan) **Then**

Calculate intercepts

Else If (intercept ≤ 0) **Then**

Compute objective function value

Else

return

End If

End For

Generate optimal solution for machines & robots

End

Figure 3.8. Algorithm to generate the optimal location of the variables

Fig. 3.8 shows algorithm 3, for the optimization of location of the point cloud models of entire workcell. The input data of the algorithm is the point cloud model of the machines and robots along with the job point data for reference and their initial localization. This algorithm can also be used for searching optimal location of any point in the coordinate frame serving as variable for the formulated problem.

This algorithm utilizes global search techniques such as simulated annealing and genetic algorithm for iteration of the optimal solution subjected to given constraints. The entire working structure of the point cloud simulation approach is constructed on these three algorithm whose flowchart is shown in Fig. 3.9. The flowchart illustrates the processing behind the point cloud simulation approach shown in Fig. 3.1.

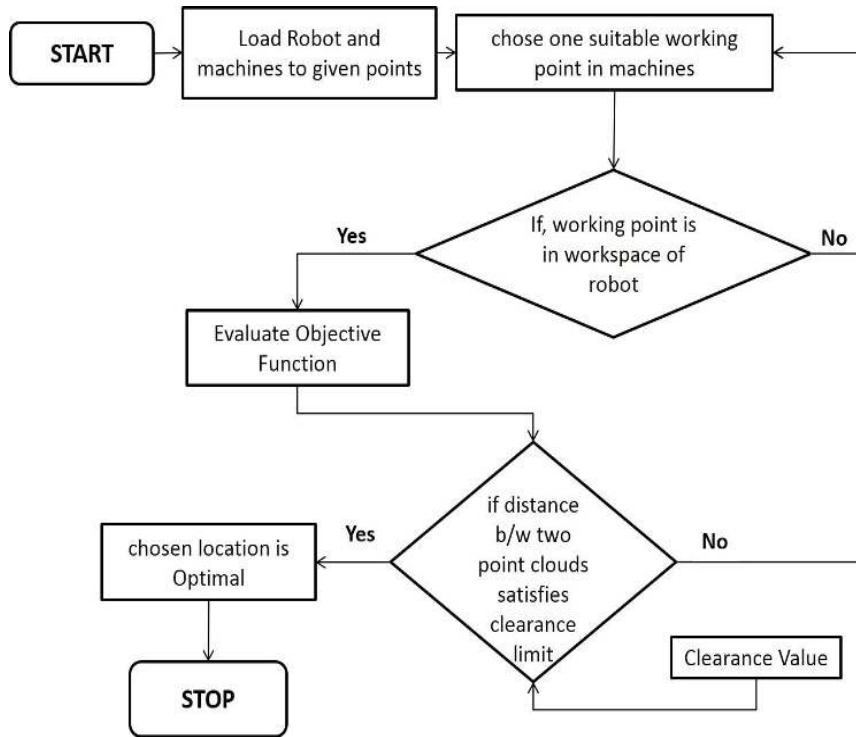


Figure 3.9. Flowchart of the point cloud simulation approach

3.4. Kinematic Calibration of Moving Object

The purpose of creating the point cloud model is to simulate the realistic collision situation between the robot and machines. The kinematic calibration of the generated point cloud model of the moving objects such as robot will ensure that its end-effector can reach the desired point within configuration space. The mechanism for calibration of the point cloud model of robot is that, the motion of each link has to synchronize with the inverse kinematics of the real robot. The RVC toolbox (Corke, 1996) was used to calculate the inverse kinematic values of the robot configuration based on the Denavit-Hartenberg Convention. Table 3.1 presents the D-H parameters of ABB IRB 6640 robot.

The accuracy of the simulated point cloud model to follow the trajectory points as traced by the real robot has been checked and calibrated by using kinematic model developed in RVC toolbox developed by Peter I. Corke. This test also confirms the relative

movement of links in the point cloud model. In the calibration process, the given point in the Cartesian space must be approached in a similar pose by developed point cloud model as approached by the line model of the RVC toolbox, joint by joint

Table 3.1. Denavit-Hartenberg parameters of ABB IRB 6640

Link i	Joint angle θ_i (rad)	Joint offset d_i (m)	Link length a_i (m)	Link twist α_i (rad)	Joint angle Range (rad)
1	θ_1	0.780	0.320	$-\pi/2$	[2.96 -2.96]
2	θ_2	0	1.075	0	[1.48 -1.13]
3	θ_3	0	0.200	$-\pi/2$	[1.22 -3.14]
4	θ_4	1.142	0	$-\pi/2$	[5.23 -5.23]
5	θ_5	0	0	$\pi/2$	[2.09 -2.09]
6	θ_6	0.200	0	0	[6.28 -6.28]

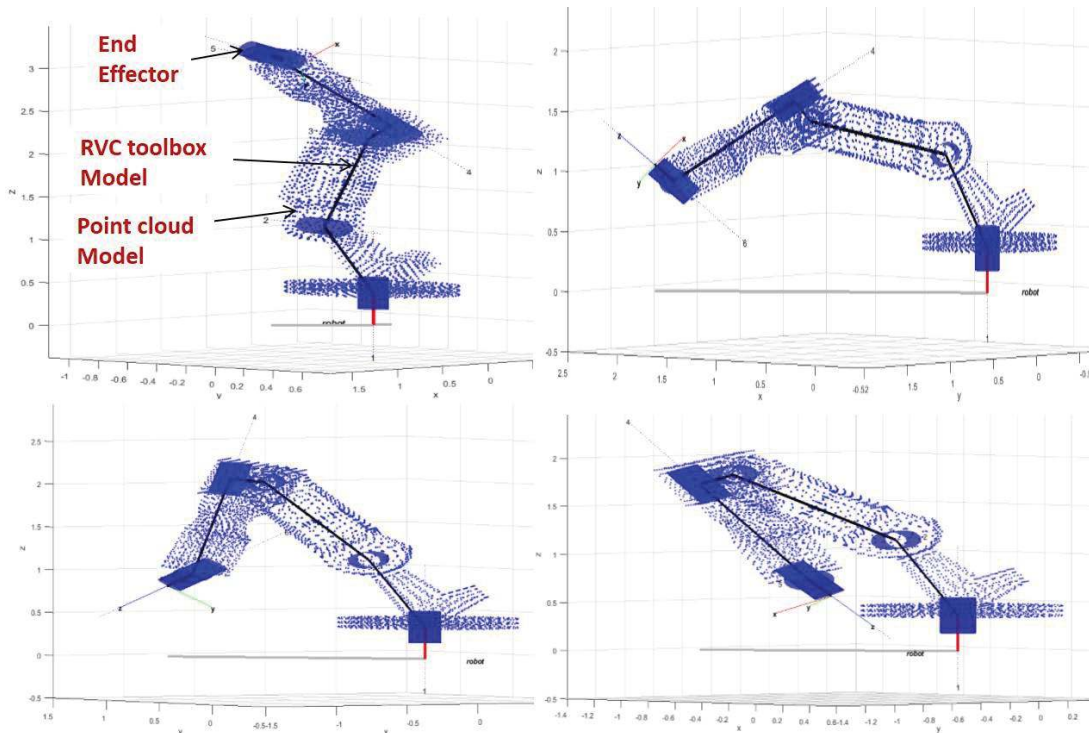


Figure 3.10. Different configurations of RVC model (black) and point cloud model (blue) of ABB IRB 6640 robot

Fig. 3.10 shows the result of the calibration test in various positions of simulated models of ABB robot. The RVC toolbox model is coinciding with the point cloud model at each and every joint position. Also, the end-effector of both model coincides with each other and thus reaches accurately at the given point in the coordinate space. This test affirms the developed point cloud model and ensures its reachability at all machines accurately during operation.

Thus, for ensuring proper working of the point cloud model calibration test by using the RVC toolbox model has been successfully conducted. In this test, the point cloud model of the robot has replicated same angular configuration as produced by the RVC model. The point cloud model produces joint motion using Algorithm 2, and the RVC model produces joint motion using the inverse kinematics of a real robot.

3.5. Industrial Case Study

For validating the developed simulation approach, a case study on a foundry robotic workcell has been conducted at M/s Electrosteel Castings Ltd, located in Dhanbad (India) as shown in Fig. 3.11. This robotic workcell produces sand cores for manufacturing cast iron pipes.



Figure 3.11. Robotic Workcell in M/s Electrosteel Castings Ltd., Dhanbad, India

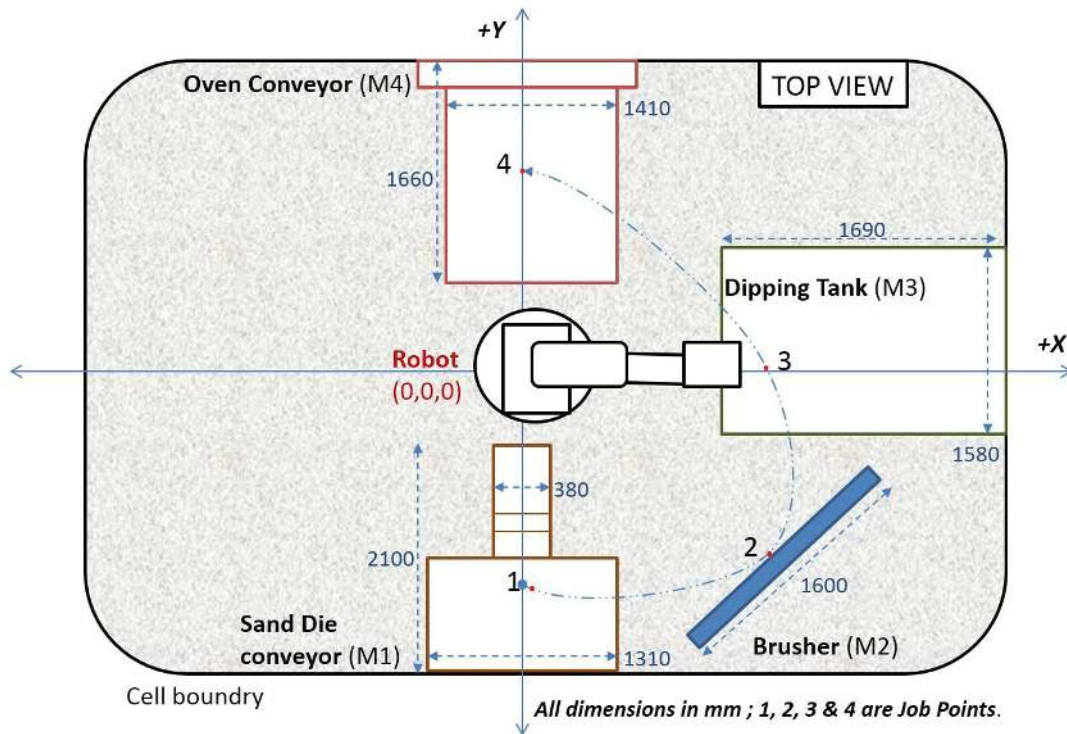


Figure 3.12. Depicted diagram of the industrial robotic workcell under study

Fig. 3.12 shows the schematic diagram of the workcell which contains four machines located in an arc type layout around ABB IRB 6640 robot manipulator arm supplied by ABB, Sweden. The task of the robotic manipulator is to pick sand cores from Sand die conveyor M1 and place them on Oven conveyor M4 via Brusher M2 and Dipping tank M3 as shown in Fig. 3.12. There are total 22 sequence points traversed by the robot end-effector while accomplishing given sequence of task in a cycle. There are a few points at which end-effector comes in contact with the machines. These points are the working sites of each machine and lie within the configuration space of the robot as indicated by 1-2-3-4 in Fig. 3.12. Therefore, for the initial feasible input required to start iteration, one point corresponding to each machine has been selected. Moreover, the movement of robot end-effector is a point-to-point motion from one machine to another, and it approaches a machine once in a cycle.

3.6. Concluding Remarks

The simulation of the manufacturing system is gaining pace, and various number of novel simulation approaches are producing promising solutions to the problems of the cellular manufacturing system. This chapter has provided the details of the new approach for modeling and simulation of the robotic workcell based on the point cloud modeling of its working components.

The beneficial outcome of the point cloud simulation approach are as follows:

- ⇒ This approach is undertaking the actual dimension for the modeling of objects. Thus the approach is working under realistic environment which will generate practically implementable solutions.
- ⇒ This approach precisely conducts the collision detection, which takes the constrain due to the geometry of the robot, tools, machine and other equipment.

However, this approach has some limitations. First is the modeling of objects is possible for the continuous surface, that can be overcome by using standard drafting packages having multifunction to produce continuous model of object. This problem occurs during generation of *.stl* file format. Second, the increase in the number of nodes in the meshed model increase the size of data matrices which increase the computational load. To overcome the problem of computational load high end processors can be used.

In the following Chapter 4, the first application of the proposed approach has been demonstrated for the case study on a robotic workcell. In this application, the layout design problem of the robotic workcell has been addressed, and optimal layout design was professionally presented.