A Cloud Based and Real Time Kinematic Sensing Solution for Automated Parking Function

David BARBÉ*, Xin YANG, Zeng-hong CHEN, Hao YE, Ting CHEN, Si-yuan FAN, Xiao-yuan CAI, Ya-ting GUO and Tao YANG

Department of Electrification, Transmission and Vehicle, AVL Technical Center (Shanghai) Co., Ltd, China

*Corresponding author

Keywords: Autonomous Driving, Vehicle Dynamic, Deep learning, Motion Control, Real Time Kinematic, Infrastructure, Connectivity, Internet of Things.

Abstract. Automated Valet Parking is foreseen as a highly automated function to be implemented on large scale in mass production in the near future. For such function, the authors have developed an innovative system concept using both real time kinematic sensing system and infrastructure IoT (Internet of things) camera to perform AVP function. As the driver activates the AVP function, the infrastructure identifies available parking spots by processing in real time camera frames and as the path is computed, the vehicle executes automated parking maneuvers. This paper introduces the system scheme and system engineering approach as well as key technology bricks to realize AVP function and proves the effectiveness and accuracy of the system through extensive vehicle testing. From system definition to final prototype validation, through Motion control algorithm description or deep learning algorithm tuning method, this paper covers a large spectrum of the engineering domains required to implement such function. The development approach is focused on a user perspective by emphasizing simplicity of usage, frugality of the solution and robustness of the system.

Introduction

Automatic parking system has been developing for several decades and is now already equipped in many types of vehicles. In recent years, a further step has been taken to move from automatic parking to automated valet parking (AVP). With the AVP function, the system is able to detect the parking spot automatically. The driver leaves the vehicle at the entrance of the parking lot and the car parks itself. When the driver wants to take the car again, the car drives itself out of the parking spot and moves to the driver automatically.

For AVP Function different scenarios exist. To simplify parking lot can be either an out-door one or a multi-storey parking facility or underground. According to the different types of parking lot, the localization method of the vehicle has to be different, as VTT uses Ultra-Wideband based localization[1] and BOSCH uses the infrastructure-camera based one[2].

The approach in this paper mainly focuses on the out-door parking environment. According to [3], the concept presented in this paper represents a mixed approach between an RTK (Real Time Kinematic) based localization and infrastructure-camera based parking spot monitoring. The V2X approach is adapted for sharing information between infrastructure and the vehicle via a Cloud server.

This paper introduces an overview of the system and the system architecture. A specific description is provided on implementation of the collision monitoring system and parking spot monitoring system. Eventually motion control method and performance results of the system as well as its limitations are herein presented.
System Schematic

Pursuer (the vehicle system’s name) is based on a BAOJUN E100 Vehicle controlled with AVL’s in-house VCU hardware and software. Fig. 1 represents the system decomposition of the so called AVP system. This complex system is decomposed in 3 main domains: Vehicle, Infrastructure (also described as Parking Area Manager) and HMI. Vehicle carries on the Both the motion control system for longitudinal and lateral dynamic control as well as sensing system used for localization and collision monitoring. Infrastructure is responsible for centralizing the communication via a Cloud Server and monitoring of Parking spot. Customer controls AVP System using portable HMI via 4G communication (Cellphone APP).

As shown in Figure 2, radar is installed in front of the car and two antennas are installed on rooftop of the car. The IMU module is installed under the driver’s seat. The VCU determines both longitudinal control by means of electric drive torque request and lateral control by bypassing the ESP PWM request leading to a steering control.

Specific System Functions Description

Emergency Braking Models

There are two types of brake strategy to avoid collision including TTC (Time to collision) and SD (Safe distance) models. Additionally, two types of TTC models are typically used defined as follow:
The $t_{TTC}$ is the time that it will take a subject vehicle to collide with the target vehicle assuming the relative velocity remains constant, as given in the following equation:

$$t_{TTC} = -\frac{x_c}{v_r}$$  \hspace{1cm} (1)

Where $v_r$ represents the relative velocity between subject vehicle and target vehicle.

The $t_{ETTC}$ is the time for a subject vehicle to collide with the target vehicle assuming the relative acceleration between the subject vehicle (SV) and target vehicle (TV) remains constant, as given in the following equation:

$$t_{ETTC} = \left[\frac{-(v_{TV} - v_{SV}) - \sqrt{(v_{TV} - v_{SV})^2 - 2(a_{TV} - a_{SV})x_c}}{(a_{TV} - a_{SV})}\right]$$  \hspace{1cm} (2)

Where $v_{TV}$ and $v_{SV}$ represent the velocity of target vehicle and subject vehicle, respectively; $a_{TV}$ and $a_{TV}$ represent the acceleration of target vehicle and subject vehicle, respectively; $x_c$ represents distance between subject vehicle and target vehicle.

In case of a SD model, ADAS controller calculates a safe distance to avoid collision via braking.

$$D = s_1 + s_0 - s_2$$  \hspace{1cm} (3)

Where $D$ refers to the distance between subject vehicle and target; $s_1$ refers to the brake distance of subject vehicle; $s_2$ corresponds to the braking distance of target vehicle; $s_0$ refers to minimum clearance between subject vehicle and target vehicle. As shown in Figure 3.

---

For implementation simplicity reason TTC model has been chosen for emergency braking strategy design.

**Obstacle Monitoring**

To ensure safety of the system usage in the geofenced area of AVL Shanghai Technical Center, a collision monitoring system has been implemented to avoid collision with pedestrian or vehicles. The subject vehicle is able to monitor the obstacle in front. When the subject vehicle detects stationary pedestrian via radar and meets emergency braking conditions, the subject vehicle would brake automatically, so that the vehicle reaches a full stop to avoid collision\(^4\).

The collision monitoring system evaluates output of radar and determines whether to activate the emergency braking flag if there’s an obstacle located in the predicted driving path of the vehicle, i.e. in Region of Interest (ROI) as shown in Fig. 4.
Fig. 5 elaborates the braking judgement process. Where $X_c$ represents the calibration value of distance between subject vehicle with obstacle, $v_c$ represents the speed of subject vehicle. ROI is the so called Region of Interest. When the obstacle appears in the ROI, the collision monitoring system activates braking flag which triggers a braking torque request to the eDrive system by the VCU Controller.

**Parking Spot Monitoring**

Function of parking spot identification and localization is handled by Infrastructure. A camera is set up (Fig. 6) to monitor the parking spots. The camera faces towards and observes several parking spots.

Bibi\textsuperscript{[5]} uses infrastructure-camera based system with traditional computer vision to identify if a parking spot is available. In our approach, deep learning technics has been integrated. It allows detection of vehicle or humans in the surrounding of the parking spots.

Existing algorithms have been chosen based on their performance in this case YOLOv3\textsuperscript{[6]}, with pretrained weights.

**Object Detection and Confidence Threshold.** An object-detection network like YOLOv3 can simultaneously output the location of an object, the classification including confidence of this classification of an object for every detected object in the input image, as visualized in Figure 5 with red boxes.

The number of false positive and false negative detections could be further reduced for a trained neural network. A threshold is applied to filter out some detections with low confidence. 119 images with up to 10 objects per image are used for testing. Applying different thresholds, the
number of false and missed detections delivers the result that setting threshold to 0.4 gives us least error (Fig. 8). This threshold is not a generic value and should not be applied in other neural networks.

![Figure 8. Relation between confidence threshold and False/Missed detections.](image1)

![Figure 9. Overlap between two boxes.](image2)

**Availability Calculation.** Calculation of the availability of a parking spot is based on the object detection above, represented as red boxes and pre-defined parking spot region, as blue boxes (Fig. 7 and Fig. 9). The parking spot region is denoted as $S_a$, the detected object as $S_b$. The $S_c$ is the overlap area of $S_a$ and $S_b$. A parking spot is not available if it fulfills the formula below.

$$\frac{S_c}{S_a} > TH_1 \text{ or } \frac{S_c}{S_b} > TH_2$$

(4)

$TH_1$ is the threshold for the first term and $TH_2$ for the second term. The first term filters out detected neighbor car which might also intersect a bit with $S_a$. The second term makes sure even small object (like human) could also set the parking spot as unavailable.

**Vehicle Motion Control**

Parking maneuver is mainly based on GPS localization and uses pursuit algorithm to follow recorded waypoints from the path planner. When GPS localization capability is limited (Not completely open environment), Chassis Information Based Control (CIBC) is selected, which control steering angle according to vehicle previous location and real-time calculated moving distance.

**Controller Principle**

**A Pure Pursuit Algorithm.** Pure pursuit model is a geometrical methodology that calculates desired steering angle based on current position and target position as shown in Fig. 10). It is widely used for tracking pre-defined path with constant vehicle speed [7][8].

![Figure 10. Sketch map of pure pursuit.](image3)

![Figure 11. Sketch map of parking.](image4)

$$\delta = \arctan(L \ast \gamma) = \arctan\left(\frac{2 \ast L \ast (d \ast \cos(\psi_e)) - \sqrt{L_d^2 - d^2 \ast \sin(\psi_e)}}{L_d^2}\right)$$

(5)
\gamma, \text{ Curvature of moving path (m}^{-1}\text{);} \delta, \text{ Wheel steering angle (rad);} \ d, \text{ Shortest distance between current position and target path (m);} \psi_e, \text{ Error between current and target heading angle (rad);} \ L_d, \text{ The distance between current position and target position (m);} \ L, \text{ Wheelbase of vehicle (m).} \\

**B Chassis Information Based Control.** CIBC uses wheel speed and steer angle to calculate vehicle’s position referring to its original position. \([x_0, y_0, \psi_0]\) is the initial position of vehicle which can be acquired from GPS. Lateral motion control of the system is then relying on these estimates to control the steering angle.

\[
\begin{align*}
[x(t), \ y(t), \ \psi(t)] &= f(v_{\text{wheel}}, \ \delta) + [x_0, y_0, \psi_0] \\
x(t) &= \int_0^t (v_{\text{wheel}} * \eta * \cos(\psi)) * dt + x_0 \\
y(t) &= \int_0^t (v_{\text{wheel}} * \eta * \sin(\psi)) * dt + y_0 \\
\psi(t) &= \int_0^t \left(\frac{v_{\text{wheel}} * \eta * \tan(\delta)}{L}\right) * dt + \psi_0 
\end{align*}
\]

\(x, \text{ Lateral position (m);} y, \text{ Longitudinal position (m);} \psi, \text{ Vehicle heading angle (rad);} v_{\text{wheel}}, \text{ Vehicle wheel speed (m/s);} \eta, \text{ Ratio between wheel and vehicle speed.} \\

**Parking Maneuver Using CIBC**

When customer select ‘go park’ on HMI, the vehicle automatically moves to the stop point A based on the available parking spot. When vehicle arrives at stop point A, shifter position will be switched from D to R automatically, and then perform reverse maneuver. When vehicle drives backward to steering point B, steering wheel turns to its max angle. Steering point B is determined based on the heading angle and distance error at stop point A. Keep reverse driving with maximum steering angle until the heading angle of vehicle close to the heading of the parking spot at point C.

\(S_{AB}: \text{ Target reverse straight drive distance in lateral direction, which is a calibration parameter considering } \psi_e \text{ and } d. \)

\[
S_{AB} = f(\psi_e, d) \\
D_{AB}: \text{ Actual reverse straight drive distance in lateral direction.} \\
D_{AB}(t) = |x_B - x_A| = \left| \int_{t_A}^{t_B} (v_{\text{wheel}} * \eta * \cos(\psi_e)) * dt \right| \\
\psi(t) = \int_{t_A}^{t_B} \left(\frac{v_{\text{wheel}} * \eta * \tan(\delta)}{L}\right) * dt + \psi_A
\]

\(t_A, \text{ Actual time at point A (s);} t_B, \text{ Actual time at point B (s);} \psi_A, \text{ Actual heading angle of vehicle at point A (rad).} \\

Steering angle can be determined based on the calculated value of \(S_{AB}, D_{AB} \text{ and } \psi \text{ as follows.} \)

\[
\delta = \begin{cases} 
0 & \text{if } D_{AB} < S_{AB} \\
\pi/6 & \text{if } D_{AB} \geq S_{AB} \text{ and } |\psi - \psi_{park}| \geq k_1 \\
0 & \text{if } D_{AB} \geq S_{AB} \text{ and } |\psi - \psi_{park}| < k_1 
\end{cases}
\]

\(\psi_{park} \text{ --- Longitudinal direction of parking spot (rad);} k_1 \text{ --- Calibration parameter of direction error threshold between vehicle and parking spot (rad).} \\

**Lookahead Distance**

Proper lookahead distance \(L_d\) is important for vehicle motion control performance. The \(\psi_e \text{ and } L_d\) is directly related to the determined target position. Vehicle will move with a smaller curvature which means more time or longer distance to reach target path when increase \(L_d\).
move with a larger curvature which means less time or shorter distance to reach target path when decrease $L_d^{[9]}$. Therefore, it is critical to determine a suitable target position for every moment.

Thus, cost function of lookahead distance is established considering distance error between target and actual path and its max value.

Cost function of straight driving:

$$J_1 = k_2 \frac{\sum_{i=1}^{n}|\xi_i|}{n*C_1} + k_3 \frac{\max(|\xi_i|)}{C_2} + k_4 \frac{\sum_{i=1}^{n}|\dot{\psi}_i|}{n*C_3} + k_5 \frac{\max(|\dot{\psi}_i|)}{C_4}$$  \hspace{1cm} (11)

Cost function of turning:

$$J_2 = k_6 \frac{\sum_{i=1}^{n}|\xi_i|}{n*C_5} + k_7 \frac{\max(|\xi_i|)}{C_6} + k_8 \frac{\sum_{i=1}^{n}|\dot{\psi}_i|}{n*C_7} + k_9 \frac{\max(|\dot{\psi}_i|)}{C_8}$$  \hspace{1cm} (12)

Cost function of normal driving:

$$J = 0.5 * J_1 + 0.5 * J_2$$  \hspace{1cm} (13)

$J$, Cost function of $L_d$; $\xi_i$, Distance error between target and actual path (m), $i$, is from 1 to $n$; $\psi_i$, Orientation change rate of vehicle (rad·s$^{-1}$), $i$, is from 1 to $n$; $n$, Number of data base; $k_2$, Weight coefficient of distance error for straight driving; $k_3$, Weight coefficient of max distance error for straight driving; $k_4$, Weight coefficient of orientation change rate for straight driving; $k_5$, Weight coefficient of max orientation change rate for straight driving; $k_6$, Weight coefficient of distance error for turning; $k_7$, Weight coefficient of max distance error for turning; $k_8$, Weight coefficient of orientation change rate for turning; $k_9$, Weight coefficient of max orientation change rate for turning; $C_1$, Acceptable mean lateral error of straight driving (m); $C_2$, Acceptable max lateral error of straight driving (m); $C_3$, Acceptable mean orientation change rate of straight driving (rad·s$^{-1}$); $C_4$, Acceptable max orientation change rate of straight driving (rad·s$^{-1}$); $C_5$, Acceptable mean lateral error of turning (m); $C_6$, Acceptable max lateral error of turning (m); $C_7$, Acceptable mean orientation change rate of turning (rad·s$^{-1}$); $C_8$, Acceptable max orientation change rate of turning (rad·s$^{-1}$).

Both lateral error and orientation change rate (OCR) are considered to evaluate vehicle motion control performance. Lateral error is always an important parameter while OCR is mostly considered during straight line for better driving experience. Thus, each weight coefficient of cost function is listed in Table 1.

| Parameter | Value | Parameters | Value  |
|-----------|-------|------------|-------|
| $k_2$     | 0.3   | $C_1$      | 0.1/m |
| $k_3$     | 0.2   | $C_2$      | 0.3/m |
| $k_4$     | 0.2   | $C_3$      | 0.1/rad·s$^{-1}$ |
| $k_5$     | 0.3   | $C_4$      | 0.2/rad·s$^{-1}$ |
| $k_6$     | 0.4   | $C_5$      | 0.2/m |
| $k_7$     | 0.3   | $C_6$      | 0.5/m |
| $k_8$     | 0.2   | $C_7$      | 0.5/rad·s$^{-1}$ |
| $k_9$     | 0.1   | $C_8$      | 2/rad·s$^{-1}$ |

Table 2. Result of cost function.

| $L_d$ (m) | $J$   |
|-----------|-------|
| 1         | 0.5956|
| 2         | 0.4343|
| 3         | 0.5724|
| 4         | 0.9136|
| 5         | 1.1455|
| 6         | 2.0836|
Different cost function results are listed in Table 2. The result is minimum when lookahead distance equals 2 m, which means best performance. This method can be applied to achieve optimal lookahead distance at different condition.

**System Functions Performance**

**Shuttle Function Performance**

Fig. 12 shows the comparison between target path and actual path in (a) around AVL Shanghai Technical Center. The error distribution between target path and actual path is displayed in Fig. 12 (b) and Fig. 12 (c) resp. for curve and straight lines.

![Figure 12](image)

Figure 12. Path actual vs Target (a), Lateral control performance during phase turning (b) and straight line (c).

This controller error (Approximately +/-20cm) is considered as acceptable within the vicinity of the AVL Shanghai Technical Center since sufficient margin exists on each side of the defined target path.

**Park Function Performance**

Fig. 13 displays the parking accuracy considering lateral and longitudinal error between the center of car and parking spot. Max error between vehicle and parking spot is within 0.2m which is an acceptable result considering side margin of the parking spots.

![Figure 13](image)

Figure 13. Parking accuracy.

**Conclusion**

This paper provides a comprehensive technical description of the system and an overview of key results of this AVP project and its limitations. This specific approach realizes the target functions by using a minimum set of sensors and minimum intrusion in the already existing system of the vehicle while guaranteeing sufficient accuracy and safety level. It proves to be an affordable solution for control of vehicles in a geofenced and opened environment. Adaptation of existing deep learning algorithms allows to detect Parking spots with a high level of confidence. Radar and camera ensure safety of potential passenger as well as surrounding pedestrians or vehicle by detecting in a robust way the obstacle on the path of the vehicle. Next step of development includes obstacle avoidance, detection of parking spots in closed environments by vehicle, additional sensor fusion between infrastructure and vehicle by means of 5G technology being rolled out in Shanghai.
Reference

[1] https://autopilot-project.eu/autopilot-partner-vtt-achieves-fully-autonomous-parking-of-robot-car-marilyn/

[2] https://www.bosch-mobility-solutions.com/media/global/highlights/automated-mobility/automated-valet-parking/avp_infografik-2.pdf

[3] Faheem, S.A. Mahmud, G.M. Khan, M. Rahman and H. Zafar, A Survey of Intelligent Car Parking System[J]. Journal of Applied Research and Technology, 2013, 11(5):714-726.

[4] Kim G, Mun H, Kim B. Performance of AEB System on a Slope Using an Extended Kalman Filter[J]. International Journal of Software Engineering and Knowledge Engineering, 2019, 29(7):955-969.

[5] Bibi N, Majid M N, Dawood H, et al. Automatic Parking Space Detection System[C]/2nd International Conference on Multimedia and Image Processing. IEEE, 2017.

[6] Redmon J, Farhadi A. YOLOv3: An Incremental Improvement[J]. 2018.

[7] Samuel M, Hussein M, Binti M. A Review of some Pure-Pursuit based Path Tracking Techniques for Control of Autonomous Vehicle[J]. International Journal of Computer Applications, 2016, 135(1):35-38.

[8] J. Giesbrecht, D. Mackay, J. Collier, S. Verret. Path Tracking for Unmanned Ground Vehicle Navigation: Implementation and Adaptation of the Pure Pursuit Algorithm[J]. Drdc Suffield Tm, 2005.

[9] Chen Z, Nie L, Guan J. Research on Lateral Tracking Control Method of Autonomous Vehicle[J]. Digital Manufacture Science, 2019, 17(1):17-20.