# A low cost genetic algorithm based control scheme for wheelchair control in hospital environment

**Karam Dad**
*Department of Mechanical Engineering, Pusan National University, 635-4Jangjeon 2 dong, Geumjeong-gu, Busan, 609-735,South Korea*

**Muhammad Jawad Khan**
*Department of Mechanical Engineering, Pusan National University, 635-4Jangjeon 2 dong, Geumjeong-gu, Busan, 609-735,South Korea*

**Wang Jie**
*Department of Mechanical Engineering, Pusan National University, 635-4Jangjeon 2 dong, Geumjeong-gu, Busan, 609-735,South Korea*

**Min Cheol Lee**
*School of Mechanical Engineering, Pusan National University, 635-4Jangjeon 2 dong, Geumjeong-gu, Busan, 609-735, South Korea*

*E-mail: karam@pusan.ac.kr, mclee@pusan.ac.kr*

**Abstract**

In this paper, a control strategy to operate a wheelchair in a hospital is proposed to assist patients. The strategy is tested using a mobile robot that is navigated in a hospital using line following scheme. The mobile robot is operated using both genetic algorithm and $A^*$ algorithm for a thorough comparison of the control scheme. The comparison of the results revealed that genetic algorithm is a better solution in controlling the wheelchair in a hospital environment.

*Keywords*: Wheelchair, Patients, Mobile Robot, Genetic Algorithm, A* Algorithm.

## 1. Introduction

Automated wheel chairs are characterized as mobile robots due to the involvement of motion planning and control schemes [1]. The automation schemes in wheel chairs are sub-divided into two major parts i.e. (i) control schemes and (ii) navigational schemes [2]. Both are used in interpretation of human gestures for assistance via control through sensors. The navigational schemes are further sub divided into offline and online schemes [3], where the offline algorithms plan the course of actions before starting the path, while online algorithms plan the path of the robot while executing the path. Control schemes provide the disturbances which are encountered due to the irregularities in the environment [4]. This paper proposes an offline scheme for navigation of the automated wheel chair.

## 2. Proposed Methodology

The main emphasis of this research is the implementation of path planning algorithms on a real time robot. Several strategies are involved in developing a real-time path following mechanism. The components used in this research are given below:

- BeInMotion® SDK Motor Control Kit
- Accelerometer
- Arduino UNO

A brief description of the methodology used for implementation of the optimal path planning in a static environment is given in this section. This methodology is divided into three significant parts:

- Optimal Path Planning for known environment
- Communication of optimum path to the Mobile Robot
- Evaluation of optimal path tracking by the Mobile

### 2.1. *Optimal Path Planning for known environment*

Optimal path planning algorithm convergence cost is higher than the ordinary algorithms; therefore Field programmable Gate Arrays (FPGA's) are generally used for implementing these algorithms, as they possess comparatively more computational power than microcontrollers [5]. But the use of an FPGA for optimal path evaluation disturbs the scalability process of the methodology. For example, consider a case where we want to change the path planning algorithm. In that case a retransform of new path planning algorithm for the FPGA may require extensive amount of work. As this research emphasis on creating a scalable method which can be used with different path planning algorithms, therefore, a more generalized approach is proposed. As the environment is completely static the path is, therefore, to be evaluated offline. It may be more advantageous if the path is preplanned and then the coordinates are fed to the mobile robot controller. For this a control scheme to evaluate the optimal path beforehand using computer is proposed. The details are given in the next section.

### 2.2. *Communication of optimum path to the mobile robot*

Once the optimum path is planned the new path coordinates are to be communicated to the Mobile Robot for further processing. Parallel communication may require many lines and is not generally preferred while communicating with the PC is done through serial communication.

### 2.3. *Evaluation of optimal path tracking by the mobile robot*

The Mobile Robot receives the path points from the PC and starts path execution command it follows the desired path and passes through each path point. In order to localize the mobile robot in the given environment data from two sensors, accelerometer and rotary encoders, is received and evaluated [6]. The mobile robot is first turned to the direction of next path point and then it travels the desired distance in a straight line. To force the Mobile Robot to follow a straight path Proportional Integral and Derivative (PID) control is proposed. The Mobile Robot stops when the designated distance in that direction is travelled and then turns to the next path point and repeats the process. The sensors inform that the planned distance has been travelled.

## 3. Implementation of Proposed Methodology

The proposed methodology is shown in the Figure 1. Flow Chart for Proposed Methodology as a flow chart which is implemented in three parts as following:

- Optimal Path Evaluation Mechanism
- Data Transfer Mechanism
- Path Following Mechanism

.

### 3.1. *Optimal Path evaluation mechanism*

This station performs the function of evaluating the optimal path from the desired algorithm. It consists of a PC only. In this research the optimal path has been found from Genetic Algorithm and A* algorithm but as it has been already discussed that the methodology is scalable so any other path planning algorithm can be used to find the optimum path. The Genetic Algorithm and A* algorithm takes the input environment and generates the path points of the optimum path. The station saves these path point coordinates for future operations.

### 3.2. *Data transfer mechanism*

This station performs the function of data transfer between the PC and the mobile robot. The path points which were evaluated in the first step are serially transferred to the mobile robot through this mechanism.

### 3.3. *Path following mechanism*

The path following mechanism includes motion of the robot from starting to the end point along the given path following the fed points on the path. This consists of both forward and turning of the robot along the path until arrives to the goal point [7].

## 4. Path Planning

The algorithms used for static environments in this study are following:

- A* Algorithm
- Genetic Algorithm

### 4.1. *A* Algorithm*

A* Algorithm finds the most cost efficient path starting from initial node to one of the destination nodes, given there are one or more destination nodes. As A* always prefers a path with the lowest expected total cost which could be distance or time or any other parameter, while keeping an already sorted priority line up of path segments that could be alternately used along the way [8]. Two functions combine to make the cost function, both of which are as follow

- Cost Function of Past Path, this is the cost between starting and current node and is generally represented by g(x)
- Cost Function of Future Path, this is a permissible "heuristic approximation" of the cost from current position to the destination and is represented by h(x).

Each node is evaluated by adding the heuristic function h(x) and g(x) i.e. the cost of reaching the present node being evaluated from the start node.

$$F(x) = g(x) + h(x) \quad (1)$$

### 4.2. *Genetic Algorithm*

Genetic algorithm is based on the principle of genetics and natural selection. It comes under the class of meta-heuristic techniques which are based on the general experiences [9]. It works on the principle of survival of the fittest. In search space it searches the optimal points intelligently by using statistical evolutionary methods. Evolutionary methods are based on natural selection where the fittest chromosomes survive, reproduce and mutate so that the resulting chromosomes in successive generations are increasingly competitive. This natural selection process is coded mathematically to generate an initial population of chromosomes and use selection procedures to converge to an optimal configuration [10]. The computation can proceed in parallel rather than in series thus resulting in high computational speed-ups. A detailed pseudo-code for the GAs are as follows:

- Define GA parameters
- Generate initial population

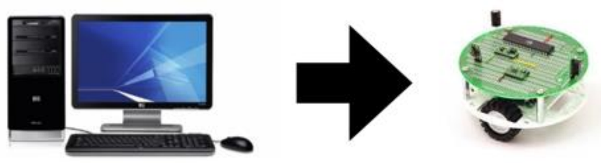

Fig. 1. Implementation Block diagram of Proposed Methodology

- Decode chromosomes
- Evaluate fitness of each chromosome
- Select mates for reproduction
- Carry out reproduction
- Carry out mutation
- Check for convergence
- If not converged go to step (iii)
- Print converged solution

GA's work on the principle of survival of the fittest. This means all the good points which give maximum value for the objective function are permissible to continue in the next generation and the points which does not give maximum values for the objective function are thrown away from our calculations. GAs maximizes a given objective function, so it is necessary to convert a minimization problem into maximization problem before applying the technique of GA. Fitness function is defined as follows:

$$F(n) = f(n) \quad \text{for a problem of maximization} \quad (2)$$
$$F(n) = 1/(1+f(n)) \quad \text{for a problem of minimization} \quad (3)$$

Equation (2) is used for the maximization of the problem whereas equation (3) is used to minimize the cost function. In path planning application, path distance is used to evaluate the fitness value of each chromosome. If the distance to the goal or path length of a certain chromosome is small, the chromosome is better than the other chromosomes in the current generation. The fitness function for path length can be defined as.

$$F_t(n) = 1/d(n) \quad (4)$$

Whereas d(n) defines the path length for the specific chromosome.

## 5. Experiments

Experiment were tested in a mobile robot instead of a wheelchair since a automatic wheel chair is an autonomous mobile robot that is used for the transportation of patients in a hospital.

### 5.1. *Testing Procedure*

First the offline navigational algorithm is run on a computer. The computer computes the optimum path and evaluates the path points for that particular path. It then serially transmits the coordinates to a host controller (Arduino UNO in this case) mounted on the mobile robot. The Arduino UNO board communicates with both the accelerometer and the computer. Firstly, it receives the coordinates from the computer and transmits it to the FPGA mounted on the mobile robot one by one. Secondly it acquires the acceleration values continuously from the accelerometer, applies Simpson's rule for double integration of the acceleration and acquires the distance covered and then transmits it to the Altera Cyclone IV E FPGA mounted on the mobile robot [11]. The Altera Cyclone IV E FPGA communicates with the Altera UNO board to get both the data from accelerometer and computer. It further evaluates the data acquired from the IR sensors mounted on the mobile robot base to apply the proportional integral derivative (PID) control which ensures that the mobile robot moves in a straight line. The IR sensors are also used for the computation of distance covered. The effective distance covered is the result of multiplication of the readings of distance covered from both sensors (accelerometer and IR sensor) with their respective weights and adding them.

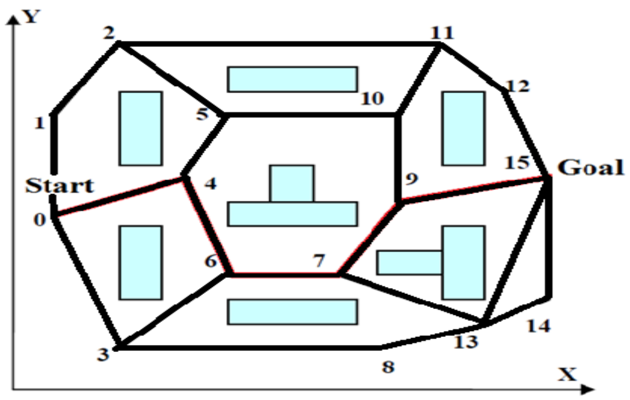


Fig. 2. Test Environment Designed

### 5.2. *Tests Designed*

The test for the mobile robot consisted of a sample which is given in Figure 2. The black lines present the linkages possible while blue boxes represent the obstacles. The start and stop points are also mentioned by "Start" and "Goal" respectively. Starting point is 0 and the goal point is 15. Each point is defined as the change in direction. There are many paths to reach the goal

## 6. Results and Discussion

The main objective of the research was the navigation of a Mobile Robot (BeInMotion Motor Control Kit) based upon the optimal path provided by an offline path planning algorithm.

### 6.1. *Implementation of Optimized Solution*

For the implementation of the path the environment under test was practically created. The created environment is shown in Figure 3. Figure shows the robot in given environment while navigating through the obstacles it starts from 0 and reaches to goal point 15 according to path generated by genetic algorithm. It first starts from 0 and reaches point 15 following the path created by points 0,4,6,7,9,15 respectively. Robot moves in two steps firstly it calculates the turning angle and then it moves forward and reaches desired point and this process is repeated for every movement from point to point.

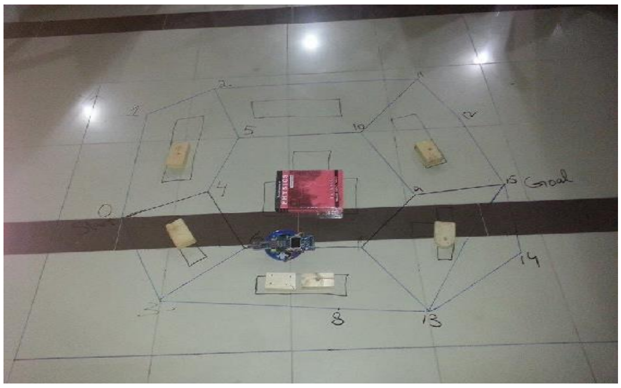

Fig. 3. Implemented Environment details

The computed path was implemented on the Mobile robot and with the help of genetic algorithm we achieved less errors and increased accuracy with each trial. The average percentage error observed from the implementation of genetic algorithm was 1.54% which can be easily reduced further by using better quality navigational sensors. By improving the quality of the sensors, more importantly the accelerometer we can achieve greater accuracy

## 7. Conclusion

The results show that this technique can be used for transporting patients from one place to another in a hospital within a controlled environment where the destination point has been loaded. The result were satisfactory and can further be improved by use of a control scheme alongside the presented scheme.

## Acknowledgements

This section should come before the References. Funding information may also be included here.

## References

1. J. Borenstein, B. Everett, and L. Feng. Navigating Mobile Robots: Systems and Techniques. A. K. Peters, 1996.
2. Borenstein J., Koren Y., "Tele-Autonomous Guidance for Mobile Robots," IEEE Tran. System, Man, and Cybernetics, Special Issue on Unmanned Systems and Vehicles, December, pp. 1437-1443. 1990,
3. J. Czyzowicz , E. Kranakis , D. Krizanc , L. Narayanan , and J. Opatrny, Optimal online and offline algorithms for robot-assisted restoration of barrier coverage, arXiv: 1410.6726v1 [cs.DS] 24 oct 2014
4. Mnif, F. & Touati, F. , An Adaptive Control Scheme for Nonholonomic Mobile Robot with Parametric Uncertanity, pp. 059 - 063, International Journal of Advanced Robotic Systems, Volume 2, Number1 (2005), ISSN 1729-8806
5. Shilpa Kale and S. S. Shriramwar, FPGA-based Controller for a Mobile Robot, (IJCSIS) International Journal of Computer Science and Information Security, Vol. 3, No. 1, 2009
6. J. Borestein, H. R. Everett, L. Feng: Where Am I? Sensors and Methods for Mobile Robot Positioning, University of Michigan, Michigan 1996.
7. Karam Dad Kallu, Muhammad Usman Rafique and Dr Zafar Ullah Koreshi, Development of a path following autonomously navigating mobile robot with parallel processing using FPGA, 1st International Young Engineers Convention (IYEC-2014) on April 18-20, 2014
8. Delling, D. and Sanders, P. and Schultes, D. and Wagner, D., *Engineering route planning algorithm* (Algorithm of large and complex networks, 2009).
9. Peter E. Hart, Nils J. Nilsson and Bertram Raphael, "A Formal Basis for the Heuristic Determination of Minimum Cost Paths," IEEE Transactions on Systems Science and Cybernetics, pp. 100-107, July. 1968.
10. Eiben, A. E. et al., *Genetic algorithm with multi-parents recombination* (PPSN III: Proceedings of the International conference on Evolutionary computation. The Third conference on parallel problem solving from nature, 1994).
11. http://www.arrownac.com/solutions/beinmotion/