Path Planning and Tracking Control for an Automatic Parking Assist System

Emese Szádeczky-Kardoss and Bálint Kiss

Dept. of Control Engineering and Information Technology
Budapest University of Technology and Economics, Hungary
szadeczky@iit.bme.hu, bkiss@iit.bme.hu

Abstract. This paper presents two major components of an automatic parking assist system (APAS). The APAS maps the environment of the vehicle and detects the existence of accessible parking place where the vehicle can park into. The two most important tasks the APAS must then realize are the design of a feasible path geometry and the tracking of this reference in closed loop such that the longitudinal velocity of the vehicle is generated by the driver and the controller determines the steering wheel angle which is realized by an Electric Steering Assist System (EPAS). We present in details the solution of the continuous curvature path planning problem and the time-scaling based tracking controller. These components are part of an implemented APAS on a commercial passenger car including an intelligent EPAS.
Keywords. automatic parking assist system, continuous curvature path planning, tracking control, time-scaling

1 Introduction

An important goal of automatic vehicle control is to improve safety and driver's comfort. The APASs provide this for parking maneuvers which are usually performed at low velocities. The APAS collects first information about the environment of the vehicle (position of the obstacles) which are necessary to find a parking lot and to complete a safe parking maneuver [1].

One may distinguish fully or semi-automated APASs according to the driver's involvement during the maneuver. In the fully automatic case the APAS influences both the steering angle and the longitudinal velocity of the car. In the case of vehicles without automatic gear the semi-automatic APAS is the only available option. In this case the driver needs to generate the longitudinal velocity of the car with an appropriate management of the pedals, while the APAS controls the steering wheel.

There exists now several APASs on the market. The Aisin Seiki Co. Ltd. has developed a semi-automatic parking system for Toyota [2] where a camera observes the environment of the car. The Evolve project resulted a fully automatic APAS for a Volvo type vehicle based on ultrasonic sensor measurements [3]. The Volkswagen Touran may be ordered with an option that also assists parking maneuvers [4]. This solution also uses ultrasonic sonars for the semi-automatic parking maneuvers. We do not address in this paper the map making and parking lot detection procedures and the signal processing problems related to the map making from distance measurement, and the position and orientation estimation of the car. The first task is realized in our setup using ultrasonic sensors and the second problem is solved by the use of the angular velocity data of the ABS sensors of the car.

To design an APAS, a mathematical model of the car has to be given. Several vehicle models are presented in the literature including kinematic and dynamic models. Let us suppose that the kinematic models such as the ones reported in [5–7] describe in a satisfactory way the behavior of the vehicle at low velocities where parking maneuvers are executed.

If the model of the vehicle is known, then the path planning method and the tracking controller algorithm can be designed based on the motion equations. To plan the motion one can use deterministic [8] or probabilistic methods [9, 10] as well. To get a continuous curvature path special curves should be used [11]. Algorithms based on soft computing methods can also calculate the reference path [12, 13].

In the case of the semi-automatic systems the design of the tracking controller is more involved than in the fully automatic case. The tracking controller in the fully automatic case may influence the behavior of the car using two inputs whereas one losses one input, namely the longitudinal velocity in the semi-automated case where the car velocity is generated by the driver hence the velocity profile during the execution of the parking maneuver may be considerably different from the one used for the path planning. This problem is addressed by [14] using time-scaling.

Our goal was to develop a parking assist system which can operate both in fully and semi-automatic mode in three different parking situations (parking in a lane, parking in a row, and diagonal parking [15]).

In this paper we present first the components of the system (Sect. 2). The kinematic vehicle model is described in Sect. 3 and the path planning and tracking controller algorithms are detailed in Sect. 4-6. Section 7 presents some results based on tests on a real car and a short summary concludes the paper.

2 Components of the System

To ensure autonomous behavior, several tasks have to be solved: the system should be able to detect obstacles in its environment; it has to measure or estimate its position and orientation; the reference motion has to be planned, and finally, this reference should be tracked as accurately as possible. These tasks are performed by separate interconnected subsystems which are depicted in Fig. 1.

The ABS sensors of the vehicle can detect the displacement of the wheels of the car. Based on these data, an estimator calculates the actual position and orientation of the car in a fixed word coordinate frame. This estimated state is used by the mapping and controller modules. To draw a map, additional data are also required about the environment. Ultrasonic sensors are used to measure the

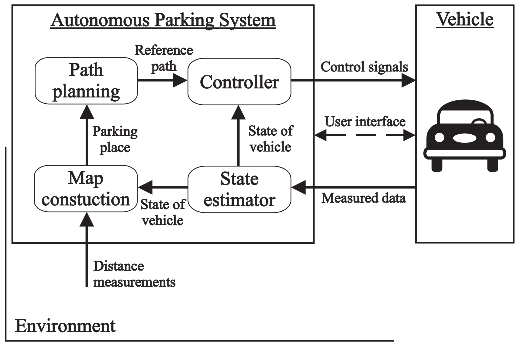

Fig. 1. Components of the Automatic Parking Assist System

distances to the surrounding obstacles. Based on these distance measurements a map can be created. One may then use simple algorithms to detect accessible parking places (if any) on the map.

During the motion planning a reference path is calculated which connects the initial and the desired final configurations. In this planning phase some constraints (e.g. the non-holonomic behavior described by the model, collision avoidance, maximal values of the actuator signals) have to be taken into consideration. Finally, the tracking control algorithm is used to track the reference path.

3 Kinematic Vehicle Model

Both the path planning and tracking control algorithms use the kinematic model of the vehicle. To calculate the geometry of the path a slightly extended model is used in order to ensure the continuous curvature property.

As it is usual in the literature, the reference point of the car is the midpoint of the rear axle denoted by R. The configuration of the vehicle (q) is described by four state variables: position of the reference point R (x, y), the orientation of the car ψ, and the curvature (κ), which is the inverse of the turning radius. Supposing that the Ackermann steering assumptions hold true, the motion of the vehicle can be described by the kinematic model of the bicycle fitted on the longitudinal symmetry axis of the car (see Fig. 2):

$$\dot{q} = \begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\psi} \\ \dot{\kappa} \end{bmatrix} = \begin{bmatrix} \cos\psi \\ \sin\psi \\ \kappa \\ 0 \end{bmatrix} v + \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} \sigma. \tag{1}$$

The longitudinal velocity of the car is denoted by v and the velocity of the change in curvature is given by σ. Let us denote the axle space by b. Then we get the following relationship between the curvature and its derivative:

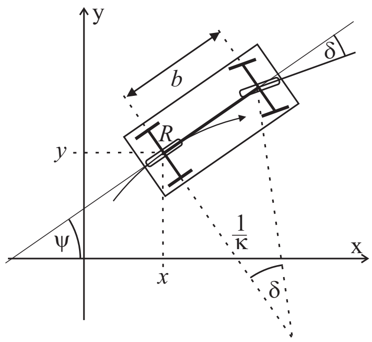

Fig. 2. Kinematic model of the vehicle in the $x - y$ plane

$$\kappa = \frac{\tan\delta}{b}, \quad \sigma = \dot{\kappa} = \frac{\dot{\delta}}{b\cos^2\delta}, \tag{2}$$

where δ is the angle between the front wheel of the bicycle and the longitudinal axis of the car. Both the curvature and the curvature derivative are limited, i.e.

$$|\kappa| \leq \kappa_{max}, \quad |\sigma| \leq \sigma_{max}. \tag{3}$$

To design the tracking controller we use a simpler version of (1) without the curvature.

$$\dot{q} = \begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\psi} \end{bmatrix} = \begin{bmatrix} \cos\psi \\ \sin\psi \\ \frac{\tan\delta}{b} \end{bmatrix} v. \tag{4}$$

In case of the semi-autonomous system the controller cannot consider the longitudinal velocity of the vehicle as a system input since it is generated by the driver and cannot be influenced by the controller. In this case we denote this driver velocity by v_d. (We suppose, that this v_d velocity can be measured or well estimated.)

$$\dot{q} = \begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\psi} \end{bmatrix} = \begin{bmatrix} \cos\psi \\ \sin\psi \\ \frac{\tan\delta}{b} \end{bmatrix} v_d. \tag{5}$$

4 Path Planning

The task of the path planning method is to determine the geometry of the reference path. Our goal is to have reference path with continuous curvature which avoids stopping the car while steering the front wheels. Further constraints are the maximal limit on the curvature and on its time derivative, since the vehicle is not able to turn with arbitrary small turning radius and the change of the turning radius (or the curvature) is also limited by the applied EPAS system.

To plan such a path which fulfills the above mentioned constraints we use three different path primitives namely straight lines, circular segments, and continuous curvature turns (CC turns) [11]. The curvature is zero along a straight line, in a circular segment it has a nonzero constant value, which does not exceed a given maximum limit, and the curvature varies linearly with the arc length in the CC turns.

The geometry of the straight lines or circular segments can be described easily if their parameters are known (e.g. lengths, turning radius). The calculation is more complicated in the case of the CC turns. If the motion is started from the initial configuration $q_0 = [0, 0, 0, 0]^T$, the velocity profile is constant (e.g. $v = 1$), and the curvature changes with the allowable maximum σ_{max} value, then the configurations in a CC turn can be described by the following equations:

$$x = \sqrt{\pi/\sigma_{max}} C_F \left(\sqrt{\kappa^2/(\pi\sigma_{max})} \right), \tag{6}$$

$$y = \sqrt{\pi/\sigma_{max}} S_F \left(\sqrt{\kappa^2/(\pi\sigma_{max})} \right), \tag{7}$$

$$\psi = \kappa^2/(2\sigma_{max}), \tag{8}$$

$$\kappa = \sigma_{max} t, \tag{9}$$

where C_F and S_F denote the Fresnel integrals, which cannot be given in a closed form. Differentiating (6-9) we can see that these equations fulfill the kinematic model of the vehicle given in (1).

Using simple mathematical operations for (6-9) we can get the configurations for motions from different initial configurations with arbitrary constant velocity. If the values of the Fresnel integrals in (6-7) can be calculated beforehand, then the remaining computations can be performed in real time [16].

Now we describe how to use the three path primitives to get the reference trajectory for parking in a lane. In this case we use seven segments to put the path together (see Fig. 3). Without loss of generality we suppose that the vehicle starts the motion backwards from the $q_s = [0, 0, 0, \kappa_s]^T$ initial configuration. First it moves along a CC turn, until the maximal curvature (or the minimal turning radius) is reached. Then it turns along a circle with the minimal turning radius, the corresponding turning angle is denoted by φ_1. After the next CC turn the curvature becomes 0, and the car turns in the opposite direction with a CC turn, a circular motion with turning angle φ_2 and one more CC turn, such that the curvature becomes 0 again. The path ends with a straight line segment whose length is denoted by l.

Such a path has three parameters: the turning angles in the circular segments (φ_1, φ_2) and the length of the straight line (l). If these parameters are known, the geometry of the path can be calculated and the goal configuration can be determined:

$$q_g = \begin{bmatrix} x_g \\ y_g \\ \psi_g \\ \kappa_g \end{bmatrix} = \begin{bmatrix} f_1(\varphi_1, \varphi_2, l) \\ f_2(\varphi_1, \varphi_2, l) \\ f_3(\varphi_1, \varphi_2) \\ 0 \end{bmatrix}. \tag{10}$$

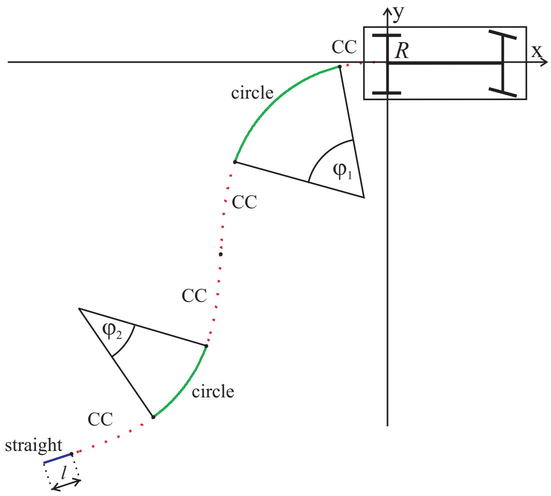

Fig. 3. Reference path for parking in a lane

The values of the path parameters can be determined from the desired goal configuration (q_d) by solving the $q_d = q_g$ equation. So the full reference path can be calculated, and the reference values for the configuration ($q_{ref} = [x_{ref}, y_{ref}, \psi_{ref}, \kappa_{ref}]^T$) and its time derivatives can be determined.

5 Time-Scaling

During the path planning we considered a preliminary reference velocity profile and to avoid involved calculations we supposed that it is constant (e.g. $v = 1$). Since the driver will generate another velocity profile ($v_d \neq v$), it is enough if one is able to track the geometry of the reference of the designed path and the time distribution along the path will be adapted in real-time to the velocity profile generated by the driver.

In the above equations the states of the configuration q were functions of time t, where $\dot{t} = 1$. In a more general form we have a state equation

$$\dot{q}(t) = f(q(t), u(t), w(t)), \tag{11}$$

where u is the vector of the inputs and w denotes the external signals. In our case, which is given by (5), we have $u = \delta$ and $w = v_d$.

We introduce a new scaled time denoted by τ such that τ is used to modify the time distribution along the path. We suggest that the relationship between t and τ should not only depend on the states of the car (as it is usually made in the literature [17]), but also on a new external input, denoted by u_s which is the so-called scaling input:

$$\frac{dt}{d\tau} = \frac{1}{\dot{\tau}} = g(q, u_s, w). \tag{12}$$

Using this time-scaling (11) can be expressed with respect to the time τ:

$$q' = \frac{dq}{d\tau} = \frac{dq}{dt}\frac{dt}{d\tau} = g(q, u_s, w) f(q, u, w). \tag{13}$$

The prime denotes differentiation according to τ, hence $\tau' = 1$.

The time-scaling defined in (12) has to satisfy some conditions:

- $\tau(0) = t(0) = 0$, since the original and the scaled trajectories should start from the same initial configuration;
- $\dot{\tau} > 0$, since time cannot stand or rewind.

During the time-scaling we modify the time distribution along the reference path, which was planned in time τ:

$$x_{ref}(t) = x_{ref}(\tau), \tag{14}$$
$$\dot{x}_{ref}(t) = x'_{ref}(\tau)\dot{\tau}, \tag{15}$$
$$\ddot{x}_{ref}(t) = x''_{ref}(\tau)\dot{\tau}^2 + x'_{ref}(\tau)\ddot{\tau}. \tag{16}$$

The further derivatives and the other state variables can be calculated in t in a similar way. It can be seen from (14) that the time-scaling does not change the geometry of the reference path, only the velocity and the further derivatives are modified if $\dot{\tau} \neq 1$.

6 Tracking Control

In this section only some key features of the tracking controller are discussed, the entire algorithm can be found in details in [14]. The literature suggests several solutions [13, 18] to control the two-input kinematic car given in (4). These methods ensure exponential tracking of the reference path. In our one-input case these algorithms cannot be used without modification since our controller cannot influence the velocity of the vehicle. Our idea is to complement the lost velocity input by the time-scaling input as in (13).

For, the following time-scaling function can be used:

$$\frac{dt}{d\tau} = \frac{u_s}{v_d}. \tag{17}$$

In this case the model equation given in (5) which evolves according to t can be transformed using the time-scaling, and we get

$$q' = \begin{bmatrix} x' \\ y' \\ \psi' \end{bmatrix} = \begin{bmatrix} \cos\psi \\ \sin\psi \\ \frac{\tan\delta}{b} \end{bmatrix} u_s. \tag{18}$$

This scaled model has now two inputs (u_s and δ), hence one of the controllers described in the literature can be used for tracking. The selected method will compute δ and u_s according to the tracking error between the real and the scaled reference trajectories. This δ input is used to control the EPAS steering system and the scaling input u_s influences the time-scaling. The time-scaling function and its derivatives, which are required for (14-16), can be calculated using the following relationships, which are based on (17)

$$\tau(t) = \int_0^t \frac{v_d}{u_s} d\vartheta, \quad \tau(0) = t(0) = 0, \tag{19}$$

$$v_d = \dot{\tau} u_s, \tag{20}$$

$$\dot{v}_d = \ddot{\tau} u_s + \dot{\tau} \dot{u}_s. \tag{21}$$

If the signs of u_s and v_d are the same than the time-scaling function satisfies the $\dot{\tau} > 0$ condition. If one of the two signals equals 0, the car is not controllable. This occurs at the very beginning and at the end of the motion.

The scheme of the closed loop control is depicted in Fig. 4. First the path planning module calculates the reference path in τ. In the next step this reference is scaled based on the longitudinal velocity of the car v_d, which is generated by the driver, and based on the scaling input u_s, which is calculated by the controller. After the time-scaling we have the scaled reference in t. The controller determines its outputs using the difference between the real and the scaled reference trajectories. So the inputs of the vehicle are the longitudinal velocity v_d generated by the driver, and δ, which is calculated by the controller. The output of the car is the position of the reference point and the orientation.

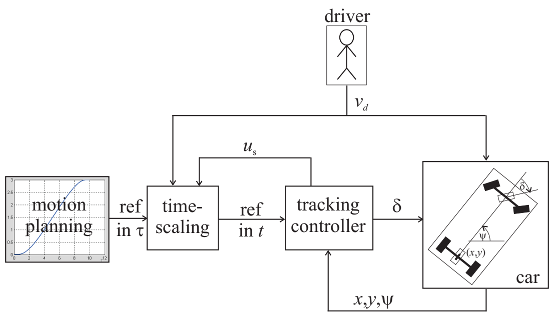

Fig. 4. Scheme of the tracking controller with time-scaling

7 Results

We implemented the presented methods on a Ford Focus type passenger car (see Fig. 5) using an EPAS provided by ThyssenKrupp to realize the steering angle for the front wheels. Both the path planner and tracking controller were realized

Fig. 5. Ford Focus

on a dSPACE hardware (dedicated Autobox) mounted in the car and connected to the EPAS actuator and to the CAN bus to read the ABS signals.

The presented motion is continuous curvature forward parking maneuver. The results are depicted in Fig. 6. The system is able to track the reference path, such that the velocity of the car is not constant, as it was supposed during the path planning.

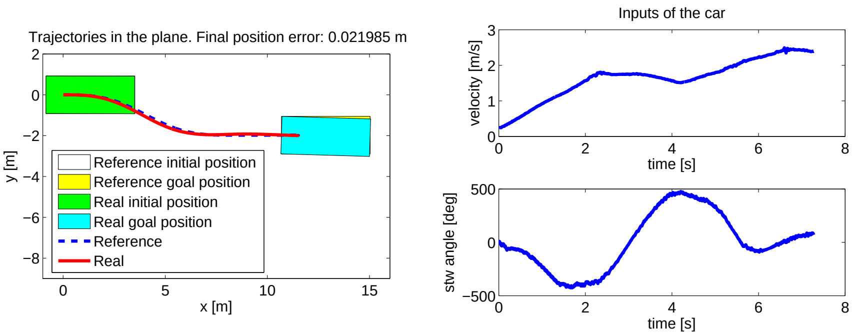

Fig. 6. Results: path in the $x - y$ plane, inputs of the car v_d, δ

8 Conclusions

This paper discussed two components of an APAS. The presented path planning method can calculate a continuous curvature path in real time. The time-scaling tracking controller is able to drive the car along the reference path such that the driver generates the car velocity. The time-scaling function can be calculated from the velocity of the car and from a scaling input, which is calculated by the controller based on the closed loop behavior. The presented algorithms were tested in a real car with encouraging results.

Acknowledgments. The research was partially funded by the Advanced Vehicles and Vehicle Control Knowledge Center under grant RET 04/2004 and by the Hungarian Science Research Fund under grant OTKA K71762.

References

1. Nissan Around View Monitor. http://www.nissan-global.com/EN/TECHNOLOGY/INTRODUCTION/DETAILS/AVM/index.html
2. AISIN SEIKI Co. Ltd. Intelligent Parking Assist. http://www.aisin.com/product/automotive/info/ot.html
3. Evolve. http://www.ikp.liu.se/evolve
4. Volkswagen Park Assist. http://www.volkswagen.com
5. Dubins, L.E.: On curves of minimal length with a constraint on average curvature, and with prescribed initial and terminal positions and tangents. American Journal of Mathematics **79**(3) (1957) 497–516
6. Reeds, J.A., Shepp, L.A.: Optimal paths for a car that goes both forwards and backwards. Pacific Journal of Mathematics **145** (1990) 367–393
7. Boissonnat, J.D., Cérézo, A., Lebond, J.: Shortest paths of bounded curvature in the plane. Journal of Intelligent and Robotics Systems **11** (1994) 5–20
8. Laugier, C., Fraichard, T., Garnier, P.: Sensor-based control architecture for a car-like vehicle. Autonomous Robot **6**(2) (1999) 165–185
9. Svestka, P., Overmars, M.H.: Motion planning for carlike robots using a probabilistic learning approach. International Journal of Robotics Research **16**(2) (1997) 119–143
10. LaValle, S.M., Kuffner, J.J.: Rapidly-exploring random trees: Progress and prospects. In: Proceedings of the 2000 International Workshop on the Algorithmic Foundations of Robotics. (2000) 293–308
11. Fraichard, T., Scheuer, A.: From Reeds and Shepp's to Continuous-Curvature paths. IEEE Transaction on Robotics and Automation **20**(6) (2004) 1025–1035
12. Zhao, Y., Collins, E.G., Dunlap, D.: Design of genetic fuzzy parallel parking control systems. In: Proceedings of the American Control Conference. Volume 5. (2003) 4107–4112
13. Cuesta, F., Ollero, A.: Intelligent Mobile Robot Navigation. Volume 16 of Springer Tracts in Advanced Robotics. Springer, Heidelberg (2005)
14. Kiss, B., Szádeczky-Kardoss, E.: Tracking control of the orbitally flat kinematic car with a new time-scaling input. In: Proceedings of the IEEE Conference on Decision and Control. (to appear)
15. Paromtchik, I.E., Laugier, C.: Motion generation and control for parking an autonomous vehicle. In: Proceedings of the IEEE International Conference on Robotics and Automation. (April 1996) 3117–3122
16. Szádeczky-Kardoss, E., Kiss, B.: Continuous-curvature turns in motion planning for car like mobile robots. In: Proceedings of the International Carpathian Control Conference. (June 2007) 684–687
17. Sampei, M., Furuta, K.: On time scaling for nonlinear systems: Application to linearization. IEEE Transactions on Automatic Control **31** (1986) 459–462
18. Fliess, M., Lévine, J., Martin, P., Rouchon, P.: Flatness and Defect of Nonlinear Systems: Introductory Theory and Examples. International Journal of Control **61**(6) (1995) 1327–1361