2019 6th International Conference on Control, Decision and
Information Technologies (CoDIT'19) | Paris, France / April 23-26, 2019

Implementation of a Parking State Machine on Vision-Based Auto Parking Systems for Perpendicular Parking Scenarios

Saeid Sedighi, Duong-Van Nguyen
Panasonic Automotive Systems Europe GmbH
Langen, Germany
e-mail: saeid.sedighi@eu.panasonic.com,
duongvan.nguyen@eu.panasonic.com

Klaus-Dieter Kuhnert
University of Siegen
Siegen, Germany
e-mail: kuhnert@fb12.uni-siegen.de

Abstract— **Recently autonomous parking has got a great attention from researchers from both Academy and Industry and being known as one of the key subsets of autonomous driving. Since autonomous driving is at its hype but with many remaining concerns, autonomous parking is more practical for daily applications and more feasible to be immediately realized mass-production in the automotive industry. Current solutions for autonomous parking require multiple maneuvers and a significant amount of time to park, which lead problematic in many real-life applications, depending on traffic situation around the ego-vehicle. In this paper, we present a parking strategy for vision-based autonomous parking systems in which the ego-vehicle is capable of completing its auto-park by one maneuver, or up to maximum three required maneuvers. Experiments show that the proposed method provides a significant improvement in reducing the number of maneuvers as well as time to park. Indeed, a rate of more than 95% successful parking in heavy traffic flows within a parking area is achieved using this method. We will also prove that the method is lightweight and very efficient for real-time applications.**

Keywords-component; Path planning; Auto parking system; Vision-based parking system; Autonomous driving.

I. INTRODUCTION

Parking and its problems are one of the actual challenges of using vehicles in daily applications. According to the annual accident report of European Road Safety Observatory [1], one of the main causes of accidents in low-speed driving inside cities is park maneuvers. This is why one of the main criteria to judge the driving skill of drivers is parking skills.

Regarding the actual applications of autonomous driving technology, parking assistant systems are the most realistic representation of autonomous driving to the automotive industry which normal people can adapt to that easier. Parking assistant systems can be used as a fully automated system or a partially automated system. Using any auto parking system assists car users to park in crowded areas where the parking maneuvers are restricted in several aspects from tight places to limited time to park. According to the General Parking Standards report [2], inside cities, the average parking place for perpendicular parking scenarios is about ~2.4 m wide and the time for parking should be as short as possible related to the traffic jam to avoid any unpredicted traffic jam behind the ego vehicle.

The problem of current parking assistant systems is the required time to park under these conditions. Finding an optimum path which is able to be executed under a certain amount of time by non-holonomic constrained vehicle is an open challenge in the automotive industry. There is a meaningful relationship between the number of maneuvers and the running time of the parking procedure. Current path planners for parking assistant systems are limited to this fact that their planners operate the same path planner algorithm (e.g. local Clothoid planner) to plan a single or multi required maneuvers (P) to finish the parking maneuver. Here the whole parking maneuver apart of single or multi goals maneuvers must be a continuous path, means:

$$P_i.s(x,y) = P_{(i\text{-}1)}.e(x,y), \tag{1}$$

$$P_i.s(yaw) = P_{(i\text{-}1)}.e(yaw), \tag{2}$$

and:

$$P_N.e(x) = X_{Goal}, \tag{3}$$

$$P_N.e(y) = Y_{Goal}, \tag{4}$$

$$P_N.e(yaw) = Yaw_{Goal}, \tag{5}$$

where s and e define the first and the last control point of each parking maneuver respectively and:

$$N=2n+1 \ (n = 1, 2, \cdots). \tag{6}$$

As the above equations illustrate, the number of maneuvers (N) for each parking procedure is minimum 3 [3]. This feature of the path planning for auto parking systems supports planning a path for complicated scenarios in which the road width or the size of the detected parking space is critical to park with just one single maneuver [4]. On the other hand, having unnecessary maneuvers creates a huge delay in parking procedure which is not optimum for the parking assistant systems in which the ego vehicle deals with real traffic situations. Less number of maneuvers means less parking time which makes the whole procedure of the parking maneuver much faster.

This paper is presenting an approach of path planning for auto parking scenarios in which the vehicle will be able to park in maximum three maneuvers for vision-based parking assistant systems.

A. Related Works

Path planning in the automotive industry is a recent research topic. Here the main challenge of path planning is finding the local optimum path to guide the vehicle from the actual or initial position (q_I) to the target position (q_G). One of the very first path planning algorithms which were used in parking applications was Reeds-Shepp local path planner [5], however, this method provides the shortest possible path (in a fully provided collision-free state space) to the target point, it cannot support linear transition for discrete maneuvers (straight to full steering) to provide a continuous path for a non-holonomic vehicle [3]. Clothoid-based local path planner is the most useful method of path planning with local optima in parking applications, however this method is not able to avoid obstacles while planning the required path to the target position [3].

To avoid any collision to the obstacles while parking, auto parking assistant systems need to plan multi required maneuvers to reach the target point. This property of Clothoid curves and the non-holonomic constraints of the vehicles limit the maneuverability of the vehicle in tight parking spaces. Due to this reason, the decision maker of auto parking systems needs to plan several maneuvers to damp these limitations.

Some recent works [4] have introduced new methods of local path planning in which the number of required maneuvers can be controlled in advance to have a deterministic motion planner algorithm. This paper introduces a path planner algorithm for perpendicular parking scenarios in which a limited number of maneuvers could be selected and planned regarding the size of the parking place, local configuration and other parameters. This method of path planning can be used in vision-based parking assistant systems where the side cameras (or others) of the ego vehicle fusing the collected camera data with gathered information from other sensors such as sonars.

II. REEDS-SHEPP PLANNER

Dubins curves is one the fundamental path planner algorithms in robotics which guarantees to find the shortest possible path from the given initial pose (q_I) to the desired goal position (q_G) for a non-holonomic vehicle. This method plans a path with considering that the vehicle is moving just forward (one direction) and in each position the planner has just three states, turning left or right with the maximum steering angle ($\o_{MAX}$) which creates a turn with the maximum possible curvature (K_{MAX}) or driving in a straight line [3]. This method has the following possible states:

$$\{LRL; RLR; LSL; LSR; RSL; RSR\}, \tag{7}$$

in which R and L mean turning with $\o_{MAX}$ (steering angle) to right and left respectfully, and S means straight driving.

Reeds-Shepp path planning has the same concept but additionally, the vehicle is able to drive in reverse direction as well. This feature gives the planner more states to select, here 46 different planning strategies for the path will be available [5].

However, this method assures finding the shortest possible path to the target position, it is still constrained by discrete states which create a non-continues path for the vehicle [6].

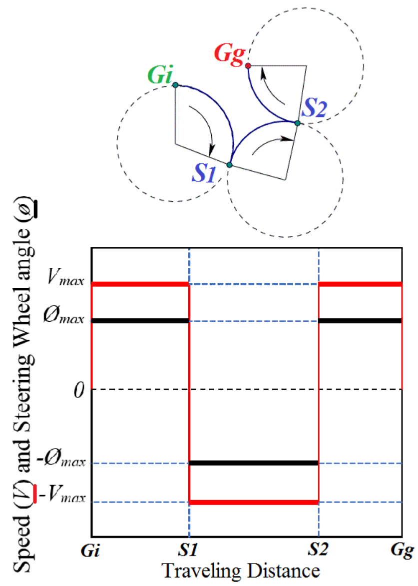


Figure 1. An example of Reeds-Shepp path including its speed-distnace gragh.

From Fig 1, the planned path in S_1 and S_2 positions is not continuous because the ego vehicle requires to change its steering angle from right full steering to full steering to left and vice versa while the vehicle is moving with a constant speed ($|\pm V_{max}|$).

III. CLOTHOID CURVE

Clothoid curves provide a smooth and drivable transition path from the straight driving to a turn with a constant curvature. Clothoid equations can be defined starting from the condition of linear relation between radius (R) and length (L):

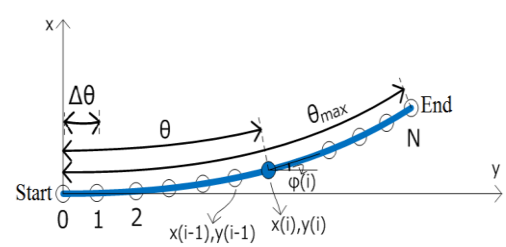
$$rl = R_i L_i = Constant = A^2, \tag{8}$$

where r and l are the radius and length of the path in each control point. This defines an infinite spiral, starting from the origin ($x = 0$, $y = 0$, $R = \infty$, $L = 0$) and spinning in two infinite loops to two points where $R = 0$ and $L = \infty$.

The constant A is called flatness or homothetic parameter of the Clothoid.

A. X-Y Coordinate

X-Y coordinate of each point of the Clothoid curve is defined as follow:



$$X = \frac{1}{2\pi}\int_0^{\theta} \cos\left(\frac{\theta^2}{2}\right) d\theta \,, Y = \frac{1}{2\pi}\int_0^{\theta} \sin\left(\frac{\theta^2}{2}\right) d\theta, \quad (9)$$

$$\theta_i = \theta_{max} \times \frac{i}{N}, \quad (10)$$

as for direction:

$$\phi(i) = \frac{\theta_i^2}{2}, \quad (11)$$

as for X coordinate:

$$X(i) = \int_0^{\theta i} cos\left(\frac{x^2}{2}\right) dx$$

$$= \int_0^{\theta i-1} cos\left(\frac{x^2}{2}\right) dx + \frac{\theta_i - \theta_{i-1}}{6}\left\{\begin{matrix} cos\left(\frac{{\theta_{i-1}}^2}{2}\right) + \\ 4\, cos(\frac{1}{2}\left(\frac{\theta_{i-1} + \theta_i}{2}\right)^2) + \\ cos\left(\frac{{\theta_i}^2}{2}\right) \end{matrix}\right\}$$

$$X(i) = x(i\text{-}1) + \frac{\theta_i - \theta_{i-1}}{6}\left\{\begin{matrix} cos\left(\frac{{\theta_{i-1}}^2}{2}\right) + \\ 4\, cos(\frac{1}{2}\left(\frac{\theta_{i-1} + \theta_i}{2}\right)^2) + \\ cos\left(\frac{{\theta_i}^2}{2}\right) \end{matrix}\right\}, \quad (12)$$

Y-coordinate in the same way:

$$Y(i) = y(i\text{-}1) + \frac{\theta_i - \theta_{i-1}}{6}\left\{\begin{matrix} sin\left(\frac{{\theta_{i-1}}^2}{2}\right) + \\ 4\, sin(\frac{1}{2}\left(\frac{\theta_{i-1} + \theta_i}{2}\right)^2) + \\ sin\left(\frac{{\theta_i}^2}{2}\right) \end{matrix}\right\}, \quad (13)$$

furthermore, it reaches the same value as the unit Clothoid concerning angle.

$$X(i) = A.x(i),\ Y(i) = A.y(i). \quad (14)$$

By benefiting from the Clothoid curve, the smooth transition for improving Reeds-Shepp algorithm to be used in motion planning for non-holonomic vehicles [7].

The above method creates a smooth and continuous speed-distance path in which the non-holonomic constraints of the vehicle are greatly considered.

IV. Vision-Based Parking System

The current work is presenting a method of path planning for perpendicular parking scenarios, appertaining to vision-based perception auto parking systems. As it is illustrated in Fig 2, our parking system is a vision-based system which benefits from robust detection of sonar sensors fusing with the collected information of two side cameras.

In a purely sonar-based parking system, the ego vehicle needs to cross the whole parking area to collect all required information then detects the parking borders while vision-based systems can detect the parking borders before crossing the whole parking area [8]. This feature of vision-based parking systems helps the decision maker of the parking systems to plan a path to the target area much earlier than a sonar-based system.

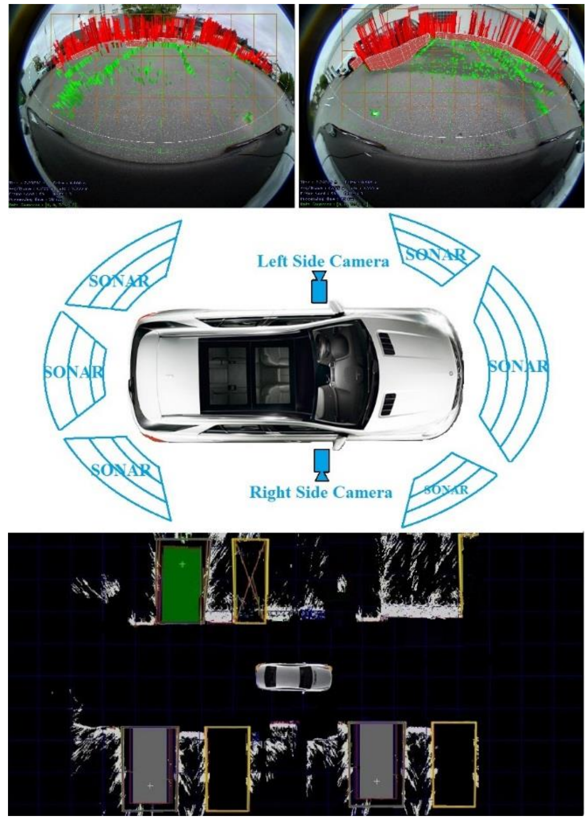


Figure 2. The used vision-based parking system and its fusion configuration (Panasonic Automotive).

In our parking system, the perception [9] part of the parking platform is able to provide the required information about the parking space as soon as the position of the mounted side camera on the vehicle (side mirrors) is in the middle of the detected parking space. At this time, the decision maker of the parking system starts to plan a path for the related parking space while the rear axle of the vehicle which mainly is used as the origin of the ego vehicle to plan a trajectory is ~1.83(m) (depending on the selected vehicle, here Mercedes-Benz C-class 2014) behind the middle of the parking space. This parking configuration gives the ego vehicle a sufficient time to plan a trajectory to park before the vehicle totally crosses the detected parking space, the available planning time can be calculated here:

$$X(distance) = V(speed) \times T(time), \quad (15)$$

where $X = L_V$ (the longitudinal distance from the side mirrors the rear axle of the vehicle) plus W_{P2} (half of the width of the detected parking place) and V_V (average speed of the vehicle which is around 15(kph) in normal parking situations). Therefore:

$$T = (L_V + W_{P2}) / V_V. \quad (16)$$

Using (16) and by considering some average values for a normal city vehicle in a normal parking situation (V_V = ~15(kph), L_V = ~1.8(m) and W_{P2} = ~1.5(m)) the available time to plan a path to park before reaching the border of the parking place is around 800 milliseconds.

Here depending on the road width (R_W) [4], the width and total length of the vehicle (W_V, L_{VT}), the side distance between the vehicle and the parking borders (D_S) and the width of the parking place (P_W), the decision maker implements different strategies to plan a path for the parking maneuvers, this deviation is presented in Table 1:

TABLE 1. PARKING STATES

Name	Kinematics Constraints	Number of Maneuvers
Wide	$R_W > 3W_V$ && $D_S > 1.5P_W$ && $P_W > (W_v + 1.2(m))$	1
Normal	$R_W > 3W_V$ && $D_S > P_W$ && $(W_V + 1.2(m)) > P_W > (W_V + 0.4(m))$	2
Narrow	$R_W > W_V$ && $(W_V + 0.4(m)) > D_S > 1.5P_W$ && $P_W > (W_V + 1.2\ (m))$	3

According to Table 1, the decision maker of the vehicle depending on the correct state of the maneuver, plans a one, two or maximum three-goal maneuvers to park. For some situations which are not listed here e.g. very short distance between the ego vehicle and the side objects or narrow road width, the decision maker can plan to ignore the parking place or implementing normal local path planner with more than 3 required maneuvers [4], [10].

V. STATE MACHINE

As it was explained in section IV, depending on the parking configuration, the state machine of the decision maker selects the correct relative maneuver to park, here all the maneuvers will be explored:

A. *One-goal maneuver*

One of the advantages of vision-based parking systems is the capability of planning a path to the target position with just one required maneuver. This feature of the planning may be difficult in purely sonar-based parking systems due to the lack of provided information of the parking place such as the correct size (width and depth) and border kind (obstacles) of the parking place. In a vision-based parking system, by benefiting from the fusion between the camera and other sensors of the ego vehicle, size and all obstacles around the detected parking place are captured easier.

By considering a vision-based parking system and selecting the one-goal maneuver due to the given criteria, decision maker using Clothoid curves combined with Reeds-Shepp planner plans a continuous path to park the vehicle with just on required maneuver under the following sequences:

Step one: longitudinal distance of the vehicle to the target position is chosen by terminating the forward driving in the approved range ($> 3W_V$).

Step two: depending on the ego vehicle (q_I, W_V, L_V, L_{VT}) and target parking position (q_G) the correct Reeds-Shepp path set is selected.

In just one goal parking maneuver, the ego vehicle does not need to change its direction of the travel, therefore, Reeds-Shepp method by considering the reverse driving direction as the constant direction of the travel plan the corresponding path.

Step three: having the Reeds-Shepp path, the Clothoid path planner needs to plan the transition curves to upgrade the selected path regarding the non-holonomic constraints of the vehicle. One of the most important criteria in this step is calculating the size of the arc (θarc) which the vehicle turns with its maximum steering angle (K_{MAX}). This parameter provides the planner with the precise length of the transition curve:

$$a = k^2/2\delta\ , \tag{17}$$

$$length = \sqrt{\frac{2\delta}{\alpha}}\ , \tag{18}$$

$$\theta_j = (length \times K_{MAX})/2, \tag{19}$$

$$\theta arc = (q_G(yaw) - q_I(yaw) - (2\times \theta_j), \tag{20}$$

where k and δ are the provided Clothoid parameters. Above steps create a continuous path with just one required maneuver to park according to the given positions (Fig. 3):

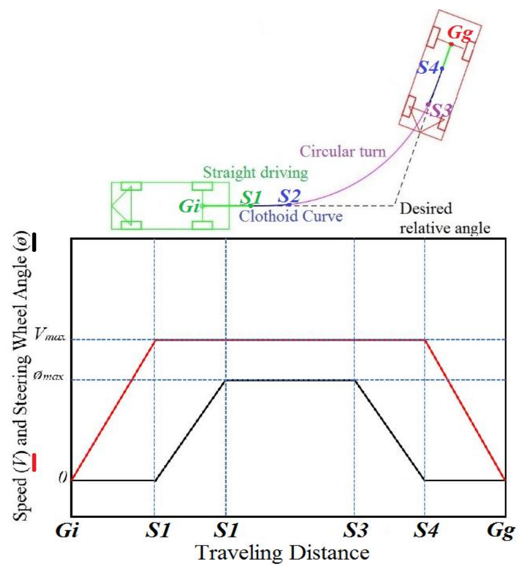


Figure 3. An example of one-goal vision-based parking maneuver and its related contincous speed-distance graph.

B. *Two-goal maenuver*

If the selected state to park is a two-goal maneuver, the decision maker of the vehicle plans two maneuvers in which the first maneuver adjusts the initial stopping position of the ego vehicle to be able to finish the parking maneuver in the second maneuver with just one move. The traditional Clothoid based local path planners, propose a three-goal maneuver to park in such a case [4], [10] whereas, a vision-based parking system is able to plan in just two required maneuvers by implementing the following sequence:

Step one: Using the forward Clothoid path planner, a forward driving path is planned to an imaginary goal position which is the tangent line of the crossed line of extended goal position and the far corner of the parking place by the relative angle of 45° degree.

Step two: a one-goal maneuver to the target position as explained in section A is used to plan a trajectory to the target position.

The planned path is generally continuous over the whole path regarding the non-holonomic constraints of the vehicle. In the transition point where the first maneuver is finished, and the second maneuver starts, the steering angle needs to change its position stationary to adapt the tire angles to the next maneuver (Fig. 4).

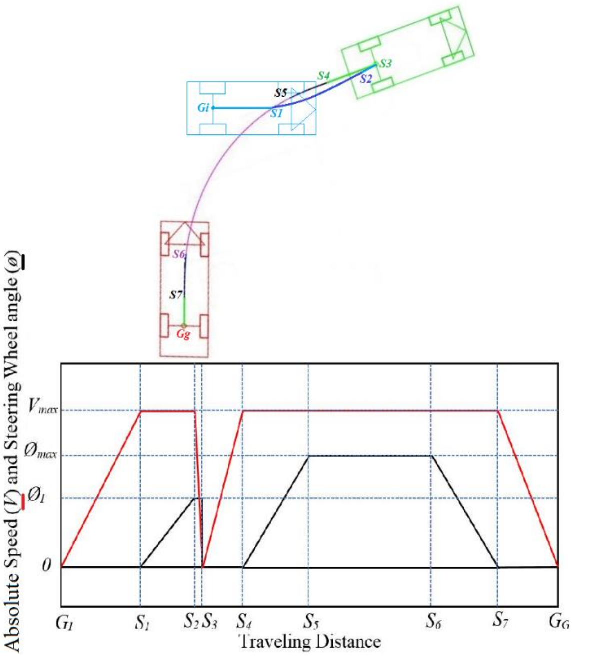


Figure 4. Vision-based parking system with two parking maneuvers.

C. Three or more goal maneuvers

An autonomous non-holonomic vehicle is constrained to its kinematic motion model. This mechanical limitation of the non-holonomic vehicles forces the decision maker of parking systems to plan several parking maneuvers for the use cases in which the maneuverability of the ego vehicle is limited in environments with limited moving spaces [4].

For this section, the three-goal maneuver for narrow perpendicular parking scenarios [4] (Fig. 5) which is a robust local planner for narrow parking scenarios may be implemented. This method provides the required parking maneuvers for very narrow parking spaces (up to W_V + 20(cm)).

For other use cases in which the above states cannot be implemented, the planner can choose between ignoring the proposed parking place due to several required parking maneuvers or using other Clothoid based planners to plan for more than three-goals maneuvers which is not recommended due to the control and actuator errors while implementing several parking maneuvers to park.

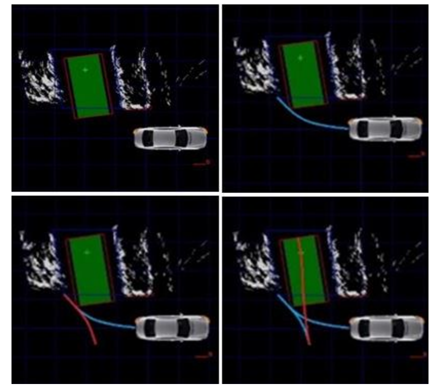


Figure 5. An example of vision-based three-goal parking maneuver (Panasonic Automotive).

By merging all above planning states, a state machine for local path planning for parking scenarios can be determined. This algorithm is explained shortly here:

Inputs: initial position of the vehicle (q_I), the goal position (q_G), road width (R_W), the width and length of the vehicle (W_V, L_{VT}), the side distance between the vehicle and the parking borders (D_V) and the width of the parking place (P_W), Vehicle data (*Veh*), Clothoid parameters (δ, k, dl).

Output: the optimum path to the target position regarding the non-holonomic constraints of the vehicle (*Path*).

a very simplified description of the algorithm for the parking scenarios can be presented here:

State Machine of Path Planning for Perpendicular Parking Slots

algorithm PathPlanner($\delta,k,dl,R_W,W_V,L_{VT},D_V,P_W,Veh,q_I,q_G$)
  State, Path, PathSeg1, PathSeg2, PathSeg3 ← Ø
  State = **StateMachine**(R_W, W_V, L_{VT}, D_V, P_W)(1)
  if *State* = stateOne **then**
    Path=**OneGoalPlanner**($\delta,k,dl,R_W,W_V,L_{VT},D_V,P_W,Veh,q_I,q_G$)
  else if *State* = stateTwo **then**
    Path=**TwoGoalPlanner**($\delta,k,dl,R_W,W_V,L_{VT},D_V,P_W,Veh,q_I,q_G$)
  else if *State* = stateThree **then**
    Path=**ThreeGoalPlanner**($\delta,k,dl,R_W,W_V,L_{VT},D_V,P_W,Veh,q_I,q_G$)
  else
    return *0*
  end if
  if *Path* ≠ Ø **then**
    return *Path*
  end if
  return *0*

OneGoalPlanner($\delta,k,dl,R_W,W_V,L_{VT},D_V,P_W,Veh,q_I,q_G$)
  Path = **ReedsSheppPlanner**(Veh,q_I,q_G)(2)
  D = Distance(q_I, q_G), T = Orientation(q_I, q_G)
  $\delta = \delta_1 D + \delta_2 T$
  $\alpha = \frac{k^2}{2\delta}$, $length = \sqrt{\frac{2\delta}{\alpha}}$, $dt = \sqrt{\frac{\alpha}{\pi}}dl$
  $t = dt{:}dt{:}\sqrt{\frac{2\delta}{\pi}}$, *Path*← Ø
  $x = \frac{\sqrt{2\pi\delta}}{k}\, cumtrapz(sin(\frac{\pi}{2}t^2))dt$

$y = \frac{\sqrt{2\pi\delta}}{k}$ *cumtrapz(cos(*$\frac{\pi}{2}t^2$*))dt*
Path ← *Path* ∪ [*x*, *y*]
[*x*, *y*] = rotate(*x*, -*y*, *π*)
Path ← *Path* ∪ [*x*, *y*]
if *Path* ≠ ∅ **then**
  return *Path*
end if
return *0*

TwoGoalPlanner($\delta, k, dl, R_W, W_V, L_{VT}, D_V, P_W, Veh, q_I, q_G$)
G_{G2}=**CrossPoint**(D_V, P_W, Veh, q_I, q_G)
PathSeg1 = **ContinousCurve** ($\delta, k, Veh, q_I, q_{G2}$)
if *PathSeg1* ≠ ∅ **then**
  Path ← *Path* ∪ *PathSeg1*
  PathSeg2= **OneGoalPlanner**($\delta, k, dl, R_W, W_V, L_{VT}, D_V, P_W, Veh, q_I, q_G$)
if *PathSeg2* ≠ ∅ **then**
  Path ← *Path* ∪ *PathSeg2*
if *Path* ≠ ∅ **then**
  return *Path*
end if
return *0*

ContinousCurve (δ, k, Veh, V, P, G,)
D = Distance(P,G), T = Orientation(P,G)
$\delta = \delta_1 D + \delta_2 T$
$\alpha = \frac{k^2}{2\delta}$, $length = \sqrt{\frac{2\delta}{\alpha}}$, $dt = \sqrt{\frac{\alpha}{\pi}}dl$
$t = dt{:}dt{:}\sqrt{\frac{2\delta}{\pi}}$
$x = \frac{\sqrt{2\pi\delta}}{k}$ *cumtrapz(sin(*$\frac{\pi}{2}t^2$*))dt*
$y = \frac{\sqrt{2\pi\delta}}{k}$ *cumtrapz(cos(*$\frac{\pi}{2}t^2$*))dt*
return *x*, *y*

CrossPoint(D_V, P_W, Veh, q_I, q_G)
[x,y] = the tangent line to the crossed line of extended goal position and the far corner of the parking place in the relative angle of 45° degree.
return *x*, *y*
ThreeGoalPlanner($\delta, k, dl, R_W, W_V, L_{VT}, D_V, P_W, Veh, q_I, q_G$)(3)

(1): As Table [1], (2): due to the limited space please refer to [5] (3): please refer to [4].

Due to the impact of the plans of decision maker of an autonomous parking system on the safety, efficiency and performance of the parking maneuvers, this proposed method was carefully tested and evaluated by Panasonic Automotive Systems Europe GmbH over almost 1000 different parking scenarios. The implemented state machine could successfully plan the correct optimum path with the desired maneuvers (path) for each scenario accordingly according to the provided positions. The running time of the algorithm (MATLAB R2015a) over several test cases was tested as Table 2:

TABLE 2. AVERAGE PERFORMING TIME

Use Case	Success Ratio	Average Running Time (ms)
Wide	100(%)	~330
Normal	95(%)	~450
Narrow	97(%)	~700

VI. CONCLUSIONS

In this paper, a method of path planning for local perpendicular parking scenarios for vision-based auto parking systems was introduced. This method uses a state machine to plan one, two or more-goal parking maneuvers for all perpendicular parking scenarios accordingly. By knowing that the ignoring of unnecessary parking maneuvers in order to save the performing time for an auto parking system is a criterion feature of the planner level, here the algorithm implements the less possible required maneuvers for each parking scenario in order to finish the parking maneuver quickly. The proved average running time of the algorithm shows the reliability of the proposed state machine for real parking scenarios where the auto parking systems struggle with several parameters such as traffic jam, limited parking space and so on.

The similar state machine can be implemented on parallel parking scenarios where an auto parking system needs to terminate the parking scenarios as quick as possible because regular parallel parking spaces are located next to driving roads. Using a state machine which leads the planner to save parking maneuvers can improve the running time of the algorithm to a great extent.

VII. REFERENCES

[1] Community database on Accidents on the Roads in Europe, "Annual Accident Report 2018", European Road Safety Observatory,2018.

[2] The Department's Planning Policy Statement, "Development Control Advice Note 15, 1999 2nd Edition Vehicular Access Standards", 1999.

[3] LaValle, Steven M. Planning Algorithms. Cambridge: Cambridge University Press, 2006. DOI= doi:10.1017/CBO9780511546877

[4] S. Sedighi, D. Nguyen, K. D. Kuhnert, "A New Method of Clothoid-Based Path Planning Algorithm for Narrow Perpendicular Parking Spaces", *5th International Conference on Mechatronics and Robotics Engineering",* Rome, Italy, 2019.

[5] J. A. Reeds and L. A. Shepp, "Optimal paths for a car that goes both forwards and backwards," Pacific Journal of Mathematics, vol. 145, no. 2, pp. 367–393, 1990.

[6] Levien, R. " The Euler Spiral: A mathematical history". University of California at Berkeley, USA, 2008.

[7] T. Fraichard, A. Scheuer, "From Reeds and Shepp's to continuous-curvature paths". IEEE Transactions on Robotics, IEEE, 2004, 20 (6), pp.1025-1035.

[8] D.-V. Nguyen, "Perceptual Inference for Autonomous Navigation", in Südwestdeutscher Verlag für Hochschulschriften, Publisher, ISBN-13: 978-3838139883, 2014. ch. 5.

[9] S. Sedighi, D. Nguyen, K. D. Kuhnert, "Guided Hybrid A-star Path Planning Algorithm for Valet Parking Applications", *The 5th International Conference on Control, Automation and Robotics* , Beijing, China, 2019.

[10] Vorobieva, Hélène, Sébastien Glaser, Nicoleta Minoiu Enache and Saïd Mammar. "Automatic Parallel Parking in Tiny Spots: Path Planning and Control." IEEE Transactions on Intelligent Transportation Systems 16 (2015): 396-410