Design and Implementation of Parking Control Algorithm for Autonomous Valet Parking

2016-01-0146

Published 04/05/2016

Yonghwan Jeong, Seonwook Kim, and Kyongsu Yi
Seoul National Univ

Sangyong Lee and ByeongRim Jo
LG Electronics Inc

CITATION: Jeong, Y., Kim, S., Yi, K., Lee, S. et al., "Design and Implementation of Parking Control Algorithm for Autonomous Valet Parking," SAE Technical Paper 2016-01-0146, 2016, doi:10.4271/2016-01-0146.

Abstract

This paper represents a parking lot occupancy detection and parking control algorithm for the autonomous valet parking system. The parking lot occupancy detection algorithm determine the occupancy of the parking space, using LiDAR sensors mounted at each side of front bumper. Euclidean minimum spanning tree (EMST) method is used to cluster that information. After that, a global parking map, which includes all parking lots and access road, is constructed offline to figure out which cluster is located in a parking space. By doing this, searching for available parking lots has been finished. The proposed parking control algorithm consists of a reference path generation, a path tracking controller, and a parking process controller. At first, route points of the reference path are determined under the consideration of the minimum turning radius and minimum safety margin with near parking. After route points is determined, the reference path is generated by connecting straight lines and arcs between pre-determined route points. A path tracking controller which determines a desired steering wheel angle is designed by the combination of road curvature based feedforward and feedback linear quadratic optimal control method using 2 DOF bicycle model. In this paper, a desired velocity of vehicle is maintained as constant during the parking process to simplify the path generation problem. Finally, parking process control algorithm determines the sequence of parking and gear change timing. The proposed control algorithm for autonomous valet parking has been validated via computer simulations and successfully implemented on a test vehicle.

Introduction

Recently, autonomous driving and advanced driver assistance systems (ADAS) have been an important research and development topic Especially, park assistant system is one of the most necessary field to produce on market since parking task is not easy for novice drivers but also for experienced drivers because of limitation of view angles and misjudgments of distance. These park assistant systems are based on various set of sensors and infrastructures [1, 2, 3, 4]. Almost commercialized parking assist system has a passive assist function such as alert the driver to obstacle while parking. Even the active park assist system such as smart parking assistant system (SPAS), parking space should be founded by driver. To improve the driver's convenience and enhance the safety in parking lot, automated valet parking system has been extensively researched.

In order to develop an autonomous valet parking system, the main research issues can be classified as two things. One is to determine the surrounding environment of the parked vehicles in parking lot, and another is to generate and track the reference path. To detect surrounding, usually on-board sensors such as camera, laser scanner, are used. At the same time, the localization should be held to achieve parking sequence and represent the environment correctly. Currently, many researches about vehicle localization are focusing on map-aided vehicle localization in automated driving [5]-[7]. This is because the on-board sensors' performance have not reached yet to a satisfactory level. Thus, many researches [5], [6], [8], [9] dealt about building a suitable digital map. In automated parking, one typical case of automated driving, has the strength that parking lot is a suitable infrastructure to provide map information. Therefore, researches about lot occupancy detecting system in infrastructure such as [10], [11], [12] have been held through last decade. These researches usually combine the occupancy detection with the infrastructure system. The advantage of these types are that the vehicle doesn't need complicated detection system of its own. However, when the vehicle has an automated driving systems, because the vehicle already has on-board sensors to perceive, there is no need to compute all the occupancy from the infrastructure. Thus in our work, we separate the occupancy detection algorithm from the infrastructure information and used simple digital map based on the lot's geometry.

Various methods from geometry based method to real-time path planning using optimal control have been used to generate the reference path. In [13], Gomez-Bravo et al. presents β-spline curve based continuous curvature path generation which is previously calculated. In this research, to guarantee the collision free and admissibility of the curves, fuzzy logic with curvature and boundary constraint is considered simultaneously [14]. Cheng et al. divided the path generation problem into two segments which is planned separately using different method [15]. Each segment selects Reeds-Shepp curve, continuous-curvature curve, or half-continuouscurvature curve based on the perception. However, the detail of the environment perception module is not suggested. In [16], Yoon et al. presents the A* algorithm using two heuristics which consider the kinematics of a vehicle with and without obstacles respectively.

In this research, we focus on designing an autonomous parking control algorithm which consist of parking lot occupancy detection, path generation and tracking algorithm. In order to detect the parking lot occupancy, parking lot map which is constructed using DGPS and aviation photo describes the drivable area, goal parking position and free driving path. The environment perception based on LiDAR module detects the parked vehicle or obstacles in the parking slot using EMST method. After that, reference path is generated using Reeds-Shepp curves with consideration of minimum turning radius and safety margin. To track the reference path, 2 DOF bicycle model based feedback LQR controller with feedforward input using curvature is designed to determine a front steering angle input. For longitudinal controller, model free adaptive controller is used to determine throttle and brake input to maintain the constant desired speed. The proposed parking control algorithm is evaluated via vehicle tests.

This paper is constituted as follows: the section II presents an overall architecture of the proposed parking control algorithm. Then Section III, laser scanner based parking lot occupancy detection algorithm is described. In section IV, Reeds-Shepp curves based path generation algorithm is proposed. The controller to track the generated path is designed in section V. Then Section VI shows the vehicle test results to validate the proposed parking control algorithm. Finally, the contribution of this research and the future works are in section VII.

Overall Architecture

The overall architecture of the proposed parking control algorithm, which consist of three stages, perception, path generation and path tracking stage is shown in figure 1. In the perception stage, the occupancy of the parking lot is measured using the measurements from laser scanner which is processed using EMST method. Information of the parking lot such as goal position, heading angle, and dimension of the lot is pre-defined in parking lot map which is constructed using DGPS and aviation photos. Then, the reference path for the parking sequence is generated based on the Reeds-Shepp car in consideration of the front margin and minimum turning radius. To tracking the path, the controller is designed to determine the desired steering wheel angle and desired throttle and brake input. The steering input and the desired throttle/brake input is determined using LQR approach and adaptive controller respectively.

Parking Lot Occupancy Detection

Parking Lot Map Construction

One of the advantage of the parking lot is that the vehicle can use the map information from the infrastructure. Map data can be used in many different ways. In our research, we constructed the map with four information: 1) parking slots' position (x, y, ψ); 2) centerline of the parking lot boundary 3) parking slots' size; 4) on-map localization information. To achieve parking sequence, our parking controller needs the slot's position information to generate the principal points. In the occupancy detection, not only the position of the slot but also the shape of it is essential. Most of the parking lots are consist of some array of rectangular shape. Thus, by adding the length and width of the slots to the map, the map can represent the exact slot shape. While in occupancy detection, the vehicle has to move in drivable area which should not go through the parking slots' line. For all of the above processes, the vehicle position on map is necessary, thus, the map needs the transformation matrix information between global position and the parking lot map. In our system, vehicle's global position is measured with high precision RTK-DGPS that allows us to analyze the accuracy up to 2 cm. So, we used offset between global vehicle position and the reference position on map as the localization information. More specifically, figure 2 shows how the digital map constructed.

For the first step, we parked the test vehicle into predefined slots manually. To gather the reference data, we used RTK-DGPS to measure the positions of the parked slots. After retaining the reference, the next sequence is constructing the digital map using the measured size of the slot and the parking lot's geometry. By comparing the position from the digital map and the DGPS, the offset data can be obtained. In our research, for instance, we measured about four slots, generates the digital map with length 5m and width 2.3m. Finally, we compute the offset of the position for the global-to-map transformation.

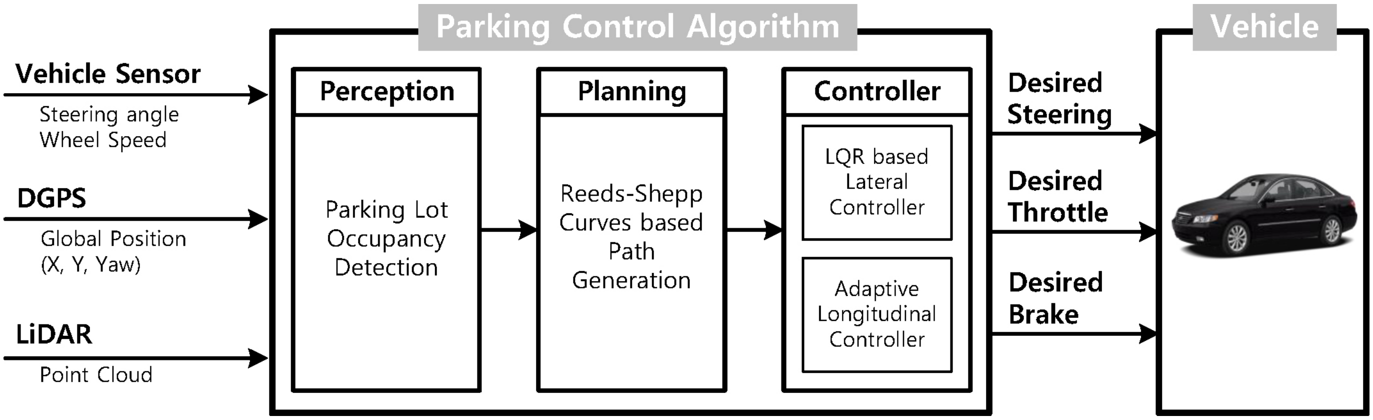
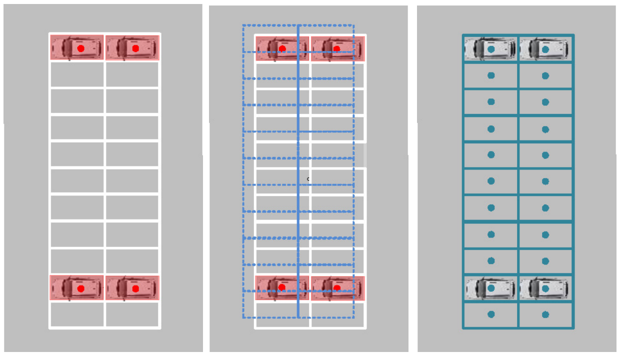


Figure 1. Overall architecture of the proposed parking control algorithm



a. b. c.

Figure 2. Processes of parking slot map construction (a) sampling for vehicle positions; (b) construct digital map from the parking lot's geometry; (c) calibrating digital map based on (a);

After these basic steps are end, we extract the boundaries of the slots and generate the centerlines of them. When generating the centerline, we mainly used the straight lines between the boundary lines and fitted with R=6m curves when there is an edge of the boundaries. Figure 3 shows one example of the constructed map. For the visibility, we add the image of the parking lot as a background. The black dot lines are the slot lines and blue dot lines shows the reference data measured by DGPS. The magenta dot lines and the thick red line are the boundary lines and the centerline, respectively.

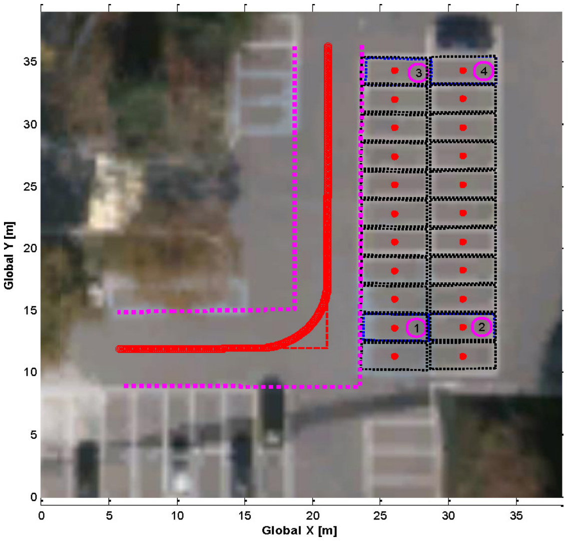


Figure 3. Constructed digital map about the parking lot

Data Processing and Detection Strategy

As we mention in the overall architecture section, two Sick sensors at the vehicle front were used for surrounding perception in our system. For every single step, the system gathers 2164 data points from the sick sensors. Our detection method doesn't need the clustering process if we only detect the occupancy, however, to extend the algorithm such as dealing with surrounding obstacles while in the sequence, we need an obstacle estimation process as an extra. For the real time performance, we used the on grid data instead of the raw data points. In parking sequence, the importance of the lateral and longitudinal length are different. Thus, we decide the lateral and longitudinal grid size as 5 cm, 10 cm each. These grid sizes are valid when the lines of the lot is aligned with the grid axis. To apply grid size properly, we transfer the points into the parking lot line aligned axis. For the clustering process, we used EMST method introduced in [17]. To improve the computational load of the algorithm, we used the distance thresholding function to transfer the Euclidean distances into simple logical numbers. In this paper, $\varepsilon = 0.5m$ is used. The distance function is as follows.

$$f\left(d_{ij}\right) = \begin{cases} 1 & d_{ij} \le \varepsilon \\ 0 & d_{ij} > \varepsilon \end{cases} \tag{1}$$

For each step, we gather the occupancy results by simply comparing the position of the obstacle information with the map constructed in map construction section and represent the available/unavailable cases as 0, 1 each. The perception range of the system is limited, thus the system needs to preserve past perception results. At the same time, to prevent the fault case, we constructed the confidence level by accumulating the occupancy results. Parking sequences are usually held in low speed condition like around 10kph. The algorithm decide the slot unavailable when the confidence level is over 10, which means the slot is decided to be unavailable for 1 sec. By using this confidence system, we guaranteed to maintain the information of the slots that are placed out of the perception range. For the final step, the algorithm gives the available slot when the distance between the vehicle and slot is smaller than predefined distance. In our research, we used 10m as the reference distance. For the vehicle's safety, the algorithm decides the slot as unavailable when the accumulated occupancy result of the slot is positive number. The summarized algorithm is as follows.

Algorithm 1. Overall Structure of Occupancy Detection algorithm

```
$C_{i,t}$ := Accumulated detected result for $i$-th slot at time $t$
$V_{i,t}$ := Detection result for $i$-th slot at time $t$
$S_i$ := Slot's poition & shape information
$O_t$ := Clustered obstacle data at time $t$
$d_{i,t}$ := Distance between the vehicle & $i$-th slot

if $S_i$ and $O_t$ are overlaped to each other
        $V_{i,t} = 1$
else    $V_{i,t} = 0$
end if
$C_{i,t} = C_{i,t} + V_{i,t}$
if $d_{i,t} \le d_{pre}$
        if $C_{i,t} = 0$
                return available
        else return unavailable
        end if
end
```

Reeds-Shepp Curves based Path Generation

Path generation problem for automated parking can be considered as a simple 2D-path generation because the slope of the parking lot is negligible. Also the total path length is relatively short compared to driving on the road, the error which occurs due to the ignoring of the road slope can be ignored. Thus, 2D-path is enough to design the reference path for automated parking control. To generate the reference path, the characteristics of the vehicle motion should be considered not to design the unreasonable path which could not be tracked. If the steering angle is constant and the velocity of the vehicle is enough low not to saturate the tire force, the motion of vehicle can be approximated as a circular motion which is traveled by the center of the rear axle.

This means that the vehicle under the parking process can be regarded as Reeds-Shepp car which has a discrete control input. The inputs for lateral motion are composed of turning left, right in constant radius, and going straight and for longitudinal motion composed of going forward and backward in constant speed. It is proved that only 48 different combination of these motions are enough to fully describe the shortest paths for the Reeds-Shepp car [18]. In view of the parking area, it is reasonable to use the simple path generation algorithm such as Reeds-Shepp car, because a size of parking lot and an interval between each parking lot is formalized by regulation. This means that the path which is defined using the parking lot standard can be utilized for almost every parking lot.

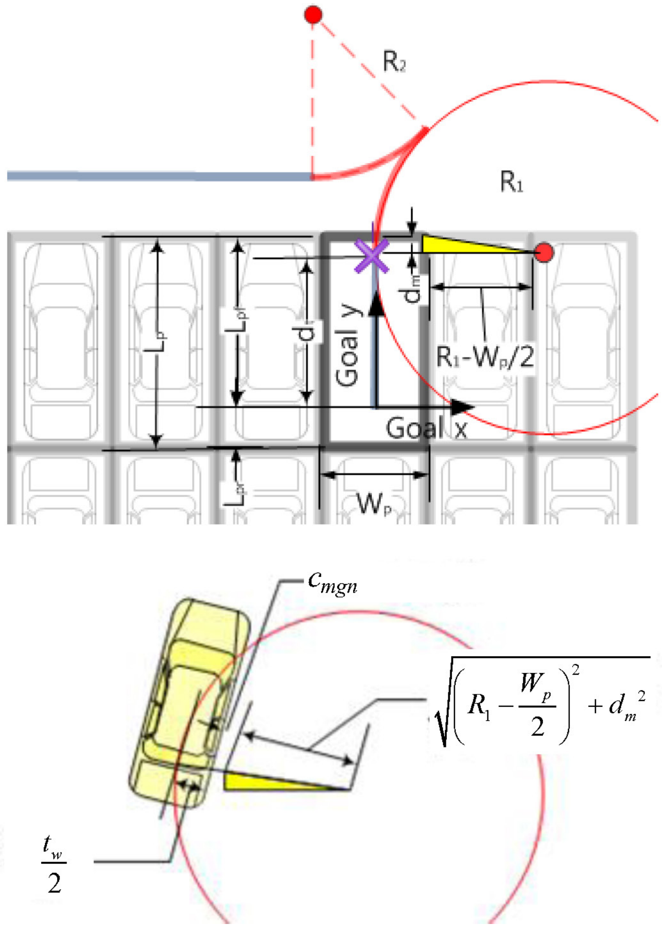


Figure 4. Reeds-Shepp curves based path generation from goal point

Among the 48 Reeds-Shepp curves, Curve-Curve-Straight line (CCS) is used to generate the reference path for perpendicular parking. Three point should be defined in addition to goal point to describe the CCS line in 2D plane. In this paper, the point between curve and straight line is defined as a tangential point, and the point between two curves is defined as a middle point. Another point which is connect between free driving path and curve is defined as a turning point. Therefore, the reference path is fully described if these three points which will be used for mode change criteria are defined. To define these points, the radius (R_1, R_2) the center angle of the arcs (θ) and a distance of straight line (d_t) are calculated. The reference path is generated backward from the position and heading information of goal parking lot, which is predefined the parking lot map. To avoid a collision with a vehicle parked next to the goal parking lot, d_t should be calculated with minimum safety margin (c_{mgn}) as follows:

$$d_m=\sqrt{\left(R_1-\frac{t_w}{2}-c_{mgn}\right)^2+\left(R_1-\frac{W_p}{2}\right)^2} \tag{2}$$

$$d_t=L_{pf}-d_m \tag{3}$$

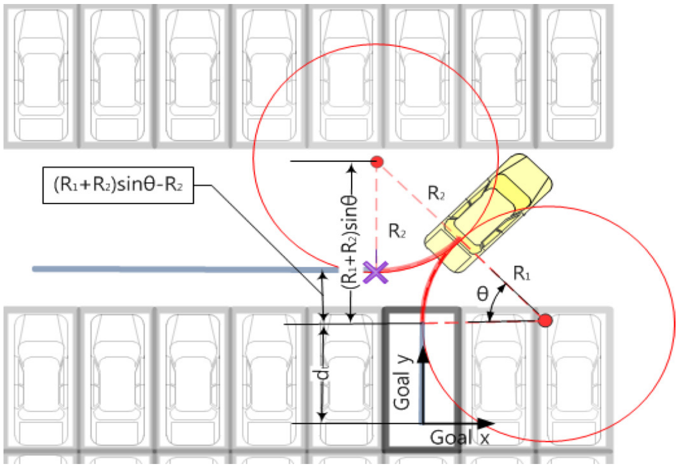


Figure 5. Determination of turning angle θ

where, W_p is the width of parking space, d_m is the longitudinal margin, t_w is the track width of the vehicle and R_1 is the radius of the arc between tangential and middle point.

After d_t is defined, θ which is described in figure 5 can be calculated from the dimension of the parking lot as follows:

$$d_t+(R_1+R_2)\sin\theta-R_2=c_{path}+\frac{t_w}{2}+L_{pf} \tag{4}$$

$$\theta=\sin^{-1}\left(\frac{c_{path}+t_w/2+L_{pf}+R_2-d_t}{R_1+R_2}\right) \tag{5}$$

where, c_{path} is the safety margin between pre-driving path which is indicated as a blue line in figure 4 and parking lot. L_{pf} is the length between the front end of the parking lot and goal point. For simplicity, R_1 and R_2 is assumed to be same and calculated by Ackerman's angle formula.

$$R_1=R_2=\frac{L}{\tan(\delta)} \tag{6}$$

where, L is a wheelbase of the vehicle, δ is a front steering angle. Now, all points which is required to define reference path can be determined as below:

$$\begin{aligned}
x_{tan} &= x_{Goal} \\
y_{tan} &= y_{Goal} + d_t \\
x_{mid} &= x_{tan} + R(1-\cos\theta) \\
y_{mid} &= y_{tan} + R\sin\theta \\
x_{turn} &= x_{mid} - R\sin\theta \\
y_{turn} &= y_{mid} - R(1-\cos\theta)
\end{aligned} \tag{7}$$

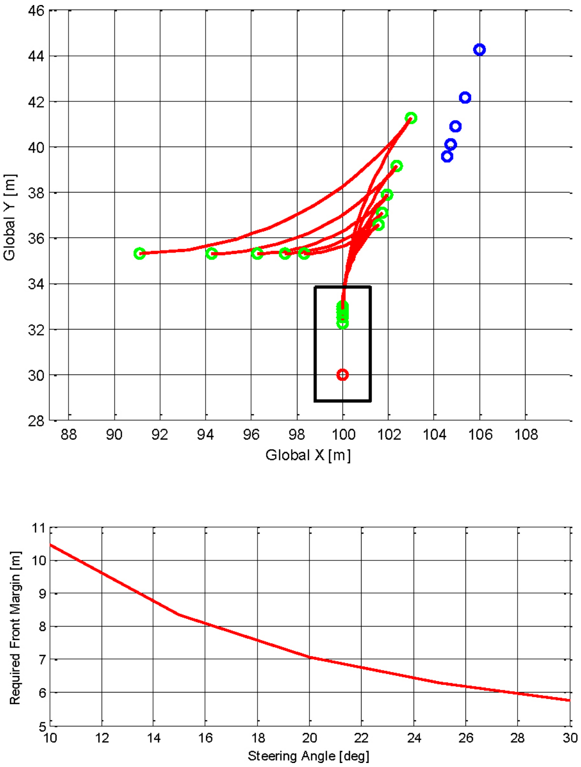


Figure 6. Required front margin according to steering angle

After margin c_{mgn} and c_{path} is properly chosen, steering angle δ is the only parameter which determine required front margin. Figure 6 shows the relationship between required front margin and front steering angle without collisions. The blue circle indicate the front end of the vehicle each path. If available margin is 7m, 20 degree of steering angle is enough to parking without collision.

Vehicle Control

Lateral Controller Design

The lateral controller calculates steering wheel angle, which is sum of feedforward and feedback input, to track the reference path. Generally, error dynamics based feedback controller is used to tracking the path in normal driving condition. However, centimeter level accuracy is needed to park the vehicle without collision because there is not enough safety margin in parking lot. Also steering input is easily increased as maximum value in parking process. This means that if the tracking error is largely increased, it is difficult to reduce error while parking sequence. To deal with this problem, curvature based feedforward input is used to impose the control input before tracking error is generated. 3rd order curve which is fitted using nearest 7 waypoints is used to compute the curvature of the reference path. After that, feedforward steering input is determined using Ackerman's angle formula as follows:

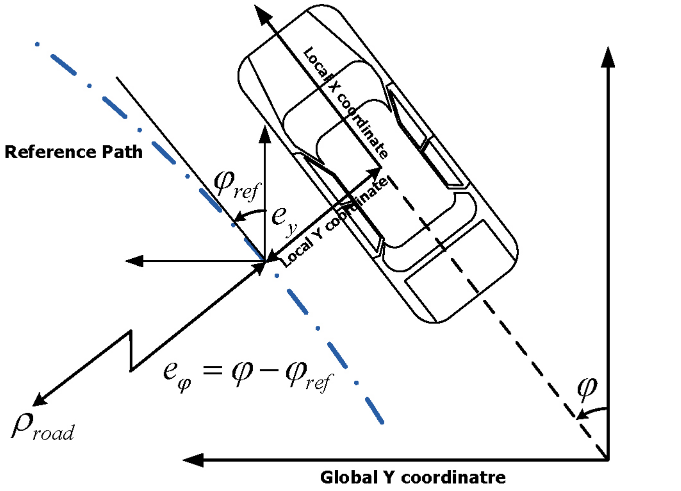


Figure 7. Error dynamics model

$$\begin{aligned}
y &= a_1x^3 + a_2x^2 + a_3x + a_4 \\
\ddot{y} &= 6a_1x + 2a_2 \\
\rho_{road} &= \left| \frac{\ddot{y}}{(1+\dot{y}^2)^{3/2}} \right| \\
\delta_{ff} &= \tan^{-1}\left(L \cdot \rho_{road}\right)
\end{aligned} \tag{8}$$

For feedback input, linear quadratic regulator (LQR) is used to determine optimal feedback control gain [19]-[20]. To formulate the LQR controller, dynamic model for path following is defined using the lateral error between the center of gravity of the vehicle and the nearest point of the reference path and yaw angle error between vehicle heading and yaw angle of the reference path as shown in the figure 7. By considering the infinitesimally small time step, the instantaneous change value of the lateral error and yaw angle error can be approximated as follows:

$$\begin{aligned}
\Delta e_y &\cong v_y\Delta t + v_x\Delta t\sin\left(\varphi - \varphi_d\right) \approx v_y\Delta t + v_x\Delta t\left(\varphi - \varphi_d\right) \\
\dot{e}_y &= v_y + v_x\left(\varphi - \varphi_d\right)
\end{aligned} \tag{9}$$

$$\begin{aligned}
\Delta\varphi_d &\cong \frac{v_x\Delta t}{\rho} \\
\dot{\varphi}_d &= \frac{v_x}{\rho}
\end{aligned} \tag{10}$$

A 2-DOF bicycle model is used to define the state equation of error dynamics. It is reasonable using the 2-DOF bicycle model because the lateral tire force is not saturated in parking process. Lateral dynamics is represented by Newton and Euler equation as follows:

$$\begin{aligned}
m\dot{v}_y + mv_x\dot{\varphi} &= F_{yfl} + F_{yfr} + F_{yrl} + F_{yrr} \\
I_z\ddot{\varphi} &= l_f\left(F_{yfl} + F_{yfr}\right) + l_r\left(F_{yrl} + F_{yrr}\right)
\end{aligned} \tag{11}$$

where, m is the mass of the vehicle and I_z is the moment of inertia of the vehicle. l_f and l_r is the distance between center of gravity and each axle. Lateral tire forces can be written as follows:

$$F_{yfl}=F_{yfr}=C_f\alpha_f=C_f\left(\delta_f-\frac{v_y+l_f\dot{\varphi}}{v_x}\right)$$
$$F_{yrl}=F_{yrr}=C_r\alpha_r=C_r\left(\frac{-v_y+l_r\dot{\varphi}}{v_x}\right)$$

(12)

where, C_f and C_r is the cornering stiffness of front and rear tire. Substituting from the equation (9) and (10) into (11) and (12), the error dynamics can be obtained as follows:

$$\begin{aligned}\dot{e}&=Ae+B\delta+F\dot{\varphi}_d\\&=\begin{bmatrix}0&1&0&0\\0&a_{22}&a_{23}&a_{24}\\0&0&0&1\\0&a_{42}&a_{43}&a_{44}\end{bmatrix}e+\begin{bmatrix}0\\b_2\\0\\b_4\end{bmatrix}\delta+\begin{bmatrix}0\\f_2\\0\\f_4\end{bmatrix}\dot{\varphi}_d\end{aligned}$$

(13)

where

$$\begin{aligned}e&=\begin{bmatrix}e_y&\dot{e}_y&e_\varphi&\dot{e}_\varphi\end{bmatrix}^T\\&=\begin{bmatrix}y_r&\dot{y}_r&\varphi-\varphi_d&\dot{\varphi}-\dot{\varphi}_d\end{bmatrix}^T\end{aligned}$$

$$\begin{aligned}a_{22}&=\frac{-2\left(C_f+C_r\right)}{mv_x},\ a_{23}=\frac{2\left(C_f+C_r\right)}{m},\\a_{24}&=\frac{-2\left(l_fC_f-l_rC_r\right)}{mv_x},\ a_{42}=\frac{-2\left(l_fC_f-l_rC_r\right)}{I_zv_x}\\a_{43}&=\frac{2\left(l_fC_f-l_rC_r\right)}{I_z},\ a_{44}=\frac{-2\left(l_f^2C_f+l_r^2C_r\right)}{I_zv_x}\\b_2&=\frac{2C_f}{m},\ b_4=\frac{2l_fC_f}{I_z}\\f_2&=\frac{-2\left(l_fC_f-l_rC_r\right)}{mv_x}-v_x,\ f_4=\frac{-2\left(l_f^2C_f+l_r^2C_r\right)}{I_zv_x}\end{aligned}$$

Instead of the feedforward input $F\dot{\varphi}_d$ term, the curvature of the road based feedforward input is used as the feedforward input. Therefore, for LQR controller design, only $\dot{e}=Ae+B\delta$ is used to formulate the controller. The performance index is defined as follows:

$$J=\lim_{t_f\to\infty}\frac{1}{t_f}E\left[\frac{1}{2}\int_0^{t_f}\left(e^TQe+\delta_{fb}{}^TR\delta_{fb}\right)dt\right]$$

(14)

where, Q and R are the weighting matrices for error and input efforts.

Longitudinal Controller Design

The longitudinal controller of autonomous vehicle usually designed as two-level structure, upper level controller and lower level controller. The upper level controller decides the desired speed or desired acceleration of vehicle to satisfy the desired motion. After that, the lower level controller calculates proper throttle and brake command to track the desired command of upper level controller. In this paper, the reference path is designed using Reeds-Shepp curves which assumed the desired velocity of the vehicle is constant. Therefore, designing the lower level controller is only needed. Previous studies have tried to develop model based lower level controller. However, the deviation of parameter could lead to the loss of control performance because most of these works rely on an accurate vehicle model. Especially, the nonlinearity of powertrain system is highly increased in low speed region, such as traffic jam condition or parking. To deal with this kind of problem, time-varying parameter adaptive vehicle speed control algorithm is used to improve the desired speed tracking performance under model uncertainty and external disturbance [21].

First, dynamics of powertrain and brake system assumes a first order system, which uses a single time-varying adaptive gain and time varying delay. Before progressing the argument about dynamics of the powertrain and brake, the modified longitudinal vehicle model are defined as follows:

$$\frac{dv(t)}{dt}=\frac{1}{m}\left(F_x(t)-G(t)\right)$$

(15)

where, $F_x(t)$ is sum of front and rear tire forces which are represented in Figure 8. $G(t)$ is the time-varying lumped external disturbance consists of the aero dynamic drag force F_a, rolling resistance R_x and gravitational force $mg\sin\theta$ by road grade as follows:

$$G(t)=F_a(t)+R_x(t)+mg\sin\theta(t)$$

(16)

With the assumption of the no slip condition, the sum of the longitudinal tire force can be expressed as sum of net torque of engine and brake as follows:

$$F_x(t)=\frac{1}{R_gR_{eff}}T_e(t)-\frac{1}{R_{eff}}T_b(t)$$

(17)

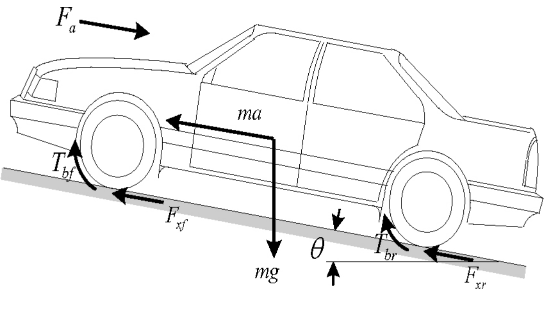


Figure 8. Longitudinal vehicle model

where, R_g is total gear ratio of transmission and final reduction gear, R_{eff} is the effective tire radius, $T_e(t)$s the engine net torque and $T_b(t)$ is the brake torque. As mentioned before, the dynamics of engine and brake can be express and follows:

$$\begin{aligned}T_e(t)&=\frac{k_{th}(t)}{1+\tau_{th}(t)s}u_{th}(t)\\T_b(t)&=\frac{k_b(t)}{1+\tau_b(t)s}u_b(t)\end{aligned}$$

(18)

where, $k_{th}\,(t)$ and $k_b\,(t)$ is a time-varying proportional gain for engine dynamics and brake dynamics. $\tau_{th}(t)$ and $\tau_b(t)$ is the throttle and brake control delay. $u_{th}(t)$ is the throttle input and $u_b(t)$ is the brake pedal position. It should be noted that time varying parameter $k_{th}(t)$ and $\tau_{th}(t)$ are largely depend on the engine speed and transmission gear ratio, respectively. Also time varying parameter $k_b(t)$ and $\tau_b(t)$ is relatively slowly varying.

Finally, the modified longitudinal vehicle model can be represented by combining eq. (2), (4), (5) and (6) as follows:

$$\frac{da(t)}{dt} = \begin{cases} \dfrac{1}{\tau_{th}(t)}\left(-a(t) - G(t) + \lambda_{th}(t)u_{th}(t)\right) - \dot{G}(t) & if\ u_{th}(t) \geq 0 \\ \dfrac{1}{\tau_b(t)}\left(-a(t) - G(t) + \lambda_b(t)u_b(t)\right) - \dot{G}(t) & if\ u_b(t) < 0 \end{cases}$$
$$\lambda_{th}(t) = \frac{k_{th}(t)}{mR_gR_{eff}},\ \lambda_b(t) = \frac{k_b(t)}{mR_{eff}},\ G(t) = \frac{F_a(t) + R_x(t) + mg\sin\theta(t)}{m} \tag{19}$$

In this vehicle model, the external disturbance and model uncertainty are included as time-varying variables of $\lambda_i \in \{\lambda_{th}(t), \lambda_b(t)\}$ and $G(t)$.

The adaptation algorithm to estimate time-varying lumped variables have been proposed to compensate the acceleration error. The schematic diagram for time-varying parameter adaptive speed controller is depicted in Figure 9. Using the linear vehicle dynamics in eq. (7), following throttle and brake control law is designed as follows.

$$u(t) = \begin{cases} \dfrac{1}{\hat{\lambda}_{th}(t)}\left(a_{des}(t) + \hat{G}(t)\right) & if\ \ a_{des}(t) + \hat{G}(t) \geq h \\ \dfrac{1}{\hat{\lambda}_b(t)}\left(a_{des}(t) + \hat{G}(t)\right) & if\ \ a_{des}(t) + \hat{G}(t) < h \end{cases} \tag{20}$$

where $a_{des}(t)$ is desired acceleration, $\hat{\lambda}_{th}(t)$, $\hat{\lambda}_b(t)$ and $\hat{G}(t)$ are the estimated value of $\lambda_{th}(t)$, $\lambda_b(t)$ and $G(t)$, respectively. h is a switching margin to prevent frequent switching between throttle and brake. The proposed control law will lead the acceleration error to converge zero in the case the estimation error ideally converges to zero. The parameter adaptation algorithms for the time-varying parameters, $\lambda_{th}(t)$, $\lambda_b(t)$ and $G(t)$, are represented as follows:

$$\begin{aligned} \dot{\hat{\lambda}}_{th}(t) &= -k_1 u_{th}(t) e_1(t) \\ \dot{\hat{\lambda}}_b(t) &= -k_2 u_b(t) e_1(t) \\ \dot{\hat{G}}(t) &= k_3 e_1(t) \end{aligned} \tag{21}$$

where, $e_1(t)$ is the acceleration error, $a_{des}(t)$-$a(t)$. k_1, k_2 and k_3 are positive adaptation gains of $\hat{\lambda}_{th}(t)$, $\hat{\lambda}_b(t)$ and $\hat{G}(t)$, respectively. The desired acceleration command has been generated as follows:

$$a_{des} = k_p\left(v_{des} - v\right) \tag{22}$$

where, k_p is a proportional gain, v_{des} is the desired velocity and v is the actual velocity.

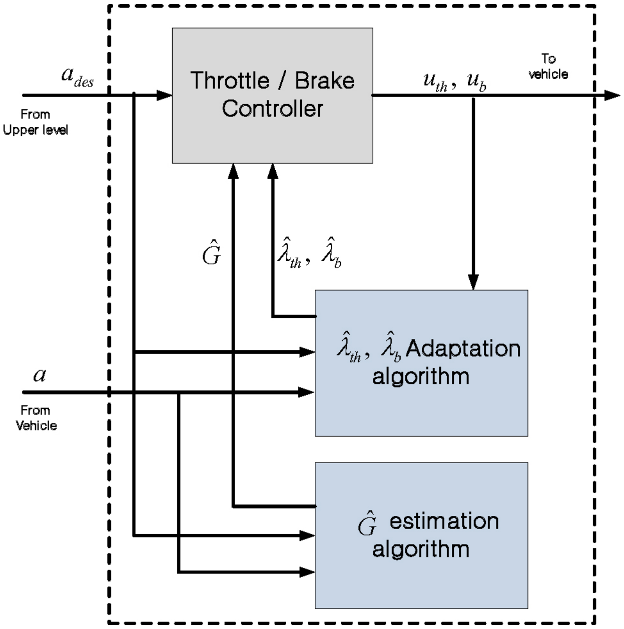


Figure 9. Time-varying parameter adaptive speed controller

Vehicle Test Results

The proposed algorithm has been evaluated by vehicle test using the test platform whose configuration is shown in Figure 10. Three LiDAR sensors, one is four-layer and others are one-layer, are equipped in the front side of the test platform to percept surrounding environment. An around view monitoring (AVM) system and ultrasonic sensors are also equipped to extract lane information and detect a relatively close obstacles. Among these environment sensor modules, only two one-layer LiDAR are used to measure the occupancy of the parking lots. To measure a global position of the test vehicle, RT 3002, differential global positioning system (DGPS) device, has been used. An industrial computer has been used to process the measurement of vehicle sensors and LiDAR, implement the proposed algorithm and record the vehicle test data. Micro-Autobox II functions as a lower level controller which determine the throttle and brake input to track a desired velocity. For steering actuator, motor driven power steering (MDPS) and control board are additionally installed. Smart motor and engine management system (EMS) board also installed for longitudinal control

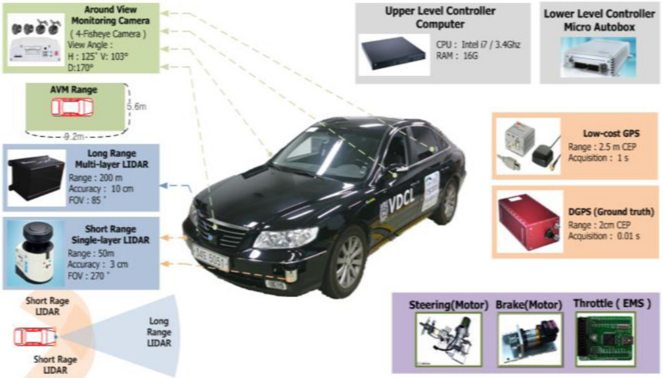


Figure 10. Configuration of test vehicle

The experimental results is shown in figure 11 to 13. In this paper, the bottom three rows are not considered as a target goal parking spaces. A snapshots of the parking lot occupancy detection algorithm is represented in figure 13. Each LiDAR measurements are illustrated as orange and cyan points. Also, the parking lot which is not available to park is indicated red box with 'X' mark. As shown in figure 13, there is no wrong decision between adjacent parking spaces about which parking lot is empty. After that, the goal parking lot is determined as shown in figure 13 (d).

In figure 11 which shows the trajectory of the vehicle, the free driving way point and Reeds-Shepp curve is represented as magenta and red line respectively. The route points and goal point is represented as a green circles. As shown in figure 12, the overall path tracking error is maintained less than 0.3m except the mode change timing. However, exceptional increase of position error was observed at 40 sec. This is because the switching of the reference path is occurred.

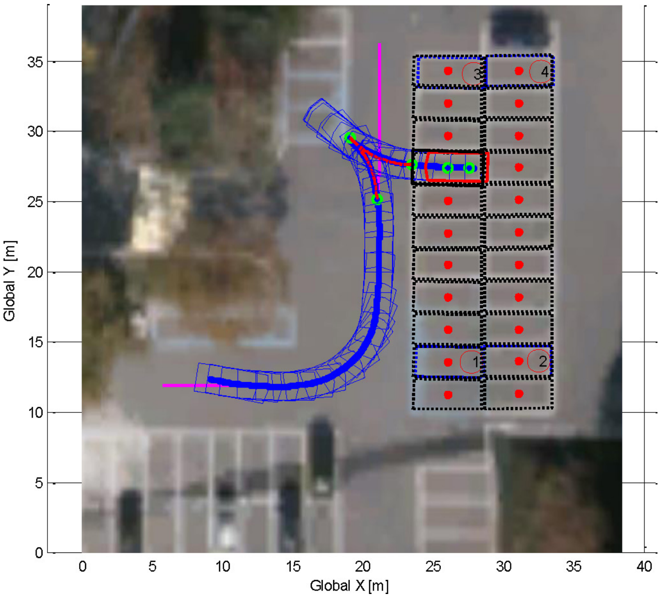


Figure 11. Trajectory of vehicle tests

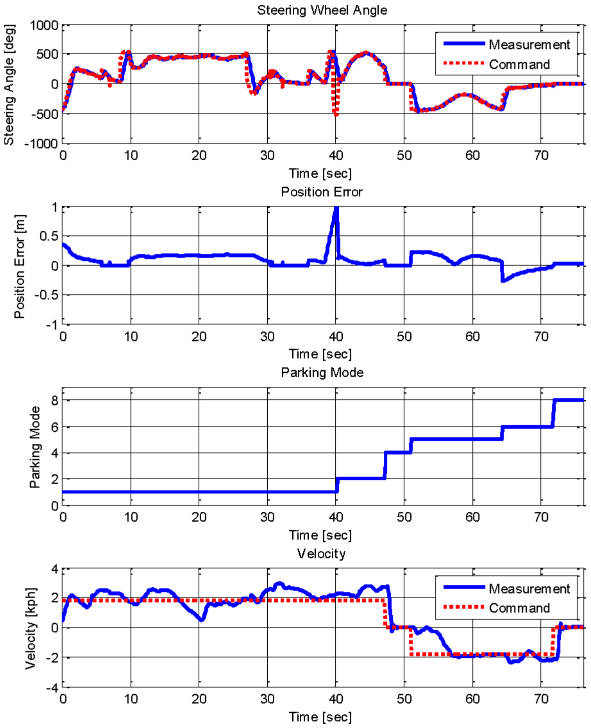


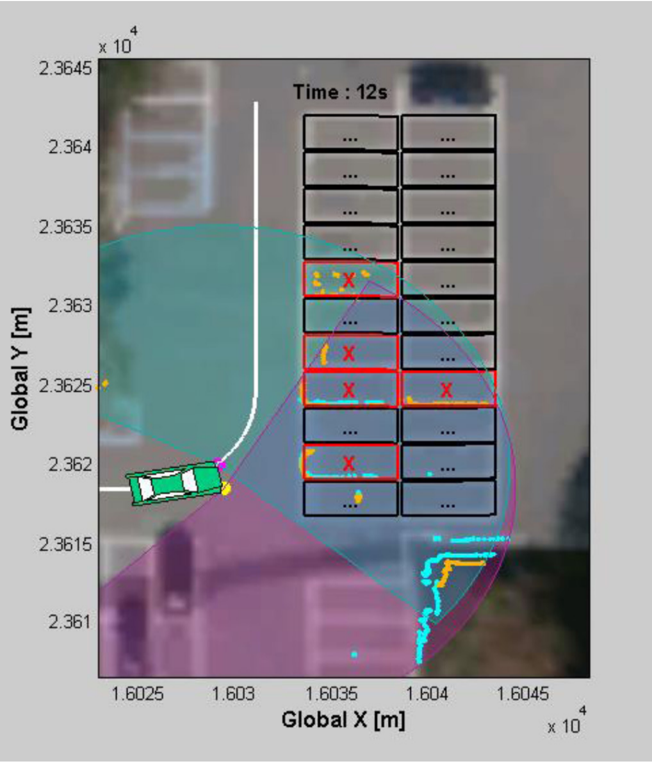
Figure 12. Vehicle test results

Table 1. Parking mode representation

Mode 1	Pre-defined path following and go to turning point
Mode 2	Go to middle point
Mode 3	Stop at middle point
Mode 4	Change gear D to R
Mode 5	Go to tangential point
Mode 6	Go to goal point
Mode 7	Stop at goal point
Mode 8	Parking complete

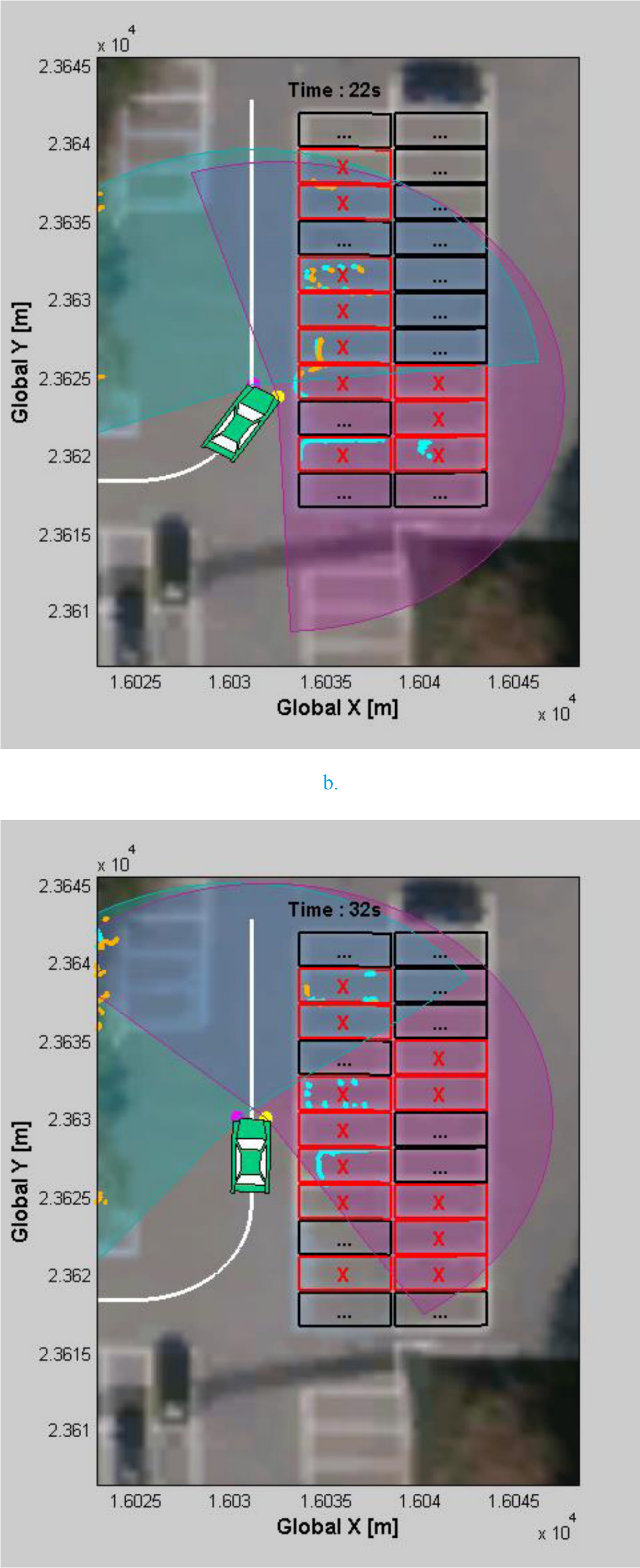
Summary/Conclusions

An autonomous valet parking algorithm based on Reeds-Shepp curves has been developed to enhance driver's convenience and prevent the collision. To avoid the collision, Reeds-Shepp curves has been designed to reflect the geometry of the parking lot. Then, LQR controller and model free adaptive controller is designed to compute the desired steering angle, throttle and brake input to track the desired velocity during the parking sequence. The performance of the proposed parking control algorithm is evaluated via vehicle test. In the future, the perception module will be replaced by mass production sensor such as camera and ultrasonic sensor. Also localization algorithm should be included.



a.

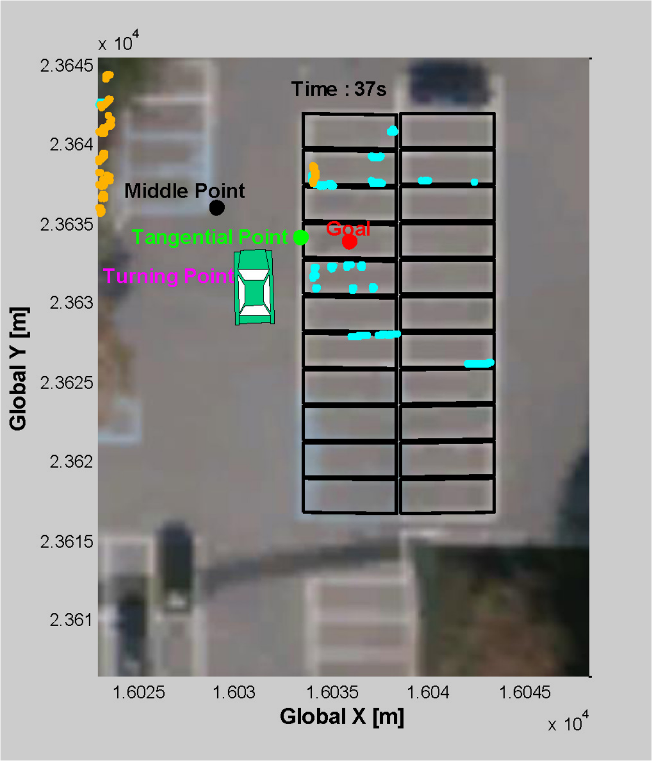
Figure 23. Snapshots of parking lot occupancy detection algorithm



b.

c.

Figure 23 (cont). Snapshots of parking lot occupancy detection algorithm



d.

Figure 23 (cont). Snapshots of parking lot occupancy detection algorithm

References

1. Kim, W., Kim, D., Yi, K., and Kim, H., “Development of a path-tracking control system based on model predictive control using infrastructure sensors,” *Vehicle System Dynamics* 50(6):1001-1023, 2012, doi:10.1080/00423114.2011.597864

2. Wang, H., and He, W., “A Reservation-based Smart Parking System,” IEEE Conference on Computer Communications Workshops, 2011, doi:10.1109/INFCOMW.2011.5928901

3. Wang, C., et al. “Automatic Parking Based on Bird’s Eye View Vision System,” *Advances in Mechanical Engineering* 6, 2014, doi:10.1155/2014/847406

4. Sans, M., “X-by-Wire Park Assistance for Electric City Cars,” Electric Vehicle Symposium and Exhibition (EVS27), 2013, doi:10.1109/EVS.2013.6915008

5. Schreiber, Markus, Knoppel Carsten, Franke Ulrik. "Laneloc: Lane marking based localization using highly accurate maps." Intelligent Vehicles Symposium (IV), 2013 IEEE, doi: 10.1109/IVS.2013.6629509.

6. Tao Z., Bonnifait Ph., Fremont V., Ibanez-Guzman J., "Lane marking aided vehicle localization." Intelligent Transportation Systems-(ITSC), 2013 16th International IEEE Conference on, doi: 10.1109/ITSC.2013.6728444.

7. Gruyer Dominique, Belaroussi Rachid, Revilloud Marc. "Map-aided localization with lateral perception." Intelligent Vehicles Symposium Proceedings, 2014 IEEE, doi: 10.1109/IVS.2014.6856528.

8. Suganuma. N., Uozumi T.., "Precise position estimation of autonomous vehicle based on map-matching." Intelligent Vehicles Symposium (IV), 2011 IEEE, doi: 10.1109/IVS.2011.5940510.

9. Cui Dixiao, Xue Jianru, Du Shaoyi, Zheng Nanning, "Real-time global localization of intelligent road vehicles in lane-level via lane marking detection and shape registration." Intelligent Robots and Systems (IROS 2014), 2014 IEEE/RSJ International Conference on, doi: 10.1109/IROS.2014.6943267.

10. Fabian, T. "An Algorithm for Parking Lot Occupation Detection." Computer Information Systems and Industrial Management Applications, 2008. CISIM '08. 7th, doi: 10.1109/CISIM.2008.53.

11. Bulan Orhan, Loce Robert P., Wu Wencheng, Wang YaoRong, et al. "Video-based real-time on-street parking occupancy detection system." *J. Electron. Imaging*. 22(4), 041109 (Aug 12, 2013), doi: 10.1117/1.JEI.22.4.041109.

12. Ichihashi H., Notsu A., Honda K., Katada T., et al. "Vacant Parking Space Detector for Outdoor Parking Lot by Using Surveillance Camera and FCM Classifier." Fuzzy Systems, 2009. FUZZ-IEEE 2009. IEEE International Conference on, doi: 10.1109/FUZZY.2009.5277099.

13. Gomez-Bravo, F., Cuesta, F., Ollero, A., and Viguria, A., "Continuous curvature path generation based on β-spline curves for parking manouevres," *Robotics and autonomous systems*, 56(4):360-372, 2008, doi:10.1016/j.robot.2007.08.004

14. Gomez-Bravo, F., Cuesta, F., and Ollero, A., "Parallel and diagonal parking in nonholonomic autonomous vehicles", *Engineering applications of artificial intelligence*, 14(4):419-434, 2001, doi:10.1016/S0952-1976(01)00004-5

15. Cheng, K., Zhang, Y., and Chen, H., "Planning and Control for a Fully-automatic Parallel Parking Assist System in Narrow Parking Spaces," IEEE Intelligent Vehicles Symposium (IV), 2013, doi:10.1109/IVS.2013.6629669

16. Yoon, S., Yoon, S. E., LEE, U., and Shim, D. H., "Recursive Path Planning Using Reduced States for Car-Like Vehicles on Grid Maps", *Transactions on Intelligent Transportation Systems* 16(5):2797-2813, 2015, doi:10.1109/TITS.2015.2422991

17. Wang D.Z., Posner I., Newman P., "What could move? Finding cars, pedestrians and bicyclists in 3D laser data." Robotics and Automation (ICRA), 2012 IEEE International Conference on, doi: 10.1109/ICRA.2012.6224734.

18. Reeds, J., and Shepp, L., "Optimal paths for a car that goes both forwards and backwards," *Pacific journal of mathematics*, 145(2):367-393, 1990, doi:10.2140/pjm.1990.145.367

19. Snider, J., M., "Automatic Steering Methods for Autonomous Automobile Path Tracking," Pittsburgh: Robotics Institute, 2009.

20. Rajamani, R., *Vehicle Dynamics and Control*, New York: Springer-Verlag, 2005.

21. Kim, H., and Yi, K., "Combined throttle and brake control for vehicle cruise control: a model free approach," IEEE Intelligent Vehicle Symposium (IV), 2013, doi:10.1109/IVS.2013.6629574

Acknowledgments

This research was supported by LG Electronics, the Korea Ministry of Land, Infrastructure, and Transport; the Korea Agency for Infrastructure Technology Advancement (Project No.: 15PTSIC054118-07)

The Engineering Meetings Board has approved this paper for publication. It has successfully completed SAE's peer review process under the supervision of the session organizer. The process requires a minimum of three (3) reviews by industry experts.

Positions and opinions advanced in this paper are those of the author(s) and not necessarily those of SAE International. The author is solely responsible for the content of the paper.

ISSN 0148-7191

http://papers.sae.org/2016-01-0146