2011 50th IEEE Conference on Decision and Control and
European Control Conference (CDC-ECC)
Orlando, FL, USA, December 12-15, 2011

Multi-optimization of $\boldsymbol{\eta}^3$-splines for autonomous parking

Gabriele Lini, Aurelio Piazzi, and Luca Consolini
University of Parma
Department of Information Engineering
Viale G.P. Usberti 181A, I-43124, Parma, Italy
emails: gabriele.lini@gmail.com, aurelio.piazzi@unipr.it, luca.consolini@unipr.it

Abstract— **This paper proposes a multi-optimization approach to the autonomous parking of car-like vehicles. It uses a polynomial curve primitive — the η^3-spline — to build up intrinsically feasible path maneuvers over which to minimize with a weighted sum method the total length of parking paths and the moduli of the maximum path curvature and curvature derivative. The approach takes into account the mandatory constraint of obstacle avoidance and maximal steering angle and the constraint of maximal curvature derivative which is a selectable limit to ensure the desired smoothness of the parking paths. Simulation results are included for a garage parking example.**

I. INTRODUCTION

Nowadays, autonomous parking of cars and wheeled robot vehicles is a significant research theme for mobile robotics and automotive industries. Indeed, the autonomous capability of parking a vehicle without, or with partial, human intervention is a key factor toward more *intelligent* vehicles [1]. This capability requires in real-time the acquisition of sensor data (from laser-scanners, cameras, etc.) to self-localize the vehicle in the parking scenario and to plan and execute motion maneuvers avoiding obstacles and to finally reach the parking slot with correct position and orientation.

Focusing on the planning of motion maneuvers of car-like vehicles, the parking problem can be theoretically introduced as follows: given an initial configuration and a final configuration of the vehicle, find a path joining the initial and final configurations such that: (1) the path is collision-free, i.e. the vehicle on the path avoids any collision with all the obstacles of the parking scenario (other cars, walls, curbs, etc.); (2) the path is feasible (or admissible), i.e. the vehicle on the path satisfies the differential constraints of the vehicle model (the nonholonomic constraints) and the actuator constraints (such as e.g. the bound on the maximal steering angle of the front wheels).

The parking problem without differential and actuator constraints becomes the so-called piano mover's problem which is a classic problem in the motion planning literature (cf. the books [2], [3] and the extensive references included). When the parking problem formulation is complete with both requirements 1 and 2, the approaches exposed in the literature are mainly based on a two-step procedure: First, a collision-free path that ignores differential (and actuator) constraints is determined. Then this path is suitably modified in order to accommodate to the constraints. In such a way, the first step just requires to pick up a solution technique for the piano mover's problem and in the second step ad hoc smoothing techniques or local steering methods are devised to accomplish a complete solution.

The two-step procedure was first proposed by Laumond et al. in [4] and subsequently a variety of variants has appeared [5]–[7] (also cf. [8] and references herein included). Other approaches to the parking problem – to cite a few of them – are based on *ad hoc* planning algorithms [9], [10], neural networks [11], [12], and fuzzy techniques [13]–[15]. In [16], parking path generation is obtained with fuzzy logic and $\boldsymbol{\beta}$-splines.

In this paper, we first address the parking problem as a smooth feedforward control problem where the vehicle's sought control inputs, the linear velocity and the front wheel steering angle, are C^1-signals, i.e. continuous time functions admitting derivatives that are still continuous. Then, the introduction of the concept of third-order geometric continuity of Cartesian paths and the procedure of dynamic path inversion as exposed in [17] permits the feedforward control problem to be reduced to a purely geometric problem followed by a velocity planning problem. This geometric problem regards the search of a sequence of feasible paths connecting the initial vehicle configuration to the final one while satisfying all the the required constraints (obstacle avoidance, maximum steering angle, etc.). In this context, a path is feasible if it is a G^3-path, i.e. a path that has continuity, along the curve, of the unit-tangent vector, curvature, and derivative of the curvature with respect to the arc length (cf. Section II).

The adoption of the $\boldsymbol{\eta}^3$-spline [18], [19], which is a polynomial curve primitive devised for the interpolation of two arbitrary Cartesian points with associated arbitrary G^3-data (unit tangent vector, curvature, and curvature derivative), allows approximating a sequence of G^3-paths with a sequence of $\boldsymbol{\eta}^3$-splines and in such a way it reduces the smooth parking problem to a search into a finite-dimensional space (cf. Section III). This search will be done in a multi-objective optimization framework [20] with constraints given by obstacle avoidance and bounds on the maximum values of the absolute curvature and curvature derivative along the parking maneuver splines. The considered indexes to be minimized using the weighted sum method are the maximum curvature modulus, the maximum curvature derivative modulus, and the total length of maneuver splines (cf. Problem 2 in Section IV).

The proposed method avoids the difficulties of the two-step procedure of mainstream approaches to autonomous parking because the approximating scheme with $\boldsymbol{\eta}^3$-splines is intrinsically feasible and the flexibility of the multi-optimization framework allows a supervisor governing the autonomous parking to trade off the various quality indexes to adapt to the current parking scenario. A garage parking example illustrating this method is reported with simulation results (cf. Section V).

II. G^3-PATHS AND THE SMOOTH PARKING PROBLEM

We consider an autonomous parking problem for a car-like vehicle. The Cartesian coordinates of the rear-axle middle-point are denoted by x, y and θ is the vehicle orientation angle with respect to the X axis. The distance between the rear-axle and the front-axle is l. With the usual modeling assumptions (no-slippage of the wheels, rigid body, etc.) the following nonlinear kinematic model of the car-like vehicle can be deduced:

$$\begin{cases} \dot{x}(t) = v(t)\ \cos\theta(t) \\ \dot{y}(t) = v(t)\ \sin\theta(t) \\ \dot{\theta}(t) = \frac{1}{l}\, v(t)\ \tan\delta(t) \end{cases} \tag{1}$$

where the vehicle control inputs are $v(t)$ and $\delta(t)$, the velocity of the rear-axle middle-point and the steering angle of the front wheels respectively. The following definition will be used along this paper.

Definition 1: A Cartesian path Γ has third order geometric continuity, and we say Γ is a G^3-path, if it has the continuity of the unit-tangent vector, the curvature, and the derivative of the curvature with respect to the arc length s (the curvilinear abscissa) (for more details see [17], [19]).

In order to obtain a smooth motion control, inputs v and δ must be functions with C^1 continuity, i.e. continuous functions with continuous first derivatives. A connection between smooth inputs and paths of the car-like vehicle is established by the following result.

Proposition 1: Assign any $t_1 > 0$. If a Cartesian path Γ is generated by the car-like vehicle described by system (1), with inputs $v(t), \delta(t) \in C^1([0, t_1])$ where $v(t) \neq 0$ and $|\delta(t)| < \frac{\pi}{2}\ \forall t \in [0, t_1]$, then Γ is a G^3-path. Conversely, given any G^3-path Γ there exist inputs $v(t), \delta(t) \in C^1([0, t_1])$ with $v(t) \neq 0$ and $|\delta(t)| < \frac{\pi}{2}\ \forall t \in (0, t_1)$, and initial conditions such that the path generated by (1) coincides with the given Γ.

Proof: It follows from an analogous result presented in [17] for unicycle mobile robots. ■

Instrumentals to the approach to path planning for the autonomous parking of car-like vehicles are the following concepts of *configuration vector* and corresponding *configuration space* [3].

Definition 2: The coordinate position (considering the middle-point of the rear-axle) and orientation of the vehicle with respect to the Cartesian plane $\{X, Y\}$ and the steering angle δ compose the *configuration vector* as follows:

$$\mathbf{q} \doteq \begin{bmatrix} q_1 \\ q_2 \\ q_3 \\ q_4 \end{bmatrix} \doteq \begin{bmatrix} x \\ y \\ \theta \\ \delta \end{bmatrix} \in \mathcal{Q}\,, \tag{2}$$

where $\mathcal{Q} \doteq \mathbb{R}^2 \times [0, 2\pi[\, \times [-\delta_M,\ +\delta_M]$, is the *configuration space*; herein δ_M is the maximum allowed value of the steering angle.

In the parking scenario, the occupancy area of the car-like vehicle is denoted by $\mathcal{A}$ which is normally a rectangle moving in the Cartesian plane $\{X, Y\}$, referred as the *parking space* $\mathcal{P}$. The car body $\mathcal{A}$ occupies a portion area of $\mathcal{P}$ that depends on the configuration vector $\mathbf{q}$, i.e. $\mathcal{A} = \mathcal{A}(\mathbf{q}) \subset \mathcal{P}$. In the parking space there are also the obstacles $\mathcal{B}_i$, $i = 1, 2, \ldots n$, considered as convex polygons without loss of generality (a non-convex polygon can be always decomposed in two or more convex polygons).

The parking problem can be introduced as a smooth feedforward control problem for model (1), i.e. the problem of devising inputs $v(t), \delta(t) \in C^1$, for which the vehicle starting from a given configuration $\mathbf{q}_s = [x_s\, y_s\, \theta_s\, \delta_s]^T$ reaches an assigned final or goal configuration $\mathbf{q}_g = [x_g\, y_g\, \theta_g\, \delta_g]^T$ while avoiding all the obstacles and satisfying at any time the constraint $|\delta(t)| \leq \delta_M$. The sought feedforward control may admit maneuvers, i.e. changes of sign in the vehicle velocity $v(t)$, so that when the velocity is positive the car performs a forward movement whereas when it is negative we have a car's backward movement.

On the grounds of Proposition 1 and of the (dynamic) path inversion concept introduced in [17], the smooth parking feedforward control problem can be reduced to a purely geometric problem, to be more specific a purely Cartesian G^3-path planning problem followed by a velocity planning on the determined paths. This means determining a sequence of (feasible) G^3-paths $\{\Gamma_1, \Gamma_2, \ldots \Gamma_h\}$ (h is the number of parking paths) that the vehicle can exactly follow by applying feedforward inputs $v(t), \delta(t)$ where $v(t) \in C^1$ can be freely designed with the constraint of having zero velocity and zero acceleration at the the start and at the end of each path Γ_i. The steering input on the path Γ_i can be simply determined by (cf. [17] and [21])

$$\delta(t) = \pm \arctan(l\kappa_i(s))|_{s=\int_{t_i}^{t} v(\xi)d\xi}$$

for a forward (+) or backward (-) movement. Herein $\kappa_i(s)$ is the curvature function of arc length s and t_i is the time instant at the beginning of Γ_i.

In the following, a path Γ to be followed by the vehicle with a forward or backward movement will be denoted by Γ^+ or Γ^- respectively. Therefore, a sequence of path $\{\Gamma_1, \Gamma_2, \ldots \Gamma_h\}$ is actually $\{\Gamma_1^+, \Gamma_2^-, \ldots \Gamma_h^+\}$ or $\{\Gamma_1^-, \Gamma_2^+, \ldots \Gamma_h^-\}$ if h is odd, and $\{\Gamma_1^+, \Gamma_2^-, \ldots \Gamma_h^-\}$ or $\{\Gamma_1^-, \Gamma_2^+, \ldots \Gamma_h^+\}$ if h is even. In the introduced sequence of paths we see an alternation of forward and backward paths, i.e. a forward path Γ_i^+ is followed by a backward Γ_{i+1}^- or viceversa. Any pair of subsequent paths $\{\Gamma_i^+, \Gamma_{i+1}^-\}$ or $\{\Gamma_i^-, \Gamma_{i+1}^+\}$ is made of paths that meet each other at a common Cartesian point corresponding to a configuration vector $\mathbf{q}_i$ ($i = 1, \ldots h-1$) which is still common for the vehicle at the end of path Γ_i and at the start of Γ_{i+1} in case of no steering at standstill, i.e. the case when $\dot{\delta}(t) = 0$ if $v(t) = 0$.

When the vehicle parking problem can be solved without maneuvers we have just one G^3-path Γ_1^+ or Γ_1^- ($h = 1$) to determine and optimize. If no solution can be found with one path because of the obstacles and the limitation given by the maximum steering angle δ_M, a solution may be sought with two chained paths $\{\Gamma_1^+, \Gamma_2^-\}$ or $\{\Gamma_1^-, \Gamma_2^+\}$ ($h = 2$). In this case there is one motion inversion of the vehicle or, in other words, one maneuver to complete the parking task. On the parking space, Γ_1 and Γ_2 meet at a cusp point whose Cartesian coordinates are given by the first two components of configuration vector $\mathbf{q}_1$. When also with $h = 2$ no solution is found we can try with more paths. In Figure 1, the case of two maneuvers ($h = 2$) is depicted.

The G^3-paths Γ_i, $1 = 1, \ldots, h$ composing the sequence $\{\Gamma_1, \Gamma_2, \ldots \Gamma_h\}$ must satisfy specific interpolation conditions at the endpoints of each Γ_i (cf. Section III) in order to guarantee the overall feasibility of the planned paths. In particular considering that the vehicle starts at the given configuration $\mathbf{q}_s = [x_s\, y_s\, \theta_s\, \delta_s]^T$ it follows that the starting point of Γ_1 satisfies: (1) Cartesian coordinates are $(x_s\, y_s)$; (2) direction angle (with respect to the X axis) of the unit-tangent vector is θ_s; (3) scalar curvature κ_s is given by (cf. [17], [22])

$$\kappa_s = \begin{cases} \frac{1}{l} \tan\delta_s & \text{if } \Gamma_1 = \Gamma_1^+ \\ -\frac{1}{l} \tan\delta_s & \text{if } \Gamma_1 = \Gamma_1^-\ ; \end{cases} \tag{3}$$

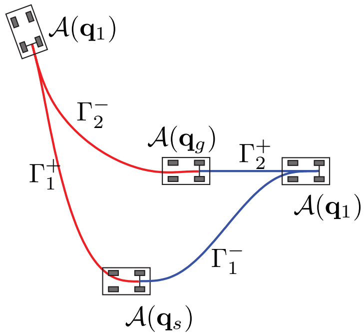

Fig. 1. The two-paths sequences $\{\Gamma_1^+, \Gamma_2^-\}$ and $\{\Gamma_1^-, \Gamma_2^+\}$ for the parking planning.

(4) the derivative of the scalar curvature with respect to the arc length, $\dot{\kappa}_s$ can be freely chosen.

Analogously, the vehicle arrives finally at the goal configuration $\mathbf{q}_g = [x_g \, y_g \, \theta_g \, \delta_g]^T$ for which the endpoint of Γ_h satisfies: (1) Cartesian coordinates are $(x_g \, y_g)$; (2) direction angle of the unit-tangent vector is θ_g; (3) scalar curvature κ_g is given by

$$\kappa_g = \begin{cases} \frac{1}{l} \tan \delta_g & \text{if } \Gamma_h = \Gamma_h^+ \\ -\frac{1}{l} \tan \delta_g & \text{if } \Gamma_h = \Gamma_h^- \, ; \end{cases} \tag{4}$$

(4) the derivative of the scalar curvature with respect to the arc length, $\dot{\kappa}_g$ is a free parameter of the planning problem.

The smooth parking problem considered in this paper can be introduced as follows.

Problem 1: **(Multi-optimization of a sequence of G^3-paths for the smooth parking problem)** Given the number h of paths, consider the space $\mathcal{F}_h$ of all the sequences of G^3-paths $\{\Gamma_1^+, \Gamma_2^-, \ldots \Gamma_h\}$ (or $\{\Gamma_1^-, \Gamma_2^+, \ldots \Gamma_h\}$) such that this sequence (a) is feasible as a whole, i.e. there exist feedforward controls $v(t), \delta(t) \in C^1$ for which the vehicle of model (1) follows the path sequence exactly, and (b) connects the given initial configuration $\mathbf{q}_s$ to the final assigned configuration $\mathbf{q}_g$.

Find the path sequence in $\mathcal{F}_h$ that minimizes the indexes

- the maximum value of the absolute curvature on the h paths,
- the maximum value of the absolute curvature derivative on the h paths, and
- the total length of the h paths $\Gamma_1, \Gamma_2, \ldots \Gamma_h$

subject to the following constraints

1) avoidance of all the obstacles $\mathcal{B}_1, \mathcal{B}_2 \ldots \mathcal{B}_n$ along the paths $\Gamma_1, \Gamma_2, \ldots \Gamma_h$;
2) {maximum value of the absolute curvature on the h paths} $\leq \kappa_M$;
3) {maximum value of the absolute curvature derivative on the h paths}$\leq \dot{\kappa}_M$;
4) avoidance of steering at standstill;

where $\kappa_M = \frac{1}{l} \tan \delta_M$ and $\dot{\kappa}_M$ is a freely chosen bound for the absolute value of the curvature derivative.

Remark 1: It is worth noting the differences among the constraints of Problem 1. Constraints 1) and 2) are *hard* constraints related to obstacle avoidance and maximal steering angle (which is a vehicle's mechanical constraint) respectively whereas constraints 3) and 4) are *soft* constraints related to path smoothness and parking modality respectively. In particular, if steering at standstill is admitted, the fourth constraint, which is considered in this exposition, can be removed without changing the proposed overall approach to the parking problem.

The constrained multi-optimization of Problem 1 is a search in the infinite-dimensional space $\mathcal{F}_h$. In the next section, an approximation scheme based on $\boldsymbol{\eta}^3$-splines will make possible to reduce the search into a finite-dimensional space for which standard parameter optimization can be used.

III. Shaping G^3-paths with $\boldsymbol{\eta}^3$-splines

The $\boldsymbol{\eta}^3$-splines, that were introduced in [18], are an effective tool to approximate Cartesian paths with third-order geometric continuity (G^3-paths, cf. Definition 1). Indeed, they can interpolate a sequence of Cartesian points over which unit-tangent vector, curvature, and curvature derivative can be arbitrarily assigned. A single $\boldsymbol{\eta}^3$-spline is a seventh-order polynomial curve

$$\mathbf{p}(u; \boldsymbol{\eta}) = [p_x(u) \, p_y(u)]^T \, , \; u \in [0, 1] \tag{5}$$

$$p_x(u) = \sum_{i=0}^{7} \alpha_i u^i \; , \;\; p_y(u) = \sum_{i=0}^{7} \beta_i u^i \tag{6}$$

that depends on a six-dimensional vector $\boldsymbol{\eta}$ (the *eta* parameter vector) and interpolates the data vectors $\mathbf{c}_a = [x_a \, y_a \, \theta_a \, \kappa_a \, \dot{\kappa}_a]^T$ and $\mathbf{c}_b = [x_b \, y_b \, \theta_b \, \kappa_b \, \dot{\kappa}_b]^T$, at the curve endpoints $\mathbf{p}(0; \boldsymbol{\eta})$ and $\mathbf{p}(1; \boldsymbol{\eta})$ respectively: $(x_a \, y_a)$ and $(x_b \, y_b)$ are the Cartesian coordinates of the endpoints, θ_a and θ_b are the direction angles of the unit-tangent vectors, κ_a and κ_b are the scalar curvatures, and $\dot{\kappa}_a$ and $\dot{\kappa}_b$ are the derivatives of the scalar curvatures with respect to the arc length. The $\boldsymbol{\eta}$ is a free vector in $\mathbb{R}_+^2 \times \mathbb{R}^4$ that can be used to shape the resulting path while maintaining the interpolation conditions at the endpoints. The complete closed-form expressions of the $\boldsymbol{\eta}^3$-spline are reported in [18] and [19].

In this paper we propose to use a simplified version of the $\boldsymbol{\eta}^3$-spline that only depends on the first two components of vector $\boldsymbol{\eta}$ (actually the most important ones, cf. Section V of [18]) while the remaining components are set to zero. Specifically, in this case $\boldsymbol{\eta}$ is redefined as the two-dimensional vector $[\eta_a \, \eta_b]^T \in \mathbb{R}_+^2$ where its positive components are the mathematical velocities of the curve at the endpoints, i.e. $\eta_a = \|\dot{\mathbf{p}}(0; \boldsymbol{\eta})\|$ and $\eta_b = \|\dot{\mathbf{p}}(1; \boldsymbol{\eta})\|$. The corresponding simplified closed-form expressions of coefficients α_i, β_i, $i = 0, 1, \ldots, 7$ appearing in (5) and (6) are detailed below:

$$\begin{aligned}
&\alpha_0 = x_a, \quad \alpha_1 = \eta_a \cos\theta_a, \\
&\alpha_2 = -\frac{1}{2}\eta_a^2 \kappa_a \sin\theta_a, \quad \alpha_3 = -\frac{1}{6}\eta_a^3 \dot{\kappa}_a \sin\theta_a, \\
&\alpha_4 = 35\,(x_b - x_a) - 20\eta_a \cos\theta_a + \left(5\kappa_a + \frac{2}{3}\eta_a \dot{\kappa}_a\right) \\
&\quad \eta_a^2 \sin\theta_a - 15\eta_b \cos\theta_b - \left(\frac{5}{2}\kappa_b - \frac{1}{6}\eta_b \dot{\kappa}_b\right)\eta_b^2 \sin\theta_b, \\
&\alpha_5 = -84(x_b - x_a) + 45\eta_a \cos\theta_a - (10\kappa_a + \eta_a \dot{\kappa}_a) \\
&\quad \eta_a^2 \sin\theta_a + 39\eta_b \cos\theta_b + \left(7\kappa_b - \frac{1}{2}\eta_b \dot{\kappa}_b\right)\eta_b^2 \sin\theta_b, \\
&\alpha_6 = 70(x_b - x_a) - 36\eta_a \cos\theta_a + \left(\frac{15}{2}\kappa_a + \frac{2}{3}\eta_a \dot{\kappa}_a\right) \\
&\quad \eta_a^2 \sin\theta_a - 34\eta_b \cos\theta_b - \left(\frac{13}{2}\kappa_b - \frac{1}{2}\eta_b \dot{\kappa}_b\right)\eta_b^2 \sin\theta_b,
\end{aligned}$$

$$\alpha_7 = -20(x_b - x_a) + 10\eta_a \cos\theta_a - \left(2\kappa_a + \frac{1}{6}\eta_a\dot{\kappa}_a\right)$$
$$\eta_a^2 \sin\theta_a + 10\eta_b\cos\theta_b + \left(2\kappa_b - \frac{1}{6}\eta_b\dot{\kappa}_b\right)\eta_b^2\sin\theta_b,$$
$$\beta_0 = y_a, \quad \beta_1 = \eta_a \sin\theta_a,$$
$$\beta_2 = \frac{1}{2}\eta_a^2\kappa_a\cos\theta_a, \quad \beta_3 = \frac{1}{6}\eta_a^3\dot{\kappa}_a\cos\theta_a,$$
$$\beta_4 = 35\,(y_b - y_a) - 20\eta_a\sin\theta_a - \left(5\kappa_a + \frac{2}{3}\eta_a\dot{\kappa}_a\right)$$
$$\eta_a^2\cos\theta_a - 15\eta_b\sin\theta_b + \left(\frac{5}{2}\kappa_b - \frac{1}{6}\eta_b\dot{\kappa}_b\right)\eta_b^2\cos\theta_b,$$
$$\beta_5 = -84(y_b - y_a) + 45\eta_a\sin\theta_a + (10\kappa_a + \eta_a\dot{\kappa}_a)$$
$$\eta_a^2\cos\theta_a + 39\eta_b\sin\theta_b - \left(7\kappa_b - \frac{1}{2}\eta_b\dot{\kappa}_b\right)\eta_b^2\cos\theta_b,$$
$$\beta_6 = 70(y_b - y_a) - 36\eta_a\sin\theta_a - \left(\frac{15}{2}\kappa_a + \frac{2}{3}\eta_a\dot{\kappa}_a\right)$$
$$\eta_a^2\cos\theta_a - 34\eta_b\sin\theta_b + \left(\frac{13}{2}\kappa_b - \frac{1}{2}\eta_b\dot{\kappa}_b\right)\eta_b^2\cos\theta_b,$$
$$\beta_7 = -20(y_b - y_a) + 10\eta_a\sin\theta_a + \left(2\kappa_a + \frac{1}{6}\eta_a\dot{\kappa}_a\right)$$
$$\eta_a^2\cos\theta_a + 10\eta_b\sin\theta_b - \left(2\kappa_b - \frac{1}{6}\eta_b\dot{\kappa}_b\right)\eta_b^2\cos\theta_b\,.$$

The infinite-dimensional space $\mathcal{F}_h$ of Problem 1 can be approximated with a finite-dimensional space by using $\boldsymbol{\eta}^3$-splines. Consider an element of $\mathcal{F}_h$, i.e. a sequence of G^3-paths $\{\Gamma_1^+, \Gamma_2^-, \dots, \Gamma_h\}$ (or $\{\Gamma_1^-, \Gamma_2^+, \dots, \Gamma_h\}$), then each Γ_i^+ or Γ_i^-, will be approximated by a single (simplified) $\boldsymbol{\eta}^3$-spline denoted as $\mathbf{p}_i^+(u;\boldsymbol{\eta}_i)$ or $\mathbf{p}_i^-(u;\boldsymbol{\eta}_i)$. Hence, the sequence of $\boldsymbol{\eta}^3$-splines $\{\mathbf{p}_1^+(u;\boldsymbol{\eta}_1), \mathbf{p}_2^-(u;\boldsymbol{\eta}_2), \dots, \mathbf{p}_h(u;\boldsymbol{\eta}_h)\}$ (or $\{\mathbf{p}_1^-(u;\boldsymbol{\eta}_1), \mathbf{p}_2^+(u;\boldsymbol{\eta}_2), \dots, \mathbf{p}_h(u;\boldsymbol{\eta}_h)\}$) will be used to set up the multi-optimization for the parking path planning.

The simplified spline $\mathbf{p}_i(u;\boldsymbol{\eta}_i)$ is defined by the interpolating conditions $\mathbf{c}_{a,i} = [x_{a,i}\, y_{a,i}\, \theta_{a,i}\, \kappa_{a,i}\, \dot{\kappa}_{a,i}]^T$ and $\mathbf{c}_{b,i} = [x_{b,i}\, y_{b,i}\, \theta_{b,i}\, \kappa_{b,i}\, \dot{\kappa}_{b,i}]^T$ at the path endpoints and by the parameter vector $\boldsymbol{\eta}_i = [\eta_{a,i}\, \eta_{b,i}]^T$.

Remark 2: In the proposed approximating scheme, a path Γ_i is actually approximated by $\mathbf{p}_i\,([0,1];\boldsymbol{\eta}_i)$, i.e. the Cartesian image over interval $[0,1]$ of the $\boldsymbol{\eta}^3$-spline curve $\mathbf{p}_i\,(u;\boldsymbol{\eta}_i)$. In the following, to simplify notation the same symbol $\mathbf{p}_i\,(u;\boldsymbol{\eta}_i)$ or even $\mathbf{p}_i$ is used to denote both the parametric curve and the corresponding path (cf. [17]).

The parking sequence of $\boldsymbol{\eta}^3$-splines $\{\mathbf{p}_1, \mathbf{p}_2, \dots, \mathbf{p}_h\}$ can satisfy the conditions (a) and (b) and the constraint 4) of Problem 1 by a proper assignment of the interpolation conditions. These assignments are exemplified below for the cases $h = 1, 2$.

Case $h = 1$ with $\{\mathbf{p}_1^+(u;\boldsymbol{\eta}_1)\}$ (one forward movement of the vehicle): The vehicle starts at configuration $\mathbf{q}_s$ and arrives at configuration $\mathbf{q}_g$ (cf. (3) and (4)). Hence the spline parameters can be set as follows:

$$\mathbf{p}_1^+(u;\boldsymbol{\eta}_1): \begin{cases} \mathbf{c}_{a,1} = \begin{bmatrix} x_s \\ y_s \\ \theta_s \\ \frac{1}{l}\tan\delta_s \\ z_1 \end{bmatrix}, \; \mathbf{c}_{b,1} = \begin{bmatrix} x_g \\ y_g \\ \theta_g \\ \frac{1}{l}\tan\delta_g \\ z_2 \end{bmatrix} \\ \boldsymbol{\eta}_1 = [z_3\, z_4]^T, \end{cases}$$

where $z_1, z_2 \in [-\dot{\kappa}_M, \dot{\kappa}_M]$ and $z_3, z_4 \in \mathbb{R}_+$ indicate the free variables to be optimized. These are packed in the vector $\mathbf{z} = [z_1\, z_2\, z_3\, z_4]^T$ that belongs to the search space $\mathcal{Z} := [-\dot{\kappa}_M, \dot{\kappa}_M]^2 \times \mathbb{R}_+^2$.

h	$\dim(\mathcal{Z})$	$\mathcal{Z}$
1	4	$[-\dot{\kappa}_M, \dot{\kappa}_M]^2 \times \mathbb{R}_+^2$
2	12	$[-\dot{\kappa}_M, \dot{\kappa}_M]^4 \times \mathbb{R}_+^4 \times \mathbb{R}^2 \times$ $[0, 2\pi) \times [-\kappa_M, \kappa_M]$
3	20	$[-\dot{\kappa}_M, \dot{\kappa}_M]^6 \times \mathbb{R}_+^6 \times \mathbb{R}^4 \times$ $[0, 2\pi)^2 \times [-\kappa_M, \kappa_M]^2$
$\vdots$	$\vdots$	$\vdots$
h	$8h-4$	$[-\dot{\kappa}_M, \dot{\kappa}_M]^{2h} \times \mathbb{R}_+^{2h} \times \mathbb{R}^{2(h-1)} \times$ $[0, 2\pi)^{h-1} \times [-\kappa_M, \kappa_M]^{h-1}$

TABLE I
THE DIMENSION AND STRUCTURE OF THE SEARCH SPACE $\mathcal{Z}$.

Case $h = 2$ with $\{\mathbf{p}_1^+(u;\boldsymbol{\eta}_1), \mathbf{p}_2^-(u;\boldsymbol{\eta}_2)\}$ (one forward movement plus a a backward one): All the spline parameters can be set as follows

$$\mathbf{p}_1^+(u;\boldsymbol{\eta}_1): \begin{cases} \mathbf{c}_{a,1} = \begin{bmatrix} x_s \\ y_s \\ \theta_s \\ \frac{1}{l}\tan\delta_s \\ z_1 \end{bmatrix}, \; \mathbf{c}_{b,1} = \begin{bmatrix} z_9 \\ z_{10} \\ z_{11} \\ z_{12} \\ z_2 \end{bmatrix} \\ \boldsymbol{\eta}_1 = [z_5\, z_6]^T, \end{cases}$$

$$\mathbf{p}_2^-(u;\boldsymbol{\eta}_2): \begin{cases} \mathbf{c}_{a,2} = \begin{bmatrix} z_9 \\ z_{10} \\ z_{11} + \pi \\ -z_{12} \\ z_3 \end{bmatrix}, \; \mathbf{c}_{b,2} = \begin{bmatrix} x_g \\ y_g \\ \theta_g + \pi \\ -\frac{1}{l}\tan\delta_g \\ z_4 \end{bmatrix} \\ \boldsymbol{\eta}_2 = [z_7\, z_8]^T, \end{cases}$$

where the free variables are z_i, $i = 1, \dots 12$, and they form the vector $\mathbf{z} \in \mathcal{Z}$ with $\mathcal{Z} := [-\dot{\kappa}_M, \dot{\kappa}_M]^4 \times \mathbb{R}_+^4 \times \mathbb{R}^2 \times [0, 2\pi) \times [-\kappa_M, \kappa_M]$ which is a twelve-dimensional search space.

When $h > 2$, the spline parameters can be set up similarly as in the presented cases. Table I reports the dimension and structure of the the search space $\mathcal{Z}$ as a function of h. In particular, when the parking is done with h splines, the dimension of the search space is $8h - 4$: every added spline increases of 8 the dimension of $\mathcal{Z}$.

Remark 3: The proposed approximation scheme replaces each path Γ_i of sequence $\{\Gamma_1, \Gamma_2, \dots, \Gamma_h\}$ with only one $\boldsymbol{\eta}^3$-spline to avoid excessive increasing of the dimension of the search space $\mathcal{Z}$. Yet, it would be possible within the same proposed framework to improve the approximation by using two or more $\boldsymbol{\eta}^3$-splines for each Γ_i.

IV. SETTING UP THE MULTI-OPTIMIZATION

In this section the multi-optimization of Problem 1 is dealt with the substitution of the infinite-dimensional space $\mathcal{F}_h$ with the finite-dimensional parameter space $\mathcal{Z}$ introduced in the previous section. This corresponds to do the searching for multi-optimization on the sequences of simplified $\boldsymbol{\eta}^3$-splines $\{\mathbf{p}_1(u;\boldsymbol{\eta}_1), \mathbf{p}_2(u;\boldsymbol{\eta}_2), \dots, \mathbf{p}_h(u;\boldsymbol{\eta}_h)\}$ instead of the sequences of G^3-paths introduced in section II.

The three indexes to be minimized using the standard weighted sum method [20] are (cf. Problem 1): the maximum value of the curvature modulus on the h splines, the maximum value of the absolute value of the curvature derivative (with respect to the arc length) on the h splines, and the total length of the h splines. These indexes are respectively denoted by κ_{max}, $\dot{\kappa}_{max}$, and s_{tot} and depend on the parameter vector $\mathbf{z} \in \mathcal{Z}$. They can be determined as follows (the dependencies on $\mathbf{z}$ are omitted for simplicity and $\mathbf{p}_i(u; \boldsymbol{\eta}_i) \equiv [p_{x,i}(u)\, p_{y,i}(u)]^T$, $i = 1, \ldots, h$, cf. (5)):

$$\kappa_{max} \doteq \max_{i=1,\ldots h} \kappa_{max,i} \tag{7}$$

where $(i = 1, \ldots, h)$ $\kappa_{max,i} \doteq \max_{u \in [0,1]} |\kappa_i(u)|$ and

$$\kappa_i(u) = \frac{\dot{p}_{x,i}(u)\ddot{p}_{y,i}(u) - \ddot{p}_{x,i}(u)\dot{p}_{y,i}(u)}{(\dot{p}_{x,i}^2(u) + \dot{p}_{y,i}^2(u))^{\frac{3}{2}}}$$

is the scalar curvature of spline $\mathbf{p}_i(u; \boldsymbol{\eta}_i)$;

$$\dot{\kappa}_{max} \doteq \max_{i=1,\ldots h} \dot{\kappa}_{max,i} \tag{8}$$

where $(i = 1, \ldots, h)$ $\dot{\kappa}_{max,i} \doteq \max_{u \in [0,1]} \left| \frac{d\kappa_i}{ds}(u) \right|$ and

$$\frac{d\kappa_i}{ds}(u) = \frac{\dot{p}_{x,i}\dddot{p}_{y,i} - \dddot{p}_{x,i}\dot{p}_{y,i}}{(\dot{p}_{x,i}^2 + \dot{p}_{y,i}^2)^2} - 3\frac{(\dot{p}_{x,i}\ddot{p}_{y,i} - \ddot{p}_{x,i}\dot{p}_{y,i})(\dot{p}_{x,i}\ddot{p}_{x,i} + \dot{p}_{y,i}\ddot{p}_{y,i})}{(\dot{p}_{x,i}^2 + \dot{p}_{y,i}^2)^3}$$

is the derivative of the curvature of spline $\mathbf{p}_i(u; \boldsymbol{\eta}_i)$ with respect to the arc length (for brevity the dependency on u is omitted in the right side of the above relation);

$$s_{tot} \doteq \sum_{i}^{h} s_{tot,i} \tag{9}$$

where $s_{tot,i} \doteq \int_0^1 [\dot{p}_{x,i}^2(\xi) + \dot{p}_{y,i}^2(\xi)]^{1/2} d\xi$.

The constraint of obstacle avoidance is dealt with the concept of *occupancy span* of the vehicle along a path planning:

Definition 3: The occupancy span of the vehicle along the spline sequence $\{\mathbf{p}_1, \mathbf{p}_2, \ldots, \mathbf{p}_h\}$ is the set defined as $\mathcal{S} \doteq \bigcup_{i=1}^{n} \mathcal{S}_i$ where

$$\mathcal{S}_i \doteq \{\boldsymbol{p} \in \mathcal{P} : \boldsymbol{p} \in \mathcal{A}(\boldsymbol{q}), q_1 = p_{x,i}(u),\, q_2 = p_{y,i}(u),\, q_3 = \arg(\dot{p}_{x,i}(u) + j\dot{p}_{y,i}(u)), u \in [0,1]\} . \tag{10}$$

Note that the occupancy span depends on $\mathbf{z} \in \mathcal{Z}$, i.e. $\mathcal{S} \equiv \mathcal{S}(\mathbf{z})$. Define the *obstacle region* $\mathcal{O}$ as the union of all the obstacles, i.e. $\mathcal{O} \doteq \cup_{i=1}^{n} \mathcal{B}_i$ and the vehicle avoids all the obstacles along a path planning if and only if the intersection of $\mathcal{S}(\mathbf{z})$ and $\mathcal{O}$ is the empty set (cf. constraint (12) below).

Now the nonlinear constrained multiobjective optimization problem for the geometric planning of autonomous parking can be stated as follows:

Problem 2: **(Multi-optimization of a sequence of $\boldsymbol{\eta}^3$-splines for the smooth parking problem)** Given the number h of paths, consider the parameter space $\mathcal{Z}$ that defines the sequences $\{\mathbf{p}_1^+, \mathbf{p}_2^-, \ldots, \mathbf{p}_h\}$ (or $\{\mathbf{p}_1^-, \mathbf{p}_2^+, \ldots, \mathbf{p}_h\}$) according to the interpolating scheme exposed in Section III. Then, the posed problem is ($\lambda_1, \lambda_2, \lambda_3 \geq 0$ and $\lambda_1 + \lambda_2 + \lambda_3 = 1$):

$$\min_{\mathbf{z} \in \mathcal{Z}} \lambda_1 \kappa_{max}(\mathbf{z}) + \lambda_2 \dot{\kappa}_{max}(\mathbf{z}) + \lambda_3 s_{tot}(\mathbf{z}) \tag{11}$$

subject to

$$\mathcal{S}(\mathbf{z}) \cap \mathcal{O} = \varnothing , \tag{12}$$

$$\kappa_{max}(\mathbf{z}) \leq \kappa_M , \tag{13}$$

$$\dot{\kappa}_{max}(\mathbf{z}) \leq \dot{\kappa}_M . \tag{14}$$

The coefficients λ_1, λ_2, and λ_3 of the composite index to be minimized in (11) can be freely chosen in order to weight the smoothness of the resulting maneuver paths (which is related to low values of both κ_{max} and $\dot{\kappa}_{max}$) versus the minimization of s_{tot}, the total length of the parking paths.

Remark 4: Note that the possible constraint of avoiding steering at vehicle's standstill does not appear in the constraints (13)-(14) because it is plainly enforced by proper assignment of the geometric interpolating conditions on the $\boldsymbol{\eta}^3$-splines (cf. Section III).

Obstacle avoidance constraint (12) can be equivalently reduced to an equality constraint by computing the *maximal collision area* of the vehicle along the spline sequence:

$$\text{mca} \doteq \max_{i=1,\ldots h} \text{mca}_i , \tag{15}$$

$$\text{mca}_i \doteq \max_{u \in [0,1]} \{\text{area}\,(\mathcal{A}(\boldsymbol{q}) \cap \mathcal{O}) : q_1 = p_{x,i}(u),\; q_2 = p_{y,i}(u),\, q_3 = \arg(\dot{p}_{x,i}(u) + j\dot{p}_{y,i}(u))\} .$$

Constraint (12) is therefore equivalent to $\text{mca}(\boldsymbol{z}) = 0$ and in such a way Problem 2 becomes a constrained minimization problem for which a standard penalty method [23] can take into account all the constraints so as to reduce the whole multi-optimization to the minimization of just one index. In a real-time scenario for autonomous parking, fast local minimization algorithms can be then implemented to solve Problem 2 provided that the following data is readily available: (1) the number h of splines; (2) the maneuver sequence to prefer: $\{\mathbf{p}_1^+, \mathbf{p}_2^-, \ldots, \mathbf{p}_h\}$ or $\{\mathbf{p}_1^-, \mathbf{p}_2^+, \ldots, \mathbf{p}_h\}$; (3) an initial estimate of the parameter vector $\boldsymbol{z}$. Reasonably, this data can be determined by using look-up tables that can be constructed off-line by extensive optimizations such as those based on methods of stochastic global multi-objective optimization [24].

V. Simulation results

An example of *garage parking* maneuver in a constrained environment is considered for a standard compact car with wheelbase and maximum steering angle of the front wheels $l = 2.3$ m and $\delta_M = 0.464$ rad. Hence, the maximum curvature of the car paths is $\kappa_M = \frac{1}{l}\tan\delta_M = 0.218$ m^{-1} (cf. Problem 1 in Section II). The allowed maximum absolute value of the curvature derivative with respect to the arc length is chosen as $\dot{\kappa}_M = 2.50$ m^{-2}. The origin of the Cartesian plane $\mathcal{P}$ is chosen to be inside the parking lot that the car has to reach. The car has start configuration $\mathbf{q}_s = [x_s\ y_s\ \theta_s\ \delta_s]^T = [7\ \ -6\ \ 3\pi/4\ \ 0]^T$ and the final goal configuration, which corresponds to a *front car parking* mode (i.e. the car can only reach the goal configuration with a forward final motion because of the surrounding obstacles, cf. Figures 2), is $\mathbf{q}_g = [x_g\ y_g\ \theta_g\ \delta_g]^T = [0.7\ 0\ \pi\ 0]^T$. The multi-optimizations for solving this parking problem are set up with weights $\lambda_1 = 0.5$, $\lambda_2 = 0.2$, and $\lambda_3 = 0.3$. All the possible spline sequences to be considered up to three splines are the following (the arguments of the $\boldsymbol{\eta}^3$-splines are omitted for compactness):

$$\begin{aligned} h = 1 &: \quad \{\mathbf{p}_1^+\},\ \{\mathbf{p}_1^-\}; \\ h = 2 &: \quad \{\mathbf{p}_1^+,\, \mathbf{p}_2^-\},\ \{\mathbf{p}_1^-,\, \mathbf{p}_2^+\}; \\ h = 3 &: \quad \{\mathbf{p}_1^+,\, \mathbf{p}_2^-,\, \mathbf{p}_3^+\}, \{\mathbf{p}_1^-,\, \mathbf{p}_2^+,\, \mathbf{p}_3^-\}. \end{aligned}$$

The sequences $\{\mathbf{p}_1^-\}$, $\{\mathbf{p}_1^+, \mathbf{p}_2^-\}$, $\{\mathbf{p}_1^-, \mathbf{p}_2^+, \mathbf{p}_3^-\}$ have to be discarded due to the fact that the last spline has to be covered with a car's forward movement (front car parking). Hence the topologically possible sequences are: $\{\mathbf{p}_1^+\}$, $\{\mathbf{p}_1^-, \mathbf{p}_2^+\}$, $\{\mathbf{p}_1^+, \mathbf{p}_2^-, \mathbf{p}_3^+\}$.

Parking with $\{\mathbf{p}_1^+\}$ is not feasible because the multi-optimization (11) fails to satisfy all the required constraints (12)-(14). Instead, both sequences $\{\mathbf{p}_1^-, \mathbf{p}_2^+\}$ and $\{\mathbf{p}_1^+, \mathbf{p}_2^-, \mathbf{p}_3^+\}$ lead to feasible parking maneuvers. For brevity, only the results for the two splines maneuver are reported. The multi-optimization of $\{\mathbf{p}_1^-, \mathbf{p}_2^+\}$ leads to a Pareto optimal solution $\mathbf{z}^* \in \mathcal{Z} = [-2.5, 2.5]^4 \times \mathbb{R}_+^4 \times \mathbb{R}^2 \times [0, 2\pi) \times [-0.218, 0.218]$ for which $\kappa_{max}(\mathbf{z}^*) = 0.143\ \mathrm{m}^{-1}$, $\dot{\kappa}_{max}(\mathbf{z}^*) = 0.260\ \mathrm{m}^{-2}$, $s_{tot}(\mathbf{z}^*) = 22.8$ m. This solution is

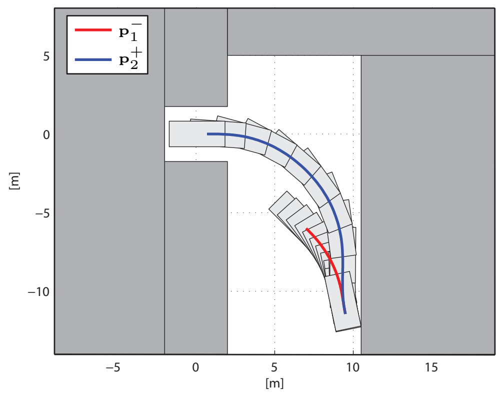

Fig. 2. Optimal parking with the two-spline maneuver $\{\mathbf{p}_1^-, \mathbf{p}_2^+\}$.

depicted with graphic simulation in Fig. 2. Plots of curvature and curvature derivative are reported in Fig. 3.

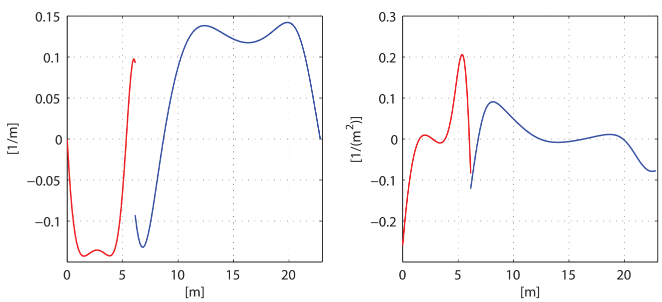

Fig. 3. Plots of curvature and curvature derivative as functions of the arc length along the entire optimal spline maneuver $\{\mathbf{p}_1^-, \mathbf{p}_2^+\}$.

The optimizations and simulations have been performed using MATLAB on a PC with an Intel Dual Core 2.8GHz processor. The computational times of the multi-optimization (11)-(14) depends on the number h of maneuvers splines mainly, but also on the chosen discretization for the spline parameters $u \in [0, 1]$, used for computing the parking quality indexes (7)-(9) and the maximal collision area (15). For example, in our case with $h = 2$, by discretizing interval $[0, 1]$ into 100 steps, the computation time is 2.5 seconds. Significant improvements on computation times are expected by abandoning MATLAB in favor of more efficient ad hoc optimization routines.

VI. Conclusion

This paper has presented a multi-optimization approach for the autonomous parking of car-like vehicles that is based on the $\boldsymbol{\eta}^3$-spline, a polynomial curve primitive devised to interpolate Cartesian points with associated unit-tangent vectors, curvatures, and curvature derivatives [18], [19]. This approach applies the multi-optimization over sequences of splines that are intrinsically feasible and in such a way it speeds up the search of a parking maneuver that avoids obstacles and satisfies the mechanical bound on the maximal steering angle. Moreover, the smoothness of the parking paths can be as desired by putting a selectable limit on the maximal curvature derivative. Simulation results are promising and with implementation improvements on optimization routines, the presented method can be applied in real-life parking scenarios.

References

[1] A. Broggi, A. Zelinsky, M. Parent, and C. Thorpe, "Intelligent vehicles," in *Springer Handbook of Robotics*, B. Siciliano and O. Khatib, Eds. Berlin: Springer, 2008, pp. 1175–1197.

[2] J.-C. Latombe, *Robot motion planning*. Kluwer, 1991.

[3] S. LaValle, *Planning Algorithms*. Cambridge University Press, 2006.

[4] J.-P. Laumond, P. Jacobs, M. Täix, and R. Murray, "A motion planner for nonholonomic mobile robots," *IEEE Transactions on Robotics and Automation*, vol. 10, no. 5, pp. 577–593, October 1994.

[5] A. Divelbiss and J.T.Wen, "A path space approach to nonholonomic motion planning in the presence of obstacles," *IEEE Transaction on Robotics and Automation*, vol. 13, no. 3, pp. 443–651, June 1997.

[6] Y. Chitour, "A continuation method for motion-planning problems," *ESAIM: Control, Optimisation and Calculus of Variations*, vol. 12, no. 1, pp. 139–168, 2006.

[7] B. Müller, J. Deutscher, and S. Grodde, "Continuous curvature trajectory design and feedforward control for parking a car," *IEEE Transactions on Control Systems Technology*, vol. 15, no. 3, pp. 541–553, May 2007.

[8] J. Minguez, F. Lamiraux, and J. Laumond, "Motion planning and obstacle avoidance," in *Springer Handbook of Robotics*, B. Siciliano and O. Khatib, Eds. Springer, 2008, pp. 827–852.

[9] I. Paromtchik and C. Laugier, "Motion generation and control for parking an autonomous vehicle," in *Proc. of the 1996 IEEE Int. Conf. on Robotics and Automation*, Minneapolis, Minnesota, April 1996, pp. 3117–3122.

[10] M. Wada, K. Yoon, and H. Hashimoto, "Development of advanced parking assistance system," *IEEE Transactions on Industrial Electronics*, vol. 50, no. 1, pp. 4–17, February 2003.

[11] D. Gorinevsky, A. Kapitanovsky, and A. Goldenberg, "Neural network architecture for trajectory generation and control of automated car parking," *IEEE Transactions on Control Systems Technology*, vol. 4, no. 1, pp. 50–56, January 1996.

[12] K. Demirli and M. Khoshnejad, "Autonomous parallel parking of a car-like mobile robot by a neuro-fuzzy sensor-based controller," *Fuzzy Sets and Systems*, vol. 160, no. 19, pp. 2876–2891, 2009.

[13] M. Sugeno and K. Murakami, "Fuzzy parking control of a model car," in *Proceedings of the IEEE 23rd Conference on Decision and Control*, Las Vegas, NV, December 1984, pp. 902–903.

[14] F. Cuesta, F. Gómez-Bravo, and A. Ollero, "Parking maneuvers of industrial-like electrical vehicles with and without trailer," *Industrial Electronics, IEEE Transactions on*, vol. 51, no. 2, pp. 257–269, April 2004.

[15] F. Cuesta and A. Ollero, *Intelligent Mobile Robot Navigation*. Springer, 2005.

[16] F. Gómez-Bravo, F. Cuesta, A. Ollero, and A. Viguria, "Continuous curvature path generation based on [beta]-spline curves for parking manoeuvres," *Robotics and Autonomous Systems*, vol. 56, no. 4, pp. 360 – 372, 2008.

[17] C. Guarino Lo Bianco, A. Piazzi, and M. Romano, "Smooth motion generation for unicycle mobile robots via dynamic path inversion," *IEEE Trans. on Robotics*, vol. 20, no. 5, pp. 884–891, October 2004.

[18] A. Piazzi, C. Guarino Lo Bianco, and M. Romano, "$\boldsymbol{\eta}^3$-splines for the smooth path generation of wheeled mobile robots," *IEEE Transactions on Robotics*, vol. 23, no. 5, pp. 1089–1095, October 2007.

[19] ——, "Smooth path generation for wheeled mobile robots using $\boldsymbol{\eta}^3$-splines," in *Motion Control*, F. Casolo, Ed. InTech www.intechopen.com, 2010, ch. 4, pp. 75–96.

[20] K. Miettinen, *Nonlinear Multiobjective Optimization*. Kluwer Academic Publishers, 1998.

[21] A. Piazzi, C. Guarino Lo Bianco, M. Bertozzi, A. Fascioli, and A. Broggi, "Quintic G^2-splines for the iterative steering of vision-based autonomous vehicles," *IEEE Trans. on Intelligent Transportations Systems*, vol. 3, no. 1, pp. 27–36, March 2002.

[22] G. Lini and A. Piazzi, "Time-optimal dynamic path inversion for an automatic guided vehicle," in *Proceedings of the 49th IEEE Conference on Decision and Control*, Atlanta, GA, December 2010, pp. 5264–5269.

[23] D. Luenberger, *Linear and Nonlinear Programming*, 2nd ed. Reading, Massachusetts: Addison–Wesley, 1989.

[24] K. Deb, *Multi-Objective Optimization using Evolutionary Algorithms*. New York: Wiley, 2001.