Chapter 2

LITERATURE REVIEW

Robots are the integral part of industries and society these days, and behind such remarkable position, lies a profound history and substantial research work that give this field its present status. The persons connected with robotics have vast opportunities, novel research ideas, expanding market, and blooming future. This scenario has made robotics, an active area, and next generation is looking towards it as the promising and stable carrier opportunity (Nocedal and Wright, 2006). Robotics comprises several disciplines of engineering and management, which have significant contribution in the development of this field. The knowledge of robot hardware such as body, motors, sensors, controllers, and software such as programming, planning, and organization, have been applied to this field in several ways. The objective of this chapter is to present an overview of the research work conducted by various authors in the field of design and planning of robotic workcell throughout the world in the past few decades. The literature review presented in this chapter mainly concentrates on the state of the art regarding computer integrated manufacturing, modeling and simulation, and the kinodynamic aspect of the robotic manipulator.

Industrial robots are reprogrammable mechanical manipulator capable of performing several operations such as material handling, welding, spraying, assembly, and inspection. Fifty years back these operations were performed by human operators, but after the second industrial revolution, automation was the chief propaganda and demand for higher productivity and efficiency, resulted in the application of robots in several industrial processes (Koren, 1987). Presently, industrial robots are powerful tool for increasing the productivity, reducing inventory and transportation cost, reducing production time, and easing working environment. However, the present position of the robot in the industrial scenario comes through decades of historical evolution.

2.1. Background

Beginning from the dramatic origin of the robot, a play titled 'Rossum's Universal Robots' by a dramatists Karel Chapek, Czech Republic, 1920, in which the concept of robot known as forced labor arises. The word robot was drived from the Czech word *Robota*, which means servitude. Therefore, the machines that are performing the given task automatically are usually designated as the robots, e.g., automated guided vehicle, manipulator arms, self-controlled domestic appliances. Contemplating the popularity of robots and to understand the entire index behind it, the description of historical development is of great importance.

The Humanoid robot *'Sophia'* is the most advanced robot created till now. However, the early foundation of robots had begun in the year 1940, with the project to create a remotely controlled mechanical manipulator arm for handling the radioactive materials in Argonne National Laboratory, USA. Before 1950, robot was a popular term used in cinema, arts, drama, television, and radio for automated machines that can be used in place of human being. In the late 1950s, the first industrial robot was patented by G. C. Devol, which gives rise to the robot as the industrial working agent. Devol's manipulator arm was able to follow a programmed sequence of industrial operations leading to the automation of industrial processes (Fu et al., 1987). Other manipulator arms such as General Electric's Handyman and the General Mills Minotaur I was the early models driven by numeric control, electric and hydraulic power drives.

Further, in 1960s, the advancement of robotic system by automatic control using sensors and feedback instruments was begun. The man and machine project, 1962 was the breakthrough in which vision and sensing devices were attached to robotic manipulator. In

1963, the VERSATRAN, a cylindrical robot was introduced by American Machine and Foundry Company and was installed in Ford Factory, Canton, USA. In 1965, Denavit & Hartenberg developed the convention to model the robot structure. In 1968, Marvin Minsky at the MIT Artificial Intelligence Laboratory developed the octopus-like tentacle arm. In 1969, the first mobile robot named as *Shakey,* for navigation was demonstrated at the Stanford Research Institute. This robot is equipped with cameras, sensors, and microphone that resembles as the first electronic person. At this time, the robots have now entered in the European and Japanese market. Unimate in collaboration with Kawasaki Heavy Industries, work to expand industrial robots to Asian Markets. Later, Hitachi developed the world first vision based fully automatic intelligent robot that assembles objects from given drawing plan.

In early 1970s, drastic rise in application of robots was begun, and many industries started using robots. German company KUKA build the first robotic production line for Daimler Benz, Sindelfingen, in 1971. In this line, the hydraulic actuated robots were used for welding purpose. The 70s was the boom in the robotic industry. Fiat and Nissan motors had installed robot assembly lines for spot welding of vehicles in 1972. In 1973, KUKA robotics developed the first six axes robot having electromechanical drives, known as *Famulus* by Vicarm Inc., USA, which was used for small part assembly operations.

Hitachi used robots for fastening and loosening bolts by using dynamic vision sensors for moving sensors in pipe and pole industry. In the year 1974, key inventions in the field of robotics had occurred all around the world. US Company, Cincinnati Milacron Corporation launched a minicomputer controlled industrial robot known as T3, The Tomorrow Tool. At this time, robotic welding arm started operation in Japan, and the first fully electric microprocessor controlled industrial robot, IRB 6 was developed by ASEA, Sweden. Hitachi developed the first precision insertion control robot, "HI-T-HAND

Expert" having motion clearance of about 10 microns. In 1976, the robots had entered in space and were used in the space exploratory projects such as Viking 1 and 2 by NASA from the Kennedy Space Centre exploring possibilities of life on Mars.

A multirobot cell with multi-cameras was developed by Hitachi (Japan) to assemble vacuum cleaners in 1977. In 1978, the two pioneer robots were developed named as PUMA by UNIMATE/Vicarm, and SCARA by Hiroshi Makino, at University of Yamanashi, Japan. These robots are lightweight assemble robots used in automobile industries for various purposes. In 1979, Dr. Raj Reddy founded the Robotics Institute at the Carnegie Mellon University, which founded the world first doctoral program in robotics and designed the first direct drive arm.

1980s is the era of robotics as a system, to develop manufacturing systems centered on robots. The first machine vision system was developed at the University of Rhode Island for bin picking. General Motors also installed machine vision system *Consight* in their foundry and now robots have started working in hazardous environments. In 1983, Westinghouse proposed flexible automated assembly lines to use robots with vision more comprehensively in positioning, orientation, and inspection. In 1984, Adept Inc., USA, introduced the *AdeptOne* a first direct drive SCARA robot. ABB Sweden developed the fastest assembly robot IRB 1000, which is 50 % faster than previous robotic arms.

In 90s, the quest was to increase productivity, efficiency, and control cost. In 1992, the Demaurex, Switzerland sold the Delta robot to Roland for packaging application. In 1994, Motoman introduced the multi-mobot control system that gives control up to 21 axes. In 1996, KUKA launched the first PC based robot control system which installed online control using teach pendant. Later, due to the advent of internet, KUKA also began remote diagnosis for robots. In 1998, ABB developed the world fastest picking robot, *Flex Picker*.

In the 20th century, the robotics research had become more concise and precise. Robots became famous for space exploration where robot works in fully automated and extremely unknown environment. The improved robot controllers such as NX100 by Motoman, Japan was launched in 2004, which can provide synchronized control between four robots up to 38 axes with a touchscreen display. In 2006, KUKA launched its first lightweight robot of just 16 kg having payload of 7 kg. RobotBar by Motoman, comprises dual arm and single armed robots, was ideal for assembly and machine tending. They also developed super speed arc welding robots in 2007. KUKA developed a heavy-duty long ranged robot with a payload of 1000 kg. In 2008, Fanuc upgrades the payload capacity to 1200 kg. The multi-robot system had expanded to eight robots in 2009 when Yaskawa Motoman introduces control system to synchronize large number of robots. In 2011, humanoids were developed such as Robonaut R2B, which is the first robot in the International Space Station. In 2015, Hanson Robotics, Hong Kong launched *Sophia* which is the most advanced artificial humanoid ever developed.

The field of robotics is a vast ocean of opportunities. The research scope and applications of robot are endless, and several new applications and approaches are coming up to solve complex real-life problems. Presently, more than 1.8 million robots are working in the industries all over the world. Industrial robots are presently engaged in more than 21 different fields, and hundreds of potential fields are waiting in the future (IFR World Robotics, 2016).

2.2. Modeling of Robot

The operating performance of the workcell is generally affected by the shape, size, weight, and structure of the robot. Indeed, there are wide varieties of robots for only welding operation. Modeling of the robot is the mathematical formulation of complex kino-dynamic relationship among various parameters of the robot. During operation, the angular motion, velocity, acceleration, jerk, torque, and inertia are to be determined for the desired task. During operation, these parameters influence each other and affect the performance of the robot. Therefore, industrial robots need an accurate modeling approach for which the design complexity is a challenging problem.

2.2.1. Robot Design and Transformation

The mechanical structure of a robot generally constitutes the links and joints. The industrial robotic manipulator, fixed to the ground is made up of base, arm, and wrist. The base of the manipulator contains the powerful actuator which bears the load of all links, motors, tool, and other auxiliary parts. Usually, the base of the robot rotates along the z-axis. The arm portion of the robot contain links which are attached to each other by the help of joints. The arm resembles like the structure of the human hand. The link is a rigid structure joined together according to the kinematic relationship between them. The shape and size of link is compatible with environment and plan of the task performed by the robot. The joint contain actuator which moves by taking input signals from the controller.

There are five types of joints used in robots: revolute, prismatic, orthogonal, twisting, and linear. Mostly used joint among them are revolute joint or pin joint having one degree of freedom in which kinematic pair have rotary motion along the longitudinal direction. Next is the prismatic joint that allows the kinematic pair to have sliding motion relative to each other. The wrist portion of the arm perform roll-pitch-yaw motion and holds the end-effector. The wrist is used to locate the position of the desired point precisely in coordinate space. The tool or end-effector is a task-specific device such as gripper or welding torch. The end-effector conducts motion in context with the last joint in the wrist and the location of that joint in the Cartesian space forms a working volume known as workspace of the robot (Craig, 2005).

The industrial robot's mechanical structure is of six types; Cartesian, SCARA, polar, cylindrical, articulated, and delta. The articulated robot with revolute joints is the most successful robot design because of its anthropomorphic characteristic. The articulated manipulator is an open chain mechanism of serially connected links that gives maximum workspace than other types of robots. It is the most versatile structure, used widely in the manufacturing industries all around the world. However, this robot structure is inherited with the drawback of error multiplication, and the inertial imbalance affects velocity of the end-effector.

The articulated manipulator conducts tasks with end-effector whose position and orientation should be well determined with respect to the fixed base. There are three coordinate frames: first is the base frame, second is the tool frame and third is the job frame. There are frames assigned to tables, stand, and if robot base is not at the origin, then the frame corresponding to the origin of coordinate system comes in to picture. The position of the end-effector is determined through the configuration of robot's joint. The angular motion of joints in the arm is responsible for the position of end-effector, while the angular motion of joints in the wrist forms the orientation (Waldron and Schmiedeler, 2008).

Degrees of freedom (*d.o.f.*) of robot is defined as the number of independent position variables required to locate all parts of mechanism (Craig, 2005). An articulated robot having *n* revolute joints, have *n* number of degrees of freedom. The necessary relationship between the position and orientation of the end-effector and the joint configuration is given by: the location of a point *P* in the coordinate frame *l* relative to the origin frame *o* can be denoted by a *3 by l* vector ${}^{l}_{o}P$ as:

$$ {}^{l}_{o}P = \begin{bmatrix} {}^{l}_{o}P_x & {}^{l}_{o}P_y & {}^{l}_{o}P_z \end{bmatrix} \qquad \ldots (2.1) $$

The components of this vector are the projections of ${}^{l}_{o}P$ onto the respective axis. This vector is the translation part of the homogeneous transformation matrix. The translation is a displacement in which no point in the rigid body remains in its initial position and all the straight lines in the rigid body remain parallel to their orientations. The translation of a body in space can be represented by the combination of its position prior to and following the translation. Conversely, the position of a body can be represented as a translation that takes the body from a position in which the coordinate frame fixed to the body coincides with the fixed coordinate frame to the current position such that the two frames are not coincident. Thus, any representation of the position can be used to create a representation of displacement and vice versa (Waldron and Schmiedeler, 2008).

The orientation of the end-effector is a displacement in which at least one point of the rigid body remains in its initial position, and not all lines in the body remain parallel to their initial orientations. The orientation matrix is a matrix that transforms a vector expressed in coordinate frame *o* to a vector expressed in a coordinate frame *l*. It represents the orientation of frame *o* relative to frame *l*. If the basis vector of the coordinate frame *o* is $(\hat{x}_o, \hat{y}_o, \hat{z}_o)$ and the $(\hat{x}_l, \hat{y}_l, \hat{z}_l)$ is the basis vector for frame *l* then the orientation matrix, ${}_o^lR$ becomes:

$${}_o^lR = \begin{bmatrix} \hat{x}_o.\hat{x}_l & \hat{y}_o.\hat{x}_l & \hat{z}_o.\hat{x}_l \\ \hat{x}_o.\hat{y}_l & \hat{y}_o.\hat{y}_l & \hat{z}_o.\hat{y}_l \\ \hat{x}_o.\hat{z}_l & \hat{y}_o.\hat{z}_l & \hat{z}_o.\hat{z}_l \end{bmatrix} \qquad \text{. . . (2.2)}$$

The orientation matrix can be represented by Euler angles or by the Roll-pitch-yaw representation. The Homogeneous transformation matrix, ${}_o^lT$ for generating the new position of the tool centre point from frame *o* to frame *l*, combines the translation and orientation matrix together, i.e.

$${}_o^lT = \begin{bmatrix} & {}_o^lR & & {}_o^lP \\ 0 & 0 & 0 & 1 \end{bmatrix} \qquad \text{. . . (2.3)}$$

The homogeneous transformation matrix sometimes generates such matrix that cannot be traced by the robot, and this problem is known as singularity. Singularity is defined as a condition caused by the collinear alignment of two or more robot axes resulting in unpredictable robot motion and velocities (ANSI/RIA R15.06-1999). This problem can be sorted out by using quaternion, which is a four-component vector. The position of the tool in the Cartesian coordinate space is achieved by joint motion as the power is supplied to actuators at the joints. Therefore, to get the values of joint configuration for the corresponding values of homogeneous transformation matrix can be determined by the kinematics that depends on the robot structure.

2.2.2. Kinematic Modeling

The kinematics deals with the mathematical formulation of relationship between time and joint variables such as position, velocity, acceleration, and jerk. The kinematic modeling deals with the motion of robot without considering the forces causing it. The equations in kinematic model are the time based parametric formulation of joint parameters of the robot (Craig, 2005; Waldron and Schmiedeler, 2008; Siciliano *et al.*, 2009b).

Motion of manipulator can be defined in two coordinate systems: joint space and Cartesian space. The joint space notation is a time based empirical relation determining the joint parameters. The Cartesian space notation describe the position and orientation of end-effector in the Coordinate frame. According to the space description, the kinematic is also classified into two ways: forward and inverse. The forward kinematics deals with the conversion of joint configuration to the Cartesian configuration and the inverse kinematics deals with the transformation of Cartesian configuration into joint configuration (Angeles, 2003).

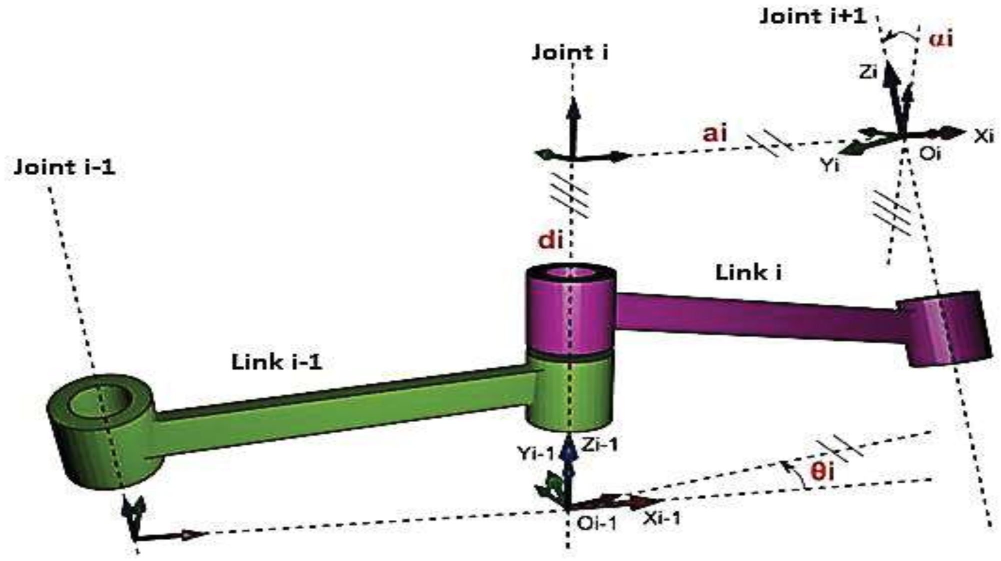
2.2.2.1. Forward Kinematics

The forward kinematics deals with the objective of computing the position and orientation of the end-effector frame relative to the base frame with known joint angle values (Paul and Shimano, 1978).

$$\begin{bmatrix} P \\ R \end{bmatrix} = \partial_{fk}\left(q,\delta_k\right) \qquad \dots (2.4)$$

Where P is the end-effector's position vector, R is the end-effector's orientation matrix, ∂_{fk} is the nonlinear forward kinematics function, q is the joint position vector and δ_k is the fixed kinematic link parameters vector consist of information regarding the parameters representing link lengths, offsets, rotation angles of each joint axes relative to the previous axis.

The joint frame should be determined in a systematic process, and it is advantageous to use a convention for assigning frame to the joints. A standard method for frame assignment is the Denavit-Hartenberg method (also known as D-H convention). D-H convention assigns frame to joints in such a manner that only four parameters were required for allocation with respect to other joint (Denavit and Hartenberg, 1955). These four parameters are, link length a_i, link twist α_i, joint offset d_i, and joint angle θ_i. Fig. 2.1 show the arrangement of links and joints for assigning frames according to the D-H convention. Later, the D-H convention was modified to remove the subscription mismatch during the axes labelling (Paul, 1972; Craig, 2005; Khalil, W., & Dombre, 2007).



Courtesy:https://en.wikipedia.org/wiki/Denavit%E2%80%93Hartenberg_parameters#/media/File:Classic-DHparameters.png

Figure 2.1. The D-H convention frame assignment

The transformation between any two joints of the robot having n number of joint is evaluated adjacently by ${}_{i-1}T^{i}$, from *i-1*th joint to *i*th joint.

$$ {}_{i-1}T^{i} = \begin{bmatrix} c\theta_i & -s\theta_i c\alpha_i & s\theta_i s\alpha_i & a_i c\theta_i \\ s\theta_i & c\theta_i c\alpha_i & -c\theta_i s\alpha_i & a_i s\theta_i \\ 0 & s\alpha_i & c\alpha_i & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \qquad ...(2.5) $$

where *s* and *c* stands for the trigonometric functions sin and cos, respectively. The final transformation between the *n*th joint and the zeroth joint (base) is obtained by successively multiplying all the transformations of the joints between them as:

$$ {}_{0}T^{n} = {}_{0}T^{1} \cdot {}_{1}T^{2} \cdot {}_{2}T^{3} \cdot {}_{3}T^{4} \ldots\ldots \cdot {}_{n-1}T^{n} \qquad ...(2.6) $$

2.2.2.2. Inverse Kinematics

The kinematics that solves for joint configuration of robot from the parameters values in Cartesian coordinate system is termed as inverse kinematics. In robotics, inverse kinematics frequently applied and have complex mathematical formulation. The inverse kinematics of the serial robot manipulator brings the end-effector to the desired location and orientation by calculating the corresponding joint values. For Cartesian coordinate value there could exist multiple solutions of a point, such as the elbow up and elbow down configuration. In addition, the number of solutions can be infinite in case of the kinematically redundant manipulator. The inverse kinematics equation can be expressed as,

$$ q = f_{ik}(X, C, \delta_k) \qquad ...(2.7) $$

Where C is the processing data required to select the feasible solution, related to the desired configuration, *X*. Numerical methods, semi-analytical methods and the closed form methods can obtain the inverse kinematic solutions.

The closed form solutions are used for robotic manipulators, as they are computationally faster and secure maximum possible solutions. The results from the kinematic computation has been supplied directly to controller and helps in rapid control and response during operation. However, closed form solutions have some disadvantages, that, it is robot dependent and involves large number of geometric parameters. Therefore, the problem of mathematical complexity is a major issue in the closed form solutions. To reduce complexity, the numerical methods or semi-analytical solutions has used as an alternative (Fu *et al.*, 1987).

In a numerical approach, the joint configuration has obtained by the iterative computation procedure. This process requires initial estimation that is subjected to uncertainty. The iteration process also suffers difficulty in converging to a correct solution and several solutions are automatically eliminated. Moreover, if the Jacobian matrix is singular, then this approach will not be able to provide any solution of the problem. In semi-analytical solution, few joint variables were determined analytically while rest of them are evaluated through iterative approach. In this approach, the problem of large computational calculations and large time consumption remains the same.

2.2.3. Dynamic Modeling

Change in force and torque during motion in a robot are measured through dynamics. The time-dependent change in position of robot end-effector is termed as dynamic behavior of the robot. In dynamics, the external force applied during motion is considered. The external forces are due to the payload at the end-effector and the forces due to gravity. The internal forces such as the frictional, Coriolis forces, and inertial forces at the joints due to links are also monitored. The actuator balances all forces by applying torque at the joints.

The dynamic modeling of the robot manipulator develop correlations between the joint torque, dynamic parameters of links, and the external forces. The dynamic analysis of the robot motion is necessary for the evaluation of the total power consumed and the torque exerted at each joint. That information is essential for robot selection, design, and optimization. Several approaches have been proposed in the literature illustrating various dynamic principles to model the relation between the dynamic parameters. Moreover, there are two types of dynamic modeling: forward dynamics and inverse dynamics.

2.2.3.1. Forward Dynamics

In forward dynamics, the angular acceleration at joints was calculated from the force or torque applied by the actuator. The numerical integration of the joint acceleration will give the joint velocity and angular position. The forward dynamics is essential for the simulation and feedback control of the robot, and it is computationally more complicated than the inverse dynamics formulation (Craig, 2005). The mathematical model given below determines the forward dynamics calculation of the acceleration:

$$\mathrm{I}\ddot{\boldsymbol{\theta}} = -\boldsymbol{C}(\boldsymbol{\theta},\dot{\boldsymbol{\theta}})\dot{\boldsymbol{\theta}} + \tau(\mathrm{t}) + \boldsymbol{\delta}(\boldsymbol{\theta},\dot{\boldsymbol{\theta}}) + \boldsymbol{\gamma}(\boldsymbol{\theta}) + \boldsymbol{J}^T\boldsymbol{w}^W \qquad \ldots (2.8)$$

Where I is the positive definite *n* x *n* generalized inertia matrix of the manipulator, the term $-\boldsymbol{C}(\boldsymbol{\theta},\dot{\boldsymbol{\theta}})\dot{\boldsymbol{\theta}}$ represents the vector of inertia forces in quadratic order, τ and $\boldsymbol{\delta}$ denote the *n* dimensional vectors of active and dissipative generalized forces, $\boldsymbol{J}^T\boldsymbol{w}^W$ takes into account the forces applied at the operating point when $\boldsymbol{w}^W$ a static wrench acts on the end-effector. The above equation is solved for the angular acceleration using recursive Newton-Euler algorithm. The calculation of $\ddot{\boldsymbol{\theta}}$ have the complexity of the order of n^3 represented as $O(n^3)$. The complexity of the system is reduced by the composite rigid-body method by Walker and Orin, 1982, and the generalized inertia matrix, *I* is computed recursively at complexity of $O(n^2)$.

Further, in 1983 Featherstone proposed an algorithm with a complexity of O(n), by neglecting the Coriolis and centrifugal forces. Later, in 1987 the same author proposed an articulated-body method by considering the neglected forces (Featherstone, 1983). In 1997, the robot dynamics solutions using the decoupled natural orthogonal complement to enable the real-time inversion of the mass matrix has been reported (Saha, 1997).

The computation of the quadratic inertia $\mathbf{C}\dot{\boldsymbol{\theta}}$ and the dissipative torque $\boldsymbol{\delta}$ is dependent on the robot and the constitutive equations used for the formulation of viscous and the Coulomb friction forces (Angeles, 2003). Thus, the term on the left-hand side of the equation 2.8 requires the calculation of I and $\ddot{\boldsymbol{\theta}}$ for which the inversion of the *n by n* generalized inertia matrix I is necessary. The inverted matrix is positive definite; therefore, the angular acceleration term can be solved by using the Cholesky-decomposition algorithm (Bejczy, 1974).

Mata *et al.* (2002) also proposed an algorithm using Gibbs-Appell equation as a starting point. They write the Gibbs-Appell equation based on the generalized coordinate and their time derivative. Nonetheless, the intricate details of the robot dynamics can be found in the texts such as (Balafoutis and Patel, 1991; Lilly, 1993; Mata *et al.*, 2002; Yamane, 2004; Featherstone, 2008; Siciliano *et al.*, 2009a).

2.2.3.2. Inverse Dynamics

In inverse dynamics, the force or torque exerted by the actuators are computed by algebraic expressions for the given kinematic configuration of the manipulator. The inverse dynamic is relevant for computing torque control equations of motion that are used in power control, mechanical structure optimization, determination of motor and gear profiles. The time-based relation in Cartesian or joint coordinates and data from inertial and structural parameters of robot constructs the time-based torque or force relations at different joints.

The two basic inverse dynamic modeling principles are Newton-Euler equation and Lagrange-Euler equation. The Newton-Euler principle of formulation is most appropriate for the inverse dynamic problem. Orin *et al.* (1979) gave the first recursive algorithm using the Newton-Euler equation. Later, Luh *et al.* (1980) developed the more efficient version of Orin's algorithm. This inverse dynamic formulation was computationally less expensive, and this approach has been improved continually (Featherstone and Orin, 2000). Newton-Euler formulation is derived from Newton's second law and the D'Alembert principle (Mittal and Nagrath, 2008). Newton's law state that the force applied on the mass moving with the rate of change in velocity can be described as,

$$\boldsymbol{F_i} = m_i\dot{\boldsymbol{v}}_{\boldsymbol{i}} \qquad \text{. . . (2.9)}$$

Where $\boldsymbol{F_i}$ is the force acting at the centre of mass of *i*th link, m_i is the mass of the *i*th link and $\dot{\boldsymbol{v}}_{\boldsymbol{i}}$ is the linear acceleration. The algorithm consists of two steps: (i) kinematic parameter determination of all the links such as velocity and acceleration (ii) dynamic parameter calculation such as torque and joint forces to determine the constraint and external wrenches (Angeles, 2003). The kinematic computations are known as inward recursions in which the time-based formulation of the velocity and acceleration has been prepared. The dynamic computations are also known as outward recursions and the equation governing the dynamic behaviour is stated as,

$$\boldsymbol{\tau} = M(\mathbf{q})\ddot{\boldsymbol{q}} + \boldsymbol{h}(\boldsymbol{q}, \dot{\boldsymbol{q}}) + \boldsymbol{G}(\boldsymbol{q}) \qquad \text{. . . (2.10)}$$

Where $\boldsymbol{\tau}$ is the joint torque vector, M is the joint space symmetric inertia matrix; $\boldsymbol{G}$ is the gravity force vector; $\boldsymbol{h}$ is the vector of centrifugal and Coriolis forces; and $\boldsymbol{q}, \dot{\boldsymbol{q}}, \ddot{\boldsymbol{q}}$ are the joint position, velocity, and acceleration vectors, respectively. The terms appeared in the equation are dissipative and are difficult to model due to friction and striction which depend strictly on the joint variables. Another approach for dynamic model is the Lagrange-Euler principle (Uicker, 1965; Kahn and Roth, 1971). Euler-Lagrange formulation is based on the Lagrange function, $\mathbf{L}$ is formulated as the difference between the total kinetic energy, $\boldsymbol{K}$ and the potential energy, $\boldsymbol{P}$ of the system, i.e.

$$\mathrm{L} = \boldsymbol{K} - \boldsymbol{P} \qquad \text{. . . (2.11)}$$

Consider m is the mass of the system, $\boldsymbol{v}$ is the velocity, I is the inertia and $\boldsymbol{\omega}$ is the angular velocity, then the total kinetic energy and the potential energy of the system is defined as,

$$\boldsymbol{K} = \frac{\mathbf{1}}{\mathbf{2}} m\, \boldsymbol{v}^{\mathbf{2}} + \frac{\mathbf{1}}{\mathbf{2}} I\, \boldsymbol{\omega}^{\mathbf{2}} \qquad \text{. . . (2.12)}$$

$$\boldsymbol{P} = \frac{\mathbf{1}}{\mathbf{2}} m\, g\, l\, \mathbf{sin}(\boldsymbol{\theta}) \qquad \text{. . . (2.13)}$$

Where g is the magnitude of gravitational acceleration, l is the link length, and $\boldsymbol{\theta}$ is the rotation angle. From above equations, it is understood that the kinetic energy is dependent on the configuration and motion of links, whereas the potential energy depends only on the configuration of manipulator. Euler-Lagrange dynamics equation of motion is stated as,

$$\frac{\boldsymbol{d}}{\boldsymbol{dt}}\left(\frac{\boldsymbol{\partial L}}{\boldsymbol{\partial \dot{q}_i}}\right) - \frac{\boldsymbol{\partial L}}{\boldsymbol{\partial \dot{q}_i}} = \boldsymbol{\emptyset_i} \quad \text{for } i = 1, \ldots\ldots, n \qquad \text{. . . (2.14)}$$

Where n is the independent generalized coordinates, $\boldsymbol{\emptyset_i}$ is the generalized forces for applied forces at given coordinates, $\boldsymbol{q_i}$ is generalized coordinates. From equations 2.11, 2.12, 2.13, 2.14, and solving for the generalized torque of the link i of n-*d.o.f* robot:

$$\boldsymbol{\tau_i} = \sum_{\boldsymbol{j=1}}^{\boldsymbol{n}} M_{ij}(\boldsymbol{q})\, \boldsymbol{\ddot{q}_j} + \sum_{\boldsymbol{j=1}}^{\boldsymbol{n}} \sum_{\boldsymbol{k=1}}^{\boldsymbol{n}} \boldsymbol{h_{ijk}} \boldsymbol{\dot{q}_j}\, \boldsymbol{\dot{q}_k} + \boldsymbol{G_i} \quad \text{for } i = 1, \ldots , n \qquad \text{. . .(2.15)}$$

Where M_{ij} is the effective coupling inertia, $\boldsymbol{h_{ijk}}$ represents the centrifugal and Coriolis forces, $\boldsymbol{G_i}$ gravitational repulsion force, and $\boldsymbol{q}, \boldsymbol{\dot{q}}, \boldsymbol{\ddot{q}}$ are the joint position, velocity, and acceleration vectors, respectively. However, above formulations were found to be computationally inefficient due to large number of the algebraic operations involved (Hollerbach, 1980).

Both approaches for modeling the inverse dynamics of the manipulator are analyzed regarding computational efficiency and in general by (Silver, 1982; Spong and Vidyasagar, 2004). The Kane's equation is also used to develop the inverse dynamic model, whose analysis can be found in (Featherstone and Orin, 2000).

2.2.4. CAD Modeling of Robotic System

The computer-aided design is a technique to develop the computer model of the product through graphical modeling software in either two or three dimension. The CAD modeling software such as AutoCAD by Autodesk 1982, CATIA by Dassault system in 1981, SolidWorks in 1995, and Autodesk Inventor in 1999 are some of the widely used modeling packages used in industries and educational institutions these days (Rao, 2004).

The CAD modeling has several advantages over conventional or manual drafting methods. The first advantage is the digitalization of the system that will help in systematic analysis and parameter optimization. CAD also helps in increasing the productivity of designer by providing tools to improve the design before implementation. It is beneficial for manufacturers, as it can increase the quality of the finished goods by applying continuous change in product design according to demand. The CAD-based system is well organized, as they keep documentation and online track of the ongoing projects. CAD packages are used in the design of complex electronic circuits and automotive parts, which are difficult to draft manually. Besides all the advantages, the CAD based systems are dependent on skilled labor and updated techniques to match the market trends (Groover, 1987).

The use of CAD in manufacturing and machine design (CAD/CAM) is a progressive area and various designing packages such as Matlab, ANSYS, Simulia, COMSOL Multiphysics, ANSYS fluent and many more are available everywhere. These design packages are enabled with programming interface through which manipulation of the system can be done, and other tools can be integrated to improve the system output. In robotics, Matlab, AutoCAD, SolidWorks, CATIA are often used as the modeling and programming package for solving various complex problems, innovatively. Presently, CAD modeling is the backbone of the simulation process. Presently, the software packages can simulate most of tasks such as robot motion and operations, workcell design and task planning (Zeid and Sivasubramanian, 1991).

The CAD-based modeling and simulation process consists of two steps: first is the development of CAD model by various techniques such as digital data acquisition system, motion capture by a camera, assembly modeling, and the human-computer interface. The second step is the data exchange between CAD system and virtual simulator (Groover and Zimmers Jr., 2006). Lee and ELMaraghy (1990) developed a CAD-based offline programming system for robotic manipulator. Nagata (2007) presented a CAD-based approach to control force applied in robotic polishing of a mold that result in highly smooth surface. The control system assisted by feedback of the operation performs stable force control during polishing of the curved mold surface. Liu *et al.* (2008) developed a CAD directed inspection and design of the laser-guided measuring robot.

Kaldestad *et al.* (2012) presented an online obstacle detection approach for the industrial robot using CAD-based expert training system. CAD model based integrated system has been successfully utilized for simulation, planning and training of an assembly task (Leu *et al.*, 2013). Neto and Mendes, 2013 developed the robot motion-planning algorithm through a CAD-based modeling package. In these approaches, robot motion is simulated by using CAD drawings and the data obtained from the drawing is then processed to obtain the optimal trajectory in the coordinate space.

An important feature of CAD package is the application programming interface that allows the extraction, analysis, and processing of the data obtained from the drafted files. Various programming languages such as Visual Basic, Visual C++, Matlab can be clubbed with the CAD package to solve complex problem efficiently. The computer-aided robotic programming was facilitating the custom analysis in prostheses surgery and preventing aseptic failure (Ming *et al*. 2015). CAD/CAM based simulation of robotic milling operation is used in the development of medical parts such as prostheses. The CT scan images of the joint are drafted into 3D CAD drawing from which the location of the robot tool point has been generated using CAM software.

Baizid *et al.* (2016) developed a simulation and optimization package for robotic path planning using SolidWorks based simulation algorithms. SolidWorks has application programming interface which can provide data of CAD models for simulating and optimizing the trajectory for continuous point-to-point operation. The CAD modeling data has been used to develop an automated robotic arc welding based additive manufacturing system (Ding *et al.*, 2016).

The CAD-based programming algorithm has also been used in ship body manufacturing. Ferreira *et al.* (2017) developed an offline robotic welding method for constructing sizeable metallic body parts in a cellular system. The hyper-flexible welding workcell with nine degrees of freedom for small batch type production has been used for simulation.

The salient feature of the CAD based simulation approach is that it allows processing of CAD data by developing algorithms that made an inexperienced user to develop desired solutions. Industrial robots aided with computer-aided designed scanned paths helps in the non-destructive testing of complex metallic shapes. The offline programmed scanned path was fed to the robotic manipulator for NDT inspection of typical skin panel of an aircraft wing. The CAD model of the reverse engineered test piece helps in generating accurate models and produce precise NDT results (Morozov *et al.*, 2018).

Moreover, the CAD based simulation approaches have been usually clubbed with some data extraction techniques such as meshing, for further simulation and analysis. The present thesis work explores the possibilities of modeling and simulation by Matlab based algorithms.

2.2.5. Point Cloud Modeling

A point cloud is a set of data points in coordinate space. The data points of an object are obtained from the scanning and computer-assisted modeling methods. The point clouds are used in numerous applications such as inspection, metrology, visualization, 3D modeling, optimization and rendering. In robotics, point clouds are used for calibration of robots, workcell, inspection, nondestructive testing, precision manufacturing, and navigation.

In point cloud modeling, the point cloud data set is transformed into the 3D surface mesh model by Delaunay triangulation, alpha shapes method, and others for different purposes. Several software related to point cloud generation and processing are 3DReshaper, RealityCapture, Agisoft Photoscan, LiMapper, Geomagic, and Point Cloud Tool.

The general approach to develop point cloud model is scanning followed by registration and then modeling. Another approach readily used for point cloud modeling is meshing of CAD model of an object. The scanning process usually forms the point cloud of an object, and laser scanning is the leading method in this area. The point cloud data was then converted into the three-dimensional model using the mathematical curves (Azariadis and Sapidis, 2005).

The point cloud models have also used for the evaluation of the finished products and automatically detect the error by point cloud registration from the measured point cloud or discretized point cloud from the CAD models. CAD models have used in various ways in the point cloud-based applications (Li *et al.*, 2013). Beserra G. *et al.* (2013) developed a point cloud acquisition method of the real world and in real time, using moving fovea based multiresolution technique that reduces the processing time by seven times.

Zou and Zhao (2013) developed a direct approach for constructing the tool travel path according to point cloud of the machines. Ali Hosseininaveh *et al.* (2014) developed an algorithm for accurate 3D point cloud modeling using stereo imaging network of captured images. Robots have been used in the point cloud registration of an object using stereovision system. The point clouds of working environment, robot arm, and an unknown object were captured by stereovision camera at different angles to obtain the complete geometrical structure of workcell (Jerbić *et al.*, 2015).

Horváth and Erdős (2017) presented a point cloud based automatic robot cell calibration approach. The measured data of the objects are compared with the CAD model of the tested object. Potó *et al.* (2017) developed a support system for the autonomous vehicle through laser scanned point cloud data. Shijun *et al.* (2017) presented a point cloud registration approach. These are the auxiliary application of the point cloud. Nowadays, in several manufacturing processes, the point cloud modeling has been preferred to perform high precision 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 a point cloud image, and the CAD model of object has been used as referral for point cloud image. Then, the point cloud data has been fed to a subsidiary package or programming interface to perform the desired task.

Point cloud modeling also helps in the inspection of the large industrial systems, as the inspection through human documentation becomes untenable. The terrestrial laser scanned data has been compared with the RANSAC point cloud data for the onsite dimensional inspection. The point cloud model of the object is in PCD (Point Cloud Data) format, and the CAD data is in STL (Stereolithography) format (Hong *et al.*, 2018). Kardos and Váncza (2018) developed a method that utilizes the point cloud based collision simulation approach to directly determine the precise disassembly of complex mesh models in assembly sequence planning. API of CAD package has been used to extract the data from the drafted model which has meshed to obtain the point cloud model.

Mineo *et al.* (2018) presented the surface-point cloud boundary detection and edge construction algorithm. Robotic mapping and point cloud modeling has been used for safety and increase productivity of construction site in which mobile robot navigation system scan the complete site for simulation. The two-dimensional simultaneous localization and mapping technique is used to obtain the locational information about robot that is transformed to high resolution 3D point cloud in real time (Kim *et al.*, 2018).

The point cloud modeling is rapidly advancing multidisciplinary field which has been explored by researcher in computer, mechanical, electrical, electronics and civil in numerous ways. This thesis work utilizes this innovative technology to enhance the quality and efficiency of the workcell design and planning system.

2.3. Robotic Workcell

Industrial robots operate with tools, machines, and equipment that forms a group based on the pre-planned task. This purposeful grouping of several machines and robots in a section of an industry is termed as a cell (Koren, 1987). The robotic working cell is often termed as workcell. The workcell is usually equipped with conveyors, loading/unloading table, tool posts and production machines. The robotic workcell has found to be useful in reducing inventory and transportation cost and provides best product quality with minimum time and cost. A workcell is flexible to change in product design, cell layout, machines change in accordance to the predefined planning and programming by experts. The workcells have installed in large number of industries, and their reliability has creating a favorable scenario for their expansion and research in the areas such as layout planning, sensors and controls, task scheduling, and trajectory planning (Groover, 1987).

Chedmail and Wenger (1990) presented the layout optimization approach and the initial works on layout design by Lueth (1992) provides an initial pace to the automated robot workcell layout planning process. Similarly, Tay and Ngoi (1996) presented an algorithm for robotic workcell layout optimization. The layout-designing task is the primary and vital aspect of the robotic workcell by which several costs and time-related factors are either directly or indirectly affected. The issues related to layout design has been detailed in work of (Kusiak and Heragu, 1987; Reinhardt, 1988; Huck, 1989; Jian and Ai-Ping, 2009) and the layout design has been taken in details in the next section.

Next task in workcell is the part scheduling and sequencing problem. Several intricate details of the scheduling such as machine environment, process restriction and objective function selection have been discussed in detail in (Cao and Sanderson, 1991; Dawande *et al.*, 2007). Further, control and sensing deals with the automation of the robotic workcell and is said to be the backbone of above mentioned tasks. The controller programming, sensor feedback control, and error tracking have been discussed in details in (Hutchinson and Kak, 1989; Brock and Khatib, 1999). Different aspects of the robotic workcell design and planning are given in details in (Groover, 1987; Kumar, 1992; Raton, 1999). However, this work aims for the simulation and optimization of an industrial workcell.

2.3.1. Simulation of Robotic Workcell

The simulation means the emulation of a real process or system that can be accomplished by creating a computer model for research and analysis purpose. Simulation of the robotic workcell deals with the modeling and analysis of the robot motion, design and replicating the internal objects of the real workcell system. A robotics simulation package is a tool for creating an embedded application of a robot without physical dependency on the actual robot and the cell environment (Shannon, 1998).

The significance of simulation has been realized in every area of robotics. The product design, operation management, advanced control and programming and other areas required simulation at various levels. The practical investigations are difficult to perform and consume enormous investment, workforce and time. Comparatively, the best simulation is that which considers the realistic data of the operating environment such that the results should be valid and applicable without further processing. The accurate and multi-functional tools for robotic simulation are in demand, and several issues are to be addressed such as computational power, accuracy, and cost optimization (Žlajpah, 2010). The available simulation packages perform similar simulation task and differs regarding accuracy, number of robot models, integration with other programming formats and licensing cost.

The robotic simulation software can be differentiated as tools based on general simulation system and tools based on special robotic system (Park, 2006). The first category tools are toolboxes, modules and individual libraries that build robotic environment within the general simulation system. This simulation system utilizes different functions available from various toolboxes such as global optimization toolbox, wavelet toolbox, neural networking and others. Also, there are general tools such as Modelica, Matlab/Simulink, Mathematica, 20-sim used in first category simulation system.

The second category special simulation tools are multi-function package having a GUI interface that can perform simulation for different robots and situations such as sea robots, mobile robots, and humanoids. The Matlab based simulation tools are readily used by researchers and provides complex problem-solving capabilities due to vast number of functions and open programming environment. These simulation tools also provide feedback through an interactive visual interface. Commercial simulation tools such as RobotStudio, MotoSim and Roboguide have both offline and online programming features.

These packages conduct simulation of robotic workcell in 3D environment, and the virtual robot modeling simulate the actual working of the real world robotic workcell. However, these packages are dedicated to a robot family only that limits their application to other areas.

The modeling and simulation packages based on various algorithms and principles were developed in 1970s starting from the OSSAM (Dillon, 1973). Most of the simulation packages deal with the kinematic and dynamic analysis of the robotic system such as Robotic Vision and Control (RVC) toolbox by P. I. Corke (Corke, 2013), Planar Manipulator toolbox (Žlajpah, 1997). The robotic workcell simulation packages capable of simulating working environment and can perform the kinodynamic analysis of robot motion are described further.

Microsoft robotics developer studio is open user 3D simulation package developed by Microsoft in 2006 having a wide range of sensors and multi-robot (Colon and Verbiest, 2008). Robologix design by Logic Design Inc., works for various robotic model such as ABB, Fanuc, Kawasaki. Webots is a 3D simulation platform by Cyberbotics, is available in Windows, Linux and Mac, and is a widely used package for research, education, industry, and services. Different programming languages can be clubbed with it, and open source libraries can be attached (Michel, 2004).

Roboguide by Fanuc is a professional package for Fanuc robots that simulates in three dimensions. MotoSim by Motoman industrial robots is a specific simulator. RobotStudio is an industrial robot simulator by ABB Inc., Sweden. It allows the user to simulate realistic working environment for ABB robots. WorkCellSimualtor by IT Robotics, Italy is a 3D simulator package applicable to packaging sorting laser cutting. Roboanalyzer by IIT Delhi, India is a 3D simulator for teaching and learning purpose. Gazebo is a simulator compatible with ROS platform and it is an open source package.

2.3.2. Optimization Algorithms for Robotic Workcell

The robotic workcell design is a vital research area for the industries to grow and compete globally. The optimization algorithms used to develop an optimal robotic workcell have two distinctions: (1) the algorithms considering global search algorithms such as Genetic Algorithm and (2) the algorithms considering local search near the initial feasible solution such as sequential quadratic programming.

For solving, a layout design problem for robotic workcell, algorithms such as simulated annealing, genetic algorithm, and sequential quadratic programming have been mostly used. These problems generally formulate nonlinear objective function with nonlinear constraints. The metaheuristic group of optimization algorithm such as nature-inspired algorithms such as artificial bee colony, differential evolution, charge search system and particle swarm optimization have also been used for workcell optimization (Yang *et al.*, 2017). These algorithms can handle the multi-objective problem and provide global solutions. The conventional optimization approaches such as sequential quadratic programming have been used to solve the similar optimization problem. However, these algorithm poses difficulty in convergence for complex cases.

Variants of optimization approach such as modified simulated annealing has been developed to solve complicated case of cell design with 3D models of machines and robot (Barral *et al.*, 2001). The genetic algorithm-based multi-objective optimization approach has been used to obtain the optimal configuration of the robotic workcell (Izui *et al.*, 2013), while the SQP algorithm has been used to solve the constrained nonlinear optimization problem (Mata and Tubaileh, 1998).

2.4. Layout Design Analysis for Robotic Workcell

The layout is a specific arrangement of machines and robots on shop floor (Lueth, 1992). Layout design is an essential aspect of workcell concerning the optimization of various parameters. The optimal placement of machines with respect to robotic manipulator in coordination with each other is a crucial task.

A broad account of publications addressing this issue are presented in this section and various problems in layout design of robotic workcell has been reported. Several approaches have specific criteria and objective for arranging robots and machines to perform a task. It provides scope for the development of several new approaches.

The early report of Kusiak and Heragu (1987), gives the detailed description of the layout design aspects of the industrial workcell. Pamanes (1989) proposed an approach for optimizing the location of the manipulator with optimal multiple kinematic performance indices. Their investigation was mainly concentrated on the proper placement of the robot concerning link dimensions and task locations. A promising solution to the problem of optimizing robot location using Monte Carlo method has been presented (Rastegar and Fardanesh, 1990; Tu and Rastegar, 1993)

In 1991, Pamanes and Zeghloul presented a method to keep the manipulator links within their bounding range to optimizes the location of the robot in the workcell. Abdel-Malek (1995) located the manipulator according to the task or operating point by using maximum orientability principle. In 1995, an approach for optimizing the position of the manipulator base in relatively small force quasi-static task has been devised by Papadopoulos and Gonthier, using redundant variables as optimization parameters.

In 1996, Tay and Ngoi developed an innovative approach to automatically design the layout of a robotic workcell using the heuristic algorithm based on the minimum total path travelled by manipulator for given sequence. They used the spatial representation system for modeling machines and orienting them in the workcell. Abdel-Malek (1997) introduced a method for quantifying the reachability of robot manipulator, that is handy for the description of robot work envelope in closed form.

Mata and Tubaileh (1998) devised a machine layout planning method for a flexible manufacturing cell. The method generates the locations and orientations of machines concerning robot reachability at each working point. The method simulates the collision between robot and machine and compare the minimum time and minimum distance solutions together. Ji and Li (1999) presented an algorithm and a dialytic elimination approach to identify the placement parameters for modular platform manipulators. It was used for guiding the placement of leg modules to obtain the most useful configuration for the intended operation.

Pires and Sá Da Costa (2000) presented an integrated programming approach for the industrial robot manipulators using object-oriented principle that has been utilized for designing the workcell architecture. Barral et al. (2001) presented an approach for workcell layout optimization by combining the constructive algorithm and simulated annealing algorithm. The specific designing approach has been adopted for the placement and orientation of machines on the workcell floor. The machine layout has been optimized by using the minimum cycle time for the given task. The approach also presents the graphical view of the optimal layout that can also provide the immediate feedback.

Guo *et al.* (2003) presented a machine cell parameters optimization and simulation system using the principle of separating the models from the algorithm. Virtual manufacturing is the prominent approach for the design of workcell. Slomp *et al.* (2005) devised a virtual cellular manufacturing system to expand the capabilities of the cellular manufacturing system. Various factors for workcell optimization such as minimum intercell movement of parts, flexibility, minimizing load imbalance, and cell size restrictions have been considered during simulation.

Chan *et al.* (2006) developed a two-stage approach for layout optimization problem and machine part grouping. In the first stage, the machine cells and part families have been identified, and in the second stage, the workcell formation problem has been formulated considering the machining sequence. The optimal layout of the workcell has been generated by solving an NP-hard problem using genetic algorithm. Jayaswal and Adil (2004) proposed a cell design algorithm using sequence data and checking machine replications and alternative process routings. The total sum of cost involved in intercell movement, machine investment and machine operating have minimized using simulated annealing and local search heuristic algorithm. Mahdavi and Mahadevan (2008) developed a sequence data based algorithm, CLASS for layout design of a workcell.

Hammond and Shimada (2009) published an approach for improvement in layout design of workcell using weighted isotropy matrices. They used redundant manipulators and considered the dynamic performance measure for high acceleration and high payload tasks. The torque weighted isotropy measure has been used as fitness metric for redundant manipulators performing high payload operations. Al-Dois *et al.* (2013) used robot motion control to optimize the layout of machines in an industrial workcell. The task-time based optimization approach minimizes the path traveled by the robot, which in turn changes the machine loading/unloading points and thus obtained a new layout according to the assigned task. Izui et al. (2013) designed a multi-objective layout optimization approach for robotic workcell. The layout design problem is based on the operation time minimization, which also minimizes the layout area to save the floor area in the industry. Zhang and Fang (2013) developed a layout optimization approach for small part assembly in 3C industry. They minimized the cycle time for a robotic workcell using response surface method and evaluated the position and orientation of the machine about robot. Nageshwaraniyer *et al.* (2013) presented a hybrid layout design approach using forecast windows.

Lim *et al.* (2016) presented the latest approach in the field of layout design for robotic workcell by introducing nature-inspired algorithms. They formulated the multi-objective layout optimization problem and solved it by using NSGA II algorithm considering layout area, operation time and manipulability minimization as their criteria. They also used five nature-inspired algorithms, i.e., genetic algorithm, differential algorithm, artificial bee colony, charge search system, particle swarm algorithm to solve the problem based on three criteria simultaneously.

2.5. Robot Trajectory Analysis

The trajectory planning is defined as generating the location of points in coordinate space followed by the end-effector while avoiding obstacles (Bayle *et al*., 2000). Robot controller has the coded trajectory plan that governs the motion of the end-effector and control the angular movement of the joint actuators. The trajectory plan is based on the mathematical profile which is formulated according to desired objective. Based on the profile, trajectory plan of the end-effector is generated which is a time-based function of joint parameters that result in the controlled movement of manipulator (Saha, 2008).

The trajectory planning involves several crucial factors that need computational and mathematical knowledge such as obstacle avoidance, constraints due to geometry and size of objects, collision-free motion, avoiding singular configurations, and workspace constraints. The trajectory can be of in two ways; first is the point-to-point, in which the end-effector moves from start to endpoint in coordinate space without involving in any live operation; second is the continuous path trajectory in which end-effector travels through a well-defined path while performing the live operation (Craig, 2005).

The trajectory planning can be formulated in two spaces: joint space and Cartesian space. Generally, data is given in Cartesian coordinate space, and the joint space trajectory planning requires converting it into equivalent joint configuration, while the results from the Cartesian space planning requires conversion of joint configuration for controller information. The trajectory planning in Cartesian space is mathematically complex and requires calculation of inverse kinematics and Hessian matrices. The singularity problem is prior in the Cartesian space and constraints are difficult to handle. To manage these problems trajectory planning generally accomplished in joint space, which determines the trajectory points regarding the joint configuration of the manipulator (Haddad *et al.*, 2007; Liu *et al.*, 2013)

The prediction of end-effector flight by joint space trajectory planning is uncertain and increase the chances of collision and undesirable motion. However, the Cartesian space trajectory planning gives the exact geometric path but fails for the singularity and out of bound configurations. The trajectory planning is subjected to several constraints such as kinematic and dynamic constraints, such as, maximum, and minimum bounds of velocity, acceleration, jerk, and torque. other constraint is due to the obstacle present in the workspace of the robot that omits several optimal configurations and increases the computational load of the problem.

Various trajectory-planning approaches have been developed in past few decades, and more, new approaches are adding. Developed approaches differs by optimization criteria, smoothness level of trajectory, computational load, and solution accuracy. The first requirement for each trajectory planning problem is smooth and hassle-free traversing of the path by the end-effector. Several algorithms have been proposed, which offer jerk-less trajectory planning algorithms (Liu *et al.* 2013, Gasparetto *et al.*, 2012; Gasparetto and Zanotto, 2007; Rubio *et al.*, 2009; Parsa and Daniali, 2010).

The smoothness of the trajectory is generally measured in terms of minimum joint jerk and sometimes regarding the joint torques (Zhihong, 2005). The smooth trajectories are favorable for long manipulator life due to minimum vibration and chatter during motion. However, the smooth trajectories come at the stake of increased traveling time caused due to reduced velocity of the manipulator. Therefore, the optimization of time is of equal importance to maintain the productivity. The balance between two parameters is essential for obtaining efficient and productive manipulator motion (Gasparetto *et al*., 2012).

The widely used profile for trajectory planning is the polynomial interpolation function. It is the time-based polynomial function for joint parameters. The polynomial curve becomes smoother as its order increases, but at the higher order the polynomial curve shows oscillations at the edges over the set of equispaced interval along the entire path and this problem is known as Runge's phenomenon. The trajectory planning is performed as the fitting of polynomial between a fixed number of specific points in Cartesian space known as knots.

Angeles (2003) shows the fitting of the piecewise polynomial curves of different orders. Initially, trajectory accelerates quickly and at the end rate of change of acceleration decreases. Three different polynomial curve combinations were applied to develop the smooth and efficient trajectories. Due to the break at the intervals, the polynomial combination suffers from discontinuity. Instead, the single higher order polynomial curve shows continuous and smooth trajectory but is computational expensive (Spong and Vidyasagar, 2004). The smoothness of trajectory depends on the mathematical profile and usually spline profile is mostly used in this case.

In spline interpolation, the polynomial of lower order has been fitted between the knots that have continuous velocity, acceleration, and jerk at each knot points. Paul (1972) introduces this approach for cubic polynomials that were smooth and have small joint displacement overshoot. Lin *et al.* 1983 developed a trajectory-planning algorithm based on the cubic polynomial. They take the minimum time criterion and uses flexible polyhedron search approach (Nelder and Mead, 1965) to iterate the path having minimum time considering the kino-dynamic constraint. Later, in 2008, trajectory planning method is developed for parallel planar manipulator (Hu *et al.*, 2008). The online trajectory planning based on cubic spline was developed using only minimum number of knots near the instantaneous position instead of entire path (Keith and Chand, 1985)

Another efficient profile used in trajectory planning is B-spline, in which the polynomial curve is stretched towards knots instead fitting in between them (Paul and Thompson, 1987). B-spline is smooth and computationally efficient and it has been applied in the online trajectory planning problem with kinematic constraints at each knot. Gaspareto *et al.* (2007) used the B-spline to plan the offline trajectory planning for the optimal time-jerk criteria. They developed the first dual objective function which minimizes together the time and jerk according to assigned weightage. The balance of the weighting factor is the crucial point for this approach. The trajectory-planning problem deals with the manipulator in motion hence it is subjected to both kinematic and dynamic constraints along the trajectory.

Chettibi *et al.*, (2004) solved the offline trajectory planning problem under kino-dynamic constraints and developed cost-based criteria considering the production time, actuator efforts and the power consumed. Gradually, by the evolution of trajectory planning approach, constraints on obstacle environment has been added to the problem formulation with multi-objective optimization approach (Saravanan *et al.*, 2009). Such methods specifically emphasize on global optimum solution search algorithms like NSGA-II, MODE, which can handle geometric constraints in Cartesian space and omits initial feasible solution requirement. However, this approach suffers from the detailed modeling of robots and machines for checking collision and has used corner points of the sphere and rectangular prism shape for simulation.

Recently developed discretized bi-level trajectory optimization searches the position of end-effector at each trajectory step under highly constrained environment. It generates a more realistic configuration of the robot and simultaneously takes advantage of avoiding singularities and obstacle in Cartesian space approach (Menasri *et al.*, 2015). This kind of approach has two significant benefits: (i) it avoids the singularity problem because forward kinematics continuously checks for the possible joint configuration under imposed constraints and (ii) infinite number of configurations can be iterated because of the redundancy of the robot manipulator. This approach is advantageous to trajectory planning which can avoid undesirable solutions and to obtain initial feasible solution for optimization.

Trajectory planning of a robotic manipulator is defined as searching for optimal flight points with in configuration space between two fixed points, traced by the centre point of end-effector. Difference occurs when the trajectory passes through a fixed via-point, which divides the trajectory into two parts. Trajectory planning with via-point location optimization adds benefit of optimum global search with redundancy. Moreover, the via-point enable wide range of motion in space that can solve nonlinear trajectory planning problems satisfying the kinematic parameters of robot. Variants of the trajectory planning occur in the way of generating solutions. Such as using fuzzy logic to control the trajectory generation and obstacle avoidance. The importance of redundant robot manipulators possessing more degree of freedom to perform a task has been noticed. Due to redundancy, the flexibility of robot increases that enhances its collision avoidance ability.

Logical methods such as fuzzy clustering, neural networking have been used to achieve singularity free trajectory planning in Cartesian space (Parsa *et al.*, 2010; Agarwal, 2012). The solutions of the trajectory planning problem from the logical algorithms is a remarkable approach for detecting a collision in the sensory environment which also have the facility of error backtracking, The smoothness of the trajectory, effects the life of actuators, as vibrations and jerk during robot movement lead to wear and tear of joints and decreases the overall running span of the robot. Searching for smooth trajectory compels the robot joint to take the minimum possible angular acceleration and optimizes the joint motion, which increases the trajectory length. Thus, increases the overall production time. For a solution, the time-jerk based optimization has been taken to optimize the traveling path of the robot end-effector while maintaining the smoothness under kino-dynamic constraints (Liu *et al.*, 2013; Yang *et al.*, 2013).

2.6. Developments in Multi-Robot Workcell

Multi-robot systems are widely prevalent in an automotive assembly line, where coordination and collaborative task capabilities of the multirobot system are extensively utilized. The multirobot systems have been proposed three decades before. Now, there are various types of multirobot system such as semi-anthropomorphic dual arm manipulators, dual industrial manipulator, double, single arm manipulator system, mobile multi-arm robot, multi-arm manipulator system (Smith *et al.*, 2012). Multirobot system can also be a team of multiple mobile robots.

The design and planning of the multirobot system as a workcell are far more complicated than the single robot workcell. However, the profit of minimizing the number of single robot cell and replacing them by multirobot workcells is a point of motivation. Various applications such as automotive assembly lines, packaging, and workcell design require planned localization of robots and machines that result in efficient and productive multirobot system. The common problems in the multirobot system vary from communication between robots and their teams, system architecture, task planning, control of robot teams, multirobot localization and mapping, motion coordination, reconfigurable multirobotic system, and autonomous learning (Arai *et al.*, 2002). In this section, the literature regarding the multirobot workcell having stationary manipulator arms have been reviewed.

Farinelli *et al.* (2004) presented the classification of the multirobot system based on robot coordination. The coordination is the crucial factor for the successful working of the multirobot system because of the complexity of the task sequence. The coordination among the robots in the workcell consider several factors such as rationale design, basic functionality and technologies used, tasks performed by the robot, and the working domain. The multirobot system is useful when the robots can perform multiple operations together with same capabilities.

The coordination of the multirobot system is demonstrated by the application of the two arm type manipulators for optimal sculpting (Owen *et al.*, 2008). The robotic machining is an alternative to CNC machining, and the robotic workcell should be designed in such a way that the robots should act as a single unit and perform the machining operation more efficiently than CNC machines. The trajectory of the robots is optimized by using the Jacobian null space and minimizing the system compliance factor. Chiddarwar and Babu (2011) developed a coordinated path planning approach for multiple robots using a dynamic path modification sequence.

The offline coordination system is a two-phase decoupled method in which the collision-free path has been determined with obstacle avoidance. The coordination among the robots is established by resolving conflicts based on path modification. Tao and Liu (2011) developed a vertical multirobot workcell layout planning method for aerospace manufacturing industry by dividing the workspace of the robot in to sector and solving a nonlinear optimization problem. The objective of the problem is to increase the overlapping of the workspace of robots without collision to define the location of each robot in three-dimensional space.

The multi-arm robotic manipulator system is most popular in the industry due to their human-like capabilities to do industrial operations. Basile *et al.* (2012) planned the motion of arm type manipulators for the cooperative task performance that involves localization and orientation of the job. The multi-arm robotic system has kinematic parameters, which were developed, and the new taxonomy of a cooperative multirobot system for industrial application has been formulated. Pellegrinelli *et al.* (2014) developed the offline planning approach for multirobot cell design and motion planning. They presents an integrated approach for layout design of spot welding workcell. In this approach, the systematic problem formulation and solving through the multistage heuristic method generates optimal combination of both robot design and plan for the end-effector motion.

Dogar *et al.* (2015) developed a multi-robot grasp planning approach for an assembly task. Several robots are programmed in a coordinated way such that the robots can perform an assembly operation. The problem is to find the optimal configuration of robot to grasp assembly parts in a cooperative assembly operation. The satisfying constraint problem is formated considering collision and transfer constraints that are solved by dividing the problem into smaller independent problems that can be solved conveniently. CAD-based automated assembly planning by multi-robot system has been developed for reconfigurable production system producing multivariate products. The CAD modeling of the products and production system helps to analyse peculiar parts of the assembly process and is known as production graph (Michniewicz *et al.*, 2016). Multirobot task allocation is an essential factor for the efficient performance due to highly complex multi operations system.

Sarker *et al.* (2014) presented a study in which they found that the local communication strategy could improve system performance better than the centralized communication system. They employed attractive field model framework a generic selforganized division of labor derived from ant, human social system. Bartelt *et al.* (2014) presented a tool-based path planning approach for the multirobot system. The path planning in the multirobot system produces difficulty for layout design. The tool-based planning omits the difficulty due to robot workspace limitation and allowing coordination between robots during tool transfer.

Multi-robot system working have their task distributed according to the work piece location and operation performed. The cooperation between the robots is an essential requirement in term of timing and task to be performed. Hernansanz *et al.* (2015) developed a multirobot cooperation platform for task-oriented teleoperation controlled by the human operator. The approach selects the robot dynamically and manage to execute task smoothly. The pick and place task in the packaging line involves multiple robots, and the coordination among them is essential for the proper packing in which pattern variation and the speed put several hurdles.

Huang *et al.* (2015) addresses the problem of multirobot coordination in the pick and place packaging line aims to reduce the parts left on the conveyor and thus to increase the assembly efficiency. The key feature is to combine the part dispatching rules to the coordinate robot strategy. Lope et al. presented the multi-robot task distribution problem and presented a solution inspired by the rules for division of labor among social insect. The group of robot is organized in a distributed or decentralized approach in which robots are responsible for autonomously selecting a particular task. Jose and Pratihar (2016) presented the solution of the task allocation problem with collision-free path planning of multirobot system. The centralized multirobot inspection system has been optimized by using genetic algorithm and A^* algorithm.

2.7. Concluding Remarks

The preceding discussion in the present chapter reveals the significance of research work in the field of robotics. The research opportunities are vast and several challenging problems are need to be tackle.

The literature review find that, two decades ago inefficient modelling approaches were used to solve both layout and trajectory planning problems. Such approaches run on number of assumptions and percentage error in the results were very high. A decade further, professional planning packages were used for developing better solutions. Professional packages have interactive GUI and multifunction facility, however they have high licensing cost and robot specific working environment. Presently, user specific packages have been developed, providing open environment for programming with any robotic manipulator. However, the challenge before these planning packages is to compete with the existing professional packages. For this task much more research work in this field is required. This has motivated to undertake research work in the field of robotic simulation and planning.

Several possibilities for research in field of robotics were outlined and finally the following topic was chosen for further study as:

"Layout Design and Trajectory Planning of Robotic Workcell using Point Cloud Approach".

The outcome of the systematic research work on the above mentioned topic is reported in the following chapters.

The next chapter presents the detailed working procedure and the mechanism of point cloud simulation approach developed to solve various problems in the field of robotic workcell design and planning.