Gómez, Javier V., …et al. (2015). Performance analysis of fast marching-based motion planning for autonomous mobile robots in ITER scenarios. *Robotics and Autonomous Systems*, v. 63, Part 1, pp.: 36-49.

DOI: https://doi.org/10.1016/j.robot.2014.09.016

Performance analysis of Fast Marching-based motion planning for autonomous mobile robots in ITER scenarios

Javier V. Gómez[a,*], Alberto Vale[b], Santiago Garrido[a], Luis Moreno[a]

[a] *RoboticsLab., Carlos III University of Madrid, Avda. de la Universidad 30, 28911, Leganés, Madrid, Spain.*
[b] *Instituto de Plasmas e Fusão Nuclear at Instituto Superior Técnico, Universidade de Lisboa, Portugal*

Abstract

Operations of transportation in cluttered environments require robust motion planning algorithms. Specially with large and heavy vehicles under hazardous operations of maintenance, such as in the ITER, an international nuclear fusion research project. The load transportation inside the ITER facilities require smooth and optimized paths with safety margin of 30 cm. The transportation is accomplished by large rhombic-like vehicles to exploit its kinematic capabilities. This paper presents the performance analysis of a motion planning algorithm to optimize trajectories in terms of clearance, smoothness and execution time in cluttered scenarios. The algorithm is an upgraded version of a previous one used in ITER, replacing the initialization implemented using Constrained Delaunay Triangulation by the Fast Marching Square. Exhaustive simulated experiments have been carried out in different levels of ITER buildings, comparing the performance of the algorithm using different metrics.

Keywords:
Motion planning, fast marching square, rigid body dynamics, ITER

*Corresponding author
Email address: `jvgomez@ing.uc3m.es` (Javier V. Gómez)
URL: `http://roboticslab.uc3m.es/` (Javier V. Gómez)

1. Introduction

Path planning is one of the key issues for hazardous transport operations using autonomous mobile robots. Not only for scenarios of disaster, but also when working in experimental scenarios testing new sources of energy where human being is not allowed. In particular, the International Thermonuclear Experimental Reactor (ITER) is a worldwide research experiment that aims to explore nuclear fusion as a viable source of energy for the coming years. The ITER project aims to make the long-awaited transition from experimental studies of plasma physics to full-scale electricity-producing fusion power plants. The largest experimental tokamak nuclear fusion reactor, depicted in Figure 1, will be located at the Cadarache facility, in the south of France.

Besides the major scientific objective of exploring the nuclear fusion as a source of energy, ITER aims to demonstrate that the future fusion power plants can be safely and effectively maintained through Remote Handling (RH) techniques, due to restrictions on human being in activated areas. The RH approach must be from the outset as flexible as possible with minimum reliance on the tokamak configuration, such as in ITER, [1].

The top level maintenance functions of RH in ITER are the exchange of blanket segments, divertor cassettes, perform in-vessel inspection and recovery tasks, allow remote maintenance of ex-vessel systems including heating and current drive systems, ex-vessel transfer casks and servo manipulators. In particular, the maintenance functions of ex-vessel transfer cask has a relevant reliance not only with the reactor, but with the entire power plant. Transportation of equipment for storage, refurbishment and repair requires vehicles of transportation that navigate along corridors of the power plant. A transport cask system (simply identified as cask) is required to accomplish the maintenance operations that includes transportation. Pre-computed paths assuming the well-known scenarios are expected for nominal operations. However, during the maintenance operations, new paths must be computed. For instance, when a cask fails, another rescue cask has to dock into the first one, remove the activated load and then drive it to the maintenance area.

The cask, depicted at the bottom of Figure 1, is a large and complex unit to transport heavy and contaminated components between the two main buildings of ITER: the Tokamak Building (TB), where the reactor is installed, and the Hot Cell Building (HCB) for refurbishment and storage. The geometry of the cask and its payload vary according to the components

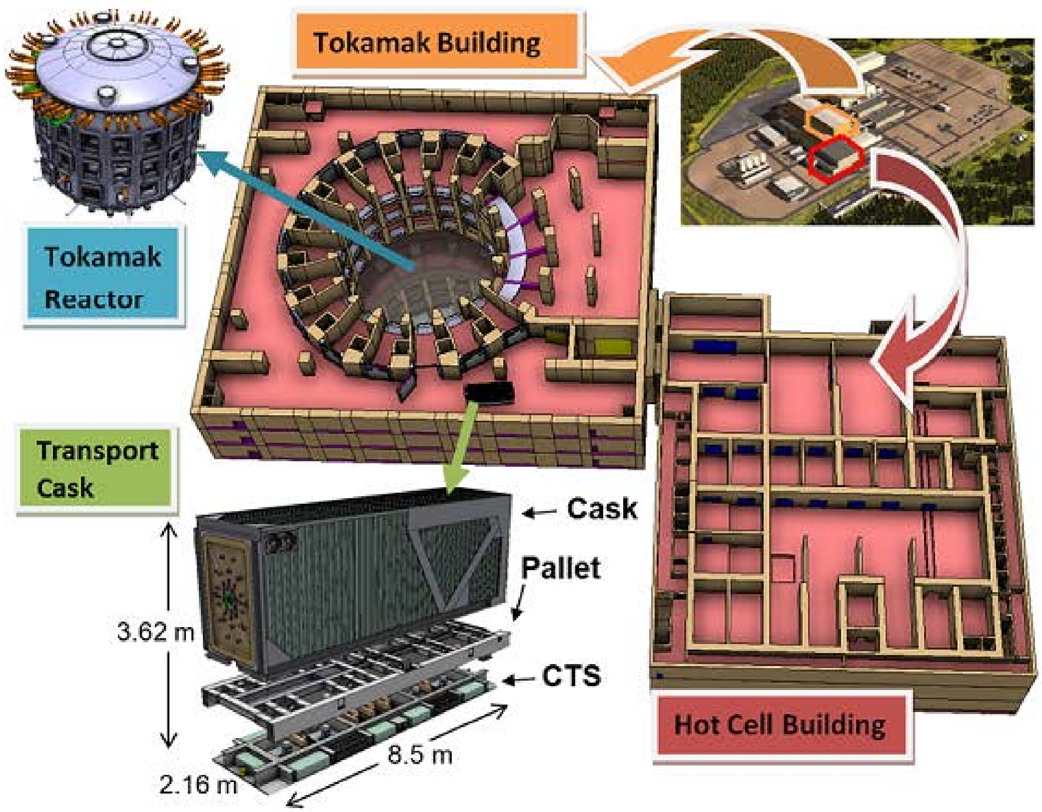

Figure 1: The ITER tokamak and the scientific buildings and facilities that will house the ITER experiments in Cadarache, South of France.

to be transported and hence, different cask typologies are expected. As a reference, the largest cask dimensions are 8.5m x 2.62m x 3.62m (length x width x height) and the total weight can reach up to 100 tons.

The maintenance operations of transportation require the cask to move throughout the cluttered environments of the TB and the HCB, equivalent to drive a truck under 30 cm of safety margins to the closest walls and pillars. The constrained space may also rise a logistic problem, where multiple vehicles have to move and different paths must be computed.

The kinematics of the cask are equivalent to a rhombic like vehicle, with two drivable and steerable wheels. Given this configuration, proposed in [2, 3], the cask has a higher maneuverability in confined spaces than the traditional cars with Ackerman or tricycle configurations [4].

From previous work of RH in ITER, the optimized paths would be implemented on the scenario using buried wired systems [2]. Presently, the buried wired systems are being superseded by other systems, as a line painted on the floor or, simply, by a virtual path. These systems are used in several Automatic Guided Vehicles (AGV) applications [5, 6, 7]. In this navigation methodology, the vehicles would follow the path by using a line guidance approach: both wheels following the same path. The proposed planning methodology returns directly the path to be followed by the center of the wheels and not the one corresponding to the center of the vehicle (identified as the free roaming, out of the scope of this paper).

A nominal operation of the vehicle for a specified environment determines a motion between two configurations (2D points with specific orientations). The first step of this planning methodology is to find an initial geometric path, i.e. a set of 2D points, connecting the initial and final configurations. The previous implemented approach was based on the Constrained Delaunay Triangulation (CDT) [8, 9]. This solution presents some limitations in terms of path smoothness. In complex scenarios, the geometric representation results in a huge number of triangles with rough initial paths still far from the optimal one, yielding to a computational effort [10]. The Fast Marching Square path planning method (FM^2) [11] is an alternative approach for the initialization. The FM^2 provides an initial path closer to the optimal solution and in a shorter period of time, resulting in an improvement of the computational effort.

Previous works have already addressed the application of the Fast Marching Method (FMM) to kinematically constrained systems. Concretely, one of the first approaches is an iterative method which in every iteration computes a different path [12]. If this path does not satisfy the kinematic constraints, the obstacles are smoothly dilated, so that the next computed path will have smoother, larger curves. This process is repeated until a valid path was found. Another different approach is to compute an initial path with FMM and then propagate a second FMM wave within a tube in the initial path surroundings [13, 14]. In this second wave expansion, the FMM is modified so that neighbors of the grid cell are no longer computing according Von Neumann neighborhood, but are computed by propagating the system with different input actions. The main drawback of this problem is its computational complexity. Ryo et al. [15] proposed a new Hamilton-Jacobi formulation for computing optimal trajectories for systems with limited curvature. This formulation has been successfully applied to Dubin's and ReedsShepp car models. However, the whole formulation needs to be done from scratch for every different kinematic system.

Authors' previous work in this topic focuses on the application of FM^2 in a 3-dimensional configuration space (two spatial dimensions and the vehicle orientation) [16]. When applying FM^2 in this configuration space, smooth paths are guaranteed. However, this approach did not take into account explicitly the kinematic constrains. However, in this work the regular 2-dimensional version is being applied since the vehicle employed (detailed in section 2.2) do not have kinematic constraints. However, the vehicle's kinematics and dimensions require a similar study to that carried out in vehicles with such constraints.

During the maintenance operations of transportation, the pose (position and orientation) of each vehicle must be continuously evaluated using sensors data. Although the first studies of localization of the CTS in ITER have been accomplished, as described in [17], in this paper it is assumed that the pose of the vehicle is always known without any uncertainty.

The paper is organized as follows. The Section 2 describes the problem statement: the scenario, the vehicle, the goals and the optimization criteria. The Section 3 introduces the FMM and how it is implemented to become the FM^2. The Section 4 describes the optimization in terms of clearance and smoothness applied to the paths returned by the FM^2. The Section 5 presents simulated results in the ITER scenarios. Conclusions and future work are pointed out in Section 6.

2. Problem Statement

The problem statement description is divided into four different issues: the environments, the vehicles, the missions (goals) and the optimization criteria.

2.1. Environments

The TB, shown in Figure 2, lodges the tokamak reactor with access by vacuum vessel port cells (from this point forward simply identified as ports). The HCB, depicted in Figure 3, will work mainly as a support area. A lift establishes the only interface between the different levels of TB and the HCB.

In ITER, the environments in all levels of TB and HCB are mostly composed by static and well structured scenarios. Each level of the buildings is modeled as 2D map representation, M, with a set of 2D points, p_i, on the

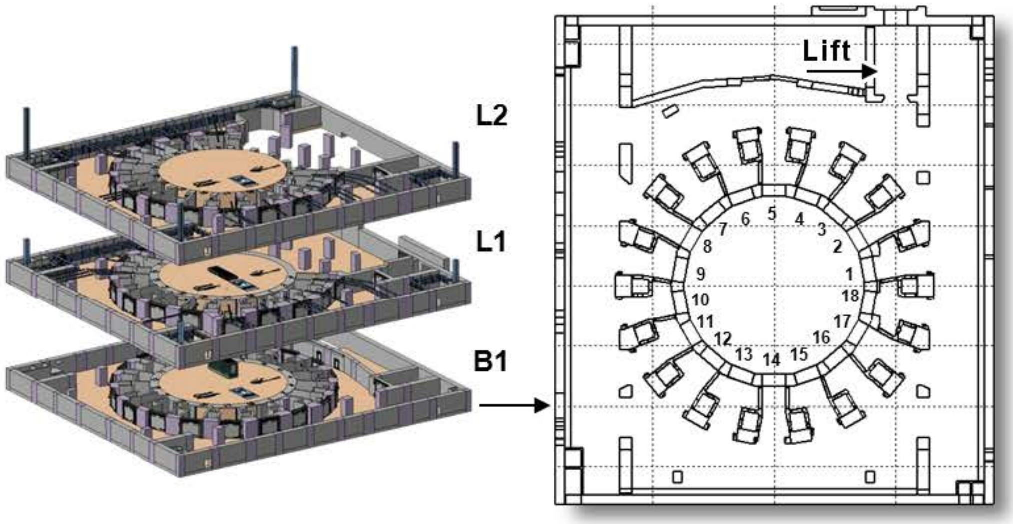

Figure 2: The three main level of Tokamak Building in ITER (left) and the 2D representation of the level B1 (right).

global Cartesian referential of ITER and a set line segments, l_{jk}, where each line segment connects two different points p_j and p_k, i.e.,

$$M = \{p_i, l_{jk} | i, j, k = 1, \ldots, M_p\} \tag{1}$$

where M_P is the number of points, $p_i = (x, y)$ and $l_{jk} = \{(1-t) \cdot p_j + t \cdot p_k | t \in [0, 1]\}$.

2.2. Transport cask

The vehicle, represented in Figure 1, is a large and complex unit to transport heavy and contaminated load between the TB and the HCB. The geometry of the vehicle and its payload vary according to the cask and the components to be transported and hence, different vehicle typologies will operate.

The vehicle is composed by three sub-systems: the cask envelope, the Cask Transfer System (CTS), and the pallet. The cask envelope is a container that enclosures the in-vessel components and the RH tools to be transported. The CTS acts as a mobile robot. The pallet is the interface between the cask and the CTS. It is equipped with a handling platform to support the cask load and help on docking procedures. When underneath the pallet the CTS

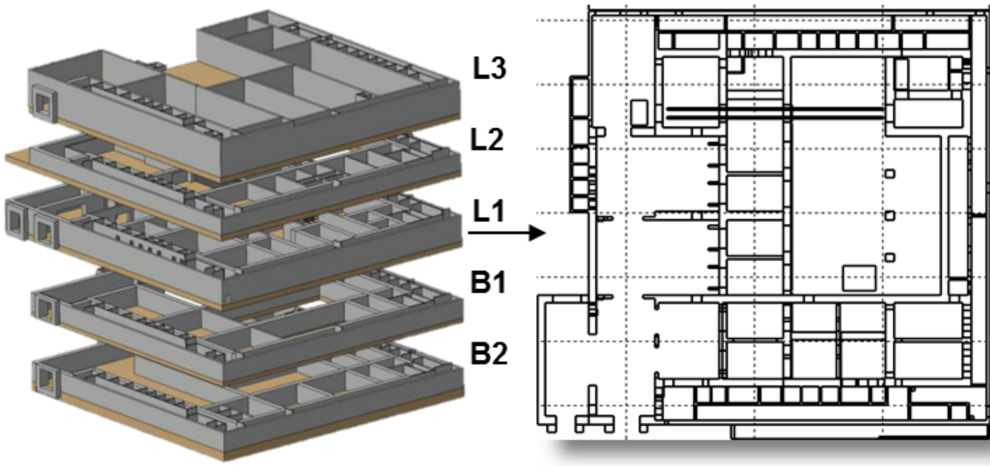

Figure 3: The five main levels of Hot Cell Building in ITER (left) and the 2D representation of the level L1 (right).

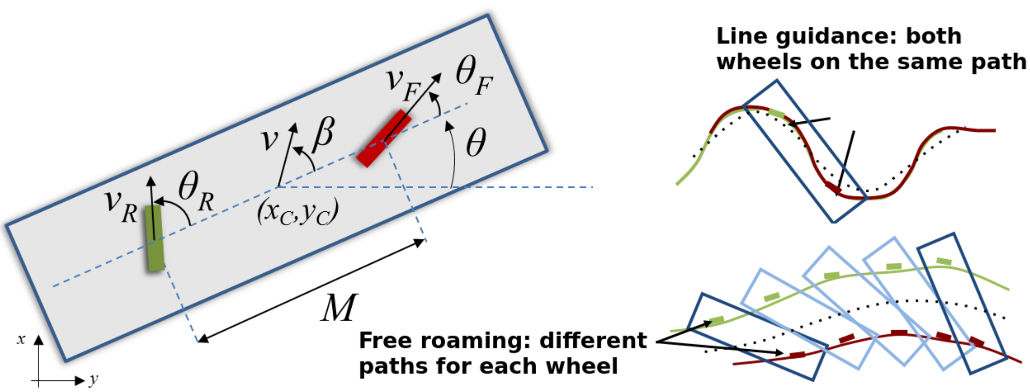

Figure 4: Rhombic vehicle model and the possible path following approaches.

transports the cask, but it can also move independently of the pallet and cask.

The CTS is equipped with two pairs of drivable and steerable wheels: one for nominal operation and the other for redundancy, operating in case of failure of the first pair, [18]. These locomotion wheels are installed along the longitudinal axis the vehicle, providing the rhombic like capabilities. Since the locomotion wheels are installed along a straight line, there are free wheels in the vicinity of the boundaries of the vehicle's shape to assure the CTS's stability. For simplicity and from this point forward, the CTS is only represented with a single pair of drivable and steerable wheels, identified as 'F'ront and 'R'ear wheels, as illustrated in Figure 4. This configuration gives the vehicle a higher maneuverability in confined spaces than the traditional cars with Ackerman or tricycle configurations [4].

As illustrated in Figure 4, consider the state vector $q = [x_c\, y_c\, \theta]$ as a representation of the vehicle pose in the frame {I}, with (x_c, y_c) the coordinates of the center of the vehicle and θ the orientation of the vehicle. Also, consider v as the longitudinal speed and β the controllable side-slip angle of the vehicle, both defined in {I}. A kinematic model for a rhombic like vehicle in {I}, that allows the simulation of the vehicle motion directly through the desired longitudinal speed v, instead of imposing an individual linear speed for each wheel, was introduced in [19] as:

$$\begin{bmatrix} \dot{x}_c \\ \dot{y}_c \\ \dot{\theta}_m \end{bmatrix} = \begin{bmatrix} \cos(\theta+\beta) \\ \sin(\theta+\beta) \\ \frac{\cos\beta\cdot[\tan\theta_F - \tan\theta_R]}{M} \end{bmatrix} \cdot v, \tag{2}$$

where

$$\beta = \arctan\left(\frac{v_F \cdot \sin\theta_F + v_R \cdot \sin\theta_R}{2 \cdot v_R \cdot \cos\theta_R}\right) \tag{3}$$

and

$$v = \frac{v_F \cdot \cos\theta_F + v_R \cdot \cos\theta_R}{2 \cdot \cos\beta}. \tag{4}$$

This modeling entails that the wheels of the vehicle roll without slipping, a constraint inherent to the nonholonomy of rhombic like vehicles, and also considers a rigid body constraint, common to this type of vehicles, as follows:

$$v_F \cos\theta_F = v_R \cos\theta_R. \tag{5}$$

The values v_F, v_R, θ_F and θ_R are the inputs to guide the vehicle, as detailed in [20].

2.3. Missions

The maintenance operations of transportation in ITER require the vehicle to move throughout the cluttered environments of the TB and the HCB, i.e. a mesh of paths between the target poses inside the buildings. For instance, a mission of transportation of a load for refurbishment requires a path between a port and the lift in TB and then between the lift and a docking port in HCB.

2.4. Optimization criteria

During a mission the vehicle describes a swept volume when follow its path. The volume is important given the free space available in the scenario, or given other parked vehicles. The speed of the path following is also relevant not only for the mission execution time, but in particularly given the dynamics of the vehicle, since it can reach up to 100 tons. As a result, each mission requires an optimized trajectory.

The trajectory optimization problem stated for the vehicle consists on evaluating a trajectory, i.e. a geometric path defined by a set of N points P_i, i.e., $S = \{P_0, P_1, \ldots, P_N\}$, combined with a speed profile. The geometric path must guarantee that the vehicle departs from the initial configuration q_S and achieves the specified goal q_F, without colliding with obstacles and keeping a safety margin. The trajectory optimization has three stages: the geometric path evaluation, the path optimization and the trajectory evaluation. The geometric path evaluation aims to find a path connecting the initial and goal configurations. This path acts as an initial condition for the path optimization stage. The geometric path evaluation is implemented using FM^2, as described in Section 3. The path optimization receives the preceding geometric solution as input and returns an optimized path. The optimization process, described in Section 4, first applies a spline interpolation to satisfy weaker differential constraints such as smoothness requirements. Afterward, a clearance based optimization is carried out to guarantee a collision free path that meets the safety requirements. In general, a minimum safety distance between the vehicle and the obstacles must be guaranteed. Finally, the trajectory evaluation defines the velocity function along the optimized path transforming it into a trajectory, which is the final output.

3. Fast Marching Methods in Path Planning

The FM^2 was firstly introduced by Garrido et al. [11]. Since then, it has been successfully applied to many different problems [21] and novel versions of the algorithm have been proposed [22]. It consists on applying twice the FMM, originally proposed by J.A. Sethian [23]. Both methods are intuitively explained in the following subsections.

3.1. The Fast Marching Method

The FMM [23] is an efficient numerical algorithm for modeling a wave front evolution through media with different propagation velocities. It is a particular case of the level set methods [24] in which the wave always expands outwards, that is, with non-negative velocity.

The FMM computes the time T a wave takes in order to reach every point of the space using a dynamic programming scheme. Let us assume a bi-dimensional grid map. The wave source point x_0 is given a value $T_0 = 0$. The FMM follows a upwind propagation procedure to solve the wave arrival time $T_{i,j}$ at each point with coordinates (i, j) according to the discrete Eikonal equation [24]:

$$\max\left(\frac{T - T_1}{\triangle x}, 0\right)^2 + \max\left(\frac{T - T_2}{\triangle y}, 0\right)^2 = \frac{1}{F_{i,j}^2} \tag{6}$$

where $\triangle x$ and $\triangle y$ are the grid spacing in the x and y directions, $F_{i,j}$ is the wave propagation speed for grid cell (i, j) and

$$\begin{aligned} &T = T_{i,j} \\ &T_1 = \min(T_{i-1,j}, T_{i+1,j}) \\ &T_2 = \min(T_{i,j-1}, T_{i,j+1}) \end{aligned} \tag{7}$$

The output is a distances map or, more properly, a times-of-arrival map. It is possible to have many wave source points, all of them with value $T = 0$.

When a constant velocity propagation is used the paths, computed as geodesics in $T(x)$, are optimal in terms of distance but they are not smooth and run too close to obstacles. In order to get a more detailed, formal description of the FMM, we refer interested readers to [23, 22].

3.2. The Fast Marching Square Path Planning Method

When abrupt changes occur in the wave velocity, geodesics lose their smoothness. In other words, they become not differentiable when there is a transition in the wave propagation speed. In fact, it is analogous to geometric optics and how light rays deform in media with different refraction indices. In case of a continuous gradient in the velocities map, the path is deformed smoothly (Figure 5).

The FM^2 is a path planning method designed to leverage the continuous curvature of the paths in the presence of velocities gradients [11]. It is based on the application of the FMM twice: firstly in order to create a velocities map of the environment and secondly to compute a path between two given points. The steps of the algorithm are the following:

- **Environment, W_0:** it is represented as a binary grid map in which obstacles are labeled as 0 (black) and the rest of the space as 1 (white).
- **Velocities map, F:** the FMM is applied using all the cells labeled as obstacles as wave sources. Then, the resulting map is rescaled so that its values are between 0 and 1. This step is actually the computation of a smooth distance transform.
- **Times-of-arrival map, T:** given an initial and goal point for the path, the FMM is applied using the goal point as a wave source. The wave propagates according to the velocities map F. The propagation is stopped once the start point of the path is reached.
- **Path extraction:** gradient descent is applied over T from the start point until the unique minimum of T, the goal point, is reached.

These steps are depicted in Figure 6. The main characteristics of the FM^2 method are path smoothness and safety in terms of obstacle clearance.

4. Path optimization and trajectory evaluation

An optimization methodology based on the elastic bands method [25] was designed [26]. The original concept associated with this approach appeared in the computer vision field, with the presentation of the so called “snakes” algorithm [27]. A snake is a deformable curve guided by artificial forces that pull it towards image features such as lines and edges. The solution herein

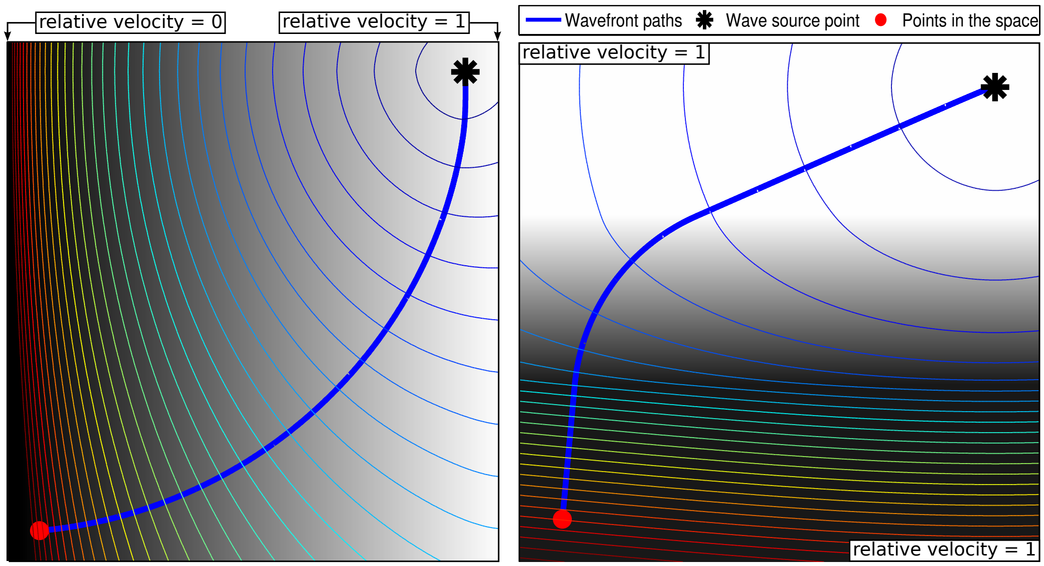

Figure 5: Examples of the geodesic paths obtained when there is a continuous gradient in the wave propagation velocities.

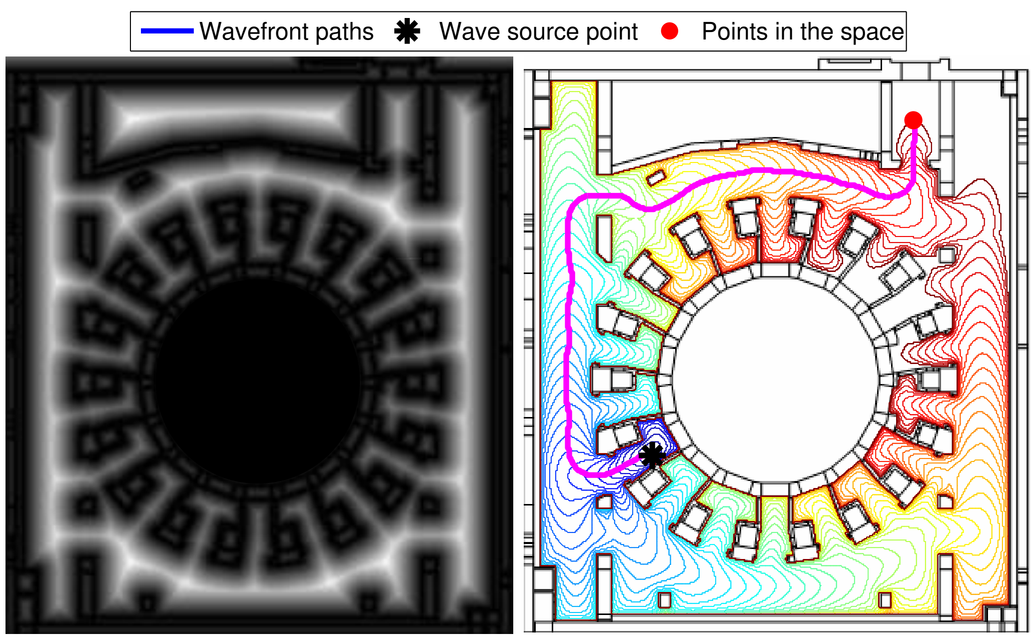

Figure 6: Velocities map for the level B1 of TB (left) and the times-of-arrival map over the initial binary map and the resulting path from the lift to the port 11 (right).

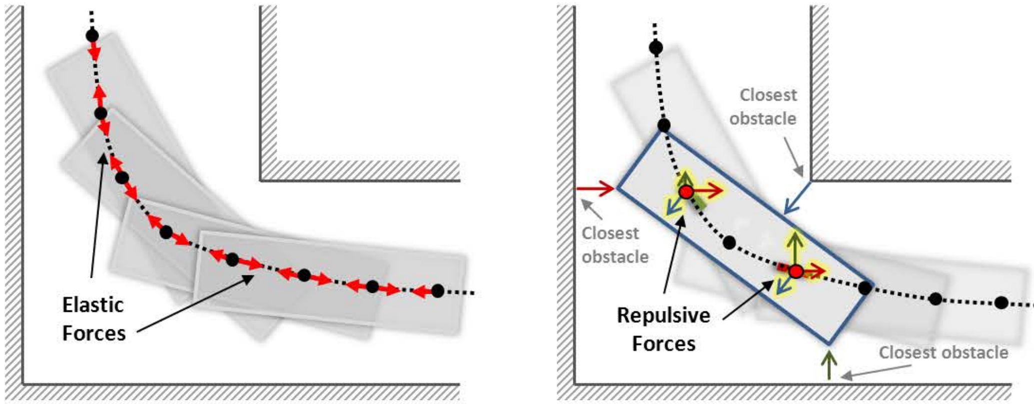

Figure 7: Elastic band concept: elastic forces to smooth the path (left) and repulsive forces generated by the closest obstacles (right).

proposed with the elastic bands methodology is similar to the snakes approach. Instead of retracting a curve to image features, in the path planning problem, it repels the path out from obstacles. Following this approach, the path is modeled as an elastic band which can be compared to a series of connected springs subjected to two types of forces, as illustrated in Figure 7:

- Internal forces or elastic forces: the internal contraction force simulates the Hooke's elasticity concept [28, 29], i.e., the magnitude force is proportional to the amplitude of displacement. This modeling approach allows the simulation of the behavior of a stretched band. This is the reason why the paths become retracted and shorter.

- External forces or repulsive forces: the obstacle clearance is achieved using repulsive forces to keep the path, and consequently the vehicle, away from obstacles.

When submitted to these artificial forces, the elastic band is deformed over time becoming a shorter and smoother path, increasing clearance from obstacles. Hooke's law evaluates the elastic force F_e applied to path point P_i as

$$F_e(P_i) = k_e \cdot [(P_{i-1} - P_i) - (P_i - P_{i+1})] \quad (8)$$

where k_e is the elastic gain and P_{i-1} and P_{i+1} are the path points adjacent to P_i. The elastic band behavior can be controlled through k_e. The band stretches with high values of k_e while low values increase the band flexibility.

To determine the external forces, a collision detector algorithm is used to evaluate the nearest obstacle point (OP) to each vehicle pose. The use of a single OP to determine the repulsive forces may not be satisfactory to maintain clearance from obstacles, and therefore, a larger set of obstacle points, such as the k-nearest (k-OPs), must be considered, as illustrated in Figure 7. This leads to a more balanced repulsive contribution ensuring effectiveness on most situations.

The repulsive force for each P_i is determined as a combination of different repulsive contributions

$$F_r(P_i) = k_r \cdot \sum_{l=\{F,R\}} \sum_{k=1}^{K} r_{l,k}(P_i) \tag{9}$$

where k_r is the repulsive gain, F and R the front and rear wheels and $r_{l,k}$ is the inverse of the distance between the k-OPs and the vehicle, considering the front and rear wheels, as detailed in [8].

Once the elastic (8) and the repulsive (9) forces are computed, an update equation procedure that defines the path evolution along each iteration is applied as

$$P_{i,new} = P_{i,old} + k \cdot F_{total}(P_{i,old}) \tag{10}$$

where k is a normalization factor adding up the total force contribution applied to all points $P_{i,old}$ and the total force contribution is given by

$$F_{total}(P_{i,old}) = F_e(P_{i,old}) + F_r(P_{i,old}) \tag{11}$$

Under the influence of these artificial forces, the elastic band is deformed over time becoming a shorter and smoother path. The stopping criteria is defined by detecting that the magnitude changes on F_{total} are smaller than a given threshold and by setting a maximum number of iterations. The path optimization is thus carried out by a path deformation approach where the computed paths are treated as flexible and deformable bands. Elastic interactions smooth the path by removing any existing slack, whereas repulsive forces improve clearance from obstacles by pushing away the points of the path.

The output of the path optimization is a collision free path suitable for execution. Then, the optimized paths are parametrized in terms of velocities, converting the paths into trajectories. Since the safety requirements are mandatory and the risk of collision shall be reduced in the cluttered environment, an initial approach defines the vehicle speed profile as a function of the distance to the obstacles. The velocity assumes low values when the vehicle is closer to the obstacles. Otherwise, the velocity is higher, under safety levels. To generate this initial speed profile, the minimum distance from the vehicle to the closest obstacle is identified for each point in the optimized path.

A maximum and minimum allowable speed are set to this profile, in order to integrate kinematic constraints. The safety margin identifies the threshold distance above which the maximum speed is considered. The speed profile thus obtained is saturated when the distance is above the threshold or below the safety margin and is referred as C-based speed profile [9]. However, the C-based speed profile is unable to handle vehicle dynamics constraints, meaning that the constraints on the admissible accelerations of the vehicle are ignored. To sidestep this issue, it has been developed a specific routine, which tests each one of the C-based speed profile transitions, checking whether the accelerations are feasible or not. Whenever a dynamic unfeasible transition is found (e.g., the calculated acceleration is out of the admissible bounds), the routine corrects the speed accordingly.

The output of the entire process, summarized in Figure 8, is an optimized path in terms of distance and smoothness, with a speed profile, assuming that the vehicle starts and ends with velocity equal to zero.

5. Simulated Results

The algorithms were implemented in MATLAB environment and integrated in the specially designed software tool Trajectory Evaluator and Simulator (TES). The TES receives the models of the buildings, generates trajectories using line guidance, evaluates the resulted 3D volume swept by the vehicle along the optimized paths and exports the optimized trajectories and the corresponding 3D swept volume directly to the CAD software. The TES provides also a GUI to preview the trajectory optimization, to manipulate the scenarios, to easily choose the vehicle typology to be used in the simulation and to generate results. The output of TES is a set of optimized trajectories which were validated by an independent software developed by ASTRIUM SAS [30]. The results achieved by the algorithms implemented in TES applied in the models of the real scenarios were important to proceed with the construction of the Tokamak Building.

The line guidance algorithm, using the FM^2 and the elastic bands optimization, was applied and tested in some levels of the TB and HCB to generate trajectories. Each optimized trajectory corresponds to a nominal

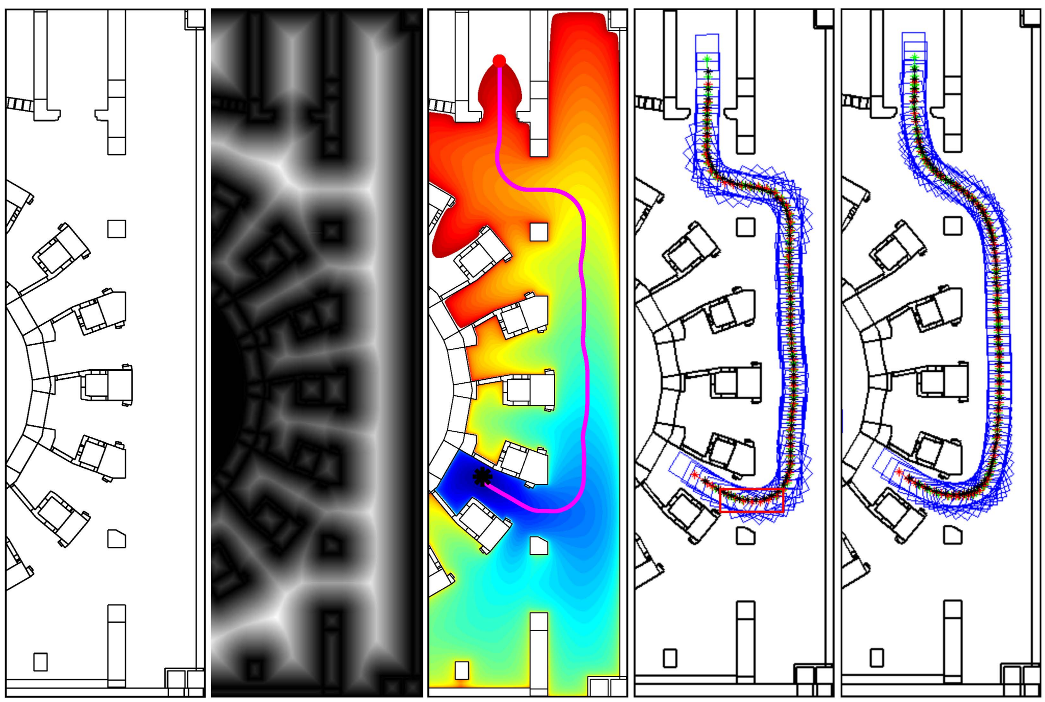

Figure 8: Steps of the proposed method on level TB/B1 (from left to right): initial map, velocities map, times-of-arrival map and FM^2 path, FM^2 path evaluated with the cask with a collision, and path after the the optimization.

operation of transportation between the lift and a port in the TB or a parking place in the HCB. Each mission only constrains the initial and final poses (positions and orientations) of the vehicle. A path is considered feasible when the minimum distance between the vehicle and the closest elements of the scenario is above a safety margin. This minimum clearance is only allowed to be infringed when entering/exiting the lift and in the docking phase.

First, an individual result of the optimization procedure is included before proceeding with the full results. This example is a mission from the lift to the port 10 of the level L1 of the TB, since it is one of the most complicated cases. Figure 9 shows that the FM^2 initialization contains points in the path with collisions, since the standard FM^2 in 2 dimensions does not take into account the size of the vehicle when planning. However, the optimization provides a shorter, smoother path without collisions. In Figure 10 the spanned areas for both initialization and optimization are shown. It is possible to see that the optimization has reduced the spanned area, since most of the FM^2 small oscillations have been reduced. Finally, Figure 11 shows the evolution of the minimum clearances and the velocities profile. Clearances are improved in those places in which the initialization had collisions. In some points of the path clearances are decreased (always above the safety margin, depicted by a dashed line) in order to reduce path length. This causes the speed profile to decrease in those places as well.

A total of 47 missions have been analyzed between the 2 buildings: 35 missions in the TB (7 in level B1, 14 in level L1, and 14 in level L2) and 12 in the HCB (7 in level L1 and 5 in level B2). Different metrics are used for both initialization and optimization: path length, swept area, total vehicle rotation angle, estimated execution time and minimum clearance along the path. The results of these metrics, except clearance, are shown in Figure 12.

The optimization procedure has reduced the path length in all cases. Also the swept area has been optimized in all the missions. The total steered angle by the vehicle is also significantly reduced since most of the oscillations created by the FM^2 initialization have been eliminated. However, it is increased in 3 missions in the TB and in all missions in the HCB/L1. These cases corresponds to sharp curves in confined spaces in which the only option for the vehicle is to get a better angle by first turning to the opposite side. As a counterpart, the estimated path execution time has been increased in all cases.

Smoothness require a more careful analysis. The smoothness metric employed creates triangles formed by consecutive path segments and compute

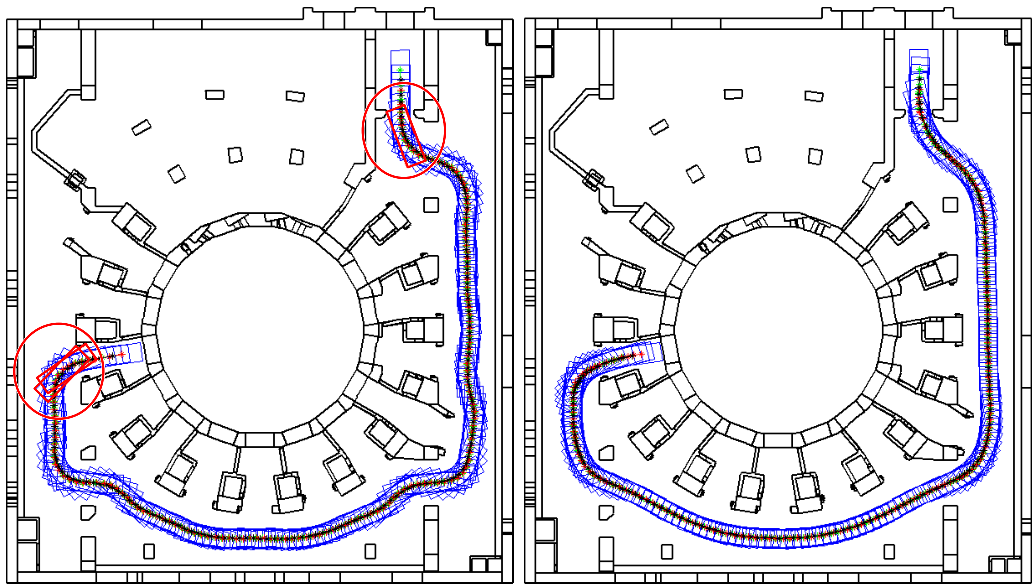

Figure 9: The path evaluation from the lift to the port 10 in level L1 of TB: the results from the initialization step (left) and from the optimization step (right).

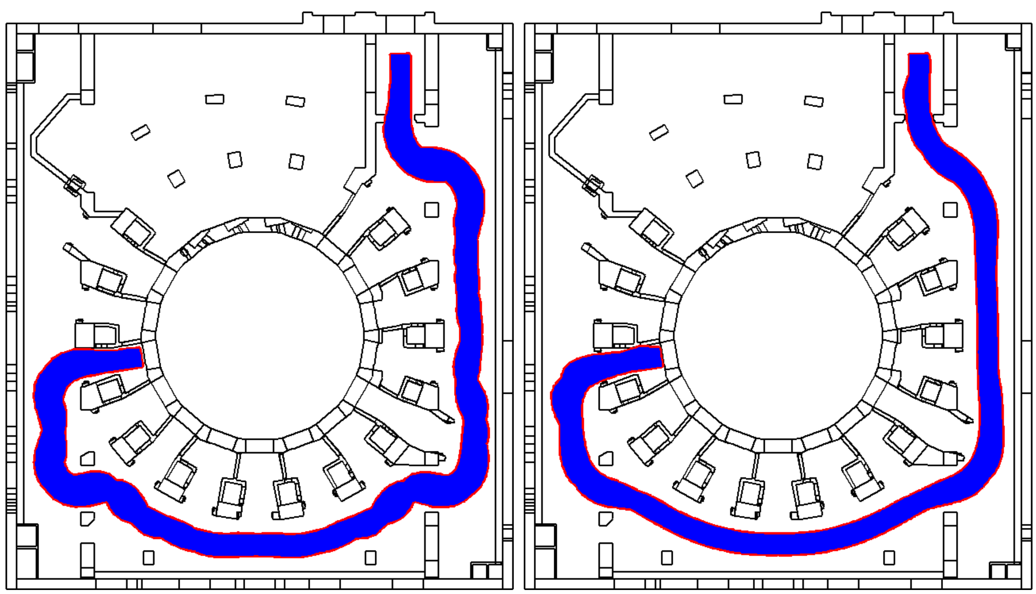

Figure 10: The spanned areas along the initialized path (left) and along the optimized path (right), from the lift to the port 10 in level L1 of TB.

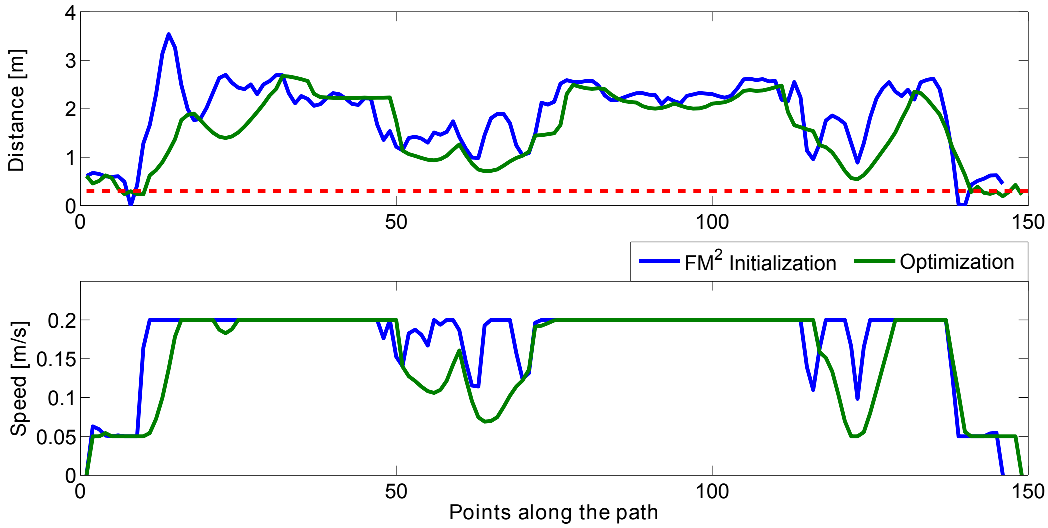

Figure 11: The minimum distance between the vehicle and the closest obstacles (top) and the speed of the vehicle (bottom) along the optimized path, from the lift to the port 10 in level L1 of TB.

the angle between those segments using the Pythagoras' theorem. Then, its conjugate angle is normalized by the path segment. Finally, all the normalized angles along the path are added. The higher this value is, the less smooth the path is. Minimum value is 0 for a straight line and there is no maximum value. Therefore, it is important that both initial and optimized paths have the same number of points. Otherwise, the comparison would not be fair. This metric is formally defined in equation 12:

$$smoothness = \sum_{i=2}^{n-1} \left(\frac{2\left(\pi - \arccos\left(\frac{a_i^2+b_i^2-c_i^2}{2a_ib_i}\right)\right)}{a_i + b_i} \right)^2 \tag{12}$$

where $a_i = dist(s_{i-2}, s_{i-1})$, $b_i = dist(s_{i-1}, s_i)$, $c_i = dist(s_{i-2}, s_i)$, s_i is the i^{th} state along the path, and $dist(s_i, s_j)$ gives the distance (Euclidean in this case) between states i and j.

Smoothness results are shown in Figure 13. Smoothness is worsened in all cases. However, this is not an actual negative result. Note that the smoothness values for the initial FM^2 paths is low (highly smooth). However, these paths present collisions. Therefore, the optimization procedure decreases smoothness only as much as required in order to satisfy the clear-

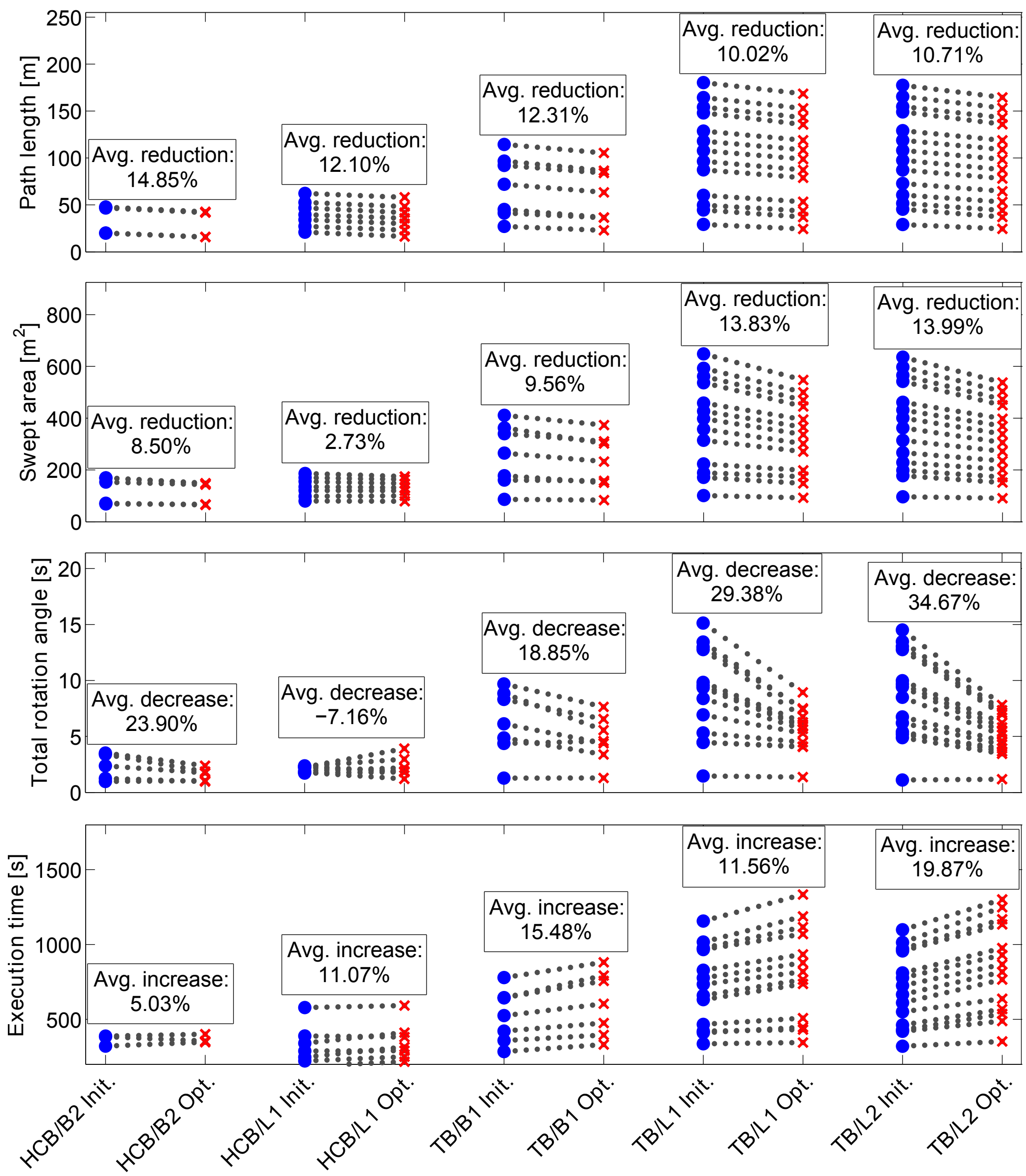

Figure 12: Metrics comparison for different levels in TB and HCB: initialization (Init.) versus optimization (Opt.).

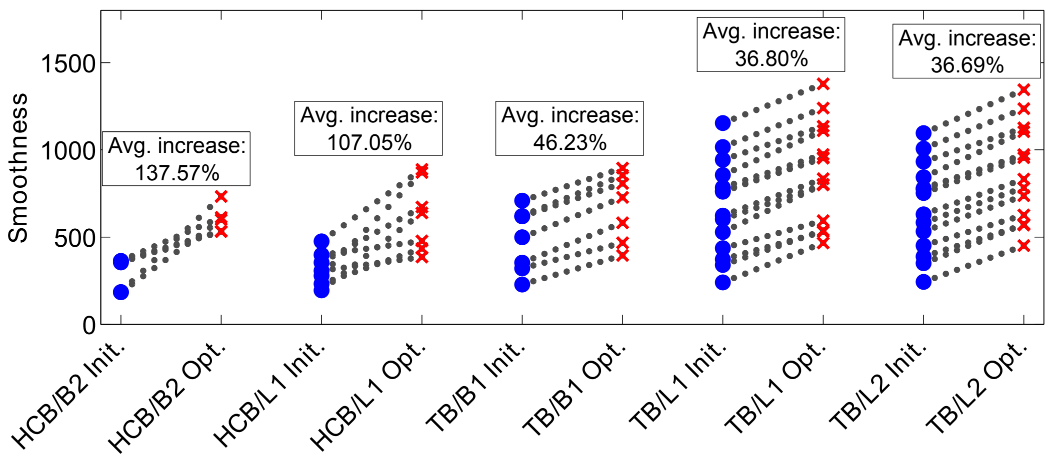

Figure 13: Smoothness metric comparison for different levels in TB and HCB: initialization (Init.) versus optimization (Opt.).

ance requirements.

The optimization provokes a redistribution of the minimum clearance along the trajectories which have to be carefully analyzed. Table 1 shows that the percentage of points with clashes have been reduced to 0 in all TB levels. However, the amount of points below the minimum clearance (0.3m) have increased. In TB there are two critical places in which it is not possible to accomplish this restriction: the lift exit and the docks gate. Therefore, the optimization *sacrifices* points in the surroundings of those critical places in order to avoid clashes by bringing them closer to the obstacles. This allows paths to have longer but smoother curves. The FM^2 method provides paths close to the optimal in terms of obstacle clearance. However, the optimization, in order to reduce oscillations and path length, decreases the clearance also in some of the points which are already far from obstacles. Thus, the percentage of points with clearance higher than a meter decreases while the group between 0.5-1m increases.

Figure 14 illustrates how close the path returned by the FM^2 is to the final solution and how fast the optimization is. As described in Section 1, a path is a set of 2D Cartesian points. During the optimization, each point *moves* as a result of the elastic and repulsive forces. Figure 14 presents, along the z-axis, the distance between each point and its final position, along the iterations represented in the x-axis. The y-axis represents the sequence of the points of the path. At iteration 30, the distance at all points are zero,

Table 1: Clearance distributions for the initialization (X_i) and optimized (X_o) trajectories.

Map / Level		Clash	$(0, 0.3)$m	$[0.3, 0.5)$m	$[0.5, 1)$m	$[1, \infty)$m
HCB	$B2_i$	0 %	5.48 %	2.74 %	44.52 %	47.26 %
	$B2_o$	0 %	2.91 %	5.81 %	42.44 %	48.84 %
	$L1_i$	0.82 %	2.45 %	3.67 %	47.35 %	45.7 %
	$L1_o$	0 %	3.69 %	4.43 %	43.17 %	50.21 %
TB	$B1_i$	1.14 %	3.42 %	3.42 %	23.29 %	68.72 %
	$B1_o$	0 %	4.26 %	9.36 %	36.17 %	50.21 %
	$L1_i$	1.49 %	1.8 %	2.82 %	14.58 %	79.31 %
	$L1_o$	0 %	6.6 %	4.83 %	24 %	64.57 %
	$L2_i$	0.66 %	0.88 %	2.86 %	19.03 %	76.57 %
	$L2_o$	0 %	2.75 %	4.48 %	36.3 %	56.47 %

since the optimized path was achieved and the points do not move anymore (or the oscillations are not perceptible, e.g., below a small threshold value). Figure 14 presents large distance values in the points that correspond to the areas of the scenario with more clearance. In the places where the scenario is very narrow, the points are stuck along all the iterations.

Figure 15 presents all the missions to the level L1 of TB. Figure 15 - left shows the paths resulted form the FM^2 algorithm applied to all ports. The paths are close to the optimized solutions, but with some clashes identified by circles. Figure 15 - right shows all the optimized paths. In some situations, as in the mission to port 11, the clearance of the path returned by the FM^2 in the vicinity of the pillars is greater when compared to the optimized path to the same port. However, the optimized path has no collision in the entrance to the same port while satisfying the clearance constrains. Figure 16 includes all the trajectories studied for the HCB. In this case the optimization is not essential in most cases since there are not clashes in most of the trajectories. However, optimization is applied in order to guarantee that the requirements are accomplished. Figure 16 - right shows the effect the optimization algorithm has in very cluttered scenarios: before entering the parking place, the cask has to get away from the wall in order to obtain a better angle to enter. As result, the total rotation angle is incremented.

Finally, an interesting point is raised. Different scenarios require different

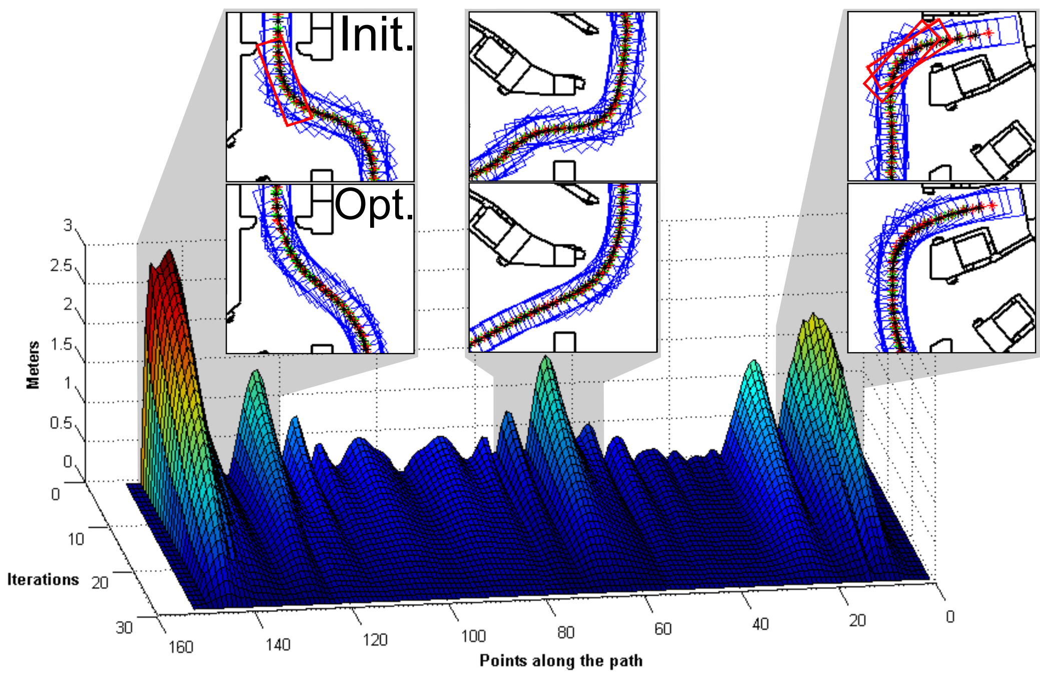

Figure 14: Evaluation of the distances between each point of the path along the iterations and its final value in the optimized path.

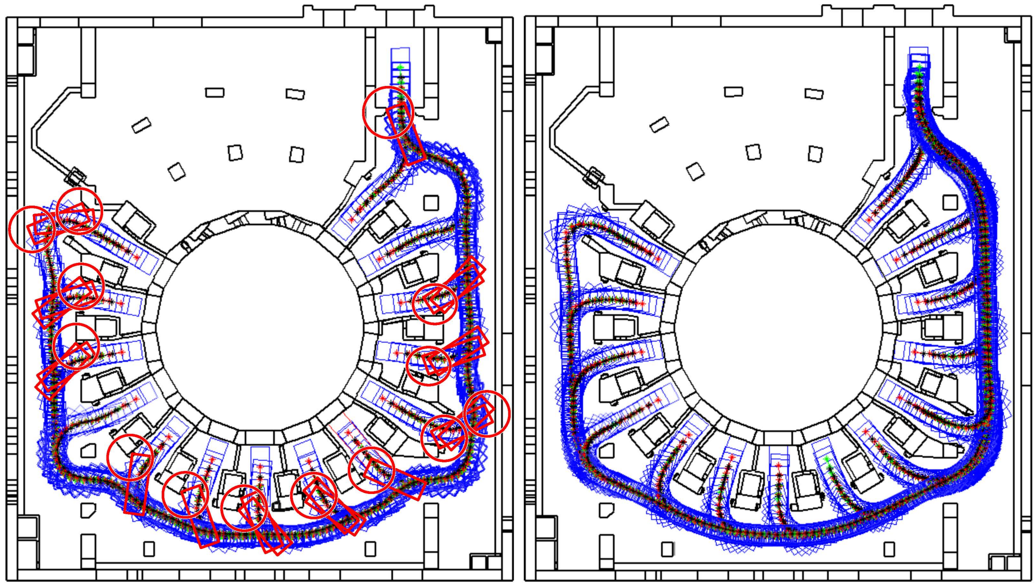

Figure 15: The path evaluation from the lift to all ports in level L1 of TB: the results from the initialization step (left) and from the optimization step (right).

FM^2 velocities maps. However, for small modifications in the scenario it is not necessary to recompute the velocities map, since the optimization procedure will successfully adapt the path. For instance, Figure 17 - left shows an optimized path between the lift and a parking place where the initial map and the respective velocity map did not considered the parked vehicle. Running again the optimization algorithm, the new optimized path is still smooth, but without clashes, as illustrated in Figure 17 - right.

The parameters k_e and k_r play an important rule for tuning the final trajectory in terms of shortness and smoothness. The k_e regulates the elastic path behavior. Higher values increase the path shortness approaching the path points connectivity. Lower values allow to increase path flexibility to obstacle-repulsive deformation. Outsized values either make the deformation process unstable or compromise path clearance. The k_r controls the repulsion behavior by determining the preponderance of the repulsive forces from obstacles. Gain increase allows to improve path clearance. Outsized values conflict with path smoothness and connectivity. These parameters were largely tested in several scenarios with similar dimensions and layout of ITER and the best results, as the ones depicted in the previous figures, were achieved with values of k_e and k_r between $[0.3; 0.4]$ and $[0.05; 0.01]$,

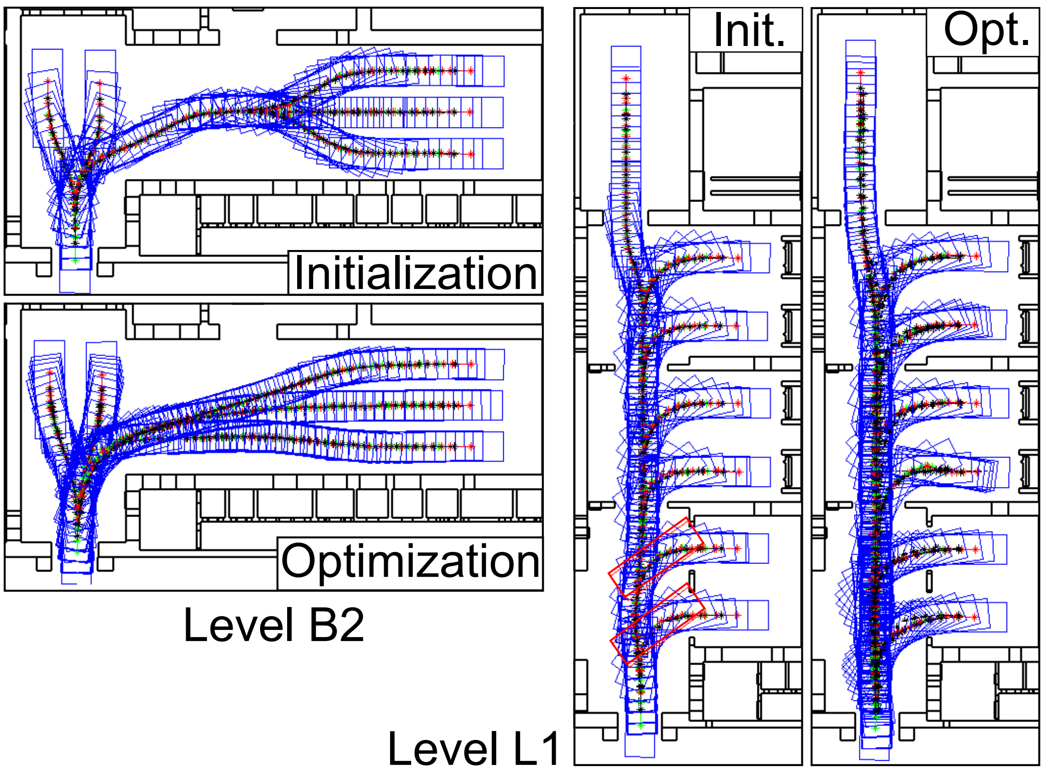

Figure 16: Path initialization and optimization from the lift to main parking places in levels L1 and B2 of HCB.

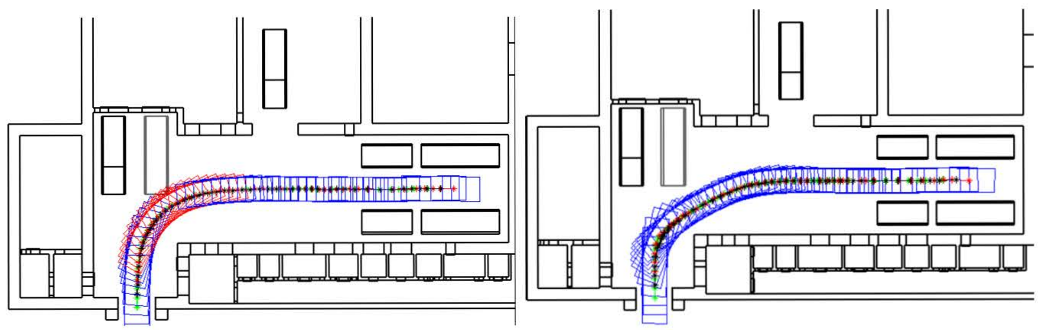

Figure 17: Path for a parking place in level B2 of HCB, in collision with a temporary vehicle (left), and the re-optimization of the path without the need of the initialization step (right).

respectively.

Lastly, the proposed framework allows to include trajectories with maneuvers [10], as shown in Figure 18. So far, the maneuver poses have to be manually defined and they will not be modified by the optimization procedure. This allows to increase clearance, find feasible paths where it was not possible before, and to accomplish restrictions about the orientation of the final poses within the ports.

6. Conclusions

This paper presented a summary of an algorithm to optimize trajectories in terms of clearance, smoothness and execution time. The algorithm has three steps: the initialization based on the FM^2, the path optimization using rigid body dynamics and the trajectory evaluation, where a velocity profile is created attending the clearance and maximum/minimum velocities and accelerations. The inputs of the 2D path planning algorithm are: the vehicle dimensions, the map of the environment and the initial and final goals of each mission. The path initialization is fast, robust and close to the final solution. Comparing with other previous approaches developed and tested by the same authors, using CDT [8], or Rapidly exploring Random Tree (RRT) [31], or only FM^2 [10], the algorithm proposed in this paper also provides trajectories with no clashes whenever possible, but with better performance in terms of computation time, smoothness and safety.

The algorithm was extensively tested using the maps of the ITER test facility: a structured, but complex and cluttered scenario. More experiments

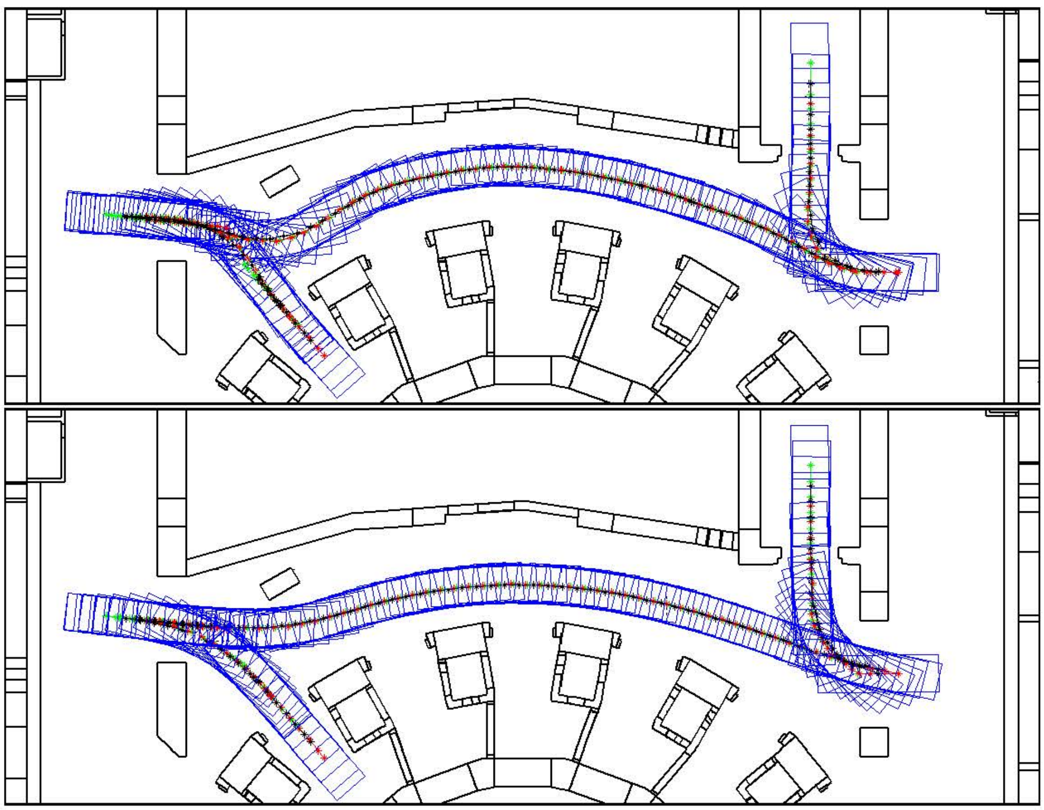

Figure 18: Example of a double maneuver in level B1 of TB, port 7: initialization (top) and optimization (bottom).

were done in the reactor building, since it is the core of the ITER. Results show how robust and flexible outputs are. The algorithm is also applicable to other environments, such as warehouses and using other vehicle kinematic configurations.

The future work focuses on extending the algorithm to a free roaming level (wheels do not follow the same path) avoiding the inclusion of maneuvers.

Acknowledgment

This work was supported by the TECHNOFUSION R&D program funded by the Community of Madrid, project DPI2010-17772 funded by the Spanish Ministry of Science. IST activities received financial support from "Fundação para a Ciência e Tecnologia" through project Pest-OE/SADG/LA0010/2013. The views and opinions expressed herein do not necessarily reflect those of the European Commission.

The authors want to gratefully acknowledge the comments on this work by David Álvarez and the reviewers and the audience of the ICRA 2013 conference. Also, the authors acknowledge the valuable contribution of future comments by the reviewers to improve both the research and the paper.

References

[1] A. Tesini, J. Palmer, The ITER remote maintenance system, Fusion Eng. Des. 83 (2008) 7–9.

[2] I. Ribeiro, P. Lima, P. Aparício, R. Ferreira, Conceptual Study on Flexible Guidance and Navigation for ITER Remote Handling Transport Casks, in: Proc. 17th IEEE/NPSS Symp. Fusion Eng., 1997, pp. 969–972.

[3] I. Ribeiro, P. Lima, P. Aparício, R. Ferreira, Active Docking of a Transport Cask Vehicle Subject to 6 Degrees of Freedom Misalignments, in: Proc. 17th IEEE/NPSS Symp. Fusion Eng., 1997, pp. 973–976.

[4] G. Dudek, M. Jenkin, Computational Principles of Mobile Robotics, Cambridge University Press (2 edition) (2010).

[5] B. R. Sarker, S. S. Gurav, Route planning for automated guided vehicles in a manufacturing facility, Int. J. Prod. Res. 43 (2005) 4659–4683.

[6] H. Martínez-Barberá, D. Herrero-Pérez, Autonomous navigation of an automated guided vehicle in industrial environments, Robot. Cim-Int. Manuf. 26 (2010) 296–311.

[7] B. Trebilcock, Automatic guided vehicle basics, Mod. Mater. Handl. 62 (2007) 46–50.

[8] D. Fonte, F. Valente, A. Vale, I. Ribeiro, A Motion Planning Methodology for Rhombic-like Vehicles for ITER Remote Handling Operations, Proc. 7th IFAC Symp. Intell. Autonomous Vehicles (2011) 106–111.

[9] F. Valente, A. Vale, D. Fonte, I. Ribeiro, Optimized Trajectories of the Transfer Cask System in ITER, Fusion Eng. Des. 86 (2011) 1967–1970.

[10] J. Gómez, A. Vale, F. Valente, J. Ferreira, S. Garrido, L. Moreno, Fast Marching in motion planning for rhombic like vehicles operating in ITER, Proc. IEEE Int. Conf. Rob. Aut. (2013) 5513–5518.

[11] S. Garrido, L. Moreno, M. Abderrahim, D. Blanco, FM2: A Real-time Sensor-based Feedback Controller for Mobile Robots, Int. J. Robot. Autom. 24 (2009) 3169–3192.

[12] C. Petres, Y. Pailhas, P. Patron, Y. Petillot, J. Evans, D. Lane, Planning for Autonomous Underwater Vehicles, IEEE Trans. Robot. 32 (2007) 331–341.

[13] Q. H. Do, S. Mita, H. T. Niknejad, L. Han, Dynamic and safe path planning based on support vector machine among multi moving obstacles for autonomous vehicles, IEICE Transactions Inf. Syst. 96-D (2013) 314–328.

[14] Q. H. Do, S. Mita, K. Yoneda, Narrow passage path planning using fast marching method and support vector machine, in: Proc. Vehicles Symposium, 2014, pp. 630–635.

[15] R. Takei, R. Tsai, Optimal trajectories of curvature constrained motion in the hamilton-jacobi formulation, J. Scientific Computing 54 (2013) 622–644.

[16] S. Garrido, L. Moreno, D. Blanco, F. Martin, Smooth path planning for non-holonomic robots using fast marching, in: Proc. Int. Conf. Mechatronics, 2009, pp. 1–6.

[17] J. Ferreira, A. Vale, I. Ribeiro, Localization of Cask and Plug Remote Handling System in ITER using multiple Video Cameras, Fusion Eng. Des. 88 (2013) 1992–1996.

[18] D. Locke, C.-G. Gutiéerrez, C. Damiani, J.-P. Friconneau, J.-P. Martins, Progress in the conceptual design of the ITER cask and plug remote handling system, Fusion Eng. Des. 89 (2014) 2419–2424.

[19] D. Wang, F. Qi, Trajectory planning for a four-wheel-steering vehicle, in: Proceedings of the IEEE International Conference on Robotics and Automation, volume 4, 2001, pp. 3320–3325.

[20] A. Vale, D. Fonte, F. Valente, I. Ribeiro, Trajectory optimization for autonomous mobile robots in ITER, Robotics and Autonomous Systems 62 (2014) 871–888.

[21] J. V. Gómez, Advanced Applications of the Fast Marching Square Planning Method, Master's thesis, Carlos III University of Madrid, 2012.

[22] A. Valero-Gómez, J. V. Gómez, S. Garrido, L. Moreno, The Path to Efficiency: Fast Marching Method for Safer, More Efficient Mobile Robot Trajectories, IEEE Robot. Autom. Mag. 20 (2013) 111–120.

[23] J. A. Sethian, A Fast Marching Level Set Method for Monotonically Advancing Fronts, Proc. Natl. Acad. Sci. 93 (1996) 1591–1595.

[24] S. Osher, J. Sethian, Fronts Propagating With Curvature-dependent Speed - Algorithms Based On Hamilton-Jacobi Formulations, J. Comput. Phys. 79 (1988) 12–49.

[25] S. Quinlan, O. Khatib, Elastic bands: connecting path planning and control, in: Proc. IEEE Int. Conf. Robotics and Automation, volume 2, 1993, pp. 802–807.

[26] A. Vale, D. Fonte, F. Valente, J. Ferreira, I. Ribeiro, C. Gonzalez, Flexible path optimization for the Cask and Plug Remote Handling System in ITER, Fusion Eng. Des. 88 (2013) 1900–1903.

[27] M. Kass, A. Witkin, D. Terzopoulos, Snakes: Active contour models, Int. J. Comp. Vis. 1 (1988) 321–331.

[28] F. Kang, S. Zhong-Ci, Mathematical Theory of Elastic Structures, Springer New York, 1981.

[29] F. P. Beer, E. R. Johnston, J. T. DeWolf, Mech. Mater., McGraw Hill, 2002.

[30] P. Ruibanys, C. Reig, E. Gazeau, J. Marmie, N. Etchegoin, Definition, Development and Operation of a Comprehensive Virtual Model of the ITER Buildings, ATS and TCS, in: 26th Symp. Fusion Technology, 2010.

[31] D. Fonte, F. Valente, A. Vale, I. Ribeiro, Path Optimization of Rhombic-Like Vehicles: An Approach Based on Rigid Body Dynamic., in: Proceedings of the 15th IEEE International Conference on Advanced Robotics, 2011, pp. 106–111.