Original Article

APPLICATIONS OF DISCRETE MATHEMATICS IN REPRESENTATION AND PATH PLANNING OF A ROBOTIC SYSTEM

Suseela Midde[1*], Dr. G. V. V. Jagannadha Rao[2]

[1*]Lecturer, Dept of Mathematics and Statistics Ch. S. D. St. Theresa's College For Women (A), Eluru, Andhra Pradesh Research Scholar, Department of Mathematics, Kalinga University, Naya Raipur.
[2]Associate Professor, Department of Mathematics, Kalinga University, Naya Raipur.
*Corresponding Author: Suseela Midde

DOI: 10.47750/pnr.2023.14.03.39

Abstract

Robots are widely used in present day situations. Since a robot consists of number of systems with different mechanisms, it needs to be planned and programmed systematically. Discrete Mathematics helps us in designing and functioning of a robotic system. For instance Robotic arm is a type of linkage and the study of which is a part of Discrete Geometry. Scheduling tasks to be completed by a set of machines use the concept of Bin-packing problem, which is a part of Discrete Optimization. Discussing functions like these, motion and path planning of the robotic system is the main concern of any industry to adapt new and modern technologies in its functioning. Literature surveys reveal that there a worldwide attempts to develop robotic systems with minimum errors and maximum efficiency. This Paper includes the usage of the Concepts of Discrete Mathematics to have a path planning for a Robotic System to work with high efficiency.

Key Words: Discrete geometry, Robotic arm, Discrete optimization, Path planning

INTRODUCTION

Robotics is no more a study restricted to defined steps of machine rather it becomes more independent and self-decision making machine with the thinking power called artificial intelligence. Robots are used in many industrial applications like assembly operations, spraying, painting, grinding, welding, pick-and-place operations, etc. Robots perform repetitive tasks with accuracy and higher efficiency. Fully autonomous robots are still under research, which can take decision on their own to perform any task and are able to perform the given tasks efficiently and effectively. In many fields where technical support is required, manhandling is either dangerous or it may not be possible. In such cases, three or more arm manipulators are generally used. Now a day they are in great demand to speed up the automation processes. Two-link robotic arm is used to locate any location, which is the essential movement in real world situations. These Robotic arms are used in small to big scale applications, namely, chip fabrications to huge mechanical actuators used in chemical processes. In these cases, the motion profile of the robot remains the same throughout the whole operation. Therefore, searching an optimal robot arm movement is a favorable solution to those problems. This paper aims to develop autonomous robot, which can compete with them on any level, and to perform tasks as human beings perform with accuracy and efficiency.

Path planning is one of the important objectives of developing autonomous robots. There is evidence that the success of the intelligent human path planner uses some intelligent, automated motion planning in Robotics that would be useful to investigate. Artificial intelligence can make it possible for robots to perform complex tasks easily. Advance computers have shown themselves to be extremely capable in many application areas, and have transformed the world in which we live. [Banga Vijay Kumar 2011]

REPRESENTATION OF GRAPH:

For the path planning of a robotic system, a plan can be drawn using the graphical representation which is part of discrete mathematics. Diagrammatic (graphical) representation of a graph is very convenient for visual study, but it is practically feasible only when the number of vertices and edges of the graph is reasonably small. So, we need some other reasonable and simple ways to represent graphs consisting large number of vertices and edges. These representations are also used in computer programming. Matrix representations of the undirected and directed graphs are discussed here below.

MATRIX (ADJACENCY MATRIX) REPRESENTATION:

The adjacency matrix is commonly used to represent graphs for computer processing. In fact, it is more suitable for dense graphs (in which each vertex of a graph has almost all the possible edges). In such representation, an $n \times n$ Boolean (1,0) matrix is used where a 1 at position (u,v) indicates that there exists an edge from vertex u to v, and a 0 at position is undirected , then its corresponding adjacency matrix will be symmetric

Matrix representation of undirected graph If an undirected graph G consists of n vertices (assuming that the graph has no parallel edge), then the adjacency matrix of G is an $n \times n$ matrix $A = [a_{i,j}]$ and defined as follows:

$$a_{i,j} = \begin{cases} 1 \text{ if there is an undirected edge between vertices } v_i and\ v_j \\ 0, \text{if there is a directed edge between vertices } v_i and\ v_j \end{cases}$$

Here are some observations from matrix representation of undirected simple graph:

(a) $a_{i,j} = a_{j,i}$ for all i and j, i.e., the matrix is symmetric.

(b) Diagonal elements of the matrix are zero (0) (as the simple graph possesses no self -loop).

(c) The degree of a vertex (represented by row number) is the sum of the 1s in that row.

(d) Let G be a graph with n vertices: $v_1, v_2, \ldots v_n$ and A be the adjacency matrix of G. Let B be the matrix computed as follows:

$$B = A + A^2 + A^3 \ldots\ldots + A^n \,(n > 1)$$

Now, B is connected if and only if B has no zero entry.

(ii) Matrix representation of directed graph If a directed graph (digraph) G consists of n vertices (assuming that the graph has no parallel edge), then the adjacency matrix of G is an $n \times n$ matrix $A = [a_{i,j}]$, and is defined as follows:

$$a_{i,j} = \begin{cases} 1 \text{ if there is a directed edge from vertex } v_i \, to \text{ vertex } v_j \\ 0, \text{if there is no edge between } v_i \, and \, v_j \end{cases}$$

Some observations from the matrix representation of directed simple graph

(a) $a_{i,j} \neq a_{j,i}$ for all i and j, i.e., if there is an edge from vertex v_i to v_j, it means that there will not necessarily exist an edge from v_j to v_i. Hence, the represented matrix A is not necessary to be symmetric .

(b) Diagonal elements of the matrix A are zero (0) (as the simple graph has no self -loop).

(c) The sum of 1 in any column j of A is equal to the number of edges directed towards vertex v_j, i.e., it is the in-degree of vertex v_j

(d) The sum of 1 in any row i of A is equal to the number of edges directed away from vertex v_i, i.e., it is the out-degree of vertex v_i.

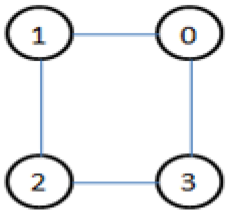
We have to note that to check whether two graphs are isomorphic or not, we may take adjacency matrix representation of the given two graphs. If both the matrices (say A and B) have the same size, then match a row of the first matrix A with any row of the second matrix B. If matching is found, then cross out those rows from A and B, and go for matching the next row of A , and so on until all the rows of A and B are crossed out. If all the rows are crossed out, then we conclude that both the graphs are isomorphic, otherwise not.

LINKED LIST (ADJACENCY LIST) REPRESENTATION

In such representation, each node u in the graph (assuming that the graph has no parallel edges) contains a linear linked list , consisting of the vertices directly reachable from u. Thus, if n vertices are present in a graph G, then the number of lists will also be n, i.e., the n rows of the adjacency matrix are represented by n lists .In particular, linked representation is a particularly good idea for sparse graphs.

The adjacency matrix and the linked list representation of the following graphs are given in figure.

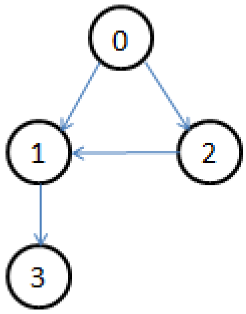
First, we present the adjacency matrix representation of the graphs in figures (a) and (b) respectively. [3]



	0	**1**	**2**	**3**
0	0	1	0	1
1	1	0	1	0
2	0	1	0	1

3	1	0	1	0

Figure (a) : Adjacency Matrix Representation of Undirected Graph



	0	**1**	**2**	**3**
0	0	1	0	1
1	1	0	1	0
2	0	1	0	1
3	1	0	1	0

Figure (b) : Adjacency Matrix Representation of Directed Graph

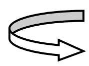
The MPD protocol proposed by Ravindranadh [7] is programmed using MATLAB for calculating robot paths moving through a time-varying environment using position information of the first robot and reachable cone of the second robot to determine possible collisions at a future time and thus in a position to avoid them by deviating from the nominal path. Here, one robot is assumed a dynamic obstacle and the other robot's collision-free path is found. In this method, the position of the reachable cone of the robot is mapped in a configuration time-space graph at discrete time intervals taking time in the z-direction. The possible collision with the obstacle is found by checking the intersection of the position circle of the obstacle (C2) with the base circle (C1) of the reachable cone of the mobile robot at that discrete time interval.

If $C1 \cap C2 = \emptyset$, there is no possible collision, and the robot path is free.

If $C1 \cap C2 \neq \emptyset$, then there is a possible collision and path deviation algorithm is to be implemented. [Pravin Kalla (2021)]

In General Robotic kinematics problems can be solved by different mathematical techniques. Most commonly used mathematical techniques are algebraic, iterative, matrix and geometrical representations. A Geometric technique is based on the coordinate system to represent the different links of the manipulator to obtain the solution of joint angle movement with rotator joints.

Table 1: Symbolic movement of the robot

Type	Notation	Symbol	Description
Revolute	R		Rotary motion about an axis

Prismatic	P	↕	Linear motion about an axis

Based on the movements of the robot the types of robots are classified as Cartesian, Cylindrical, Spherical, SCARA (Selective Compliance Assembly Robot Arm) and articulated Robots.

Table 2: Robotic Classification based on the major axes

Robot	Axis1	Axis 2	Axis 3	Total revolute
Cartesian	Prismatic	Prismatic	Prismatic	0
Cylindrical	Revolute	Prismatic	Prismatic	1
Spherical	Revolute	Revolute	Prismatic	2
SCARA	Revolute	Revolute	Prismatic	2
articulated	Revolute	Revolute	Revolute	3

Matrix technique uses 4 x 4 homogeneous matrices for obtaining joint solutions for various manipulators. A generalized 3x3 rotation matrix is used to represent the rotational operations of the coordinate frame to define the position of the manipulator in the three-dimensional space. (Banga vijay Kumar 2011)

A learning process of an automated robotic system is specified as start, end and intermediate stages of motion. An automated robotic system must possess ability to get programmed for variety of arm movement for all type of sensor data. The robotic arm movements can be controlled by efficient algorithms using simple programming languages.

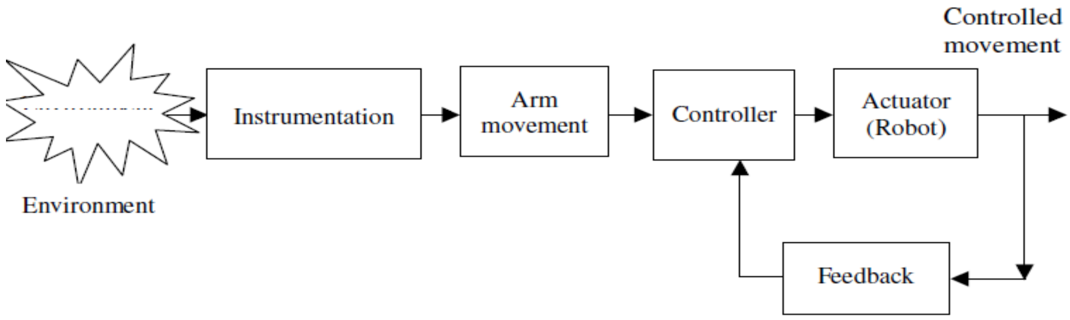


Fig. 1.1 block diagram of an automated robotic system.

To move robotic arm for particular task, programmer generates appropriate sequence of steps taking information of its environment and actuates on the basis of system databases. To make programming simple it is required to move manipulator to the task level. The manipulator moves according to the task of the task of the robotic arm to achieve the target. In robot path planning care should be taken that it moves free between start and target points without any collision with obstacles in the environment, which is a key factor of the motion planning. This is the core problem for the researcher in an automated robot to achieve collision free path planning in the wake of non- deterministic parameters like energy consumption, friction and settling time, etc.

Path Planning of the robotic System

A robotic arm system is characterized by degree- freedom. Performance of the robotic arm is also indexed in terms of "speed". Motion planners operate on- line and off- line. On –line planners are used in an automated robotic system, which operate in "real time". A" real time" automated robotic system needs to have very high speed as the path planner iterates its planning process several times. An off- line planner requires an operator for path planning. Literature survey reveals that the recently many path planning methods have been developed. Path planning can be done very conveniently for robotic arms with lower degree- of- freedom.

Complex industrial robotic system posses higher degree of freedom (DoF). For example with 4 DOF may have one translational and three rotational motions. The motion of robotic arm is characterized and constrained by number of joints and their types. Every joint contributes one DOF. The concept of configuration space provides for meaningful solution for motion and path planning for the robotic arm having multiple DOF.

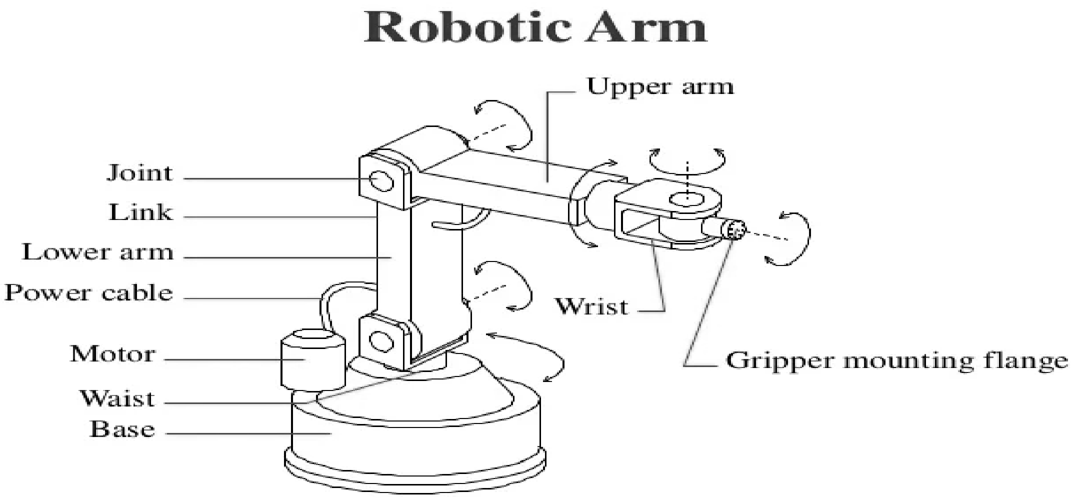


Fig 2; Components of Robotic Arm

Basically, robotic arm is configured in terms of three parameters, which are defined over a local coordinate system. Out of three parameters; two parameters contribute local origin in terms of x-y positions. Third parameter gives the details about the orientation of local coordinates system. It is also known as local frame. Each coordinate gives a unique and possible robot configuration depending upon the number of DOFs. For example, a two DOF robot can be characterized by its two dimensional configuration space. Three DOF robotic arm can be characterized by its dimensional configuration space. Four DOF robotic arm can be characterized by its four-dimensional configuration space.

We introduce the radius vector of points Oi (i =1,2,3,4) in the i-local coordinate system:

$$R_i = [x_i \ y_i \ z_i \ 1]^T$$

The communication of radius vectors in the coordinate system i-1 and i by means of the transition matrix

$$A_{i-1,i} : R_{i-1} = A_{i-1} R_i$$

The transition matrix from point O0 to point O1 is of the form:

$$A_{01}=\begin{bmatrix}\cos(q_1) & -\sin(q_1) & 0 & 0\\ \sin(q_1) & \cos(q_1) & 0 & 0\\ 0 & 0 & 1 & 0\\ 0 & 0 & 0 & 1\end{bmatrix}$$

The matrices of the transition from point O1 to point O2 and of the transition from point O2 to point O3 are of the form:

$$A_{12}=\begin{bmatrix}-1 & 0 & 0 & 0\\ 0 & 0 & 1 & 0\\ 0 & 1 & 0 & q_2\\ 0 & 0 & 0 & 1\end{bmatrix}, A_{23}=\begin{bmatrix}1 & 0 & 0 & 0\\ 0 & 1 & 0 & 0\\ 0 & 0 & 1 & q_3\\ 0 & 0 & 0 & 1\end{bmatrix}$$

The transition matrix from point O3 to point O4 is of the form:

$$\mathrm{A}_{34}=\begin{bmatrix}\cos(q_4) & 0 & -\sin(q_4) & 0\\ \sin(q_4) & 0 & \cos(q_4) & 0\\ 0 & -1 & 0 & s\\ 0 & 0 & 0 & 1\end{bmatrix}$$

The transition matrix from point O0 to point O2 is defined as the product of matrices:

A 02= A01 A12,

$$A_{02}=\begin{bmatrix}-\cos(q_1) & 0 & -\sin(q_1) & 0\\ -\sin(q_1) & 0 & \cos(q_1) & 0\\ 0 & 1 & 0 & q_2\\ 0 & 0 & 0 & 1\end{bmatrix}$$

The transition matrix from point O0 to point O3 is obtained as the product of matrices:

A 03 =A 01 A 12 A 23,

$$A_{03}=\begin{bmatrix}-\cos(q_1) & 0 & -\sin(q_1) & -q_3\sin(q_1)\\ -\sin(q_1) & 0 & \cos(q_1) & q_3\cos(q_1)\\ 0 & 1 & 0 & q_2\\ 0 & 0 & 0 & 1\end{bmatrix}$$

The transition matrix from point O0 to point O4 is defined as the product of matrices:

A04 = A01 A12 A23 A34,

$$A_{04} = \begin{bmatrix} -\cos(q_1)\cos(q_4) & \sin(q_1) & \cos(q_1)\sin(q_4) & -q_3\sin(q_1) - s\sin(q_1) \\ -\cos(q_4)\sin(q_1) & -\cos(q_1) & \sin(q_1)\sin(q_4) & q_3\cos(q_1) + s\sin(q_1) \\ \sin(q_4) & 0 & \cos(q_4) & q_2 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

The origin of coordinates O4 associated with gripping device in the fixed coordinate system O0 associated with a base of rack is defined by coordinates:

$$x_{04} = q_3\sin(q_1) - s\sin(q_1),$$
$$y_{04} = q_3\cos(q_1) + s\cos(q_1),$$
$$z_{04} = q_2$$

We denote the coordinates of the gripper fingers in a coordinate system O4 as (x4 y4 z4), then in a fixed coordinate system O0 the coordinates of the gripper fingers can be represented in the form of

$$x_{04} = -q_3\sin(q_1) - x_4\cos(q_1)\cos(q_4) + z_4\cos(q_1)\sin(q_4) - s\sin(q_1) + y_4\sin(q_1)$$
$$y_{04} = q_3\cos(q_1) - x_4\sin(q_1)\cos(q_4) + z_4\sin(q_1)\sin(q_4) + s\cos(q_1) - y_4\cos(q_1)$$
$$z_{04} = q_2 + x_4\sin(q_4) + z_4\cos(q_4)$$

Robotic System's movements are calculated using the above formulae.

CONCLUSION

Creating an automated robotic system and to define its path is a challenging task. It requires various skills like programming, mechanical modeling of robotic systems, electronic devices, their interfaces and control. An integrated design based on above skills and engineering areas will help in realization of such an automated robotic system with specifically defined path planning to assign different tasks. An automated robotic system must plan its movement optimally. So, to plan such movements, we have the mathematical representation in forms of graphs and matrices.

REFERENCES

1. A kuzmina (2020) the use of mathematical models to improve the effective functioning of enterprises issn 0974-3154

2. Banga, Vijay Kumar (2011), Performance Improvement in Robotic Arm Movement using Fuzzy and Genetic Algorithms. http://hdl.handle.net/10603/301582.

3. Ganiyat salawu et.al (2020) mathematical modelling and simulation of throughput in a robotics manufacturing system

4. Igor manoel garbim cruz (2022) symbolic and numerical mathematical modeling of the jacobian of the yaskawa motoman - gp7 robot using matlab© issn: 2831 – 6789

5. Ivanov, s.e., 2016 "mathematical modeling of nonlinear dynamic system of the truck crane," contemporary engineering sciences, 9, pp. 487-495.

6. Praveen, Kalla [2021], Robot path planning techniques a comparative study, Ph.D Thesis, Andhra University, India. http: //hdl.Handle.net /10603/395093.

7. Ravindranadh, P. (2006) Some studies on mobile robots path planning in the presence of static and dynamic obstacles, Ph. D. Thesis, Andhra University, India.

8. Sergei evgenievich ivanov et.al (2020) mathematical modeling of the dynamics of 3-dof robot-manipulator with software control

9. S. K.Chakraborthy and B.K.Sarkar " Discrete Mathematics" Text Book Published in 2011

10. Sekar, s., vijayarakavan, m., 2015, “observer design of singular systems (robot arm model) using leapfrog method,” global journal of pure and applied mathematics, 11(1), pp. 68-71.

11. Wilson, j., charest, m., dubay, r., 2016, “non-linear model predictive control schemes with application on a 2-link vertical robot manipulator,” robotics and computer-integrated manufacturing, 41, pp. 23-30.

12. Yuri zhukov et.al (2019) mathematical modeling of the robot motion control system for monitoring deep sea communications