Perception, Planning and Control for Self-Driving System Based on On-board Sensors

Yanyan Dai and Suk-Gyu Lee

Abstract
The autonomous vehicle can recognize and understand environment, self-control, and achieve the driving level of the human driver. To create this kind of system, the following work was carried out: (a) a real time lane detection system is proposed, based on vision system functions, using webcam camera; (b) In order to detect curve lanes, deep learning is applied for lane detection, based on fully Convolutional Neural Network (CNN); (c) The cubic spline interpolation method is used for path generation, based on Global Positioning System (GPS) data, where distance between two adjacent path generation points is same. Compared with the connection method of the cubic polynomial fitting algorithm, the curve fitted path by the cubic spline interpolation method is smoother and more satisfied with the vehicle motion pattern; (d) Based on frenet coordinate frame, optimize trajectory planning method is used to plan safe, easy and comfort trajectory. Compared to the Cartesian coordinate frame, frenet coordinate frame simplifies the solution of road curve fitting problems, especially in the case of complex road environment; (e) Fuzzy sliding mode control method is proposed for vehicle steering control. Finally, the simulation and experiment results are used to prove the effectiveness of the proposed methods.

Keywords
Autonomous vehicle system, real time lane detection, lane detection using fully CNN, path generation, trajectory planning, fuzzy sliding mode control method for steering control

Date received: 3 March 2020; accepted: 24 July 2020

Handling Editor: James Baldwin

Introduction
Autonomous vehicle systems have been a popular topic of research. The technical preparation of self-driving vehicles is maturing, with the increase in available computing power and the reduction in sensor costs. The field of self-driving technology is not only related to vehicle control, path planning, sensory fusion, but also forewords such as artificial intelligence, machine learning, deep learning, and reinforcement learning. Self-driving will start a new technology and market revolution, in the next 5 to 10 years. Autonomous driving has been of interest to researches to push the boundaries of automotive racking, that is limited by the abilities of human drivers, with several companies and academics. Stanford University developed an off-road autonomous vehicle for the DARPA Grand Challenge. Hanyang University developed an autonomous off-road Hyundai “A1” to compete in the Autonomous Vehicle Competition (AVC). Google has tested its self-driving vehicles over 2 million miles, using
100 self-driving vehicles. Tesla provides self-driving function in their 2016 model S cars. Uber’s plans to eventually replace the human driven fleet with self-driving cars.

The core of the self-driving system, including three parts: perception, planning, and control. Self-driving system technology means that: self-driving car can recognize the surrounding environment and status of the vehicle through a variety of on-board sensors (camera, lidar, millimeter wave radar, GPS, inertial sensors, etc.), and make analysis and judgment, then autonomously control vehicle movement, and finally achieve self-driving based on the acquired environmental information (including road information, traffic information, vehicle location and Obstacle information, etc.) This paper proposes methods for self-driving system in perception, planning and control, based on on-board sensors.

In lane detection and tracking, several issues should be considered, namely choosing an appropriate lane model, developing an image processing for features extraction, probability of lane tracking, and effectiveness of real-time image processing ability. Based on the three main methods for lane detection using cameras includes: model of the lane markings, color information, and feature information method including edge, gradient and intensity. About road information, it makes the mathematical model of the lane markings, such as B-spline, where candidate points are extracted from the lane markings. About color information, it applies a threshold to the image based on RGB and HSL color space, and places an 8-bit image. To extract a lane, color threshold is based on RGB color space or HSL color space. Due to simplification of computer monitor design, the RGB color space is the most commonly used color space. However, it cannot be used for all applications. In order to describe a color in the RGB color space, the red, green, and blue color components are all used, so RGB is not as intuitive as other color spaces. The HSL color space uses only the hue component to describe color. Therefore, HSL is the best color space for image processing applications, for example color matching. In this paper, a real-time lane detection system is proposed, which uses the capabilities of vision system functions, based on combination of feature information and color information.

The real-time lane detection system works well for clear straight lane. However, on curved lanes or sharp turns, the real time lane detection system would fail, due to complexity of the lane markers. The lane detector cannot distinguish the lane edge well. The lane detection system based on Convolutional Neural Network (CNN) can overcome this problem. CNN is the most popular deep learning architecture, due to the effectiveness of convnets. The main advantage of CNN is that it can automatically detect the important features without any human supervision. CNN is also computationally efficient. This enables CNN models to run on any device.

Path planning can be divided into two steps: global path planning and local path planning. Global path planning deals with generating the most ideal route to reach the final destination. Local path planning is responsible for obstruction avoidance and path planning for each segment generated by the global path planner. There are some methods for local path planning, such as artificial potential field method, A* algorithm, and discrete optimization approach. The artificial potential method is calculated by the attractive and repulsive potential functions. However, the computations are complex and sometimes unsolvable, because the path is described in high order polynomial. In addition, it does not always yield a path to the goal due to local minima in the vector field. For A* algorithm, it does not work well on lane tracking based on driving with few obstacles. The discrete optimization approach calculates a finite number of paths. A cost is computed for each path, and finally the path with least cost is selected. In this paper, a discrete optimization method is used, based on the frenet coordinate system for autonomous vehicle path planning, which has strong practicability in assisted driving and self-driving.

Fuzzy logic is widely used in motion control, because it provides the ability to control system with construction of complex mathematical model. Fuzzy logic is used for mobile robot motion control, where sensors of mobile robot contain uncertain and incomplete information. Fuzzy logic technology is also used in traffic management. There is applicability issue to create models of traffic, but IF-THEN logic can deal with vague expectations and respond to situations. In this paper, fuzzy sliding mode control method is used to minimize vehicle’s steering angle error. The vehicle’s cross track error is defined as $\delta_e$. An angular velocity control law is designed to make the cross-track error to zero.

The contribution of this paper is: (a) A real time lane detection system is proposed based on vision system functions, where image acquisition and processing functions are used to extract lanes. Although there exists lane detection algorithms in previous work, the proposed lane detection system processes fast and the performance is good. (b) In order to detect curve lanes, deep learning is applied for lane detection, based on fully CNN. It shows how to perform lane segment detection by training a deep neural network for lane segmentation. Compared with, in this paper, the number of layers is increased, and the number of convolution kernels is modified in each layer. A parallel jump structure is used to connect the encoding feature map to the decoding feature map. UpSampling2D is changed to Conv2D Transpose to implement the upsampling process. (c) The cubic spline interpolation
method is used for path generation, based on GPS data, where distance between two adjacent path generation points is same. Compared with the connection method of the cubic polynomial fitting algorithm, the curve fitted path by the cubic spline interpolation method is smoother and more satisfied with the vehicle motion pattern. (d) Optimize trajectory planning method depending on frenet coordinate frame is used to plan the safe, easy and comfort trajectory. In addition, compared to the Cartesian coordinate frame, frenet coordinate frame simplifies the solution of road curve fitting problems, especially in the case of complex road environment. Compared with pure pursuit algorithm, the result shows that the path planned by optimize trajectory planning method is more closed to the predefined path. (e) fuzzy sliding mode control method is used to design angular velocity control law to minimize vehicle’s steering angle error. (f) the simulation and experiment results show effectiveness of the proposed methods.

In this paper, section 2 gives the modeling of the vehicle. Section 3 describes two lane detection systems. Section 4 discusses the path generation and path planning methods. Section 5 outlines the fuzzy sliding mode method for vehicle steering control. Section 6 and Section 7 show the simulation result and experiment result, respectively.

**Modeling**

The local and global coordinate of the vehicle is shown as Figure 1. $X_0MY_1$ is the local coordinate frame. $X_gOY_g$ is the global coordinate frame. $M_r$ is the central point of rear axle, which is the original point of local coordinate frame. $M_f$ is the central point of front axle. The angular difference between the global and local coordinate is given by $\theta$. The orientation of the local coordinate frame about the global coordinate frame is expressed by the rotation matrix as

$$R(\theta) = \begin{bmatrix}
\cos \theta & \sin \theta & 0 \\
-\sin \theta & \cos \theta & 0 \\
0 & 0 & 1
\end{bmatrix}$$ (1)

The local coordinate of the system is shown in Figure 2, where $l$ is the wheelbase, $\phi$ is wheel steering angle, $d$ is the distance between two rear wheels, ICR is instantaneous center of rotation, and $d_r$ and $d_f$ are instantaneous radius of rotation of $M_r$ and $M_f$, respectively. The overall speed of the vehicle $v_r$ is the speed of rear wheels, along with the axes direction of $X_l$ of local coordinate frame. $v_f$ is the speed of front wheels. The relationship of $v_r$ and $v_f$ is:

$$v_f = v_r / \cos \phi$$ (2)

In Figure 3, point w is the center gravity point of the car. The configuration of the vehicle about point W is defined as $[x, y, \delta]^T$ in global coordinate. $x$, $y$ and $\delta$ are
the x coordinate, y coordinate, and orientation of the vehicle, respectively. $v$ is the speed of the vehicle. $w_f$ and $w_r$ are the distances between the center gravity point and the front wheel and the rear wheel, respectively. The kinematic model of the vehicle is calculated as

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\delta} \end{bmatrix} = \begin{bmatrix} v \cos(\delta + \beta) \\ v \sin(\delta + \beta) \\ \frac{x}{w_r} \sin(\beta) \end{bmatrix}$$

where $\beta$ is calculated as:

$$\beta = \arctan\left(\frac{w_r}{w_r + w_f} \tan(\varphi)\right)$$

**Lane detection system**

**Real time lane detection system**

A webcam camera is equipped in front of the vehicle. Using the camera, the vehicle detects lane, and tracks lane. The real time lane detection system flowchart is shown in Figure 4. Firstly, the camera acquires image. In the second block, the image is resized to 640*480 for image processing. Because on the road, there are always two lanes. The lanes are straight or curve. The lanes are also solid lanes or dash lanes. Based on lane characteristics, four rectangle regions of interest (ROI) are defined to focus on lane area and analyze the lane data. Two ROIs are used to focus on left lane, and two ROIs are for the right lane, in order to detect dash and curve lane. Color threshold block converts a color image to a binary image. If the color components are in the specified range, the pixel value of the original image will be set to 1. Otherwise, the pixel value is equal to 0. The color threshold, based on HSL color space, is used to extract lane. Hue channel is used to detect yellow lane. Luminance channel is for white lane detection. In order to remove or enhance isolated feature, grayscale morphology is used. Two functions are used in this block: erosion and dilation. Erosion reduces the brightness of pixels surrounded by neighbors with a lower intensity. Dilation increases the brightness of pixels, which are surrounded by neighbors with a higher intensity. Based on the environment condition, erosion or dilation is chosen for gray morphology. Finding edge block is used to find a line describing the edge with points along the edge, and output the coordinate information of the points.

Converting pixel to real world is applied to transform pixel coordinate to real-world coordinate. For obtaining lane tracking points block, the algorithm is processed as Figure 5. $B_i(x_{hi},y_{hi})$ and $C_i(x_{ci},y_{ci})$ are the edge points coordinates, based on finding edge and converting pixel to real world blocks. The tracking points $A_i(x_{ai},y_{ai})$ is based on its configuration in (3) and (4), and calculated as:

$$x_{ai} = \frac{1}{2}(x_{hi} + x_{ci})$$

$$y_{ai} = \frac{1}{2}(y_{hi} + y_{ci})$$

**Lane detection system using fully CNN**

In this section, deep learning is applied to detect lanes based on fully CNN. It shows how to perform lane segment detection by training a deep neural network for lane segmentation. SegNet is an encoding-decoding structure. It uses different method in upsampling and downsampling. Based on, Figure 6 shows pipeline used in lane detection system. The neural network architecture includes seven convolutional layers, three polling layers, three upsampling layer and seven deconvolution layers. However, this model is not stable during lane detection, so the model is improved in this paper as shown in Figure 7. Different from network in Figure 6, the number of layers is increased in the network. The number of convolution kernels is modified in each layer. A parallel jump structure is used to connect the encoding feature map to the decoding feature.
map, so that multi-layer information can be used in classification prediction. UpSampling2D is changed to Conv2D Transpose to implement the upsampling process, since Conv2DTranspose has the process of learning, which has better result. The BDD100K data is used, consisting of images and labels. For training, the dataset is divided into training set and validation set. The neural network architecture includes 10 convolutional layers, four pooling layers, four concatenate layers and 10 deconvolution layers. The model is compiled with mean squared error as the loss function. And the model is trained on the training set, and validated on the validation dataset. Images are inputs of prediction process. Using the trained model result, the prediction process outputs images with regions of the detected lane.
Path generation and path planning

Cubic spline interpolation model

Autonomous vehicles do path planning and tracking algorithm based on GPS data. However, due to communication delay, the GPS data cannot be obtained continuously. Distances between each two conjunction GPS points are not same. Therefore, if the GPS data is directly used to do the path planning and tracking algorithm, without path generation algorithm, the speed of the vehicle is always changed. In order to control autonomous vehicle smoothly and safely movement, constant speed is pursued for the vehicle. The cubic spline interpolation method is used for path generation, based on GPS data, where distances between two adjacent path generation points are the same. Compared with the connection method of the cubic polynomial fitting algorithm, the curve fitted path by the cubic spline interpolation method is smoother and more satisfied with the vehicle motion pattern. The cubic polynomial fitting algorithm produces a curve arc in some straight line's segments. This is not the expected vehicle operating characteristic. Based on vehicle kinematics, the vehicle should take a straight line as much as possible. Only in the direction of the transition, a curved path, which matches the vehicle turning characteristics, is needed. The feature of cubic spline interpolation method is as:

1. The cubic spline curve is continuous and smooth at the joint.
2. The first and second derivatives of the cubic spline are continuous.
3. The second derivative of the boundary of the natural cubic spline is continuous.
4. A single point does not affect the entire interpolation curve.

Three points \( S_1 = (x_1, y_1) \), \( S_2 = (x_2, y_2) \), and \( S_3 = (x_3, y_3) \) are defined to be fitted. The cubic functions to fit points \( S_1 \) and \( S_2 \), to fit points \( S_2 \) and \( S_3 \) are calculated as:

\[
y = ax^3 + bx^2 + cx + d \quad (7)
\]

\[
y = ex^3 + fx^2 + gx + h \quad (8)
\]

Equation (7) passes through \( S_1 \) and \( S_2 \), and equation (8) passes through \( S_2 \) and \( S_3 \), so the following equations can be obtained:

\[
y_1 = ax_1^3 + bx_1^2 + cx_1 + d \quad (9)
\]

\[
y_3 = ex_3^3 + fx_3^2 + gx_3 + h \quad (10)
\]

\[
a x_1^3 + bx_1^2 + cx_1 + d = ex_2^3 + fx_2^2 + gx_2 + h = y_2 \quad (11)
\]

Based on feature 2, the first derivative of both sides of equation (11) can be obtained as:

\[
3ax_1^2 + 2bx_1 + cx_1 = 3ex_2^2 + 2fx_2 + g \quad (12)
\]

Based on feature 3, the second derivative at the start and end points should be continuous, the following are obtained:

\[
6ax_1 + 2b = 0 \quad (13)
\]

\[
6ex_3 + 2f = 0 \quad (14)
\]

Supposed that there are \( n + 1 \) path points: \( (x_0, y_0), (x_1, y_1), (x_2, y_2), \ldots, (x_n, y_n) \). The difference between two conjunction points is calculated as

\[
h_i = x_{i+1} - x_i, i = 0, 1, \ldots, n + 1 \quad (15)
\]

Combining the above six equations (9) to (14), the parameters \( a, b, c, d \) in the cubic spline interpolation function can be calculated.

\[
a_i = y_i \quad (16)
\]

\[
b_i = \frac{y_{i+1} - y_i}{h_i} - \frac{h_1}{2}m_i - \frac{h_i}{6}(m_{i+1} - m_i) \quad (17)
\]

\[
c_i = \frac{m_i}{2} \quad (18)
\]

\[
d_i = \frac{m_{i+1} - m_i}{6h_i} \quad (19)
\]

where \( m_i \) is the solution of the following matrix equation.

\[
\begin{bmatrix}
1 & 0 & 0 & 0 & \cdots & 0 \\
h_0 & 2(h_0 + h_1) & h_1 & 0 & \cdots & 0 \\
0 & h_1 & 2(h_1 + h_2) & h_2 & \cdots & 0 \\
0 & 0 & h_2 & 2(h_2 + h_3) & \cdots & h_3 \\
\vdots & \vdots & \vdots & \ddots & \ddots & \vdots \\
0 & 0 & 0 & \cdots & 2(h_{n-2} + h_{n-1}) & h_{n-1} \\
0 & \cdots & 0 & \cdots & 0 & 1
\end{bmatrix}
\begin{bmatrix}
m_0 \\
m_1 \\
m_2 \\
m_3 \\
\vdots \\
m_{n-1} \\
m_n
\end{bmatrix} = \begin{bmatrix}
0 \\
\frac{y_2 - y_0}{h_2} \\
\frac{y_3 - y_1}{h_3} \\
\frac{-y_0 + y_2}{h_1} \\
\frac{y_1 - y_3}{h_2} \\
\frac{-y_3 + y_{n-2}}{h_n} \\
\frac{y_{n-1} - y_{n-3}}{h_2}
\end{bmatrix} \quad (20)
\]
Optimization of trajectory planning method based on frenet coordinate frame

Frenet coordinate frame. The centerline of the road is defined as the road reference line. As shown in Figure 8, a frenet coordinate frame is established using the tangent vector $t$ of the reference line and the normal vector $n$. The coordinate frame is based on the starting road center line, and the coordinate axes are perpendicular to each other. The direction of the road reference line is $s$ direction, which is called longitudinal direction. The normal direction of the reference line is $d$ direction, which is called lateral direction. Compared to the Cartesian coordinate frame, frenet coordinate frame simplifies the solution of road curve fitting problems, especially in the case of complex road environment, since it is easy to find the road reference line and describe the location of the vehicle, based on the longitudinal direction and lateral direction. Compared with the Cartesian coordinate system, the frenet coordinate system simplifies the solution of the road curve fitting problem. In the road environment, we can always find the reference line of the road, which is the center line of the road. Therefore, the vehicle position, based on the reference line, can be simply represented using the distance along the longitudinal direction and the distance along the lateral direction. The calculation of the speed in both directions is also relatively simple, based on frenet coordinate system.

Optimization trajectory planning method. Due to lane change or obstacle avoidance, there is a certain lateral offset between the car and the desired lane. The car should return to its desired lane, based on the best compromise result between the comfort experience in the car and the time required to reach the desired lane position. In longitudinal direction, if the car is moving too fast or too close to the front car, it must slow down significantly without rushing. Comfort can be described by the jerk, which is difference of acceleration. Based on, a quantic polynomial is the jerk-optimal connection of a start state $P_0 = [d_0, \dot{d}_0, \ddot{d}_0]$ and an end state $P_1 = [d_1, \dot{d}_1, \ddot{d}_1]$ within the time interval $T = t_1 - t_0$ in a one-dimensional problem. The jerk can be described as:

$$J_s(p(t)) = \int_{t_0}^{t_1} p^2(\tau)d\tau$$

where $p(t)$ is described as:

$$p(t) = \alpha_0 + \alpha_1t + \alpha_2t^2 + \alpha_3t^3 + \alpha_4t^4 + \alpha_5t^5$$

Trajectory plan in lateral direction. In Lateral Direction, define the start state as $D_0 = [d_0, \dot{d}_0, \ddot{d}_0]$, based on equation (22), the lateral offset, lateral velocity, and lateral acceleration of the start state are described as

$$d_0 = d(t_0) = \alpha_{d0} + \alpha_{d1}t_0 + \alpha_{d2}t_0^2 + \alpha_{d3}t_0^3 + \alpha_{d4}t_0^4 + \alpha_{d5}t_0^5$$

$$\dot{d}_0 = \dot{d}(t_0) = \alpha_{d1} + 2\alpha_{d2}t_0 + 3\alpha_{d3}t_0^2 + 4\alpha_{d4}t_0^3 + 5\alpha_{d5}t_0^4$$

$$\ddot{d}_0 = \ddot{d}(t_0) = 2\alpha_{d2} + 6\alpha_{d3}t_0 + 12\alpha_{d4}t_0^2 + 20\alpha_{d5}t_0^3$$

Define the end state as $D_1 = [d(t_1), \dot{d}(t_1), \ddot{d}(t_1)]$, the lateral offset, lateral velocity, and lateral acceleration of the end state are described as

$$d_1 = d(t_1) = \alpha_{d0} + \alpha_{d1}t_1 + \alpha_{d2}t_1^2 + \alpha_{d3}t_1^3 + \alpha_{d4}t_1^4 + \alpha_{d5}t_1^5$$

$$\dot{d}_1 = \dot{d}(t_1) = \alpha_{d1} + 2\alpha_{d2}t_1 + 3\alpha_{d3}t_1^2 + 4\alpha_{d4}t_1^3 + 5\alpha_{d5}t_1^4$$

$$\ddot{d}_1 = \ddot{d}(t_1) = 2\alpha_{d2} + 6\alpha_{d3}t_1 + 12\alpha_{d4}t_1^2 + 20\alpha_{d5}t_1^3$$

Define $t_0 = 0$, $\alpha_{d0}$, $\alpha_{d1}$, and $\alpha_{d2}$ are calculated as

$$\alpha_{d0} = d(t_0)$$

$$\alpha_{d1} = \dot{d}(t_0)$$

$$\alpha_{d2} = \ddot{d}(t_0)$$

$\alpha_{d3}$, $\alpha_{d4}$, and $\alpha_{d5}$ can be calculated based on the following matrix

![Frenet coordinate frame](image-url)
where $k_j, k_t, k_d \geq 0$. Item $k_p J_s(d(t))$ is used to penalize the candidate trajectory with higher Jerk. Item $k_t T$ represents the breaking time. Item $k_d d_i^2$ describes the difference between the goal state with the reference line. For the optimizing trajectory, the vehicle should move in parallel to the center line. So $d_i$ is a constant and $d_i = d_i^* = 0$. Combining different end conditions $d_i \in (d_{\min}, d_{\max})$ and $T_j \in (T_{\min}, T_{\max})$, the candidate trajectories can be calculated as

$$
[d_i, d_i^*, d_i^*, T_j]_{ij} = [d_i, 0, 0, T_j]
$$

where $d_{\min}, d_{\max}$ and $T_{\min}, T_{\max}$ are the constraints for $d_i$, $d_i^*$, and $T_j$, respectively. Define $\Delta d$ and $\Delta T$ as sampling density, equations (35) and (36) can be described as:

$$
d_{i+1} = d_i + \Delta d
$$

$$
T_{j+1} = T_j + \Delta T
$$

Trajectory plan in longitudinal direction. In order to plan trajectory in longitudinal direction, three conditions are considered: following, merging and stopping, and keeping at a constant speed. To calculate the cost function, for the following, merging and stopping cases, the start state is set as $S_0 = [s_0, s_0, s_0]$, and the target position $s_{\text{target}}$ is defined. For the velocity keeping case, the target velocity $s_{\text{target}}$ is defined.

For the following cases, the target position is defined as:

$$
s_{\text{target}}(t) = s_{lv}(t) - [D_0 + \tau \dot{s}_{lv}(t)]
$$

where $D_0$ and $\tau$ are constants. $s_{lv}$ and $\dot{s}_{lv}$ are the position and velocity of the leader vehicle, which are described as below, with assumption $s_{lv}(t) = \ddot{s}_{lv}(t) = \text{const.}$

$$
\dot{s}_{lv}(t) = \ddot{s}_{lv}(t_0) + \dot{s}_{lv}(t_0)[t - t_0]
$$

$$
s_{lv}(t) = s_{lv}(t_0) + \dot{s}_{lv}(t_0)[t - t_0] + \frac{1}{2} \ddot{s}_{lv}(t_0)[t - t_0]^2
$$

The first derivative and second derivative of $s_{\text{target}}(t)$ are given as

$$
\dot{s}_{\text{target}}(t) = \dot{s}_{lv}(t) - \tau \ddot{s}_{lv}(t)
$$

$$
\ddot{s}_{\text{target}}(t) = \ddot{s}_{lv}(t_1) - \tau \dddot{s}_{lv}(t) = \dddot{s}_{lv}(t)
$$

For merging case, the target position is defined as:

$$
s_{\text{target}}(t) = \frac{1}{2} [s_0(t) + s_h(t)]
$$

where $s_0(t)$ and $s_h(t)$ are pair of vehicles’ positions, which the vehicle should merge.

For stopping case, the target position is defined as:

$$
s_{\text{target}}(t) = s_{\text{stop}}(t)
$$

where $s_{\text{stop}} = s_{\text{stop}} = 0$. Combining different end conditions $\Delta d_s$ and $T_j$ as below, candidate trajectories can be computed.

$$
[s_1, \dot{s}_1, s_1, T]_{ij} = [s_{\text{target}} T_j + s_1, \dot{s}_{\text{target}} T_j + \dot{s}_1, T_j]
$$

For these two cases, the cost function is expressed as:

$$
C_s = k_p J_1(s(t)) + k_s [s_1 - s_{\text{target}}]_2
$$

In the keeping at a constant speed case, for a start state $S_0 = [s_0, \dot{s}_0, s_0]$ and an end state $S_1 = [s_1, \dot{s}_1]$, combining different end conditions $\Delta d_t$ and $T_j$ as below, candidate trajectories can be computed.

$$
[s_1, \dot{s}_1, T]_{ij} = [s_{\text{target}} T_j + \Delta s_1, 0, T_j]
$$

The cost function is expressed as:

$$
C_v = k_p J_1(s(t)) + k_s [\Delta s_1 - s_{\text{target}}]_2
$$

Finally, comparing the cost function in lateral direction and the cost function in longitudinal direction as below, the best trajectory can be selected.

$$
C_{\text{total}} = k_{\text{lat}} C_d + k_{\text{lon}} C_v
$$

where $k_{\text{lat}}$ and $k_{\text{lon}}$ are the constants. $C_d$ and $C_v$ are the cost function in lateral direction and in longitudinal direction. The trajectory, which has the minimize value of $C_{\text{total}}$, is the optimization trajectory.

Fuzzy sliding control method for vehicle steering

The vehicle movement along a predefined trajectory is expected in constant speed. If constant steering angle is used to control the vehicle, the vehicle’s movement will oscillate. For solving the problem, we propose a fuzzy sliding mode control method to control vehicle’s
steering angle. Define the vehicle’s cross track error as $\delta_{IE}$. The switching function $s_i$ is as:

$$s_i = \delta_{IE}$$ (49)

The exponential reaching law is designed as

$$\dot{s}_i = -k_1 \text{sgn} s_i - k_2 s_i$$ (50)

where $k_1 > 0$, $k_2 > 0$, $\text{sgn} s_i = \frac{s_i}{|s_i|}$, $\sigma$ is a positive constant.

The control law is defined as

$$\omega_i = \omega_{iw} + k_1 \text{sgn} s_i + k_2 s_i$$ (51)

In order to prove the stability of the control law, the Lyapunov function is designed as

$$V_i = \frac{1}{2} \delta_{IE}^2$$ (52)

The derivative of the Lyapunov function is

$$\dot{V}_i = \theta_{IE} \theta_{IE} s \dot{s} = -k_1 |s_i| - k_2 s_i^2 \leq 0$$ (53)

When $\delta_{IE} = 0$, $\dot{V} = 0$. When $\theta_{IE} \neq 0$, $\dot{V}_i < 0$. Therefore, the system is asymptotically stable.

$k_1$ should be large to have fast trending motion from system status to sliding surface, if a moving point is far from the sliding surface. Otherwise, $k_1$ should be small to have slow trending motion to weaken the charting. Therefore, the fuzzy sliding mode controller is designed, where the fuzzy rule is described as Table 1. Seven fuzzy variables are used to describe the fuzzy partitions, as positive big (PB), positive medium (PM), positive small (PS), zero (ZO), negative small (NS), negative medium (NM), negative big (NB).

The triangular function is defined as:

$$f(x) = \begin{cases} \max \left(0, 1 - \frac{2(a-x)}{w}\right), & x \leq a \\ \max \left(0, 1 - \frac{2(x-a)}{w}\right), & x > a \end{cases}$$ (54)

The output is based on centroid defuzzification method, and is given as (55). $\mu(\bar{k}_1)$ and $\mu(\bar{k}_2)$ are the membership value for point $\bar{k}_1$ and point $\bar{k}_2$, respectively.

$$\bar{k}_1 = \frac{\sum_{i=1}^{7} \bar{k}_1 \mu(\bar{k}_1)}{\sum_{i=1}^{7} \mu(\bar{k}_1)}, \quad \bar{k}_2 = \frac{\sum_{i=1}^{7} \bar{k}_2 \mu(\bar{k}_2)}{\sum_{i=1}^{7} \mu(\bar{k}_2)}$$ (55)

The fuzzy exponential reaching law is defined as

$$\dot{s}_i = -\bar{k}_1 \text{sgn} s_i - \bar{k}_2 s_i$$ (56)

The fuzzy sliding mode control law is as

$$\omega_i = \omega_{iw} + \bar{k}_1 \text{sgn} s_i + \bar{k}_2 s_i$$ (57)

**Simulations**

**Real time lane detection simulation**

The powerful capabilities of LabVIEW are used to simulate lane detection. The original figure is shown in Figure 9. The vehicle should detect yellow lane and white lane, simultaneously. To show the effectiveness of lane detection system, the figure is chosen, where lanes are not shown clearly. The vehicle has to use vision functions to remove noise. As shown in Figure 10, a rectangle image is extracted from the figure, which only contains lane’s information. The rectangle contains the left-top and right-bottom coordinates as (0, 240) and (640, 480), respectively. Based on Figure 4, four ROIs are defined as up-left, down-left, up-right, and down-right. In Figure 10, four ROIs are shown as green rectangles. NI Image Acquisition (IMAQ) color threshold function is applied to threshold HSL image into an 8-bit image. Hue, saturation and Luminance ranges are set as (85, 252), (106, 252), and (0, 255) to threshold the color, and the lanes are presented as white color.

![Figure 9](image-url) **Figure 9.** Original figure for lane detection.
IMAQ find edge function is used to find straight edges within a region in an image. In up-left, down-left and down-right ROIs, the edges of lanes are detected and the detected lane is marked as red color. Gray morphology function performs grayscale morphological transformations. It is used to remove noise. IMAQ edge tool is used to find edges along the detection lane in the image. The detection lane’s points are as pink points in Figure 9. The lane’s information is in pixel coordinates. The lane’s information is transformed in real world coordinates using the calibration information. Based on Figure 5 and equations (5) and (6), the yellow lane in Figure 9 is the tracking lane for vehicle.

**Fully CNN lane detection simulation**

Based on description in Figure 7, the model is trained. After 10 epochs, the model is finished with MSE for training of 0.0024 and validation of 0.0024. After training, the result is saved in a json along with its weights. Two images are tested for lane detection, using deep learning based on CNN. The original images are sized as 640×480. Firstly, the images are resized to 80×160. Based on the training result, after prediction process, two images are shown in Figures 11 and 12, where the detected lane region is filled in green.

Based on the fully CNN method in,21 after training the model, test results are shown in Figures 13 and 14. After 20 epochs, the model finished with MSE for training of 0.0046 and validation of 0.0048. The results are worse than the results in Figures 11 and 12.

**Trajectory planning and tracking simulation**

In Figure 15, the reference points are defined as (0, 0), (0, –4), (20.5, 1), (30, 6.5), (40.5, 8), (50, 10) and (60, 6) in meters to generate the trajectory. Based on cubic spline interpolation method, the white trajectory is generated, where the white trajectory passes through the reference points. In the white trajectory, distances between two adjacent points are the same. Therefore, the vehicle can move in the constant speed. Obstacles are located at (20, 10), (30, 6), (30, 8), (35, 8), and (50, 3). When the vehicle tracks the white trajectory, the
vehicle should avoid the obstacles. Using the optimization trajectory planning method based on frenet coordinate frame, the best trajectory is planned as the red lane. The trajectory avoids obstacles and finally tracks the generated path. Using the fuzzy sliding mode method, the vehicle’s yaw angle error is minimized as shown in Figure 16. In the frenet coordinate frame, initially, the vehicle is located at (1, 0). Firstly, the vehicle should track the white path from its initial point, so error occurs. At time step 26-42, the vehicle should avoid obstacle, instead of tracking the white trajectory. Therefore, the vehicle should change its orientation to avoid obstacle, error occurs. However, finally, the vehicle yaw angle is close to zero.

As shown in Figure 17, pure pursuit trajectory planning method is compared with frenet optimized trajectory planning method. Passing through points (0, 0), (8, –4), (12, 2), (16, 4), (20, 2), (24, 0) and (27, 3) in meter, path is generated as blue lane in Figure 17. Based on frenet optimized trajectory planning method, the trajectory is planned as red lane. It is almost the same as the generated blue lane. However, based on pure pursuit trajectory planning method, green lane is planned. There is difference between the planned green lane and the generated blue lane, in x range of (0, 3).

In order to show the effectiveness of the trajectory, the proposed trajectory planning and tracking method is tested using a simulator. The simulator is made to demonstrate the performance on an outdoor autonomous vehicle. As shown in Figure 18(a), the left window is the simulator window, and the right window is shown the path based on the algorithm processing in LabVIEW. Based on the proposed path planning algorithm and fuzzy sliding control method for vehicle steer, the speed and steer commands are calculated. Then the commands are sent to the simulator based on RS232 communication. The vehicle in the simulator will move based on the speed and steer commands. Initially, the vehicle is located at (10 m, 0, 0). The light blue obstacle is located at (28 m, 0). The length of the vehicle is
around 1.1 m. The target speed is defined as 10 km/h. The green lane is described as the generated path, which guarantees that the vehicle avoids the obstacle well. The dark blue lane is shown as the vehicles actual path. In Figure 18(b), the vehicle arrives at (20 m, –1 m) to avoid the predefined obstacle. In Figure 18(c), the vehicle tracks back to its predefined straight line. In Figure 18(d), the vehicle moves forward to its predefined straight line. The test simulator video is shown in https://www.dropbox.com/s/5up9w26hra799xv/path%20tracking.mp4?dl=0.

**Experiment**

The experiment is used to show the effectiveness of the trajectory planning method. As shown in Figure 19, the length of curved lanes is 4 m. The distance between two black lanes is 1.1 meter. Firstly, a curve path is generated, using cubic spline interpolation model. Then, the vehicle plans and tracks the trajectory successfully, based on optimization trajectory planning method in frenet coordinate frame. At t = 20 s, the vehicle passes through the curved part of the lane. At t = 30 s, the vehicle finishes the full path.

**Conclusions**

In this paper, a real time lane detection system is proposed based on vision system functions, using webcam camera. In order to detect curve lanes, deep learning is applied for lane detection, based on fully CNN. The cubic spline interpolation method is used for path generation, based on GPS data, where distances between two adjacent path generation points are the same. Optimization trajectory planning method depending on frenet coordinate frame is used to plan a safe, easy and comfort trajectory. In addition, compared to the Cartesian coordinate frame, frenet coordinate frame simplifies the solution of road curve fitting problems, especially in the case of complex road environment. Fuzzy sliding control method for
vehicle steering control is proposed. Finally, the simulation and experiment results show the effective performance of the proposed methods.

Declaration of conflicting interests
The author(s) declared no potential conflicts of interest with respect to the research, authorship, and/or publication of this article.

Funding
The author(s) received no financial support for the research, authorship, and/or publication of this article.

ORCID iD
Yanyan Dai https://orcid.org/0000-0003-2484-4585

References
1. Pendleton S, Andersen H, Du X, et al. Perception, planning, control, and coordination for autonomous vehicle. *Machines* 2017; 5: 1–54.
2. Thrun S, Montemerlo M, Dahlkamp H, et al. Stanely: the robot that won the DARPA Grand Challenge. *J Field Robot* 2006; 23: 661–692.
3. Chu K, Lee M and Sunwoo M. Local path planning for off-road autonomous driving with avoidance of static obstacles. *IEEE Trans Intell Transp Syst* 2012; 13: 1599–1616.
4. Waymo. Google self-driving car project, https://www.google.com/selfdrivingcar/
5. Tesla. Tesla motors: Model S press kit, https://www.tesla.com/presskit/
6. Newton C. Uber will eventually replace all its drivers with self-driving cars. *Verge* 2014; 5: 2014.
7. Peleshko D, Ivanov Y, Sharov B, et al. Design and implementation of visitors queue density analysis and registration method for retail video surveillance purposes. In: *Proceedings of the IEEE first international conference on data stream mining & processing*, Lviv, 23–27 August 2016, pp. 159–162. IEEE.
8. Nguyen YQ, Kim HS, Jun SC, et al. A study on real-time detection method of lane and vehicle for lane change assistant system using vision system on highway. *Int J Eng Sci Technol* 2018; 21: 822–833.
9. Gao DZ, Li W, Duan JM, et al. A practical method of road detection for intelligent vehicle. In: *Proceedings of the 2009 IEEE international conference on automation and logistics*, Shenyang, 5–7 August 2009, pp. 980–985. IEEE.
10. Borkar A, Hayes M and Smith M. A novel lane detection system with efficient ground truth generation. *IEEE Trans Intell Transp Syst* 2012; 13: 365–374.
11. Li WH, Gong XH, Wang Y, et al. A lane marking detection and tracking algorithm based on sub-regions. In: *Proceedings of the 2014 international conference on informative and cybernetics for computational social systems*, Qingdao, 9–10 October 2014, pp. 68–73. IEEE.
12. Son JI, Yoo HJ, Kim SH, et al. Real-time illumination invariant lane detection for lane departure warning system. *Expert Syst Appl* 2015; 24: 1816–1824.
13. National Instruments. *IMAQ vision concepts manual*. National Instruments, October 2000.
14. Towards Data Science. Applied deep learning – Part 4: convolutional neural networks, https://towardsdatascience.com/applied-deep-learning-part-4-convolutional-neural-networks-584bc134c1e2
15. Ozguner U, Redmill K and Acarman T. *Autonomous ground vehicles*. Norwood: Artech House, 2011.
16. Dai YY, Choi KS and Lee SG. Adaptive formation control and collision avoidance using a priority strategy for nonholonomic mobile robots. *Int J Adv Robotic Syst* 2013; 10: 1–14.
17. French M. *Advanced path planning for an autonomous SAE electric race car*. Project report, 2014. Western Australia University.
18. Werling M, Ziegler J, Kammel S, et al. Optimal trajectory generation for dynamic street scenarios in a Frenet Frame. In: *Proceedings of the IEEE international conference on robotics and automation*, Anchorage, AK, 3–7 May 2010, pp. 987–993. IEEE.
19. Ivan T, Iryna V, Roman T, et al. Designing and simulation a motion control system of mobile robot based on fuzzy logic. *Int J Adv Res Comput Eng Technol* 2014; 3: 3143–3147.
20. Iryna V and Roman T. Grantry and bridge cranes neuro-fuzzy control by using neural-like structures of geometric transformations. *Technical Transactions, Automatic Control* 2013; 3-AC: 53–68.
21. Towards Data Science. Lane detection with deep learning (Part 2), https://towardsdatascience.com/lane-detection-with-deep-learning-part-2-3ba559b5c5af
22. Takahashi A, Honggo T, Ninomiya Y, et al. Local path planning and motion control for agv in positioning. In: *Proceedings of the IEEE international workshop on intelligent robots and systems*, Tsukuba, 4–6 September 1989, pp. 392–397. IEEE.