Genetic Algorithm based Trajectory Planning and Control of Multiple Mobile Robot Navigation

Dayal Ramakrushna Parhi\textsuperscript{a}, Jagadish Chandra Mohanta\textsuperscript{b} and Saroj Kumar Patel\textsuperscript{b}

\textsuperscript{a}Professor, Department of Mechanical Engineering, N. I. T., Rourkela-769008, Orissa, India
\textsuperscript{b}Engineering Officer, Mechanical Engineering Division, C. P. R. I., Bangalore-560080, India

This paper discusses about navigation of multiple mobile robots using knowledge based genetic algorithm (GA) method. This method is used extensively for obstacles avoidance and targets seeking. The classical GA is dependent only on the relative distances between the robots and the surrounding obstacles. The new scheme introduces a variable, which takes care of the obstacles' and targets' influences on the path of the robots thereby produces an optimal or near optimal path. The control system combines repelling influences related to the distances between robots and nearby obstacles and with attracting influences between the robots and targets. The variables are dependent on the obstacles positions, both in angles and distances with respect to the robots. It has been observed that using knowledge based genetic algorithm (GA) method robots are able to navigate successfully in a highly cluttered environment.

Key words: mobile robots, obstacle avoidance, target seeking, navigation, genetic algorithm

1. INTRODUCTION

Genetic algorithm (GA) is rapidly gaining popularity in optimal path generation and obstacle avoidance applications for mobile robots. This method is particularly attractive because of its elegance and simplicity. The following research works have been reported in literature in recent years for navigation of mobile robot using this method.

It is obvious that path planning can be viewed as an optimization problem (e.g., shortest distance) under certain constraints (e.g., the given environment and collision-free condition). Since the appearance of genetic algorithms (GA) in 1975 \[1\], GAs have been used in solving many optimization problems successfully. GA is stochastic search technique analogous to natural evolution based on the principle of survival of the fittest. The potential solutions of a problem are encoded as chromosomes, which form a population. Each individual of the population is evaluated by a fitness function. A selection mechanism based on the fitness is applied to the population and the individuals strive for survival. The fittest ones have more chance to be selected and to reproduce offspring by means of genetic transformations such as crossover and mutation. The process is repeated and the population is evolved generation by generation. After many generations, the population converges to solutions of good quality, and the best individual has good chance to be the optimal or near optimal solution. The feature of parallel search and the ability of quickly locating high performance region \[2\] contribute to the success of GAs on many applications.

Many researchers applied GAs for path planning of mobile robots \[3-5\]. However, like most early GA applications, most of those methods adopt classical GAs that use fixed-length binary strings and two basic genetic operators, and few modifications were made to the algorithms. Genetic algorithm based path planning with fix-length binary string chromosomes based on cell representation of mobile robot environment has been proposed \[6\]. Its binary encoding is biased and inefficient. Besides, in order to use the standard GA, the path planning solutions are restricted to X-monotone or Y-monotone. The classical GAs use binary strings and two basic genetic operators. After encoding solutions to a problem, the classical GAs are more like “blind” search, and perform well when very little prior knowledge is available. However, GAs do not have to be “blind” search, when additional knowledge about problem is available, it can be incorporated into GAs to improve the efficiency of GA\[7-8\]. Path planning is such a problem that requires...
knowledge incorporation into the GAs for the problem. Graph technique is a traditional way of representing the environment where a mobile robot moves around. A genetic algorithm based on MAKLINK graph environment representation is proposed by many authors [9-11]. In this genetic algorithm, the path is represented by variable length chromosomes formed by mid-points of the free-links, which is a more natural way of encoding than binary strings. This graph based method needs to form a configuration space before applying the genetic algorithm.

Both Hocaolu et al. [12] designed specialized genetic operators with some heuristic knowledge. A path is represented by a hierarchically ordered set of vectors that define path vertices generated by a modified Gram-Schmidt orthogonalization process [13]. Liu et al. [14] have proposed a novel Genetic Algorithms (GAs) approach for a near optimal path planning of a mobile robot in a greenhouse. The chromosome encoding features in inverse proportion between research spaces of GAs and complexity of obstacles. They designed the fitness evaluation for both incomplete and complete paths to guide the evolutional direction. Geisler et al. [15] have improved genetic algorithm performance by developing a more efficient genotype structure for a known environment with static obstacles. Motion was constrained to only row-wise navigation. Sedighi et al. [16] improved on the above work and presented results of a genetic algorithm based path-planning model developed for local obstacle avoidance of a mobile robot in a given search space while Simionescu et al. [17] discussed a new approach to solve constrained nonlinear programming problems using evolutionary computations. However, in all such studies, a limited effort was made to find an optimal controller (instead, an GA was designed based on a particular user-defined function and rules) for mobile robots navigation with multiple targets.

This paper proposes an efficient genetic algorithm (GA) method for motion planning of mobile robots in a cluttered environment. Here the new GA function and the corresponding virtual force are defined. This method enables the multiple robots to find the target in an unknown cluttered environment in a shortest path. Computer simulation results demonstrate the effectiveness of the dynamic motion planning scheme based on the proposed genetic algorithm (GA) approach.

2. DESIGN OF MOBILE CONTROLLER USING GA

2.1. Basic Approach for Obstacle Avoidance

As specified earlier a genetic algorithm (GA) based obstacle avoidance scheme has been used here for path planning of multiple robots with multiple targets in presence of obstacles. Genetic algorithms are heuristic optimization methods whose mechanisms are analogous to biological evolution. The evolutionary procedure employed in the simulations consists in programming a standard genetic algorithm (GA) system. The speed of genetic algorithm depends heavily on the encoding scheme of the chromosomes and on the genetic operators that work on these chromosomes [18]. In order to speed up a GA, the chromosome’s and gene’s structures need to be as simple as possible. In addition, only a few, but very effective, reproduction operators should be applied on the chromosomes. A GA operates on a population of chromosomes, which represent possible solutions for a given problem. This implementation is a new approach to the path-planning and obstacle avoidance problem, representing each chromosome as a group of basic attitudes. These attitudes define the robot’s movements in agreement with the feedback generated by its environment. Each feedback, which is used as input to the system, is based on the sensors reading and on the robot’s direction to its goal location.

The sensors reading are presented to the GA system in a simplified form. The proposed simulator provides 6 sets of ultrasonic and 4 sets of IR sensors for detecting the obstacles and bearing of the targets. However, the distances are calculated at each instantaneous position of the robots. These distances are calculated based on the position and orientation of the obstacles with respect to robots instantaneous position.

For this purpose the basic inputs to the genetic controller are front obstacle distances (FOD), left obstacle distances (LOD) and right obstacle distances (ROD) to all the GAC and output heading angles (HA) are expressed in terms of encoded generation function distributions by crisp values. Then, these crisps values are converted to binary value in order to facilitate the interface the machine language for further processing of the controller. To visualise the above genetic controller in real sense the problem has been addressed in different stages. The stages are analysed below:

Stage 1: Formation of pool set for obstacle avoidance: From the sensors out puts (FOD, LOD and
Genetic Algorithm based Trajectory Planning and Control of Multiple Mobile Robot Navigation

ROD) distances an initial population pool is created with a predefined population size. The population contains number of individuals (i.e., chromosomes). Each individual represents a solution for the problem under study. In our case, each solution is in terms of a heading angle between the current directions of the robots’ steering with respect to targets’ directions from its start to end point in the search space. The initial population with size n can be presented as follows:

Initial Population = \(<P_1, P_2, \ldots, P_n>\)

Each structure have the elements \(p_{(i)}\) which are simply an integer string of length \(L\), in general.

Each population have 5-sets of chromosomes which are represented by Element numbers 1 to 5.

Element: 1 Element: 2 Element: 3 Element: 4 Element: 5
\(P_1 = \{p_{1,1}, p_{1,2}, p_{1,3}, p_{1,4}, p_{1,5}\}\)
\(P_2 = \{p_{2,1}, p_{2,2}, p_{2,3}, p_{2,4}, p_{2,5}\}\)

\(\ldots\)

\(P_n = \{p_{n,1}, p_{n,2}, p_{n,3}, p_{n,4}, p_{n,5}\}\)

Where, Element No. 1 (\(p_{1,1}\) to \(p_{n,1}\)) represents the left obstacle distance (FOD)
Element No. 2 (\(p_{1,2}\) to \(p_{n,2}\)) represents the front obstacle distance (LOD)
Element No. 3 (\(p_{1,3}\) to \(p_{n,3}\)) represents the front obstacle distance (ROD)
Element No. 4 (\(p_{1,4}\) to \(p_{n,4}\)) represents the instantaneous heading angle (HA) with respect to target potion.
Element No. 5 (\(p_{1,5}\) to \(p_{n,5}\)) represents the sign conversion (‘+ve’, ‘zero’ and ‘–ve’) for clock wise, straight and anti clockwise based on the direction of HA \(f\) respectively which is shown in Fig. 1.

Beyond 511 mm radius the region is treated as obstacles free and in this case robot considered that, there is no obstacle in the particular direction and starts moving towards target till find any obstacle on the way within the range. In this case heading angle will be zero (Case 5 in Table 1). In case 2 left obstacle distance is medium, front obstacle distance is near and right obstacle distance is near, so robot will take a left turn of 12degree (-ve) in order to avoid obstacle. Similarly in case 8, left obstacle distance is near, front obstacle distance is very near and right obstacle distance is medium, so robot will take a right turn of 12degree (+ve) in order to avoid obstacle. For the heading angle each degree of rotation is taken as \((511/180)\)th part for both clock wise and anti clockwise movement of robot. For simplicity a set of 10 populations has been shown in tabular form (Table. 1).

Table 1

<table>
<thead>
<tr>
<th>Case</th>
<th>FOD (mm)</th>
<th>LOD (mm)</th>
<th>ROD (mm)</th>
<th>HA (Degree)</th>
<th>Direction (+/-)ve</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>250</td>
<td>200</td>
<td>150</td>
<td>18</td>
<td>-ve</td>
</tr>
<tr>
<td>2</td>
<td>300</td>
<td>340</td>
<td>260</td>
<td>12</td>
<td>-ve</td>
</tr>
<tr>
<td>3</td>
<td>400</td>
<td>300</td>
<td>150</td>
<td>10</td>
<td>-ve</td>
</tr>
<tr>
<td>4</td>
<td>270</td>
<td>280</td>
<td>160</td>
<td>15</td>
<td>-ve</td>
</tr>
<tr>
<td>5</td>
<td>600</td>
<td>100</td>
<td>120</td>
<td>0</td>
<td>straight</td>
</tr>
<tr>
<td>6</td>
<td>505</td>
<td>400</td>
<td>280</td>
<td>5</td>
<td>-ve</td>
</tr>
<tr>
<td>7</td>
<td>480</td>
<td>220</td>
<td>450</td>
<td>14</td>
<td>+ve</td>
</tr>
<tr>
<td>8</td>
<td>120</td>
<td>200</td>
<td>360</td>
<td>12</td>
<td>+ve</td>
</tr>
<tr>
<td>9</td>
<td>500</td>
<td>100</td>
<td>450</td>
<td>16</td>
<td>+ve</td>
</tr>
<tr>
<td>10</td>
<td>320</td>
<td>450</td>
<td>180</td>
<td>10</td>
<td>-ve</td>
</tr>
</tbody>
</table>

Stage 2: Analysis of fitness function for obstacle avoidance: Fitness function represents an important part of any evolutionary process using GAs. Appropriate selection of the fitness function will lead the search towards the optimal solution. The optimal obstacle avoidance, in our case, is the possible collision free motion of robot with optimum heading angle (HA) with respect to target location, thereby optimising the trajectory between the start and end point in the environment. Thus, the fitness function is responsible for optimal obstacle avoidance. The proposed GA knowledge based controller helps computing the total number of steps the mobile robot need to take to reach the ending point. Consequently, the fitness value for a complete solution will be computed as:

\[ f_{total} = 0.4(f_1) + 0.15(f_2) + 0.15(f_3) + 0.15(f_4) + 0.15(f_5) \]

Where,

\[ f_1 = \sqrt{(C_{FOD} - p_{c,1})^2 + (C_{LOD} - p_{c,2})^2 + (C_{ROD} - p_{c,3})^2} \]
\[ f_2 = |C_{FOD} - p_{c,1}| \]
\[ f_3 = |C_{LOD} - p_{c,2}| \]
\[ f_4 = |C_{ROD} - p_{c,3}| \]
\[ f_5 = |TA - HD| \]
and \((C_{FOD} - P_{ci,1}), (C_{LOD} - P_{ci,2})\) and \((C_{ROD} - P_{ci,3})\) are the best distances (child) obtained from the given pool set of front, left and right obstacles distances from instantaneous obstacle position with respect to initial position. These distances are the output of sensors data.

TA and HD are the target angle and heading direction respectively.

**Stage 3: Crossover of parameters and its analysis:**
During the operation of reproduction crossover is applied on the chosen parent chromosomes only within a certain probability, the crossover probability. In the chosen crossover operator, two parent chromosomes are combined applying a single-cross-point, value encoding crossover. The crossover operator has been modified to produce two offspring chromosomes with each crossover operation. This is achieved by using the gene information, which were not used to build offspring one, in order to build a second chromosome. In the proposed controller we used the crossover operators for front, left and right obstacle distances as well as for the heading angle. The function of the crossover operator for case first two pool set shown in table 1 is illustrated in Fig. 2.

**Stage 4: Mutation:** For mutation, almost every operation that changes the order of genes within a chromosome or that changes a gene’s value (such as location or direction) is a valid mutation operator. The mutation operator has been designed according to the addressed obstacle avoidance problem. The chosen mutation operator checks with a mutation probability for every single gene whether it should be mutated or not. If a gene is to be mutated, a random number between 1 and the total number of population in the search space is assigned to location and a random direction, either clockwise, anticlockwise or straight, is based on reference direction. This mutation variant has the advantage that it gives the opportunity for a chromosome to become significantly altered. That means that the complete search space will be explored and it therefore prevents the GA from getting stuck in a local optimum. The fitness of all affected genes (steps) is re-evaluated and stored in the variable feasibility immediately after the changes in location and direction are made. Each step’s fitness is therefore always up to date with each instant position of robot.

**Stage 5: Evaluation of fittest child according to fitness function:** The evaluation of fittest child is computed as per the fitness function described in stage 2. According to the fittest child, the heading angle of the robot will be decided. The obstacle avoidance behaviour of the robot using genetic algorithm will be incorporated in the controller of the robot for successful navigation. The detail flow chart of the proposed GA model is given in Appendix A.

3. **RESULTS AND DISCUSSIONS**
In the present work, navigation problems of multiple robots in presence of obstacles are solved using developed navigation schemes, namely novel genetic-system. The knowledge based GA-approaches are tuned using a GA module with the help of 30 training scenarios generated at random. A particular training scenario is different from the other, in terms of the initial position of the obstacles, their size, speed and direction of movement. The robot is assumed to have a maximum and minimum acceleration of 50 and 5mm²/s, respectively. The performances of the proposed approaches are tried among the obstacles in different scenarios to show the effectiveness of the developed control scheme. The results obtained are discussed below.
3.1. Obstacle Avoidance and Target Seeking Approach

Fig. 1 relates to an exercise designed to demonstrate that the robots reach their targets in a highly cluttered environment without colliding with obstacles. This exercise involves four mobile robots having three targets initially assembled in a highly cluttered environment for avoiding twenty obstacles placed randomly within an enclosure.

In the above simulations two types of environmental scenarios have been presented for robots navigation. It is clear that the robots reach the targets without any collision among themselves at the same time avoiding the obstacles. In Fig. 1 an environmental scenario has been presented for target seeking behavior of three mobile robots respectively for collision-free movement. Fig. 2 shows the navigational path of six mobile robots when they are tracking the target which clearly shows the obstacle avoidance and collision free movement by multi-robots and multi-targets systems.

In Fig. 3, the 3-D obstacles functions have been shown with corresponding contour plot.

From the above results it can be observed that using proposed method, the robots are able to navigate successfully in a cluttered environment.

3.2. Collision-free Movements in a Cluttered Environment by Several Robots

Obstacle avoidance by six mobile robots in a highly cluttered environment is shown in Fig. 2. This shows the situation where six robots with three targets and each robot reach its nearest target efficiently without colliding between obstacles and among themselves. It can be noted that, during the navigation robots stay well away from the obstacles.

In the above simulations two types of environmental scenarios have been presented for robots navigation. It is clear that the robots reach the targets without any collision among themselves at the same time avoiding the obstacles. In Fig. 1 an environmental scenario has been presented for target seeking behavior of three mobile robots respectively for collision-free movement. Fig. 2 shows the navigational path of six mobile robots when they are tracking the target which clearly shows the obstacle avoidance and collision free movement by multi-robots and multi-targets systems.

From the above results it can be observed that using proposed method, the robots are able to navigate successfully in a cluttered environment.

4. CONCLUSIONS

In this paper, a new GA based controller has been developed for navigation of mobile robots in a highly cluttered environment. Firstly, we have proposed a method for adjustment of fitness values to avoid statistical variation due to randomness in initial positions and orientations of obstacles. Then the distances of obstacles from three directions (viz. front, left and right) were evaluated by using suitable fitness function and optimized by the proposed algorithm based upon an iterative non-linear search, which utilizes matches between observed geometry of the environment and a-priori map of position locations, thereby correcting the position and orientation of the robot to find targets. The developed strategies have been checked via simulations, which show the ability...
of proposed controller to solve the multiple robot navigation tasks in an optimized way and to obtain a strategy for this purpose.

The above navigation method can be applied suitably for autonomous cooperative task handling by mobile robots in space mission, hazardous environments and factory shop floors.

References


Appendix A

Flow Chart for GA Program

- Start
  - Initialize a population of strings Iteration I=0
  - I=I+1
  - Assign fitness to all strings in the population
  - If I > Imax?
    - Yes
      - Assign fitness to all strings in the population
      - Output best solution
      - End
    - No
      - Reproduction
      - Crossover
      - Mutation

- Yes
  - No