CHAPTER X

PERFORMANCE STUDY OF THE FM2 PLANNING METHOD FOR REMOTE HANDLING OPERATIONS IN ITER

Javier V. Gómez[1], Santiago Garrido[1], Luis Moreno[1], Alberto Vale[2], Filipe Valente[2] and João Ferreira[2]

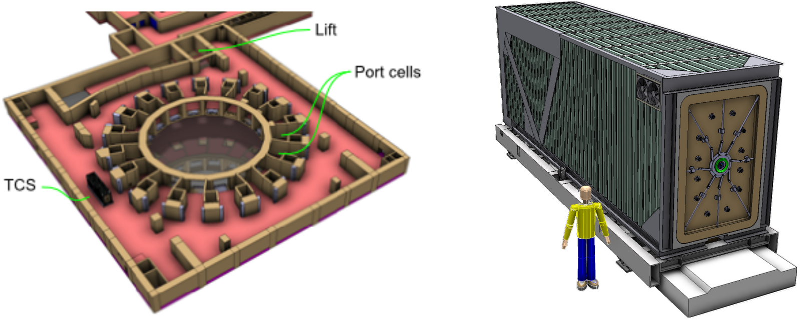
[1]RoboticsLab., Carlos III University of Madrid (UC3M).
[2]Instituto de Plasmas e Fusão Nuclear (IPFN), Instituto Superior Técnico, Universidade Técnica de Lisboa.

Along this chapter a path planning method called Fast Marching Square is proposed to be applied in ITER scenarios. This method is characterized by its robustness and its smooth and safe paths, including even velocity commands. Simulation results show that the proposed method provides promising results in the main building of ITER, the Tokamak Building.

1 Introduction

The current largest project in nuclear fusion technology is the ITER (International Thermonuclear Experimental Reactor). This project aims to make the long-awaited transition from experimental studies of plasma physics to full-scale electricity-producing fusion power plants. Among all the challenges it leads, this chapter focuses in Remote Handling (RH) operations of maintenance.

During the Tokamak operation, human being is not allowed to be inside in any level of the TB (as depicted in figure 1 a)). Hence, RH systems must guarantee maintenance and deactivation autonomous operations.



a) b)

Fig. 1. a) Level B1 of Tokamak Building. b) Cask and Plug Remote Handling System .

The Cask and Plug Remote Handling System (CPRHS) , as illustrated in figure 1 b) is the RH solution for maintenance operations of transportation of heavy and activated loads. During nominal operation, it is expected to move autonomously between the TB and its adjacent building, the Hot Cell Building (HCB). The total area is equivalent to a soccer field. The largest CPRHS has dimensions 8.5x2.62x3.62 meters (length, width, height) with a total weight up to 100 tons.

The nature of the ITER project requires robust path planning and motion algorithms. Beyond the obstacle avoidance, the cluttered environment and the large dimensions of the CPRHS convert this typical navigation problem in which the typical solutions are not enough and new, improved path planning algorithms are required.

In this context, this chapter studies the application of the Fast Marching Method to the path planning problem in the ITER scenarios. The next section describes the principles of this method. In section 3 the Fast Marching Method is applied to the ITER scenarios and the results are analyzed. Section 4 outlines another solution based on Fast Marching Square: using 3-

dimensional path planning. Finally, in section 5 the conclusions are presented.

2 Fast Marching Method and Path Planning

The Fast Marching Method (FMM), and more concretely the Fast Marching Square Method (FM2), is a very versatile one when applied to path planning problems. It has been successfully applied to many different problems such as autonomous exploration (Garrido, 2008), outdoor motion planning (DeSanctis, 2009) or even robot formation motion planning (Garrido, 2011). This leads to the application of this method to the ITER project.

2.1 Fast Marching Method

At the beginning, this method suggested by J. A. Sethian in 1996 was proposed to approximate the solution of the Eikonal equation (Sethian, 1996). Let us assume that a wave starts propagating in T = 0 with velocity F always non-negative. The Eikonal equation allows updating the time T for each position x according to:

$$|\nabla T(x)|F(x) = 1 \tag{1}$$

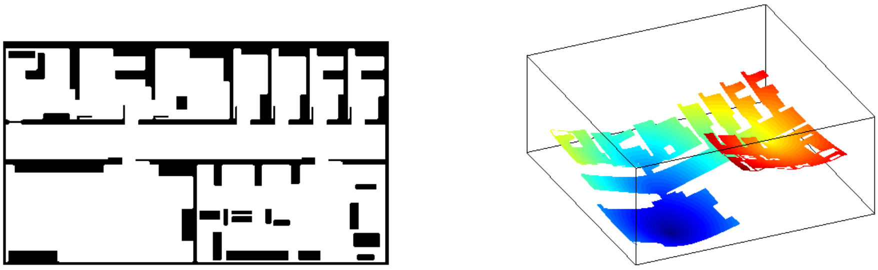
Discretizing the gradient ∇T according to (Osher, 1988) it is possible to solve the Eikonal equation at each point $p(x_i, y_j)$, which corresponds to the row *i* and column *j* of a grid map, as follows:

$$\begin{aligned} T_1 &= \min(T_{i-1,j}, T_{i+1,j}) \\ T_2 &= \min(T_{i,j-1}, T_{i,j+1}) \end{aligned} \tag{2}$$

$$\left(\frac{T_{i,j} - T_1}{\Delta x}\right)^2 + \left(\frac{T_{i,j} - T_2}{\Delta y}\right)^2 = \frac{1}{{F_{i,j}}^2} \tag{3}$$

The Fast Marching method consists on solving $T_{i,j}$ for every point of the map starting at the source point of the wave (or waves) where

$T_{i_0,j_0} = 0$. Once it has finished, the algorithm outputs a distance map as shown in figure 2.



a) b)

Fig. 2. a) Example grid map. b) Distance map provided by Fast Marching.

FMM can be directly used as a path planner algorithm. By applying gradient descent from any point of the distance map, a path will be obtained with the source of the wave as a goal point. This is valid only if one wave has been employed to generate the distance map. Otherwise, local minima will appear. The main advantage of this method is that the path obtained is optimal in distance. An example of a path provided by Fast Marching is shown in figure 3.

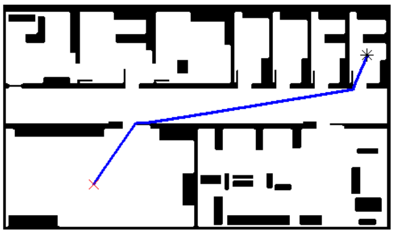


Fig. 3. Example of a path obtained with Fast Marching.

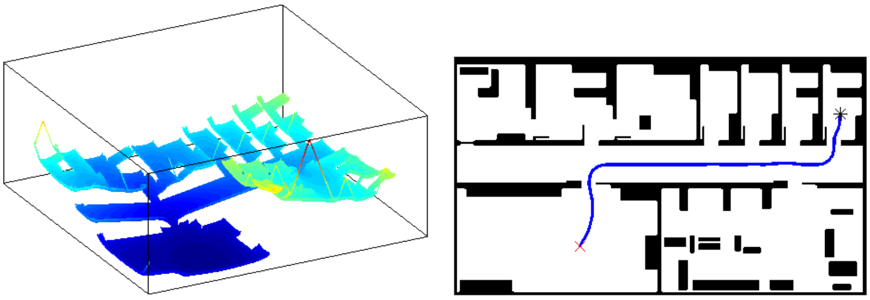
2.1 Fast Marching Square Method

Although the paths provided by the FMM are optimal in distance terms, they do not accomplish the smoothness and safety constraints that most of robotic applications require. These paths run too close to obstacles and walls and they have sharp curves. In view of these drawbacks, it turns out that the FMM algorithm is not enough in most of the situations. However, the FM2 algorithm (Garrido, 2009) solves these two main disadvantages.

The FM2 method is based on creating velocities (or slowness) maps depending on the environment map in which the velocity of the expanding wave varies depending on the distance to the closest obstacle. The FMM is applied in order to obtain these velocities maps. In this case, all the obstacles and walls are labeled as wave sources. Those waves expand with constant speed. The result is a distances map in which those cells in the grid that are farther from the obstacles have a higher value (figure 4).

Once this distance map is computed, it is normalized in order to have values between 0 and 1, i.e., full stopped or full speed, respectively. The FMM is then applied with the goal point as wave source. During the expansion, the wave will propagate with the velocities indicated in the map generated previously. The resulting distances map will be similar to the one obtained with the standard FMM, but with slight differences which make the paths very smooth when gradient descent is applied (figure 5).

Fig. 4. Velocities map obtained from the map of the figure 1 a).



a) b)

Fig. 5. a) Distances map after applying FMM over the velocities map. b) Final path obtained: smooth and safe.

It is possible to use the analogy with light waves propagation. The velocities map acts as a non-homogeneus medium in which the refraction index depends on the position. Therefore, according to the Fermat's principle (least time principle*): the path taken between two points by a ray of light is the path that can be traversed in the least time.* Assuming that the robot moves at the speed indicated by the velocities map, the FM2 method provides the shortest paths in terms of time.

Apart from the smoothness and safety, FM2 has other properties worth to mention:

- Absence of local minima. As long as only one wave is employed to generate the distances map, Fast Marching ensures that there will only be one global minimum at the source point of the wave (goal of the path).
- Completeness. The method will find a path if it exists. It will point out also if there is not a possible path.
- Fast response. The velocities map has to be calculated only once and the Fast Marching method has complexity order of O(n) (Yatziv, 2005). Even more, the simplicity with which the environment is modelled avoids complex calculations or costly sensory treatment.

3 Application of the Fast Marching Square Algorithm to the ITER Environment

Although the ITER scenarios are 2D, the structure of the building and the dimensions of the CPRHS turn this problem into a very hard one. Also, the ITER safety and operating requirements are quite strong. Hence, classical path planning methods need to be improved in order to accomplish all the restrictions.

Concretely, the ITER requirements focus on safety and smoothness. The large dimensions of the CPRHS, the cluttered environment and the contaminated nature of its load turn the motion planning into a very complex one, demanding a robust planner. The trajectories must be also the shortest in time while satisfying energy optimization for the CPRHS. Another very important requirement is that the minimum safety distance to obstacles to be guaranteed is 300 mm (Fonte, 2010 and Valente F, 2011).

In a previous study, it was observed that the most suitable kind of vehicle is the rhombic-like vehicle (Ribeiro, 1997). This type of vehicle is holonomic, that is, it can move in any direction, even rotating around its center thanks to its motion system, in which both are drive wheels and also steer (figure 6).

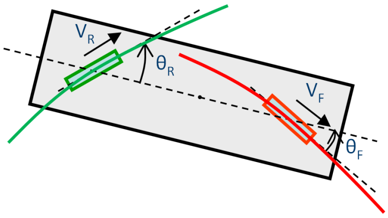
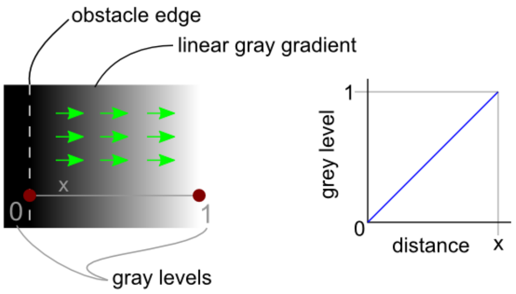


Fig. 6. Schema of a rhombic-like vehicle. This study is focused on both wheels following the calculated path.

Therefore, the dimensions of the vehicle and its spanned area (orientation dependent) while moving does not allow enlarge the obstacles and consider the robot as a single point. How this is faced with the FM2 is detailed in next sections.

3.1 Collision Detection based on FM2

Thanks to the fact that the velocities map has a value for every point directly proportional to the distance to the closest obstacle, it can be easily translated into a distances map. The most important point is that, calculating the gradient in each point of this map, the direction to the closest obstacle is also obtained. Since these operations are computationally expensive, it is important to stress that these maps should be calculated offline and once per map (a “gradients map” can be previously computed). These concepts are illustrated in figures 7 and 8.



a) b)

Fig. 7. a) Example of a velocities map and its gradient. b) Conversion from gray level to distance.

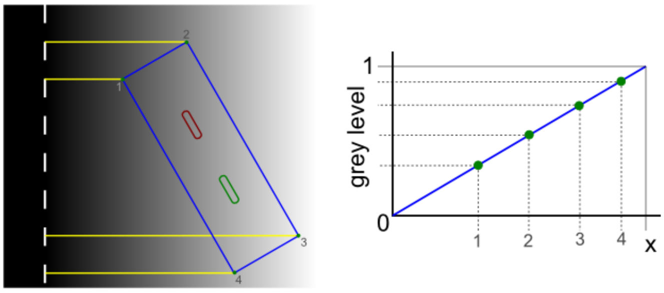
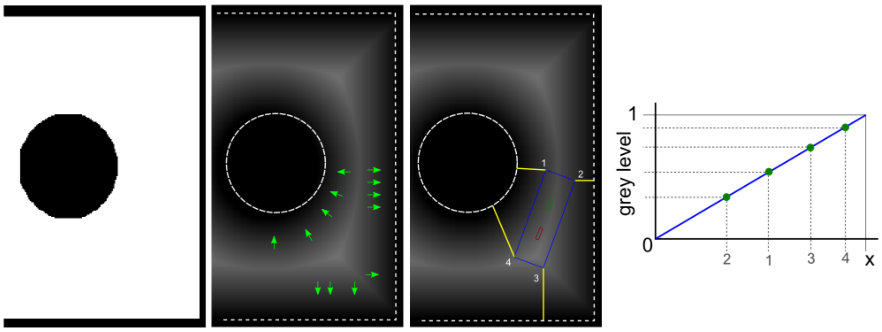


Fig. 8. a) Example of distance computation on the CPRHS. b) Distances in the corner points.

Therefore, to compute the distance to the closets obstacle (and also where it is), a set of as many as desired points are distributed along the perimeter of the CPRHS. The gray level of all these points is checked and the lowest one is chosen. Thus, it is already possible to obtain the distance to the closest obstacle and also its direction, according to figure 9.



a) b) c) d)

Fig. 9. a) Initial binary map. b) Some gradient arrows over the velocities map. c) Directions to the closest obstacles (only evaluated in the corners) d) Distances calculated from the velocities map for each corner.

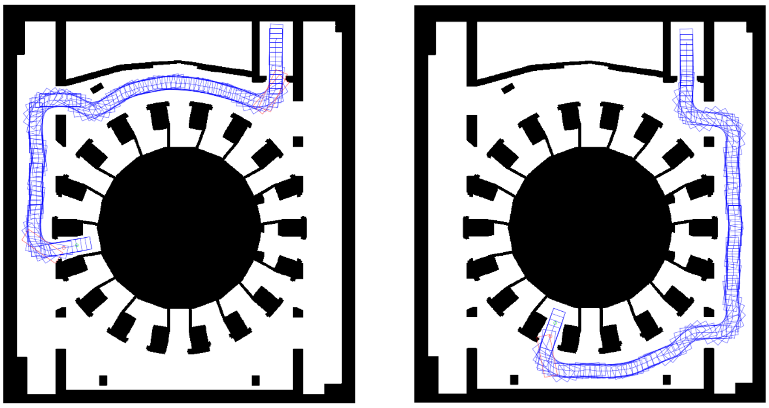
The results of the application of this algorithm with real ITER building map and CPRHS's dimensions are shown in the next section.

3.2 Collision Detection Validation and Simulation

In order to validate the proposed collision detection algorithm, simulation tests are carried out. In these, both wheels of the CPRHS have to follow the computed path. Also, the velocity command applied to the CPRHS is according to the position of the front wheel of the CPRHS (green) within the velocities map (thanks to the FM2 characteristics).

For the collision detection, one point every centimeter is places around the CPRHS's perimeter (a total of 2224 points). Also, the map dimensions are 1648x1450 pixels, which mean a grid size of 5 centimeters. All paths were splitted in 100 iterations.

Among the 18 possible trajectories, the simulation of those two (port cells 10 and 13) which could be more meaningful are shown in figure 11, 12 and 13, including their velocity profiles and also the distance to the closest obstacle of all iterations.



a) b)

Fig. 11. Path and CPRHS's poses from the lift to the port cell 10 and 13. Red CPRHS means collision.

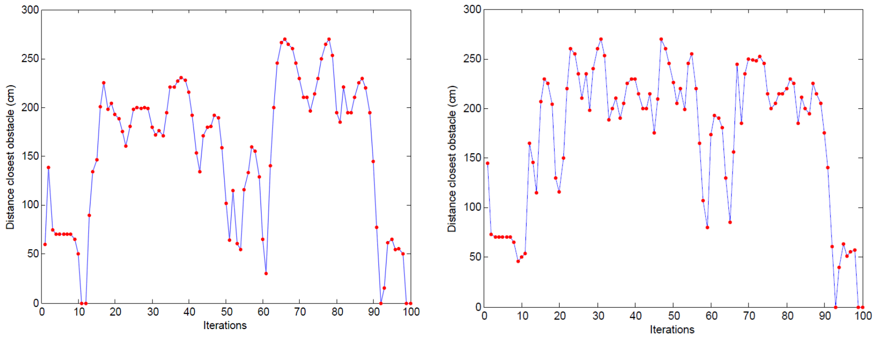


Fig. 12. a) Minimum distances during the motion for paths 10 and 13.

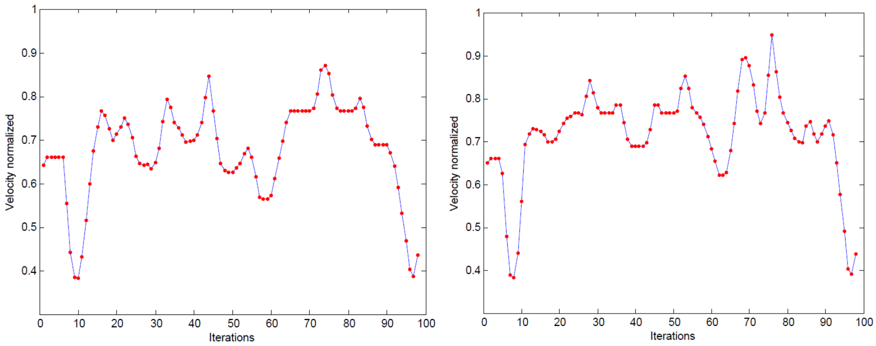


Fig. 13. a) Velocities profile for the same trajectories 10 and 13.

Simulations show that the FM2 method, with both wheels of the CPRHS following the path, has collision points in the lift and when entering the port cells. This is due the fact that FMM and FM2 does not take into account the kinematics of the vehicle when planning. However, thanks to the high mobility of the CPRHS and the FM2 collision checking algorithm, the best way to avoid these collisions is to detect the collision points and allow the wheels to move outside the path prior the collision occurs.

Therefore, it is mandatory to include a new layer in the path planning algorithm to remove these conflict points, leaving the rest of the path as is.

As additional information, the computation time of the trajectories use to be approximately 3.5 seconds (using Matlab in an Intel Core 2 Duo 3 GHz PC). Even more, the time it takes to check collisions is between 20 and 40 milliseconds.

4 Using a 3-Dimensional FM2 Method

With the collision detection algorithm proposed above, it is possible to create a 3D C-space of the environment, with the two dimensions of the CPRHS's position and the vehicle's orientation as the third dimension. Computing a trajectory along the C-Space built taking into account the vehicle's dimensions, it is possible to guarantee the absence of collisions.

Since the FM2 is based on the FM algorithm, it can be applied to more than 2 dimensions easily. The C-space has been built iteratively placing the CPRHS in every position and with every possible angle. This is a slow task, but it can be done offline and once per map. Results are shown in figure 14, in which maneuvers where forced to prove the collisions absence.

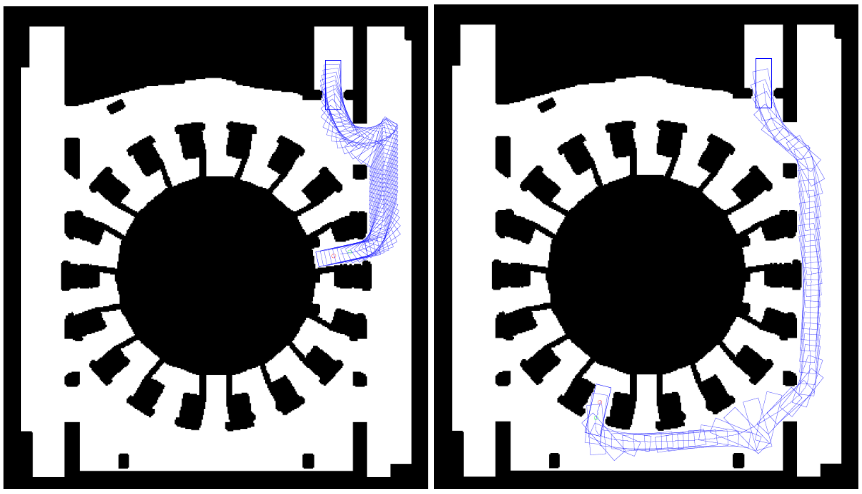


Fig. 14. a) Path computed in 3D to the port cell 1. b) Path computed in 3D to the port cell 13.

5 Conclusions and Future Work

Throughout this chapter the FM2 path planning method has been outlined and applied to the ITER-RH problem. Results show that a standalone FM2 algorithm is not enough to satisfy the ITER requirements: there are collisions in very concrete points of the scenario. However, the path quality (in terms of smoothness and safety) is high in the rest of the points.

A variation of the standard FM2 method has been proposed, planning 3-dimensional trajectories. These trajectories are collision-absent but the space spanned by the CPRHS could become a problem.

The application of FM2 to the ITER has still many challenges: the grid map discretization can become a problem is more resolution is required, making the algorithm slower. Even more, although the FM2 is safe enough, the dimensions of the CPRHS provokes some collisions, which is the main point to be solved.

Acknowledgements

This work is funded by the project number DPI2010-17772, by the Spanish Ministry of Science and Innovation. Also, this work was supported by FCT in the frame of the Contract of Associate Laboratories of Instituto de Plasmas e Fusão Nuclear/IST (Pest-OE/SADG/LA0010/2011).

References

(DeSanctis, 2009) De Sanctis, L. et al. 2009. Outdoor motion planning using Fast Marching. In Proc. Int. Conf. Climbing and Walking Robots and the Support Technologies for Mobile Machines, Istambul, Turkey.

(Fonte, 2010) Fonte, D. et al. 2010. A motion planning methodology for rhombic-like vehicles for ITER remote handling operations. Proceedings of the 7th IFAC Symposium on Intelligent Autonomous Vehicles, Lecce, Italy, pp. 210–215.

(Garrido, 2008) Garrido, S. et al. 2008. Exploration of a cluttered environment using Voronoi transform and Fast Marching method. Journal of Robotics and Autonomous Systems – 56(12), pp: 1069–1081.

(Garrido, 2009) Garrido, S. et al. 2009. FM2: a real-time sensor-based feedback controller for mobile robots. Int. Journal of Robotics and Automation – 24(1): 3169–3192.

(Garrido, 2011) Garrido, S. et al. 2011. Robot formation motion planning using Fast Marching. Journal of Robotics and Autonomous Systems– 59(9), pp: 675–683.

(Osher, 1988) Osher, S. et al. 1988. Fronts Propagating with Curvature-Dependent Speed: Algorithms Based on Hamilton-Jacobi Formulations. Journal of Computational Physics – 79: 14–49.

(Ribeiro, 1997) Ribeiro, M. I. et al. 1997. Conceptual study on flexible guidance and navigation for ITER remote handling transport casks. Pro-

ceedings of the 17th IEEE/NPSS Symposium on Fusion Engineering, San Diego, USA.
(Sethian, 1996) Sethian, J. A. 1996. A fast marching level set method for monotonically advancing fronts. Proceedings of the National Academy of Science – 93(4): 1591–1595

(Valente F, 2011) Filipe Valente, Alberto Vale, Daniel Fonte, Isabel Ribeiro. Optimized Trajectories of the Transfer Cask System in ITER.
Fusion Engineering and Design, 86 (2011) 1967–1970.

(Yatziv, 2005) Yatziv, L. et al. 2005. A fast O(n) implementation of the Fast Marching algorithm. Journal of Computational Physics –212(1): 393–399.