

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

Karam Dad<sup>1</sup>, Muhammad Jawad Khan, Wang Jie, Min Cheol Lee<sup>2</sup>

<sup>1</sup> 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

<sup>2</sup> Control and Measurement system,

### Abstract

In this paper, a control strategy is proposed for assisting patients, with wheelchairs, to move around within a hospital. The strategy is tested using a mobile robot that is navigated in a hospital while using obstacle avoidance scheme. The mobile robot is operated using both genetic algorithm and A\* algorithm for a thorough comparison of both schemes. The comparison of the results confirmed that genetic algorithm give a far better solution for motion planning of a 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 based on genetic algorithm and A\* algorithm. The GA method is heuristic and is based on a random search of permissible paths, in a 2D grid with obstacles, to determine the 'best' paths from a 'fitness' function which expresses the objective function in an optimization formulation. Other procedures, such as cross-over and mutation, are used from genetics in an analogous manner to improve upon an estimate of the optimal path from one

generation to the other until convergence is established. A\* is an informed search algorithm, or a best-first search, meaning that it solves problems by searching among all possible paths to the solution (goal) for the one that incurs the smallest cost (least distance travelled, shortest time, etc.)

### 2. Proposed Methodology

In this research a scalable method is presented for navigation of a mobile robot in a static environment. In the static environment, map of the surrounding and designed path is assumed to be known. The proposed navigational scheme uses an Altera® FPGA based mobile platform, BeInMotion™. Use of an FPGA based platform allows parallel and fast implementation of the optimal path algorithm. The system uses PID controller for trajectory correction. The proposed FPGA is programmed using the soft processor because of easy sequential programming. The parallel processing capability speeds up in acquiring feedback from servo motor, reading accelerometer data and controlling motors through PID simultaneously. The developed navigational scheme has been implemented based on genetic algorithm [1] for path planning on FPGA based mobile robot. The same navigational scheme can be

used with any other offline path planning algorithm to implement its path practically on a mobile robot in real environment. The main emphasis of this research is the implementation of path planning algorithms on a mobile 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 with known surroundings 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. 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.

### **2.2. 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 Figure1.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. Data transfer 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]. The main component of this mechanism is the distance computation of the mobile robot in a particular direction.

#### Distance Approximation

Distance covered by the mobile robot is estimated using multiple schemes and then the data of both schemes is combined to acquire more sophisticated results. The distance is measured by using an encoder and an accelerometer at the same time.

#### Measurement using Encoders

Using an encoder for distance computation is straight forward. We attach high resolution encoders to each wheel and then calculate the distance by using the wheels' number of revolutions but this technique has the disadvantage that it doesn't account for the slippage in the wheels of the moving robot

#### Measurement Using Accelerometer

Acceleration,  $a(t)$ , can be found by taking the second derivative of position or first derivative of velocity

$$a = \frac{dv}{dt} = \frac{dx^2}{dx}$$

However, it is of interest to reverse this process and find the position signal given an acceleration signal. To do that, integration must be performed twice on the acceleration signal. In principle, using double integration on an acceleration signal to get a position signal, the initial position and initial velocity must be known. After the first integration, the initial velocity should be added to the result, as the initial position should be added after the second integration. These operations are illustrated in the following equations:

$$v(t) = v(t_0) + \int_{t_0}^t a(T) dT$$

where  $t_0$  is the initial time and  $v(t_0)$  is the initial velocity, which is a constant. To get the position signal from velocity, formula is used:

$$x(t) = x(t_0) + \int_{t_0}^t v(T) dT$$

Therefore, for a double integration to be performed on acceleration, the two initial conditions (velocity and position) must be known to avoid integration errors. However, the only way to get these initial conditions is through direct measurement, which is often impractical or unobtainable. An important part of this project is to

develop an approach that doesn't require knowledge of initial conditions

#### Digital Integration Methods

There are a number of discrete integration algorithms available to perform integration numerically. The acceleration signal is sampled, making it a discrete function of time having a sampling frequency ( $f_s$ ), associated with it. The simplest way to perform numerical integration is to use the rectangular integration method. This method uses an accumulator to sum all past sampled inputs and the current input sample and divide by the sampling rate. Rectangular integration is represented by the following difference equation

$$y(n) = \frac{1}{f_x} \sum_{k=0}^n x(n-k) = y(n-1) + \frac{1}{f_x} x(n)$$

Where  $x$  is the integrand,  $y$  is the output of the integrator, and  $f_s$  is the sampling frequency

There is another method of integrating that uses Simpson's rule. It is defined by the following difference equation:

$$y(n) = y(n-1) + \frac{1}{f_x} * \frac{x(n-1) + 4x(n) + x(n+1)}{6}$$

$$v(n) = v(n-1) + \frac{1}{f_x} * \frac{a(n-1) + 4a(n) + a(n+1)}{6}$$

$$x(n) = x(n-1) + \frac{1}{f_x} * \frac{v(n-1) + 4v(n) + v(n+1)}{6}$$

Unlike the other methods, this one requires a future sample of the integrand,  $x$ , to get the current sample of the integrated signal,  $y$ , so it can't be performed in real time. This approach is used in this research as the environment is already known therefore it is the most suitable candidate

## 4. Path Planning



Fig. 1. Implementation Block diagram of Proposed Methodology

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 in the graph 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.

$$F(x) = g(x) + h(x) \tag{1}$$

#### 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. The fitness function used in this research is briefly described in the proposed methodology. The value of the fitness function of a specifically coded string is called its string's fitness. This particular fitness value selects whether that string transfer on to the next coming generation or not.

### 5. Experiments

Experiments were tested in a mobile robot instead of a wheelchair since an 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.

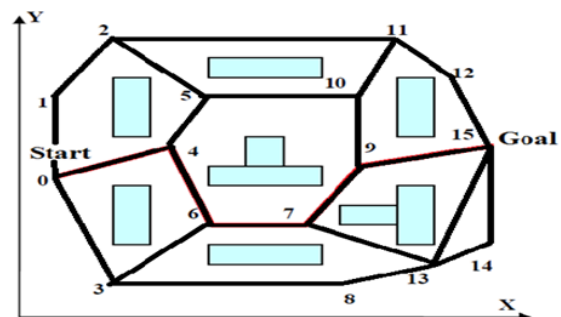


Fig. 2. Test Environment Designed

**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 this research is to provide an implementation based performance analyzing solution, for harnessing newly developed path planning algorithms, for general purpose static environment motion planning problems. Motion planning for an autonomous wheelchair carrying a patient is a commonly referred variant of this kind of problem.

**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 the point marked 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.

Table 1 : Experimental results of proposed algorithm

Trial #	Theoretical Distance (cm.)	Observed Distance (cm.)	Error in Distance (cm.)	Percentage Error (%)
1	449	440	9	2
2		445	4	0.9
3		441.4	7.6	1.7
4		444	5	1.4
5		440.1	8.9	2

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. As a very low cost accelerometer was used to implement the desired methodology, therefore, the results were not as accurate as one may expect from a system using multiple sensing technologies.

**6.2. Comparison of Genetic and A\* Algorithm**

Both Genetic Algorithm and A\* Algorithm were tested for the same environment, given in Fig 2, both algorithms produced the same path for the robot to follow, but the convergence time for genetic algorithm was found very low as compared to the convergence time of A\* algorithm

A timing comparison for both algorithms is provided in Table 2.

Table 2: Time Comparison of Genetic Algorithm and A\* Algorithm

Trial #	No of Solutions per trial	Genetic Algorithm convergence time (s)	A* Algorithm convergence time (s)
1	100	0.05	0.7
2		0.08	0.71
3		0.06	0.69
4		0.04	0.75
5		0.03	0.78

The comparison given in Table 2 was carried out on a system with Dual core 2.4MHz processor, 2GB Ram. Both algorithms were tested in MatLab 2015.

The average observed convergence time for Genetic Algorithm was 520 Us while the average convergence time for A\* Algorithm was 726 Us. The results prove that Genetic Algorithm is much faster in solving a path planning problem than the A\* Algorithm of the same order

## 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 was satisfactory and can further be improved by use of a control scheme alongside the presented scheme.

## Acknowledgements

This research was supported by the MOTIE (Ministry of Trade, Industry & Energy), Korea, under the Industry Convergence Liaison Robotics Creative Graduates Education Program supervised by the KIAT (N0001126).

This work was supported by the National Research Foundation of Korea (NRF) grant funded by the Korea Government (MSIP) (No.NRF-2012M2B2B 1055503).

## 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 Uncertainty, 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. Borenstein, 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/>