OPTIMAL PATH PLANNING FOR AN AUTONOMOUS MOBILE ROBOT USING
AGENT BASED APPROACH

M. Bala Subramanian¹, Dr.K.Sudhagar², G.Rajarajeswari³
¹Research Scholar, Department of Computer Science and Engineering, Bharath Institute of Higher Education and Research, Bharath University, Chennai, India,
²Agni College of Technology, Department of Mechanical Engineering, Chennai, India,
³Rase college of Engineering, Department of Information Technology, Chennai, India.

Received on: 15.10.2016
Accepted on: 22.11.2016

Abstract

Modelling agent based architecture through a heterogeneous agents and designing the best possible path for an autonomous mobile robots are one of the key area, which gains much attention to the researchers in this field. This paper aim’s the study on optimal path analysis of an autonomous mobile robot, based on agent based system. The key objective of this paper is to find an optimal path for an autonomous mobile robot using a known environment based on agents. The sub-optimal path is determined by a heuristics approach, called A Star (A*) algorithm. The communication control between a robot and its server will be accomplished based on the defined open agent architecture, which will be treated as a master-subordinate architecture. The robots used to communicate with its server to update the optimized path details, required to achieve its target position, in an optimized way. The system evaluation was done using a graphical user interface (GUI) based test-bed environment for robots, called Mobile Sim and the efficiency of the system was measured, via the simulation results.

Introduction

The mobile robots are designed for wide range of mission on critical application, in which most of the applications are depend on its navigation from the initial position to target position, in a known or unknown environment [1]. In an autonomous mobile robot (AMR) research community many approaches have been proposed and under proposal for optimal path planning problem. In an AMR research area, two challenging factors that depend on each other are Planning and Reactive. When a Mobile robot (MR) is in execution, there are high chances for path obstacles before it reaches to its destination and without reacting to these unforeseen events, it will not reach to its goal point. So, in order to reach the target position, the MR must include its current location details and its environmental details, that it
gains during its execution, in which it instantly produced new plan along with the new information details to, met its target goal point precisely.

**Literature Study:** A novel approach by introducing a new data structure called multi agent navigation graph, which is constructed from voronoi diagrams was discussed [2]. The agent based approach also collects, much attention in this literature study. Ugur et-al included a study on agent based approach in a static environment [3]. Kawamura et-al presented a user centric human robot interface for supervisory control of a group of MR [4]. The basic bi-directional human robot-interaction and event-triggered adaptation between the system’s agents was demonstrated [5]. Rushan et-al applied agent approach to a heterogeneous MR team to find a solution for localization problem [7]. Wavelet based path planning problem using agent based approach was proposed [8]. Co-ordination among the MR is needed among them [9]. MR coordinated localization, mapping and exploration is a challenging topic [10]. The structure of the paper is organized as follows: In Section 2, the frame work study on constructing the multi agent system is shortly presented. Also, a study on heuristic based approach called A* algorithm for sub-optimal navigational system and the application details are explained in Section 3. In Section 4, presents simulation study and the obtained results. Section 5 presents concluding remarks and guidelines for further work. It is assumed that the reader has basic knowledge on mobile agents.

**Framework Study:** Agent is a concept used by wide range of people in wide range of application. In this agent based research community, this term is has wide range of variants. The concept of agents comes from developing a thinking machine with the capability of solving a problem on its own [12]. An agent can be a piece of software that is capable of accomplishing tasks on behalf of its user. Agents are capable of perceiving their environment, but only a limited extent [13].

**Path Planning Flow Diagram:** The path planning agents, uses a Heuristic based A* algorithm for optimal path generations. Fig 1 shows a scenario of re-planning when two MR have potential obstacles/collision on their paths.

![Fig. 1 Path Planning Flow Diagram.](image-url)
The Robot agents start executing their current navigational plans and continuously sense the probable impact/obstacles for the next movements. Once a probable collision/obstacle is sensed, the robot agents send signals to stop the robot execution and retrieve the current robot’s configurations. The robot agents communicate with the coordinator agent to collect the information about the obstacle and to send information about their current paths and tasks to the coordinator agent.

Path Planning Algorithm

A robot navigation system includes a planner module which determines appropriate path based on its environment map and the obstacle avoidance algorithm determines a suitable direction of motion based on recent sensor data. Path planning considers a model or a map of the environment to determine the geometric path points for the MR to track from its initial starting position to the target or goal position. Many algorithms are proposed by many researchers for static and dynamic environment. There are reasonable number of algorithm, for search graphs for the shortest path between the nodes like Dijkstra’s algorithm, Variations of Moore algorithm, and A* algorithm. A Star algorithm (A *) is a best first graph search algorithm that finds a least cost path from a given initial node to one goal node. A * is generally considered to be the best path finding algorithm. Also in comparison the Dijkstra’s algorithm is essentially the same as A *, except that there is no heuristic (H is considered to be ‘0’ always). Since it has no heuristic, it searches/explores by expanding out equally in every direction. So it usually ends up exploring a much larger area before the target is found. This generally makes it slower than A * and less optimized compared to A* which is much preferred in identifying the optimal path planning.

A * search is one of the widely used informed search strategies. It uses heuristic functions, h(x) which provides approximation for the cost of the best route that goes through the nodes. It will always find a path if a path exists and report failure if a path does not exist and the path that A* returns will be optimal in terms of the heuristic function. This heuristic function is used to accelerate the search process.

It is assumed that, the environment is divided into 1m by 1m square grids. The center point of each square is considered as a node of the graph. Then A * uses this graph to construct the path. The cost for each grid is calculated as the sum of two costs:

\[ f(n) = h(n) + g(n) \]  \hspace{1cm} (1)

Here, the function g(n) is the cost to reach from the starting node to the node n, h(n) is the cost to reach from the node n to the goal node. Since, g(n) gives the path cost from the start node to node n and h(n) is the heuristic distance.
which is the estimated cost of the closest path from \( n \) to goal, \( f(n) \) becomes the estimated cost of the shortest solution through node \( n \). A * is an optimal search strategy if \( h(n) \) is an admissible heuristic that is provided that \( h(n) \) never overestimates the exact cost to reach the goal. The input for A * search algorithm is the graph and the output is a back-pointer path which is a sequence of nodes starting from the goal and back to the start. In the case of \( O \) is the open set which is a priority queue and \( C \) is the closed set containing all processed nodes, the A * search algorithm can be expressed as below [3].

**WHILE O is not empty**

Find \( n \) from \( O \) such that \( f(n) \leq f(n) \) for all \( n \) in \( O \)

Remove \( n \) from \( O \)

Add \( n \) to \( C \)

IF \( n = n \) to Goal

END

for all \( x \), neighbour of \( n \) and not

Expand \( n \) in \( C \)

IF \( x \) is not in \( O \)

Add \( x \) to \( O \)

ELSE IF \( g(n) + \text{dist}(n, x) < g(x) \)

Update \( x \)'s back-pointer to point the \( n \) best

END
Implementation point of view, we need to have the Initial and target position; Dimensional details (includes the workspace cell details); Map file of the workspace; output the optimal path as a list of waypoints. The region of a grid cell uses eight-point connectivity relation and the heuristic distance (h(n)), which is calculated by using Euclidian distance which is always smaller than or equal to the actual distance.

Simulation Results

A numerous number of MR simulation systems were used in the past, all with the goal of creating an artificial robot environment, as real as possible, considered to be the test bed for the implementing the MR algorithms. The realistic error models is for sensors and actuator used in the design model allows the code development, testing and debugging the robustness of the robot programs that can handle with the real environments [6]. The majority of the robot simulation tools focus on the motion of the robotic manipulator in different environments. As the motion simulation has a central role in all simulation systems they all include the kinematic or dynamic models of robot manipulators.

Which type of models will be used depends on the objective of the simulation system. For example, trajectory planning algorithms rely on kinematics models [11]. The effectiveness of the above defined approach is tested and evaluated in Mobile Sim simulation environment, which shows promising results in the existence and nonexistence of anonymous obstacles.

MobileSim is a test-bed for Pioneer robots and the programs are written with Aria framework. The programs that work correctly in simulator also work in the real life. Working environment is drawn by Active Media Mapper program and imported to MobileSim program [6] Using this information and the map of the environment, the A* algorithm constructs a path between the starting point and the goal point. And the constructed path is passed through communication channel to the robot. The trace of the path followed by the robot is shown in below Fig.2, Fig.3. As a part of simulation activity, the input for this simulation is defined as grid position co-ordinates. The input details are passed to the robot agents.

The initial position is of grid, (X, Y) is (1, 1). And the target point is (19, 9). The robot agent will identified the optimal path using the A* algorithm. The trace details of different simulation scenarios and environment are also captured. Fig 2, 3 shows the behavior movement aspects of a pioneer robot, defined with its initial position and the
movement from initial to the target positions are captured. The path trace details are captured and shown in Fig 4.

The MR movement details are captured and shown in below figures for different case scenario.

Fig 2. Behavioural map details of Pioneer robots position

Fig 3. MR movements from Initial to Target

Fig 4. Ray Trace of Pioneer MR with movement details.

Conclusion

A comprehensive study was made on agent based real-time optimal path planning algorithm for AMR. The navigation path is constructed by A* heuristic algorithm, which is considered to be one of the best paths finding algorithm, based on its environment exploration. The discussed approach is simulated on a GUI based MobileSim simulation tool. The results are prominent and accurate. The robot agent navigates from its source point to the destination as per the defined input values. This simulation study is based on the static environment and further it can be extended for the dynamically changing environment and also it can be applied in real time scenario as future extended work.

References


