MITSUBISHI ELECTRIC RESEARCH LABORATORIES
http://www.merl.com

# Reinforcement Learning for Vision-Based Lateral Control of a Self-Driving Car

Huang, M.; Zhao, M.; Parikh, P.; Wang, Y.; Ozbay, K.; Jiang, Z.-P.

TR2019-074 July 25, 2019

**Abstract**

Lateral control design is one of the fundamental components for self-driving cars. In this paper, we propose a learning-based control strategy that enables a mobile car equipped with a camera to perfectly perform lane keeping in a road on the ground. Using the method of adaptive dynamic programming, the proposed control algorithm exploits the structural knowledge of the car kinematics as well as the collected data (images) about the lane information. An adaptive optimal lateral controller is obtained through a data-driven learning algorithm. The effectiveness of the proposed method is demonstrated by theoretical stability proofs and experimental evaluations.

*IEEE International Conference on Control and Automation (ICCA)*

# Reinforcement Learning for Vision-Based Lateral Control of a Self-Driving Car

Mengzhe Huang, Mingyu Zhao, Parthiv Parikh, Yebin Wang, Kaan Ozbay, and Zhong-Ping Jiang

***Abstract*—Lateral control design is one of the fundamental components for self-driving cars. In this paper, we propose a learning-based control strategy that enables a mobile car equipped with a camera to perfectly perform lane keeping in a road on the ground. Using the method of adaptive dynamic programming, the proposed control algorithm exploits the structural knowledge of the car kinematics as well as the collected data (images) about the lane information. An adaptive optimal lateral controller is obtained through a data-driven learning algorithm. The effectiveness of the proposed method is demonstrated by theoretical stability proofs and experimental evaluations.**

## I. Introduction

Vehicle motion control is a classical problem in the community of robotics and autonomous driving [1], [2]. The nonholonomic constraints associated with vehicles pose difficulties in the motion planning and control design for self-driving cars. To perform different maneuvers, such as lane keeping and lane change, the control tasks can be categorized into two subtasks: 1) path following; 2) trajectory tracking. In the task of path following, the reference path is independent of time. On the contrary, time is a constraint in the reference path for the trajectory tracking problem formulation [3]. During the last decades, extensive research efforts have been devoted to develop intelligent control algorithms that can solve the above two problems in the presence of modeling error and other forms of uncertainty.

For academic research, differential wheeled and car-like robots serve as affordable and easy-to-use hardware platforms to validate the performance of developed control algorithms. For the trajectory tracking problem, many control methods have been proposed in prior work, such as backstepping [4] and adaptive control [5]. These tracking controllers rely on the exact or estimated global position measurement. On the other hand, the path following problem can be solved based only on the local information of the mobile car, e.g., distance to the curve and heading angle error. Among the earliest path following approaches is pure pursuit [6]. Due to its simple implementation, pure pursuit has been employed in the DARPA Grand Challenge [7]. A chain-form based method was proposed in [8] to achieve path following for a car connected with several trailers. Recent investigations for robot motion control have focused on conventional model-based control solutions [9]–[11]. In [10], the authors designed a proportional controller and discussed the existence and practical stability of an equilibrium trajectory given that the reference path is a circular track. In the stability analysis, the exact curvature needs to be known in advance in order to eliminate the lateral offset. This limitation motivates this study.

In this paper, without knowing the exact curvature, we aim to design an image-based control method that allows a mobile car to perfectly perform lane keeping on a circular track, where the lateral deviation converges to zero as time goes to infinity. As shown later in this paper, the equilibrium point of the motion system is dependent on the curvature. In other words, unknown curvature gives rise to unknown kinematic model for path following. In addition, parametric uncertainties, e.g., unknown motor parameters, may undermine the control performance in practice. For the purpose of simultaneous adaptivity and optimality, this study presents a data-driven learning-based control algorithm using adaptive dynamic programming (ADP). ADP is an integration of reinforcement learning [12] and control theory. Adaptive optimal controllers can be learned from measurable data without the exact knowledge of system dynamics [13]–[18]. Recently, ADP-based control algorithms have been applied to the data-driven adaptive optimal control of connected and autonomous vehicles [19]–[21].

The main contribution of this paper is as follows: by employing the error kinematics with respect to the lane centerline, the lane keeping problem is solved by a data-driven learning-based control method. In particular, a combination of ADP and internal model principle is proposed to improve the lane keeping performance in order to 1) avoid the curvature estimation of the road; 2) overcome model uncertainties caused by unknown motor parameters. Compared to the early work [8]–[10], [22], our approach can achieve simultaneous adaptivity and optimality. Specifically, it intends to improve the control performance by reducing lateral offset and energy consumption. The efficacy of the proposed algorithm is validated through rigorous stability analysis and physical experiment using a Raspberry Pi based self-driving car. The experimental results show that the mobile car achieves lane keeping with minimal lateral offset.

**Notations.** Throughout this paper, $\mathbb{R}$ and $\mathbb{Z}_+$ denote the sets of real numbers and non-negative real numbers, respectively. $|\cdot|$ represents the Euclidean norm for vectors,

This work has been supported in part by the National Science Foundation under Grant ECCS-1501044, in part by a gift from the Mitsubishi Electric Research Laboratories and in part by C2SMART Center.

M. Huang, M. Zhao, P. Parikh and Z. P. Jiang are with the Department of Electrical and Computer Engineering, Tandon School of Engineering, New York University, Brooklyn, NY, 11201 USA (Contact email: {m.huang, zjiang}@nyu.edu).

Y. Wang is with Mitsubishi Electric Research Laboratories, Cambridge, MA 02139 USA, which funded his work on this topic.

K. Ozbay is with C2SMART Center, Tandon School of Engineering, New York University, Brooklyn, NY, 11201 USA.

or the induced matrix norm for matrices. $\otimes$ indicates the Kronecker product. $\text{vec}(A) = [a_1^T, a_2^T, \dots, a_m^T]^T$, where $a_i \in \mathbb{R}^n$ are the columns of $A \in \mathbb{R}^{n\times m}$. $I_n$ represents the $n \times n$ identity matrix. $0_{n\times m}$ denotes the $n \times m$ zero matrix. For a symmetric matrix $P \in \mathbb{R}^{m\times m}$, $\text{vecs}(P) = [p_{11}, 2p_{12}, \cdots, 2p_{1m}, p_{22}, 2p_{23}, \cdots, 2p_{m-1,m}, p_{mm}]^T \in \mathbb{R}^{\frac{1}{2}m(m+1)}$. For an arbitrary column vector $v \in \mathbb{R}^n$, $\text{vecv}(v) = [v_1^2, v_1v_2, \cdots, v_1v_n, v_2^2, \cdots, v_{n-1}v_n, v_n^2]^T \in \mathbb{R}^{\frac{1}{2}n(n+1)}$. $diag(a_1, a_2, \cdots)$ indicates a diagonal matrix where the entries on the main diagonal are $a_1, a_2, \cdots$ and the entries outside the main diagonal are all zero. Capital or lower-case letters in bold font refer to vectors.

## II. Kinematic Model of a Mobile Car

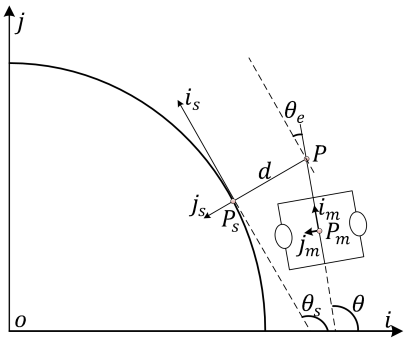



Fig. 1: Kinematic model of a mobile car

In this study, the mobile car is approximately modeled by a two-wheel differential-driven robot. Assuming that the mobile robot satisfies the nonslipping condition during the motion. Then, the simplified kinematic model of the unicycle-type mobile robot is

$$\dot{X} = v\cos\theta, \quad \dot{Y} = v\sin\theta, \quad \dot{\theta} = \omega, \tag{1}$$

where $(X, Y)$ represents the coordinates of the middle point $P_m$ located at the car's center of mass, $\theta$ specifies the orientation angle of the car's chassis in a reference frame, $v$ is the linear velocity of $P_m$, and $\omega$ is the chassis instantaneous velocity of rotation. Furthermore, the linear velocity $v$ and the angular velocity $\omega$ are regulated by the actuated wheels via the following relations

$$v = \frac{r}{2}(\omega_r + \omega_l), \tag{2a}$$

$$\omega = \frac{r}{2L}(\omega_r - \omega_l), \tag{2b}$$

where $r$ is the wheels' radius, $L$ is the distance between wheels of both sides, $\omega_l$ and $\omega_r$ are the angular velocities of the left and right wheels. The relationship between the motors (that drive the robot) and the angular velocity of wheels can be simplified as

$$\omega_i = bm_i, \tag{3}$$

where $i = \{l, r\}$, $m_i$ is the duty cycle of the micro-controller that regulates the motor speed, and $b$ is the coefficient. In short, a linear relationship between the duty cycle and the angular velocity of the wheel is assumed.

In this study, we focus on the controller design of the robot's lateral motion, while the linear velocity $v$ is constant. Define the duty-cycle difference between both sides as the control input $u$, i.e., $u = m_r - m_l$. Note that $u \in [-100\%, 100\%]$. Combining (2b) and (3), we have $\omega = b_m u$, where $b_m = \dfrac{rb}{2L}$. Thus, from (1), we have

$$\dot{\theta} = b_m u. \tag{4}$$

Recall that the objective here is to achieve lane keeping, which is an example of path following. We follow the same procedure as in [23, Chapter 34] and [10] to generalize the kinematic model (1) and (4) into a model in Frénet frame. As shown in Fig. 1, $\mathcal{F}_o = \{o, \boldsymbol{i}, \boldsymbol{j}\}$ is a fixed reference frame. $\mathcal{F}_m = \{P_m, \boldsymbol{i_m}, \boldsymbol{j_m}\}$ is a frame attached to the point $P_m$. $P$ is the look-ahead point such that $\boldsymbol{P_mP} = l_1\boldsymbol{i_m}$ with the previewed distance $l_1 > 0$. $\mathcal{F}_s = \{P_s, \boldsymbol{i_s}, \boldsymbol{j_s}\}$ is the frame of the tracked centerline at the target point $P_s$ such that the unit vector $i_s$ tangents the curve. $P_s$ is uniquely determined if the point $P$ is close enough to the curve.

In addition, we introduce the following variables to characterize the vehicle motion with respect to the curve:

1) $s$ is the curvilinear coordinate of $P_s$;
2) $d$ is the distance between $P$ and the curve, i.e., the ordinate of the point $P$ in the frame $\mathcal{F}_s$;
3) $\theta_e = \theta - \theta_s$ is the orientation angle error of the car with respect to the road;
4) $c(s)$ is curvature of the path at $P_s$.

By the definition of curvature, we have $c(s) = \dfrac{\partial\theta_s}{\partial s}$. It follows that

$$\begin{aligned}\dot{\theta}_e &= \dot{\theta} - \dot{\theta}_s \\ &= b_m u - \dot{s}c(s).\end{aligned} \tag{5}$$

Then, we note that $\boldsymbol{OP} = \boldsymbol{OP_s} + \boldsymbol{P_sP}$. Differentiating the equation with respect to time $t$ gives

$$\begin{aligned}\frac{\partial \boldsymbol{OP}}{\partial t} &= \frac{\partial \boldsymbol{OP_s}}{\partial t} + \frac{\partial \boldsymbol{P_sP}}{\partial t} \\ &= \dot{s}(1 - dc(s))\boldsymbol{i_s} + \dot{d}\boldsymbol{j_s},\end{aligned} \tag{6}$$

where the last equality is obtained from the Frénet formula. On the other hand, we have $\boldsymbol{OP} = \boldsymbol{OP_m} + \boldsymbol{P_mP}$ and $\frac{\partial \boldsymbol{OP_m}}{\partial t} = v\boldsymbol{i_m}$ which imply that

$$\begin{aligned}\frac{\partial \boldsymbol{OP}}{\partial t} =& \frac{\partial \boldsymbol{OP_m}}{\partial t} + \frac{\partial \boldsymbol{P_mP}}{\partial t} \\ =& (v\cos\theta_e - l_1 b_m u \sin\theta_e)\boldsymbol{i_s} \\ &+ (v\sin\theta_e + l_1 b_m u\cos\theta_e)\boldsymbol{j_s}.\end{aligned} \tag{7}$$

According to (6)-(7), the model describing the motion of the self-driving car

$$\dot{\xi} = f(\xi, u) = \begin{bmatrix} f_1(\xi, u) \\ f_2(\xi, u)\end{bmatrix} = \begin{bmatrix} v\sin\theta_e + l_1 b_m u\cos\theta_e \\ b_m u - \dot{s}c(s)\end{bmatrix}, \tag{8}$$

where $\xi = \begin{bmatrix} d, \theta_e\end{bmatrix}^T$, and $\dot{s} = \dfrac{1}{1 - dc(s)}\left[v - l_1 b_m u \sin\theta_e\right]$.

## III. Model-based Optimal Control of the Mobile Car

The control objective here is to achieve lane keeping on a track with constant curvature, which is to force $d$ to zero, i.e, $\lim_{t\to\infty} d = d^* = 0$. In this section, we present a model-based control design method with the exact knowledge of the track curvature $c$.

### A. Linearization and Discretization of the Motion System

To simplify our controller design procedure, the system (8) can be linearized around the equilibrium $\xi^* = \left[d^*, \theta_e^*\right]^T$ and $u^*$ which satisfy $f_1(\xi^*, u^*) = f_2(\xi^*, u^*) = 0$. After some algebraic manipulations, we have the following linearized model:

$$\dot{\xi} = F\xi + Gu + E, \tag{9}$$

where

$$F = \left.\begin{bmatrix} \dfrac{\partial f_1}{\partial d} & \dfrac{\partial f_1}{\partial \theta_e} \\ \dfrac{\partial f_2}{\partial d} & \dfrac{\partial f_2}{\partial \theta_e} \end{bmatrix}\right|_{\substack{\xi=\xi^* \\ u=u^*}}, G = \left.\begin{bmatrix} \dfrac{\partial f_1}{\partial u} \\ \dfrac{\partial f_2}{\partial u} \end{bmatrix}\right|_{\substack{\xi=\xi^* \\ u=u^*}}$$

$$E = \left.\begin{bmatrix} -\dfrac{\partial f_1}{\partial \theta_e}\theta_e^* - \dfrac{\partial f_1}{\partial u}u^* \\ -\dfrac{\partial f_2}{\partial \theta_e}\theta_e^* - \dfrac{\partial f_2}{\partial u}u^* \end{bmatrix}\right|_{\substack{\xi=\xi^* \\ u=u^*}}.$$

Note the constant disturbance matrix $E$ is introduced by the non-zero equilibrium point $(\xi^*, u^*)$ at steady state, because the curvature $c$ is also a non-zero constant for a circular track. Hence, for a mobile car, the motion system matrices $F$, $G$ and $E$ are determined by the track curvature $c$, the longitudinal velocity $v$ and the motor coefficient $b_m$.

To achieve digital control of the micro-motors, the linear motion system (9) can be discretized as follows

$$\xi_{k+1} = F_d\xi_k + G_d u_k + E_d, \tag{10}$$

where $F_d = e^{Fh}$, $G_d = \left(\int_0^h e^{F\tau}d\tau\right) G$, $E_d = (\int_0^h e^{F\tau}d\tau)E$, and $h > 0$ is the sampling period.

### B. Output Regulation of the Mobile Car

The output regulation problem aims to design a feedback controller to achieve asymptotic tracking with disturbance rejection for a reference signal while maintaining the closed-loop stability [24]. In our study, output regulation is a powerful tool to help reject the constant disturbance $E_d$ caused by the non-zero curvature $c$.

First, we define a compensator variable $z$ such that

$$z_{k+1} = z_k + d_k, \tag{11}$$

which essentially incorporates an internal model of the constant disturbance, see [24, Chapter 1]. The internal model plays an important role to define the augmented system that combines (10) and (11). Letting $x_k = \left[\xi_k, z_k\right]^T$, we have the following augmented system

$$x_{k+1} = Ax_k + Bu_k + D \tag{12}$$

where $A = \begin{bmatrix} F_d & 0 \\ C & 1 \end{bmatrix}, B = \begin{bmatrix} G_d \\ 0 \end{bmatrix}, D = \begin{bmatrix} E_d \\ 0 \end{bmatrix}, C = \begin{bmatrix} 1 & 0 \end{bmatrix}$. In particular, the matrix $C$ is the output matrix. The following theorem provides a potential controller design procedure to achieve path following on a circular track with the knowledge of the curvature $c$.

**Theorem 1.** *Suppose a controller $K = \begin{bmatrix} K_\xi & K_z \end{bmatrix}$ satisfies that all the eigenvalues $\lambda$ of the closed-loop system $A_{cl} = A - BK$ are located in the unit disk, i.e., $|\lambda| < 1$. Then, considering (12), we have $\lim_{k\to\infty} d_k = 0$.*

*Proof.* First, the existence of such a controller $K$ is guaranteed by [24, Lemma 1.37]. The closed-loop system (12) with controller $K$ is

$$x_{k+1} = (A - BK)x_k + D. \tag{13}$$

Then, the steady state $x^*$ is defined as

$$\begin{aligned} x^* &= (I_3 - (A - BK))^{-1}D \\ &= \left[\xi^*, z^*\right]^T, \end{aligned} \tag{14}$$

where $z^*$ is the steady state value of the compensator $z$. In particular, $z^*$ is determined by the parametrization of $K$. Accordingly, the steady-state input is $u^* = -Kx^*$.

Next, let the error state and the error input be $\bar{x}_k = x_k - x^*$ and $\bar{u}_k = u_k - u^*$, respectively. From (13), we have

$$\bar{x}_{k+1} = (A - BK)\bar{x}_k \tag{15}$$

which implies that $\lim_{k\to\infty} \bar{x}_k = 0$. Therefore, we have $\lim_{k\to\infty} x_k = x^*$, which gives that $\lim_{k\to\infty} d_k = d^* = 0$ and $\lim_{k\to\infty} u_k = u^*$. □

Note that the compensator $z$ is essentially an integrator of the distance $d$. Thus, the controller in Theorem 1 is equivalent to a proportional-integral (PI) controller.

### C. Optimal Control Formulation for Path Following

In this section, an optimal control formulation of path following is presented, aiming to minimize the lateral deviation and heading angle error as well as the energy consumption. Now, we introduce the following optimal control problem known as the linear quadratic regulator (LQR) problem:

$$\begin{aligned} \underset{\bar{u}}{\text{minimize}} \quad & \sum_{k=0}^{\infty} \bar{x}_k^T Q \bar{x}_k + r\bar{u}_k^2 \\ \text{subject to} \quad & \bar{x}_{k+1} = A\bar{x}_k + B\bar{u}_k \end{aligned}$$

where $Q = Q^T \geq 0$, $r > 0$ and $(A, \sqrt{Q})$ is observable. By linear optimal control theory [25], the gain $K^*$ of the optimal controller $\bar{u}_k = -K^*\bar{x}_k$ is given by

$$K^* = (r + B^T P^* B)^{-1} B^T P^* A, \tag{16}$$

where $P^* = P^{*T} > 0$ is the unique solution to the following algebraic Riccati equation

$$A^T P^* A - P^* + Q - A^T P^* B(r + B^T P^* B)^{-1} B^T P^* A = 0. \tag{17}$$

Here, an iterative algorithm is recalled from [26] to solve (17).

**Algorithm 1** Model-based Value Iteration Algorithm [26]

1: Choose a sufficiently small constant $\epsilon > 0$. $j \leftarrow 0$. $P_j \leftarrow 0_{3\times 3}$.
2: **repeat**
3: Compute $P_{j+1}$ and $K_{j+1}$ by

$$P_{j+1} \leftarrow A^T P_j A + Q - A^T P_j B(r + B^T P_j B)^{-1} B^T P_j A \tag{18}$$

$$K_{j+1} \leftarrow (r + B^T P_{j+1} B)^{-1} B^T P_{j+1} A \tag{19}$$

4: $j \leftarrow j + 1$
5: **until** $|P_j - P_{j-1}| < \epsilon$

**Remark 1.** *The following property holds for the sequences $\{P_j\}_{j=1}^{\infty}$ and $\{K_j\}_{j=1}^{\infty}$ obtained from Algorithm 1:*

$$\lim_{j\to\infty} K_j = K^*, \lim_{j\to\infty} P_j = P^*.$$

We refer the interested reader for the convergence proof of the sequences in Remark 1 to [26], [27]. Hence, by iteratively solving (18)-(19), the optimal control gain $K^*$ can be obtained.

Hitherto, a model-based optimal control design has been proposed to accomplish path following with the exact knowledge of the curvature $c$ and the motor coefficient $b$. However, these parameters, including $r$ and $L$ in (2), can constitute uncertainties in practice, while the model-based method cannot guarantee the stability/optimality of closed-loop system with unknown $c$, $b$, $r$ and $L$. We shall overcome this limitation by a data-driven learning-based approach presented in the next section.

## IV. Data-driven Learning-based Approach for Lateral Control

In this section, we develop a data-driven approach to obtain the controller $K^*$ using an ADP method which assumes no knowledge of the curvature of the track.

First, we define the following matrix $H_j$:

$$\begin{aligned} H_j &= \begin{bmatrix} H_j^{11} & H_j^{12} & H_j^{13} \\ (H_j^{12})^T & H_j^{22} & H_j^{23} \\ (H_j^{13})^T & (H_j^{23})^T & H_j^{33} \end{bmatrix} \\ &:= \begin{bmatrix} B^T P_j B & B^T P_j D & B^T P_j A \\ D^T P_j B & D^T P_j D & D^T P_j A \\ A^T P_j B & A^T P_j D & A^T P_j A \end{bmatrix}. \end{aligned}$$

Then, the augmented system (12) can be rewritten into

$$\begin{aligned} x_{k+1} &= Ax_k + Bu_k + D \\ &= A_j x_k + B(K_j x_k + u_k) + D, \end{aligned} \tag{20}$$

where $A_j = A - BK_j$. From (18) and (20), it follows that

$$\begin{aligned} & x_{k+1}^T Q x_{k+1} \\ =& - x_{k+1}^T \mathcal{F}(P_j) x_{k+1} + x_{k+1}^T P_{j+1} x_{k+1} \\ =& - x_{k+1}^T \left[H_j^{33} - (H_j^{13})^T (r + H_j^{11})^{-1} H_j^{13}\right] x_{k+1} \\ & + \left( \begin{bmatrix} u_k \\ 1 \\ x_k \end{bmatrix} \otimes \begin{bmatrix} u_k \\ 1 \\ x_k \end{bmatrix} \right)^T \mathrm{vec}(H_{j+1}) \\ =& x_{k+1}^T \left[H_j^{33} - (H_j^{13})^T (r + H_j^{11})^{-1} H_j^{13}\right] x_{k+1} \\ & + \left[ \mathrm{vecv} \left( \begin{bmatrix} u_k \\ 1 \\ x_k \end{bmatrix} \right) \right]^T \mathrm{vecs}(H_{j+1}) \\ =& - \phi_{k+1}^j + \psi_k^T \mathrm{vecs}(H_{j+1}), \end{aligned} \tag{21}$$

where

$$\begin{aligned} \mathcal{F}(P_j) &= A^T P_j A - A^T P_j B(r + B^T P_j B)^{-1} B^T P_j A, \\ \phi_{k+1}^j &= x_{k+1}^T \left[H_j^{33} - (H_j^{13})^T (r + H_j^{11})^{-1} H_j^{13}\right] x_{k+1}, \\ \psi_k &= \mathrm{vecv}\left( \begin{bmatrix} u_k^T & 1 & x_k^T \end{bmatrix}^T \right). \end{aligned}$$

Note that (21) holds for any time instant $k \in \mathbb{Z}_+$. In order to determine $H_{j+1}$, measurable data, i.e., $x$ and $u$, are collected at multiple time steps $k_0 < k_1 < \cdots < k_N$ where $N$ is a sufficiently large positive integer. In particular, we can define

$$\begin{aligned} \Psi &= \left[\psi_{k_0}, \psi_{k_1}, \ldots, \psi_{k_N}\right]^T, \\ \Phi_j &= \left[x_{k_0+1}^T Q x_{k_0+1} + \phi_{k_0+1}^j, \ldots, x_{k_N+1}^T Q x_{k_N+1} + \phi_{k_N+1}^j\right]^T. \end{aligned}$$

Then, (21) can be put into the following matrix equation:

$$\Psi \mathrm{vecs}(H_{j+1}) = \Phi_j. \tag{22}$$

**Assumption 1.** *There exists a positive integer $N^*$ such that for all $N > N^*$ and for time instant $k_0 < k_1 < \cdots < k_N$, $\Psi$ has full column rank.*

**Remark 2.** *To make Assumption 1 satisfied, some exploration noise $\eta$ can be added to the input [15].*

Under Assumption 1, (22) can be solved by the least-squares method to find $H_{j+1}$

$$\left[\mathrm{vecs}(H_{j+1})\right] = (\Psi^T \Psi)^{-1} \Psi^T \Phi_j. \tag{23}$$

Using obtained $H_{j+1}$ and (19), the control policy $K_{j+1}$ can be updated as follows

$$K_{j+1} = (r + H_j^{11})^{-1} H_j^{13}. \tag{24}$$

Finally, the ADP-based control algorithm for path following is summarized.

**Algorithm 2** ADP-based Algorithm for Path Following

1: Choose a sufficiently small constant $\epsilon > 0$.
2: Apply an arbitrary initial control policy with exploration noise $\eta$ and collect data $x$ and $u$ until the rank condition in Assumption 1 is satisfied. $j \leftarrow 0$. $H_j \leftarrow 0_{5\times 5}$.
3: **repeat**
4: Compute $H_{j+1}$ and $K_{j+1}$ by (23) and (24)
5: $j \leftarrow j+1$
6: **until** $|H_j - H_{j-1}| < \epsilon$

**Theorem 2.** *Under Assumption 1, the obtained sequences* $\{H_j\}_{j=1}^{\infty}$ *and* $\{K_j\}_{j=1}^{\infty}$ *from (23) and (24) satisfy*

$$\lim_{j\to\infty} H_j = H^*, \ \lim_{j\to\infty} K_j = K^*,$$

*where* $H^*$ *is defined as*

$$H^* := \begin{bmatrix} B^TP^*B & B^TP^*D & B^TP^*A \\ D^TP^*B & D^TP^*D & D^TP^*A \\ A^TP^*B & A^TP^*D & A^TP^*A \end{bmatrix}.$$

*Proof.* Let $P_{j+1}$ be the solution of (18), which uniquely determines $H_{j+1}$. By (21), it is obvious that $H_{j+1}$ satisfies (23). On the other hand, let $H$ be a solution of (23). Since $\Psi$ is of full rank, we have $H = H_{j+1}$, which implies that the least square solution of (23) is equivalent to the one in Algorithm 1. Thus, the convergence of $H_{j+1}$ is obtained. The convergence of $K_{j+1}$ follows directly, which completes the proof. □

Next, we give the local stability analysis when the learned controller is applied to the mobile car.

**Theorem 3.** *Let* $K_{j^*}$ *be the learned controller from Algorithm 2, where* $j^*$ *is the index when the algorithm is terminated. After applying* $K_{j^*}$ *to the vehicle motion system (12), the lane keeping task is accomplished, i.e.,* $\lim_{k\to\infty} d_k = 0$.

*Proof.* By Theorem 2, the closed-loop system becomes

$$x_{k+1} = (A - BK_{j^*})x_k + D, \tag{25}$$

which is a stable system, i.e., all the eigenvalues of $(A - BK_{j^*})$ are located in the unit disk. According to Theorem 1, we have $\lim_{k\to\infty} d_k = 0$. □

## V. Experimental Results

### A. Description of the Mobile Car

The mobile car has two identical wheels on either side. During the movement, the control signal to the front wheel of either left or right side is always the same as the rear one. Hence, the rotation speeds of the front wheels are the same as the rear ones. See Fig. 2.

The vision device consists of a wide angle camera with a resolution of 640×480. It is fixed on the mobile car with a lateral angle of view 110 degrees.

A computer vision program is designed to detect the two lane boundaries of the track, which determines the centerline as the path to follow (see Fig. 3). The output of the vision system is $\xi = \begin{bmatrix} d, \theta_e \end{bmatrix}^T$.

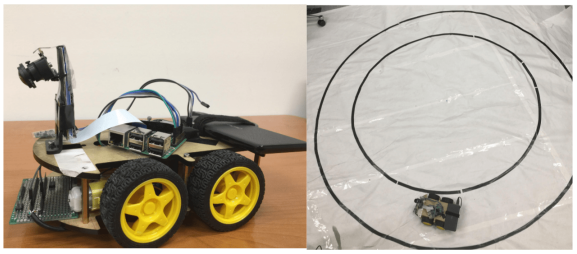


Fig. 2: Experimental mobile car and a test circular track

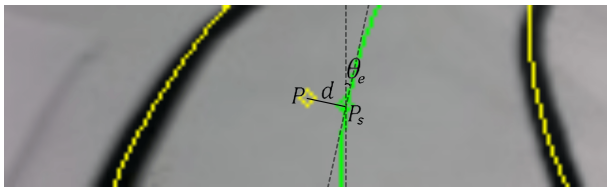



Fig. 3: Processed image including detected lane boundaries, lane centerline and $d$, $\theta_e$

### B. Performance Evaluation

In order to compare the learned control policies with an initial control policy, the initial condition for each trial is fixed at $\xi_0 = \begin{bmatrix} d_0, \theta_{e,0} \end{bmatrix}^T$ where $d_0 = 20$ [cm] and $\theta_{e,0} = 0.4$ [rad]. The longitudinal velocity $v$ is also fixed. The weighting matrices are set as $Q = diag(8, 0.1, 0.1)$ and $r = 1$.

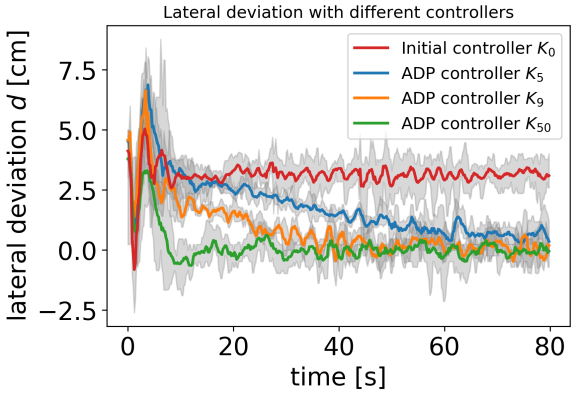
The initial controller is chosen as $K_0 = \begin{bmatrix} 6 & 0 & 0 \end{bmatrix}$. The resulting performance is depicted in Fig. 4, where the initial controller is tested for 5 trials. The solid line is the average trajectory of these 5 trials, and the shaded area indicates the variations between each trial. It is observed that the initial control policy cannot regulate the lateral deviation to 0 as desired, which approaches 2.7 [cm] as time goes. All the data of these 5 trials are collected to initiate our proposed Algorithm 2.

The ADP algorithm is terminated at 50th iteration. As illustrated in Algorithm 2, the controller is updated at each iteration. Thus, we select three controllers in these iterations to demonstrate the learning and improved performance. In particular, controllers $K_5$, $K_9$ and $K_{50}$ are tested on the same track, where $K_5 = \begin{bmatrix} 4.46 & 33.31 & 0.02 \end{bmatrix}$, $K_9 = \begin{bmatrix} 4.61 & 34.65 & 0.05 \end{bmatrix}$ and $K_{50} = \begin{bmatrix} 5.31 & 39.14 & 0.28 \end{bmatrix}$. Each of the controllers is tested for 5 trials. The resultant performance is also included in Fig. 4.

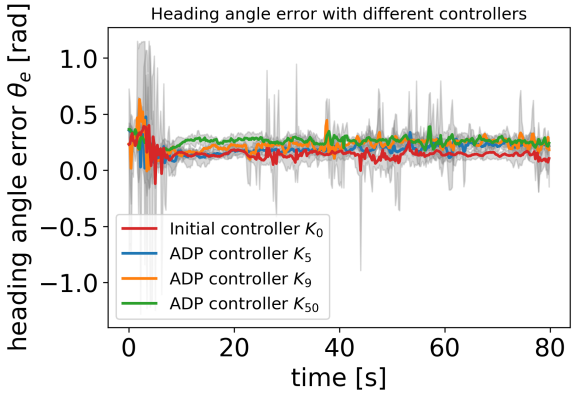
We note that all three controllers can force the lateral deviation to the neighborhood of 0, which achieves the goal of lane keeping. ADP controller $K_5$ regulate the lateral error slower, and $K_{50}$ makes the mobile car converge to the equilibrium in less than 10 seconds. The convergence of $K_{50}$ is consistent with the theoretical result in Theorem 3.

## VI. Conclusion and Future Work

In this paper, a data-driven learning-based algorithm has been proposed to solve the lateral motion control problem for a mobile car, using image as the feedback signal. Based on




(a) Trajectories of lateral deviation $d$




(b) Trajectories of heading angle error $\theta_e$

Fig. 4: Performance evaluation with different controllers

a synthesis of ADP and internal model method, an adaptive optimal controller is learned from collected data, such as lateral offset and heading angle error extracted from the image. Using the proposed approach, the self-driving car achieves nearly optimal lane keeping performance without knowing the exact curvature of the circular track and the coefficient of motor system. Rigorous stability proofs are presented in the study and our experimental results are consistent with the theoretical results. Our future work will focus on the learning-based control design of a platoon of self-driving cars in order to accomplish different maneuvers, such as car following, lane changing and collision avoidance.

## Acknowledgment

The authors would like to thank Manuel Serrano Rebuelta and Yutong Liu for discussions on the experimental design of computer vision and motor control systems.

## References

[1] I. Kolmanovsky and N. H. McClamroch, "Developments in nonholonomic control problems," *IEEE Control Systems Magazine*, vol. 15, pp. 20–36, Dec 1995.

[2] B. Paden, M. Cáp, S. Z. Yong, D. Yershov, and E. Frazzoli, "A survey of motion planning and control techniques for self-driving urban vehicles," *IEEE Transactions on Intelligent Vehicles*, vol. 1, pp. 33–55, March 2016.

[3] A. P. Aguiar and J. P. Hespanha, "Trajectory-tracking and path-following of underactuated autonomous vehicles with parametric modeling uncertainty," *IEEE Transactions on Automatic Control*, vol. 52, pp. 1362–1379, Aug 2007.

[4] Z. P. Jiang and H. Nijmeijer, "Tracking control of mobile robots: A case study in backstepping," *Automatica*, vol. 33, pp. 1393–1399, 1997.

[5] K. Wang, Y. Liu, and L. Li, "Visual servoing trajectory tracking of nonholonomic mobile robots without direct position measurement," *IEEE Transactions on Robotics*, vol. 30, pp. 1026–1035, Aug 2014.

[6] R. C. Coulter, "Implementation of the pure pursuit path tracking algorithm," tech. rep., Carnegie-Mellon UNIV Pittsburgh PA Robotics Institute, 1992.

[7] M. Buehler, K. Iagnemma, and S. Singh, *The DARPA Urban Challenge: Autonomous Vehicles in City Traffic*. New York, NY, USA: Springer, 2009.

[8] C. Samson, "Control of chained systems application to path following and time-varying point-stabilization of mobile robots," *IEEE Transactions on Automatic Control*, vol. 40, pp. 64–77, Jan 1995.

[9] M. Egerstedt, X. Hu, and A. Stotsky, "Control of mobile platforms using a virtual vehicle approach," *IEEE Transactions on Automatic Control*, vol. 46, pp. 1777–1782, Nov 2001.

[10] J. B. Coulaud, G. Campion, G. Bastin, and M. D. Wan, "Stability analysis of a vision-based control design for an autonomous mobile robot," *IEEE Transactions on Robotics*, vol. 22, pp. 1062–1069, Oct 2006.

[11] C. Chen, J. Li, M. Li, and L. Xie, "Model predictive trajectory tracking control of automated guided vehicle in complex environments," in *IEEE International Conference on Control and Automation*, pp. 405–410, June 2018.

[12] R. S. Sutton and A. G. Barto, *Reinforcement learning: An introduction*. Cambridge, MA: MIT press, 2018.

[13] Y. Jiang and Z. P. Jiang, *Robust Adaptive Dynamic Programming*. Hoboken NJ: Wiley-IEEE Press, 2017.

[14] T. Bian and Z. P. Jiang, "Value iteration and adaptive dynamic programming for data-driven adaptive optimal control design," *Automatica*, vol. 71, pp. 348 – 360, 2016.

[15] W. Gao, Y. Jiang, Z. P. Jiang, and T. Chai, "Output-feedback adaptive optimal control of interconnected systems based on robust adaptive dynamic programming," *Automatica*, vol. 72, pp. 37 – 45, 2016.

[16] B. Kiumarsi, K. G. Vamvoudakis, H. Modares, and F. L. Lewis, "Optimal and autonomous control using reinforcement learning: A survey," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 29, pp. 2042–2062, June 2018.

[17] D. Wang, H. He, and D. Liu, "Adaptive critic nonlinear robust control: A survey," *IEEE Transactions on Cybernetics*, vol. 47, pp. 3429–3451, Oct 2017.

[18] B. Pang, T. Bian, and Z. P. Jiang, "Adaptive dynamic programming for finite-horizon optimal control of linear time-varying discrete-time systems," *Control Theory and Technology*, vol. 17, pp. 73–84, Feb 2019.

[19] W. Gao, Z. P. Jiang, and K. Ozbay, "Data-driven adaptive optimal control of connected vehicles," *IEEE Transactions on Intelligent Transportation Systems*, vol. 18, pp. 1122–1133, May 2017.

[20] M. Huang, W. Gao, Y. Wang, and Z. P. Jiang, "Data-driven shared steering control of semi-autonomous vehicles," *IEEE Transactions on Human-Machine Systems*, 2019. doi: 10.1109/THMS.2019.2900409.

[21] Y. Zhu, D. Zhao, and Z. Zhong, "Adaptive optimal control of heterogeneous CACC system with uncertain dynamics," *IEEE Transactions on Control Systems Technology*, 2018. doi: 10.1109/TCST.2018.2811376.

[22] A. K. Das, R. Fierro, V. Kumar, B. Southall, J. Spletzer, and C. J. Taylor, "Real-time vision-based control of a nonholonomic mobile robot," in *IEEE International Conference on Robotics and Automation*, vol. 2, pp. 1714–1719, May 2001.

[23] A. Eskandarian, *Handbook of intelligent vehicles*. London, U.K.: Springer, 2012.

[24] J. Huang, *Nonlinear Output Regulation: Theory and Applications*. Philadelphia, PA: SIAM, 2004.

[25] F. L. Lewis, D. Vrabie, and V. L. Syrmos, *Optimal Control*. John Wiley & Sons, 2012.

[26] P. Lancaster and L. Rodman, *Algebraic Riccati Equations*. Clarendon press, 1995.

[27] D. P. Bertsekas, *Dynamic Programming and Optimal Control*, vol. 1. Belmont, MA: Athena scientific, 2005.