

# Development of a Small-Scale Research Platform for Intelligent Transportation Systems

Hung Manh La, Ronny Salim Lim, Jianhao Du, Sijian Zhang, Gangfeng Yan, and Weihua Sheng, *Senior Member, IEEE*

**Abstract**—In this paper, we propose and develop a small-scale research platform for intelligent transportation systems (ITSs). Our platform has four main parts, i.e., an arena, an indoor localization system, automated radio-controlled (RC) cars, and roadside monitoring facilities. First, to mimic traffic environments, we build an arena with a wooden floor, mock buildings, and streets. Second, to facilitate feedback control for trajectory following, an indoor localization system is set up to track the RC cars. Third, both autonomous driving RC cars and human driving RC cars are developed, based on an automated RC car design. The automated RC cars can receive control signals from a computer through an Xbee RF module and control the front and rear wheels through motors. A new control algorithm is developed to allow the RC cars to track predefined trajectories. Finally, we implement an example of roadside monitoring, which uses a fish-eye camera associated with advanced video processing for image segmentation, object identification, and tracking. Experiments are performed to demonstrate the effectiveness of the designed platform. We also discuss possible ITS research problems that can be studied in this testbed.

**Index Terms**—Automated highway systems, autonomous vehicles, driver assistance, intelligent transportation systems (ITSs), intervehicle communications, vehicle control.

## I. INTRODUCTION

WITH NEARLY 43 000 deaths a year on U.S. roads [1], [2] and increasing traffic delays in major metropolitan areas, a need exists for countermeasures to reduce the number and severity of traffic accidents and relieve traffic jams. Intelligent transportation systems (ITSs) have attracted increasing attention in recent years due to their great potential in meeting this need [3]–[5]. There are many areas of ITS that are undergoing rigorous research, such as the reduction in vehicle accidents, realization of automatic driving, relief of traffic congestion, and improvement of the environment [1].

Manuscript received April 4, 2011; revised November 30, 2011 and March 11, 2012; accepted May 29, 2012. Date of publication July 25, 2012; date of current version November 27, 2012. This work was supported in part by the Oklahoma Transportation Center under Grant OTCREOS10.1-43 and Grant OTCREOS9.1-15 and in part by the National Science Foundation under Grant CISE/CNS MRI 0923238. The Associate Editor for this paper was D. J. J. Dailey.

H. M. La and R. S. Lim are with the Center for Advanced Infrastructure and Transportation, Rutgers University, Piscataway, NJ 08854 USA (e-mail: hung.la11@rutgers.edu; ronny.lim@okstate.edu).

J. Du and W. Sheng are with the School of Electrical and Computer Engineering, Oklahoma State University, Stillwater, OK 74078 USA (e-mail: jianhao.du@okstate.edu; weihua.sheng@okstate.edu).

S. Zhang and G. Yan are with Zhejiang University, Zhejiang 310027, China (e-mail: zsjzju@yahoo.com.cn; ygf@zju.edu.cn).

Color versions of one or more of the figures in this paper are available online at <http://ieeexplore.ieee.org>.

Digital Object Identifier 10.1109/TITS.2012.2207454

ITSs have attracted many researchers around the world [1], [2], [6]–[9]. The overview of main issues, technological challenges, developments, and achievements of ITSs can be found in [10] and [11]. For example, Tubaishat *et al.* [12] studied the use of wireless sensor networks in ITSs to collect traffic data for traffic planning and management. Recently, autonomous vehicles such as Google's self-driving cars [13] and the fully automated Buick LeSabre vehicles have been developed [14], [15]. Integrated in an autonomous vehicle, a real-time hierarchical simultaneous localization and mapping system was developed by Schleicher *et al.* [16]. The integrated vehicle-based safety systems (IVBSSs) have been developed and integrated in a fleet of 16 passenger cars and 10 heavy trucks. Their goal is to examine the effect of the prototype of the IVBSS integrated crash warning system on driving behavior and driver acceptance [17], [18]. The Partners for Advanced Transportation Technology (PATH) project conducted in California [19] mainly focuses on research issues in traffic operations, transportation safety, and modal applications.

There are many challenges in conducting full-scale ITS research, particularly when vehicles are involved. For example, modifying conventional vehicles to autonomous vehicles is usually costly in terms of money and time. Testing in real traffic environments is usually dangerous. Such challenges have posed significant obstacles to many researchers in ITSs and forced them to use pure computer simulation [20], in which it is extremely hard to mimic certain aspects such as communication, vehicle dynamics, and driving experience. For many ITS research problems, a scale-down platform may be very useful for preliminary study and feasibility tests. However, there is very limited previous work in developing such a multipurpose ITS research platform. Therefore, in this paper, we adopt an approach that strikes a balance between real-size vehicles and pure computer simulations. We aim to build a small-scale testbed that can simulate real traffic environments, vehicle communication, and human driving experience. The testbed will be easy to access, easy to duplicate, and cost effective while maintaining a reasonable degree of fidelity to real worlds. The testbed can be used to conduct research in ITSs, e.g., intelligent traffic monitoring, autonomous vehicle collision avoidance, vehicle black box investigation, and mixed human and autonomous driving in future ITSs.

As an important part of this testbed, it is desirable to develop autonomous driving RC cars that can follow predefined trajectories so that different traffic scenarios can be realized. However, controlling nonholonomic RC cars with low-accuracy steering angles and nonsmooth velocity is a challenging

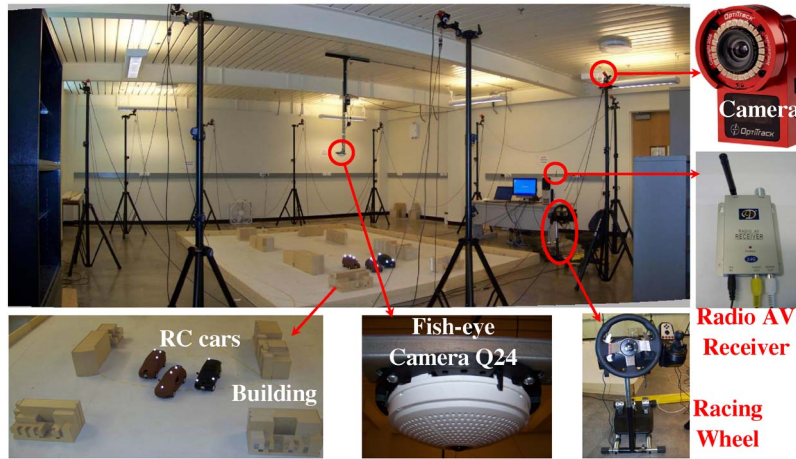


Fig. 1. Overall platform for ITS research.

problem. We propose an efficient feedback control algorithm based on virtual vehicles to allow the RC cars to track predefined trajectories.

The rest of this paper is organized as follows. In the next section, we present the hardware setup of the testbed. Section III shows the control hardware design of the automated RC car. Section IV describes the RC car kinematic model, the tracking control algorithm, and the control of multiple RC cars, respectively. Section V presents the vision-based monitoring of traffic. Section VI shows the experimental results. Section VII presents the potential research problems we can investigate using this testbed. Finally, Section VIII concludes this paper.

## II. HARDWARE SETUP OF THE TESTBED

### A. Overview

The scale-down ITS testbed we developed has four main parts.

- 1) arena;
- 2) indoor localization system;
- 3) automated radio controlled (RC) cars;
- 4) roadside monitoring facilities.

To mimic typical traffic environments, we build an arena with a wooden floor, mock buildings, and streets. An indoor localization system built from an optical motion capture system is developed. Automated RC cars with both autonomous driving and human driving capability are developed. For the roadside monitoring facilities, an overhead fish-eye camera is used, and the associated advanced video processing algorithms are developed, which include image segmentation, object identification, and tracking. The overall testbed is shown in Fig. 1.

### B. Arena

We built an arena with a dimension of 16 ft by 12 ft, which can be used to create various mock environments. The arena is based on a wooden floor on which streets, roads, and intersections can be set up. A carpet on top of the wooden floor is used to mimic concrete or asphalt road surfaces. We can also

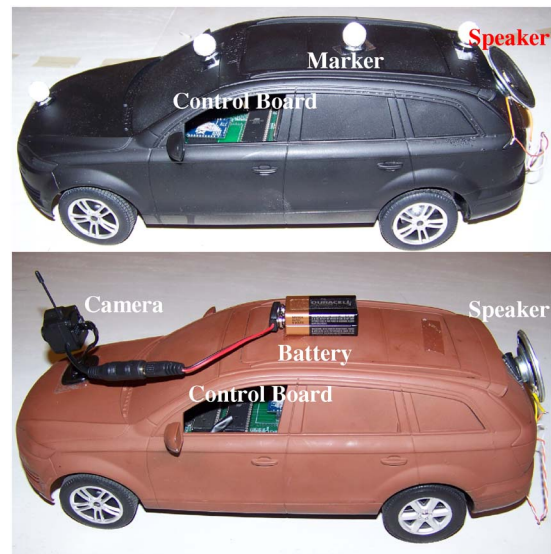


Fig. 2. Two automated RC car. (Top) Autonomous driving RC car. (Bottom) Human driving RC car.

place mock buildings, trees, and other decorations to make the scene more realistic.

### C. Indoor Localization System

An indoor localization system is built up to localize RC cars in the simulated traffic environment. The purpose of this system is to provide location feedback of the cars to control them to move along predefined tracks. This indoor localization system can mimic the function of the Global Positioning System (GPS) in the real world. This system is developed from an optical motion capture system (Opti-Track) from NaturalPoint Inc. [21]. There are 12 cameras mounted on tripods to cover the whole arena. The Opti-Track system has the capability of capturing 100 frame/s, so that the location and orientation information can be obtained in real time and with high accuracy (above 95%). The Opti-Track system tracks each RC car via the markers (see Fig. 2) mounted on top of the RC car. To simulate real-world GPS signals, we can down sample the data and even inject noise into the RC car location and orientation estimates.



Fig. 3. Setup for manually driving RC cars.

D. Automated RC Cars

We have developed both autonomous driving and human driving RC cars. They are based on the design of an automated RC car, which will be explained in Section III.

For the autonomous driving RC car [see Fig. 2 (top)], four markers are mounted on top of an automated RC car to build a rigid body, so that the location and orientation of the car can be tracked. A speaker is mounted on the back of the car for the purpose of mimicking collision sound, which can be used in automated collision detection research. The tracking control algorithm that allows the RC car to track predefined trajectories is developed in the computer, and the control commands are sent to the RC car via the Xbee wireless communication.

For the human driving RC car [Fig. 2 (bottom)], a miniature wireless camera is mounted on the hood of the RC car to provide visual inputs. It is used to observe the environment in front of the car and send the video stream through wireless communication to the personal computer (PC). The human driver sits in front of a wheel stand and drives the RC car while he/she observes the video stream on the monitor. We developed a program using the software development kit of the wheel stand to read the data from the steering wheel, which include the wheel-turning angle, brake, gas pedal, and gear shift status. Based on that, we send control commands, such as “move forward,” “backward,” “turn left,” “turn right,” “speed up,” or “slow down,” through the Xbee wireless communication to the automated RC car. Such a human driving setup can partially mimic the human driving experience. The whole setup of this human driving system is shown in Fig. 3. The human driving RC car is also equipped with speakers to generate collision or other sounds to mimic real traffic.

E. Roadside Monitoring Facilities

A Mobotix Q24 fish-eye camera as shown in Fig. 1 is mounted over the arena to serve as a roadside monitoring facility. This camera is capable of simultaneously provid-

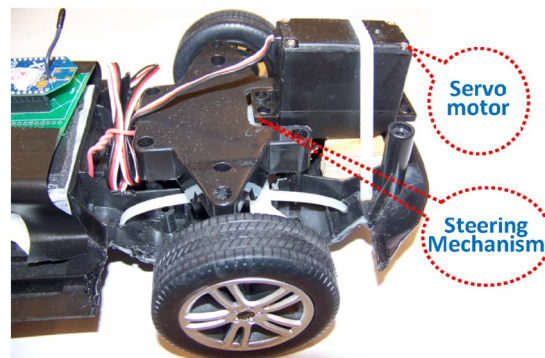


Fig. 4. Servo motor is mounted in the RC car.

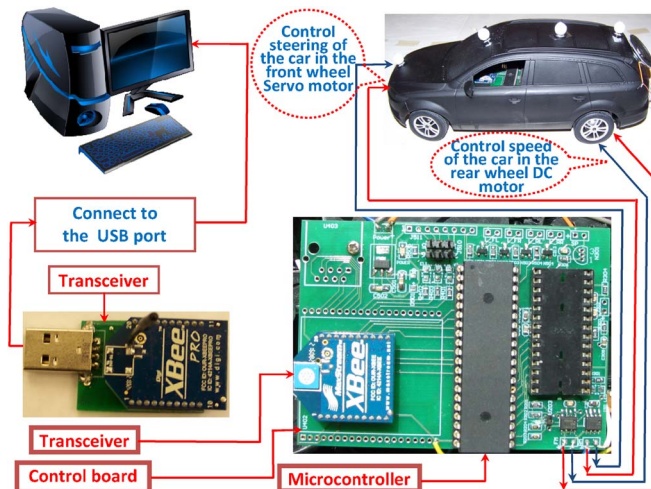


Fig. 5. Hardware setup for the RC car control.

ing different views, including a full 360° all-around view; hence, it can cover the whole arena to monitor the traffic underneath it. This camera uses an Internet Protocol (IP)-based interface. The stream of live images from the camera is obtained through a socket connection. The features of the camera (including resolutions, frame rates, etc.) can be easily modified by sending a web request. The zooming and panning of the camera lenses can be controlled by virtual pan, tilt, and zoom functions. The camera provides the highest resolution of 3M pixels, and the color images are scalable from 160×120 to 2048×1536. The highest frame rate is 30 frame/s. This camera can be used in research projects involving traffic monitoring, such as automated collision detection or anomaly detection through visual surveillance. In the future, more sensors, such as microphone arrays, will be deployed to further enhance the monitoring capability.

III. HARDWARE DESIGN OF AUTOMATED RADIO CONTROLLED CAR

We used commercial off-the-shelf RC cars with a scale of 1 : 14. The RC car comes with two dc motors: a front dc motor for steering control and a rear dc motor for speed control. After testing the front dc motor, we find that it has very poor steering performance and cannot be used in our project. Therefore, the front dc motor is replaced by a servo motor, which is mounted

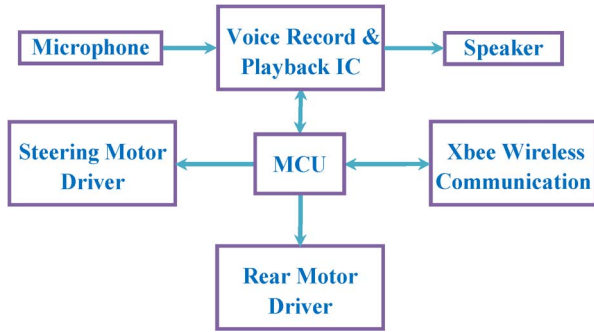


Fig. 6. Function blocks of the control board.

in the RC car, as shown in Fig. 4. The overall hardware design of the automated RC car control is shown in Fig. 5. There are two major parts in the hardware design: 1) an XBee wireless module and 2) a control board embedded in the RC car body.

**XBee wireless module** This module is using small low-power radio based on the IEEE 802.15.4 standard and originally targeted for wireless personal area networks. Due to the limited size of our testbed, we find that it is a good solution to wireless communication in our testbed. This XBee wireless module has a data rate up to 250 kb/s and can serve as the communication channel between RC cars and between RC cars and roadside infrastructures. For the purpose of automated RC car control, one XBee module is connected to the PC, and another is mounted on the control board as shown in Fig. 5, enabling the wireless communication between the PC and automated RC cars.

**Control board** An embedded control board is developed to replace the original circuit board inside the RC car. Its design is shown in Fig. 6. The microcontroller unit (MCU) (ATmega 162) at the middle of Fig. 6 functions as the control unit for the automated RC car. A speaker is used for playing recorded sounds to mimic real-world traffic sounds while a microphone is used to record environmental sounds. Both the speaker and the microphone are driven by a voice record IC (ISD 1700). The pulsewidth modulated (PWM) output from the MCU is used to drive the front servo motor and the rear DC motor so that the orientation and the velocity of the RC car can be controlled, respectively.

#### IV. AUTONOMOUS RADIO CONTROLLED CAR CONTROL

To create various traffic scenarios, we need control of the autonomous RC cars, so that they can follow predefined trajectories. This way, we can develop scripts that describe the desired movement of the vehicles in the arena. For example, we can create a scenario of head-on collision between two cars exactly the way we want.

To simultaneously control multiple RC cars, the control algorithm is implemented using multithread programming. The PC connects to a USB wireless adapter (XBee) for communicating with the RC cars. The architecture of the multicar control program is shown in Fig. 7. The controller for each RC car is independently implemented in a separate thread. These threads can also realize intervehicle communication.

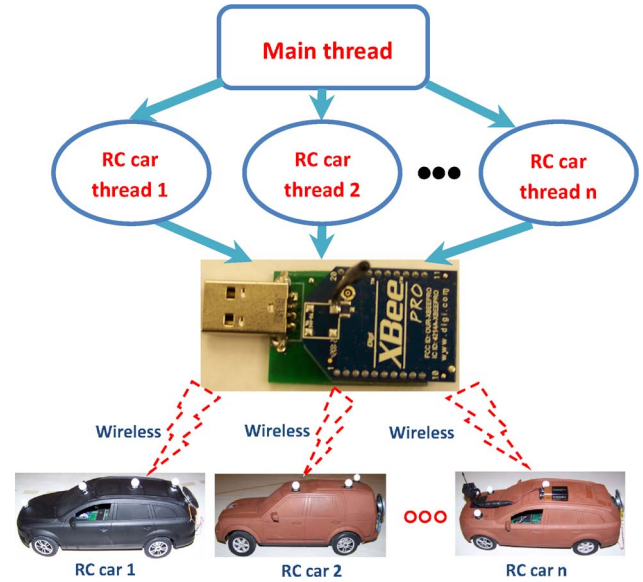


Fig. 7. Architecture of the multicar control program.

The problem of controlling a nonholonomic vehicle is well studied [22]–[25], but controlling a nonholonomic vehicle with low accuracy and nonsmooth velocity is a challenging problem. In this section, we first build a model of the RC car and then focus on developing efficient control algorithms for the RC car to track a predefined trajectory.

#### A. RC Car Model

As we know, most existing models of nonholonomic vehicles [22]–[24] are usually described as

$$\begin{cases} \dot{x} = v \cos(\Psi) \\ \dot{y} = v \sin(\Psi) \\ \dot{\Psi} = \omega \end{cases} \quad (1)$$

where  $\Psi$  is the orientation angle of the vehicle (see Fig. 8), and  $\omega$  is the angular velocity. The model in (1) is simple and does not consider the actual constraints on the range of steering angle and the sliding angle, which reflects the sliding errors between the center point of the car and the center point of the front axial.

Since the RC car has low accuracy on steering, we model it as

$$\begin{cases} \dot{x} = v \cos(\Psi + \theta + \beta) \\ \dot{y} = v \sin(\Psi + \theta + \beta) \\ \dot{\Psi} = \omega \end{cases} \quad (2)$$

where  $\theta$  is the steering angle of the front wheels (see Fig. 8), and  $\beta$  is the sliding angle that is obtained based on the center point of the car and velocity vector  $v$ . The  $\beta$  angle is computed as  $\beta = \Psi_c - \Psi$ ; here,  $\Psi_c$  is the heading of the vehicle at the center point.

#### B. RC Car Control Algorithm

To let the RC car track a predefined trajectory, we use a virtual-vehicle-based approach. The virtual vehicle is a reference point that is moving on the path we want the RC car to follow. The virtual vehicle  $s(t)$  is designed to move along the path with  $x_d = p(s)$  and  $y_d = q(s)$ . To track the virtual vehicle, two constraints are considered in the following inequalities (3)

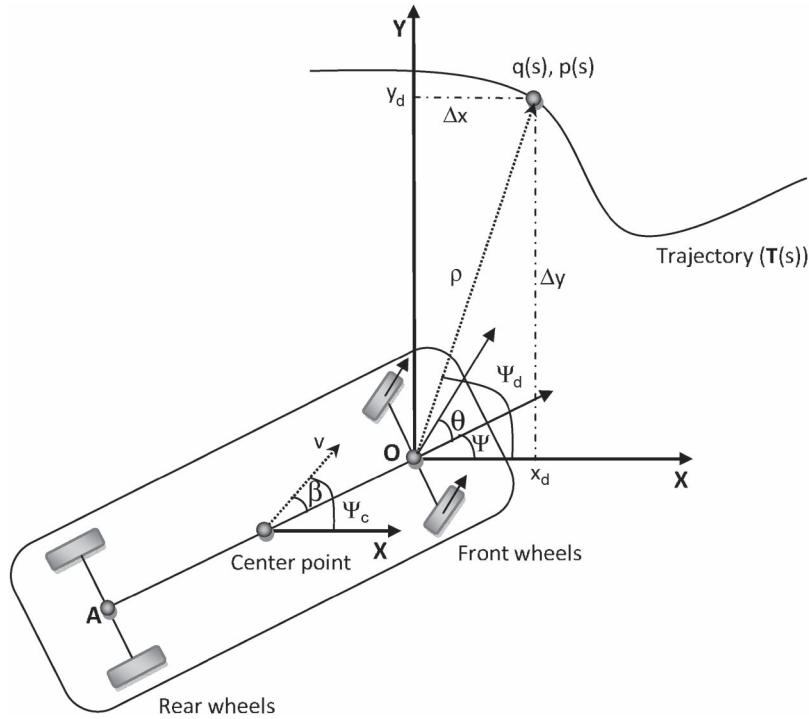


Fig. 8. RC car tracking the virtual vehicle (p(s), q(s)) moving in a predefined trajectory  $T(s)$ .

and (4), which are related to the difference between the actual heading and the desired heading of the RC car, and the distance between the actual and the virtual vehicle, respectively:

$$\lim (|\Psi(t) - \Psi_d(t)|)_{t \rightarrow \infty} \leq d_\Psi. \quad (3)$$

Here,  $\Psi_d$  is the desired angle, and  $d_\Psi$  is a small angle threshold

$$\lim (\rho(t))_{t \rightarrow \infty} \leq d_\rho. \quad (4)$$

Here,  $d_\rho$  is a small distance threshold.  $\rho(t)$  is the Euclidean distance between the RC car and the virtual vehicle (see Fig. 8). It is computed as

$$\rho(t) = \sqrt{\Delta x^2 + \Delta y^2}. \quad (5)$$

Here,  $\Delta x = x_d - x$ , and  $\Delta y = y_d - y$ .

To handle (3), the steering angle control for the RC car is based on proportional-derivative control (PD control) as follows:

$$\theta(t) = -k_p [\Psi(t) - \Psi_d(t)] - k_d [\dot{\Psi}(t) - \dot{\Psi}_d(t)].$$

Here,  $k_p$  and  $k_d$  are positive constants.

As mentioned before, although we replaced the front DC motor by a servo motor to obtain a controllable orientation and wider steering angles ( $20^\circ$ ), the left and right steering angles are not the same because the mechanical steering part is not rigid. Additionally, the velocity of the RC car is not smooth. The low-accuracy steering angle of the RC car is shown in Fig. 9. In this model, since the front wheels mounted on the car are not stable, the left steering angle range (see the top of Fig. 9) is different from the right steering wheel (see the bottom of Fig. 9). Specifically, in our experiment, we use an RC car that has  $\theta_L^{\max}$  between  $22^\circ$  and  $27^\circ$  and  $\theta_R^{\max}$  between  $15^\circ$  and  $20^\circ$ . This difference on the left and right steering angles

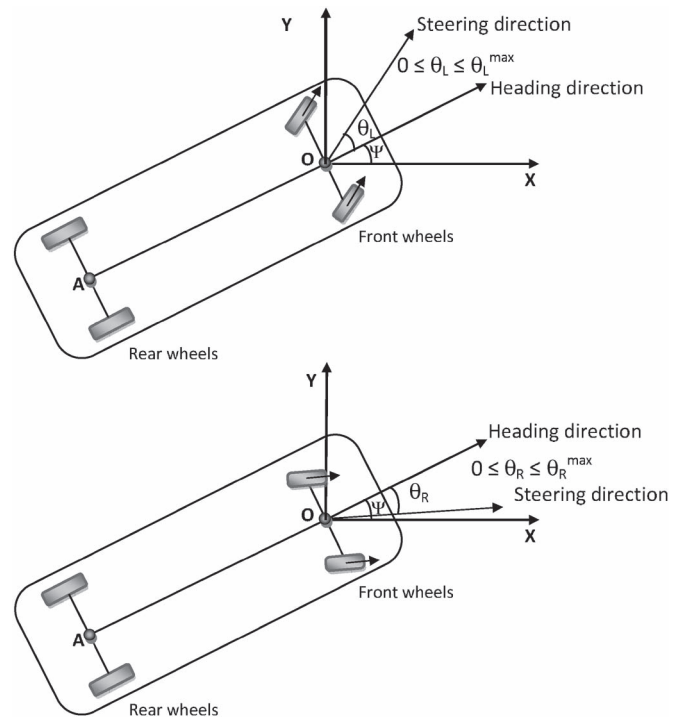


Fig. 9. Low-accuracy steering angle of the RC car. The left steering angle range is different from the right steering angle range.

is one of the reasons that make the car not able to track the predefined trajectories when applying any traditional tracking control algorithm.

Based on the aforementioned analysis, to handle the inequality (4), parameter  $\gamma$  is introduced [26], [27]

$$\dot{\rho} - \dot{d}_\rho = -\gamma(\rho - d_\rho). \quad (6)$$

Here,  $\gamma$  is a positive constant.

From (5), we can obtain

$$\begin{aligned}\dot{\rho} &= \frac{1}{\sqrt{\Delta x^2 + \Delta y^2}} (\Delta x \Delta \dot{x} + \Delta y \Delta \dot{y}) \\ &= \frac{1}{\rho} [\Delta x (\dot{x}_d - \dot{x}) + \Delta y (\dot{y}_d - \dot{y})].\end{aligned}\quad (7)$$

Here,  $\dot{x}_d = \dot{p}(s)\dot{s}$ , and  $\dot{y}_d = \dot{q}(s)\dot{s}$ .

Substituting  $\dot{\rho}$  from (7) into (6) while noticing that  $d_\rho$  is a constant,  $\dot{d}_\rho = 0$ , we have

$$\frac{1}{\rho} \dot{s} (\Delta x \dot{p}(s) + \Delta y \dot{q}(s)) = \frac{1}{\rho} (\Delta x \dot{x} + \Delta y \dot{y}) - \gamma (\rho - d_\rho) \quad (8)$$

or

$$\dot{s} = \frac{1}{\Delta x \dot{p}(s) + \Delta y \dot{q}(s)} [(\Delta x \dot{x} + \Delta y \dot{y}) - \gamma (\rho - d_\rho)]. \quad (9)$$

From (9), it is easy to see that, if  $\Delta x \dot{p}(s) + \Delta y \dot{q}(s) = 0$ , then  $\dot{s} \rightarrow \infty$ , or the car cannot track the virtual vehicle. In order to have  $\Delta x \dot{p}(s) + \Delta y \dot{q}(s) \neq 0$ , the car should stay close to and behind the virtual vehicle. From this analysis, we should make the virtual vehicle move with constant velocity at initial time when the car is far from it. This means that we need to have  $s(t) = s(t-1) + c$  ( $c$  is a positive constant) if  $t < t_{\text{threshold}}$ . The whole tracking control algorithm is shown in Algorithm 1.

---

**Algorithm 1:** The virtual vehicle tracking algorithm

---

**Initialization phase:**

- Create a trajectory of the virtual vehicle that the RC car wants to track.
- Initialize parameters:  $v, k_p, k_d, d_\rho, \gamma, s, \dot{s}, \Delta t$ .

**Implementation phase:**

**if**  $t < t_{\text{threshold}}$  **then**

Let the virtual vehicle move with constant velocity (to relax the assumption  $\Delta x \dot{p}(s) + \Delta y \dot{q}(s) = 0$ )

$$s(t) = s(t-1) + c.$$

Here,  $c$  is a positive constant.

**else**

- Compute the velocity of the virtual vehicle:

$$\dot{s}(t) = \frac{1}{\Delta x \dot{p}(s) + \Delta y \dot{q}(s)} (\Delta x \dot{x} + \Delta y \dot{y} - \gamma \rho (\rho - d_\rho)).$$

- Compute the steering angle for the RC car:

$$\theta(t) = -k_p [\Psi(t) - \Psi_d(t)] - k_d [\dot{\Psi}(t) - \dot{\Psi}_d(t)].$$

- Update the position of the virtual vehicle based on its velocity  $\dot{s}$ :

$$s(t) = s(t-1) + \dot{s}(t)\Delta t.$$

**end**

---

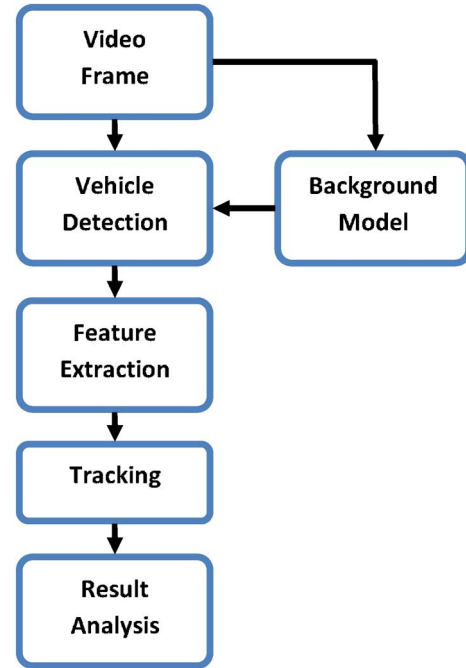


Fig. 10. Flowchart of the vision-tracking process.

## V. VISION-BASED MONITORING OF TRAFFIC

In this section, we present vision-based monitoring of traffic. Typical applications of vision-based traffic monitoring are vehicle detection, tracking, and anomaly detection. The Mobotix Q24 fish-eye camera is chosen to cover the entire arena. We set the camera at a resolution of  $800 \times 600$  pixels, which ensures real-time processing. Based on the live image streams received from the fish-eye camera over the Ethernet, the vision processing of detecting and tracking the RC cars is conducted in the PC with a processing speed of around 8 frame/s.

The flowchart of the vision-tracking algorithm is shown in Fig. 10. First, a threshold-based background model is applied to obtain foreground from image sequences for the purpose of detecting the motion regions of the cars. After the regions are detected, color features are extracted to label different cars, and then, a Kalman-filter-based tracking algorithm is applied to get a more accurate estimate of the state of the cars.

The system (state) model is defined as

$$\begin{cases} x(t) = x(t-1) + \Delta x(t-1) + \epsilon_x(t) \\ y(t) = y(t-1) + \Delta y(t-1) + \epsilon_y(t) \\ \Delta x(t) = \Delta x(t-1) + \zeta_x(t) \\ \Delta y(t) = \Delta y(t-1) + \zeta_y(t). \end{cases} \quad (10)$$

Here,  $(x(t), y(t))$  is the location of the car at time  $t$ , and  $\epsilon(t) = [\epsilon_x(t) \ \epsilon_y(t)]^T$  and  $\zeta(t) = [\zeta_x(t) \ \zeta_y(t)]^T$  are Gaussian and uncorrelated noises with zero mean at time  $t$ , respectively.

The observation model is defined as

$$\begin{cases} z_x(t) = x(t) + v_x(t) \\ z_y(t) = y(t) + v_y(t) \end{cases} \quad (11)$$

where  $(z_x(t), z_y(t))$  is the observation of the location of the car at time  $t$ , and  $v(t) = [v_x(t) \ v_y(t)]^T$  are the Gaussian and uncorrelated noises with zero mean at time  $t$ .

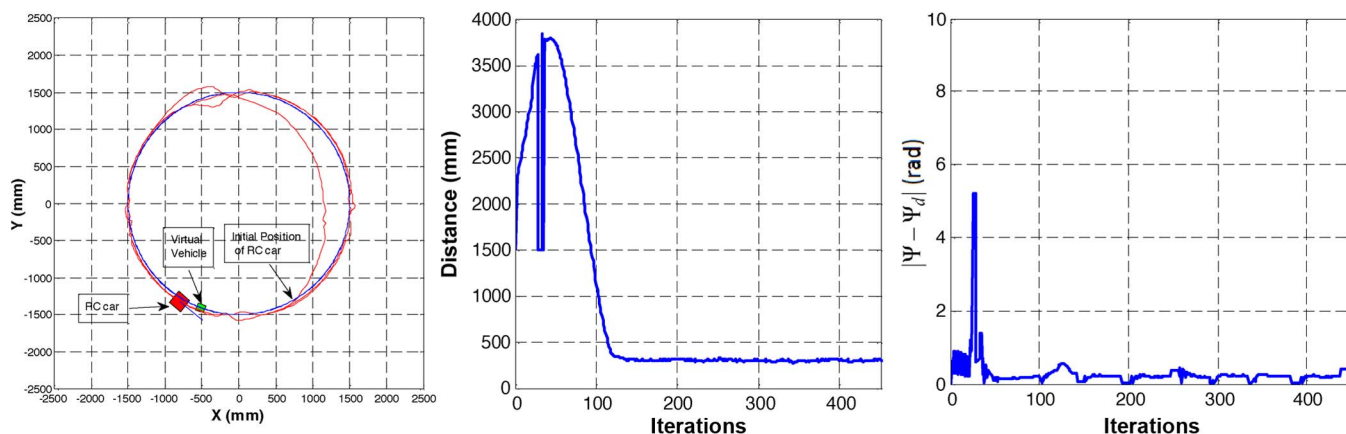


Fig. 11. (Left) Trajectories of the RC car tracking the virtual vehicle moving in circle. (Middle) Distance between the RC car and the virtual vehicle. (Right) Difference between the actual heading of the RC car and the desired one.

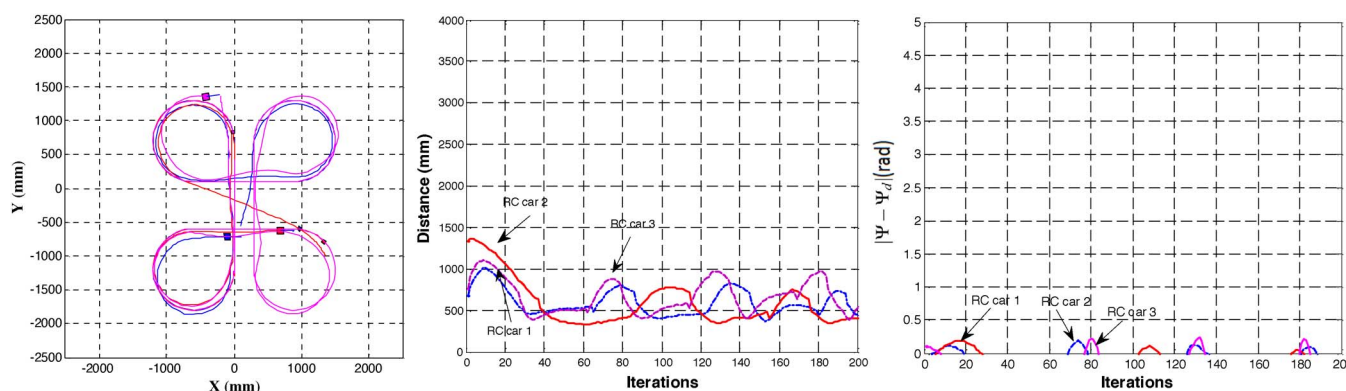


Fig. 12. (Left) Trajectories of the three RC cars tracking the virtual vehicles moving in desired trajectories. (Middle) Distance between the RC car and the virtual vehicle. (Right) Difference between the actual heading of the RC cars and the desired ones.

The vision information is obtained and saved, and it can be used to verify the trajectory and velocity information determined by the RC car control system. The vision algorithm is implemented in C/C++ and OpenCV [28].

## VI. EXPERIMENTAL RESULTS

In this section, we test our proposed control algorithm for a single RC car. Then, we extend the test to three RC cars based on multithread programming. We also provide the results of car tracking using the vision-tracking algorithm.

The parameters for RC control algorithm (see Algorithm 1) are given as follows: The desired distance between the RC car and the virtual vehicle  $d_p$  is 300 mm; the initial velocity of the virtual vehicle  $\dot{s}(0)$  is 0; the constants for the PD steering controller  $k_p = 1$ , and  $k_d = 0.8$ ; the constant  $\gamma$  for computing the velocity of the virtual vehicle is 2; and other parameters are  $v = 67$  and  $\Delta_t = 0.00056$ . The parameters of the virtual vehicle moving in a circle are given as follows:  $[x, y] = [R\cos(s), R\sin(s)]$  with its radius  $R = 1500$  mm.

The tracking results of Algorithm 1 are shown in Fig. 11. That is, the left side of Fig. 11 shows the RC car tracking the virtual vehicle, which moves in the circle trajectory. Fig. 11 (middle) shows the evaluation of the distance between the RC car and the virtual vehicle, and we can see that this distance converges to the predefined value of  $d_p = 300$  mm. Hence, this result meets our control goal in (4). In addition, we evaluate

the difference between the actual heading of the RC car and the desired one, as shown on the right side of Fig. 11. This result also satisfies the requirement in (3).

We also evaluate Algorithm 1 for three RC cars running through an intersection, as shown in Fig. 12. In this test, we design the trajectories that are more complicated than the circle trajectory. These trajectories have sharp changing points at the transition from circle trajectory to line trajectory. The left side of Fig. 12 shows the three RC cars tracking the three virtual vehicles, which move in the mock streets with an intersection. Fig. 12 (middle) shows the evaluations of the distance between the RC cars and the virtual vehicles. The right side of Fig. 12 shows the difference between the actual heading of the RC cars and the desired ones, respectively. Observing these figures, we can see that, at these sharp turning points on the trajectories, the tracking performance deteriorates.

For vehicle detection and tracking through the overhead Q24 camera, we first let two RC cars track the predefined trajectories based on Algorithm 1. One tracks the virtual vehicle moving in a circle trajectory, and the other moves in a figure “8” trajectory. Then, the vision-tracking algorithm is applied to track these two cars. The snapshots of the tracking result are shown in Fig. 13. The result of the circle tracking is plotted in red/black color, and the result of the figure “8” tracking is plotted in green/white color. The red/black squares represent the RC cars. From this figure, we can see that our vision-tracking algorithm can track multiple RC car with a reasonable accuracy. Video clips of the

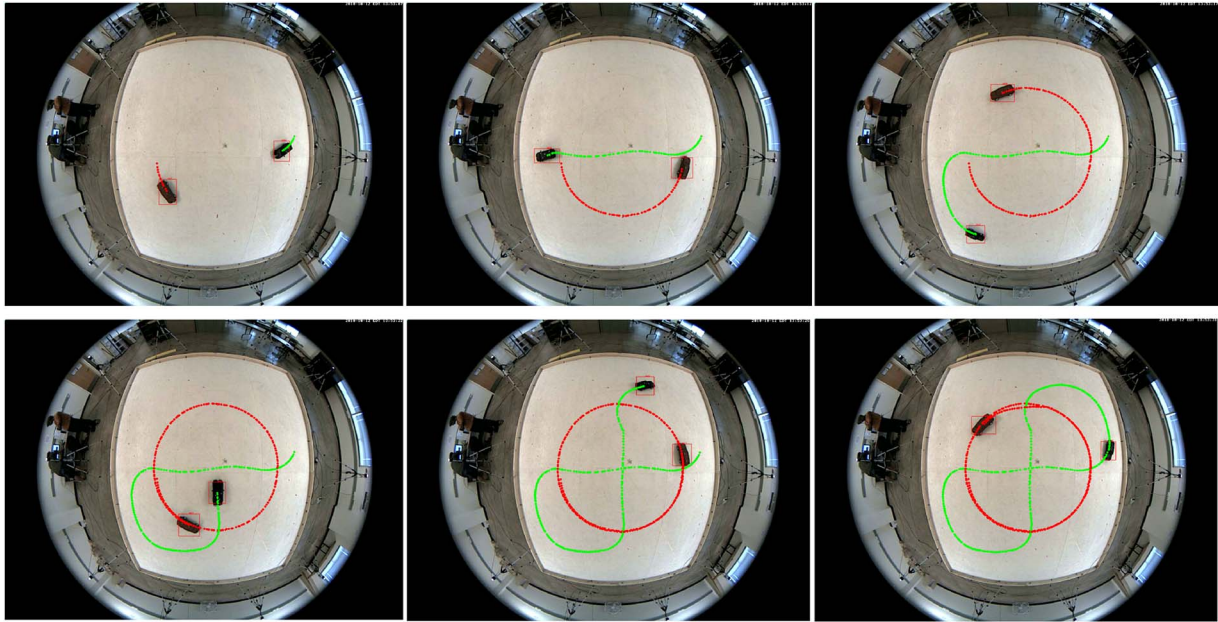


Fig. 13. Snapshots of vision-tracking algorithm.



Fig. 14. Platform has been upgraded to simulate more realistic street environments.

running of autonomous RC cars are available on our laboratory website at <http://ascc.okstate.edu/projectshung.html>

To make the platform better mimic real-world environments, we recently upgraded the platform by replacing the tripods with rails for mounting the motion capture system and adding intersections, sidewalks, model trees, model buildings, etc. A picture of the upgraded platform is shown in Fig. 14.

### VII. RESEARCH DIRECTIONS ON THIS PLATFORM

The developed platform can be used to investigate many problems in ITSs. Some examples are listed here.

#### A. Intelligent Traffic Monitoring

It is desirable to develop smart monitoring systems that can automatically detect anomalies in the traffic, e.g., detecting the collision of two cars at a street intersection, which makes it possible for automatic accident reporting to transportation authorities or calling for medical help. Using this testbed, we can

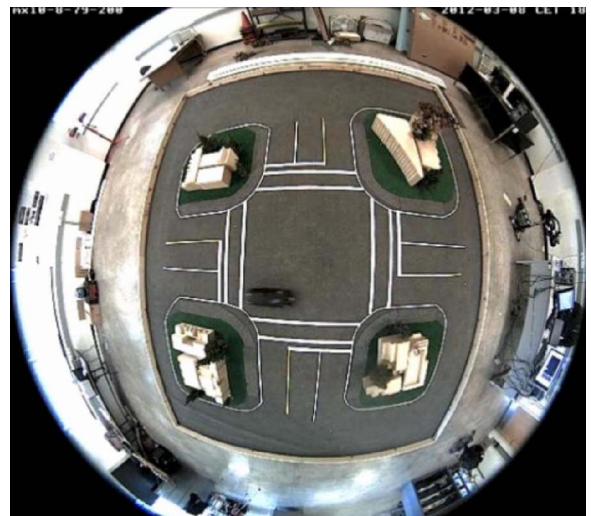


Fig. 15. Topdown view of the mock street, which can be used to detect traffic anomaly such as collisions and driving in wrong directions.

easily create different abnormal traffic scenarios (such as car collisions) that may be difficult to create using real vehicles due to the safety and cost concerns. The desired traffic scenarios can be easily realized through a script file detailing the movement of multiple RC cars. Fig. 15 shows a topdown view of the mock street while a car is driving through it. A recent project we are working on is to realize collision detection through audio/video fusion. A microphone array has been used to detect the angle of arrival of the collision sound, which is fused with the video data for improved detection accuracy and confidence.

#### B. Intelligent Vehicle Collision Avoidance

Collision avoidance is one of the main challenges in future ITSs. It is desirable to develop autonomous vehicles that are equipped with collision avoidance capability. Such a capability can be developed based on innovative control

algorithms, vehicle–vehicle communication, vehicle–roadside communication, etc. Our platform can be easily configured for such purposes. Due to the use of multithread programming, it is very straightforward to implement intervehicle vehicle–roadside communication while the vehicle control algorithms can be developed on the PC or the microcontroller in the cars. One of the authors' previous work [29] used a coordination diagram to enable multiple cars to pass through an intersection without collision. This research can be further extended and implemented in this new testbed.

### C. Vehicle Black Box

Black box has been typically used in airplanes to help accident investigation. Similarly, a vehicle “black box” can also benefit the investigation of vehicle accidents. Through the collection of important data regarding the state of the vehicle and the driver immediately before and during the accident, it is possible to reconstruct the whole procedure of the accident. Although the wide adoption of such vehicle black boxes may be a lengthy process due to many practical concerns, it is worthwhile to investigate such technologies and study its impact in future ITSSs. Our testbed can be easily configured to equip each car with a black box. For example, we can install wireless inertial measurement sensors in each vehicle to collect the motion data such as acceleration, angular velocity, and orientation. The indoor localization system can provide simulated GPS data, and the steering wheel can provide the driver's control input including wheel turning, gas pedal, brake and gear shift status, etc. By streaming all the data to a single computer, we can develop algorithms to accurately reconstruct the accident.

### D. Autonomous Driving in Mixed Traffic

Although we are confident that more and more autonomous vehicles will appear on our streets and roads, it is expected that the transition from the current human driving to future autonomous driving may take a very long time. Therefore, it will be of great interest to investigate autonomous car driving in mixed traffic with both human driving cars and autonomous driving cars. For an autonomous vehicle, it may be beneficial to model the human driving behaviors in developing its control algorithms. On the other hand, for a human driver, it may be helpful to understand the autonomous driving behavior in making his/her decisions for vehicle control. There will be many challenging and interesting problems we can study that crosscut human psychology, cognitive sciences, and engineering. Since we have both human driving cars and autonomous cars, our testbed can be used to conduct such preliminary research.

## VIII. CONCLUSION

This paper has proposed a small-scale research platform for ITSSs. Four main parts of the platform (an arena, an indoor localization system, automated RC cars, and roadside monitoring facilities) have been developed. The arena has been built to simulate the traffic environments. The Opti-Track system has been set up for localizing and controlling the RC cars. The hardware

has been designed and integrated in both autonomous driving and human driving RC cars. The fish-eye Q24 camera associated with the vision-tracking algorithm has been developed as an example of roadside monitoring facilities. Along with this new platform development, this paper has studied the problem of controlling a nonholonomic vehicle that has low steering accuracy and nonsmooth velocity. The tracking control algorithm for such a vehicle has been proposed. Experimental results have been collected to demonstrate the effectiveness of our proposed tracking control algorithm and the new platform design. We have also proposed research problems that can be investigated with this testbed. In the future, we will consider some adaptive and learning algorithms to automatically tune the controlled parameters to improve the tracking performances. We will also make this testbed available to remote users who want to access this testbed to collect the data for their research on ITSSs.

## REFERENCES

- [1] S. Hollborn, “Intelligent transport systems (ITS) in Japan,” Tech. Univ. Darmstadt, Darmstadt, Germany, Tech. Rep., 2002.
- [2] J. Njord, J. Peters, M. Freitas, B. Warner, K. C. Allred, R. Bertini, R. Bryant, R. Callan, M. Knopp, L. Knowlton, C. Lopez, and T. Warne, “Safety applications of intelligent transportation systems in Europe and Japan,” in *Proc. Int. Technol. Scanning Program*, 2006, pp. 1–54.
- [3] F. Y. Wang, “Building an intellectual highway for ITS research and development,” *IEEE Trans. Intell. Transp. Syst.*, vol. 11, no. 1, pp. 2–3, Mar. 2010.
- [4] F. Y. Wang, “Parallel control and management for intelligent transportation systems: Concepts, architectures, and applications,” *IEEE Trans. Intell. Transp. Syst.*, vol. 11, no. 3, pp. 630–638, Sep. 2010.
- [5] J. Zhang, F. Y. Wang, K. Wang, W. H. Lin, X. Xu, and C. Chen, “Data-driven intelligent transportation systems: A survey,” *IEEE Trans. Intell. Transp. Syst.*, vol. 12, no. 4, pp. 1624–1639, Dec. 2011.
- [6] L. Li, X. Li, Z. Li, D. D. Zeng, and W. T. Scherer, “A bibliographic analysis of the IEEE transactions on intelligent transportation systems literature,” *IEEE Trans. Intell. Transp. Syst.*, vol. 11, no. 2, pp. 251–255, Jun. 2010.
- [7] O. Andrisano, R. Verdone, and M. Nakagawa, “Intelligent transportation systems: The role of third generation mobile radio networks,” *IEEE Commun. Mag.*, vol. 38, no. 9, pp. 144–151, Sep. 2000.
- [8] D. Chuan-Fei, M. Xu, and W. Bing-Hong, “Weighted congestion coefficient feedback in intelligent transportation systems,” *Phys. Lett. A*, vol. 374, no. 11/12, pp. 1326–1331, Mar. 2010.
- [9] Y. Zhao, “Mobile phone location determination and its impact on intelligent transportation systems,” *IEEE Trans. Intell. Transp. Syst.*, vol. 1, no. 1, pp. 55–64, Mar. 2000.
- [10] T. G. Crainic, M. Gendreau, and J.-Y. Potvin, “Intelligent freight-transportation systems: Assessment and the contribution of operations research,” *Transp. Res. Part C, Emerging Technol.*, vol. 17, no. 6, pp. 541–557, Dec. 2009.
- [11] L. Figueiredo, I. Jesus, J. A. T. Machado, J. R. Ferreira, and J. L. Martins de Carvalho, “Towards the development of intelligent transportation systems,” in *Proc. IEEE Intell. Transp. Syst. Conf.*, 2001, pp. 1206–1211.
- [12] M. Tubaishat, P. Zhuang, Q. Qi, and Y. Shang, “Wireless sensor networks in intelligent transportation systems,” *Wirel. Commun. Mobile Comput.*, vol. 9, no. 3, pp. 287–302, Mar. 2009.
- [13] Google, “Autonomous driving,” *The New York Times*, Oct. 10, 2010.
- [14] H. Kress-Gazit, D. C. Conner, H. Choset, A. Rizzi, and G. J. Pappas, “Courteous cars,” *IEEE Robot. Autom. Mag.*, vol. 15, no. 1, pp. 30–38, Mar. 2008.
- [15] *Vehicle Platooning and Automated Highways*, California PATH, Richmond, CA, 2010, pp. 1–4.
- [16] D. Schleicher, L. M. Bergasa, M. Ocaña, R. Barea, and M. E. López, “Real-time hierarchical outdoor SLAM based on stereovision and GPS fusion,” *IEEE Trans. Intell. Transp. Syst.*, vol. 10, no. 3, pp. 440–452, Sep. 2009.
- [17] J. R. Sayer, S. E. Bogard, D. Funkhouser, D. J. LeBlanc, S. Bao, A. D. Blankespoor, M. L. Buonarosca, and C. B. Winkler, “Integrated vehicle-based safety systems, heavy-truck field operational test key findings report,” Univ. Michigan Transp. Res. Inst., Ann Arbor, MI, Tech. Rep. Dept. Transp. HS 811 482, 2010.

- [18] Integrated Vehicle-Based Safety Systems (IVBSS). [Online]. Available: <http://www.its.dot.gov/ivbss/>
- [19] PATH project, Mar. 2011. [Online]. Available: <http://www.path.berkeley.edu/>
- [20] S. Panwai and H. Dia, "Comparative evaluation of microscopic car-following behavior," *IEEE Trans. Intell. Transp. Syst.*, vol. 6, no. 3, pp. 314–325, Sep. 2005.
- [21] [Online]. Available: <http://www.naturalpoint.com/optitrack/>
- [22] C. Canudas de Wit and O. J. Sordalen, "Exponential stabilization of mobile robots with nonholonomic constraints," *IEEE Trans. Autom. Control*, vol. 11, no. 11, pp. 1791–1797, 1992.
- [23] G. Indiveri, "Kinematic time-invariant control of a 2d nonholonomic vehicle," in *Proc. IEEE CDC*, Phoenix, AZ, 1999, pp. 2112–2117.
- [24] Y. Zhu and U. Ozguner, "Constrained model predictive control for non-holonomic vehicle regulation problem," in *Proc. 17th World Congr., Int. Fed. Autom. Control*, Seoul, Korea, 2008, pp. 9552–9557.
- [25] J. Cochran and M. Krstic, "Nonholonomic source seeking with tuning of angular velocity," *IEEE Trans. Autom. Control*, vol. 54, no. 4, pp. 717–731, Apr. 2009.
- [26] M. Egerstedt, X. Hu, and A. Stotsky, "Control of mobile platforms using a virtual vehicle approach," *IEEE Trans. Autom. Control*, vol. 46, no. 11, pp. 1777–1782, Nov. 2001.
- [27] S. V. Gusev and I. A. Makarov, "Stabilization of program motion of transport robot with track laying chassis," *Proc. LSU*, vol. 1, no. 3, pp. 1–15, 1989.
- [28] G. Bradski and A. Kaehler, *Learning Open CV*, 1st ed. Sebastopol, CA: O'Reilly, 2008.
- [29] W. Sheng, Q. Yang, and Y. Guo, "Cooperative driving based on inter-vehicle communications: Experimental platform and algorithm," in *Proc. IEEE/RSJ Int. Conf. Intelligent Robots Syst.*, 2006, pp. 5073–5078.



**Hung Manh La** received the M.S. and B.S. degrees in electrical engineering from Thai Nguyen University of Technology, Thai Nguyen, Vietnam, in 2003 and 2001, respectively, and the Ph.D. degree in electrical and computer engineering from Oklahoma State University, Stillwater, in 2011.

He is currently a Postdoc with the Center for Advanced Infrastructure and Transportation, Rutgers University, Piscataway, NJ, and a Lecturer with the Department of Electronics Engineering, Thai Nguyen University of Technology. He has been

actively involved in research projects with the Federal Highway Administration, Department of Transportation, Department of Defense, and National Science Foundation. His research interests include mobile robots, mobile sensor networks, cooperative control, learning and sensing, and intelligent transportation systems.

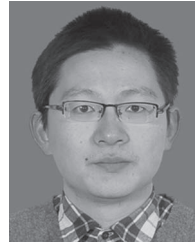
Dr. La received two Best Paper Awards at the 2009 and 2010 Conferences on Theoretical and Applied Computer Science, respectively, and one Best Presentation of Session Award at the 2009 American Control Conference.



**Ronny Salim Lim** received the Bachelor's degree from Pelita Harapan University, Tangerang, Indonesia, in 2006 and the M.S. degree in electrical and computer engineering from Oklahoma State University, Stillwater, in 2011.

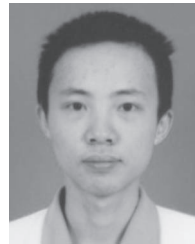
He is currently an Application Developer with the Center for Advanced Infrastructure and Transportation, Rutgers University, Piscataway, NJ. He was a Research Assistant with the Laboratory for Advanced Sensing, Computation and Control, where he conducted research on robotics crack inspection,

intelligent transportation systems, and multiagent control.



**Jianhao Du** received the M.S. and B.S. degrees in electrical engineering from Zhejiang University, Hangzhou, China, in 2007 and 2010, respectively. He is currently working toward the Ph.D. degree with the School of Electrical and Computer Engineering, Oklahoma State University, Stillwater.

His research interests include computer vision, camera networks, and mobile robotics.



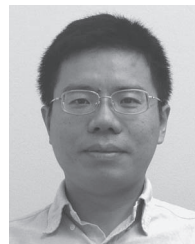
**Sijian Zhang** received the B.S. degree in electrical engineering, in 2007, from Zhejiang University, Hangzhou, China, where he is currently working toward the Ph.D. degree in control theory and control engineering.

His current research interests include sensing and wireless sensor networks.



**Gangfeng Yan** received the B.S. and M.S. degrees in control theory and control engineering from Zhejiang University, Hangzhou, China, in 1981 and 1984, respectively.

He is currently a Professor with the Department of Systems Science and Engineering, Zhejiang University. His research interests include hybrid systems, neural networks, and cooperative control.



**Weihua Sheng** (SM'08) received the B.S. and M.S. degrees in electrical engineering from Zhejiang University, Hangzhou, China, in 1994 and 1997, respectively, and the Ph.D. degree in electrical and computer engineering from Michigan State University, East Lansing, in 2002.

He is currently an Assistant Professor with the School of Electrical and Computer Engineering, Oklahoma State University, Stillwater. During 1997–1998, he was a Research Engineer with the R&D Center, Huawei Technologies Co., China. During

2002–2006, he taught with the Department of Electrical and Computer Engineering, Kettering University (formerly General Motor Institute), Flint, MI. He was promoted to Associate Professor there before he joined Oklahoma State University. He has participated in organizing several IEEE international conferences and workshops in the area of intelligent robots and systems. He is the author of more than 110 papers in major journals and international conference proceedings. He is the holder of one U.S. patent. His research is supported by the National Science Foundation, Department of Defense, Defense Experimental Program to Stimulate Competitive Research, Department of Transportation, etc. His current research interests include wearable computing, mobile robotics, human robot interaction, and intelligent transportation systems.

Dr. Sheng is currently an Associate Editor for IEEE TRANSACTIONS ON AUTOMATION SCIENCE AND ENGINEERING.