

Dynamic Motion Planning Algorithm for a Biped Robot Using Fast Marching Method Hybridized with Regression Search

Ravi Kumar Mandava, Katla Mrudul and Pandu R Vundavilli

School of Mechanical Sciences, IIT Bhubaneswar, Bhubaneswar, Odisha, India-752050. E-mail: rm19@iitbbs.ac.in, km16@iitbbs.ac.in, pandu@iitbbs.ac.in

Abstract: In the past few years, studies of biped robot locomotion and navigation have increased enormously due to its ease in mobility in the terrains that are designed exclusively for the humans. To navigate the biped robot in static and dynamic environments without hitting obstacles is a challenging task. In the present research, the authors have developed a hybridized motion planning algorithm that is, fast marching method hybridized with regression search (FMMHRS) methodology. In this work, initially the fast marching algorithm has been used to observe the environment and identify the path from start to final goal. Later on, the regression search method is combined with the fast marching method (FMM) algorithm to optimize the path without hitting any obstacles. The main objective of the present research work is to generate the path for both the static and dynamic scenarios in simulation and in a real environment. To conduct the testing of the proposed algorithm, the authors have chosen an 18-DOF two legged robot that was developed in our laboratory.

Keywords: Biped robot; static and dynamic environment; fast marching method; regression search

1 Introduction

Path planning plays an important role in the navigation of autonomous vehicles and assisted systems. But, the significant property in this field is that the path planning is developed to satisfy the non-holonomic constraints raised due to its motion. During initial stages of research on path planning, investigators had only considered the length of the path as the major cost, and majority of them were worked extensively to obtain an efficient method, that can generate a collision free path. In general, path planning for mobile robots can be categorized into various classes [1]. Roadmap based methods were used to extract the network representation from the environment and then they apply graph based search algorithms to determine the path. Exact cell decomposition methods were used to

construct the non-overlapping regions that cover free space and encode cell connectivity in graph. The approximate cell decomposition method was similar to the exact cell decomposition method, but the cells were assumed to have a predefined shape and it did not exactly cover the entire free space. The potential field method [2], which is different from the other methods in which the robot was assumed to be a point robot which was moving under the influence of attractive forces generated between the goal and the pushing away from the obstacles due to the influence of repulsive forces generated between the obstacles and robot.

The essential requirement for solving the motion planning problem is the creation of appropriate terrain/environment. Once the environment is created, motion planning algorithms can be implemented in an effective manner. This section presents a brief overview of the different types of path planning methods. The available path planning algorithms are categorized into two types [3]: The first category deals with the classic approaches and the second one is focused on heuristic approaches. In classical approaches, the algorithms are designed to calculate the optimal solution, if one exists, or to prove that no feasible path exists. These algorithms are generally very expensive, computationally. But in heuristic approaches, the algorithms are anticipated to search for good quality solutions within a short time. However, heuristic algorithms can fail to determine the good solution for a difficult problem. There exist few variations of classical methods, such as cell decomposition, roadmap, artificial potential field and mathematical programming to generate the path for the mobile robots. These methods alone and along with their combinations are often used to develop more successful paths. In the roadmap approach [4], feasible paths were mapped onto a network and searched for the desired path. However, the searched path was limited to a network and those path planning algorithms become a graph based search algorithm. Moreover, some of the researchers had developed some well-known roadmaps after using visibility graphs, voronoi diagram [5] and sub-goal networks. In [6], the visibility graph algorithm was used to calculate the optimal path between the start and goal points. In that approach, the authors did not consider the size of the mobile robot, and the mobile robot was moved very close to the vertex of the obstacles. However, the computational time for planning the path using the above method was too long. Later on, researchers developed various methods, such as the voronoi diagram [7] and sub-goal network [8] algorithms that were performed in a better manner when compared with the visibility graph.

In addition to the above approaches, Cai and Ferrai [9] proposed the cell decomposition method, which was the simplest method for planning the path for mobile robot, but the algorithm was inefficient in terms of planning time and managing the computational memory according to the cell size. However, the hybridization of roadmap method with cell decomposition method was seen to provide better efficiency and worked based on the concept of free configuration space (C-space). Due to the lack of adaptation and robustness, the conventional

approaches were not suitable to solve the motion planning problems in dynamic environments. Further, among heuristic approaches, researchers had used A-star algorithm [10] to calculate the shortest path for a given map. In general A-star algorithm was a classical heuristic search algorithm and it was applied on a C-space for planning the path of a robot. The search efficiency of the A-star algorithm was low and the planned path was optimal, when compared with the cell decomposition method. Stenz [11] used D-star algorithm, which was not heuristic, to perform search equally in all directions. When compared with the A-star algorithm, it was found to perform slowly, because this algorithm searches large areas before reaching the goal. In conjunction with, researchers also developed soft computing based approaches for obtaining the optimal path in cluttered environments. In [12], a genetic algorithm was used to obtain the best feasible path after many iterations. It happened due to the complex structure of GA, which requires a huge time to process the data. When dealing with the dynamic environments, this fact lead to the premature convergence while obtaining the optimal solution. To improve the performance of GA, some researchers had suggested different types of optimization algorithms such as, combining genetic algorithm and simulated annealing [13], ant colony optimization [14], particle swarm optimization [15], cuckoo search algorithm [16], invasive weed optimization [17], bacterial foraging optimization [18] and firefly algorithm [19]. In addition to the above approaches, some researchers have also used the soft computing approaches, such as fuzzy-genetic algorithm [20] and neural network based algorithms [21] to solve the motion planning problems of biped robots.

Further, Santiago et al. [22] proposed a robust algorithm called Voronoi fast marching method for obtaining the smooth and safe path in a cluttered environment. It worked based on the phenomenon of local-minima-free planner. Lucas et al. [23] developed a fast marching tree using FMM algorithm for obtaining the optimal path in a high dimensional configuration space. Moreover, a novel path planning algorithm was introduced [24] with non-holonomic constraints for a car-like robot. In this approach, the FMM was used to investigate the geometric information of the map, and support vector machine was used to find the information related to the clearance of the obstacles. The FMM was guided by this function to generate the vehicle motion under kinematic constraints. In [25], the authors explained a detailed overview of fast marching method and also recalled the methods that is, FM2 and FM2* developed and used by the same authors in path planning applications. Garido et al. [26] applied the FMM algorithm to simulate the electromagnetic wave propagation. Here, the wave starts from a point and continuous to iterate until it reaches the end point. The generated field had only one global minima point, which was located at the center point. Petres et al. [27] developed anisotropic fast marching method, which was an improved version of FMM with higher computational efficiency than level set method. Further, Song et al. [28] proposed a novel multi-layered fast marching (MFM) method to generate the practical trajectories for the unmanned surface

vehicles while operating in a dynamic environment. To design an optimal path planning algorithm in [29], the authors developed an effective and improved artificial potential field method combined with regression search. Simultaneously, Ravi et al. [30] established a hybridized path planning algorithm for static and dynamic environments. In this work, they used a 3-point smoothing method to generate the optimal path.

The main objective of the present research is to minimize the path length subjected to constraints on different curvature properties. In order to determine the path in the global map, the authors have presented a novel hybridized path planning method (that is, FMMHRS). It is important to note that the developed FMMHRS will help in achieving the shortest path due to the inherent characteristics of regression search. Initially, FMM has been applied to solve path planning problems [31]. In certain scenarios, the path trajectories obtained are not safe because the path is very close to the obstacles. In order to improve the safety of the path trajectories calculated by using the fast marching method, it is possible to give two solutions: The first possibility is to avoid unrealistic trajectories, generated when areas are narrower than the robot. Therefore, the minimum clearance that should be maintained between obstacles and walls is at least half the size of the robot. The second possibility that has been used in this work is to enlarge the distance between walls and objects to a safe distance so that the robot will not collide with an obstacle. Therefore, initially the entire path is generated with the help of FMM from start to final goal. Once the FMM path is generated, it is split into number of equally spaced nodes. Then regression search is initiated on the nodal data by connecting the present node to the next and by checking the clearance distance between the path and obstacle. This procedure is continued until, the robot reaches the target. To the best of the author's knowledge, this combination has not been tried by any researchers in the field of motion planning of biped robots. Moreover, the proposed algorithm is implemented on static and dynamic environments in both computer simulations and in real time environment. Further, an 18-DOF biped robot has been used to tackle the real time situations. The advantages of this method are ease of implementation, speed and quality of the path. Moreover, this method can work in both 2D and 3D environments, and it can also be used in a local or global scale path planning problems.

2 The Fast Marching Method

The FMM is an efficient numerical algorithm developed by Osher and Sethian in 1988, which was used for tracking and modeling the motion of a physical wave front interface. In general, the algorithm has been applied in various research fields including medical imaging [32], computer graphics [33], image processing [34], computational fluid dynamics and computation of trajectories etc.

The wavefront interface can be modeled as a 3D surface or a flat curve in 2D. The FMM calculates the time T that a wave needs to reach every point of the space. The wave can be originated at one point or more than one point at a given time. If it is originated at more than one point, then each source point generates one wave front and all the source points are associated with time $T=0$. In the context of the FMM the authors have assumed that the wave front (r) grows by motion in the direction normal to the surface. But, the wave speed F which is not same everywhere and it is always non-negative. At a certain point, the motion of wavefront is designated by the Eikonal equation, which is given below.

$$1 = F(x) |\nabla T(x)| \quad (1)$$

where x indicates the position of origin, $F(x)$ represents the wave propagation speed and $T(x)$ denotes the time required by the wave interface to reach x . Further, the magnitude of the gradient of the arrival function $T(x)$ is assumed to be inversely proportional to the velocity of the wave front.

In order to understand the present research paper, it is significant to highlight the property of wave's propagation. It is important to note that the function that represent the time required by the wave interface to reach x . i.e. $T(x)$, only represent a global minima from one single point. Further, as the wave front only expands ($F>0$), the locations away from the source should have greater arrival time T . Moreover, the problem of local minima will arise only if a particular point has a lesser arrival time (T) than the neighbor point which is closer to the source, which is not possible, as the wave must have already reached this neighbor before. In [31], Sethaian established a discrete solution for the Eikonal equation in a 2D area discretized using a grid map. According to [35] the discretization of gradient ∇T can be achieved with the help of the equations given below.

$$\left\{ \begin{array}{l} \max(S_{ij}^{-x} T, 0)^2 + \min(S_{ij}^{+x} T, 0)^2 \\ \max(S_{ij}^{-y} T, 0)^2 + \min(S_{ij}^{+y} T, 0)^2 \end{array} \right\} = \frac{1}{f_{ij}^2} \quad (2)$$

where

$$S_{ij}^{-x} = \frac{T_{i,j} - T_{i-1,j}}{h_x}, \quad S_{ij}^{+x} = \frac{T_{i+1,j} - T_{i,j}}{h_x}, \quad S_{ij}^{-y} = \frac{T_{i,j} - T_{i,j-1}}{h_y}, \quad S_{ij}^{+y} = \frac{T_{i,j+1} - T_{i,j}}{h_y} \quad (3)$$

In the above expression, i and j indicate the rows and columns of the grid map, respectively, h_x and h_y are the grid spacing in x and y directions, respectively. Now substitute Eq. (3) in Eq. (2) and simplify to produce Eq. 4 shown below.

$$T_h = \min(T_{i-1,j}, T_{i+1,j}) \quad \text{and} \quad T_v = \min(T_{i,j-1}, T_{i,j+1}) \quad (4)$$

The revised form of the Eikonal equation, in 2D space after solving the above quadratic equation is given in equation (5).

$$\max\left(\frac{T_{ij} - T_h}{h_x}, 0\right)^2 + \max\left(\frac{T_{ij} - T_v}{h_y}, 0\right)^2 = \frac{1}{f_{ij}^2} \quad (5)$$

It is to mention that the speed of the wave front is assumed to be positive ($F > 0$), T must be greater than T_h and T_v , whenever the wave front has not already passed over the coordinates i, j . Subsequently, Eq. (5) can be rewritten as follows.

$$\left(\frac{T_{ij} - T_h}{h_x}\right)^2 + \left(\frac{T_{ij} - T_v}{h_y}\right)^2 = \frac{1}{f_{ij}^2} \quad (6)$$

In the above equation, whenever $T_{ij} > T_h$ and $T_{ij} > T_v$, always choose a greater value for T_{ij} when solving the Equation. (6). Further, if $T_{ij} < T_h$ or $T_{ij} < T_v$, the corresponding member in Equation (5) will become zero. Moreover, while solving equation. (7), if $T_{ij} < T_h$, then the Eq. (6) is written as follows.

$$\left(\frac{T_{ij} - T_h}{h_x}\right)^2 = \frac{1}{f_{i,j}^2} \quad (7)$$

Further, if $T_{ij} < T_v$, the equation (6) will be written as follows.

$$\left(\frac{T_{ij} - T_v}{h_y}\right)^2 = \frac{1}{f_{i,j}^2} \quad (8)$$

To demonstrate the execution of solution of Eikonal equation, let us consider the following two Figs. 1(a) and 1(b) in which the wave is originated at one and two source points, respectively. In both the figures, the frozen zones are indicated by red colour and their T values are not changed. The points for narrow band and unknown zone are marked by yellow and white colour, respectively. It is also important to note that the wave propagates concentrically in Fig. 1(a) and it propagates in Fig. 1(b). This process continuous and the cells expand as the physical wave grows. The cells that have less T value will expand first. If two cells have different arrival time, then the cell first addressed by the wave front will expand first.

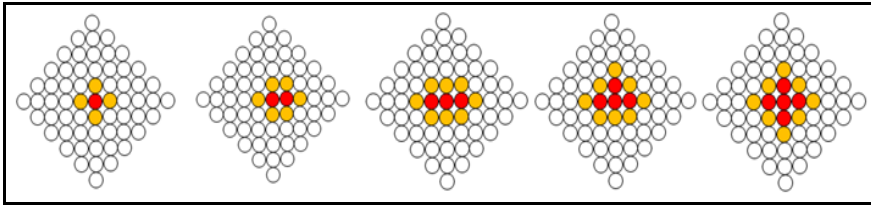


Figure 1 (a)

Wave expansion with one source point

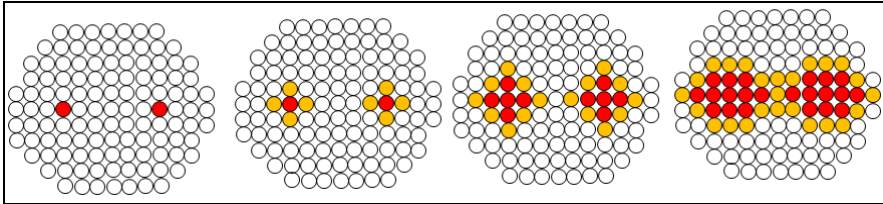


Figure 1 (b)

Wave expansion with two source points

2.1 Regression based Search Method

Although, the FMM algorithm has developed the collision free path in a cluttered environment, it will consume more time and energy of the robot to execute the path. Therefore, to optimize the obtained path in the present research work the authors have implemented a regression search method. In order to obtain the shortest path, the regression search algorithm tries to establish straight lines between the start point and goal point via interconnect points, which are connected with the latter inter points. If the straight line does not cross any obstacle, then the inter start point connects with the next later point with a new straight line until this line crosses any obstacle or the distance between the line and obstacle is less than the d_0 .

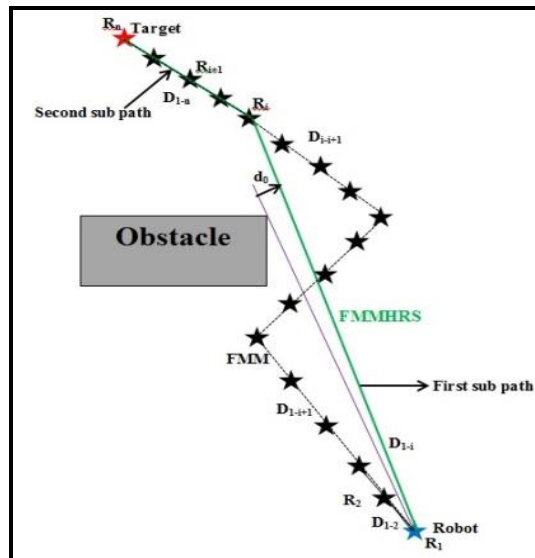


Figure 2

Schematic diagram showing the operation of regression search method

The entire proposed algorithm is given below.

Algorithm 1 Fast marching method hybridized with regression search method

*** Fast marching method***

Input : A grid map of size $m \times n$

Input : Set a node on the grid, where the wave will be originated

Output: Set the value of T for all nodes

Initialization:

T (start point) $\leftarrow 0$

$Far \leftarrow$ all grid points

$Known \leftarrow$ Identify all grid points with known cost

for each adjacent point k find known point

 | $Trail \leftarrow a$
 | $T(a) = \text{cost update}$

end

while

 | sort check
 | the point $n \leftarrow$ point with low cost in checking
 | remove n from check
 | $Known \leftarrow n$
 | for each neighbor point k of n
 | $T(a) = \text{cost update}(a)$
 | | If $a \in Far$ then
 | | Remove a from Far
 | | $Trail \leftarrow a$
 | end
 end

end

*** Regression search based method***

R_1 (start point) $\leftarrow$ connects with next point

From $R_j \in \{R_2, R_3, \dots, R_n\}$

If $D_{1,j} \leftarrow$ does not cross any obstacle connect next point $j=j+1$

 | Check the distance $D_{1,j}$ from obstacle $> d_0$ then $j=j+1$

 | else

 | previous point is the next start point

 | From $R_k \in \{R_{j+1}, R_{j+2}, \dots, R_n\}$

 | Check the distance $D_{j,k}$ from obstacle $> d_0$ then $k=k+1$

End

Obtain the optimal path

The schematic diagram showing the principle of operation of regression search method is shown in Fig. 2. Let us consider that the set of sequential points generated by FMM are assumed as $R_1, R_2, \dots, R_i, R_{i+1}, \dots, R_n$. If the regression search method is applied on the said points, the algorithm tries to connect the initial point (that is, inter start point) R_1 with the next sequential point R_2 with the

help of a straight line forming D_{1-2} . Then the algorithm tries to determine that whether D_{1-2} is crossing any of the obstacles existing in the terrain or not. Once it determines this, the algorithm finds the shortest distance between the line D_{1-2} and the obstacles. If D_{1-2} is not crossing any obstacle or the shortest distance is greater than d_0 , then the algorithm reconnect with R_1 with R_3 as D_{1-3} , and this procedure is repeated until D_{1-i+1} crossing any obstacle. Then the local optimal path up to this point is denoted by D_{1-i} . If there are no further obstacles in the terrain, then the similar procedure as mentioned above is repeated between the inter start point R_i and the target R_n to obtain the path D_{i-n} . Now the robot has to move along the optimal path (D_{1-i} to D_{1-n}), which consumes less energy when compared with the path obtained by FMM algorithm.

3 Results and Discussions Related to Simulation Studies

In this section, the simulation results related to the FMM and FMMHRS in two dimensional work spaces under static and dynamic environments are presented. Once the path planning algorithms have been developed, the effectiveness in generating the collision free paths on various scenarios is studied in computer simulations. These computer simulations are conducted in Python programming environment with the help of a PC that consists of Intel i5 processor running on 2.2 GHz. The 2D simulation space considered in the programming environment is fixed at 500×500 pixel.

3.1 Simulation Results in the Static Environment

In the present section FMM and FMMHRS algorithms are compared with an artificial potential field method (APF) combined with particle swarm optimization (PSO) and three point smoothing method available in the literature [30] are shown in Fig. 3. From Fig. 3, it can be observed that both the proposed approaches and the approach available in [30] are found to generate the collision-free paths in both the scenarios shown above. Further, Table 1 gives the path lengths and time taken to generate the path during the said simulation study. It is seen that the length of the path generated by FMM and FMMHRS algorithms are seen to be less when compared with the APF with PSO and three point smoothing method.

Moreover, it has also been observed that the hybrid algorithm (that is, FMMHRS) proposed in the present research has performed better than the other algorithms considered in this study in terms of both the path length and time taken to generate the path. Further, the authors have tested the proposed algorithms on new scenarios/maps in both static and dynamic environments.

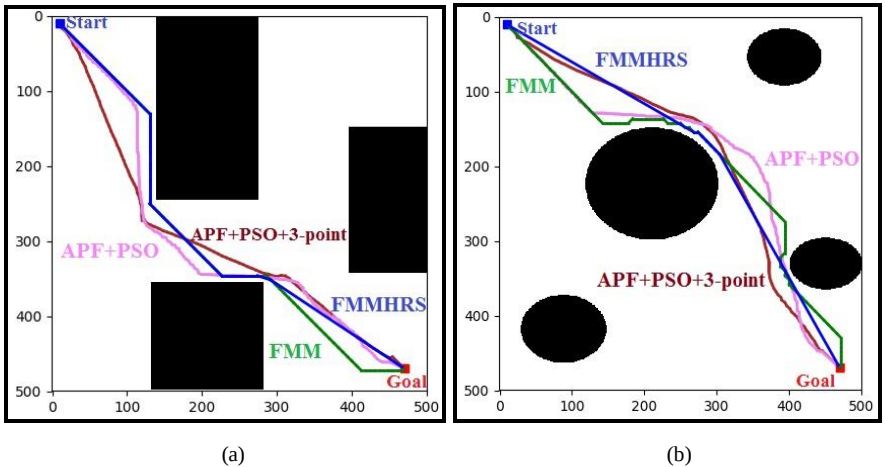


Figure 3
Simulation results of various approaches on different scenarios (a) map 1 and (b) map 2

Table 1
Comparison of path length and time needed to generate paths during simulation

Maps	Path length in pixels				Time taken to generate the path in 'sec'			
	FMM	FMM HRS	APF+ PSO	APF+ PSO+3-point	FMM	FMM HRS	APF+ PSO	APF+ PSO+3-point
map 1	676.08	664.26	819. 87	753.26	30	28	46	40
map 2	708.74	663.34	822. 49	715.35	33	27	47	35

The results related to the generation of path on new terrains after employing FMM and FMMHRS are shown in Fig. 4. It can be observed that in every case, the path developed by the FMMHRS is shortest and optimal in nature when compared with the path obtained by standard FMM algorithm. It is to be noted that in all the maps the obstacles are marked with black color and certain amount of clearance is provided around the obstacles. The path developed by the FMM and FMMHRS are indicated with green and blue color lines, respectively.

Further, Table 2 gives the path length and time required to generate the path for various terrains. From the results of Fig. 4 and Table 2, it has been observed that the FMMHRS approach is seen to provide a shorter path when compared with the standard FMM approach. This may be due to the fact that in FMM approach initially the collision-free path is obtained basically by not considering the shortest route. Later, regression search is employed in which it is always trying to draw a straight line between interstate point and goal point. Then the algorithm will try to determine the location and providing certain clearance around the obstacle to safely navigate the robot without any collision. This fact led to the generation of shortest path.

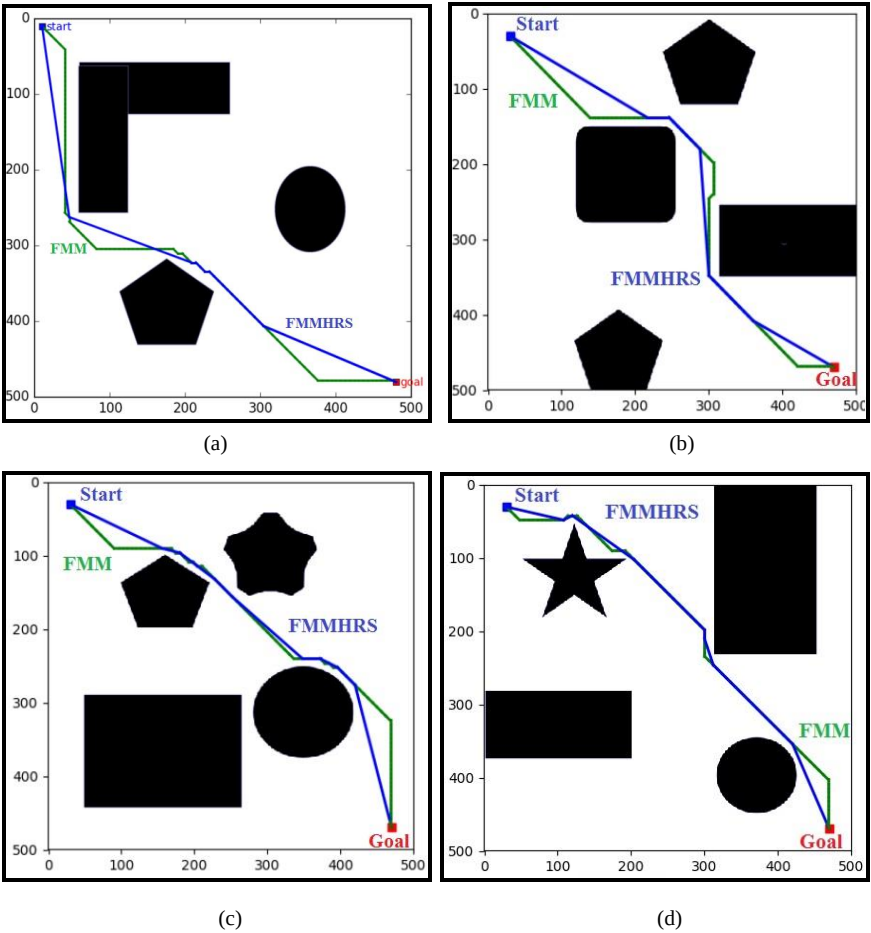


Figure 4

Simulation results related to the static environment at different scenarios (a) map-1, (b) map-2, (c) map-3 and (d) map-4

Table 2

Simulation results related to the static obstacles

Maps	Path length in pixels		Time taken to generate the path in	
	FMM	FMMHRS	FMM	FMMHRS
map 1	791.8965	749.0306	40.23	35.56
map 2	789.7787	684.0287	39.52	32.21
map 3	697.7787	627.8427	34.42	28.25
map 4	678.1463	622.2539	30.25	27.56

3.2 Simulation Results in the Dynamic Environment

The developed FMM and FMMHRS algorithms are also used to generate collision-free optimal path in some dynamic environments. The results of simulation for the scenarios involving one, two and three dynamic obstacles are shown in Figs. (5) and (6), respectively. In all the cases, the scenarios are created in such a way, that the straight line path of the robot will be disturbed, so that once again the robot will plan its future course of its action without deviating from the goal.

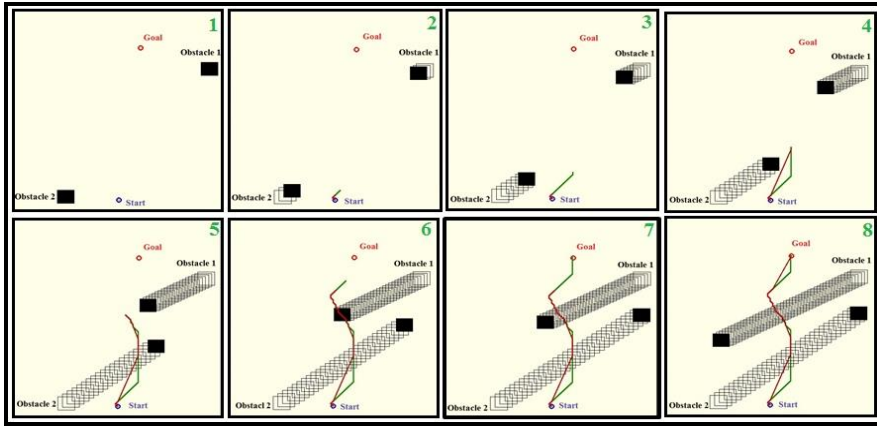


Figure 5

Simulation results related to the dynamic environment with two obstacles

The obstacles are painted with black color and the start and goal points are indicated by blue and green color circles, respectively. The paths generated by the FMM and FMMHRS algorithms are indicated with green and brown color lines, respectively. It can be observed that, in all the scenarios the robot is trying to avoid the collision with the obstacles. The path length and time of travel for the robot to reach the goal are given in Table 3. In this case also FMMHRS algorithm is seen to provide optimal path when compared with FMM algorithm. The reason for this is also same as the one explained above for the static obstacle case.

Table 3

Simulation results related to the dynamic obstacles

Obstacles	Path length in pixels		Time taken to generate the path in	
	FMM	FMMHRS	FMM	FMMHRS
two	441.5878	418.8691	23.21	21.14
three	590.8082	569.0802	38.23	36.12

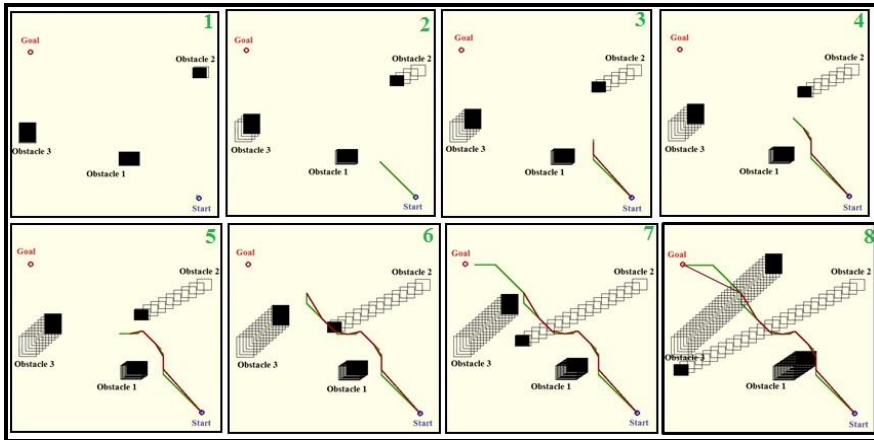


Figure 6

Simulation results related to the dynamic environment with three obstacles

4 Experiments in the Real Environment

In the present work, the effectiveness of the developed motion planning algorithms is verified by conducting real time experiments. To execute the paths developed by FMM and FMMHRS algorithms, the authors have chosen a biped robot [36].

4.1 Experimental Results in the Static Environment

To find the effectiveness of the developed motion planning algorithms, in the present study two different scenarios shown in Figs. (7) and (8) are considered. For ease in identification during image processing, the terrain and the obstacles are painted in white and red color, respectively. The obstacles (that is, static) are located on the terrain in a particular fashion to reflect different scenarios. Further, the start and goal points are marked using marker pen on the terrain. An overhead camera is mounted at the top of the terrain to capture the video of the scene. The two shoulders of the biped robot are marked with green color to indicate the location in the terrain through image processing. The algorithms are implemented using Python software and the image processing technique is used to detect the locations of the obstacles and robot in the scene. While conducting the real time experiments, a wired communication has been employed between the robot and computer terminal to transmit the data related to the on-line path developed by the algorithms, and the required gait angles that are generated to track the path

decided by using the said algorithms. The paths developed by the FMM and FMMHRS algorithms are marked by thick yellow and green lines which are shown in Figs. (7) and (8), respectively. The path length, distance travelled by the robot and time taken by the robot to reach the goal position are given in Table 5.

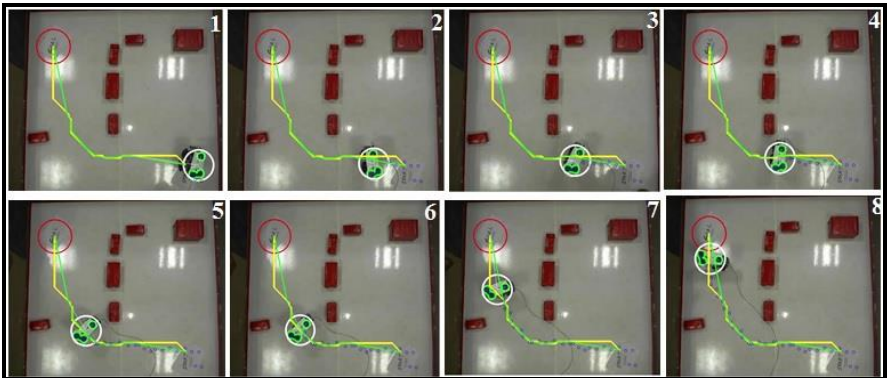


Figure 7

Experimental results for navigation of the biped robot in real time static environment (Scenario 1)

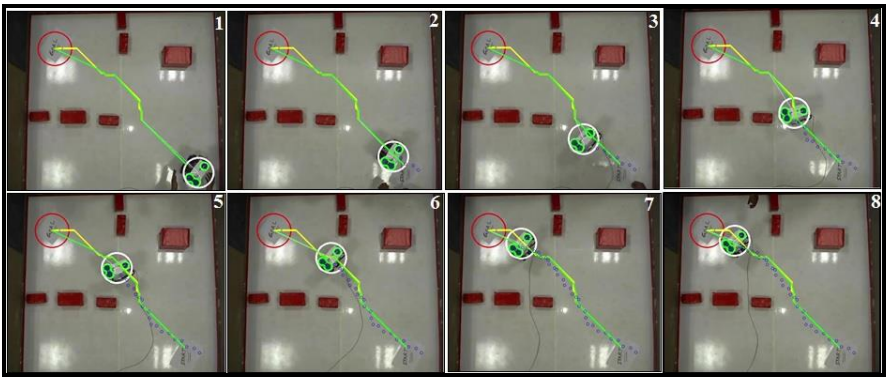


Figure 8

Experimental results for navigation of the biped robot in real time static environment (Scenario 2)

Table 5
Results related to the path length and time travel for various scenarios

Scenarios	Path length in pixels			Robot travel time in sec
	FMM	FMMHRS	Robot	
Scenario 1	586.6761	564.5822	630.0924	215.45
Scenario 2	504.3817	494.9290	657.3502	221.16

It can be observed that the path decided by FMMHRS is seen to be the most optimal when compared with the path developed by the FMM algorithm. The reason for this is same as the one explained earlier. Further, the distance travelled by the robot is seen to be more when compared with FMMHRS. This may be due to the fact that the biped robot is a mechanism that is supported on discrete foot holds and the balancing is a serious problem. Therefore, when the path generated by the algorithm is curved in nature, it will be difficult for the robot to track the exact path due to the following reasons.

1. The robot cannot make sharp turns due to balancing problems raised by the changes in inertia of the robot.
2. The play exists in the transmission mechanism between the motor and the joint also allows for certain misalignment of the path while tracking.

Further, it has been observed that the time taken by the robot to travel from start to finish is seen to be very high when compared with the time required to generate the path by the algorithm. It might have happened because the mobility of the biped robot is very slow due to the balancing problems. The other reason could be the wired transmission of data between the computer terminal and the robot. However, the biped robot has successfully navigated the path among the static obstacles.

4.2 Experimental Results in the Dynamic Environment

The real experiments related to the execution of motion planning in dynamic environments consists of two and three obstacles as shown in Figs. 9 and 10, respectively. As it is a dynamic environment, the obstacles used in this study are allowed to move slowly to meet the requirements of slow walking of the biped robot. During dynamic walking, the path will be updated at regular intervals of time after considering the new location of the robot and moving obstacles. The path update rate can be varied based on the velocity of the biped robot and obstacles. Figures 9 and 10 show the path generated by the FMM and FMMHRS algorithms and tracked by the robot on the terrain while moving among two and three obstacles, respectively. Table 6 shows the result related to the distance covered by the robot from start to the goal and the time taken by the robot to reach the goal position. The results shows that both the algorithms that is, FMM and FMMHRS are capable of generating the path in real time for the environments that contain dynamic obstacles. Further, the biped robot is seen to follow the optimal path generated by FMMHRS with little deviation. The reasons for this are explained in the experiments related to the cases involving static obstacles.

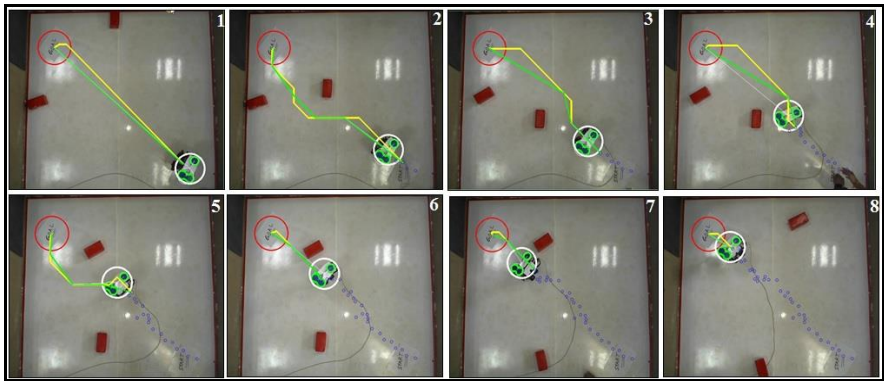


Figure 9

Experimental results for navigation of the biped robot in real time dynamic environment (Scenario 1)

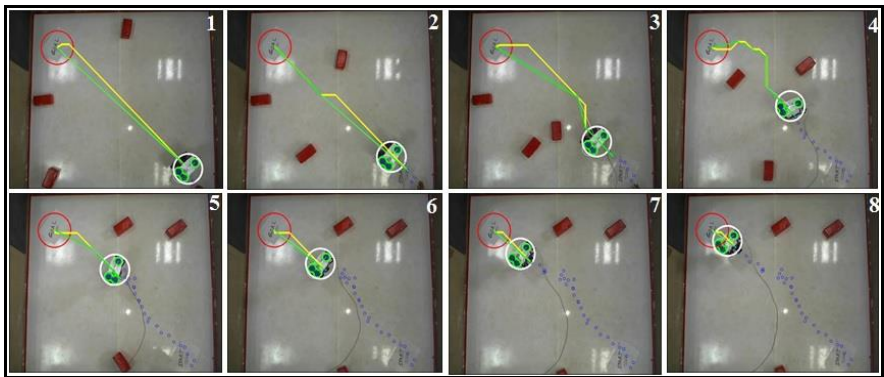


Figure 10

Experimental results for navigation of the biped robot in real time dynamic environment (Scenario 2)

Table 6

Results related to the distance covered and time of travel for various scenarios

Scenarios	Path covered by the robot in pixels	Time travel in 'sec'
Scenario	607.554	210.52
Scenario	647.112	218.23

4.3 Comparison with Other Work

Based on the literature, the authors performed some qualitative comparisons with the approaches reported in [24, 31, and 37-40]. Till date, some of the researchers had used a FMM algorithm to generate the path in computer simulations. In this work, the authors not only used this algorithm to generate the path in a cluttered

environment, but also developed an optimal path after combining FMM with regression search (FMMHRS). Moreover, some of the researchers [29, 30 and 41] had worked on the generation of collision-free path in both the static and dynamic environments in computer simulations only. In the present research, the authors have implemented the said algorithms in both computer simulations and in real time environments. Further, a majority of research in motion planning involves the usage of wheeled robots for validation, which has having better mobility and stability. The only drawback is that, it only can navigate on a continuous terrain, whereas the biped robots are planned to use in the environments that are non-continuous. It is important to note that the mobility and stability of the biped robot is poor while in motion and it can navigate on a discontinuous terrain. In the present study, the proposed FMM and FMMHRS algorithms are successfully implemented on the biped robot in both the static and dynamic environments.

5 Conclusions

This paper explains the features of the FMM and FMMHRS algorithms used to generate a path in both the static and dynamic obstacles environments. Initially, both the algorithms are used to solve the motion planning problem in simulations and in various scenarios. Based on the results of simulation, it can be observed that the developed algorithms are capable of generating collision free paths from start to the goal point. It has been observed that the FMMHRS algorithm is seen to perform better than the FMM approach, for both the static, as well as dynamic scenarios. It may be due to the fact that FMMHRS always tries to provide a straight-line path between the start point and the goal point, when there is no obstacle in the line of path. Further, the real-biped robot is seen to track the path with little deviation and reach the goal point safely.

References

- [1] J.-C. Latombe, Robot motion planning, Dordrecht, Netherlands: Kluwer Academic Publishers (1991)
- [2] Istvan N, Behaviour study of a multi-agent mobile robot system during potential field building, Acta Polytechnica Hungarica, 6 (4) (2009) 111-136
- [3] Masehian, E. and Sedighizadeh. D, Classic and heuristic approaches in robot motion planning – a chronological review, World Academy of Science Engineering and Technology, 29 (5) (2007) 101-106
- [4] Oh, J. S., Choi, Y. H. and Park, J. B, Complete coverage navigation of cleaning robots using triangular-cell-based map, IEEE Trans on Industrial Electronics, 51 (3) (2004) 718-726

- [5] Voronoi, G.F, Nouvelles applications des paramètres continus à la théorie de formes quadratiques, *Journal für die reine und angewandte Mathematik*, (1908) 134-198
- [6] Tarjan, R. E, A unified approach to path problem', *Journal of the Association for Computing Machinery*, 28 (3) (1981) 577-593
- [7] Takahashi, O. and Schilling, R. J., Motion planning in a plane using generalized Voronoi diagrams, *IEEE Trans on Robotics and Automation*, 5 (2) (1989) 143-150
- [8] Avneesh, S., Erik, A. and Sean, C., Real-time path planning in dynamic virtual environment using multi-agent navigation graphs, *IEEE Trans on Visualization and Computer Graphics*, 14 (3) (2008) 526-538
- [9] Cai, C. H. and Ferrai, S., Information-driven sensor path planning by approximate cell decomposition, *IEEE Trans on Systems, Man, and Cybernetics, Part B: Cybernetics*, 39 (3) (2009) 672-689
- [10] Nilsson, N. J., Problem-solving methods in artificial intelligence, *Artificial Intelligence: A New Synthesis*, Morgan Kaufmann Publishers (2000)
- [11] Stentz, A., The focused D* algorithm for real-time re-planning, *Proceeding of the International Joint Conference on Artificial Intelligence* (1995) 1995-2000
- [12] Sedighi, K. H., Ashenayi, K. and Manikas, T. W., Autonomous local path planning for a mobile robot using a genetic algorithm, *Proceedings of Congress on Evolutionary Computation*, 2 (2004) 1338-1345
- [13] Blackowiak, A. D. and Rajan, S. D., Multipath arrival estimates using simulated annealing: application to crosshole tomography experiment, *IEEE Journal of Oceanic Engineering*, 20(3) (1995) 157-165
- [14] Garcia, M. A. P., Montiel, O. and Castillo, O., Path planning for autonomous mobile robot navigation with ant colony optimization and fuzzy cost function evaluation, *Applied Soft Computing*, 9 (3) (2009) 1102-1110
- [15] A. Ayari and S.Bouamama, A new multiple robot path planning algorithm: dynamic distributed particle swarm optimization, *Robotics and Biomimetic*, 4 (8) (2017)
- [16] Prases K. Mohanty & Dayal R. Parhi, Optimal path planning for a mobile robot using cuckoo search algorithm, *Journal of Experimental & Theoretical Artificial Intelligence*, 28 (1-2) (2016) 35-52
- [17] Panda, M. R., et al., Hybridization of IWO and IPSO for mobile robots navigation in a dynamic environment. *Journal of King Saud University – Computer and Information Sciences* (2017)
<https://doi.org/10.1016/j.jksuci.2017.12.009>

- [18] Md. Arafat Hossain and Israt Ferdous, Autonomous Robot Path Planning in Dynamic Environment Using a New Optimization Technique Inspired by Bacterial Foraging Technique, 2013 International Conference on Electrical Information and Communication Technology (2013) 1-6
- [19] Liu, C., Gao, Z. and Zhao, W., A new path planning method based on firefly algorithm. In Fifth international joint conference on computational sciences and optimization (CSO) Harbin, China (2012) 775-778 doi:10.1109/CSO.2012.174
- [20] D. K. Pratihar, K. Deb & A. A Ghosh, Fuzzy-Genetic Algorithms and Time-Optimal Obstacle-Free Path Generation for Mobile Robots, Engineering Optimization, 32 (1) (1999) 117-142
- [21] I. Engedy and G. Horváth, Artificial Neural Network based Mobile Robot Navigation, 6th IEEE International Symposium on Intelligent Signal Processing (2009) 241-246
- [22] S. Garrido, L. Moreno, J. V. Gómez and P. U. Lima, General Path Planning Methodology for Leader-Follower Robot Formations, International Journal of Advanced Robotic Systems, 10 (2013) 1-10
- [23] L. Janson and M. Pavone, Fast marching trees: a fast marching sampling-based method for optimal motion planning in many dimensions, 16th International Symposium on Robotics Research (2013)
- [24] Q. H. Do, S. Mita and K. Yoneda, Narrow Passage Path Planning Using Fast Marching Method and Support Vector Machine, 2014 IEEE Intelligent Vehicles Symposium (2014) 630-635
- [25] Alberto Valero-Gomez, Javier V. Gomez, Santiago Garrido and Luis Moreno, Fast Marching Methods in Path Planning, IEEE Robotics & Automation Magazine, 20 (2013) 111-120
- [26] Garrido, S., Moreno, L., Blanco, D., Exploration of a cluttered environment using voronoi transform and fast marching. Robotics Autonomous Systems 56 (12) (2008) 1069-1081
- [27] Petres, C., Pailhas, Y., Petillot, Y., Lane, D., Underwater path planning using fast marching algorithms. In: Proceedings of the Oceans – Europe (2005) 814-819
- [28] R. Song, Y. Liu , R. Bucknall, A multi-layered fast marching method for unmanned surface vehicle path planning in a time-variant maritime environment, Ocean Engineering 129 (2017) 301-317
- [29] G. Li, Y. Tamura, A. Yamashita and H. Asama, Effective improved artificial potential field-based regression search method for autonomous mobile robot path planning, Int. J. Mechatronics and Automation, 3 (3) (2013) 141-170

- [30] R. K. Mandava, S. Bondada, P. R. Vundavilli, An Optimized Path Planning for the Mobile Robot using Potential Field Method and PSO algorithm, 7th International Conference on soft computing and problem solving (socpros-2017) Bhubaneswar, India, 2017
- [31] J. A. Sethian, A fast marching level set method for monotonically advancing fronts, *Proc. Nat. Acad. Sci. U.S.A.* 93 (4) (1996) 1591-1595
- [32] S. Jbabdi, P. Bellec, R. Toro, J. Daunizeau, M. Pélérini-Issac, and H. Benali, Accurate anisotropic fast marching for diffusion-based geodesic tractography, *Int. J. Biomedical Imaging*, 2008 (2) (2008)
- [33] H. Li, Z. Xue, K. Cui, and S. T. C. Wong, Diffusion tensor-based fast marching for modeling human brain connectivity network, *Comp. Med. Imag. and Graph.* 35(3) (2011) 167-178
- [34] K. Yang, M. Li, Y. Liu, and C. Jiang, Multi-points fast marching: A novel method for road extraction, *The 18th International Conference on Geo-informatics: GI Science in Change, Geo-informatics*, (2010) 1-5
- [35] S. Osher and J. A. Sethian, Fronts Propagating with Curvature Dependent Speed: Algorithms based on Hamilton-Jacobi Formulations, *Journal of Computational Physics*, 79 (1) (1988) 12-49
- [36] R. K. Mandava and P. R. Vundavilli, Whole body motion generation of 18-DOF biped robot on flat surface during SSP and DSP, *International Journal of Modeling Identification and Control*, In Press (2018)
- [37] Santiago Garrido, Luis Moreno, Dolores Blanco and Piotr Jurewicz, Path Planning for Mobile Robot Navigation using Voronoi Diagram and Fast Marching, *International Journal of Robotics and Automation (IJRA)* 2 (1) (2011) 42-64
- [38] 1P. Melchior, B. Orsoni, O. Lavialle, A. Poty and A. Oustaloup, Consideration of obstacle danger level in path planning using A* and fast-marching optimization: comparative study, *Signal Processing*, 11 (2003) 2387-2396
- [39] C. H. Chiang, P. J. Chiang, A comparative study of implementing Fast Marching method and A* search for mobile robot path planning in grid environment: effect of map resolution, in *Proc. IEEE Advanced Robotics and Its Social Impacts*, (2007) 1-6
- [40] Q. H. Do, S. Mita and K. Yoneda, A practice and optimal path planning for autonomous parking using fast marching algorithm and support vector machine, *IEICE Trans. Inf. & Syst.* 96 (12) (2013) 2795-2804
- [41] J. Vascak and M. Rutrich, Path planning in dynamic environment using Fuzzy Cognitive Maps, in *2008 6th International Symposium on Applied Machine Intelligence and Informatics* (2008) 5-9