Path Planning of A Parallel Manipulator Using A* Algorithm

A. Harish [1], Dr.Chandrashekhar.A [2], Dr.G. Satish Babu [3]

Asst. Prof., Mechanical Engineering Department, CMR College of Engineering & Technology, Hyderabad, India[1],

Asst. Prof., Faculty of Science & Technology, ICFAI Foundation for Higher Education, Hyderabad, India[2].

Professor, Department of Mechanical Engineering, JNTUH College of Engineering, Hyderabad, India[3]

ABSTRACT:

Manipulators are now used in many fields as automation became a most important part of all the fields. Manipulators are being used mostly in many hazardous places where it is dangerous or laborious for a man to work in that places like industries, hospitals, rescue operations etc. Manipulators have a great application in industries where they are used for carrying loads, pick and place and assembling. Two types of manipulators are in use they are Serial and Parallel manipulators. In this paper I am going to deal with Parallel Manipulators and their advantages over serial manipulator and also do path planning for a parallel manipulator that is used in industries using an artificial intelligence algorithm called A algorithm. This algorithm helps a robot to find the optimal path on its own depending on the obstacles present and layout of the industry.*

1. INTRODUCTION:

Robots or Manipulators are playing a very important role in fulfilling human needs. Generally, manipulators are of two types Serial and Parallel. Serial manipulators are the most common industrial robots. They are designed as a series of links connected by motor-actuated joints that extend from a base to an end-effector. Often they have an anthropomorphic arm structure described as having a shoulder, an elbow and a wrist. Serial robots usually have six joints, because it requires at least six degrees of freedom to place a manipulated object in an arbitrary position and orientation in the workspace of the robot. n its most general form, a serial robot consists of a number of rigid links connected with joints. Simplicity considerations in manufacturing and control have led to robots with only revolute or prismatic joints and orthogonal, parallel and/or intersecting joint axes.

Whereas a parallel manipulator is a mechanical system that uses several computer-controlled serial chains to support a single platform, or end-effector. Perhaps, the best known parallel manipulator is formed from six linear actuators that support a movable base for devices such as flight simulators. This device is called a Stewart platform or the Gough-Stewart platform in recognition of the engineers who first designed and used them. A parallel manipulator is designed so that each chain is usually short, simple and can thus be rigid against unwanted movement, compared to a serial manipulator. Errors in one chain's positioning are averaged in conjunction with the others, rather than being cumulative. Each actuator must still move within its own degree of freedom, as for a serial robot. However, in the parallel robot the off-axis flexibility of a joint is also constrained by the effect of the other chains. It is this closed-loop stiffness that makes the overall parallel manipulator stiff relative to its components, unlike the serial chain that becomes progressively less rigid with more components.

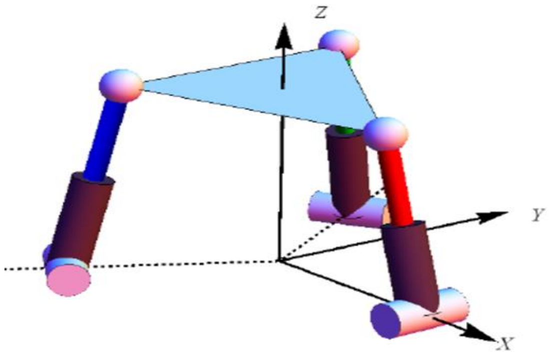

Fig.1 Parallel Manipulator

2. ADVANTAGES OF PARALLEL MANIPULATOR OVER A SERIAL MANIPULATOR

Most robot applications require rigidity. Serial robots may achieve this by using high-quality rotary joints that permit movement in one axis but are rigid against movement outside this. Any joint permitting movement must also have this movement under deliberate control by an actuator. A movement requiring several axes thus requires a number of such joints. Unwanted flexibility or sloppiness in one joint causes a similar sloppiness in the arm, which may be amplified by the distance between the joint and the end-effector: there is no opportunity to brace one joint's movement against another. Their inevitable hysteresis and off-axis flexibility accumulates along the arm's kinematic chain; a precision serial manipulator is a compromise between precision, complexity, mass (of the manipulator and of the manipulated objects) and cost. On the other hand, with parallel manipulators, a high rigidity may be obtained with a small mass of the manipulator (relatively to the charge being manipulated). This allows high precision and high speed of movements, and motivates the use of parallel manipulators in flight simulators (high speed with rather large masses) and electrostatic or magnetic lenses in particle accelerators (very high precision in positioning large masses).

A further advantage of the parallel manipulator is that the heavy actuators may often be centrally mounted on a single base platform, the movement of the arm taking place through struts and joints alone. This reduction in mass along the arm permits a lighter arm construction, thus lighter actuators and faster movements. This centralisation of mass also reduces the robot's overall moment of inertia, which may be an advantage for a mobile or walking robot. As their speed of action is often constrained by their rigidity rather than sheer power, they can be fast-acting, in comparison to serial manipulators. All these features of parallel manipulators make them more suitable for a work like load carrying which requires more rigidity.

3. PROBLEM STATEMENT:

A parallel manipulator used in industries is considered for carrying loads. For this manipulator an algorithm has to be fed so that it can plan its path based on the layout of the industry. This manipulator has to consider all the obstacles that come on its way and should be able to choose path for different layout of the industry and different setup of obstacles.

4. METHODOLOGY:

For path planning most of the widely used algorithm is A* algorithm. A* is an artificial intelligence algorithm which can make the manipulator think like a human considering all the possibilities and find the optimal path from all the possible paths. MATLAB is used for coding and running the A* algorithm.

5. A-STAR ALGORITHM:

ARTIFICIAL INTELLIGENCE:

According to the father of Artificial Intelligence, John McCarthy, it is "The science and engineering of making intelligent machines, especially intelligent computer programs". Artificial Intelligence is a way of making a computer, a computer-controlled robot, or a software think intelligently, in the similar manner the intelligent humans think. AI is accomplished by studying how human brain thinks, and how humans learn, decide, and work while trying to solve a problem, and then using the outcomes of this study as a basis of developing intelligent software and systems.

Philosophy of AI

While exploiting the power of the computer systems, the curiosity of human, lead him to wonder, "Can a machine think and behave like humans do?"

Thus, the development of AI started with the intention of creating similar intelligence in machines that we find and regard high in humans.

Goals of AI

To Create Expert Systems − The systems which exhibit intelligent behaviour, learn, demonstrate, explain, and advice its users.

To Implement Human Intelligence in Machines − Creating systems that understand, think, learn, and behave like humans.

A-STAR ALGORITHM:

Hart et al. (1968) developed a search strategy, called A*, to solve for minimum cost paths. The A* approach differs from other methods as it incorporates an estimate of the cost of 'path-completion.' For certain classes of

estimating functions, A* will find the optimal path. Over the past 50 years, there has been continued and sustained interest in developing faster algorithms for solving the shortest path problem, because this problem is applied in a wide variety of areas including telephone call routing on communication networks and vehicle routing on road networks.

The A* algorithm was originally presented by Hart, et al. (1968). It was designed to solve for the shortest path between an origin and a destination. Suppose that in addition to the network of nodes and arcs that we have information in which we can estimate the distance from any one node to the destination. Formally, let h(i) = the estimate of completing the path from node i to the destination node t. One candidate for h (*) is the Euclidean distance measure. The Euclidean distance measure is always a lower bound on what it would take in distance to complete a path from i to node t on a Cartesian plane (for a large region it would be the distance of the great circle arc). The A* algorithm for solving the one-to-one shortest path problem can be described as follows:

Step 1 : Begin the problem by setting $d(i) = +\infty$ for each $i \in N$; next set $g(s) = 0$ *and* $d(s) = g(s) + h(s)$; finally let $s=\{s\}$;

Step 2 : Node selection : identify node $v \in S$ where $d(v) \leq d(i)$ for all $i \in S$; set $S=S-\{v\}$;

Step 3: Stopping criteria: if $v=t$, stop as the shortest path to destination t has been found; otherwise go to step 4.

Step 4: Expanding: for each node w where arc$(v, w) \in A$, if $g(w) > g(v)+l(v, w)$ then update $g(w) = g(v) + l(v, w)$, set $d(w) = g(w) + h(w)$ and when $w \notin S$ let $S = S+ \{w\}$; after all updates have been made go to step 2.

By using the above steps the algorithm is developed and coded in the MATLAB. When we run the code it asks to give all the inputs like target position, obstacles and the starting point. When these all are given algorithm calculates the optimal path and follows that path to reach the target. This way it can calculate optimal path for all the valid layouts in an industry and plan its path optimally.

6. RESULTS:

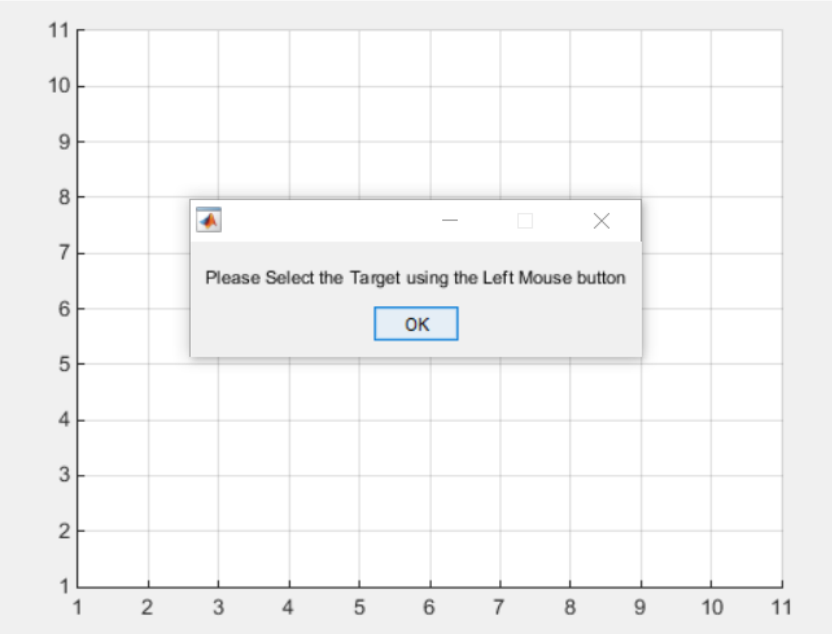

Fig.2 GUI for giving the target point in a layout

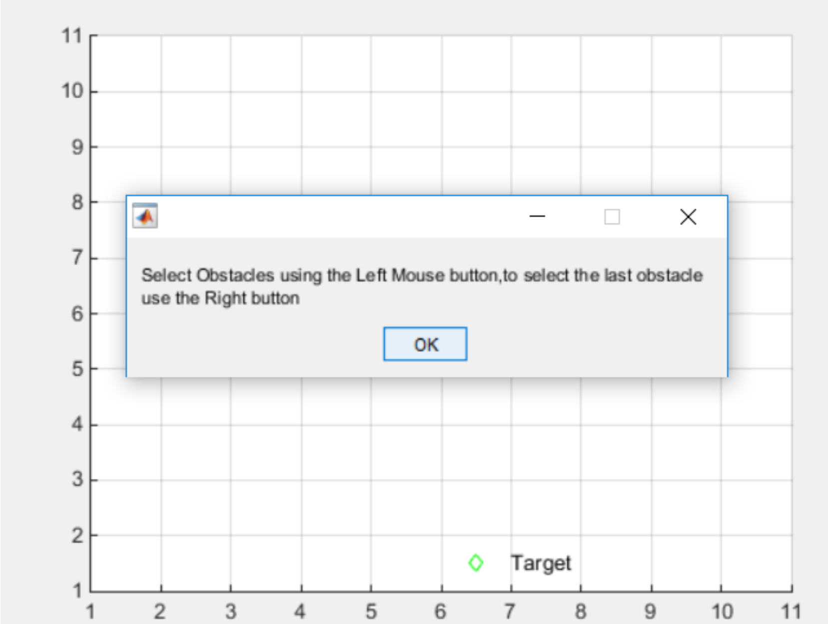

Fig. 3 GUI for giving the obstacles

When the code is run, a graph with nodes appears as GUI where user need to mention the target point first. After giving the target point user has to update the GUI with all the obstacles he wishes to with left click of the mouse and for the last mark use right click. When the starting point is given the algorithm makes the robot to go in all the ways possible from the starting point and traces different routes. It calculates the number of nodes it is going through in each path and measures the count of nodes. Finally calculates the shortest path used to reach the target and displays that path which is the optimal path.

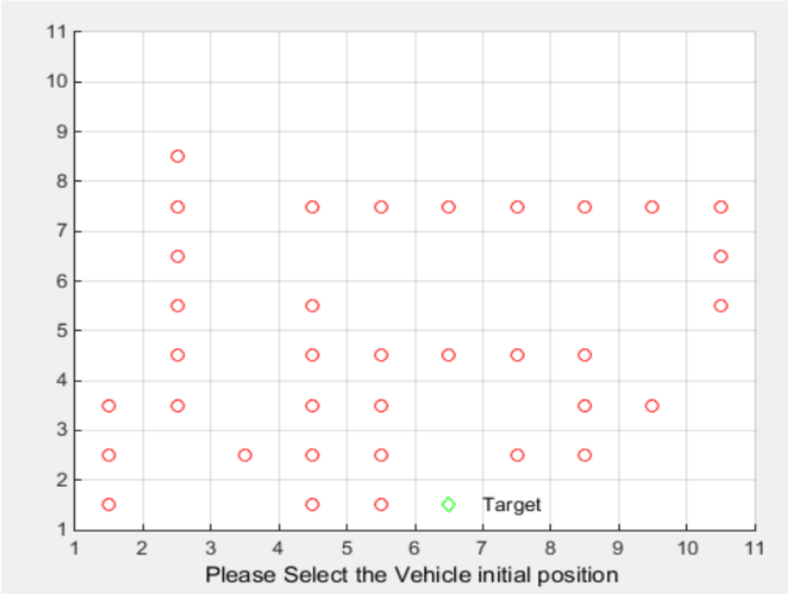

Fig.4 Layout with Target position and obstacles

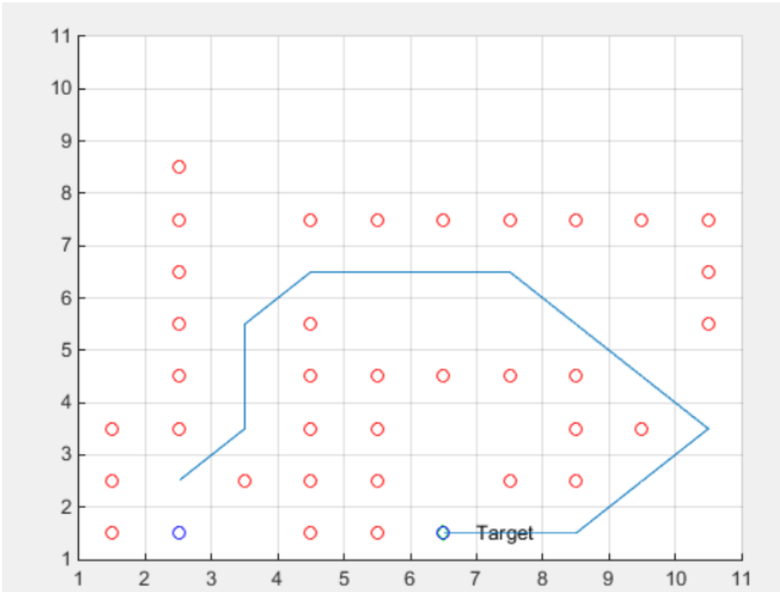

Fig. 5 Optimal Path from starting point to target point

7. FUTURE SCOPE:

This algorithm is fed into a working model of a parallel manipulator and used for tool path planning, network planning, path planning on roads and maps etc. in real time. This algorithm will be fed into a real time parallel manipulator and used in a real environment.

8. CONCLUSION:

Parallel manipulators become one of the best helpers for human beings in works where there is a need of high rigidity and limited workspace. Due to their robustness they are good for carrying loads and working in hazardous uneven paths. Path planning can help these manipulators in saving the time, money and energy. Artificial Intelligence algorithms will definitely make the job easy for a human being.

9. REFERENCES:

[1]. BELLMAN, R., 1958, On a routing problem. Quarterly of Applied Mathematics, 16, pp. 87-90.

[2]. ZHAN, F.B. and NOON, C.E., 1998, Shortest path algorithms: an evaluation using real road networks. Transportation Science, 32, pp. 65-73.

[3]. Barrett, C., Bisset, K., Holzer, M., Konjevod, G., Marathe, M.V., Wagner, D.: Engineering Label-Constrained Shortest-Path Algorithms. In: Fleischer, R., Xu, J. (eds.) AAIM 2008. LNCS, vol. 5034, pp. 27–37. Springer, Heidelberg (2008)

[4]. Bast, H., Funke, S., Matijevic, D., Sanders, P., Schultes, D.: In Transit to Constant Shortest-Path Queries in Road Networks. In: Proceedings of the 9th Workshop on Algorithm Engineering and Experiments (ALENEX 2007), pp. 46–59. SIAM, Philadelphia (2007)

[5]. Chandrashekhar A, G.Satish Babu, “Jacobian Matrix and Kinematic Aanalysis of the Parallel Robots- A Survey” IUP Journal of Mechanical Engineering, August 2015Vol VIII, No.3

[6]. Chandrashekhar A, G.Satish Babu, “Optimal Configurations of Kinetostatic Spatial Parallel Robots Based on Properties and Point of Isotropy” IASTER's International Journal of Research in Mechanical Engineering,3,3, May -June, 2015, Pg. 33 – 45.

[7]. Chandrashekhar A, G.Satish Babu, “Force isotropy of three-limb spatial parallel manipulator”, International Journal of Mechanical Engineering & Technology (IJMET),6,6, June 2015, Pg.01-08.

[8]. Chandrashekhar A, G.Satish Babu, “ Nearly isotropic 6-upu parallel manipulator” International Journal of Design and Manufacturing Technology (IJDMT) Volume 6, Issue 2, July-December 2015, pp. 54-60,

[9]. Chandrashekhar A, G.Satish Babu, “ Optimum design of 4-UPU parallel robot using condition number” International Journal of Industrial Engineering Research and Development (IJIERD) Volume 6, Issue 2, July-Dec 2015, pp. 34-38.

[10]. Chandrashekhar A, G.Satish Babu, “ Generation of Manipulability Ellipsoids for Different Configurations Using the Yoshikawa’s Manipulability Index and Manipulability Ellipsoid”, International Journal of Theoretical and Applied Mechanics, ISSN 0973-6085 Volume 12, Number 3 (2017) pp. 623-635.

[11]. P. Abhilash and A. Chandrashekhar “Analysis of Performance Indices of Planar Parallel Manipulators”, Indian Journal of Science and Technology, Vol 11(23),(2018) pp 1-6.

[12]. https://www.tutorialspoint.com/artificial_intelligence/artificial_intelligence_overview.htm

[13]. https://www.hackerearth.com/practice/notes/a-search-algorithm/

[14]. https://www.geeksforgeeks.org/a-search-algorithm/