Automatic parallel parkinkg maneuver with geometric continuous-curvature and trajectory regenation

Hélène Vorobieva, Nicoleta Minoiu-Enache, Sébastien Glaser, Saïd Mammar

Automatic Parallel Parkinkg Maneuver with Geometric Continuous-Curvature and Trajectory Regenation

Hélène Vorobieva [1)] Nicoleta Minoiu-Enache [2)] Sébastien Glaser [3)] Saïd Mammar [4)]

1)-2) Renault
1 avenue du Golf, 78084 Guyancourt, France (E-mail: hvorobieva@gmail.com and nicoleta.minoiu-enache@renault.com))
3) IFSTTAR, LIVIC
77 rue des Chantiers 78000 Versailles, France (E-mail: sebastien.glaser@ifsttar.fr)
4) IBISC, Université d'Evry Val d'Essonne
91025 Evry, France (E-mail: said.mammar@ibisc.univ-evry.fr)

ABSTRACT: This paper presents a trajectory generation method for the automatic parallel parking. In the first place, a continuous-curvature path satisfying geometric constraints is generated. During the automated tracking of the trajectory, following errors might appear. Then, if necessary, a new trajectory is generated to correct these errors. The regeneration method presents the same advantages as for the initial path generation (continuous curvature) and allows correcting the deviation of the vehicle. Moreover, the complete functional architecture including the path generation is presented in this article to illustrate the execution of the parking maneuver.

KEY WORDS: **electronics and control,** autonomous driving system, parking assist system / path planning [E1]

1. Introduction

Since parking spots have become very narrow in big cities, maneuvering the vehicle can be difficult and stressful even for experimented drivers. Therefore, automatic parking can increase the comfort and the safety of the driver. In this contribution, we consider the path generation for the automatic parallel parking.

The methods to generate the trajectory for the parking problem can be divided into two groups: geometric methods with circle arcs and continuous-curvature methods.

The geometric methods involve easy geometrical equations and create trajectories based on admissible collision-free circular arcs, which lead the vehicle in the parking. Some methods have only been presented for the parking in one maneuver as in [(1)] and [(2)]. Other methods allow the parking in several maneuvers as in [(3)], where a forbidden zone has been defined and the car can park traveling outside it. Another approach is based on retrieving a vehicle from the parking spot and reversing this procedure. This approach is particularly instinctive and adapted to the parking problems [(4)] and [(5)]. However, the curvature of path based on circle arcs is discontinuous: the vehicle has to stop to reorient its front wheels. This is particularly demanding for the steering column, induces a faster wear of the tires and leads to an undesirable delay.

Continuous-curvature methods overcome these problems. Some of these methods are based on two phases path planning. First, a collision-free path is created and then subdivided to create an admissible path. For example, [(6)] uses clothoidal curves for this purpose. However, these methods often need multiple optimizations of the created path when applied to the parking problem and a possible high computational cost. Moreover, the optimal solution is not always guaranteed.

To reduce the complexity and create a path adapted to the parking problem with low computational cost, we recently have combined the simplicity of the geometric path planning with the advantages of a continuous- curvature path by using clothoids [(7)]. The resulted curve answers the constraints of admissibility for a car-like vehicle and describes very well its behavior. Thus, generally, a simple open-loop control provides a satisfactory path following. However, experiments showed that for parking spots where multiple maneuvers are necessary, the accumulation of the errors with respect to the desired pose might induce significant deviations and prevent the vehicle from parking correctly. For these cases the trajectory regeneration can be a solution in order to correct the deviations.

In the next section, we present a brief state of the art of the sensors and the actuators and explain why with these technologies open loop parking is not always satisfactory. In the section 3, the path planning method with continuous curvature is presented. In the sections 4, the control of the steering angle and the control of the longitudinal speed are outlined. The section 5 details the regeneration of the path and gives a simulation example. In the section 6, we present the full sequence of the parking maneuver with the regeneration of the trajectory. Finally, the section 7 is devoted to the conclusions.

2. Why Trajectory Regeneration is Necessary

2.1. State of the art of the sensors for the parking

For the autonomous parking the vehicle needs to know the size of the parking, the position of the obstacles and its own pose on the road (exteroceptive information). It also should be able to measure the travelled distance when it parks (proprioceptive information).

Modern cars are often equipped with an ABS (Anti-lock Braking System), which measures the speed of the wheels at each time step. Moreover, the steering angle is also generally available via a steering angle sensor or by using an EPS (Electric Power Steering) internal estimator. This information can be used to calculate the travelled distance, as in [(10)].

However, the vehicle has also to be equipped with other sensors to have information about its environment and to correct the drift of the odometry. There are a lot of sensors which provide exteroceptive information, but we will list only some of them more appropriate for a mass-produced autonomous parking system.

The ultrasound sensors are very popular for the parking problem [11] [12] for their simplicity of use and low price. However, they have a blind zone (some centimeters), which prevent them to detect very close obstacles. Moreover, lane markings cannot be detected and that is why only spots defined by surrounding obstacles can be detected.

Cameras present the advantage to detect lane marking and with stereovision (or with a mono-camera when the vehicle is moving) it is also possible to calculate the distance to the obstacles. However, difficulties can be encountered for the information proceeding and in low light conditions.

Modern cars are also often equipped with anti-collision systems which use radars or LIDARs. These sensors could be used for the parking problem only for the beginning of the maneuver as they cannot detect close obstacles (for example, for Continental the anti-collision systems don't work below 1 m [13]).

A good solution to minimize the errors is to have different sort of sensors and proceed to a fusion of the information, for example with a camera and 16 ultrasound sensors as in [14].

2.2. State of the art of the actuators for the parking

The control of the actuators to achieve the desired steering angle and vehicle speed is a difficult task for low parking speeds.

Let consider first the steering column and the steering actuation. This is carried out mainly by controlling the electric machine installed on the steering column or on the rack and used for the driver assistance. The steering angle error is transformed into a steering torque for the electric machine. For low speeds and high angles, the nonlinear behavior of the steering system and of the tires is difficult to neglect and the steering control law has to deal with it. Moreover, the limitation of the steering dynamics imposed for the drivers' safety when a driver is in the car has to be taken into consideration as in [16] .

The automatic longitudinal control of the vehicle for low speeds is carried out today mainly for vehicles with thermal engine and automatic transmission or for electric vehicles. For a thermal engine one difficulty is to manage very low throttle inputs for low accelerations that are comfortable for the passengers. This operation is easier for an electric engine. Either the type of engine, a problem arises when controlling the brake, since the small brake differentials are also hard to obtain. Moreover, designing an appropriate switching between the throttle and the brake remains challenging. Studies of different control approaches for automatic speed control at low speeds are given in [18] and [19].

Moreover the lateral and longitudinal vehicle control can be impacted by the tires deformation. The hypothesis of rolling without slipping might not be completely respected on all surfaces, especially for the driven wheels. A comparison of the kinematic vehicle model with a dynamic vehicle model for a low speed trajectory tracking maneuver is given in [17].

2.3. Trajectory regeneration necessity

Having the pose of the vehicle and its environment, it is possible to generate a path and the associated control signals for the automatic parking. However, as previously shown, the use of different sensors, even with a data fusion, cannot provide a perfect detection of the environment and a perfect pose of the vehicle. The resulting detection errors could be corrected during the maneuvers if the environment is better detected when the configuration of the vehicle changes. However, the initial trajectory might not be valid anymore. Moreover, the environment can change during the maneuver (apparition or disappearance of an obstacle, for instance). In that case, the initially generated path is no more valid. In addition, as previously outlined, the actuators control is challenging for low speeds and this leads to deviations from the initially generated parking path.

To solve these problems, two solutions exist. The first one is to have a closed loop system for each time step that ensures a good trajectory tracking. This approach does not solve however the problem of changing environment. The second approach is to regenerate the path at some key points of the trajectory. In the sequel, we present our path planning method, enhanced by the regeneration method to correct the trajectory deviations.

3. Initial Path Planning

3.1. Geometric path planning with circle arcs

The parking maneuver is a low-speed movement. Consequently, the Ackerman steering is considered with the four wheels rolling, without slipping around the instantaneous center of rotation. Different turning radiuses are calculated. For example, with E being respectively the center of the rear track, $R_E=a/\tan\delta$. The other radiuses can be deduced geometrically. The notations used for the vehicle are presented in Fig. 1 and Fig. 2 and we denote $R_{min}=R_{Emin}$ the radius associated with the maximal steering.

First a geometric path is created with the reversed method [5]. Then the trajectory will be processed to obtain a continuous curvature path. The vehicle is considered in the parking spot, and a retrieving path composed by circle arcs representing forward and backward moves of the vehicle is defined and reversed. A forward move means steering at maximum toward the outside of the spot and then moving forward until approaching the front obstacle. A backward move means steering at maximum toward the inside of the spot and then moving backward until approaching the rear or lated obstacle. When a forward move allows the vehicle to retrieve without collision, the last two circle arcs connecting the real initial pose of the vehicle are calculated (Fig. 2).

3.2. Clothoid turns

In this section, we briefly describe the general properties of clothoids applied to the path planning for vehicles [8]. A clothoid is a curve whose curvature $\kappa=1/R$ varies linearly with its arc length L and that is defined by its parameter A with $A^2=RL$.

Without loss of generality, consider the start configuration of the vehicle $q_i=(x_i,y_i,\psi_i,\delta_i)=(0,0,0,0)$. A vehicle moving with constant positive longitudinal and steering velocity is describing a clothoid whose parameter A depends on these velocities. The configuration for any position q of the vehicle at a distance L from the initial configuration is then (Fig. 3):

$$q=\left(x=A\sqrt{\pi}C_f\left(\frac{L}{\pi A}\right),\quad y=A\sqrt{\pi}S_f\left(\frac{L}{\pi A}\right),\quad \psi=A^2/(2R^2),\quad \delta=1/R\right) \tag{1}$$

where C_f and S_f are the Fresnel integrals:

$$C_f(x)=\int_0^x \cos\frac{\pi}{2}u^2du \quad \text{and} \quad S_f(x)=\int_0^x \sin\frac{\pi}{2}u^2du \tag{2}$$

The center of the circular arc C, located at a distance R from a configuration q, in the direction normal to ψ is:

$$x_C=x_R-R\sin\psi \quad , \quad y_C=y_R+R\cos\psi \tag{3}$$

In addition, we define the radius R_1 and the angle μ between the orientation of q_i and the tangent to the circle of center C and radius R_1:

$$R_1=\sqrt{x_C{}^2+y_C{}^2} \quad , \quad \mu=\arctan x_C/y_C \tag{4}$$

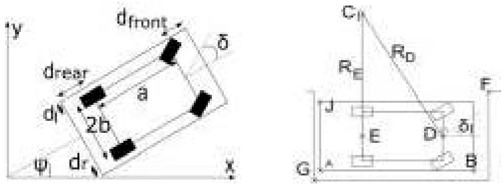

Fig. 1 Vehicle in global (x,y,ψ)-coordinates (left) and in a parking spot (right)

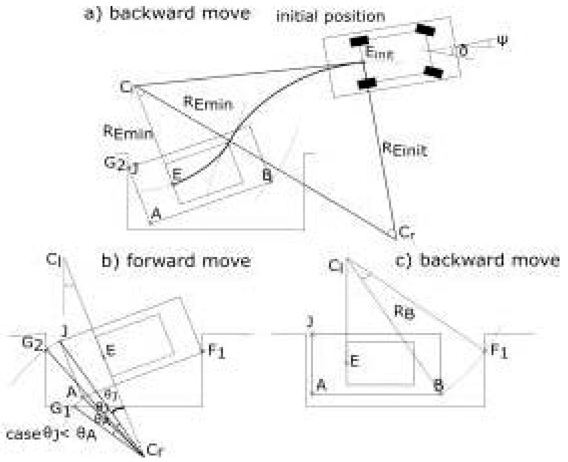

Fig. 2 Example of geometric parking in 3 maneuvers

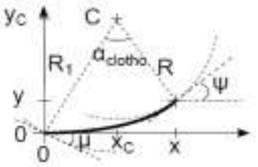

Fig. 3 Clothoid turns

Then we define a clothoidal sequence CAC(A,L,θ):

1) clothoid of parameter A, length L, initial null curvature and final curvature equal to $1/R$;

2) optional circle arc of radius R and angle θ;

3) clothoid of parameter A, length L, initial curvature equal to $1/R$ and final null curvature.

To make the most of the maximum steering angles of the vehicle, we want the radius R_1 to be as small as possible. The minimum time to steer from null steering to maximum steering is $t_{min}=\delta_{max}/v_{\delta}$, where v_{δ} is the maximum desired steering velocity. The minimum associated length of the path is then defined as $L_{min}=v_{longi}t_{min}$, where v_{longi} is the maximum desired longitudinal velocity. The parameter of the used clothoid is then $A_{min}^2=R_{min}L_{min}$ and R_1 is calculated using Eq. 1 - 4. Let's also call α_{clotho} the angle formed by a clothoid started with a null curvature and finished with a curvature $1/R_{min}$.

3.3. From circle arcs to a continuous-curvature path

As previously mentioned, the first step of the parking path generation is to create a geometric path composed of circle arcs as in [5]. In order to prepare the transformation of the generated path in a continuous curvature path, the radius R_1 is used instead of the usual radius R_{min}. Another modification concerns C the instantaneous center of rotation. In the geometrical methods, it is located on the same line with the rear track. Here, in order to use clothoids, the orientation of the vehicle situated on the circles of the geometric path always presents an angle μ with the tangent to the circle in E, so C is transformed by the rotation of center E and angle μ (Fig. 4).

These two modifications effectuated, the second step is to transform each circle arc of angle α_{tot} of the geometric method into a sequence CAC. Three cases are classically considered [8] (see Fig. 5 for the cases A and B).

Case A) $\alpha_{tot} \geq 2\alpha_{clotho}$: We use the sequence CAC(A_{min},L_{min}, α_{tot}-$2\alpha_{clotho}$).

Case B) $2\alpha_{clotho} \geq \alpha_{tot} \geq 2\mu$: We use the sequence CAC(A_{new},L_{new},0), with:

$$A_{new}=\sqrt{\frac{R_1^2\sin(\frac{\beta}{2}+\mu)^2}{\pi\left(\cos\frac{\beta}{2}C_f\left(\sqrt{\beta/\pi}\right)+\sin\frac{\beta}{2}S_f\left(\sqrt{\beta/\pi}\right)\right)^2}} \qquad (5)$$

$$L_{new}=A_{new}\sqrt{\beta} \qquad (6)$$

Where $\beta=(\psi_{end}-\psi_{start}) \bmod 2\pi$ is the deflection of the turn.

Case C) $\alpha_{tot} < 2\mu$: In this case, it is no more possible to use the method of the case B. Nevertheless, a left steering is still possible. A clothoid of parameter A_{min} and of length L_{caseC} is calculated to satisfy the following criteria: $\alpha_{tot} = 2\alpha_{clothocaseC}$, and at the end position it is no more possible to go forward (or backward, depending on the case) without a collision. As this calculus includes parametric curves with Fresnel integrals, an approached solution is considered. Finally, the sequence CAC(A_{min},L_{caseC},0) is used.

The last step is to calculate the starting point of the first circle arc. Indeed, this starting point is different with respect to the starting point calculated with the geometric method based on circle arcs. To minimize lateral offset, the vehicle has to go backward or forward without steering from its initial position E_{init} until it arrives at a point $E_{correct}$ on a circle of radius R_1 connected with a tangential point to the second circle (Fig. 4).

4. Control Steering and Speed Signals

Open-loop control signals of the steering angle δ and the longitudinal velocity v are generated in real time in function of the traveled distance d. The speed control purpose is that the vehicle accelerates to a desired speed and then stays at this speed to finally decelerate and stop when the vehicle has traveled the total distance of a forward move or a backward move. This control signal could be a trapezoid signal. The steering control signal for each sequence CAC(A,L,θ) is:

$$\delta(d)=\begin{cases} k\left|\arctan(a/R_{d1})\right| & d\in[0,L] \\ k\left|\arctan(a/R_{\min})\right| & d\in\left]L,L+L_{arc}\right[\\ k\left|\arctan(a/R_{d2})\right| & d\in\left[L+L_{arc},L_{CAC}\right] \end{cases} \qquad (7)$$

With $k=\pm 1$ the left or right steering, $L_{arc}=\theta R_{min}$, $L_{CAC}=2L+L_{arc}$, $R_{d1}=A^2/d$, $R_{d2}=A^2/(L_{CAC}-d)$.

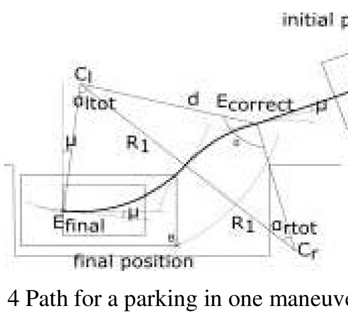

Fig. 4 Path for a parking in one maneuver with the geometric method using the angle μ and radius R_1

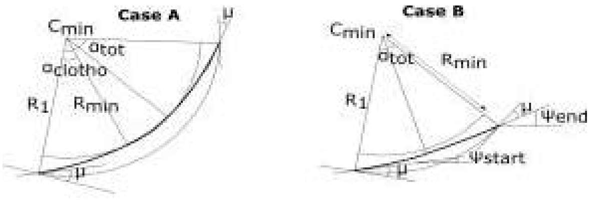

Fig. 5 Evolution from a circle arc to an A or B sequence

5. Regeneration of the Trajectory

5.1. General Strategy

Different methods exist to ensure that the parking is correct and can be divided into two groups: methods which use reference trajectories and these which doesn't use one.

The first step for the methods which use reference trajectories is to create a trajectory. Then, robust control laws , e.g. [20], ensure that the trajectory is well executed. However, these methods need to calculate explicitly the (x,y)-points of the trajectory, which would need a lot of computational cost for our method, and the sensors need to measure precisely the pose of the vehicle at each moment, what is not always possible. Moreover, if the environment changes, all the plannification has to be redone entirely.

Some methods don't use any referece trajectory. For example, [21] uses fuzzy logic to create the control laws knowing at each moment the pose of the vehicle. However, these methods also need to have good sensors enough to give precise information of the pose of the vehicle at each moment.

Our strategy for the correction of the trajectory brings novelty with respect to the state of the art. As we have shown in [15], big deviations from the initial path planning are observable only after 4 maneuvers (a maneuver is a sequence in the same direction). Consequently, we consider that our control method is precise enough at least for one maneuver and it does not need to be corrected at every time step. Moreover, an emergency stop can prevent the vehicle from collision if the deviation is dangerous, thus we don't need to know the precise pose of the vehicle at each moment. As we will show, our regeneration method need the precise pose of the vehicle only when it stops. Our path planning method and associated control signal being simple and efficient, it would be therefore interesting to have the same advantages for the regenerated path.

The method we present relies on the generation of a trajectory between two poses. Each time the vehicle stops (end of a backward or forward maneuver or emergency stop due e.g. to an obstacle being too near), the position of the vehicle is estimated and compared to the calculated position of the path planning. If the error makes unacceptable the next maneuvers, the regeneration of the path occurs. However, if the current pose of the vehicle is acceptable for a final pose, the parking maneuver is considered as finished and no further maneuver is performed.

The method to regenerate the path has to respond to the same constraints as the initial path planning: that means to satisfy the geometric constraints, to have a low computational cost and, if possible, a continuous-curvature.

An interesting solution is to use the reversed geometrical method as for the initial path planning. In that case, there is no need to re-calculate the trajectory (as the method of retrieving from the parking spot never changes) for all the maneuvers except for the one that the vehicle should have done immediately after the stop. The problem consists thus in finding a trajectory from the real pose of the vehicle (position and orientation), to the next stop pose calculated during the initial path planning. For example in Fig. 6, for a parking in three maneuvers, the algorithm generates a trajectory characterized by the first real position (1r), a second, third and fourth wished positions (2s, 3s, 4s). If at the end of the first maneuver the real position of the vehicle is 2r instead of 2s, the problem is to generate a path between 2r and 3s. The path between 3s and 4s already exists. In the section 5, the pose of the vehicle is represented as an arrow, which starts at the point E of the vehicle and whose direction represents the orientation of the vehicle.

5.2. Continuous-curvature solution

5.2.1. Bi-elementary paths

In order to keep simple control signals and admissibility of the trajectory, clothoids are used. In [9], bi-elementary paths composed by two sequences CAC (without the optional arc of circle) are presented. They can link any two poses of the vehicle thanks to an intermediary pose, symmetric to both the initial and final poses of the vehicle (Fig. 7).

To create a bi-elementary path between two poses of the vehicle $q_A=(x_A,y_A,\varphi_A)$ and $q_B=(x_B,y_B,\varphi_B)$, a new frame R_{AB} is defined, it is centered in the middle of the segment *[AB]* and its *x*-axis is oriented along the vector $\vec{AB}$. In this frame, the coordinates of q_A and q_B are respectively $(-r/2, 0, \alpha)$ and $(r/2, 0, \alpha')$, with $\alpha=\varphi_A-\theta$, $\alpha'=\varphi_B-\theta$ and θ is the orientation of R_{AB} in the global frame (Fig. 8).

The results of [9] show that it is possible to find a CAC sequence (without the optional arc of circle) between two poses q_A and q_B only if theses poses are symmetric. This means that the orientations of these poses are symmetric with respect to the line joining their positions. Formally, the coordinates of q_A and q_B must verify:

$$(x_B - x_A)\sin\left(\frac{\varphi_B+\varphi_A}{2}\right) = (y_B - y_A)\cos\left(\frac{\varphi_B+\varphi_A}{2}\right) \qquad (8)$$

In practice, two poses are rarely symmetric, so an intermediary pose q_I, symmetric to both q_A and q_B, is needed to link q_A and q_B by CAC sequences. The authors of [9] show that the set of such poses has a constant curvature $\kappa=2\sin(\beta)/r$, with $\beta=(\alpha'-\alpha)/2$. Consequently, this set is a circle of center C and radius $R=1/\kappa$ if $\kappa\neq 0$ or a line otherwise (Fig. 9). This set is defined by (*s* refers to the length along the set of positions):

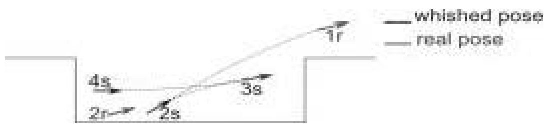

Fig. 6 Examples for the regeneration of the trajectory

Fig. 7 Examples of bi-elementary paths

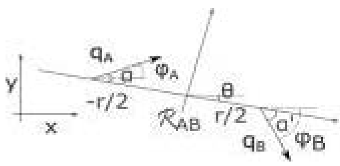

Fig. 8 Frames involved in bi-elementary paths

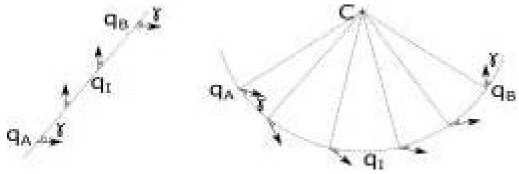

Fig. 9 Configurations symmetric to *qA* and *qB*

Remark: the angle between the tangent of the curve in respectively A and B, and the poses q_A and q_B is $\gamma=|\alpha+\alpha'|/2$. The angle between the tangent to the curve in I and the pose q_I, is $-\gamma$.

$$q_I(\kappa,s)=\begin{cases} x_I(\kappa,s)=\sin(\kappa s)/\kappa \\ y_I(\kappa,s)=(\cos(\beta)-\cos(\kappa s))/\kappa \\ \theta_I(\kappa,s)=\kappa s-(\alpha+\alpha')/2 \end{cases} \quad (9)$$

and if $\kappa=0$:

$$q_I(0,s)=\begin{cases} x_I(0,s)=s \\ y_I(0,s)=0 \\ \theta_I(0,s)=-\alpha \end{cases} \quad (10)$$

Having two symmetric poses, it is possible to create a CAC($A,L,0$) sequence taking the following parameters [9]:

$$A=\sqrt{1/\sigma} \quad (11) \qquad L=\sqrt{2\beta/\sigma} \quad (12)$$

with $\sigma=\frac{4\pi\left(\cos(\beta)C_f(\sqrt{2\beta/\pi})+\sin(\beta)S_f(\sqrt{2\beta/\pi})\right)^2}{r^2}$.

5.2.2. Strategy

When the regeneration of the trajectory has to be done between two poses q_A and q_B, the method presented above is applied to find a trajectory composed by one CAC sequence (if the poses are symmetric) or by a bi-elementary path (if the poses are not symmetric). Once this trajectory is created, we verify that it is admissible and collision free. If it is admissible and collision free, the created trajectory is kept. Otherwise, if it is possible to use curvature discontinuous paths, the solution presented in 5.3. is implemented. If it is not allowed, the solution presented in 5.4. can be used. In the sequel, we study the conditions of admissibility for the created trajectory.

If the poses are symmetric, the CAC sequence is created using the Eq.(11) and Eq.(12). It is admissible if the needed steering does not exceed the maximal steering of the vehicle, which means that $A_{AB}^2/L_{AB}\geq R_{min}$.

If the poses are not symmetric, there exists a set of intermediary poses q_I. The difficulty resides on finding an admissible intermediary pose q_I, for which $A_{AI}^2/L_{AI}\geq R_{min}$ and $A_{IB}^2/L_{IB}\geq R_{min}$. Due to the high computational cost of Fresnel integrals involved in the creation of the CAC sequences, all the set cannot be analyzed. The geometric study of the functions give the following results (see Fig. 10). Let P be the intersection point between the line containing qA, the circle of center C and radius R. If the point I goes from A to B, when it goes close to the point P, the steering needed for the first circle arc decreases (which mean that A_{AI}^2/L_{AI} decreases). If P is not situated between A and B, when the point I goes from A to B, the steering needed for the first circle arc increases. Moreover, wherever the point P is situated, when the point I goes from A to B, the steering needed for the second circle arc decreases (which mean that A_{IB}^2/L_{IB} decreases). These variations allow us to proceed by dichotomy to find an admissible pose q_I. If P is situated between A and B, the dichotomy is done on the circle arc between A and P. Otherwise, it is done on the circle arc between A and B. The conditions to stop the algorithm of dichotomy are:

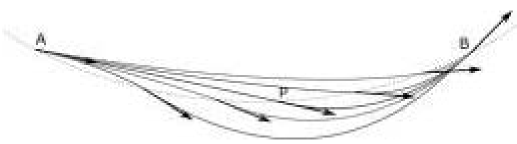

Fig. 10 Variation of the maximal curvatures of the CAC sequences in function of the intermediary pose

1) An admissible pose q_I is found.
2) A pose q_I such as $A_{AI}^2/L_{AI}<R_{min}$ and $A_{IB}^2/L_{IB}<R_{min}$ is found. In that case, giving the geometric study of the variations of the functions, no admissible pose q_I exists.
3) The dichotomy exceeds a given number of iterations. In that case, we consider that no admissible pose q_I exists.

5.3. Solution based on circle arcs

As it was done in the previous section with clothoids, two poses q_A and q_B can be linked by a circle if the poses are symmetric or by two circles thanks to an intermediate pose q_I if the poses q_A and q_B are not symmetric. The set of poses q_I is defined by the equations (9) and (10) and is a line or a circle of center C and of radius R as defined in the previous section. If such a created path is admissible and collision free, the trajectory is used. Otherwise, the solution presented in 5.4 is implemented. In the sequel, we study the conditions of admissibility for the created trajectory.

If the poses q_A and q_B are symmetric, the circle arc linking these poses is easily defined: its center is the intersection of the normals to the poses q_A and q_B.

If the poses q_A and q_B are not symmetric, a subset of intermediate poses q_I has to be found (with $I\neq A$ and $I\neq B$), such as $R_A\geq R_{min}$ and $R_B\geq R_{min}$, with R_A and R_B the radiuses of respectively the circle linking q_A and q_I and the circle linking q_I and q_B. To find this subset, a new frame centered in C with the x-axis oriented along the vector $\vec{CA}$ is defined (Fig. 11). In this new frame, the coordinates of the points A, B and I are given in function of the angle $\lambda=A\hat{C}I$ and we have:

$$A\begin{cases} x_A=R\cos(\lambda_A)=R \\ y_A=R\sin(\lambda_A)=0 \end{cases} \quad B\begin{cases} x_B=R\cos(\lambda_B) \\ y_B=R\sin(\lambda_B) \end{cases} \quad (13)$$
$$I\begin{cases} x_I=R\cos(\lambda) \\ y_I=R\sin(\lambda) \end{cases}$$

with $\lambda_B=acos(1-r^2/(2R^2))$, $\lambda_A=0$ and $\lambda\in]\ \lambda_A\ ,\ \lambda_B\ [$ and r defined as in the section 5.2.1.

The equation of the normal to the poses q_A, q_B and q_I are then respectively:

$$D_{NA}(\tau_A)\begin{cases} x_{DNA}=R(\cos(\lambda_A)-\cos(\lambda_A+\gamma)\tau_A) \\ y_{DNA}=R(\sin(\lambda_A)-\sin(\lambda_A+\gamma)\tau_A) \end{cases}$$
$$D_{NB}(\tau_B)\begin{cases} x_{DNB}=R(\cos(\lambda_B)-\cos(\lambda_B+\gamma)\tau_B) \\ y_{DNB}=R(\sin(\lambda_B)-\sin(\lambda_B+\gamma)\tau_B) \end{cases} \quad (14)$$
$$D_{NI}(\tau)\begin{cases} x_{DNI}=R(\cos(\lambda)-\cos(\lambda-\gamma)\tau) \\ y_{DNI}=R(\sin(\lambda)-\sin(\lambda-\gamma)\tau) \end{cases}$$

where the parameters τ_A, τ_B, τ are real numbers.

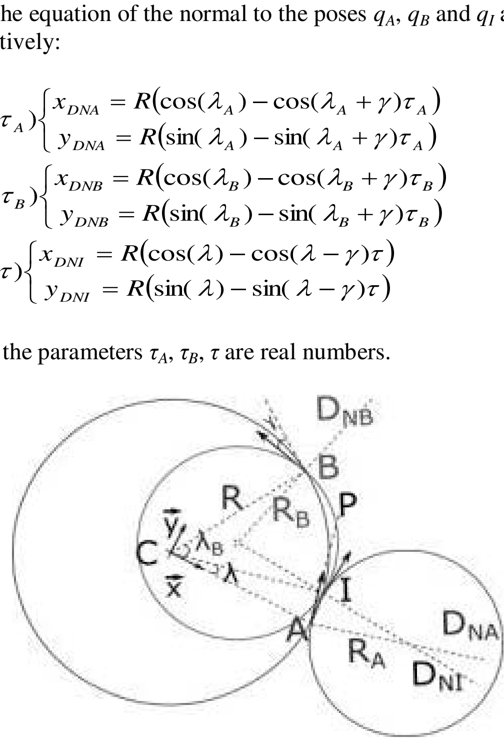

Fig. 11 Solution based on two circle arcs

In the sequel, as all the formulas for the calculations for respectively first the points A and I and then the points B and I are of the same shape, the formulas are written in function of *X*, where *X* is to be replaced respectively by *A* or *B*.

The intersection C_{XI} of D_{NX} and D_{NI} is calculated. The point C_{XI} represents the center of the circle of radius R_X and joining the points *X* and *I*. By writing $D_{NX}(\tau_X) = D_{NI}(\tau)$, the point C_{XI} and the values τ_X and τ are found. The value τ is:

$$\tau(\lambda) = \frac{\sin(\lambda_X) + N_X(\cos(\lambda) - \cos(\lambda_X) - \sin(\lambda))}{N_X \cos(\lambda - \gamma) - \sin(\lambda - \gamma)} \quad (15)$$

with $N_X = \tan(\lambda_X + \gamma)$.

The problem is to find all the radiuses R_X such as $R_X \geq R_{min}$ and we have $R_X = R|\tau|$. Consequently, to solve the problem it is sufficient to study the variations of R_X in function of λ and to solve the equations (16) and (17):

$$R\tau - R_{\min} = 0 \quad (16) \qquad R\tau + R_{\min} = 0 \quad (17)$$

Proceeding by implications, two solutions are found for Eq. (16):

$$\lambda = 2e_1 \tan\left(\frac{-g_1 \pm \sqrt{g_1{}^2 - e_1{}^2 + f_1{}^2}}{e_1 - f_1}\right) \quad (18)$$

with:

$$e_1 = R(\sin(\lambda_X) - N_X \cos(\lambda_X))$$
$$f_1 = N_X R - R_{\min} N_X \cos(\gamma) - R_{\min} \sin(\gamma)$$
$$g_1 = -R - R_{\min} N_X \sin(\gamma) + R_{\min} \cos(\gamma)$$

The unique solution is then found by testing the two values of λ.

With the same method, the Eq. (17) is solved and the unique solution is to find by testing the two solutions of the implications:

$$\lambda = 2e_2 \tan\left(\frac{-g_2 \pm \sqrt{g_2{}^2 - e_2{}^2 + f_2{}^2}}{e_2 - f_2}\right) \quad (19)$$

with:

$$e_2 = R(\sin(\lambda_X) - N_X \cos(\lambda_X))$$
$$f_2 = N_X R + R_{\min} N_X \cos(\gamma) + R_{\min} \sin(\gamma)$$
$$g_2 = -R + R_{\min} N_X \sin(\gamma) - R_{\min} \cos(\gamma)$$

The study of the variations of the functions $R_A(\lambda)$ and $R_B(\lambda)$ gives the following--g results in $]\lambda_A, \lambda_B[$ with $]\lambda_A, \lambda_B[\subset [0, 2\pi[$:

- $R_A(\lambda)$ increases if zero or one solution of the equations (16) and (17) is in this range. It increases and then decreases if both the solutions of the equations (16) and (17) are in this range.
- $R_B(\lambda)$ decreases in this range.

Thanks to the study of the functions' variations $R_A(\lambda)$ and $R_B(\lambda)$, and, the solution of the Eq. (16) and (17), we found a range $[\lambda_1, \lambda_2]$ (it may be null or open if $\lambda_1 = \lambda_A$ or $\lambda_2 = \lambda_B$) in which R_A and R_B are admissible for the vehicle. If this range is empty, the solution described in the next section is used. If it is not empty, the λ for the pose q_I can be chosen, with one of the radiuses R_A or R_B being the biggest possible.

5.4. Solution with a forward or backward move

If the previous methods do not provide any admissible and collision free-path, a simple forward or backward move with maximum steering is performed until the vehicle reaches the front or back obstacle. After this maneuver, as the vehicle doesn't have the pose calculated by the path planning, the routine of the regeneration of the trajectory has to be performed for the next maneuver. The forward or backward move has to be done with as much steering as possible.

5.5. Simulation

In this section, we illustrate a situation when the regeneration is needed with Matlab simulation results. To consider that a parking is correct, there is no need for the vehicle to achieve the exactly planned pose. We assume that a final pose is correct as long as the vehicle is entirely inside the spot. The initial conditions and the parameters of the vehicle for the simulation are presented in the Table 1.

The Fig. 12 presents a case of a parking in 3 maneuvers with a perfect following of the initially planned path.

To simulate perturbations during a real parking, we calculate the perfect trajectory and the perfect final pose for each maneuver and then add random perturbations to this final pose. The Fig. 13 and Fig.14 present parking with the same initiall random perturbation. In the Fig. 13, there is no regeneration of the trajectory. Consequently, at the end of each maneuver, even if the final pose is not the whished one, the vehicle continues to execute the control signals as it was initially planned, which leads to an incorrect parking. In the Fig. 14, the regeneration of the trajectory is used, which allows the vehicle to adapt the path without collisions and to achieve a correct final pose.

Table 1 Parameters of the vehicle and initial conditions

Parameters	**Value**	**Initial Conditions**	**Value**
a	2588 mm	$Length_{parking\ spot}$	6,31 m
$2b$	1511 mm	$Width_{parking\ spot}$	2,3 m
d_{front}	839 mm	x_{init}	7,5 m
d_{rear}	657 mm	y_{init}	4 m
$d_r = d_l$	130 mm	φ_{init}	0°
δ_{max}	33°		

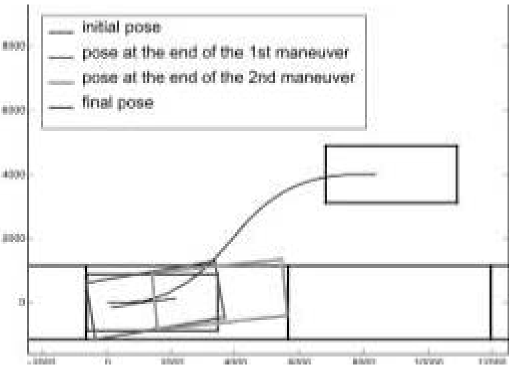

Fig. 12 Perfect parking in 3 maneuvers

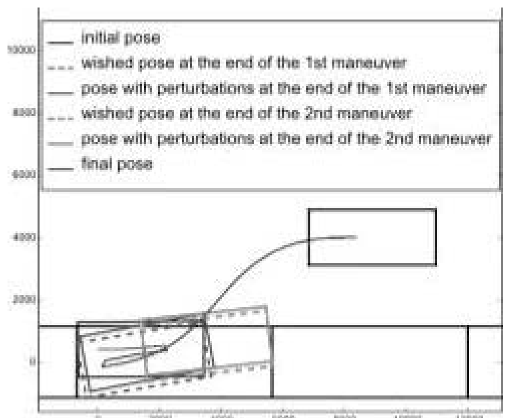

Fig. 13 Parking with perturbations without regeneration

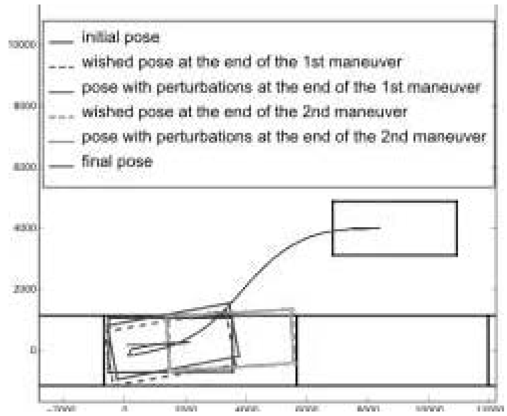

Fig. 14 Parking with perturbations with regeneration

6. Vehicle Parking Maneuver with Trajectory Regeneration

The previous sections presented the path planning and regeneration methods based on the environment data and the geometry of the vehicle. Moreover, control speed and steering signals have been presented. To be integrated into the vehicle as a parking system, these modules have to be linked in a more complete functional architecture that will be presented in this section. Moreover, a supervisor has to make decisions about the maneuver, oversee its proceedings and provide to the modules the data they need.

6.1. Global functional architecture

In this section we refer to the Fig. 15. The parking system has 4 modules: the supervisor, the perception of the vehicle in the external world, the path planning and the control signals.

The module of the vehicle in the external world gives to the supervisor the information about the size of the parking spot, its distance from the vehicle and the obstacles. Moreover, the calculations of the travelled distance are also done in this module thanks to the proprioceptive sensors, for example with the speed of each wheel.

The module of path planning calculates the initial path with the parameters imposed by the supervisor, for example if the maneuvers have to begin by a forward or a backward move, the choice of the method and of the curves. This module also calculates the regeneration of the trajectory when needed.

The module of the control signals calculates the speed and steering signals at each time step and insures that these signals are executed by the actuators.

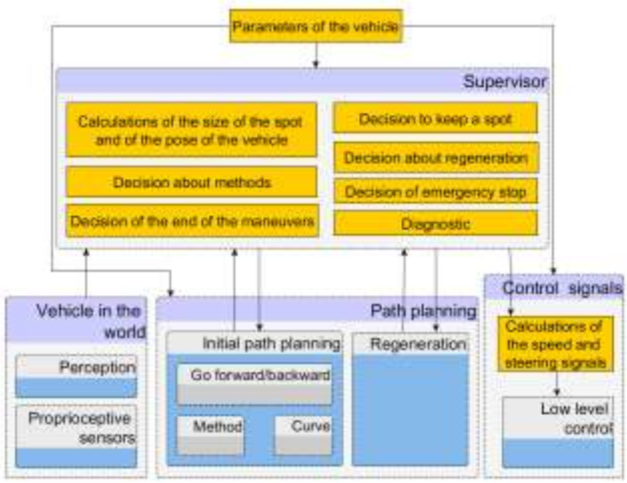

Fig. 15 Global functional architecture

Finally, the supervisor has to calculate the pose of the vehicle and of the parking spot in a local map from the information provided by the units. It also has to make decisions about the methods for the path calculations or the regeneration if the spot is big enough to initiate a parking. It has to decide when to stop the maneuvers (end or emergency stop) and ensure the global diagnostic of the system.

6.2. Proceedings of the parking maneuver

In this section we describe the proceedings of the automatic parking. The pseudo-code for this algorithm is in the Fig. 16.

6.2.1. First step of the parking maneuver

The first step is to find a suitable spot for the parking maneuver and might be restarted until such a spot is found

Then, when a spot is detected by the vehicle in the world module, all the information about this spot is sent to the supervisor. The supervisor determines the size of the spot and decides to keep this spot if its size is at least equal to a predetermined minimal size. If the spot is not kept, another spot has to be searched.

If the spot is considered, the supervisor determines the pose of the vehicle with an optional data fusion. Then, it chooses the methods and options for the initial path planning and selects the information to give to the path planning module, which generates an initial path and sends to the supervisor the parameters of the trajectory. Afterwards, the supervisor analyses these data (number of maneuvers, eventual collisions with the environment, overtaking on the lane) and decides to finally keep or not the spot and proceed to the parking maneuvers. If the spot is not kept, the first step of the parking maneuver has to be restarted. Otherwise, the second step of the parking maneuver is carried out.

```
While suitable spot == false
    Vehicle in the world : environment
    Supervisor : size of spot
    If size of spot > minimal size
        Supervisor : pose of vehicle
        Supervisor : choice of methods
        Path planning : intial path
        Supervisor : decides if the initial path is suitable
        If initial path suitable
            Break
        End
    End
End
While parking end == false
    While |speed|>0 OR (speed == 0 AND distance to travel
    for the current maneuver < travelled distance AND
    emergency == false)
        Vehicle in the world : obstacles, odometry, speed
        Supervisor : emergency ?
        Control signals : new speed and steering
    End
    Vehicle in the world : environment, pose
    Supervisor : size of spot
    Supervisor : end of parking ?
    If parking end == false
        Supervisor : regeneration ?
        If regeneration needed
            Supervisor : choice of methods
            Path planning : new path
        End
    End
End
```

Fig. 16 Algorithm of the parking maneuver

6.2.2. Second step of the parking maneuver

This step considers each maneuver one by one and consequently, it has to be restarted until the supervisor decides to end the parking.

First, a loop is done until the vehicle stops or until the vehicle is stopped but the distance which had to be travelled for the considered maneuver has not been travelled yet and no emergency stop was imposed. During this loop, the module of the vehicle in the world gives to the supervisor, at each time step, the information about the proximity of the obstacles and the travelled distance. If the supervisor considers that the obstacles are too close, an emergency stop is triggered and a null speed signal is sent to the control module. Otherwise, the parameters needed for the calculations of the speed and steering signals are sent to the control module.

When the vehicle stops because of an emergency stop or because the distance to travel for the considered maneuver is finished, the vehicle in the world module sends to the supervisor the information about the spot and the obstacles. Then, the supervisor determines again the size of the spot and the pose of the vehicle.

Afterwards, the supervisor decides if the parking is ended. Two ends are possible: the vehicle is considered as well parked or the parking maneuvers have to be given up (for example if the sensors show a new obstacle which prevents a collision-free parking).

If the parking is not ended, the supervisor tests if a regeneration of the trajectory is needed. If positive, the supervisor sends to the path planning module all the needed data. Then the regeneration is calculated and the path planning module sends to the supervisor the new parameters of the path.

The second step of the parking maneuver is then restarted until the parking is ended.

7. Conclusion

A lateral parking based on a continuous curvature trajectory in several maneuvers is presented. The trajectory generation starts with a geometric path based on circle arcs that is former transformed thanks to the clothoid curves to gain a continuous curvature path. An algorithm for the regeneration of the trajectory for the automatic parallel parking is further proposed, for instance in case of important deviation or obstacles. The initial path generation and the regeneration are included in a global functional architecture of the automatic parking. Next foreseen steps are to link this trajectory generation algorithms to the perception algorithms and to embed the calculus in a real control unit.

References

(1) A. Gupta and R. Divekar, "Autonomous parallel parking methodology for Ackerman configured vehicles," in *Proc. of Int. Conf. on Control, Communication and Power Engineering*, Chennai, India, July 2010.

(2) T. Inoue, M. Q. Dao, and K.-Z. Liu, "Development of an autoparking system with physical limitation," in *Proc. of SICE Annuel Conf.*, Sapporo, Japan, August 2004.

(3) K. Jiang, D. Z. Zhang, and L. D. Seneviratne, "A parallel parking system for a car-like robot with sensor guidance," in *Proc. of the Institution of Mechanical Engineers, Part C: Journal of Mechanical Engineering Science*, 1999.

(4) S. Choi, et al. "Easy path planning and robust control for automatic parallel parking in tiny spots", in *Proc. of IFAC World Congress*, Milano, Italy, August-September 2011.

(5) H. Vorobieva, et al., "Geometric path planning for automatic parallel parking in tiny spots," in *Proc. 13th IFAC Symp. Control in Transportation Systems*, Sofia, Bulgaria, September 2012.

(6) B. Mueller, J. Deutscher, and S. Grodde, "Continuous curvature trajectory design and feedforward control for parking a car," in *IEEE Transactions on Control Systems Technology*, Vol. 15, No. 3, pp 541-553 (2007)

(7) H. Vorobieva, et al., "Geometric continuous- curvature path planning for automatic parallel parking", in *Proc. of IEEE ICNSC*, Evry, France, April 2013.

(8) T. Fraichard and A. Scheuer, "From Reeds and Shepp's to continuouscurvature paths," in *IEEE Trans. Robotics*, Vol. 20, No. 6, pp 1025-1035 (2004)

(9) A. Scheuer and T. Fraichard, "Collision-Free and Continuous-Curvature Path Planning for Car-Like Robots", in *Proc. of IEEE ICRA*, Albuquerque, NM, USA, April 1997.

(10) P. Bonnifait, P. Bouron, P. Crubillé, and D. Meizel, "Data fusion of four abs sensors and gps for an enhanced localization of car-like vehicles," in *Proc. of IEEE Int. Conf on Robotics and Automation*, Seoul, Korea, May 2001.

(11)K. Jiang, D. Z. Zhang, and L. D. Seneviratne, "A parallel parking system for a carlike robot with sensor guidance*," Proc. of the Institution of Mechanical Engineers, Part C : Journal of Mechanical Engineering Science*, 1999.

(12) T.-H. Hsu, J.-F. Liu, P.-N. Yu, W.-S. Lee, and J.-S. Hsu, "Development of an automatic parking system for vehicle," in *Proc. of IEEE Vehicle Power and Propulsion Conf.*, Harbin, China, 2008.

(13) (2012) SRR 20X /-2 /-2C /-21 Short Range Radar. Continental. Allemagne. [Online]. Available : http://www.conti-online.com/www/download/industrial_sensors_de_en/themes/download/srr20x_datasheet_en.pdf (Access Date 2015.03.27). Access date/Web site only.

(14) Y.-W. Ryu, S.-Y. Oh, and S.-Y. Kim,"Robust automatic parking without odometry using an evolutionary fuzzy logic controller," *Int. Journal of Control, Automation, and Systems*, Vol. 6, No. 3, pp 434-443 (2008)

(15) H. Vorobieva, S. Glaser, N. Minoiu-Enache, and S. Mammar, "Automatic parallel parking with geometric continuous-curvature path planning," in *Proc. IEEE Intelligent Vehicles Symp.*, Dearborn, MI, USA, June 2014.

(16) D. Kim, B. Song, "Lateral Vehicle Control for Semi-Autonomous Valet Parking with Consideration of Actuator Dynamics", *Int. Journal of Control Science and Engineering*, Vol. 2, No. 6, pp 150-156 (2012)

(17) N. Minoiu Enache, "Trajectory Tracking Control by LMI-based Approach for Car-like Robots", in *Proc. of 9th Int. Conf. on Informatics in Control, Automation and Robotics*, Vol. 2, Rome, Itlay, pp 38-47 (2012)

(18) V. Milanés, et al., "Low-Speed Longitudinal Controllers for Mass-Produced Cars: A Comparative Study", *IEEE Transactions on Industrial Electronics*, Vol. 59, No. 1, pp 620-628 (2012)

(19) P. Ioannou and Z. Xu, "Throttle And Brake Control Systems For Automatic Vehicle Following", *Journal of Intelligent Transportation Systems,* Vol.1, issue 4, pp 345-377 (1994)

(20) O. Amidi, "Integrated mobile robot control", Robotic Insitiute, Carnegie Mellon University, Pittsburgh, Pa, USA, Tech. Rep. *Tech. report* cmu-ri-tr-90-17, 1990

(21) H. Miyata, M. Ohki, Y. Yokouchi, and M. Ohkita, "Control of the autonomous mobile robot dream-1 for a parallel parking", *Mathematics and Computers in Simulation*, Vol. 41, No. 1-2 (1996)