Optimal Velocity Planning of Wheeled Mobile Robots on Specific Paths in Static and Dynamic Environments

María Prado
Dept. Mechanical Engineering, University of Malaga
Spain

1. Introduction

The control system of a mobile robot generally comprises two different modules: a trajectory planner and a trajectory tracking controller, although some researchers have proposed algorithms that integrate both tasks.

To completely solve the trajectory planning problem is to define an open-loop path and its velocity profile from an initial to a final posture, while avoiding any potential obstacles.

In time-optimal planning of a wheeled mobile robot (WMR), the problem is solved by defining control inputs for the wheels that minimize navigation time from the origin to the target posture. This goal implies two tasks, which can be carried out simultaneously or sequentially: path-planning (PP), which involves the computation of the shortest feasible path; and velocity-planning (VP), which involves the computation of the fastest feasible velocity profile for the entire domain of the path.

Several approaches have been developed to perform both tasks. The most widely used approaches are free configuration-time space based methods, (Reinstein & Pin, 1994), but these algorithms are computationally expensive, even when one is only dealing with PP or VP separately. To reduce the computational cost, researchers have recently published methods which do not require computing the C-space obstacles (Wang et al., 2004), as well as methods that search for a probabilistic road map (LaValle & Kuffner, 2001). Some other approaches that use intelligent computing-based methods have also been presented, such as those that use artificial potential fields-based methods (Liu & Wu, 2001), fuzzy logic (Takeshi, 1994), genetic algorithms (Nerchaou, 1998) or neural networks (Zalama et al., 1995).

In order to find an optimal and feasible solution for the two problems, mechanical, kinematic and dynamic characteristics of the WMR that limit its motion must be taken into account, as well as other environmental, task-related and operational issues. These constraints can be summarized by upper boundary functions of the velocity, acceleration and deceleration of the WMR. In general, the functions are not constant, nor are they even continuous. They are therefore nonintegrable constraints, and the time optimal planning is a nonholonomic problem.

A significant number of nonholonomic constraints, which include not only mechanical and kinematic but also dynamic characteristics of the WMR, are difficult to deal with when PP and VP are approached simultaneously. The vast majority of existing algorithms consider

only kinematic constraints or some dynamic conditions derived from simplified models of the WMR and/or its environment. But the resulting trajectory may be unexecutable, or tracked by the robot with high spatial and temporal errors. However, when PP and VP are approached sequentially, the difficulty of both problems is significantly reduced. Such approaches make it possible to include more complex constraints for the WMR's velocity and acceleration, especially with regards to its kinematic and dynamic characteristics.

To our knowledge, the first references addressing VP with kinematic and dynamic constraints for WMR is (O'Dunlaing, 1987). This paper, like a number of other algorithms to solve the VP stage, is based on constant maximum values for robot velocity and acceleration, set to arbitrary constants which are unrelated to the mechanical characteristics of the system. More recent works seek to find more efficient bounds for these operating variables, but never in a global way and always based on simplified dynamic robot models. (Weiguo et al., 1999) propose a velocity profile planner for WMRs on flat and homogeneous terrains, where velocity and acceleration are limited only by the outer motor torques and by the absolute slippage of the vehicle on the ground. (Choi & Kim, 2001) develop another planner where velocity and acceleration are constrained by dynamic characteristics related to the performance of the robot's electric motors and its battery's power. (Guarino Lo Bianco & Romano, 2005) present a VP algorithm for specific paths that generate a continuous velocity and acceleration profile, both into safety regions limited by upper boundary functions not described in the paper. The method involves an optimization procedure that has a significant computational cost.

Some other limitations have been studied, mainly within the framework of projects for planetary exploration. (Shiller, 1999) deals with some dynamic constraints: sliding restrictions, understood as the avoidance of absolute vehicle slippage, tip-over and loss of wheel-ground contact constraints, which are important issues when dealing with irregular outdoor terrains. The author works with a very simplified robot model, neglecting sideslip and assuming pure rolling, so wheel deformations and microslippages which can cause important tracking errors are not quantified. (Cheriff, 1999) also proposes a set of kinematic and dynamic constraints over the robot's path, dealing specifically with 3D irregular and non-homogeneous grounds. The resulting trajectory planner directly incorporates a complete dynamic WMR model, considering non-linear motions and specifically accounting for wheel-ground interactions, which makes it necessary to run complex algorithms that significantly increase computational cost.

(Lepetic et al., 2003) present a VP method that considers dynamic constraints by bounding the acceleration by the maximum wheel-ground adherence capacity. This maximum is computed as a function of a constant friction coefficient for every posture and of the weight borne by the wheel. Load transfer due to lateral forces is considered to calculate the weight on the wheel, but only as a constant maximum value, derived from a simplified model of the WMR, that reduces the lateral maximum acceleration to the same value for every posture. The VP method published by (Krishna et al., 2006) builds a trajectory continuous in space and velocity, which incorporates environment and sensory constraints by setting a maximum velocity for the entire path of the robot that is decreased when an obstacle is detected within its visibility circle. The velocity constraint is computed as a function of the position and velocity of the obstacle and of a maximum acceleration or deceleration value of the WMR, established as constant values for every posture.

This chapter deals with time-optimal planning of WMRs when navigating on specific spatial paths, i.e., when the PP is previously concluded. First, the computation of the upper

boundary functions of its velocity, acceleration and deceleration are described. Then a method for time-optimal planning is proposed, the main goals of which are:

- To fully exploit velocity, acceleration and deceleration constraints, avoiding the planning of velocities or accelerations that lead to dangerous motions.
- To plan a feasible trajectory, with continuous velocity and deceleration
- To bound the jerk of the WMR
- To be of low computational cost.

The method firstly deals with velocity planning in static environments and then presents an algorithm to modify the resulting trajectory to avoid moving obstacles. Special attention is paid to the efficiency of the second algorithm, an advantage which makes it highly useful for local and/or reactive control systems.

2. Problem definition

Problem 1: Given a WMR's path, computed to navigate in a static and known environment, plan the fastest, feasible and safe trajectory, considering the constraints imposed by the mechanical configuration, kinematics and dynamics of the robot and by environmental and task-related issues.

Problem 2: Modify the trajectory quickly and locally to avoid moving obstacles.

A generalized posture of a WMR, parameterizing by the path length, s, can be defined by the vector $q(s) = [X(s), Y(s), \theta(s), \delta(s)]^T$. $[X(s), Y(s)]$ is the position and $\theta(s)$ the orientation of the WMR's guide point on a global frame (Z coordinate is constant by assuming navigation is on flat ground). $\delta(s)$ is a function kinematically related to the curvature of the trajectory, $\kappa(s)$; specifically, it is a function of the steer angles of the wheels of WMRs with steering wheels or a function of the difference between the angular velocities at the traction wheels for WMRs with differential drive.

The path, P(s), can be defined by a continuous series of generalized postures from the initial posture, q_0, to the final one, q_f. Therefore, if S is the total length of the path:

$$P(s) = \{q(s)\} : [0, S] \quad \rightarrow \quad \Re^4 ; P(0) = q_0 \vee P(S) = q_f \tag{1}$$

To transform P(s) into a trajectory, a velocity function must be generated for the entire path domain. It must be defined in positive real space (if the WMR is only required to move forward, as is the usual case) and planned to make the robot start from a standstill and arrive at the final posture also with null velocity. That is:

$$V(s) = \{v(s)\} : [0, S] \quad \rightarrow \quad \Re^+ ; V(0) = 0 \vee V(S) = 0 \tag{2}$$

Additional conditions are strongly required of V(s) to obtain a feasible trajectory:

1. Continuity, since the kinematics of WMR make it impossible to develop other types of maneuvers.
2. Confinement into a safety region of the space-velocity plane (s×v), upper limited by a boundary function of the velocity, $V_{Lim}(s)$.
3. Confinement of its first derivative with respect to time, acceleration or deceleration, into a safety region of the space-acceleration plane (s×a), upper limited by a boundary function of the acceleration, $a_{Lim}(s)$, and lower limited by the negative value of a boundary function of deceleration $d_{Lim}(s)$.

4. Continuity of acceleration or deceleration: this condition ensures that the jerk of the robot, the second derivative of its velocity, is finite, so that the robot's movements are smooth. High jerk is not recommended for WMRs for a number of reasons: it causes the robot to shake significantly and thus complicates on-board tasks; it makes tracking control more difficult, since wheel microslippage increases and wheel behavior becomes less linear (Wong, 2001); and it increases the error of on-board sensor systems.

5. Additionally, low computational cost is beneficial for the generation of the velocity profile. This goal is especially pursued when solving problem 2, for the purpose of possibly incorporating the algorithm into local controls or reactive planners, to adjust the trajectory in the presence of new unexpected obstacles that appear in the visibility area of the robot's sensorial systems (Krishna et al., 2006).

3. Velocity constraints

This section deals with constructive characteristics, kinematic configuration and the dynamic behaviour of a WMR, as well as operational matters, in order to identify the constraints that influence the maximum velocity of a WMR's guide point.

For all the constraints detailed in the following subsections, an upper boundary function of velocity, parametrized by s, can be generated. The function is built by assigning the lowest upper bound of all the velocity constraints to each posture:

$$V_{Lim}(s) = \min\{V_{Lim1}, V_{Lim2} \ldots \ldots V_{Limn}\} \quad / s \subset [0, S] \tag{3}$$

This chapter addresses the case of a WMR guided by steering wheels; in the case of WMRs with differential drive, the approach will be similar and therefore the constraints can easily be deduced under the same considerations.

3.1. Construction constraints

Thermal and mechanical characteristics of motors and batteries impose maximum rotational velocities on the tractive and steering servomotors, ω_{tm}^{max} and ω_{sm}^{max}, respectively (Choi & Kim 2001). Thus, if ξ_t is the reduction ratio of the drive-train and R the wheel's radius, the maximum linear velocity of driven wheels on the ground is:

$$v_{tw}^{max} = \xi_t \omega_{tm}^{max} R \tag{4}$$

Further, if ξ_s is the reduction ratio of the steering-train, the maximum velocity of variation of the steering angle, i.e. the maximum steering gain, is:

$$G_s^{max} = \xi_s \left| \omega_{sm}^{max} \right| \tag{5}$$

3.2. Kinematic constraints

With regards to kinematic linkages between the driven wheels and the guide point, if d_{tw}^{max} is the position vector on the ground of the most distant driven wheel with respect to the guide point, an upper bound for the WMR's velocity is given by:

$$V_{Lim1} = v_{tw}^{max} \frac{|1/\vec{\kappa}|}{\left|\frac{1}{\kappa} + \vec{d}_{tw}^{max}\right|} \tag{6}$$

On the other hand, by considering kinematic linkages between the steering wheels and the guide point, a second boundary function for the velocity is found as:

$$V_{Lim2} = \frac{\left|\frac{d\kappa}{dt}\right|}{\left|\frac{d\kappa}{ds}\right|} \tag{7}$$

The numerator must be calculated from a kinematic model of the robot, whereas the denominator can be directly computed from the known spatial path.

3.3. Dynamic constraints

A dynamic model of the robot is needed to generate boundary functions relating to its dynamic characteristics. Since this model is only used to fully define the VP algorithm, specifically when defining V_{Lim}, a_{Lim} and d_{Lim}, but not when computing the trajectory, it can be as complex as needed for successful results without increasing computational cost. One may therefore use a model which is not limited by its degrees of freedom, geometric non-linearities, integration tolerances, etc...

3.3.1. Maximum velocity to bound spatial error

Let the quadratic spatial error of a WMR be the square of the distance from the actual position of its guide point tracking a trajectory to the position planned by the PP, measured on the ground plane and parameterised by the normalised arc length, $\tilde{s}$, defined as the ratio of s to the total path length, S, i.e. $\tilde{s} = s/S.$

Let the actual tracked trajectory, which will involve a side-slip angle with a value that is generally non-zero, be expressed by a two dimensional function on a world reference frame as:

$$P_A(\tilde{s}) = \left[X_\beta, Y_\beta, \theta_\beta, \delta_\beta\right]: \quad [0,1] \to \mathfrak{R}^4 \tag{8}$$

Then, the quadratic spatial error can be calculated by:

$$E_s^2(\tilde{s}) = \sqrt{(X(\tilde{s}) - X_\beta(\tilde{s}))^2 + (Y(\tilde{s}) - Y_\beta(\tilde{s}))^2} \tag{9}$$

And the total quadratic spatial error is the integral of (9) over the entire path:

$$TE_s^2(\tilde{s}) = \int_0^1 E_s^2(\tilde{s})d\tilde{s} \tag{10}$$

If the planned path, P(s) in (1) is particularized for stationary manoeuvres, i.e. with constant velocity and curvature, the WMR's planned position in the same world reference frame can be expressed as a function of $\tilde{s}$ as:

$$[X(\tilde{s}), Y(\tilde{s})] = \left[\frac{\sin(2\pi\tilde{s})}{\kappa}, \frac{1}{\kappa}(1 - \cos(2\pi\tilde{s}))\right] \tag{11}$$

The actual tracked trajectory, with side-slip angle β_A and curvature κ_A, will generally differs from the planned trajectory and is given by (Prado et al., 2002):

$$\left[X_\beta(\tilde{s}), Y_\beta(\tilde{s}) \right] = \left[\frac{1}{\kappa_A} \left[\sin(\beta_A + 2\pi\tilde{s}) - \sin(\beta_A) \right], \frac{1}{\kappa_A} \left[\cos(\beta_A) - \cos(\beta_A + 2\pi\tilde{s}) \right] \right] \tag{12}$$

κ_A and β_A can be approximated by the trajectory obtained as simulation output of a sufficiently accurate dynamic model or by experimental results. If enough simulations or tests are performed to characterize the dynamics of the WMR in stationary trajectories, it is possible to fit κ_A and β_A to functions of the planned curvature and velocity. Therefore, by substituting (11) and (12) into (10) and considering the simulation or experimental results to compute κ_A and β_A, the total spatial error of the WMR when navigating a whole stationary cycle can be calculated as a function of the planned V and κ. Although it is known that in general the planned variables V and κ do not stay constant at adjacent postures, it is understood that they will experience smooth variations when smooth paths are planned. Therefore, the error at each posture will be approximated by the error in (10) computed as described in this section, i.e. considering neighbourhoods where the V and κ are kept constant.

Finally, TE_s^2 will be upper limited by a magnitude relative to the total area of the circle that defines the stationary planned trajectory. Therefore, if tol_s is the percentage of the admissible tolerance for the spatial error, the following constraint is imposed:

$$TE_s^2 \leq \frac{\pi}{\kappa^2} \left(1 - (tol_s)^2 \right) \tag{13}$$

When (13) is applied to (10), a velocity constraint V_{Lim3} for the WMR is obtained.

3.3.2. Maximum velocity to bound temporal error

When a WMR is navigating, it must do more than consider position error; temporal error can also be important if one wishes to fit or synchronise several objects. Let the temporal error of a WMR be the time gap between the actual time when the robot arrives at a posture, t_A, and the time when it is planned to arrive, t:

$$E_t = t_A - t \tag{14}$$

For a stationary trajectory of length S tracked with actual velocity V_A, this error is:

$$E_t = S \left(\frac{1}{V_A} - \frac{1}{V} \right) \tag{15}$$

V_A in a stationary trajectory can be approximated by the velocity obtained in experimental tests or simulations of a sufficiently accurate dynamic model of the WMR. As stated for the spatial error, such outcomes make it possible to express V_A as a function of the characteristics of V and κ.

The velocity planner fixes an upper bound for the temporal error associated to each posture, given by a value relative to the time that the path tracker estimates the robot will spend in the stationary trajectory, with relative tolerance t_t^{rel}. Then the following inequality must be satisfied:

$$E_t \leq tol_t \ t = tol_t \frac{S}{V} \tag{16}$$

By substituting (15) in (22), a new upper boundary function V_{Lim4} is generated as:

$$V_{Lim4} = (tol_t + 1)V_A \tag{17}$$

3.3.3. Tip-over limitation

Tip-over occurs when the robot's entire weight shifts to one side of the vehicle, and the other wheels are about to lose contact. Thus, the robot is at risk of tipping-over when its total weight is entirely borne by the outer wheel (Shiller, 1999). The extreme situation, depicted in Fig. 1 for positive κ, where h is the height of the centre of gravity (c.g.) of the robot and B_1 and B_2 are the lateral distances between the outer wheel and the c.g. for positive and negative κ, respectively (although generally $B_1=B_2$), yields a relation between the lateral force, F_y, and the vertical force, F_z, given by:

$$\begin{cases} F_y = F_z \dfrac{B_1}{h} \quad \text{......if } \kappa \geq 0 \\ F_y = F_z \dfrac{B_2}{h} \quad \text{......if } \kappa < 0 \end{cases} \tag{18}$$

By neglecting gyroscope torques, the lateral force, F_y, on flat grounds is simply the centrifugal force, while F_z is the robot's weight. Thus, if g is the gravity constant, equation (18) requires V to be lower than:

$$\begin{cases} V_{Lim5} = \sqrt{\dfrac{gB_1}{h}\dfrac{1}{\kappa}} \quad \text{......if } \kappa \geq 0 \\ V_{Lim6} = \sqrt{\dfrac{gB_2}{h}\dfrac{1}{(-\kappa)}} \quad \text{......if } \kappa < 0 \end{cases} \tag{19}$$

3.4. Operational constraints

The need to fit and synchronise the robot's motion with its environment, whether static or dynamic, makes operational constraints necessary.
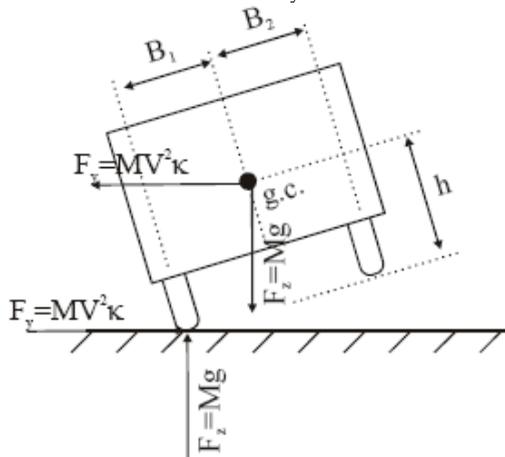
Fig. 1. Tip-over of the WMR.

3.4.1. Maximum velocity to prevent collisions

V is limited by a value that ensures the WMR will come to a complete stop at a distance greater than a safety distance from any obstacle, L_{safe}. Such a stop will be performed at the maximum deceleration, b^{max}, a constant calculated in section 4. Therefore, the distance run by the robot until it stops, is:

$$s = \frac{V^2}{2\, b^{max}} \tag{20}$$

When V_{obs} is the maximum estimated velocity of the obstacle towards the robot (0 in static environments), the distance covered by the object is:

$$s_{obs} = \frac{V_{obs}}{b^{max}} V \tag{21}$$

If D_{obs} is the distance from the robot guide point to the obstacle in completely known environments, or the radius of vision of the external sensor system in partially known environments (Krishna et al., 2006), in order to ensure that the robot maintains its safety distance, it must satisfy:

$$s + s_{obs} = D_{obs} - L_{safe} \tag{22}$$

By replacing (20) and (21) in (22), a new upper limit for the velocity is found as:

$$V_{Lim7} = \sqrt{(V_{obs})^2 + 2b^{max}(D_{obs} - L_{safe})} - V_{obs} \tag{23}$$

3.4.2. Maximum velocity to approach the target posture

In the same way, in order to ensure safe stopping at the target point of the path, another upper boundary function is given by:

$$V_{Lim8} = \sqrt{2\, C_2 (S-s)\, b^{max}} \tag{24}$$

Where C_2 is an arbitrary constant greater than 1, which reflects a security percentage for stopping, and S is the total path length.

3.4.3. Environmental constraints

A set of velocity constraints which are solely dependent on the robot's working environment can be defined as a function which assigns a maximum speed V_i to each portion of the path, with expressions such as:

$$V_{Lim9} = \begin{cases} V_1 & \text{if } 0 \le s \le s_1 \\ \vdots & \vdots \\ V_n & \text{if } s_n s \le s_{n+1} \end{cases} \tag{25}$$

4. Acceleration and deceleration constraints

The same constructive, kinematic, dynamic and environmental topics which were analysed for velocity are studied for acceleration and deceleration in this section. From all the constraints detailed in next subsections an upper boundary function of acceleration, a_{Lim}, and deceleration, d_{Lim}, can be generated as:

$$a_{Lim}(s) = \min\{a_{Lim1}, a_{Lim2} \ .... \ a_{Limn}\} \quad /s \subset [0, S] \tag{26}$$

$$d_{Lim}(s) = \min\{d_{Lim1}, d_{Lim2} \ .... \ d_{Limn}\} \quad /s \subset [0, S] \tag{27}$$

4.1. Constructive constraints

The maximum torques of tractive motors, T_{tm}, steering motors, T_{sm}, and the braking mechanism, T_{bm}, dictate the maximum torques achievable at the wheels. If η_t and η_s are the efficiency of the drive-train and of the steering-train, these values are given by:

$$T_{tw}\big|^{mot} = \frac{\eta_t}{\xi_t} T_{tm}; \qquad T_{sw}\big|^{mot} = \frac{\eta_s}{\xi_s}\frac{T_{sm}}{2}; \qquad T_{bw}\big|^{mot} = \frac{\eta_t}{\xi_t} T_{bm}\big|^{max} \tag{28}$$

4.2. Kinematic constraints

As occurs with velocity, the robot's kinematics would make its acceleration be a function of the acceleration of the wheels. But as is argued in section 4.4, this value is limited exclusively by the dynamic capabilities relative to the resisting loads.

4.3. Dynamic constraints

4.3.1. Wheel-ground adhesion constraint

In order to avoid slippage, the maximum effort that the wheel-ground contact can support in a direction j is limited by the wheel-ground friction coefficient,μ, as:

$$F_{wi}^j\Big|^{ad} = \mu_j F_{wi}^n \tag{29}$$

μ can be assumed to be constant for uniform rubber wheels if slippage does not occur and terrain characteristics are uniform (Wong, 2001). F_{wi}^n is the vertical load borne by the i-th wheel, which changes with: ground irregularities, transients for non-stationary manoeuvres, lateral load transference produced by the centrifugal force on curved paths and longitudinal load transference on accelerated or decelerated paths. The first two phenomena can be neglected, especially for navigation in industrial environments. Regarding the two dynamic load transfers, F_{wi}^n can be computed as a function of the static weight borne by the i-th wheel, $F_{wi}^n\big|_0$, as:

$$F_{wi}^n = F_{wi}^n\big|_0 + \frac{h}{\vec{L}_{wi}} Ma + \frac{h}{\vec{B}_{wi}} MV^2 \vec{\kappa} \tag{30}$$

Where $\vec{L}_{wi}$ is the vector of longitudinal position of the centre of the i-th wheel with respect to the c.g. of the WMR and $\vec{B}_{wi}$ is the vector of its lateral position.

The maximum lateral effort that can be borne by the steering wheel is computed by replacing in (29) the sum of (30) extended for those wheels:

$$F_y\Big|^{ad} = \mu_y \sum_{\substack{i=wheel \\ steering}} F_{wi}^n \tag{31}$$

Therefore, if d_c is the castor distance, i.e. the longitudinal distance from the rotation axis of the steering system to the wheel centre, the available steering torque limited by the wheel-ground adherence capacity is:

$$T_s \Big|^{ad} = \mu_y d_c \sum_{\substack{i=steering \\ wheel}} F_{wi}^n \tag{32}$$

Regarding driven and braking wheels, by replacing in (29) the sum of (30) for all those wheels, the longitudinal driven and braking efforts limited by the wheel-ground adherence capacity are given by the following equations, respectively:

$$F_t \Big|^{ad} = \mu_x \sum_{\substack{i=driven \\ wheel}} F_{wi}^n \;\; ; \qquad F_b \Big|^{ad} = \mu_x \sum_{\substack{i=braking \\ wheel}} F_{wi}^n \tag{33}$$

4.3.2. Maximum tractive and braking efforts

The maximum acceleration is that which is reached applying the maximum available effort when rolling, grade and aerodynamic resistances are overcome. For WMRs, aerodynamic resistance is habitually neglected, because of their low navigation velocity. Therefore, the maximum acceleration when negotiating a grade j, and with rolling resistance coefficient f_r, is:

$$a^{max} = g \left(\frac{F_{long} \Big|^{max}}{Mg} - f_r \cos[a\tan(j/100)] - \sin[a\tan(j/100)] \right) \tag{34}$$

$F_{long}|^{max}$ is the maximum longitudinal effort, limited either by the motors or by the wheel-ground adherence capacity. Therefore, by introducing the power capacities computed in (28) and the maximum adhesion of (33) into (34), the following upper boundary functions are defined:

$$a_{Lim1} = \frac{1}{M} \cdot \min \left\{ 2 \frac{T_{wt} \Big|^{mot}}{R}, \quad F_t \Big|^{ad} \right\} - g \left(f_r \cos[a\tan(j/100)] + \sin[a\tan(j/100)] \right) \tag{35}$$

$$d_{Lim1} = \frac{1}{M} \cdot \min \left\{ 2 \frac{T_{bw} \Big|^{mot}}{R}, \quad F_b \Big|^{ad} \right\} - g \left(f_r \cos[a\tan(j/100)] + \sin[a\tan(j/100)] \right) \tag{36}$$

These are constant functions as long as f_r can be considered constant, which occurs when the operating variables stay within a small interval, which is the most common situation for WMR motion in industrial environments (Wong, 2001).

4.3.3. Maximum steering efforts

The maximum acceleration available for the steering angle, δ, can be calculated as:

$$\left\| \frac{d^2\delta}{dt^2} \right\|^{max} = \frac{2 T_{steer} \Big|^{max} - \sum_{\substack{i=steering \\ wheel}} T_{wi}^{res}}{I_s} \tag{37}$$

$T_{steer}|^{max}$ is the maximum steering torque at the wheel, limited either by power of the steering system, in (28) or by adhesion, in (32); I_s is the mass moment of inertia of the whole steering system; and T_{wi}^{res} is the self-aligning torques of the i-th wheel.

Looking at the left-hand side of (37), it can be expressed by:

$$\frac{d^2\delta}{dt^2} = \frac{d^2s}{d^2t} \frac{d^2\kappa}{ds^2} \frac{d^2\delta}{d\kappa^2} \tag{38}$$

Thus, the acceleration of δ depends on three terms: the acceleration of the trajectory; the spatial acceleration of curvature, a characteristic directly derived from the spatial path; and

on the last term, which is a characteristic of the robot that can be approximated from its kinematic model.

By replacing (38) and in (37) and by isolating the acceleration, a new boundary function is given by:

$$a_{Lim2} = \frac{\min\left\{2T_w^s \mid^{mot}, T^s \mid^{ad}\right\} - \frac{MV^2|\kappa|}{2}d_c}{I_s} \frac{1}{\left|d^2\delta / d\kappa^2\right|} \frac{1}{\left|d^2\kappa / ds^2\right|} \tag{39}$$

This upper bound depends on the velocity, which would take the VP to an iteration loop. In order to avoid this costly procedure, V is substituted by the single-valued upper boundary function of velocity, $V_{LIM}(s)$, defined in section 3, making the acceleration boundary even more restrictive.

5. Velocity Planning in Static Environments

5.1. Path Segmentation

In order to reduce its complexity, the proposed method divides the path into segments where all of the velocity constraints are continuous and monotone or constant, i.e. where the velocity of the path can be continuous and monotone or constant when it matches its upper boundary function. The velocity profile for the entire path is generated sequentially from the first segment to the last one according to the method describe later on in this section.

A set of p+2 segmentation points, $P_s=\{0, ^1ss, ... ^pss, S\}$, sorted in increasing order, divides P(s) into p+1 segments, $SG=\{^1Sg, ... ^{p+1}Sg\}$, where ^iSg is the subset of P(s) corresponding to the domain $s \subset [^{i-1}ss, ^iss]$.

P_s, comprises: the initial point of the path, s=0; its final point, s=S; and every point that satisfies at least one of the following conditions:

- To be a local extremum of the velocity boundary function: consequently V_{Lim} is monotone over ^iSg, so that V(s) can also be monotone when it matches V_{Lim}.
- To be a point of discontinuity of V_{Lim}: at the first stage, the VP algorithm proposed herein will only deal with the velocity limits at the end points of the segment. Later it will check if $V_{Lim}(s)$ is exceeded at any intermediate point and, in such a case, a time-consuming iterative process will be carried out. Since any discontinuity of V_{Lim} increases the risk of failing to meet the velocity constraint, they are shifted to the ends of the segments by selecting them as segmentation points.

5.2. VP of a segment of the path in static environments

The piece of V(s) for ^iSg is generated from the time-space (t×s) function $^i\sigma(t)$, which computes the path length navigated as:

$$s = {}^i\sigma(t) \quad ; s \subset \left[^{i-1}ss, {}^iss\right] \tag{40}$$

Thus, the velocity profile for the segment is:

$$^iV(s) = \frac{d}{dt}\left({}^i\sigma\left({}^i\sigma^{-1}(s)\right)\right) \quad ; s \subset \left[^{i-1}ss, {}^iss\right] \tag{41}$$

$^i\sigma(t)$ must start at the initial position of ^iSg and arrive at its final position. If the origin of time is shifted to the first point of ^iSg, without loss of generality, and if ^it is the time taken by the WMR to navigate ^iSg, the position boundary conditions are:

$$^i\sigma(0) = {}^{i-1}ss$$
$$^i\sigma\left({}^i t\right) = {}^i ss \qquad (42)$$

The first derivative of $^i\sigma(t)$ with respect to time, $^i\sigma'(t)$, must also satisfy the velocity boundary conditions, which are given at the edge points of the segment by:

$$^i\sigma'(0) = {}^{i-1}v = \begin{cases} 0 & \text{if } i = 1 \\ {}^{i-1}\sigma'\left({}^{i-1}t\right) & \text{if } i \neq 1 \end{cases}$$

$$^i\sigma'\left({}^i t\right) = {}^i v = V_{Lim}\left({}^i s\right) \qquad (43)$$

The second equation of (43) sets the velocity at the end of ^iSg to the maximum value permitted by the V_{Lim} at this point, in order to obtain the fastest trajectory; while the first equation of (43) compels the WMR to start from a standstill for the first segment of the path, i=1, or ensures continuity between adjacent segments for any other case. Note that $^{i-1}\sigma'(^{i-1}t)$ was set to its highest feasible value when planning the previous segment ^{i-1}Sg.

The cubic polynomial is selected to generate $^i\sigma(t)$, since it has just enough parameters to satisfy the boundary conditions in (42) and (43) and it has inverse, so that (41) can be computed. Thus, the t×s function of the path can be expressed as:

$$^i\sigma(t) = {}^i\sigma_0 t^3 + {}^i\sigma_1 t^2 + {}^i\sigma_2 t + {}^i\sigma_3 \qquad (44)$$

By applying (42) and (43) to (44), the boundary conditions can be summed up as:

$$\begin{bmatrix} 0 & 0 & 0 & 1 \\ \left(^i t\right)^3 & \left(^i t\right)^2 & {}^i t & 1 \\ 0 & 0 & 1 & 0 \\ 3\left(^i t\right)^2 & 2\,{}^i t & 1 & 0 \end{bmatrix} \begin{bmatrix} {}^i\sigma_0 \\ {}^i\sigma_1 \\ {}^i\sigma_2 \\ {}^i\sigma_3 \end{bmatrix} = \begin{bmatrix} {}^{i-1}ss \\ {}^i ss \\ {}^{i-1}v \\ {}^i v \end{bmatrix} \qquad (45)$$

$^i t$ must be computed to confine $^iV(s)$ into the safety zone of the s×v plane limited by [0≤V(s), V(s)≤V_{Lim}(s)], and its first derivative into the safety zone of the s×a plane limited by [-d_{Lim}(s)≤V'(s), V'(s)≤a_{Lim}(s)]; further, $^i t$ must be computed to ensure the continuity of $^i\sigma(t)$ up to its second derivative for the entire domain of the path, specifically between adjacent segments.

The magnitude $^i\sigma''(t)$ is important because local extrema of V_{Lim} are always located at the ends of the segments that partition the path. The velocity planned for these points is the maximum possible, under (43). If a maneuver with positive acceleration is planned for a segment whose end point is a local maximum of V_{Lim}, the velocity boundary will be violated at the beginning of the next segment. A similar situation would occur for a local minimum when negative acceleration is planned.

A direct approach to the problem would involve solving a large system of non-linear inequalities, a process of very high computational cost (Muñoz, 1995). Therefore, a method is proposed, working mainly with closed mathematical expressions and thereby reducing computational cost significantly. The method is based on setting the acceleration to zero at the segmentation points, so that acceleration continuity is ensured and the problem of failing to meet the velocity constraints just after local extrema of V_{Lim}, explained in a previous paragraph, is overcome.

The maximum acceleration of ^iSg, ^iA, is planned to be reached at an intermediate point, $^i t_c$, which yields one of the acceleration profiles showed in Fig. 2, therefore:
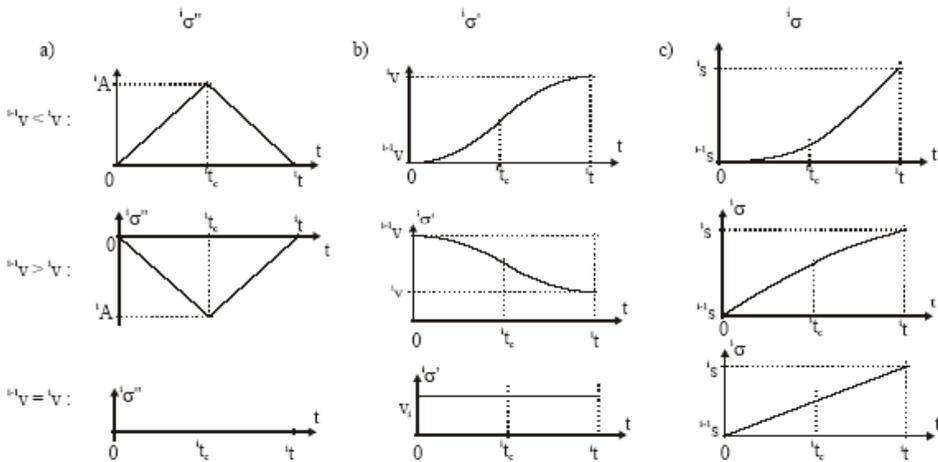
Fig. 2. Possible solutions in static environments. A) Acceleration profile; b)Velocity profile;
c)Trajectory.

$$^i\sigma''(t) = {}^i\sigma''_1(t) \cap {}^i\sigma''_2(t); \quad \begin{cases} {}^i\sigma''_1(t) = \dfrac{{}^iA}{{}^it_c}\,t & ....t \leq {}^it_c \\[3mm] {}^i\sigma''_2(t) = \dfrac{{}^iA}{{}^it - {}^it_c}\left({}^it - t\right) & ....t > {}^it_c \end{cases} \quad (46)$$

By integrating (46) and calculating the integration constants to satisfy the initial velocity
boundary condition of (43) and to ensure continuity between $^i\sigma'_1(t)$ and $^i\sigma'_2(t)$, the velocity
profile of the segment, also plotted in Fig. 2, is:

$$^i\sigma'(t) = {}^i\sigma'_1(t) \cap {}^i\sigma'_2(t); \quad \begin{cases} {}^i\sigma'_1(t) = {}^{i-1}v + \dfrac{{}^iA}{2\,{}^it_c}\,t^2 & ....t \leq {}^it_c \\[3mm] {}^i\sigma'_2(t) = {}^{i-1}v + \dfrac{{}^iA}{{}^it - {}^it_c}\left({}^it\,t - \dfrac{t^2}{2} - \dfrac{{}^it_c\,{}^it}{2}\right) & ....t > {}^it_c \end{cases} \quad (47)$$

Finally $^i\sigma(t)$ is computed by integrating (47) with the integration constants to satisfy the first
position boundary condition of (42) and to ensure its continuity at ^it_c:

$$^i\sigma(t) = {}^i\sigma_1(t) \cap {}^i\sigma_2(t); \quad \begin{cases} {}^i\sigma_1(t) = {}^{i-1}ss + {}^{i-1}v\,t + \dfrac{{}^iA}{6\,{}^it_c}\,t^3 & ....t \leq {}^it_c \\[3mm] {}^i\sigma_2(t) = {}^{i-1}ss + {}^{i-1}v\,t - {}^iA\,\dfrac{t^3 - 3\,{}^it\,t^2 + 3\,{}^it\,{}^it_c\,t - {}^it\left({}^it_c\right)^2}{6\left({}^it - {}^it_c\right)} & ....t > {}^it_c \end{cases} \quad (48)$$

The proposed algorithm initially selects ^it_c at the half-way point of ^iSg. So:

$$^it_c = \dfrac{{}^it}{2} \quad (49)$$

In this case, by taking into account the second velocity condition in (43) it is found that:

$$^it = 2\,\dfrac{{}^iv - {}^{i-1}v}{{}^iA}$$

In order to arrive at the position given by the second equation of (50), ^iA must be:

$$^i A = 4\,\frac{^i ss - \,^{i-1} ss}{\left(^i t\right)^2} - 4\,\frac{^{i-1} v}{^i t} \tag{51}$$

Therefore, by replacing (51) in (50), the value of $^i t$ needed for navigating $^i Sg$ with the selected velocity profile is computed as:

$$^i t = 2\,\frac{^i ss - \,^{i-1} ss}{^i v + \,^{i-1} v} \tag{52}$$

While the maximum acceleration must be:

$$^i A = \frac{\left(^i v\right)^2 - \left(^{i-1} v\right)^2}{^i ss - \,^{i-1} ss} \tag{53}$$

$^i\sigma(t)$ is fully defined by replacing (49), (52) and (53) in (48). It satisfies the boundary conditions of position and velocity at the initial and end points of $^i Sg$ and ensures the continuity up to its second derivative over both $^i Sg$ and the entire path domain. But $a_{Lim}(s)$ and $d_{Lim}(s)$ constraints must also be satisfied, hence it is necessary to check that:

$$\left.\begin{array}{l} \dfrac{d}{dt}\,^i V(s) \le a_{Lim}(s) \\[2mm] -\dfrac{d}{dt}\,V(s) \ge d_{Lim}(s) \end{array}\right\} \quad \forall s \subset \left[^{i-1} ss,\ ^i ss\right] \tag{54}$$

When $^{i-1}v < {}^i v$, an acceleration maneuver is planned for the segment. So if the acceleration restriction fails, it occurs because its upper boundary, the first inequality of (54), is violated. This problem can be solved by decreasing the final velocity to the maximum permitted by the maximum feasible acceleration in the segment. Thus, the velocity at the end of the segment is modified to be:

$$^i v* = \sqrt{\left(^{i-1} v\right)^2 + \,^i A^*\left(^i ss - \,^{i-1} ss\right)} \tag{55}$$

where the modified acceleration, $^i A^*$, is computed as:

$$^i A* = \min\left\{a_{Lim}(s) \,/\, s \subset \left[^{i-1} ss,\ ^i ss\right]\right\} \tag{56}$$

And the time that the WMR takes to navigate the i-th segment is recomputed as:

$$^i t* = 2\,\frac{^i v* - \,^{i-1} v}{^i A*} \tag{57}$$

Obviously, $^i v* < {}^i v$, since $^i A* < {}^i A$, and it is not necessary to check the velocity constraint at the end point of $^i Sg$ again.

On the other hand, when $^{i-1}v < {}^i v$, a deceleration maneuver is planned for the segment. Therefore, if the acceleration restriction fails, it occurs because its lower boundary, the second inequality of (54), is violated. In this case it is not possible to reduce deceleration by increasing the final velocity, since $^i v$ was selected as the maximum permitted by V_{Lim}. As a consequence, it becomes necessary to decrease the initial velocity to the maximum permitted by the maximum feasible deceleration:

$$^{i-1} v* = \sqrt{\left(^i v\right)^2 - \,^i A^*\left(^i ss - \,^{i-1} ss\right)} \tag{58}$$

where $^i A^*$ is computed as:

$$^i A^* = -\,^i d* = -\min\left\{d_{Lim}(s) \,/\, s \subset \left[^{i-1} ss,\ ^i ss\right]\right\} \tag{59}$$

And the new time it takes to navigate the i-th segment is recomputed by:

$$^{i}t* = 2 \frac{^{i-1}v - ^{i}v*}{^{i}d*} \tag{60}$$

This strategy requires the velocity profile of ^{i-1}Sg to also be modified by planning $^{i-1}v*$ as the velocity at its final posture, in order to ensure the continuity of $V(s)$.

The proposed algorithm ensures the velocity constraint will be satisfied at the end points of ^{i}Sg, but not along the entire segment. Thus, it must be checked:

$$^{i}V(s) \leq V_{Lim}(s) \quad \forall s \subset \left[^{i-1}ss, \ ^{i}ss\right] \tag{61}$$

If this constraint is violated now, the iterative processes detailed in section 5.3 must be performed. When $^{i}V(s)$ was computed by the proposed algorithm, inequality (61) failed in very few cases for the tests carried out with the WMR RAM, mentioned in section 7, and therefore the iterative strategies were necessary very infrequently. The same results can be expected for any WMR that works with spatial paths planned as smooth curves (continuous in curvature), because V_{Lim} depends on the curvature of the path and its first derivative with respect to the path length, and these functions are smooth if the spatial path is smooth.

The last task to build $^{i}\sigma(t)$ involves undoing the time shifting, i.e., setting the time at the initial point of ^{i}Sg equal to the time at the final point of ^{i-1}Sg. That is:

$$\left. \begin{array}{ll} \text{for} \quad i = 1 & ^{i}\sigma(t) = ^{i}\sigma(t) \\ \text{for} \quad i = 2 \text{ to } p+1 & ^{i}\sigma(t + ^{i-1}t) = ^{i}\sigma(t) \end{array} \right\} \quad t \subset \left[0, \ ^{i}t\right] \tag{62}$$

5.3. Velocity profile modification to satisfy velocity constraints in segments.

5.3.1. Acceleration maneuvers

When (61) is not satisfied in ^{i}Sg and $^{i-1}v < ^{i}v$, $^{i}V(s)$ is iteratively slowed down by delaying $^{i}t_c$ from the half-way point of ^{i}Sg to a value $^{i}t_c*$, given by:

$$^{i}t_c* = \frac{N-1}{N} \ ^{i}t \quad ; N = 3, 4, 5 \dots \tag{63}$$

Then $^{i}\sigma(t)$ is recomputed under (48) but substituting $^{i}t_c$ with $^{i}t_c*$ and with the maximum acceleration recomputed to be a value that satisfies the conditions in (42) and (43):

$$^{i}A* = 2 \frac{2\left(^{i}v\right)^2 - ^{i-1}v \ ^{i}v - \left(^{i-1}v\right)^2}{3\left(^{i}ss - ^{i-1}ss\right) + ^{i}t_c*\left(^{i}v - ^{i-1}v\right)} \tag{64}$$

And ^{i}t is substituted with $^{i}t*$ computed under (57). N in (63) keeps on increasing and $^{i}\sigma(t)$ being modified until (61) is satisfied.

By deriving (64) with respect to $^{i}t_c*$:

$$\frac{d}{d \ ^{i}t_c*} \ ^{i}A* = -\frac{^{i}A*}{3\left(^{i}ss - ^{i-1}ss\right) + ^{i}t_c*\left(^{i}v - ^{i-1}v\right)}\left(^{i}v - ^{i-1}v\right) \tag{65}$$

It is observed that for acceleration maneuvers the derivative is always negative. Thus, delaying $^{i}t_c*$, i.e. increasing $^{i}t_c*$ in (73), implies reducing the maximum acceleration needed to satisfy the boundary conditions of the segment, although the time consumed in navigating it is increased. Hence a long trajectory with low velocity at all points is planned, and the velocity constraint is satisfied. Obviously, if the acceleration constraint was satisfied before the modification, it is also verified now and does not need to be checked again.

5.3.2. Deceleration maneuvers

When (61) is not satisfied in ^{i}Sg and $^{i-1}v>^iv$, a similar strategy to the one described in the previous section for acceleration maneuvers is applicable, but in this case, in order to plan a maneuver of lower deceleration, ^it_c is advanced to a value $^it_c^*$, which is given by:

$$^i t_c^* = \frac{1}{N}\, ^i t \quad ; N = 3, 4, 5 \dots \dots \tag{66}$$

Then $^i\sigma(t)$ is recomputed by (48), substituting ^it_c with $^it_c^*$, maximum deceleration recomputed by (64) to be a value that satisfies the conditions in (50) and (51), and substituting ^it with $^it^*$ computed under (57). N in (66) keeps on increasing and $^i\sigma(t)$ being modified until (61) is satisfied.

It can be observed that the derivative of the modified deceleration, $^id^*=-^iA^*$, given by (65), is always positive when $^{i-1}v>^iv$. Therefore, advancing the control point implies reducing the maximum deceleration and therefore increasing the time consumed in tracking the trajectory. Consequently if the deceleration constraint was satisfied before the modification, it is also verified now and does not need to be checked again. This modification leads to planning lower velocities at all points of the segment, except for points that are very close to the end point. Only when the velocity constraint fails in this region but not at the end point is the proposed method unable to find a feasible solution. But such a situation is not expected to occur when the spatial path is planned as smooth curves and the segmentation of section 5.1 is applied.

5.4. Velocity planning algorithm in static environments

The comprehensive algorithm proposed in this chapter for VP in static environments is summarized in the following flowchart:

1. *Compute* $V_{Lim}(s)$; $s \subset [0, S]$
2. *Compute* $a_{Lim}(s)$; $s \subset [0, S]$
3. *Compute* $d_{Lim}(s)$; $s \subset [0, S]$
4. *Create Ps={0, 1ss, ... pss, S}*
5. *VP for iSg.* **for** *i=1 to p+1* **do**
6. *Set position boundary conditions, (42)*
7. *Set velocity boundary conditions, (43)*
8. *Compute iσ(t), (48), with it, (52), itc, (49), and iA, (53)*
9. **if** *acceleration constraint, (54), fails* **then**
10. **if** $^{i-1}v<^iv$ **then**
11. *Modified iσ(t) with iv*, (55), iA*, (56), and it*, (57)*
12. **end if**
13. **if** *i-1v>iv* **then**
14. *Modified iσ(t) with i-1v*, (58),, iA*, (59), and it*, (60)*
15. *Recompute $^{i-1}V(s)$ for ^{i-1}Sg with new $^iv^*$, (58)*
16. **end if**
17. **end if**
18. **if** *velocity constraint, (61), fails* **then**
19. **if** *i-1v<iv* **then**
20. *N=1*
21. **repeat**
22. *N=N+2.*
23. *Advance ^it_c, (63)*
24. *Compute $^iA^*$, (64) and $^it^*$, (57)*
25. *Compute $^i\sigma(t)$, (48), with values in* **Error! Reference source not found.**
26. **until** *satisfy velocity constraint, (61)*
27. **end if**
28. **if** *$^{i-1}v>^iv$* **then**
29. *N=1*
30. **repeat**
31. *N=N+2*
32. *Delay ^it_c, (66)*
33. *Compute $^iA^*$, (64), and $^it^*$, (57)*
34. *Compute $^i\sigma(t)$, (48), with values in* **(Error! Reference source not found.)**
35. *until satisfy velocity constraint (61)*
36. **end if**
37. **end if**
38. *Shift the origin of the time, (62)*
39. *Compute $^iV(s)$, (41)*
40. **end for**

6. Velocity Planning in Dynamic Environments

Crossing points of moving obstacles with the WMR's path can be represented by a set $R=[^1R$ $^2R \ldots {}^lR]$ of rectangular forbidden regions in the s×t plane (Liu & Wu, 2001). Each region ^kR is defined by a segment of the space with end points $[^ks_{ini}, {}^ks_{fin}]$ and an interval of time limited by $[^kt_{ini}, {}^kt_{fin}]$, as shown in Fig. 3.

The function $^i\sigma(t)$, computed previously, is not a valid solution of the VP problem if it intersects a forbidden region.
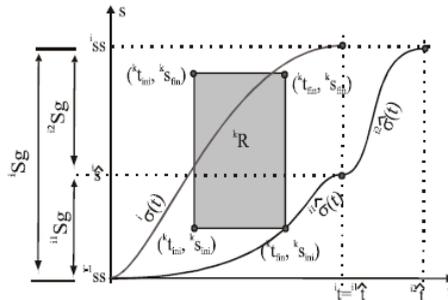
Fig. 3. Velocity planning in dynamic environments.

The problem can be solved by planning a slower trajectory that allows the moving obstacle to cross the robot's path before it arrives at the crossing region. The opposite solution, to plan a faster trajectory so that the robot passes through the dangerous space before the obstacle arrives, it is not possible, since $^i\sigma(t)$ was built to achieve the highest feasible velocity profile.

A secondary aim of the modification strategy, as stated in section 2, is to achieve a low computational cost. This is accomplished by both modifying only the segments adjacent to where the obstacle is found and by avoiding iterative strategies.

If $^i\sigma(t)$ intersects the forbidden region ^kR, the trajectory is slowed down by dividing iSg into two subsegments: $^iSg = {}^{i1}Sg \cup {}^{i2}Sg$. The first one, ^{i1}Sg, plans a velocity profile, , that makes the WMR avoid the obstacle. The second segment, ^{i2}Sg, plans a velocity profile, $^{i2}\hat{V}(t)$, that makes the WMR arrive at the final position imposed by the second position boundary condition of (42) with the velocity imposed by the second velocity boundary condition of (43).

6.1. Planning the velocity profile to avoid moving obstacles

The first piece of the modified sxt function, $^{i1}\hat{\sigma}(t)$, is planned to make the WMR avoid ^kR by compelling it to pass through its first point at its last time, i.e.:

$$^{i1}\hat{\sigma}\left(^k t_{fin}\right) = {}^k s_{ini} \qquad (67)$$

The total time for the subsegment ^{i1}Sg, $^{i1}\hat{t}$, is set equal to ^it, the time planned for ^iSg in static environments, and its point of maximum acceleration, $^{i1}\hat{t}_c$, to the same time as for static environments, ^it_c. This goal can be achieved by selecting the maximum acceleration in ^{i1}Sg so that $^{i1}\hat{\sigma}(t)$ is similar to $^i\sigma(t)$ computed by (48) but substituting ^iA with a lower

acceleration, $^{il}\hat{A}$, calculated to satisfy (67). This value is given by:

$$\text{if } ^{k}t_{fin} \leq {}^{il}\hat{t}_{c} \dots\dots \quad {}^{il}\hat{A} = 6\left({}^{k}s_{ini} - {}^{i-1}ss - {}^{i-1}v \, {}^{k}t_{fin} \right) \tag{68}$$

$$\text{if } ^{k}t_{fin} > {}^{il}\hat{t}_{c} \dots\dots \quad {}^{il}\hat{A} = 6\left({}^{k}s_{ini} - {}^{i-1}ss - {}^{i-1}v \, {}^{k}t_{fin} \right) \frac{{}^{il}\hat{t} - {}^{il}\hat{t}_{c}}{3\,{}^{il}\hat{t}\left({}^{k}t_{fin}\right)^{2} - \left({}^{k}t_{fin}\right)^{3} - 3\,{}^{il}\hat{t}_{c}\,{}^{il}\hat{t}\,{}^{k}t_{fin} + \left({}^{il}\hat{t}_{c}\right)^{2}{}^{il}\hat{t}}$$

The WMR's position at the end of ^{il}Sg, $^{i}\hat{s}$, is a lower value than is, since a lower acceleration is used during the same time. Therefore the WMR does not arrive at the final position condition, ^{i}ss, given by the second equation of (42), but rather at:

$$^{i}\hat{s} = {}^{i-1}ss + {}^{i-1}v \, {}^{il}\hat{t} + \frac{{}^{il}\hat{A}}{6} \frac{\left({}^{il}\hat{t}\right)^{3}}{{}^{il}\hat{t}_{c}} \tag{69}$$

Likewise, the velocity of the WMR at the final posture of ^{il}Sg, $^{i,il}\hat{v}(t)$ is also a lower value than ^{i}v, so it does not satisfy the final velocity condition of (43), but is rather:

$$^{i,il}\hat{v} = {}^{i-1}v + \frac{{}^{il}\hat{A}}{2} \frac{\left({}^{il}\hat{t}\right)^{2}}{{}^{il}\hat{t}_{c}} \tag{70}$$

Since $^{il}\hat{\sigma}(t)$ is computed from $^{i}\sigma(t)$ by applying a lower acceleration for the same time and the velocity and acceleration constraints were successfully checked for $^{i}\sigma(t)$, the two upper limits are also satisfied by $^{il}\hat{\sigma}(t)$. But two lower limits can be violated and they must be checked: the positive magnitude of the velocity and the deceleration constraint:

$$^{il}\hat{\sigma}'(t) \geq 0 \quad \forall t \subset \left[0, \, {}^{il}\hat{t}\right] \tag{71}$$

$$^{il}\hat{A} \geq -\max\left\{d_{Lim}(s) \quad / \, s \subset \left[{}^{i-1}ss, \, {}^{i}\hat{s}\right]\right\} \tag{72}$$

If it $^{il}\hat{\sigma}(t)$ fails to meet one of these constraints, the processes detailed in subsections 6.1.1 or 6.1.2 must be carried out, respectively.

The last task to build $^{il}\hat{\sigma}(t)$ involves setting the time at the initial point of the segment equal to the time at the final point of the previous segment:

$$^{il}\hat{\sigma}\left(t + {}^{i-1}t\right) = {}^{il}\hat{\sigma}(t) \tag{73}$$

Finally, the velocity profile $^{il}\hat{V}(s)$ is computed by applying (41).

6.1.1. Modification of the velocity profile to keep velocity positive

When the lower boundary of velocity, (71), is violated, the initial velocity and the maximum deceleration of ^{il}Sg are modified in order to plan for the WMR to arrive at the end posture at zero velocity, i.e.:

$$^{il}\hat{\sigma}'\left({}^{i}\hat{t}\right) = {}^{i,il}\hat{v}* = 0 \tag{74}$$

By making $^{i1}\hat{\sigma}(t)$ fulfill (74) while still satisfying (67) to avoid ^{k}R, the initial velocity and maximum acceleration of ^{i1}Sg are modified to be $^{i-1,i1}\hat{v}*(t)$ and $^{i1}\hat{A}*$:

$$
\text{if} \quad {}^{k}t_{fin} \le {}^{i}\hat{t}_{c} \ldots \begin{cases} {}^{i1}\hat{A}* = -6\left({}^{k}s_{ini} - {}^{i-1}ss\right)\dfrac{{}^{i1}\hat{t}_{c}}{{}^{k}t_{fin}\left(3\,{}^{i1}\hat{t}\,{}^{i1}\hat{t}_{c} - \left({}^{k}t_{c}\right)^{2}\right)} \\[4mm] {}^{i-1,i1}\hat{v}* = 3\left({}^{k}s_{ini} - {}^{i-1}ss\right)\dfrac{{}^{i1}\hat{t}_{c}\,{}^{i1}\hat{t}}{{}^{k}t_{fin}\left(3\,{}^{i1}\hat{t}\,{}^{i1}\hat{t}_{c} - \left({}^{k}t_{c}\right)^{2}\right)} \end{cases}
$$

$$
\text{if} \quad {}^{k}t_{fin} > {}^{i}\hat{t}_{c} \ldots \begin{cases} {}^{i1}\hat{A}* = -6\left({}^{k}s_{ini} - {}^{i-1}ss\right)\dfrac{{}^{i1}\hat{t} - {}^{i1}\hat{t}_{c}}{3\,{}^{k}t_{fin}\left({}^{i1}\hat{t}\right)^{2} + \left({}^{k}t_{fin}\right)^{3} - 3\,{}^{i1}\hat{t}\left({}^{k}t_{fin}\right)^{2} - \left({}^{i1}\hat{t}_{c}\right)^{2}\,{}^{i1}\hat{t}} \\[4mm] {}^{i-1,i1}\hat{v}* = 3\left({}^{k}s_{ini} - {}^{i-1}ss\right)\dfrac{\left({}^{i1}\hat{t} - {}^{i1}\hat{t}_{c}\right){}^{i1}\hat{t}}{3\,{}^{k}t_{fin}\left({}^{i1}\hat{t}\right)^{2} + \left({}^{k}t_{fin}\right)^{3} - 3\,{}^{i1}\hat{t}\left({}^{k}t_{fin}\right)^{2} - \left({}^{i1}\hat{t}_{c}\right)^{2}\,{}^{i1}\hat{t}} \end{cases}
$$

(75)

Obviously, in this case the previous segment, ^{i-1}Sg, must be recomputed following the method stated in section 5 with a new velocity boundary condition, given by:

$$
{}^{i1}\hat{\sigma}'(0) = {}^{i-1}v = \begin{cases} 0 & \text{if } i = j \\ {}^{i-1}\sigma'({}^{i-1}t) & \text{if } i \ne j \end{cases}
$$

$$
{}^{i1}\hat{\sigma}'\left({}^{i1}\hat{t}\right) = {}^{i-1,i1}\hat{v}*
$$

(76)

6.1.2 Modification of the velocity profile to satisfy the deceleration constraint

When the lower boundary of acceleration, (72), is violated, the maximum deceleration is decreased by setting it to its maximum feasible value. Thus, $^{i1}\hat{A}$ is replaced by:

$$
{}^{i1}\hat{A}* = -\max\left\{d_{Lim}(s) \quad / s \subset \left[{}^{i-1}ss,\ {}^{i}\hat{s}\right]\right\}
$$

(77)

Additionally, the initial velocity of the subsegment must be sufficiently reduced to keep on satisfying (76) when the maximum deceleration is $^{i1}\hat{A}*$, yielding:

$$
\text{if } {}^{k}t_{fin} \le {}^{i1}\hat{t}_{c} \ldots \quad {}^{i-1,i1}\hat{v}* = \frac{{}^{k}s_{fin} - {}^{i-1}ss}{{}^{k}t_{fin}} - \frac{{}^{i}\hat{A}*}{6}\frac{\left({}^{k}t_{fin}\right)^{2}}{{}^{i1}\hat{t}_{c}}
$$

(78)

$$
\text{if } {}^{k}t_{fin} > {}^{i1}\hat{t}_{c} \ldots \quad {}^{i-1,i1}\hat{v}* = \frac{{}^{k}s_{fin} - {}^{i-1}ss}{{}^{k}t_{fin}} - \frac{{}^{i}\hat{A}*}{6}\frac{3\,{}^{i1}\hat{t}\left({}^{k}t_{fin}\right)^{2} - \left({}^{k}t_{fin}\right)^{3} - 3\,{}^{i1}\hat{t}_{c}\,{}^{i1}\hat{t}\,{}^{k}t_{fin} + \left({}^{i1}\hat{t}_{c}\right)^{2}\,{}^{i1}\hat{t}}{\left({}^{i1}\hat{t} - {}^{i1}\hat{t}_{c}\right){}^{k}t_{fin}} -
$$

Thus, $^{i1}\hat{\sigma}(t)$ is recomputed from $^{i}\sigma(t)$ by replacing ^{i}A with $^{i1}\hat{A}*$ and ^{i-1}v with $^{i-1,i}\hat{v}*$. The position and velocity of the WMR at the end of ^{i1}Sg must be also recomputed by using (78) and (79) respectively, with the new values for $^{i1}\hat{A}*$ and $^{i-1,i1}\hat{v}*$.

Finally, it is necessary to recompute the previous segment, ^{i-1}Sg, making its final velocity equal to $^{i-1,i1}\hat{v}*$ to avoid discontinuities between adjacent segments.

6.2. Planning the velocity profile to arrive at the final position

A second function $^{i2}\hat{\sigma}(t)$ is planned to be attached to in order to make the WMR arrive at ^{i}ss. The following position boundary conditions must be satisfied by $^{i2}\hat{\sigma}(t)$:

$$
\begin{aligned}
{}^{i2}\hat{\sigma}\!\left({}^{i1}\hat{t}\right) &= {}^{i}\hat{s} \\
{}^{i2}\hat{\sigma}\!\left({}^{i2}\hat{t}\right) &= {}^{i}ss
\end{aligned} \tag{79}
$$

Additionally, ${}^{i2}\hat{\sigma}(t)$ must ensure continuity between ${}^{i1}Sg$ and ${}^{i2}Sg$. Therefore:

$$
\begin{aligned}
{}^{i2}\hat{\sigma}'\!\left({}^{i1}\hat{t}\right) &= {}^{i,i1}\hat{v} \\
{}^{i2}\hat{\sigma}'\!\left({}^{i2}\hat{t}\right) &= {}^{i,i2}\hat{v} = V_{Lim}\!\left({}^{i2}\hat{t}\right)
\end{aligned} \tag{80}
$$

${}^{i2}\hat{\sigma}(t)$ is generated by the procedure presented in section 5, but with the boundary conditions given by (88) and (89).

6.3. Velocity planning algorithm in dynamic environments

The comprehensive algorithm for VP in dynamic environments is summarized as follows:

1. Compute , (56) with , (77)
2. **if** lower boundary of velocity,(80) ,fails **then**
3. Compute , (84)
4. Compute , (84)
5. Compute ,(56) ,with values of 3, 4
6. Replan i-1V(s) of i-1Sg with velocity boundary conditions of (85)
7. **end if**
8. **if** boundary of deceleration, (81)fails **then**
9. Compute , (86)
10. Compute , (87)
11. Compute ,(56),with values 9, 10
12. Replan i-1V(s) of i-1Sg with velocity boundary conditions of (85)
13. **end if**
14. Shift the origin of time, (82)
15. Compute iV(s), (49)
16. Compute , (78)
17. Compute , (79)
18. Vp for i2Sg:
19. Algorithm for static environments with position boundary condition of (88) and velocity boundary conditions of (89).

7. Experimental and simulation results

The velocity planner described in previous sections is applied to the robot RAM (Ollero et al., 1993), a WMR designed for navigation with high maneuverability in indoor and outdoor industrial environments. The maximum velocity its traction motors can reach is 1.7m/s and their maximum acceleration is 10.8m/s2

The spatial paths of the robot are generated as cubic spirals, since this is curve that shows the best mechanical characteristics to be tracked by a WMR (Prado, 2000): continuity of position, orientation and curvature; bounded curvature; arbitrary selection of position and orientation at the initial and final postures; best dynamic behavior of the WMR, specifically lowest tracking errors, when it navigates with open loop control.

Fig. 4 plots the planned trajectory and the trajectory actually tracked by RAM in a static environment. The time consumed for the trajectory which was planned using the proposed algorithm is 62.2s, 24.7% longer than the time that would be consumed if RAM navigated the entire path at its maximum velocity. Results show very low tracking errors: spatial errors are lower than 0.1m for a path of 84.4m, and temporal errors are lower than 1s for a trajectory of 62.2s.
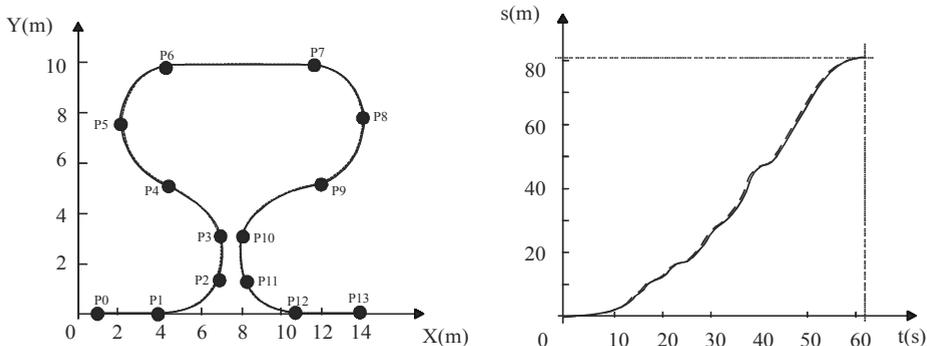
Fig. 4. Planned and tracked trajectory with maximum velocity 1.7m/s.
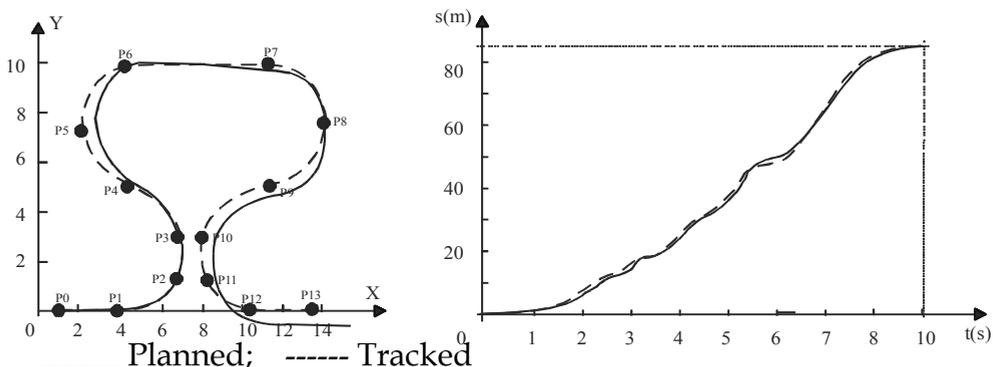
_____ Planned;   ------ Tracked

Fig. 5. Planned and tracked trajectory with maximum velocity 17m/s.

In order to test the VP algorithm in trajectories with high dynamic requirements, simulation experiments have been performed with a model of RAM (Prado et al., 2002) that increases the maximum velocity of its tractive motors up to 17m/s, also increasing its acceleration and deceleration performances by the same rate.

Fig. 5 shows the planned trajectory and the trajectory tracked by the model of RAM when executing the same path as in Fig. 4. The total time planned by the proposed algorithm is 35.5% longer than the time that would be consumed if the entire path was navigated at maximum velocity. But it is only 15.2% longer than the time that would be consumed if the velocity was set to its upper boundary function, disregarding the acceleration constraint and continuity. Moreover, the greatest difference occurs in the first segment, because the acceleration constraint causes the planned trajectory to take 3.7s (26.2% of the total time) to reach the velocity boundary function. Low errors are found again: spatial errors are always

lower than 0.80m and temporal errors are lower than 2s for a trajectory of 14.2 s.

The velocity and acceleration profiles of the trajectory of Fig. 5, along with their boundary functions are plotted in Fig. 6. It can be observed that both velocity and acceleration remain within the safety limits while still ensuring their continuity.

Similar results were found for other simulated trajectories with paths generated as cubic splines, but also as clothoids or polar splines (Prado et al. 2003). Particularly regarding velocity constraints, not one path of RAM planned as curves with continuous curvature was found for which the iteration strategies of section 5.3 were needed. Similar results can be expected for any WMR if its spatial path is planned as smooth curves.
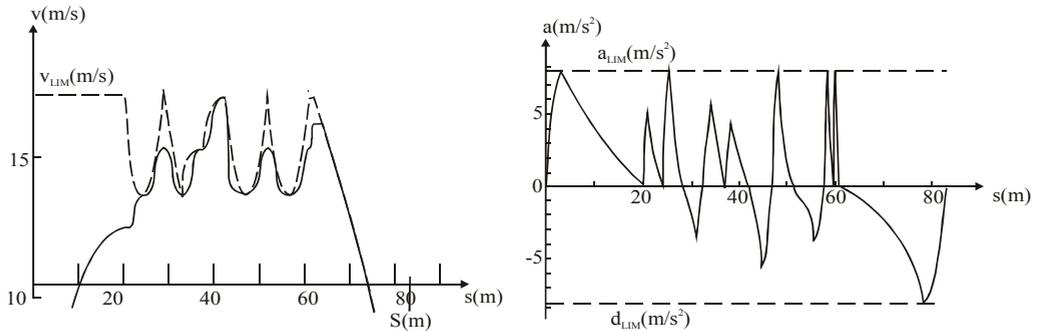
Fig. 6. a) Velocity profile of the trajectory of Fig. 5; b) acceleration profile of the trajectory of Fig. 5

_____ In static environments; ------- Modified for dynamic environments
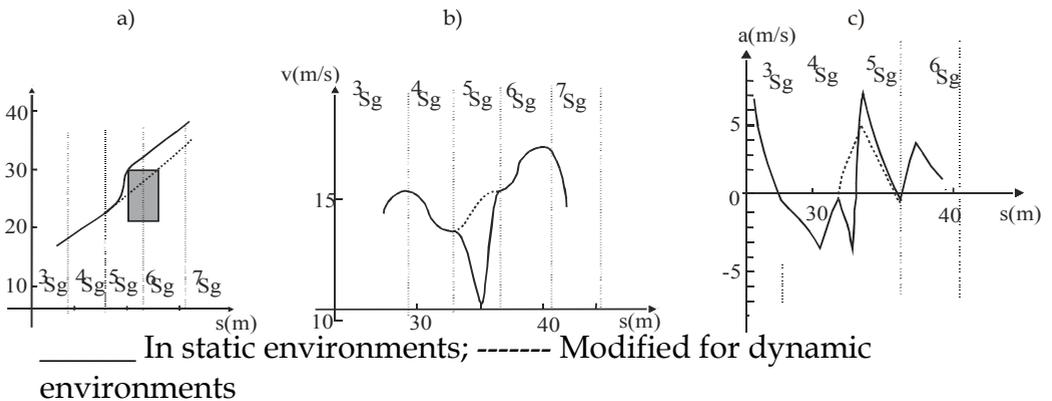
Fig. 7. a) Planned path; b)Planned velocity profile; c) Planned acceleration profile

When a moving obstacle intersects the WMR's path, its velocity profile is slowed down, as explained in section 6. Fig. 7 shows the effect of including a forbidden region that intersects the 5th segment of the fast trajectory of Fig. 5. The maneuver involves an increase of 4.6s in navigation time. It can be observed that the obstacle is successfully avoided by only modifying the trajectory at the segment that intersects with the forbidden region. The modified trajectory is also continuous in velocity and acceleration and both functions remain within their respective bounded regions.

8. Conclusion

A method to compute velocity, acceleration and deceleration upper boundary functions for WMRs is described in this chapter. These functions confine the operating variables into safety regions in order to generate trajectories that are feasible, safe and with low tracking errors. The method first deals with issues related to the performances of steering, tractive and braking subsystems of wheeled robots, as well as mechanical properties of wheel-ground contact. Next, the motion capabilities of the robot's guide point are calculated as functions of the previous results, attending to the kinematics and dynamics of the complete robot system. Operational constraints caused by the need for fitting and synchronising the robot's motion with its environment are also defined.

The upper limits for velocity based on robot dynamics are fixed not only to avoid total vehicle slippage, as was previously done by other authors, but also to bound the spatial and temporal errors of a trajectory in which the space path generator and the trajectory tracker work under the kinematic problem solution. The definition of these boundary functions depends on either simulation outcomes of a sufficiently accurate dynamic model of the WMR, as complex as needed since it works offline, or on an appropriate number of experimental tests.

Topics involving navigation over 3D terrains and vehicles with very high velocities, such as the presence of ground irregularities, transients for non-stationary manoeuvres, aerodynamic effects, gyroscope torques, etc., are not dealt with in this chapter, though they should be taken into account for robots used in tasks such as outdoor exploration or vehicles developed to run on roads without a driver.

The resulting bounds are included in an algorithm to plan time optimal velocity profiles for a WMR on specific paths. Dealing with PP and VP simultaneously may make it possible to plan faster trajectories in some situations, but such methodologies involve more complex algorithms where it is very difficult to include a significant number of velocity and acceleration constraints that include nonintegrable functions.

The velocity planner proposed in this chapter is able to generate a trajectory with favorable tracking conditions, since:
-It confines the velocity and its first derivative into safety zones limited by functions that can consider any velocity, acceleration and deceleration constraint which can be expressed as a

function of the spatial path.
- It ensures the continuity of the velocity and acceleration over the entire path.
- The jerk of the WMR is bounded.
- It is of low computational cost. Nonetheless, iteration strategies are proposed to solve some specific situations, but such situations are not expected to occur when the spatial path is planned as smooth curves.

Initially, the problem is addressed for static environments, and later an algorithm is proposed to locally modify the velocity profile when crossing points where moving obstacles are detected. It is also an algorithm of low computational cost, thereby providing a beneficial characteristic for its possible use in local control systems.
The method was tested on the WMR RAM with paths planned as cubic spirals. Its successful performance was confirmed by experimental results for the feasible velocities of the WMR prototype and by simulation results for higher velocities.

9. References

Cherif, M. (1999). Motion planning for all-terrain vehicles: a physical modelling approach for coping with dynamic and contact interaction constraints, IEEE Transactions on Robotics and Automation, Vol. 15, No. 2, 202-218, ISSN 0882-4967

Choi, J.S & Kim, B.K. (2001). Near-time-optimal trajectory planning for wheeled mobile robots with translational and rotational sections, IEEE Transactions on Robotics and Automation, Vol. 17, No. 1, 85-90, ISSN 0882-4967.

Gill, M.A.C. & Zomaya, A.Y. (1998). Obstacle avoidance in multi-robot systems: experiments in parallel genetic algorithms. World Scientific Publishing Company, ISBN 9-8102-3423-6, New Jersey, USA.

Guarino Lo Bianco, C. & Romano, M. (2005), Bounded velocity planning for autonomous vehicles, Proceedings of 18th IEEE/RSJ International Conference on Intelligent Robots and Systems, pp. 685 – 690, ISBN 0-7803-8913-1, Edmonton, Canada, August 2005, IEEE, New York, USA.

Kiguchi, K.; Watanabe, K. & Fukuda T. (2004) Trajectory planning of mobile robots using DNA computing. Journal of Advanced Computational Intelligence and Intelligent Informatics, Vol. 8, No. 3, 295-301, ISSN 1343-0130.

Krishna, K.M., Alami, R. & Simeon, T.(2006) Safe proactive plans and their execution. Robotics and Autonomous Systems, Vol. 54, No. 3, 244-255, ISSN 0921-8890.

LaValle, S.M. & Kuffner, J.J. (2001). Randomized kinodynamic planning". International Journal of Robotic Research, Vol. 20, No. 5, 278-400, ISSN 0278-3649.

Lepetic, M.; Klancar, G.; Skrjanc, I.; Matko, D. & Potocnik, B. (2003) Time optimal planning considering acceleration limits. Robotics and Autonomous Systems, Vol. 45, N0. 3-4, 199-210, 2003, ISSN 0921-8890.

Liu, G.Y. & Wu, C.J.(2001). A discrete method for time optimal motion planning a class

of mobile robots, Journal of Intelligent and Robotic Systems, Vol.32,No.1, 75-92, ISSN 09210296.

Muñoz, V.F. (1995) Trajectory planning for mobile robots, Ph.D. Thesis, University of Málaga, Spain (in Spanish)

Nearchou A. (1998). Path planning of a mobile robot using genetic heuristics, Robotica, Vol. 16, No. 5, 575-588, ISSN 0263-5747.

O´Dunlaing, C. (1987). Motion Planning with Inertial constraints, Algorithmica, Vol. 2, No. 1, 431-475, ISSN 0178-4617

Ollero, A.; Simon, A.; García F. & Torres, V. (1993). Integrated mechanical design of a new mobile robot, Proceedings of the First Symposium on Intelligent Components and Instruments for Control Applications, 461-466, Málaga, Spain, May 1992 Pergamon Press, United Kingdom

Prado, M. (2000) Dynamic modeling of mobile robots. Application to the trajectory planning, Ph.D. Thesis, University of Málaga, Spain.

Prado, M.; Simón, A. & Ezquerro, F. (2002). Velocity, acceleration and deceleration bounds for a time-optimal planner of a wheeled mobile robot, Robotica, Vol. 2, N0. 2, 181-193, ISSN 0263-5747.

Prado, M.; Simon, A. Carabias, E.; Perez, A. & Ezquerro, F. (2003). Optimal velocity planning of wheeled mobile robots on specific paths in static and dynamic environments. Journal of Robotic Systems, Vol. 20, No. 12, 737-754, ISSN 0741-2223

Reisten, D.B. & Pin, F.G. (1994). Time-optimal trajectories for mobile robots with two independent acceleration-driven wheels. International Journal of Robotic Research, Vol. 13, No. 1, 38-54, ISSN 0278-3649.

Shiller, Z. (1999). Motion Planning for Mars Rover, Proceedings of I Workshop Robot Motion and Control, pp. 257-262, ISBN 0780356551, Kiekrz, Poland, June 1999, IEEE, New York, USA.

Takeshi, A. (1994). Motion planning for multiple obstacles avoidance of autonomous mobile robot using hierarchical fuzzy rule, Proceedings of IEEE International Conference on Multisensor Fusion and Integration for Intelligent System, pp. 265-271, ISBN 0780320727, Las Vegas, USA, October 1994, IEEE, New York, USA.

Wang, Y.; Han, L.; Li, M.; Wang, Q.; Zhou, J. & Cartmell, M. (2004). A real-time path planning approach without computation of Cspace obstacles. Robotica, Vo. 22, No. 2, 173-187, ISSN 0263-5747.

Weiguo, W.; Huitang, C. & Peng-Yung, W. (1999). Optimal Motion Planning for a Wheeled Mobile Robot, Proceedings of IEEE International Conference on Robotics and Automation, pp. 41-46, ISBN 0-7803-5180-0 , Michigan, USA, June 1999, IEEE, New York, USA.

Wong, J.Y. (2001), Theory of Ground Vehicles, John Wiley & Sons, ISBN 0-471-35461-9, New York, USA.

Zalama, E.; Gaudiano, P. & Lopez-Coronado, J. (1995). A real-time unsupervised neural network for the low level control of a mobile robot in nonstationay environment. Neural Networks, Vol. 8, No. 1, 103-1203, ISSN 0893-6080

Mobile Robots: Perception & Navigation

Edited by Sascha Kolski

Today robots navigate autonomously in office environments as well as outdoors. They show their ability to beside mechanical and electronic barriers in building mobile platforms, perceiving the environment and deciding on how to act in a given situation are crucial problems. In this book we focused on these two areas of mobile robotics, Perception and Navigation. This book gives a wide overview over different navigation techniques describing both navigation techniques dealing with local and control aspects of navigation as well es those handling global navigation aspects of a single robot and even for a group of robots.

How to reference

In order to correctly reference this scholarly work, feel free to copy and paste the following:

Maria Prado (2007). Optimal Velocity Planning of Wheeled Mobile Robots on Specific Paths in Static and Dynamic Environments, Mobile Robots: Perception & Navigation, Sascha Kolski (Ed.), ISBN: 3-86611-283-1, InTech, Available from:

http://www.intechopen.com/books/mobile_robots_perception_navigation/optimal_velocity_planning_of_wheeled_mobile_robots_on_specific_paths_in_static_and_dynamic_environme