Int. J. Mech. Eng. & Rob. Res. 2014

International Journal of Mechanical Engineering and Robotics Research

ISSN 2278 – 0149 www.ijmerr.com
Vol. 3, No. 2, April, 2014

Research Paper

INTELLIGENT CONTROL OF MOBILE ROBOT USING C++

D Elayaraja[1]*, R Ramakrishnan[2], M Udhayakumar[3] and S Ramabalan[4]

Corresponding Author: **D Elayaraja, ✉ cape.duraisamy@gmail.com*

A robot is an electro-mechanical device that can perform autonomous or pre-programmed task. A robot can work or carry out an assigned task successfully with the help of a program. The objective of this work is to make the navigation of the robot more precise and add intelligence to it by using programming software. In this work C++ compiler is used to stimulate the mobile robots within these environments. Controller programs helped to read input values and show the required simulation in a graphic window. The goal reaching character of the mobile robot is successfully carried out using C++ program.

Keywords: Mobile robot, Intelligent Control, Goal reaching, C++ Program

INTRODUCTION

Autonomous mobile robots need to bring everything along with them, including a power supply and a brain. The power supply is typically an array of batteries, which adds a lot of weight to the robot. The brain is also constrained because it has to fit into the robot, not weigh a ton, and be frugal about sucking power out of the batteries.

Parhi et al used Rule base technique and petri net modelling to avoid collision among robots one model of collision free path planning and also modelled fuzzy-logic technique and both the models are compared. They found that the rule-based technique has a set of rules obtained through rule induction and subsequently with manually derived heuristics. This technique employs rules and takes into account the distances of the obstacles around the robots and the bearing of the targets in order to compute the change required in steering angle (Dayal Ramakrushna Parhi *et al*.,).

Fainekos et al. provided a tractable solution to the motion planning problem for dynamics models of mobile robots and temporal logic motion planning problem for mobile robots are modelled by second order dynamics (Georgios E *et al*.,).

Raquel Ros et al interested in the action selection and coordination for joint multi-robot tasks, motivated by a prototype environment of robot soccer. They have successfully applied Case-Based Reasoning (CBR)

[1] Associate Professor, K.Ramakrishnan College of Engineering, Samayapuram, Trichy, Tamilnadu, India.
[2] PG student, K.Ramakrishnan College of Engineering, Samayapuram, Trichy, Tamilnadu, India.
[3] Asst.Professor, K.Ramakrishnan College of Engineering, Samayapuram, Trichy, Tamilnadu, India.
[4] Principal, E.S.Pillay Engineering College,Nagapattinam, Tamilnadu, India.

techniques to model the action selection of a team of robots within the robot soccer domain. However, their previous approach did not address the dynamic intentional aspect of the environment, in particular, in robot soccer, the presence of adversaries. Many efforts aim at modelling the opponents in particular when the perception is centralized. Instead, they addressed a robot soccer framework in which the robots are fully distributed, without global perception or global control (Raquel Ros *et al*.,).

Yun-Hui Liu et al proposed a tangent graph for path planning of mobile robots among obstacles with a general boundary. The tangent graph is defined on the basis of the locally shortest path. It has the same data structure as the visibility graph, but its nodes represent common tangent points on obstacle boundaries, and its edges correspond to collision-free common tangents between the boundaries and convex boundary segments between the tangent points. The tangent graph includes all locally shortest paths and is capable of coping with path planning not only among polygonal obstacles but also among curved obstacles (Liu YH, Arimoto S,1992).

Kamal Kant et al present a novel approach to solving the trajectory planning problem (TPP) in time-varying environments. The essence of their approach lies in a heuristic but natural decomposition of TPP into two sub-problems: (i) planning a path to avoid collision with static obstacles and (ii) planning the velocity along the path to avoid collision with moving obstacles (Kant K, Zucker SW, 1986).

Fujimura K et al. performed a study on problems in planning a collision-free path for a robot in a time varying environment. It is assumed that an obstacle moves along a known path. The formulation also allows the destination point (the point to be reached) to move along a pre-determined path. A new concept of 'accessibility' from a point to a moving object is introduced and is used to determine a collision free path. An environment which contains some uncertainty in which the robot needs to see and possibly plan an alternative path is also considered. The ability to deal with moving obstacles is useful in a variety of visual tasks such as the navigation of an autonomous vehicle and the interaction between robot arms (Fujimura K, Samet H, 1988).

Lamadrid JG et al. addressed the problem of planning motions for a mobile robot in the presence of objects moving on unknown trajectories and with unknown velocities. The robot must follow a predefined path with a given tolerance and reach its destination by a given time without colliding with any obstacle. Objects are represented by nonintersecting discs in the plane, and the robot by a point. The proposed method relies on the use of sensors to detect obstacles, and interleaves path planning with execution. Experimental results obtained both in simulation and with a real robot are shown by them (Lamadrid JG, Gini ML, 1990).

Slack MG et al. presented an integrated route planning and spatial representation system that allows paths to be calculated in dynamic domains. The path planner finds the "best route" through a given n-dimensional space.The algorithm takes into account the following: Capabilities of the robot executing generated plans, traversability of space, and interactions with both predictable and unpredictable dynamic objects. Planning robot movement in dynamic environments demands that the dynamic aspects of the environment be modelled in at least as much detail as the movements of the robot. They have created a representation system that

allows dynamic aspects of the environment and performance aspects of the robot to be easily modelled. It also integrates this model with a route-planning algorithm. This system has been extended into an incremental route planner which can be used for real-time tactical planning in unpredictable domains (Slack MG, Miller DP, 1987).

Fiorini P et al. presented a simple and efficient approach to the computation of avoidance manoeuvres among moving obstacles. The method is discussed for the case of a single manoeuvring object avoiding several obstacles moving on known linear trajectories. The original dynamic problem is transformed into several static problems using the relative velocity between the manoeuvring object and each obstacle (Fiorini P, Shiller Z, 1993).

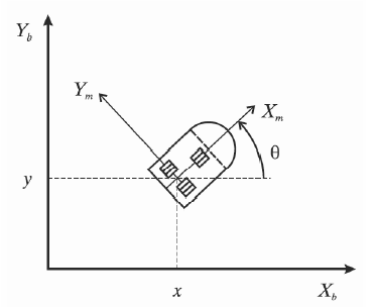
A paper by LaValle SM et al. presents the first randomized approach to kinodynamic planning (trajectory planning/design). The task is to determine control inputs to drive a robot from an initial configuration and velocity to a goal configuration and velocity while obeying physically based dynamical models and avoiding obstacles in the robot's environment. They considered generic systems that express the nonlinear dynamics of a robot in terms of robot's high dimensional configuration space. Kinodynamic planning is treated as motion planning problem in a higher- dimensional state space that has both first order, differential constrains and obstacle-based, global constraints (LaValle SM, Kuffner JJ, 200).

Ahuactzin et al. has formulated a genetic algorithm for motion planning of robots which shows that the path planning problem can be expressed as an optimization problem and thus solved with a genetic algorithm. We illustrate this approach by building a path planner for a planar arm with two degree of freedom, and then we demonstrate the validity of the method by planning paths for a holonomic mobile robot (Ahuactzin, El-Ghazali Talbi *et al*.,).

KINEMATICS OF WHEELED MOBILE ROBOT

A unicycle mobile robot is an autonomous, wheeled vehicle capable of performing missions in fixed or uncertain environments. The robot body is symmetrical around the perpendicular axis and the centre of mass is at the geometrical centre of the body. It has two driving wheels that are fixed to the axis that passes through and one passive wheel prevents the robot from tipping over as it moves on a plane.

Figure 1: A Unicycle Mobile Robot



The nonholonomic kinematic constraints are described by

$$A(q)\dot{q} = 0 \quad ...(1)$$

And the following relation can be obtained

$$A(q)S(q) = 0 \quad ...(2)$$

Where

$$S(q) \in R^{n\times(n-m)} = R^{n\times r}$$

is composed of linearly independent vectors in the null space of a (q) from (1) and (2), (n-m)-dimensional vector exists such that

$$\dot{q} = S(q)z \quad ...(3)$$

where z corresponds to the internal state variable. Differentiating (3) we have

$\ddot{q} = S(q)\dot{z} + S(q)z$

Defining variables q and z in (3) as

$q^T = [x_c y_c \theta_c]$ and $z^T = [v_c w_c]$

a nonholonomic constraint on the rolling motion of the robot wheel without slipping, given by (1), can be expressed as

$x_c \sin\theta_c — y_c \cos\theta_c = 0$

The kinematic equations in Cartesian coordinates corresponding to (3) is

$$\begin{pmatrix} \dot{x}_c \\ \dot{y}_c \\ \dot{\theta}_c \end{pmatrix} = \begin{pmatrix} v_c \cos\theta_c \\ v_c \sin\theta_c \\ \omega_c \end{pmatrix}$$

where x_c and y_c are position variables, θ_c is a heading direction angle, υc is a forward linear velocity, ω_c is an angular velocity of the mobile robot. Also, the kinematic equations in polar coordinates become

$$\begin{pmatrix} \dot{\rho}_c \\ \dot{\phi}_c \\ \dot{\theta}_c \end{pmatrix} = \begin{pmatrix} v_c \cos(\phi_c - \theta_c) \\ -\frac{v_c}{\rho_c} \sin(\phi_c - \theta_c) \\ \omega_c \end{pmatrix}$$

Where

$\rho_c = \sqrt{(x_c^2 + y_c^2)}$ and $\phi_c = \tan^{-1}(y_c/x_c)$

Obstacle Avoidance and Goal Reaching Task

To provide the robot with ability to navigate autonomously by avoiding obstacles in crowded and unpredictably environment; the following behaviors are designed: Goal reaching, Obstacle avoidance.

Goal Reaching Task

The goal reaching behavior tends to drive the mobile robot from a given initial position to a stationary or moving target goal.

Obstacle Avoidance Task

For obstacle avoidance behavior, the robot needs to acquire information about the environment. This task tends to steer the robot in such a way as to avoid collision with obstacles that happens to be in the vicinity of the robot. If the environment of the robot contains one or more obstacles, the robot must be able to avoid collisions. The robot uses the obstacle avoidance controller in order to reach the final destination safely without collision with the objects.

ALGORITHM FOR OBSTACLE AVOIDANCE AND GOAL REACHING

- User has to be asked to input the type of obstacle. Depending on the type of obstacle, input the number of sides if it is a polygon.
- Then depending on the type of obstacle it asks to enter the co-ordinate of the vertices if it is a polygon and the co-ordinate of centre and the radius if it is a circle.
- User has the flexibility to add any number of obstacles in the arena. The arena has a co-ordinate system which ranges from 0 to 700 in the horizontal axis (x-axis) and from 0 to 500 in the vertical axis(y- axis).
- After inserting the obstacles it will ask to enter the starting point and the destination point of the mobile agent.
- In the programme the starting point co-ordinates are taken as (a, b) and the destination point is taken as (x1, y1).
- For shortest distance path the robot has to travel to the destination along a straight line. So the slope of the line joining (a, b) and (x1, y1) is mo = (y1-b)/(x1-a). The equation of the line in which the robot has to move: yi = (mo)*(xi-a) + b.

- Now for loop" for the programme will start incrementing the value of xi by1 and getting the value of yi from the above mentioned equation.
- According to the value of the co-ordinates given to the obstacles the equation and the range of their side or curves are obtained.
- While the "for loop" is running the values of xi and yi were continuously being checked with the equations of sides or curves (circle) of the obstacles.
- If the value of xi and yi satisfies the preset values of equation then the robot will move till the end point of that side with the same slope as that of side in order to avoid collision.

SIMULATION

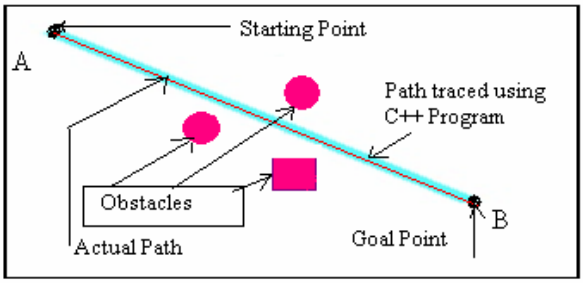
In this section, examples of mobile robot navigation in indoor environment are presented to verify the validity of the proposed schemes. The designed environment takes into account several situations such as: free space and an area with many circular shapes as obstacles. These typical cases in which the robot task is to move from a given current position to a desired goal position in obstacle free environment and in the presence of obstacles.

Figure 5.1: Output of C++ Program

Enter the starting points co-ordinates in (x,y) form 25 30

Enter the target points co-ordinates in (x,y) form 350 300_

Figure 2: Graphical Form of Goal Reaching with Obstacle Avoidance



CONCLUSION AND FUTURE WORK

The robot motion towards the target point is achieved using C++ program in the 1st phase of the project. Phase 2 consists of developing the program to navigate the robot by avoiding the fixed obstacles and improving the robot intelligence to avoid moving and oscillating obstacles.

REFERENCES

1. A case-based approach for coordinated action selection in robot soccer, by Raquel Ros, Josep Lluís Arcos, Ramon Lopez de Mantaras , Manuela Veloso.
2. Fiorini P, Shiller Z (1993), Motion planning in dynamic environments using the relative velocity paradigm. In: Proceedings of IEEE conference on robotics and automation; pp. 560-5.
3. Fujimura K, Samet (1988), H Accessibility: a new approach to path planning among moving obstacles. In: Proceedings of IEEE conference on computer vision and pattern recognition, Ann Arbor, MI, pp. 803-7.
4. Kant K, Zucker SW (1986), towards efficient planning: the path velocity decomposition. The International Journal of Robotics Research; 5(1):72-89.

5. Lamadrid JG, Gini ML (1999), Path tracking through uncharted moving obstacles. IEEE Transactions on Systems, Man and Cybernetics, 20(6): pp. 1408-22.
6. LaValle SM, Kuffner JJ (2001), Randomized kinodynamic planning. Inter national Journal of Robotics Research, 20 (5): 378-400.
7. Liu YH, Arimoto S (1992), Path planning using a tangent graph for mobile robots among polynomial and curved obstacles. The International Journal of Robotics Research, 11(4):376-82.
8. Slack MG, Miller DP (1987), Path planning through time and space in dynamic domains. In: Proceedings of the international joint conference on artificial intelligence; pp. 1067-70.
9. Temporal logic motion planning for dynamic robots, by Georgios E. Fainekos, Antoine Girardb, Hadas Kress-Gazit b, George J. Pappasa. a 3330 Walnut Street, GRASP Laboratory, University of Pennsylvania, PA 19104, United States Laboratory Jean Kuntzmann, Université Joseph Fourier, France.
10. The stable and precise motion control for multiple mobile robots, Dayal Ramakrushna Parhi (a), Saroj Kumar Pradhan (b), Anup Kumar Panda (c), Rabindra Kumar Behera (d), (a)(d) Department of Mechanical Engineering, N.I.T., Rourkela, Orissa, India (b) Department of Mechanical Engineering, N.I.T., Hamirpur, H.P., India (c) Department of Electrical Engineering, N.I.T., Rourkela, Orissa, India.
11. Using genetic algorithms for robot motion planning, Juan Manuel Ahuactzin, El-Ghazali Talbi, Pierre Bessiere, and Emmanuel Mazer .1-10.