Hindawi Publishing Corporation
Mathematical Problems in Engineering
Volume 2010, Article ID 578406, 20 pages
doi:10.1155/2010/578406

Research Article

Design and Lyapunov Stability Analysis of a Fuzzy Logic Controller for Autonomous Road Following

Yi Fu, Howard Li, and Mary Kaye

Department of Electrical and Computer Engineering, University of New Brunswick, Fredericton, NB, Canada E3B 5A3

Correspondence should be addressed to Howard Li, howard@unb.ca

Received 14 September 2009; Accepted 27 January 2010

Academic Editor: Irina N. Trendafilova

Autonomous road following is one of the major goals in intelligent vehicle applications. The development of an autonomous road following embedded system for intelligent vehicles is the focus of this paper. A fuzzy logic controller (FLC) is designed for vision-based autonomous road following. The stability analysis of this control system is addressed. Lyapunov's direct method is utilized to formulate a class of control laws that guarantee the convergence of the steering error. Certain requirements for the control laws are presented for designers to choose a suitable rule base for the fuzzy controller in order to make the system stable. Stability of the proposed fuzzy controller is guaranteed theoretically and also demonstrated by simulation studies and experiments. Simulations using the model of the four degree of freedom nonholonomic robotic vehicle are conducted to investigate the performance of the fuzzy controller. The proposed fuzzy controller can achieve the desired steering angle and make the robotic vehicle follow the road successfully. Experiments show that the developed intelligent vehicle is able to follow a mocked road autonomously.

1. Introduction

Intelligent transportation systems (ITSs) are an emerging global phenomenon benefiting public and private sectors alike. ITSs and intelligent vehicles can relieve congestion, improve safety, and enhance productivity. The field of intelligent vehicles is rapidly growing worldwide. Technologies involved with intelligent vehicles include sensing and control technologies, communications, and computer informatics. Intelligent vehicles offer the potential to significantly enhance safety and operational efficiency. As one component of ITSs, intelligent vehicles use sensing and intelligent algorithms to understand the vehicle's immediate environment, either assisting the driver or fully controlling the vehicle [1]. Intelligent vehicles function at the control layer to enable the driver vehicle subsystem to operate more effectively. Intelligent vehicles can help drivers in various ways: (1) collision warning—to advise or warn the driver; (2) driver assistance—to partially control the vehicle as an emergency intervention to avoid a collision; (3) automated driving—to fully control the vehicle. Vehicles that can navigate autonomously in everyday traffic will become a reality in a few decades. In this paper, we will focus on the development of embedded systems to realize automated road following for intelligent vehicles.

A lot of highway accidents are caused by deviation from the road. Autonomous road following is one of the major research topics in the area of intelligent vehicles. Because of its potential in preventing accidents caused by human fatigue and performing tasks in environments unreachable by human beings, a lot of road following methods for intelligent vehicles have been proposed in the recent decades. Intelligent vehicles are robotic systems that perceive the driving environment to assist the driver in safe vehicle operation by providing driving assistance or full control of the vehicle. With the proposed research, we will be able to develop an autonomous control system that can be used on cars to assist driving or take the control of driving in order to avoid fatal accidents. Furthermore, the developed intelligent vehicles can be used to perform tasks in an environment that is unreachable by human beings.

Generally speaking, autonomous road following requires two major steps: road feature extraction, and speed/steering control of the vehicles. The success of autonomous road following requires the ability to continuously detect and extract useful road features, to analyze road features, and to perform steering and speed control based on road conditions.

Computer-vision-based road feature extraction has been applied in intelligent vehicles for road following widely because of its advantages in low-power consumption, compact size, availability as a commercial product, cost effectiveness, and robustness [2]. Numerous image processing and feature extraction methods can be found in [3–7].

Control algorithms should be considered as an important issue in road following to ensure safe and smooth rides. Although a lot of researches have been done on this topic, most of them are based on traditional control theories such as PID [2] and nonlinear controllers [8]. The kinematic behavior of autonomous road following is typically nonlinear. Therefore linear models usually fail to describe these systems efficiently. However, it is difficult to analyze nonlinear mathematical models for autonomous road following schemes. Other methods such as neural networks [9, 10] and reinforcement learning (RL) [11] approaches have also been used in road following, but these approaches require learning procedures which consume extra computation time.

Human drivers can drive a car smoothly with their driving expertise rather than knowledge on control theories. This fact leads us to the fuzzy logic solution. Fuzzy logic control is known to be an organized method to emulate human expertise in dealing with imprecise data. It attempts to apply a human-like way of thinking in the application areas and allows intermediate values to be defined with linguistic terms besides conventional evaluations. It has been proven to be an effective and active method in solving control problems during the past decades. Fuzzy logic is a logic much closer to human thinking than traditional logic. It is a precise logic of imprecision and approximate reasoning. Fuzzy logic controllers provide a means of converting a linguistic control strategy based on expert knowledge into an automatic control strategy [12]. In areas where conventional control methods have difficulties because of the lack of precise quantitative data regarding the system inputs and outputs, fuzzy logic controllers can process imprecise data and make rational decisions by emulating human thinking and decision making capabilities. Fuzzy logic control has been applied for intelligent vehicles in many areas such as car parking and vehicle guidance. Various applications of fuzzy logic control in the field of intelligent vehicles can be found in [13–17].

Safety and reliability are of great importance for intelligent vehicles especially if the vehicles are used for transportation. Therefore, before applying any dynamical systems to intelligent vehicles, the stability of these dynamical systems should be studied to make sure they are stable. Several well-known methods could be used to analyze the stability of a system, such as the Routh-Hurwitz stability criterion, the Lyapunov stability criterion, the Nyquist criterion, finding poles directly, the root locus method, or the Jury stability test for discrete-time control systems. Lyapunov stability theory is probably the most used tool for stability analysis. A dynamic system is Lyapunov stable if all solutions of the system which start near an equilibrium point remain near the point for all time. However, it is difficult to find the suitable Lyapunov functions. There is no general method to construct a Lyapunov function. Trial-and-error or mathematical/physical methods are often used. Some useful approaches for finding Lyapunov functions are Sum of Squares Decomposition [18, 19], convex search for storage functions [20], and variable gradient method [21]. However, these methods do not always lead to a desired Lyapunov function.

In this paper, we will design an autonomous road following embedded system for multiple intelligent vehicles. An intelligent vehicle which is capable of moving between two lines on the road will be designed. A fuzzy logic controller (FLC) will be developed to control the steering wheel of the vehicle for autonomous road following. An FLC on a nonlinear system requires less computational power compared to traditional nonlinear system applications. The resources required for building the embedded system will be significantly reduced. Lyapunov's direct method will be used to analyze the stability of the control system. The "variable gradient" approach proposed in [21] will be used to construct the Lyapunov function of this system. This control system will be implemented on a vision-based intelligent vehicle which is able to perform road following autonomously.

The rest of the paper is organized as follows. The kinematic model of the vehicle is studied in Section 2. Section 3 presents the design of the FLC. Section 4 provides the proof of the stability using Lyapunov's direct method. Experimental setup for the vision-based intelligent vehicle is described in Section 5. Section 6 discusses the simulation and experimental results. The last section of this paper concludes the research.

2. Kinematic Model of the Car

The road following problem of the intelligent vehicle with kinematic constraints in the two-dimensional workspace is studied. A nonholonomic constraint for a robotic vehicle is a nonintegrable equation involving the configuration parameters and their derivatives (velocity parameters) [22]. Such a constraint does not reduce the dimension of the space of configurations attainable by the robot but reduces the dimension of the space of possible motions at any given configuration. Considering the robotic vehicle modeled as Figure 1, the rear wheels are aligned with the vehicle while the front wheels are allowed to pivot about the axes. In a sufficiently large empty space, a robotic vehicle can be driven to any position with any orientation, hence the robot's configuration space has four dimensions, two for translation, one for rotation and one for the steering angle. Let (x, y, ϕ, θ) denote the

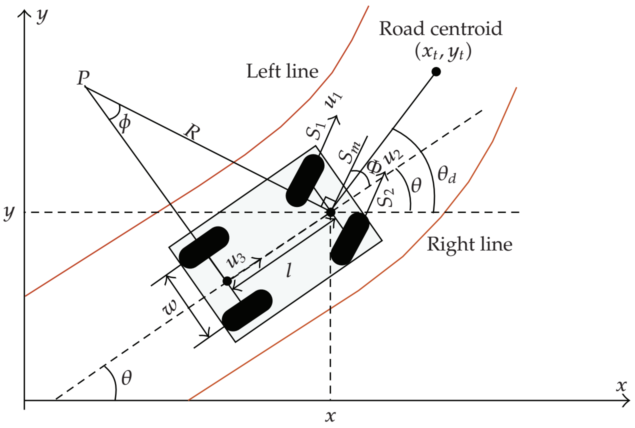


Figure 1: Kinematic model of the car.

configuration of the robot, parameterized by the location of the front wheels. The kinematic model of the robotic vehicle can be represented as

$$\begin{aligned} \dot{x} &= u_3 \cos\theta, \\ \dot{y} &= u_3 \sin\theta, \\ \dot{\theta} &= \frac{u_3}{l} \tan\phi, \end{aligned} \tag{2.1}$$

where u_3 corresponds to the forward velocity of the vehicle and the angle of the vehicle body with respect to the horizontal line is θ, the steering angle with respect to the vehicle body is ϕ, $(x,\ y)$ is the location of the center point of the front wheels, l is the length between the front and the rear wheels.

Since the turn radius of the robot is quite large compared with the radius of the wheels, referring to Figure 1, we have the following relations:

$$\begin{aligned} S_1 &= \left(R - \frac{w}{2}\cos\phi\right)\phi, \\ S_m &= R\phi, \\ S_2 &= \left(R + \frac{w}{2}\cos\phi\right)\phi, \end{aligned} \tag{2.2}$$

where S_1 and S_2 give the displacement (distance traveled) of the front left and front right wheel, respectively, R is the turn radius of the center point of the front wheels, w is the distance between wheels (from center-to-center along the length between the two font wheels or two back wheels), and ϕ is the angle of the turn in radians. S_m is the displacement at the center point of the front wheels. Once we have established the simple geometry for the robotic vehicle system, it is easy to develop algorithms for controlling the robot's steering angle ϕ, thus controlling the robot's orientation θ. As the robot is considered as having a rigid body, to develop a forward kinematic equation for the steering system, we start by specifying a frame of reference in which an arbitrarily chosen point is treated as stationary. All other points in the system are treated as moving relative to the reference point. Here we treat the center point of the front wheels as the origin of the simulated robot's frame of reference.

For the robotic vehicle that we are modeling, the turn angle of the front wheels has a range of $[-\pi/6,\ \pi/6]$, that is, $-\pi/6 \leq \phi \leq \pi/6$. By adjusting the steering angle ϕ, we can control the velocity of both the front wheels

$$\begin{aligned} \dot{S}_1 &= \dot{\phi}\left(R - \frac{w}{2}\cos\phi\right) + \frac{w}{2}\phi\dot{\phi}\sin\phi, \\ \dot{S}_2 &= \dot{\phi}\left(R + \frac{w}{2}\cos\phi\right) - \frac{w}{2}\phi\dot{\phi}\sin\phi, \end{aligned} \tag{2.3}$$

so that

$$u_2 - u_1 = \dot{\phi}w(\cos\phi - \phi\sin\phi), \tag{2.4}$$

where u_1 and u_2 correspond to the forward velocity of the front left wheel and the front right wheel, respectively.

The real-time path of the vehicle can be obtained by integrating (2.1). The angle of the vehicle body with respect to the horizontal line at time $(t + \Delta t)$ could be derived first. If $\dot{\phi} \neq 0$, that is, $u_2 \neq u_1$, the angle of the vehicle body with respect to the horizontal line at time $(t + \Delta t)$ is given as

$$\begin{aligned} \theta(t + \Delta t) &= \theta(t) + \frac{u_3}{l}\int_t^{t+\Delta t} \tan\phi dt \\ &= \theta(t) + \frac{u_3}{l\dot{\phi}}\left(-\ln|\cos\phi|\right)\Big|_t^{t+\Delta t} \\ &= \theta(t) + \frac{u_3}{l\dot{\phi}}\left(-\ln|\cos\phi(t + \Delta t)| + \ln|\cos\phi(t)|\right), \end{aligned} \tag{2.5}$$

and if $\dot{\phi} = 0$, that is, $u_2 = u_1$, the angle of the vehicle body at time $(t + \Delta t)$ is given as

$$\theta(t + \Delta t) = \theta(t) + \frac{u_3}{l}\tan\phi\Delta t. \tag{2.6}$$

Next the position of the vehicle at time $(t + \Delta t)$ could be derived. If $\dot{\theta} \neq 0$, that is, $\phi \neq 0$, the position of the moving vehicle at time $(t + \Delta t)$ is given as

$$\begin{aligned} x(t + \Delta t) &= x(t) + \frac{u_3}{\dot{\theta}}(\sin\theta(t + \Delta t) - \sin\theta(t)), \\ y(t + \Delta t) &= y(t) - \frac{u_3}{\dot{\theta}}(\cos\theta(t + \Delta t) - \cos\theta(t)), \end{aligned} \tag{2.7}$$

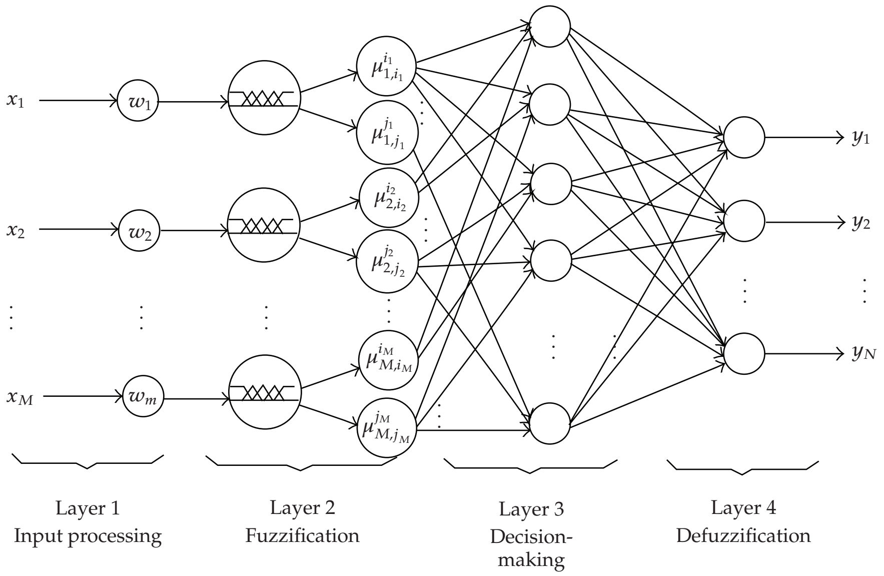


Figure 2: Structure of the FLC.

and if $\dot{\theta} = 0$, that is, $\phi = 0$, the position of the vehicle at time $(t + \Delta t)$ is given as

$$\begin{aligned} x(t+\Delta t) &= x(t) + u_3 \cos\theta\Delta t, \\ y(t+\Delta t) &= y(t) + u_3 \sin\theta\Delta t. \end{aligned} \tag{2.8}$$

3. Design of the FLC

In this section, the general structure of the proposed FLC is presented first. Then based on simulation and experiments, the FLC is designed to perform the road following task. The developed FLC not only enhances performance for road following but also saves computational resources.

3.1. The Structure of the FLC

Figure 2 shows the structure of the proposed FLC which has four layers: input processing layer, fuzzification layer, decision-making layer, and defuzzification layer. These layers form an FLC following the Mamdani model. There are M inputs and N outputs for the FLC. Each rule in the system has the following form:

> Rule k: IF x_1 is $S_1^{(k)}$ AND $\ldots$ AND x_m is $S_m^{(k)}$ $\ldots$ AND x_M is $S_M^{(k)}$, THEN y_1 is $R_1^{(k)}$ AND $\ldots$ y_n is $R_n^{(k)}$ $\ldots$ AND y_N is $R_N^{(k)}$ $\quad (k = 1, 2, \ldots, K)$,

where x_m is the mth input, $m = 1, 2, \ldots, M$, y_n is the nth output, $n = 1, 2, \ldots, N$, $S_m^{(k)}$ is the fuzzy linguistic sets for input x_m, $R_n^{(k)}$ is the fuzzy rule sets for output y_m, and K is the number of rules in the rule base. The design of each layer is presented as follows.

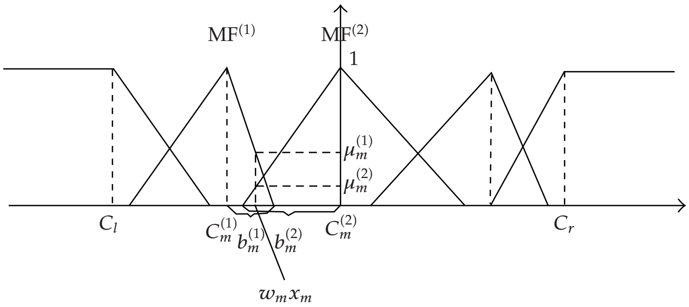


Figure 3: Input MFs.

Input Processing Layer

The inputs of the FCL are crisp values. The input vector can be presented as

$$[x_1 \ \ x_2 \ \ \cdots \ \ x_M]. \tag{3.1}$$

Each input is adjusted by a weight factor. Then the input vector can be presented as

$$[w_1x_1 \ \ w_2x_2 \ \ \cdots \ \ w_Mx_M], \tag{3.2}$$

where $w_1, w_2, \ldots, w_m$ are weight factors.

Fuzzification Layer

It converts the weighted crisp inputs to fuzzy variables. This layer has three nodes for each input: one node defines the fuzzy membership functions (MFs) for the input; the other two nodes represent the two triggered MFs for the input. Triangle MFs are used for fuzzy sets except for the leftmost and the rightmost fuzzy set. The leftmost and rightmost MFs are trapezoidal. Without loss of generosity, MFs for different fuzzy sets can have different triangle or trapezoidal shapes. We constrain the maximum overlapping degree of two MFs to 50%. Therefore the maximum number of triggered MFs is 2 and these two MFs must be adjacent. An example of the MFs with five fuzzy sets is shown in Figure 3. We define MF$^{(1)}$ and MF$^{(2)}$ as the first and second triggered MF, and $\mu_m^{(1)}$ and $\mu_m^{(2)}$ as the membership value of the first and second triggered MF of the input x_m, respectively, then $\mu_m^{(1)}$ and $\mu_m^{(2)}$ can be calculated as

$$\begin{gathered}
\mu_m^{(1)} = 1, \quad \mu_m^{(2)} = 0, \quad x_m \le C_l, \\
\mu_m^{(1)} = 0, \quad \mu_m^{(2)} = 1, \quad x_m \ge C_r, \\
\mu_m^{(1)} = \max\left(0, 1 - \frac{w_m x_m - C_m^{(1)}}{b_m^{(1)}}\right), \\
\mu_m^{(2)} = \max\left(0, 1 - \frac{C_m^{(2)} - w_m x_m}{b_m^{(2)}}\right), \quad C_m^{(1)} \le x_m \le C_m^{(2)},
\end{gathered} \tag{3.3}$$

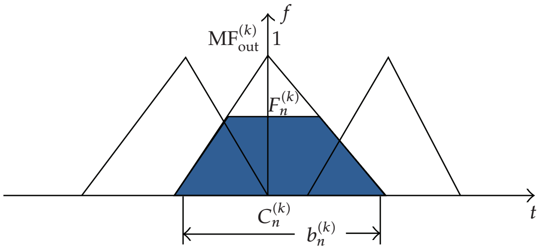


Figure 4: An example of output MFs.

where $C_m^{(1)}$ and $C_m^{(2)}$ are the centers of the two triggered MFs and $C_m^{(1)} < C_m^{(2)}$, C_l and C_r are the leftmost and the rightmost limits, respectively, $b_m^{(1)}$ is the width of the right-half base of $\mathrm{MF}^{(1)}$, and $b_m^{(2)}$ is the width of the left half base of $\mathrm{MF}^{(2)}$. If one of the membership values is zero, the MF with respect to this membership value is not triggered, then there is only one triggered MF. The outputs of this layer are membership values of triggered MFs for all inputs.

Decision-Making Layer

For every output, each node in this layer selects a fuzzy control rule from the rule base of this output and obtains the antecedent value of this rule based on triggered input MFs. Each output has one rule base, which means in the case of N outputs there are N different rule bases in total. For each output, rules are selected according to the predefined rule base corresponding to this output. Thus, each node in this layer selects N rules from N different rule bases. Each control rule is selected based on a combination of MFs. Each of these MFs is one of the triggered MFs of an input. For example, if there are three inputs, two of which have two triggered MFs while the other input has only one triggered MF, the number of selected rules for each output is $4\ (2 \times 2 \times 1)$. The antecedent value of the selected rule k is computed using AND fuzzy logic. Therefore, "min" operation is considered for composition of the FLC:

$$F^{(k)} = \min(\mu_1, \mu_2, \ldots, \mu_m, \ldots, \mu_M), \tag{3.4}$$

where μ_m is one of the triggered input MFs (either $\mu_m^{(1)}$ or $\mu_m^{(2)}$) of $x_m,\ k = 1, 2, \ldots, K^*$, and K^* is the number of selected rules.

Defuzzification Layer

This layer converts fuzzy values to crisp values and send them out as control outputs of the FLC. Each node in this layer performs defuzzification for an output. Triangles are used for the output MFs. Figure 4 shows an example of MFs for an output with three fuzzy sets. The center of gravity (COG) algorithm is used to compute output crisp values. Let $C_n^{(k)}$ be the center of the kth triangle. The output y_n can be computed as

$$y_n = \frac{\sum_k C_n^{(k)} \int F_n^{(k)} dt}{\sum_k \int F_n^{(k)} dt}, \tag{3.5}$$

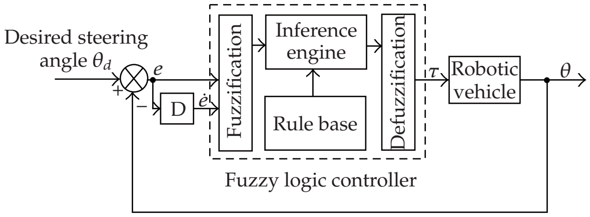


Figure 5: Block diagram of the specified FLC.

where $F_n^{(k)}$ is the antecedent value of the kth selected fuzzy rule for y_n. As all the output MFs are triangles, $\int F_n^{(k)} dt$ can be obtained by computing the area of the trapezoid under the line $f = F_n^{(k)}$. Thus, (3.5) can be modified as

$$y_n = \frac{\sum_k C_n^{(k)} b_n^{(k)} \left(F_n^{(k)} - \left(F_n^{(k)} \right)^2 / 2 \right)}{\sum_k b_n^{(k)} \left(F_n^{(k)} - \left(F_n^{(k)} \right)^2 / 2 \right)} \tag{3.6}$$

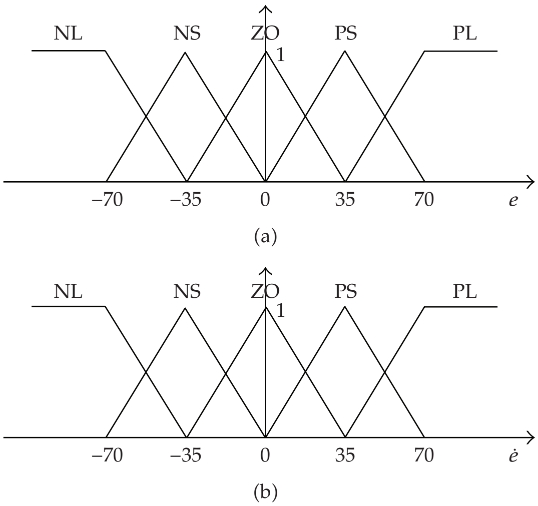
where $b_n^{(k)}$ is the base length of the kth output MF. Equation (3.6) simplifies the defuzzification calculation by replacing derivatives with the computation of trapezoid areas.

3.2. The Developed FLC for Road following

A block diagram of the specified FLC for the road following task is shown in Figure 5. The desired orientation of the center line of the car should be aligned with the road centroid. The error is the angle between the desired orientation of the center line and the actual center line of the car. The error is represented by $e = \theta_d - \theta$. To reduce the error to zero, the steering angle should be equal to ϕ. ϕ is determined by the FLC. The error and the change-in-error are calculated and fed into the FLC. The FLC is designed to output control signals corresponding to the control torque to the front steering motor to control the front wheels' steering angle ϕ.

The fuzzification procedure maps the crisp input values to the linguistic fuzzy terms with the membership values between 0 and 1. In this study, we use five MFs for both error $e = \theta_d - \theta$ and change-in-error $\dot{e} = \dot{\theta}_d - \dot{\theta}$ because five MFs can balance between the performance and the complexity of the FLC. The isosceles triangle membership function (MF) is used except for the leftmost and the rightmost input MF which are trapezoidal. Each MF has a 50% overlapping with its neighboring MFs. Figure 6 illustrates the input MFs for e and $\dot{e}$, respectively. Note that the scale for the X coordinate can be changed easily in the program.

The inference engine is responsible for decision making in the control system using approximate reasoning. The fuzzy control rules are designed based on expert knowledge and testing. Furthermore, the control rules also meet the stability requirements derived from Lyapunov's direct method. Table 1 represents abstract knowledge that an expert uses to control the steering angle given information about the error and its derivative. For example, if e is positive large "PL" and is increasing rapidly ($\dot{e}$ is positive large), then the vehicle should turn left sharply, that is, ϕ should be left large (LL). The rule base stores rules which define



(a)

(b)

Figure 6: Input Membership functions. (a) Error MF. (b) Change-in-error MF.

Table 1: Fuzzy rule base.

				$\dot{e}_\phi$		
		PL	PS	ZO	NS	NL
	PL	RL	RL	RL	RS	MD
	PS	RL	RL	RS	MD	LS
e_ϕ	ZO	RL	RS	MD	LS	LL
	NS	RS	MD	LS	LL	LL
	NL	MD	LS	LL	LL	LL

Note: PL: Positive Large; PS: Positive Small; ZO: Zero; NS: Negative Small; NL: Negative Large; RL: Right Large (turn right sharply); RS: Right Small (turn right gently); MD: Middle (no turn); LL: Left Large (turn left sharply); LS: Left Small (turn left gently).

the relation between the inputs and the output. Based on the membership functions of the error and the change in error, twenty-five fuzzy rules are obtained.

Figure 7 shows output MFs. All MFs for the output are isosceles triangles with 50% overlap between two neighboring MFs.

The defuzzification procedure maps the fuzzy output from the inference mechanism to a crisp signal. We use the COG defuzzification method to combine the recommendations represented by the implied fuzzy sets from all the rules. The crisp value is computed according to (3.6).

4. Stability Analysis Using Lyapunov's Direct Method

The autonomous road following system can be represented as

$$\begin{aligned} e &= \theta - \theta_d = x_1, \\ \dot{e} &= \dot{\theta} - \dot{\theta}_d = x_2, \end{aligned} \tag{4.1}$$

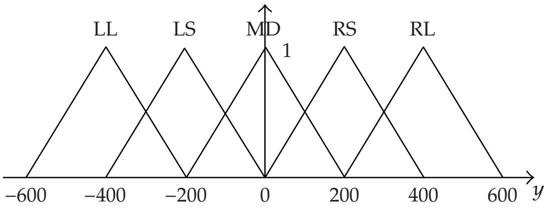


Figure 7: Steering control output MF.

where θ_d is the desired angle, and θ is the actual angle. From (2.1), we know that we need to design the FLC to control the steering wheel so that θ can approach θ_d. The output of the fuzzy controller $\Phi(\cdot)$ is a function of x_1 and x_2 that can be represented by $\Phi(x_1,\ x_2)$. On the intelligent vehicle, the control signal $\Phi(x_1,\ x_2)$ is used to steer the front wheels. We assume that the low-level tracking controller can output the motor torque to steer the front wheels accordingly. Recall (2.1), we have

$$\begin{gathered}\dot{x}_1 = x_2 = \dot{\theta} - \dot{\theta}_d = \frac{u_3}{l}\tan\phi - \dot{\theta}_d, \\ \dot{x}_2 = \ddot{\theta} - \ddot{\theta}_d = \frac{u_3}{l}\frac{1}{\cos^2\phi}\dot{\phi} - \ddot{\theta}_d.\end{gathered} \tag{4.2}$$

Assume that at the equilibrium point $e = 0$ and $\dot{e} = 0$, so that the equilibrium is preserved. The main shortcoming of Lyapunov theory is the difficulty associated with the construction of Lyapunov functions [21]. To overcome this problem, we use the "variable gradient" approach proposed in [21] to construct the Lyapunov function. The essence of this method is to assume the gradient of the unknown Lyapunov function $V(\cdot)$ is known up to some adjustable parameters. Then finding $V(\cdot)$ becomes integrating the assumed gradient.

The system can be represented as

$$\dot{x} = f(x) = \begin{bmatrix}\dot{x}_1 & \dot{x}_2\end{bmatrix}^T = \begin{bmatrix}x_2 & \dot{x}_2\end{bmatrix}^T. \tag{4.3}$$

We proceed to find a suitable Lyapunov function for this system as follows.

Step 1. Assume that $\nabla V(x) = g(x)$ has the form

$$g(x) = \begin{bmatrix}h_1^1 x_1 + h_1^2 x_2 & h_2^1 x_1 + h_2^2 x_2\end{bmatrix}. \tag{4.4}$$

Step 2. Impose the symmetry conditions

$$\frac{\partial^2 V}{\partial x_i x_j} = \frac{\partial^2 V}{\partial x_j x_i}, \tag{4.5}$$

or equivalently,

$$\frac{\partial g_i}{\partial x_j} = \frac{\partial g_j}{\partial x_i}. \tag{4.6}$$

In our case, the symmetry condition is

$$\begin{aligned}\frac{\partial g_1}{\partial x_2} &= x_1\frac{\partial h_1^1}{\partial x_2} + h_1^2 + x_2\frac{\partial h_1^2}{\partial x_2},\\ \frac{\partial g_2}{\partial x_1} &= h_2^1 + x_1\frac{\partial h_2^1}{\partial x_1} + x_2\frac{\partial h_2^2}{\partial x_1}.\end{aligned} \tag{4.7}$$

To simplify the solution, we assume that g_i's are constant. Then we have

$$\begin{aligned}&\frac{\partial h_1^1}{\partial x_2} = \frac{\partial h_1^2}{\partial x_2} = \frac{\partial h_2^1}{\partial x_1} = \frac{\partial h_2^2}{\partial x_1} = 0,\\ &\frac{\partial g_1}{\partial x_2} = \frac{\partial g_2}{\partial x_1} \Longleftrightarrow h_1^2 = h_2^1 = k\\ &\Longrightarrow g(x) = \begin{bmatrix} h_1^1x_1 + kx_2 & kx_1 + h_2^2x_2\end{bmatrix}.\end{aligned} \tag{4.8}$$

If we choose $k = 0$, we have

$$g(x) = \begin{bmatrix} g_1 & g_2\end{bmatrix} = \begin{bmatrix} h_1^1x_1 & h_2^2x_2\end{bmatrix}. \tag{4.9}$$

Step 3. Find $\dot{V}$:

$$\begin{aligned}\dot{V}(x) &= \nabla V \cdot f(x)\\ &= g(x)\cdot f(x)\\ &= \begin{bmatrix} h_1^1x_1 & h_2^2x_2\end{bmatrix}\begin{bmatrix} x_2\\ \dot{x}_2\end{bmatrix}\\ &= h_1^1x_1x_2 + h_2^2x_2\dot{x}_2.\end{aligned} \tag{4.10}$$

Step 4. Find V from ∇V by integration:

$$\begin{aligned}V(x) &= \int_0^{x_1} g_1(s_1,0)ds_1 + \int_0^{x_2} g_2(x_1,s_2)ds_2\\ &= \int_0^{x_1} h_1^1s_1ds_1 + \int_0^{x_2} h_2^2s_2ds_2\\ &= \frac{1}{2}h_1^1x_1^2 + \frac{1}{2}h_2^2x_2^2.\end{aligned} \tag{4.11}$$

Therefore, we have

$$V(x) = \frac{1}{2}h_1^1 x_1^2 + \frac{1}{2}h_2^2 x_2^2, \tag{4.12}$$

$$\dot{V}(x) = h_1^1 x_1 x_2 + h_2^2 x_2 \dot{x}_2. \tag{4.13}$$

Step 5. Find h_1^1 and h_2^2 that make $V > 0$. From (4.12), $V(x) > 0$ only if $h_1^1,\ h_2^2 > 0$. Let $h_1^1 = h_2^2 = 1$, then we have

$$\dot{V}(x) = x_1 x_2 + x_2 \dot{x}_2 = \left(\dot{\theta} - \dot{\theta}_d\right)\left(\theta + \ddot{\theta} - \theta_d - \ddot{\theta}_d\right). \tag{4.14}$$

Based on the "variable gradient" approach, we can derive the following Lyapunov function:

$$V(x) = \frac{1}{2}\ x_1^2 + \frac{1}{2}x_2^2, \tag{4.15}$$

where $V : B(\epsilon) \rightarrow \Re$ for some $\epsilon > 0$, where $B(\epsilon) = \{x \in \Re^2 :\ |x| < \epsilon\}$ is a ball centered at the origin with a radius of ϵ and $|\cdot|$ is a norm on $\Re^2$. Then the gradient of the Lyapunov function is

$$\nabla V(x(t)) = [\ x_1 \quad x_2], \tag{4.16}$$

$$\begin{aligned} \dot{V} &= [\ x_1 \ \ x_2]\begin{bmatrix} x_2 \\ \dot{x}_2 \end{bmatrix} \\ &= x_2(x_1 + \dot{x}_2) \\ &= \left(\dot{\theta} - \dot{\theta}_d\right)\left(\theta + \ddot{\theta} - \theta_d - \ddot{\theta}_d\right) \\ &= \left(\frac{u_3}{l}\tan\phi - \dot{\theta}_d\right)\left(\theta - \theta_d + \frac{u_3}{l}\frac{1}{\cos^2\phi}\dot{\phi} - \ddot{\theta}_d\right). \end{aligned} \tag{4.17}$$

We would like $\dot{V} \leq 0$, to prove stability, that is, to show that the fuzzy controller can achieve and maintain the desired θ_d. In (4.17), $u_3/l > 0$, $\cos^2\phi > 0$. We assume that $\dot{\theta}_d$ is bounded, and $\ddot{\theta}_d = 0$. On the RC car, the steering angle $-30° \leq \phi \leq 30°$.

We can always design the FLC, $\Phi(x_1, x_2)$, to adjust ϕ and $\dot{\phi}$ to achieve the following conditions:

$$\begin{aligned} &\frac{u_3}{l}\frac{1}{\cos^2\phi}\dot{\phi} \geq \theta_d - \theta \quad \text{if } \frac{u_3}{l}\tan\phi - \dot{\theta}_d > 0 \\ &\frac{u_3}{l}\frac{1}{\cos^2\phi}\dot{\phi} \leq \theta_d - \theta \quad \text{if } \frac{u_3}{l}\tan\phi - \dot{\theta}_d < 0. \end{aligned} \tag{4.18}$$

Therefore, $\dot{V} \leq 0$ can always be achieved and hence the stability holds.

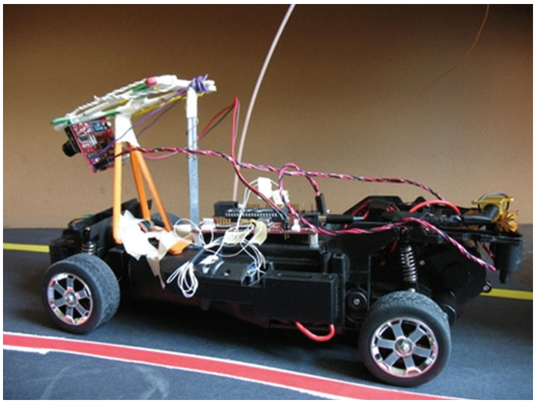


Figure 8: The steering mechanism of the RC car.

5. Experimental Setup

In this section, the hardware setup of the prototype vehicle for autonomous road following is presented. A fully automated robotic system is designed to control a motor vehicle using an on board camera. The hardware system includes a camera associated with a frame grabber. The camera is then connected to a Microchip PIC controller which controls the vehicle motion. This system utilizes the camera with a frame-grabber and a built-in processor to grab pictures of the road and to find the center of the road. Then it calculates the deviation between the center of the car and the center of the road, and sends the result to an FLC. The FLC decides how to turn the vehicle and sends out control signals to control the steering wheel. This is implemented in a way that the vehicle can remain between two lines while simulating highway motion.

5.1. The Remote Control Car

The Audi S4 radio control (RC) car platform is chosen because it offers a fully functional steering and drive system. The control signal for the steering and drive system is generated by the microcontroller which emulates the car′s control signals. The speed and steering of the car are controlled by varying the pulse-width modulation (PWM) signals. The steering mechanism of the RC car is shown in Figure 8.

5.2. The Mocked Road

The lane used in this project is the space between two lines as shown in Figure 8. To illustrate the control process, we use a red line and a yellow line. The distance between these two lines is constant.

5.3. Computer Vision

The CMUCam3 camera is chosen for the project. The camera uses a built in microcontroller and a frame grabber to allow postprocessing on each captured image. As the two lines which

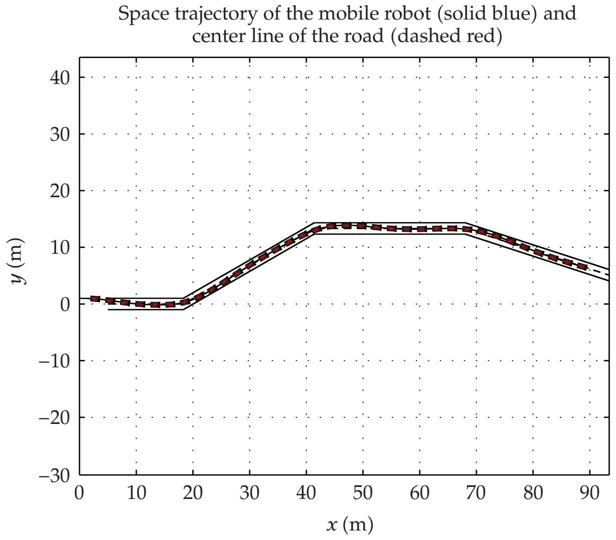
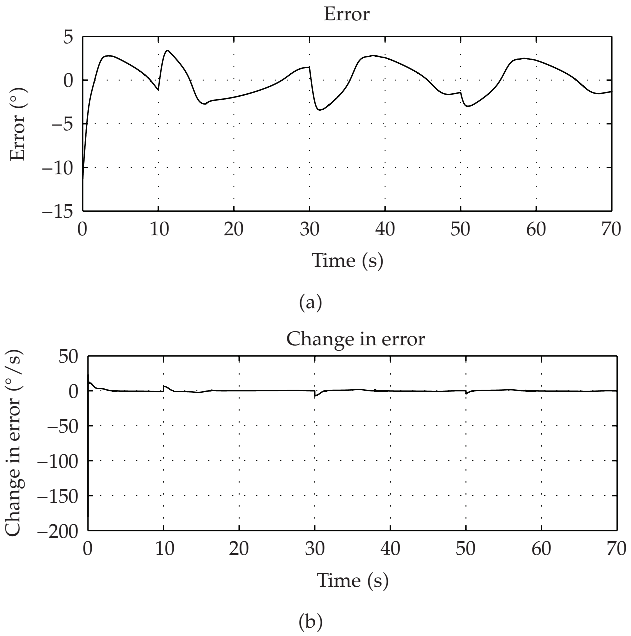


Figure 9: Following a road in Cartesian space.



(a)

(b)

Figure 10: Error e_{ϕ} and change-in-error $\dot{e}_{\phi}$.

mark the road are in different colors from the rest of the road, using a multiple color tracking algorithm, we are able to track each line individually. The minimum and maximum RGB values for each line color are set in the camera code. Pixels within the RGB value range are regarded as a part of the line. A filter is applied to filter out noise. The camera finds the location and the width of each colored line in terms of pixels and calculates the centroid of each colored line. The centroid of two colored lines is then used to calculate the position

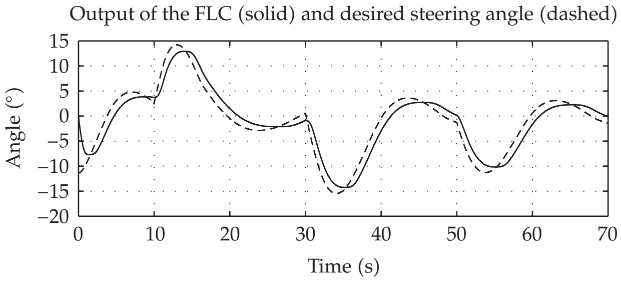


Figure 11: Output of the fuzzy controller.

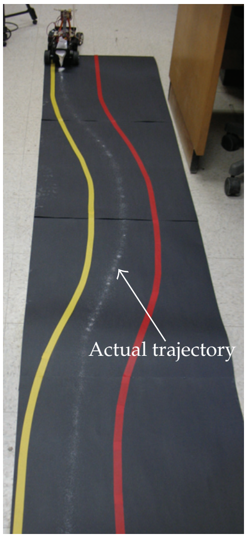


Figure 12: The road following result using the proposed method.

error which is sent to the microcontroller. Since the vehicle is supposed to remain in the center of the two colored lines, error can be calculated if the centroid is not at the center of the image. The camera is programmed using a variation of the C language. The error is calculated according to one of the states described in Table 2. Centroids in the table refer to the x coordinate of the image only. In the table, Cr represents the centroid of the red line, Cy represents the centroid of the yellow line, Cw represents the centroid of the image, C represents the centroid of the two lines, respectively.

5.4. Microcontroller

The PIC18F4431 microcontroller is used to interface with the camera and the car's existing circuitry. It receives the incoming position error from the RS-232 port located on the CMUCam3 board. The microcontroller then uses the error to calculate the appropriate steering signal, using the proposed FLC written in C language. The steering signal is then output to the RC car's circuit.

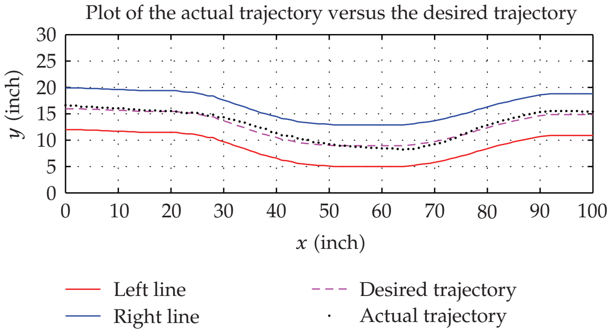


Figure 13: The actual trajectory and the desired trajectory.

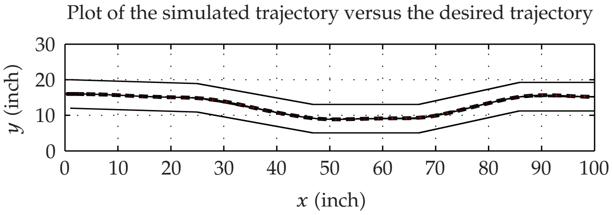


Figure 14: The simulated trajectory and the desired trajectory.

6. Simulation Studies and Experiment Results

In order to verify the performance of the proposed FLC and the system, simulations and experiments have been conducted.

Simulations have been done to test the performance of the fuzzy controller. The results show that the intelligent vehicle is able to follow the road with a speed of 13 m/s. Figure 9 shows the tracking performance of the robotic vehicle in the Cartesian space. The road has two sharp curves and the robot starts from (0, 1) which is off the center of the road. The robot is capable of following the road. It can respond to sudden changes on the road by changing its steering angle accordingly.

Figure 10 shows the steering error and the change-in-error. Figure 11 shows the output of the fuzzy controller. It can be seen that the fuzzy controller can output smooth control signals to sudden changes in e and $\dot{e}$ to make the system stable.

Experiments have been performed to verify the fuzzy logic controller design. In our experiments, we use a mocked road with two lines to guide the navigation of the robot. The actual trajectory of the robot is marked on the road using white powder dropping from a container attached to the rear end of the robot. Figure 12 shows the robot performing the autonomous road following task on the mocked road with curves. The robot moves at a speed of 0.9 m/s. In the figure, the line in the center of the road is the trajectory marked by the powder.

In our experiments, all data, such as errors and mean square errors, are calculated based on four trials. Sample points on the actual trajectories are recorded and compared with the desired trajectory which is the center line of the road. One sample point is taken for every inch along the road. Figure 13 shows the desired trajectory and the actual trajectory we have recorded during the experiment. In the figure, the dashed line represents the desired

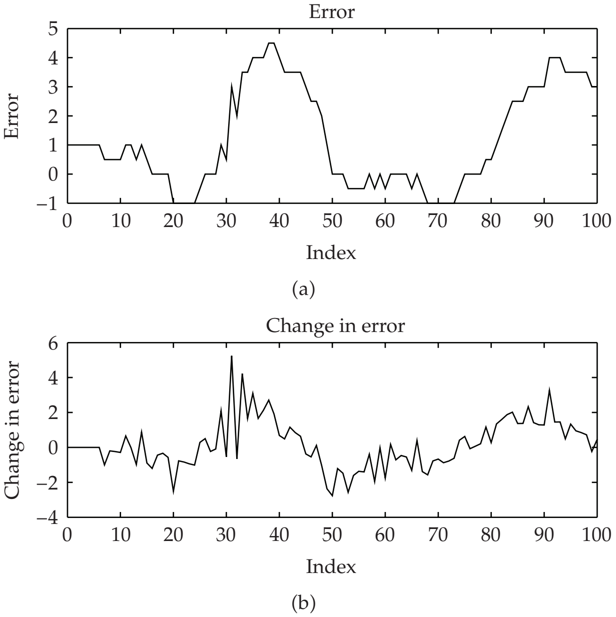


Figure 15: Error and change in error in the experiment.

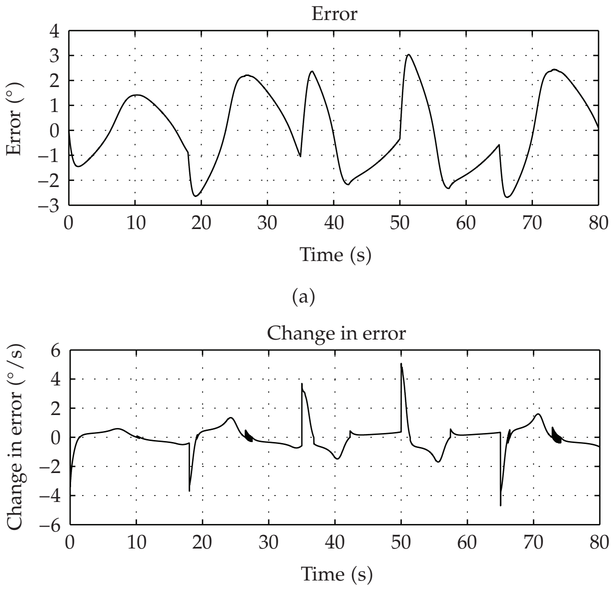
Table 2: Error calculation table.

Red centroid	Yellow centroid	Error
$Cr! = 0$	$Cy! = 0$	Error $= C - Cw$
$Cr! = 0$	$Cy = 0$	Error $= 176 - Cr$
$Cr = 0$	$Cy! = 0$	Error $= 0 - Cy$

trajectory while the dotted line represents the actual trajectory. The maximum absolute error between the center line of the road and the actual trajectory of the intelligent vehicle is 1.55 inches. The root mean square error is 0.73 inches which is relatively small compared with the width of the road which is 8 inches.

As a comparison, we also obtained simulation results for the vehicle moving at 0.9 m/s. Figure 14 shows the tracking performance of the vehicle. The desired trajectory is the one used in the experiment. It can be seen that experiment results in Figure 13 match the simulation results in Figure 14 very well. The maximum error in the experiment results is a little larger. This is because in the real world, the camera takes some time to process the raw data in order to obtain the error while in simulation the error is calculated instantaneously.

The error and change in error for both the experiment and the simulation are plotted in Figures 15 and 16. From the figures, it can be seen that the error is larger when there is a curve in the desired trajectory. This is because the desired trajectory is changing while the actual trajectory is deviated from the desired trajectory. The designed fuzzy controller adjusts to the changes and controls the vehicle to follow the trajectory successfully. It is also noted that in the experiment results, the change in error is filtered thus it is smoother than the simulated change in error.



(a)

(b)

Figure 16: Error and change in error in the simulation.

7. Conclusion

In this paper, an intelligent robotic vehicle which is capable of performing autonomous road following is successfully developed. A vision system installed on the vehicle is used to detect a curved road. An FLC is developed to control the steering wheel of the robotic vehicle according to the deviation from the rods. The deviation of the vehicle from the road is regarded as the error. The error and the change-in-error are used as the input of the FLC. Based on human driving experience and experiments, the FLC makes decision on the steering angle accordingly. An analysis and design of fuzzy control laws for steering control of the nonholonomic robotic vehicle are presented. Lyapunov's direct method is used to guarantee the convergence of the steering error. Simulations using the four degree of freedom nonholonomic robotic vehicle model are also conducted to investigate the performance and stability of the fuzzy controller. Experiments demonstrate that the vehicle with the proposed fuzzy controller can automatically follow the curved road.

Acknowledgment

Peter Miller, Matt Webb, and Chris Desjardins helped with the hardware setup.

References

[1] R. Bishop, "Intelligent vehicle applications worldwide," *IEEE Transactions on Intelligent Systems*, vol. 15, no. 1, pp. 78–81, 2000.

[2] A. M. Zhang and R. A. Russell, "Dominant orientation tracking for path following," in *Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 3885–3889, August 2005.

[3] D. Raviv and M. Herman, “A new approach to vision and control for road following,” in *Proceedings of the IEEE Workshop on Visual Motion*, pp. 217–225, October 1991.

[4] C. Thorpe, M. H. Hebert, T. Kanade, and S. A. Shafer, “Vision and navigation for the Carnegie-Mellon Navlab,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 10, no. 3, pp. 362–373, 1988.

[5] D. Raviv and M. Herman, “A unified approach to camera fixation and vision-based road following,” *IEEE Transactions on Systems, Man and Cybernetics*, vol. 24, no. 8, pp. 1125–1141, 1994.

[6] Z. Zhu, S. Yang, G. Xu, X. Lin, and D. Shi, “Fast road classification and orientation estimation using omni-view images and neural networks,” *IEEE Transactions on Image Processing*, vol. 7, no. 8, pp. 1182–1197, 1998.

[7] R. Chapuis, R. Aufrere, and F. Chausse, “Accurate road following and reconstruction by computer vision,” *IEEE Transactions on Intelligent Transportation Systems*, vol. 3, no. 4, pp. 261–270, 2002.

[8] L. Beji and Y. Bestaoui, “An adaptive control method of automated vehicles with integrated longitudinal and lateral dynamics in road following,” in *Proceeding of the 2nd International Workshop on Robot Motion and Control*, pp. 201–206, October 2001.

[9] S. Baluja, “Evolution of an artificial neural network based autonomous land vehicle controller,” *IEEE Transactions on Systems, Man, and Cybernetics, Part B*, vol. 26, no. 3, pp. 450–463, 1996.

[10] D. Y. Jeong, S. J. Park, S. H. Han, M. H. Lee, and T. Shibata, “A study on neural networks for vision-based road following of autonomous vehicle,” *IEEE International Symposium on Industrial Electronics*, vol. 3, pp. 1609–1614, 2001.

[11] S.-Y. Oh, J.-H. Lee, and D.-H. Choi, “A new reinforcement learning vehicle control architecture for vision-based road following,” *IEEE Transactions on Vehicular Technology*, vol. 49, no. 3, pp. 997–1005, 2000.

[12] C. C. Lee, “Fuzzy logic in control systems: fuzzy logic controller I,” *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 20, no. 2, pp. 404–418, 1990.

[13] T.-H. S. Li, S.-J. Chang, and Y.-X. Chen, “Implementation of human-like driving skills by autonomous fuzzy behavior control on an FPGA-based car-like mobile robot,” *IEEE Transactions on Industrial Electronics*, vol. 50, no. 5, pp. 867–880, 2003.

[14] J. L. Arroyabe, G. Aranguren, L. A.L. Nozal, and J. L. Martin, “Autonomous vehicle guidance with fuzzy algorithm,” in *Proceedings of 26th Annual Conference of IEEE Industrial Electronics Society (IECON '00)*, vol. 3, pp. 1503–1508, Nagoya, Japan, October 2000.

[15] S. Yasunobu and Y. Murai, “Parking control based on predictive fuzzy control,” in *Proceedings of the 3rd IEEE Conference on Fuzzy Systems*, vol. 2, pp. 1338–1341, Orlando, Fla, USA, June 1994.

[16] W. A. Daxwanger and G. K. Schmidt, “Skill-based visual parking control using neural and fuzzy networks,” in *Proceedings of the IEEE International Conference on Systems, Man and Cybernetics*, vol. 2, pp. 1659–1664, Vancouver, Canada, October 1995.

[17] M. C. Leu and T.-Q. Kim, “Cell mapping based fuzzy control of car parking,” in *Proceedings of IEEE International Conference on Robotics and Automation*, vol. 3, pp. 2494–2499, Leuven, Belgium, May 1998.

[18] P. A. Parrilo, *Structured semidefinite programs and semialgebraic geometry methods in robustness and optimization*, Ph.D. thesis, Caltech, Pasadena, Calif, USA, 2000.

[19] A. Papachristodoulou and S. Prajna, “On the construction of Lyapunov functions using the sum of squares decomposition,” in *Proceedings of the 41st IEEE Conference on Decision and Control*, vol. 3, pp. 3482–3487, Las Vegas, Nev, USA, December 2002.

[20] A. Megretski, *Dynamics of Nonlenear Systems*, Lecture Notes, Massachusetts Institute of Technology, Cambridge, Mass, USA, 2003.

[21] H. J. Marques, *Nonlinear Control Systems: Analysis and Design*, John Wiley & Sons, New York, NY, USA, 2003.

[22] Y. Fu, H. Li, and M. Kaye, “Design and stability analysis of a fuzzy controller for autonomous road following,” in *Proceedings of IEEE Intelligent Vehicles Symposium*, pp. 66–71, Xi’an, China, June 2009.