2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)
October 14-18, 2024. Abu Dhabi, UAE

Real-Time Path Generation and Alignment Control for Autonomous Curb Following

Yuanzhe Wang, Yunxiang Dai and Danwei Wang *Fellow, IEEE*

***Abstract*— Curb following is a key technology for autonomous road sweeping vehicles. Currently, existing implementations primarily involve pre-recording waypoints during human driving and subsequently retracing them autonomously. Moreover, existing research related to this topic predominately focuses on curb detection for driver assistance, yet the resultant curb detection outcomes remain underutilized in the development of autonomous curb following systems. To fill this gap, this paper proposes a real-time path generation and alignment control approach to facilitate autonomous curb following. Firstly, a segmented path generation algorithm is introduced that progressively generates reference path segments while ensuring the overall continuity of the reference path. Secondly, a parameterized alignment control algorithm is developed to accurately navigate the vehicle along the planned reference path with proved stability. Real public road experiments have been conducted to validate the proposed approach. The experimental results demonstrate the efficacy of the proposed methodologies across various curb following scenarios, including common concave, convex, and straight-concave curbs, thereby showcasing the practical viability of our methods in real-world applications.**

I. Introduction

With the rapid development of self-driving technologies in the past decade, autonomous vehicles have found successful applications in various domains, including warehouse management, port operation, and environmental service. Among them, road sweeping, a common environmental service, involves deploying sweeping vehicles in designated areas for cleaning purposes. A fundamental aspect of road sweeping is to precisely trace curbs on public roads to facilitate effective cleaning. With the support from the National Robotics Programme of Singapore, our group has also developed an autonomous environmental service vehicle with the capability of road sweeping [1]. Current autonomous sweeping technologies generally require a human driver guiding the vehicle along designated road curbs beforehand, recording waypoints for subsequent autonomous sweeping. During these operations, the recorded waypoints serve as the reference for the vehicle, ensuring reliable and repeatable operation. However, this approach has limitations in environments lacking prior manual preparations. To enhance system flexibility and intelligence, a fully autonomous curb following solution with seamlessly integrated perception and decision-making capabilities is necessary.

Currently, most existing research efforts related to curb following mainly center around curb detection [2]–[27]. Despite extensive investigation, the detected results primarily serve road boundary identification for driver assistance, with limited exploration of their potential to facilitate autonomous vehicle operations. While a few studies in the literature leverage curb detection outcomes for vehicle motion planning and control [28]–[34] across various applications, the problems they address differ significantly from curb following. Consequently, their methodologies cannot be directly applied to realize autonomous curb following. Moreover, in practical applications, curb detection encounters various uncertainties, including sensor noise and roadside vegetation, resulting in numerous outliers among the detected points. Directly following such points may yield nonsmooth vehicle movements, a scenario deemed unacceptable in practice. At present, fully autonomous curb following with perception and decision-making capabilities tightly integrated still remains a challenge.

Motivated by the discussions presented above, this paper delves into the challenging domain of autonomous curb following and proposes a real-time path generation and alignment control approach tailored to this problem. The proposed approach comprehensively accounts for the interactions between perception and decision-making. Firstly, a segmented path generation algorithm is introduced to progressively establish a globally continuous reference path for the vehicle. Secondly, a parameterized alignment control algorithm is developed to guide the vehicle along the reference path, with proved stability. The efficacy of the proposed methodology is substantiated by a series of public road experiments. The primary contribution of this paper lies in the development of a real-time path generation and alignment control approach explicitly designed for autonomous curb following, compatible with existing curb detection algorithms. Specifically, the contribution encompasses:

- The introduction of a segmented path generation algorithm, which provides a piecewise yet globally continuous reference path for the vehicle.
- The development of a parameterized alignment control algorithm tailored for nonholonomic vehicles, facilitating precise curb following.

The remainder of this paper is organized as follows.

The first two authors contribute equally.

This research is supported by the Agency for Science, Technology and Research (A*STAR) under its National Robotics Programme (Project No. W2122d0245 and M22NBK0109)

Y. Wang is with the School of Control Science and Engineering, Shandong University, Jinan, China, 250061, and was with the School of Electrical and Electronic Engineering, Nanyang Technological University, Singapore, 639798.

Y. Dai and D. Wang are with the School of Electrical and Electronic Engineering, Nanyang Technological University, Singapore, 639798.

Corresponding Author: Yuanzhe Wang, wang0951@e.ntu.edu.sg

Sec. II presents existing studies related to this paper. Sec. III formulates the curb following problem. Sec. IV presents the proposed approach. Experimental results are discussed in Sec. V, while Sec. VI concludes this paper.

II. Related Work

This section provides a brief survey of existing research efforts related to autonomous curb following. Within the literature, the majority of related studies predominantly concentrate on curb detection. Consequently, this section begins with a discussion of curb detection. Subsequently, we delve into alignment control studies, which exhibit similarities to curb following.

A. Curb Detection

Curb detection is a well-established research area with a significant body of work in the literature. In particular, a recent survey paper comprehensively summarizes and discusses existing curb detection approaches [35]. These approaches vary with the sensor modalities employed, mainly including *vision based* [2]–[6], *point cloud based* [7]–[20], and *multi-modal* [21]–[25]. Traditional curb detection primarily relies on image processing, where researchers have developed vision based techniques to extract curb information from images. Given that the most distinguishing feature between a curb from a road is the elevation difference, point cloud based methods dominate in existing related work. Some researchers use stereo cameras to generate 3D point clouds [7]–[11], while others utilize LiDAR [12]–[20]. Considering the fact that both vision based and point cloud based approaches have certain limitations, multi-modal methods are proposed by fusing sensing information from heterogeneous sensors, particularly camera and LiDAR [21]–[25]. Besides these three main categories of curb detection approaches, there are a few exceptions which employ ultrasonic sensors [36], leverage Google Street View images [26], or utilize offline high-resolution aerial images [27]. Despite the extensive research conducted in recent years, the detected results have predominantly served road boundary identification in driver-assistance systems, with limited exploration of their potential for enabling autonomous vehicle operations.

B. Alignment Control

In the literature, we can find a few papers which employ curb detection results to plan and control vehicle motions [28]–[30]. In [28], curb detection outcomes are utilized to identify potential collisions and initiate a transition between teleoperation and autonomous navigation modes. Reference [29] leverages curb detection results to prevent the robot from colliding with curbs by correcting its motion. Experimental findings demonstrate the effectiveness of the proposed controller along straight road curbs but reveals limitations in handling curved ones. Reference [30] develops a curb recognition and negotiation system for robotic wheelchairs, enabling the recognition of curb characteristics and autonomous curb negotiation. While these studies address applications involving curb detection, it is obvious that they are different from the specific problem of curb following. An existing research problem that may relate to curb following is wall following, a well-established area in mobile robotics with quite a few studies in the literature [31]–[34]. In these studies, distance sensors such as sonar are generally employed, and raw sensing data is directly incorporated into control algorithms. However, it is worth highlighting that curb following presents distinct challenges compared to wall following. The distance from a vehicle to a wall is typically straightforward to measure, ensuring continuous and reliable observations. In contrast, curb detection involves substantial uncertainties, posing great challenges to direct sensory feedback control. Therefore, wall following approaches cannot be directly applied in curb following scenarios.

III. Problem Formulation

This paper addresses the challenge of autonomous curb following, as shown in Fig. 1. The objective of the vehicle is to detect and follow a road curb while maintaining a specified distance from it. This entails a threefold process: curb detection, path generation, and alignment control. Initially, the vehicle must detect the curb, followed by the planning of a reference path at a predetermined distance from the curb. Subsequently, the vehicle must achieve precise alignment with this reference path. Given the extensive prior research on curb detection, this paper primarily concentrates on path generation and alignment control.

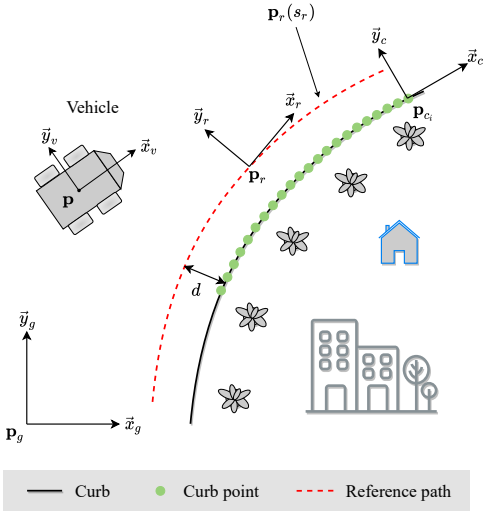


Fig. 1. The diagram of autonomous curb following problem

To facilitate problem formulation and algorithm design, we define the following three coordinate frames and a convention:

- *Global frame* is a stationary coordinate frame attached on the ground, defined as $\{g\} = \{\vec{x}_g,\ \vec{y}_g\}$.
- *Vehicle frame* is a mobile coordinate frame attached on and moving with the vehicle, denoted as $\{v\} = \{\vec{x}_v,\ \vec{y}_v\}$ with $\vec{x}_v$ pointing towards the moving direction.

- *Frenet frame* is a mobile coordinate frame that follows a curved path. In this paper, we refer to the Frenet frame on the curb curve as $\{c\} = \{\vec{x}_c,\ \vec{y}_c\}$ and that on the reference path as $\{r\} = \{\vec{x}_r,\ \vec{y}_r\}$.

The frames $\{g\}$, $\{v\}$, $\{c\}$ and $\{r\}$ are visualized in Fig. 1. This paper uses the pre-superscript to describe the coordinate frame where a variable is expressed.

A. Vehicle Kinematics

This paper considers the following first-order nonholonomic model of vehicle kinematics:

$$\begin{aligned} \dot{x} &= v\cos\theta \\ \dot{y} &= v\sin\theta \\ \dot{\theta} &= \omega \end{aligned} \tag{1}$$

where $\boldsymbol{p} = [x\ y]^\top$ and θ represent the position and the orientation of the vehicle, respectively, $\boldsymbol{u} = [v\ \omega]^\top$ is the control input of the vehicle, v and ω are the velocity and the angular velocity of the vehicle, respectively.

B. Path Generation

In practical scenarios, the contour of a curb can be approximated by a curved path, denoted as $\boldsymbol{p}_c(s_c)$ with s_c as the parameter. To locate a curb, it is necessary to identify a series of discrete points along it, called curb points. As depicted in Fig. 1, $\boldsymbol{p}_{c_i} = [x_{c_i}\ y_{c_i}]^\top, i = 1, 2, \cdots$ is used to denote the position of curb point i. Curb detection algorithms estimate $\boldsymbol{p}_{c_i}$ and get $\hat{\boldsymbol{p}}_{c_i} = [\hat{x}_{c_i}\ \hat{y}_{c_i}]^\top$.

The curb following task requires the vehicle to move along the curb and maintain a constant distance d from it. Therefore, we need to find the mathematical representation of the curb curve $\hat{\boldsymbol{p}}_c(s_c) = [\hat{x}_c(s_c)\ \hat{y}_c(s_c)]^\top$ based on the curb detection results $\hat{\boldsymbol{p}}_{c_i}, i = 1, 2, \cdots$, and then generate a reference path $\boldsymbol{p}_r(s_r) = [x_r(s_r)\ y_r(s_r)]^\top$ for the vehicle, where s_r is the parameter of the reference path. The reference path should satisfy:

$$d_{rc} = d \tag{2}$$

where $d_{rc} = \min \left\| \boldsymbol{p}_r\left(s_r\right) - \hat{\boldsymbol{p}}_c\left(s_c\right) \right\|,\ \forall s_r, s_c$.

C. Alignment Control

To align with the generated reference path, a virtual target vehicle is introduced which moves along the reference path. The Frenet frame $\{r\}$ is attached on the virtual target vehicle. Thus, the reference path alignment problem is transformed to the virtual target tracking problem. The alignment error is formulated as:

$$^r\boldsymbol{p} = R_*^r\left(^*\boldsymbol{p} - {}^*\boldsymbol{p}_r\right),\ ^r\theta = {}^*\theta - {}^*\theta_r \tag{3}$$

where $R_*^r = \begin{bmatrix} \cos{}^*\theta_r & \sin{}^*\theta_r \\ -\sin{}^*\theta_r & \cos{}^*\theta_r \end{bmatrix}$ is the rotation matrix from $\{*\}$ to $\{r\}$. $\{*\}$ represents the reference frame for control design, which can be $\{g\}$ or any other frame. $^*\boldsymbol{p}_r$ is the position of the virtual target vehicle relative to $\{*\}$. $^*\theta_r$ is the orientation of $\vec{x}_r$ expressed in $\{*\}$, i.e. the moving direction of the virtual target vehicle. The objective of alignment control is to design a control algorithm to determine the movements of the virtual target vehicle and the vehicle such that the alignment error $[^r\boldsymbol{p}^\top\ {}^r\theta]^\top$ converges to zero asymptotically.

D. Problem Statement

Based on the formulations made above, the curb following problem is stated as follows.

Problem. *Consider an autonomous vehicle modeled by Eq.* (1)*, tasked with navigating along a curb while maintaining a prescribed distance d from it. Assume that online curb detection is available, enabling the localization of curb points as $\hat{\boldsymbol{p}}_{c_i}$. This problem is composed of the following two key parts:*

- *Path Generation. Develop an algorithm to generate a reference path $\boldsymbol{p}_r(s_r)$ for the vehicle, satisfying the constraint Eq.* (2)*.*
- *Alignment Control. Develop an algorithm to guide the vehicle along the reference path $\boldsymbol{p}_r(s_r)$ with the alignment error Eq.* (3) *converging to zero asymptotically.*

IV. Proposed Approach

To address the curb following problem formulated in Sec. III, this paper proposes an integrated framework outlined in Fig. 2. The proposed framework consists of curb detection, path generation, and alignment control, and is compatible with any existing curb detection algorithm. In this paper, we primarily focus on introducing our developed path generation and alignment control approach.

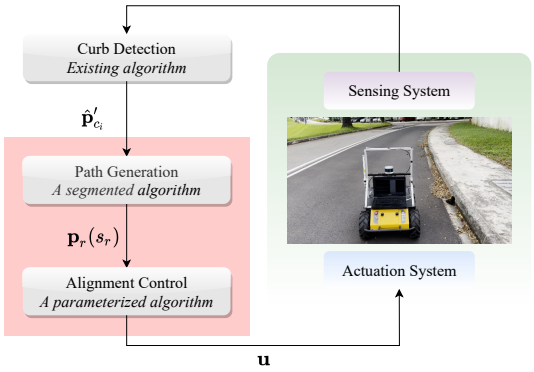


Fig. 2. The schematic diagram of the proposed curb following framework

A. A Segmented Path Generation Algorithm

This subsection introduces the proposed path generation algorithm, as depicted in Fig. 3 and Fig. 4. The algorithm comprises three key modules: sliding window processing, key point extraction, and path smoothing. Due to sensing range limitations in practical implementations, only curb points within a specific range from the vehicle can be detected. Thus, global path generation is impossible. To overcome this challenge, we generate the reference path in a piecewise and progressive manner. Specifically, the vehicle plans a local path segment based on the current curb point detection and extends the reference path with a new segment

when it reaches the end of the current one. Each reference path segment is associated with a path frame $\{p_k\}$. For example, in Fig. 3, $\{p_{k-1}\} = \{\vec{x}_{p_{k-1}}, \vec{y}_{p_{k-1}}\}$ and $\{p_k\} = \{\vec{x}_{p_k}, \vec{y}_{p_k}\}$ are the reference frames for the $k-1$th and kth path segments, respectively. The origin of $\{p_k\}$ locates at the position of the vehicle when it plans the kth segment, and $\vec{x}_{p_k}$ aligns with the vehicle's heading direction at that moment. In this paper, path generation for the kth segment is initiated when the vehicle completes the $k-1$th segment. Therefore, the generated reference path comprises a sequence of segments, denoted as $\{{}^{p_k}\boldsymbol{p}_{r_k}(s_{r_k})\}, k = 1, 2, \cdots$ with each path segment ${}^{p_k}\boldsymbol{p}_{r_k}(s_{r_k})$ expressed in $\{p_k\}$.

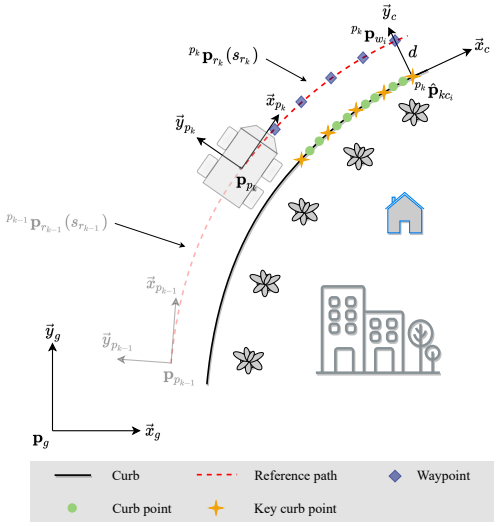


Fig. 3. The illustrative diagram of the proposed path generation algorithm

As mentioned earlier, the proposed framework is compatible with existing curb detection algorithms. In this paper, we assume the vehicle is equipped with a 3D LiDAR and employs an open-source point cloud based curb detection algorithm [18], which operates at the frame level. However, single-frame detection may contain outliers due to the sparsity of point clouds. Furthermore, it overlooks road curb continuity, which can be used to filter outliers by assessing curb points over a certain time period. To address this challenge, we introduce a temporal refinement mechanism, which analyzes the detection results in a most recent time window of size N_f. At each time step, the latest detection results are pushed into the window, while the earliest ones are popped out. Within the window, DBSCAN clustering is applied on the detected curb points, followed by a RANSAC filter imposed on the largest cluster. The final inliers are designated as filtered curb points $\hat{\boldsymbol{p}}_{c_i}, i = 1, 2, \cdots$, while the outliers are removed from the sliding window.

Next, we use the kth path segment generation as an example to illustrate the proposed algorithm. When the vehicle reaches the end of the $k-1$th segment, the filtered curb points $\hat{\boldsymbol{p}}_{c_i}, i = 1, 2, \cdots, N_{ck}$ are transformed to $\{p_k\}$ using ${}^{p_k}\hat{\boldsymbol{p}}_{c_i} = R_g^{p_k}(\hat{\boldsymbol{p}}_{c_i} - \boldsymbol{p}_{p_k})$, where $i = 1, 2, \cdots, N_{ck}$, N_{ck} is the number of filtered curb points for the kth path segment, $R_g^{p_k}$ is the rotation matrix from $\{g\}$ to $\{p_k\}$, $\boldsymbol{p}_{p_k}$

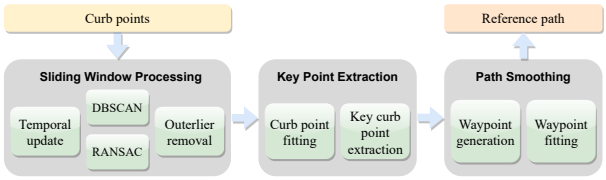


Fig. 4. The flow chart of the proposed path generation algorithm

is the origin position of $\{p_k\}$, and θ_{p_k} is the orientation of $\vec{x}_{p_k}$.

Based on the filtered curb points ${}^{p_k}\hat{\boldsymbol{p}}_{c_i}, i = 1, 2, \cdots, N_{ck}$, polynomial regression is performed to estimate the curb curve ${}^{p_k}\hat{\boldsymbol{p}}_c(s_c)$, ${}^{p_k}\hat{y}_c = \sum_{j=0}^{n} a_j {}^{p_k}\hat{x}_c^j$. Specifically, the following optimization problem is solved:

$$\underset{\boldsymbol{a}}{\text{minimize}} \quad \left\| \hat{\boldsymbol{y}}_c - \hat{X}_c \boldsymbol{a} \right\| \tag{4}$$

where $\hat{\boldsymbol{y}}_c = \textbf{col}[{}^{p_k}\hat{y}_{c_i} \,|_{i=1}^{N_{ck}}]$, $\boldsymbol{a} = \textbf{col}[a_j \,|_{j=0}^{n}]$, $\hat{X}_c$ is a $N_{ck} \times (n+1)$ matrix with ${}^{p_k}\hat{x}_{c_i}^j$ as the element in the ith row and the $j+1$th column, $\textbf{col}\,[x_i|_{i=1}^{n}]$ denotes the vector stacked by $x_i|_{i=1}^{n}$ with its ith row element being x_i. The solution of this optimization problem is $\boldsymbol{a} = \left(\hat{X}_c^T \hat{X}_c\right)^{-} \hat{X}_c^T \hat{\boldsymbol{y}}_c$. Thus, the curb curve ${}^{p_k}\hat{\boldsymbol{p}}_c(s_c)$ is estimated with ${}^{p_k}\hat{x}_c$ as the parameter, i.e. $s_c = {}^{p_k}\hat{x}_c$.

Then, we select key curb points by sampling the curb curve along ${}^{p_k}\hat{x}_c$ with a designated interval $I_{\hat{x}_c}$, resulting in ${}^{p_k}\hat{\boldsymbol{p}}_{kc_i} = [{}^{p_k}\hat{x}_{kc_i} \; {}^{p_k}\hat{y}_{kc_i}]^\top, i = 1, 2, \cdots, N_{kck}$, where N_{kck} is the number of key curb points after sampling. Waypoints are then generated by translating these key curb points along $\vec{y}_c$ of the Frenet frame $\{c\}$ which moves along the estimated curb curve. Mathematically, waypoints ${}^{p_k}\boldsymbol{p}_{w_i} = [{}^{p_k}x_{w_i} \; {}^{p_k}y_{w_i}]^\top, i = 1, 2, \cdots, N_{kck}$ can be obtained using ${}^{p_k}\boldsymbol{p}_{w_i} = R_c^{p_k} {}^{c}\boldsymbol{p}_{wl} + {}^{p_k}\hat{\boldsymbol{p}}_{kc_i}$, where ${}^{c}\boldsymbol{p}_{wl} = [0 \; d]^\top$ if the vehicle moves on the left side of the curb, and ${}^{c}\boldsymbol{p}_{wl} = [0 \; -d]^\top$ otherwise.

Finally, polynomial regression is performed on these waypoints to generate the kth reference path segment while ensuring continuity with the $k-1$th segment. This involves solving the following optimization problem:

$$\begin{aligned} \underset{\boldsymbol{b}}{\text{minimize}} \quad & \|\boldsymbol{y}_w - X_w \boldsymbol{b}\| \\ \text{subject to} \quad & {}^{p_k}y_{re} = {}^{p_k}\boldsymbol{x}_{re}^\top \boldsymbol{b} \end{aligned} \tag{5}$$

where $\boldsymbol{y}_w = \textbf{col}[{}^{p_k}y_{w_i} \,|_{i=1}^{N_{kck}}]$, $\boldsymbol{b} = \textbf{col}[b_j \,|_{j=0}^{n}]$, X_w is a $N_{kck} \times (n+1)$ matrix with ${}^{p_k}x_{w_i}^j$ as the element in the ith row and the $j+1$th column, ${}^{p_k}\boldsymbol{x}_{re} = \textbf{col}[{}^{p_k}x_{re}^j \,|_{j=0}^{n}]$, ${}^{p_k}\boldsymbol{p}_{re} = [{}^{p_k}x_{re} \; {}^{p_k}y_{re}]^\top$ is the endpoint position of the $k-1$th reference path segment expressed in $\{p_k\}$. This optimization problem can be simplified to the following unconstrained optimization problem:

$$\underset{\boldsymbol{b'}}{\text{minimize}} \quad \|\boldsymbol{y}_w - X'_w \boldsymbol{b'}\| \tag{6}$$

where $\boldsymbol{b'} = \textbf{col}[b_j \,|_{j=1}^{n}]$, X'_w is a $N_{kck} \times n$ matrix with ${}^{p_k}x_{w_i}^j - {}^{p_k}x_{re}^j$ as the element in the ith row and the jth column. The solution of this optimization problem is $\boldsymbol{b'} = \left(X_w'^T X'_w\right)^{-} X_w'^T \boldsymbol{y}_w$ with $b_0 = {}^{p_k}y_{re} - \sum_{j=1}^{n} b_j {}^{p_k}x_{re}^j$.

Thus, the kth reference path segment ${}^{p_k}\boldsymbol{p}_{r_k}(s_{r_k})$ is obtained, parameterized by ${}^{p_k}x_{r_k}$, i.e. $s_{r_k} = {}^{p_k}x_{r_k}$, and expressed as ${}^{p_k}y_{r_k} = \sum_{j=0}^{n} b_j {}^{p_k}x_{r_k}^j$.

B. A Parameterized Alignment Control Algorithm

This subsection introduces the proposed path alignment control algorithm. As the reference path is planned piecewisely and each segment is expressed in its own path frame, the path alignment control algorithm is discussed with respect to the corresponding path frame as well. In this subsection, we illustrate the proposed control algorithm using the kth path segment tracking as an example. Thus, $\{*\}$ in (3) becomes $\{p_k\}$, and the vehicle pose is transformed to $\{p_k\}$ by ${}^{p_k}\boldsymbol{p} = R_g^{p_k}(\boldsymbol{p} - \boldsymbol{p}_{p_k})$ and ${}^{p_k}\theta = \theta - \theta_{p_k}$, where $R_g^{p_k}$ represents the rotation matrix from $\{g\}$ to $\{p_k\}$.

Since the generated reference path is parameterized by ${}^{p_k}x_r$, the alignment control command should be driven by ${}^{p_k}x_r$ as well. The dynamics of path alignment error parameterized by ${}^{p_k}x_r$ can be derived from (3) as follows:

$$\begin{aligned} {}^{r}\dot{\boldsymbol{p}} &= \dot{R}_{p_k}^{r}\left({}^{p_k}\boldsymbol{p} - {}^{p_k}\boldsymbol{p}_r\right) + R_{p_k}^{r}\left({}^{p_k}\dot{\boldsymbol{p}} - {}^{p_k}\dot{\boldsymbol{p}}_r\right) \\ {}^{r}\dot{\theta} &= \omega - {}^{p_k}\dot{\theta}_r \end{aligned} \tag{7}$$

Since $\dot{R}_{p_k}^{r} = S\left({}^{p_k}\dot{\theta}_r\right) R_{p_k}^{r} = R_{p_k}^{r} S\left({}^{p_k}\dot{\theta}_r\right)$ with $S(a) = \begin{bmatrix} 0 & a \\ -a & 0 \end{bmatrix}$ and (3), ${}^{r}\dot{\boldsymbol{p}}$ can be rewritten as:

$${}^{r}\dot{\boldsymbol{p}} = S\left({}^{p_k}\dot{\theta}_r\right){}^{r}\boldsymbol{p} + R_{p_k}^{r}\,{}^{p_k}\dot{\boldsymbol{p}} - R_{p_k}^{r}\,{}^{p_k}\dot{\boldsymbol{p}}_r \tag{8}$$

where $S\left({}^{p_k}\dot{\theta}_r\right){}^{r}\boldsymbol{p}$ and $R_{p_k}^{r}\,{}^{p_k}\dot{\boldsymbol{p}}$ are easy to compute. Now, we focus on $R_{p_k}^{r}\,{}^{p_k}\dot{\boldsymbol{p}}_r$, which can be rewritten as $R_{p_k}^{r}\,{}^{p_k}\dot{\boldsymbol{p}}_r = R_{p_k}^{r}\,{}^{p_k}\boldsymbol{p}_t {}^{p_k}\dot{x}_r$, where ${}^{p_k}\boldsymbol{p}_t = [1\ \mathrm{d}^{p_k}y_r/{}^{p_k}\mathrm{d}x_r]^\top$. Considering that the vector ${}^{p_k}\dot{\boldsymbol{p}}_r$ aligns with the direction of $\vec{x}_r$, and given that the length of a vector remains unchanged after coordinate transformation, it can be obtained that $R_{p_k}^{r}\,{}^{p_k}\dot{\boldsymbol{p}}_r = \begin{bmatrix} \sqrt{1+\left(\frac{\mathrm{d}^{p_k}y_r}{\mathrm{d}^{p_k}x_r}\right)^2} \\ 0 \end{bmatrix} {}^{p_k}\dot{x}_r$.

To express ${}^{p_k}\dot{\theta}_r$ in terms of ${}^{p_k}x_r$, we write the curvature of the reference path as $\kappa_r = \frac{\mathrm{d}^{p_k}\theta_r}{\mathrm{d}l_r}$, where l_r is the arc length parameter and $\mathrm{d}l_r = \sqrt{\left(\mathrm{d}^{p_k}x_r\right)^2 + \left(\mathrm{d}^{p_k}y_r\right)^2} = \sqrt{1+\left(\frac{\mathrm{d}^{p_k}y_r}{\mathrm{d}^{p_k}x_r}\right)^2}\mathrm{d}^{p_k}x_r$. This leads to $\mathrm{d}^{p_k}\theta_r = \kappa_r\sqrt{1+\left(\frac{\mathrm{d}^{p_k}y_r}{\mathrm{d}^{p_k}x_r}\right)^2}\mathrm{d}^{p_k}x_r$ and subsequently ${}^{p_k}\dot{\theta}_r = \kappa_r\sqrt{1+\left(\frac{\mathrm{d}^{p_k}y_r}{\mathrm{d}^{p_k}x_r}\right)^2}{}^{p_k}\dot{x}_r$.

Based on the above derivations, the dynamics of path alignment error can be simplified as:

$$\begin{aligned} {}^{r}\dot{x} &= {}^{p_k}\dot{x}_r\sqrt{1+\left(\frac{\mathrm{d}^{p_k}y_r}{\mathrm{d}^{p_k}x_r}\right)^2}\left(\kappa_r\,{}^{r}y - 1\right) + v\cos{}^{r}\theta \\ {}^{r}\dot{y} &= -{}^{p_k}\dot{x}_r\sqrt{1+\left(\frac{\mathrm{d}^{p_k}y_r}{\mathrm{d}^{p_k}x_r}\right)^2}\kappa_r\,{}^{r}x + v\sin{}^{r}\theta \\ {}^{r}\dot{\theta} &= \omega - {}^{p_k}\dot{x}_r\sqrt{1+\left(\frac{\mathrm{d}^{p_k}y_r}{\mathrm{d}^{p_k}x_r}\right)^2}\kappa_r \end{aligned} \tag{9}$$

To achieve the alignment control objective, the following control algorithm is proposed:

$$\begin{aligned} v =& v_f \\ \omega =& {}^{p_k}\dot{x}_r\sqrt{1+\left(\frac{\mathrm{d}^{p_k}y_r}{\mathrm{d}^{p_k}x_r}\right)^2}\kappa_r + {}^{r}\dot{\theta}_d \\ & - \gamma_\theta v\,{}^{r}y\frac{\sin{}^{r}\theta - \sin{}^{r}\theta_d}{{}^{r}\theta - {}^{r}\theta_d} - k_\theta\left({}^{r}\theta - {}^{r}\theta_d\right) \end{aligned} \tag{10}$$

where v_f is a positive constant because curb following task generally requires the vehicle to move forward with a constant speed, $\gamma_\theta > 0$ and $k_\theta > 0$ are design parameters. The progression rate of the parameter ${}^{p_k}x_r$ is designed as ${}^{p_k}\dot{x}_r = \frac{v\cos{}^{r}\theta + k_x\,{}^{r}x}{\sqrt{1+\left(\frac{\mathrm{d}^{p_k}y_r}{\mathrm{d}^{p_k}x_r}\right)^2}}$, where $k_x > 0$ is a design parameter. It controls the movement of the virtual target vehicle along the reference path. The desired approaching angle is designed as ${}^{r}\theta_d = -\bar{\theta}_d\tanh\gamma_d\,{}^{r}y$, where $\bar{\theta}_d > 0$ and $\gamma_d > 0$ are design parameters. It shapes the maneuver of how the vehicle converges to the reference path. And it is not difficult to obtain that ${}^{r}\dot{\theta}_d = -\bar{\theta}_d\left[1-\tanh^2\left(\gamma_d\,{}^{r}y\right)\right]\gamma_d\,{}^{r}\dot{y}$. Since the planned reference path is a polynomial and parameterized by ${}^{p_k}x_r$, the curvature κ_r can be computed as $\kappa_r = \frac{\mathrm{d}^2{}^{p_k}y_r/\mathrm{d}^{p_k}x_r^2}{\left(1+(\mathrm{d}^{p_k}y_r/\mathrm{d}^{p_k}x_r)^2\right)^{\frac{3}{2}}}$.

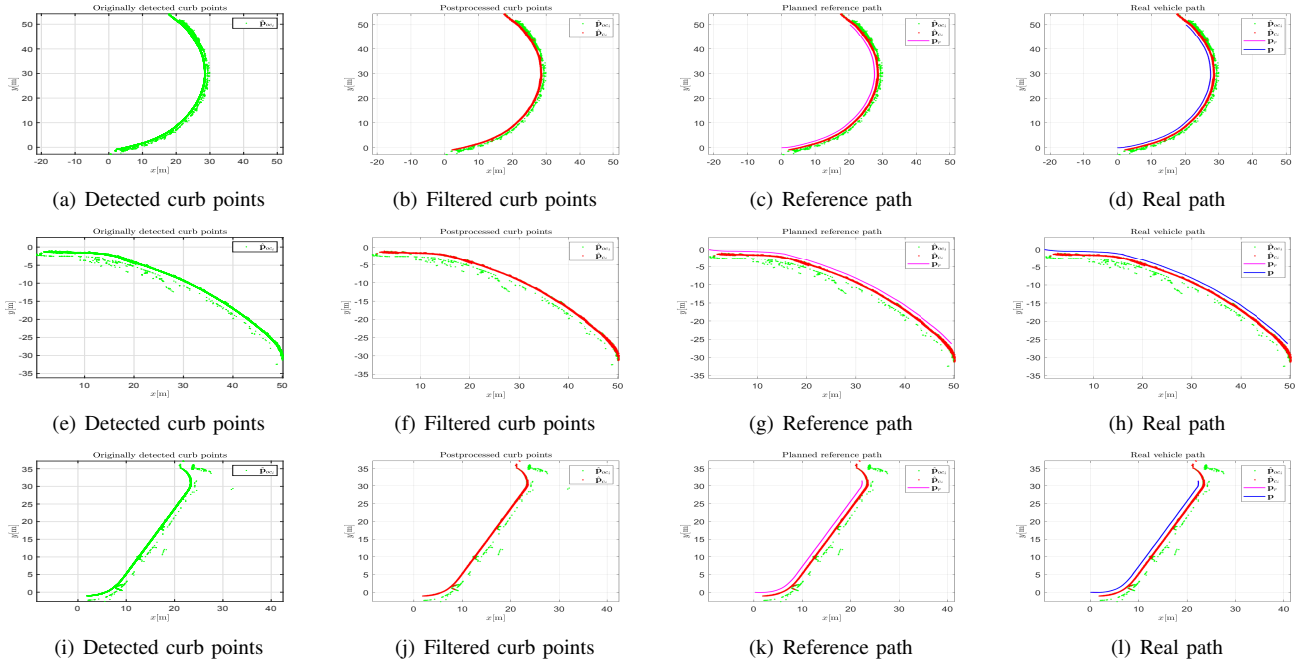
The performance of the proposed alignment control algorithm is analyzed in the following theorem.

Theorem 1. *Consider a nonholonomic vehicle modeled by (1) and a reference path. When the control algorithm (10) is applied, the path alignment error (3) converges to zero asymptotically.*

Proof. Consider Lyapunov function candidate $V = \frac{1}{2}\left({}^{r}x^2 + {}^{r}y^2\right) + \frac{1}{2\gamma_\theta}\left({}^{r}\theta - {}^{r}\theta_d\right)^2$. By taking time derivative on both sides of it, we obtain $\dot{V} = {}^{r}x\,{}^{r}\dot{x} + {}^{r}y\,{}^{r}\dot{y} + \frac{1}{\gamma_\theta}\left({}^{r}\theta - {}^{r}\theta_d\right)\left({}^{r}\dot{\theta} - {}^{r}\dot{\theta}_d\right)$. Considering (9) and (10), we can get $\dot{V} = -k_x\,{}^{r}x^2 + v_f\,{}^{r}y\sin{}^{r}\theta_d - \frac{k_\theta}{\gamma_\theta}\left({}^{r}\theta - {}^{r}\theta_d\right)^2$. As both $\sin(\cdot)$ and $\tanh(\cdot)$ are odd functions, $v_f\,{}^{r}y\sin{}^{r}\theta_d \le 0$. Therefore, $\dot{V} \le 0$, and $\mathbf{R} = \{{}^{r}\boldsymbol{p}, {}^{r}\theta \mid \dot{V} = 0\} = \{{}^{r}\boldsymbol{p} = \boldsymbol{0}, {}^{r}\theta = 0\}$. It is not difficult to find that $\mathbf{R}$ is an invariant set. Therefore, the maximum invariant set in $\mathbf{R}$ is itself. According to the LaSalle's invariance principle, ${}^{r}\boldsymbol{p}$ and ${}^{r}\theta$ converges to zero asymptotically. This completes the proof. □

V. Experimental Results

To validate the proposed approach, experiments have been conducted on public roads in the campus of Nanyang Technological University, Singapore, using a Husky UGV equipped with a Velodyne Puck Lidar. LOAM was employed for vehicle localization. The algorithms were programmed using C++ under Robot Operating System (ROS), kinetic release, with key parameters set as follows: $N_f = 20$, $d = 1$m, $n = 2$, $I_{\hat{x}_c} = 0.25$m, $v_f = 0.7$m/s, $k_x = 1.0$, $k_\theta = 1.0$, $\gamma_\theta = 1.0$, $\bar{\theta}_d = \frac{\pi}{8}$, and $\gamma_d = 1.0$. The results of three representative experiments are presented below, with the outcomes of the proposed path generation algorithm in



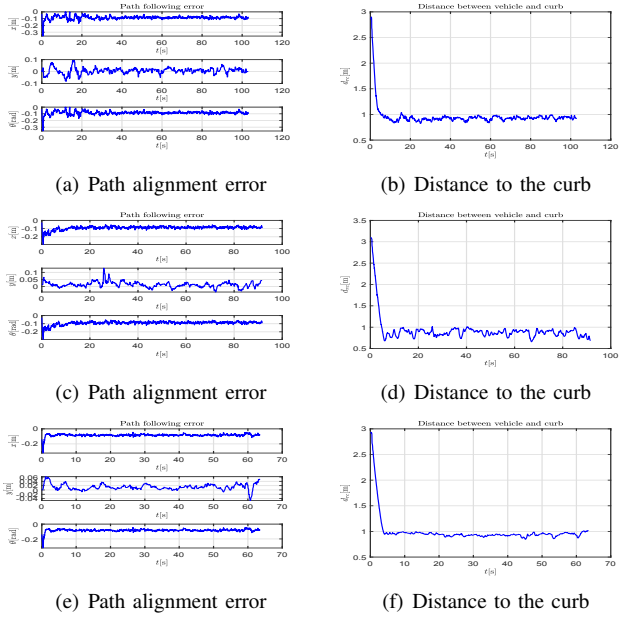
(a) Detected curb points (b) Filtered curb points (c) Reference path (d) Real path

(e) Detected curb points (f) Filtered curb points (g) Reference path (h) Real path

(i) Detected curb points (j) Filtered curb points (k) Reference path (l) Real path

Fig. 5. Results of path generation [(a)-(d) are results of Experiment 1; (e)-(h) are results of Experiment 2; (i)-(l) are results of Experiment 3]

Fig. 5 and the results of the proposed alignment control algorithm in Fig. 6.



(a) Path alignment error (b) Distance to the curb

(c) Path alignment error (d) Distance to the curb

(e) Path alignment error (f) Distance to the curb

Fig. 6. Results of alignment control [(a)-(b) are results of Experiment 1; (c)-(d) are results of Experiment 2; (e)-(f) are results of Experiment 3]

In Experiment 1, the vehicle was tasked to follow a concave curb as shown in Figs. 5(a) to 5(d). Fig. 5(a) showcases the accumulated detected curb points expressed in the global frame, from which numerous noisy points can be observed. These noisy points pose great challenges to path planning, emphasizing the importance of the introduced sliding window processing method. The effectiveness of this method is evident in Fig. 5(b), where the red points retained after the sliding window processing align well with the curb, facilitating subsequent path generation. Figs. 5(a) and 5(b) demonstrate the effectiveness of the proposed sliding window processing module. Fig. 5(c) displays the generated reference path expressed in the global frame, showcasing its continuity and alignment with the curb even planned piecewisely. This validates the proposed path generation algorithm. The real path of the vehicle shown in Fig. 5(d) closely overlaps with the reference path, confirming the efficacy of the alignment control algorithm. Quantitative path alignment performance in Fig. 6(a) demonstrates that position errors along both x and y axes are bounded by 0.1m for most of the time and orientation errors are bounded by 0.1rad after convergence. Figs. 5(d) and 6(a) substantiate the effectiveness of the proposed alignment control algorithm. Fig. 6(b) shows the distance from the vehicle to the curb, indicating consistent maintenance of a distance slightly below 1m. In this figure, we use the nearest originally detected curb points from the vehicle to compute the distance. It is worth noting that due to detection uncertainties, the detected curb points do not locate exactly on the real curb. This causes the error in the vehicle-to-curb distance. These figures demonstrate the effectiveness of the proposed approach in concave curb following scenarios.

Experiment 2 and 3 validate the proposed approach in convex curb following and straight-concave curb following scenarios, respectively, yielding results similar to the first experiment. Due to space limitations, we do not provide detailed explanations here.

VI. Conclusion

This paper has explored the challenge of autonomous curb following and introduced a real-time path generation and

alignment control approach tailored to this problem. The proposed approach can seamlessly integrate with existing curb detection algorithms. Validation of the proposed approach has been conducted through three public road experiments, demonstrating their effectiveness, robustness and practicality. While the primary motivation of this research stems from an environmental service application, road sweeping, it is worth noting that the developed techniques hold relevance for diverse applications, such as side parking, boundary patrolling, and more. In practical scenarios, environments often present complexities and may feature discontinuous curbs, posing significant challenges to consistently accurate curb detection. Future research will delve into this problem.

REFERENCES

[1] NEA, "Proof-of-concept trials of autonomous environmental service vehicles commence in designated testbed environments," 2021. [Online]. Available: https://www.nea.gov.sg/media/news/news/index/proof-of-concept-trials-of-autonomous-environmental-service-vehicles-commence-in-designated-testbed-environments

[2] A. S. Huang and S. Teller, "Lane boundary and curb estimation with lateral uncertainties," in *2009 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2009, pp. 1729–1734.

[3] M. Enzweiler, P. Greiner, C. Knöppel, and U. Franke, "Towards multi-cue urban curb recognition," in *2013 IEEE Intelligent Vehicles Symposium (IV)*. IEEE, 2013, pp. 902–907.

[4] M. Cheng, Y. Zhang, Y. Su, J. M. Alvarez, and H. Kong, "Curb detection for road and sidewalk detection," *IEEE Transactions on Vehicular Technology*, vol. 67, no. 11, pp. 10 330–10 342, 2018.

[5] H. Zhou, H. Wang, H. Zhang, and K. Hasith, "Lacnet: Real-time end-to-end arbitrary-shaped lane and curb detection with instance segmentation network," in *2020 16th International Conference on Control, Automation, Robotics and Vision (ICARCV)*. IEEE, 2020, pp. 184–189.

[6] J. Yu and Z. Yu, "Mono-vision based lateral localization system of low-cost autonomous vehicles using deep learning curb detection," in *Actuators*, vol. 10, no. 3. MDPI, 2021, p. 57.

[7] F. Oniga, S. Nedevschi, and M. M. Meinecke, "Curb detection based on a multi-frame persistence map for urban driving scenarios," in *2008 11th International IEEE Conference on Intelligent Transportation Systems*. IEEE, 2008, pp. 67–72.

[8] J. Siegemund, D. Pfeiffer, U. Franke, and W. Förstner, "Curb reconstruction using conditional random fields," in *2010 IEEE Intelligent Vehicles Symposium*. IEEE, 2010, pp. 203–210.

[9] F. Oniga and S. Nedevschi, "Polynomial curb detection based on dense stereovision for driving assistance," in *13th International IEEE Conference on Intelligent Transportation Systems*. IEEE, 2010, pp. 1110–1115.

[10] ——, "Curb detection for driving assistance systems: A cubic spline-based approach," in *2011 IEEE Intelligent Vehicles Symposium (IV)*. IEEE, 2011, pp. 945–950.

[11] M. Kellner, M. E. Bouzouraa, and U. Hofmann, "Road curb detection based on different elevation mapping techniques," in *2014 IEEE Intelligent Vehicles Symposium Proceedings*. IEEE, 2014, pp. 1217–1224.

[12] G. Zhao and J. Yuan, "Curb detection and tracking using 3d-lidar scanner," in *2012 19th IEEE International Conference on Image Processing*. IEEE, 2012, pp. 437–440.

[13] J. Maye, R. Kaestner, and R. Siegwart, "Curb detection for a pedestrian robot in urban environments," in *2012 IEEE International Conference on Robotics and Automation*. IEEE, 2012, pp. 367–373.

[14] H. Lee, J. Park, and W. Chung, "Curb feature based localization of a mobile robot in urban road environments," in *2015 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2015, pp. 2794–2799.

[15] S. Xu, R. Wang, and H. Zheng, "Road curb extraction from mobile lidar point clouds," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 55, no. 2, pp. 996–1009, 2016.

[16] Y. Zhang, J. Wang, X. Wang, and J. M. Dolan, "Road-segmentation-based curb detection method for self-driving via a 3d-lidar sensor," *IEEE Transactions on Intelligent Transportation Systems*, vol. 19, no. 12, pp. 3981–3991, 2018.

[17] Y. Jung, S.-W. Seo, and S.-W. Kim, "Curb detection and tracking in low-resolution 3d point clouds based on optimization framework," *IEEE Transactions on Intelligent Transportation Systems*, vol. 21, no. 9, pp. 3893–3908, 2020.

[18] G. Wang, J. Wu, R. He, and B. Tian, "Speed and accuracy tradeoff for lidar data based road boundary detection," *IEEE/CAA Journal of Automatica Sinica*, vol. 8, no. 6, pp. 1210–1220, 2020.

[19] J. Guerrero, R. Chapuis, R. Aufrère, L. Malaterre, and F. Marmoiton, "Road curb detection using traversable ground segmentation: Application to autonomous shuttle vehicle navigation," in *2020 16th International Conference on Control, Automation, Robotics and Vision (ICARCV)*. IEEE, 2020, pp. 266–272.

[20] Y. Jung, M. Jeon, C. Kim, S.-W. Seo, and S.-W. Kim, "Uncertainty-aware fast curb detection using convolutional networks in point clouds," in *2021 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2021, pp. 12 882–12 888.

[21] K. Kodagoda, W. S. Wijesoma, and A. P. Balasuriya, "Cute: Curb tracking and estimation," *IEEE Transactions on Control Systems Technology*, vol. 14, no. 5, pp. 951–957, 2006.

[22] J. Tan, J. Li, X. An, and H. He, "Robust curb detection with fusion of 3d-lidar and camera data," *Sensors*, vol. 14, no. 5, pp. 9046–9073, 2014.

[23] S. E. C. Deac, I. Giosan, and S. Nedevschi, "Curb detection in urban traffic scenarios using lidars point cloud and semantically segmented color images," in *2019 IEEE Intelligent Transportation Systems Conference (ITSC)*. IEEE, 2019, pp. 3433–3440.

[24] I. Baek, T.-C. Tai, M. M. Bhat, K. Ellango, T. Shah, K. Fuseini, and R. R. Rajkumar, "Curbscan: Curb detection and tracking using multi-sensor fusion," in *2020 IEEE 23rd International Conference on Intelligent Transportation Systems (ITSC)*. IEEE, 2020, pp. 1–8.

[25] S. Das, N. Mahabadi, S. Chatterjee, and M. Fallon, "Multi-modal curb detection and filtering," *arXiv preprint arXiv:2205.07096*, 2022.

[26] K. Hara, J. Sun, J. Chazan, D. Jacobs, and J. E. Froehlich, "An initial study of automatic curb ramp detection with crowdsourced verification using google street view images," in *First AAAI Conference on Human Computation and Crowdsourcing*, 2013.

[27] Z. Xu, Y. Sun, L. Wang, and M. Liu, "Cp-loss: Connectivity-preserving loss for road curb detection in autonomous driving with aerial images," in *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2021, pp. 1117–1123.

[28] S.-H. Kim, C.-W. Roh, S.-C. Kang, and M.-Y. Park, "Outdoor navigation of a mobile robot using differential gps and curb detection," in *Proceedings 2007 IEEE International Conference on Robotics and Automation*. IEEE, 2007, pp. 3414–3419.

[29] S.-H. Kim, "Tracking control for reliable outdoor navigation using curb detection," *Recent Advances in Mobile Robotics*, 2011.

[30] S. Sivakanthan, J. Castagno, J. L. Candiotti, J. Zhou, S. A. Sundaram, E. M. Atkins, and R. A. Cooper, "Automated curb recognition and negotiation for robotic wheelchairs," *Sensors*, vol. 21, no. 23, p. 7810, 2021.

[31] J. Lee, S. N. Sponberg, O. Y. Loh, A. G. Lamperski, R. J. Full, and N. J. Cowan, "Templates and anchors for antenna-based wall following in cockroaches and robots," *IEEE Transactions on Robotics*, vol. 24, no. 1, pp. 130–143, 2008.

[32] M. Katsev, A. Yershova, B. Tovar, R. Ghrist, and S. M. LaValle, "Mapping and pursuit-evasion strategies for a simple wall-following robot," *IEEE Transactions on robotics*, vol. 27, no. 1, pp. 113–128, 2011.

[33] C.-H. Hsu and C.-F. Juang, "Evolutionary robot wall-following control using type-2 fuzzy controller with species-de-activated continuous aco," *IEEE Transactions on Fuzzy Systems*, vol. 21, no. 1, pp. 100–112, 2012.

[34] C.-F. Juang, Y.-H. Chen, and Y.-H. Jhan, "Wall-following control of a hexapod robot using a data-driven fuzzy controller learned through differential evolution," *IEEE Transactions on Industrial electronics*, vol. 62, no. 1, pp. 611–619, 2014.

[35] L. M. Romero, J. A. Guerrero, and G. Romero, "Road curb detection: a historical survey," *Sensors*, vol. 21, no. 21, p. 6952, 2021.

[36] J. H. Rhee and J. Seo, "Low-cost curb detection and localization system using multiple ultrasonic sensors," *Sensors*, vol. 19, no. 6, p. 1389, 2019.