

Semiautonomous Vehicular Control Using Driver Modeling

Victor A. Shia, Yiqi Gao, Ramanarayan Vasudevan, *Member, IEEE*, Katherine Driggs Campbell, Theresa Lin, Francesco Borrelli, and Ruzena Bajcsy, *Fellow, IEEE*

Abstract—Threat assessment during semiautonomous driving is used to determine when correcting a driver's input is required. Since current semiautonomous systems perform threat assessment by predicting a vehicle's future state while treating the driver's input as a disturbance, autonomous controller intervention is limited to a restricted regime. Improving vehicle safety demands threat assessment that occurs over longer prediction horizons wherein a driver cannot be treated as a malicious agent. In this paper, we describe a real-time semiautonomous system that utilizes empirical observations of a driver's pose to inform an autonomous controller that corrects a driver's input when possible in a safe manner. We measure the performance of our system using several metrics that evaluate the informativeness of the prediction and the utility of the intervention procedure. A multisubject driving experiment illustrates the usefulness, with respect to these metrics, of incorporating the driver's pose while designing a semiautonomous system.

Index Terms—Intelligent vehicles, nonlinear control systems, predictive control, vehicle safety.

I. INTRODUCTION

DESPITE the development of numerous vehicular safety features, the number of traffic accidents and fatalities remains in the several millions and hundreds of thousands, respectively [1]. Although it is difficult to determine exactly why these accidents occur, there is considerable evidence that in-vehicle information systems and devices, such as cell phones, make the driver prone to distraction, which degrades driving performance. Several studies have estimated, using data from vehicles instrumented with cameras, that driver distraction contributes to between 22% and 50% of all crashes [2], [3]. Other studies have verified that driver distraction due to cell phones is correlated with an inability to maintain central lane

position and led to lower overall response time [4]. Alarming, a recent report has indicated that the percentage of fatal crashes due to distraction is rising [5].

A potential resolution to the driver distraction problem is the autonomous vehicle [6]–[9]. Although autonomous systems have garnered much of the public's attention, the deployment of these systems has been hindered due to the lack of formal methods that are capable of verifying the safety of such systems in arbitrary situations. Recognizing this shortcoming of theoretical verification algorithms in the face of a growing number of accidents due to driver distraction, several car companies have begun designing active safety systems that operate only over short time horizons in order to avoid accidents. These systems rely on threat assessment to determine when control intervention that corrects the driver's input is required. Volvo, for example, has developed one of the more advanced active safety systems, which works at low speeds to detect potential accidents by treating the driver's input as a disturbance, computing the reach set of the car over a short time horizon (approximately 500 ms), and warns the driver if it detects an intersection of the reach set with an obstacle [10]. If the driver does not react in time, the system will automatically brake to avoid or mitigate an accident. Confidence in this autonomous breaking maneuver and its threat assessment system that treats a driver's input as a disturbance was achieved via extensive testing over several million kilometers. In these types of semiautonomous active safety system, the driver assumes control for the majority of the time, with the control system acting as the copilot.

As these semiautonomous systems perform more complicated maneuvers at higher speeds, threat assessment must occur over a longer time horizon. In this context, treating the driver as just a disturbance results in an exorbitantly sized reach set, which results in a semiautonomous system that begins to behave like a fully autonomous one. Therefore, as this shift occurs, the driver's behavior becomes fundamental during the design of a provably safe semiautonomous vehicle. Importantly, a single universal driver model for threat assessment is insufficient. Instead, a framework that adapts to different drivers by exploiting real-time empirical observations of the driver is critical. Empirical validation becomes difficult due to the reliance of the system on the driver model itself. Fortunately, recent advances in numerical optimization have shown that provably safe controllers can be designed rapidly and robustly given a particular circumstance [11], [12]. In this paper, we extend the theoretical framework that we described in [13], by constructing a real-time system that utilizes empirical observations of a driver to keep a vehicle safe.

Manuscript received July 26, 2013; revised January 9, 2014 and April 13, 2014; accepted May 3, 2014. Date of publication June 10, 2014; date of current version December 1, 2014. This material is based upon work supported by the National Science Foundation under Awards ECCS-0931437, ECCS-1239323, and DGE-0903711 and by the Hyundai Center for Excellence, University of California at Berkeley. The Associate Editor for this paper was S. A. Wadaw.

V. A. Shia, K. D. Campbell, and R. Bajcsy are with the Department of Electrical Engineering and Computer Sciences, University of California at Berkeley, Berkeley, CA 94720 USA (e-mail: vshia@eecs.berkeley.edu; krde@eecs.berkeley.edu; bajcsy@eecs.berkeley.edu).

Y. Gao, T. Lin, and F. Borrelli are with the Department of Mechanical Engineering, University of California at Berkeley, Berkeley, CA 94720 USA (e-mail: yiqigao@berkeley.edu; tlin051@berkeley.edu; fborrelli@berkeley.edu).

R. Vasudevan is with the Computer Science and Artificial Intelligence Laboratory, Massachusetts Institute of Technology, Cambridge, MA 02139 USA (e-mail: ramv@csail.mit.edu).

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.2014.2325776

A. Driver Modeling and Prediction

Monitoring and modeling a driver's behavior has been considered by different communities. The human factors community has focused on quantifying the amount of attention required to accomplish a variety of tasks, and the level of distraction due to common preoccupations that arise during driving [14], [15]. **These studies, however, do not provide a method to predict the driver's behavior given a particular level of distraction and task complexity.**

Other communities have used body sensors on the driver [16], eye trackers [17], facial tracking [18], [19], or other sensors [20]–[22] to predict driver behavior to provide warnings of impending danger. **Although these methods predict driver intent with 70%–90% accuracy over varying prediction horizons, they do not describe how to employ this prediction during control design or how to quantify the utility of a given prediction, with respect to preventing accidents [23].**

Trajectory prediction models, on the other hand, restrict themselves to specific maneuvers [24], [25] or scenarios [26] and do not consider the driver state [27], [28]. Existing methods break the trajectory into two components for prediction: **speed and steering.** Oftentimes, the high-level driver behavior is linked with hybrid or hidden Markov models to predict driver intentions [29], [30] and incorporate human factors such as comfort and skill level [24], and even day-to-day experiences [31]. **However, the actual driver behavior is never incorporated in the control framework.**

B. Paper Contributions and Structure

In the interest of thoroughness, Section II describes the **theoretical framework**, and Section II-C describes several **metrics to evaluate the performance of the theoretical framework**, which was originally described in [13]. Our contributions in this paper are organized as follows: Section II describes a new component to the theoretical framework that designs an autonomous controller to render a vehicle safe. Section III describes **the implementation of the driver and vehicle model and autonomous controller design.** In Section IV, we construct a real-time hardware-and-driver-in-the-loop experiment to illustrate the utility of incorporating empirical observations of the driver in the control loop to prevent accidents. In Section V, we present our experimental results and empirically validate our proposed architecture's ability to prevent accidents, and Section VI concludes the paper.

II. METHODOLOGY

Here, we present a semiautonomous real-time vehicle framework that relies on driver modeling. A specific instantiation of this framework is described in subsequent sections.

A. Semiautonomous Vehicle Framework

The framework controls the evolution of a vehicle, which is described as the solution to the *vehicle dynamics* as

$$x(k+1) = f(x(k), u(k)) \quad \forall k \in \mathbb{N} \quad (1)$$

where $x(k) \in \mathbb{R}^n$ is the state of the vehicle at time k ; $u(k) \in U$ is the input into the vehicle dynamics at time k , where $U \subset \mathbb{R}^m$ is a compact and connected set containing the origin; and $x(0)$ is assumed given. In addition, for a fixed horizon $\bar{N} \in \mathbb{N}$, let $\mathcal{U}_{\bar{N}} = \{\{u(k)\}_{k=0}^{\bar{N}} \mid u(k) \in U, \forall k \in \{0, \dots, \bar{N}\}\}$. The **constraint function**, i.e., $\varphi_{\bar{N}}: \mathbb{R}^n \times \{0, \dots, \bar{N}\} \rightarrow \mathbb{R}$, describes the safe part of the environment within which the vehicle travels. **That is, the state of the car remains outside of obstacles in the environment at a time k if $\varphi_{\bar{N}}(x(k), k) \leq 0$.** Observe that if we know the location of the obstacles at $k=0$ but are only aware of bounds on the movement of each obstacle, then we can still encapsulate these bounds within the constraint function $\varphi_{\bar{N}}$.

We measure the driver's state by using observations of his motion inside of the vehicle. Several recent advances in **computer vision** have made possible the real-time robust articulated tracking of a human [32]. Employing these latest insights, we presume the existence of a system capable of articulated tracking in our vehicle and describe a specific implementation in Section IV.

Assumption 1: Given $\underline{N} \in \mathbb{N}$ and $N > 0$ joints (e.g., elbows or knees), we assume there exists a function $\theta_{\underline{N}}: \{-\underline{N}, \dots, 0\} \rightarrow \mathbb{R}^{3N}$ called the **driver state function**, which describes the evolution of the positions of the joints of the driver.

The driver state function $\theta_{\underline{N}}$ describes how the driver has moved in the previous $\underline{N} + 1$ time steps.

B. Problem Formulation

Our semiautonomous vehicle framework divides the problem of determining when to intervene into **three components**. To appreciate our proposed architecture, consider the “simplest” semiautonomous architecture that **treats the driver's input as a disturbance, constructs the reach set of the vehicle, and intervenes in the operation of the vehicle whenever the reach set of the vehicle intersects with an obstacle and whenever safe intervention is feasible.** Unfortunately, by treating the driver as a disturbance, the simplest architecture predicts a large potential set of behavior and therefore acts too often. That is, suppose, for example, that the semiautonomous controller employed a breaking maneuver and did not want to decelerate the vehicle quickly. In this instance, the controller would need to act over a long time horizon (e.g., more than 2 s). The set of reachable states at highway speeds for a 2-s time horizon is large and, in all likelihood, would intersect with some obstacle all the time (e.g., a vehicle in a neighboring lane), although the driver would most likely never have an accident with this obstacle.

The first component of the proposed semiautonomous vehicle framework addresses this deficiency by **designing a control algorithm, which incorporates the prior observations denoted by \mathcal{O} (this may include past states of the driver, past vehicle trajectories, and other sensor data) and the current information $\mathcal{I} = \{\bar{N}, \underline{N}, f, x(0), \varphi_{\bar{N}}, \mathcal{U}_{\bar{N}}, \theta_{\underline{N}}\}$, in order to predict a smaller and more useful set of potential states for the vehicle and the future expected steering.** This component is needed as the poor performance of not incorporating observations and future predictions of the driver was shown in [13].

The component is defined formally as follows.

Component 1: Fixing \bar{N} , $\underline{N} \in \mathbb{N}$ and choosing $\alpha \in [0, 1]$, construct the **vehicle prediction multifunction (VPM)** denoted by $\Delta(\alpha, k, \mathcal{O}, \mathcal{I}) \subset 2^{\mathbb{R}^n}$ as the solution to

$$\begin{aligned} \Delta(\alpha, k, \mathcal{O}, \mathcal{I}) &= \arg \min_{\Omega \subset \mathbb{R}^n} |\Omega| \\ \text{s.t. } P((X(k) - x(0)) \in \Omega | \mathcal{O}, \mathcal{I}) &\geq \alpha, \\ \forall k \in \{0, \dots, \bar{N}\} \end{aligned} \quad (2)$$

where $|\Omega|$ denotes the area of the set Ω ; $P(A|B)$ is the conditional probability of A given B ; and $P((X(k) - x(0)) \in \Omega | \mathcal{O}, \mathcal{I})$ denotes the probability that the difference between a solution to (1) beginning at $x(0)$ in k time steps, denoted by $X(k)$, and $x(0)$ is inside of the set Ω , given the prior observations and prior information (note that $X(k)$ is capitalized, in order to distinguish it as the random variable).

The VPM is the smallest size set that contains all α probable vehicle trajectories in k time steps, given prior observations and current information. In order to construct this function, we require a model linking $X(k)$ with prior observations. We assume that the minimizer of (2) exists and is unique. In Section III-C, we describe how we construct the required conditional distribution and how we construct the VPM.

Using the VPM and the constraint function, the second component of the semiautonomous vehicle framework is a binary-valued function that is defined formally as follows.

Component 2: The vehicle intervention function (VIF) denoted by g is 1, when the driver requires aid, and is defined as

$$g(\alpha, \mathcal{O}, \mathcal{I}) = \begin{cases} 1, & \text{if } \bigcup_{k=0}^{\bar{N}} (\Pi(\Delta(\alpha, k, \mathcal{O}, \mathcal{I})) \cap \mathcal{C}(k, \mathcal{I})) \neq \emptyset \\ 0, & \text{otherwise} \end{cases} \quad (3)$$

where $\mathcal{C}(k, \mathcal{I}) = \{z \in \mathbb{R}^n | \varphi_{\underline{N}}(z - x(0), k) > 0\}$ and $\Pi : 2^{\mathbb{R}^n} \rightarrow 2^{\mathbb{R}^n}$. Π is a projection operator that allows for the selection of certain trajectories if required. The VIF is 1 if the projection Π of the α probable vehicle trajectory intersects an obstacle in the next $\bar{N} + 1$ time steps, given our past observations and current information. In Section III-C, we describe how we define the VIF.

When intervention is required, the third component of the semiautonomous vehicle framework computes an input to the vehicle for the next $\bar{N} + 1$ time steps that renders the vehicle safe, as follows.

Component 3: The autonomous controller, denoted by $\lambda : \mathbb{R}^n \times \mathbb{R}^n \rightarrow \mathcal{U}_{\bar{N}} \cup \emptyset$, determines an input to the vehicle for the next $\bar{N} + 1$ time steps and is defined as

$$\delta u = \lambda(\Pi(\Delta(\alpha, k, \mathcal{O}, \mathcal{I})), \mathcal{C}(k, \mathcal{I})) \quad (4)$$

where δu can be either an augmented input to the driver's steering or the actual steering for fully autonomous control. If the autonomous controller is unable to construct a control that renders the vehicle safe over the horizon \bar{N} , then δu returns \emptyset .

In Section III-D, we describe how we construct the autonomous controller.

Algorithm 1 describes our semiautonomous vehicle framework. Notice that, since we do not demand the VPM to perfectly predict the potential behavior of the vehicle, the driver may get into an accident because the semiautonomous vehicle framework did not intervene. This means that the safety of the semiautonomous vehicle framework is guaranteed in the sense that, when the semiautonomous controller intervenes, it is able to do so in a provably safe way under the environment model prescribed by $\psi_{\bar{N}}$.

Algorithm 1 Semiautonomous Vehicle Framework

- 1: Data: \bar{N} , $\underline{N} \in \mathbb{N}$ and $x(0) \in \mathbb{R}^n$
 - 2: Let $u(k) = 0, \forall k \in \{0, \dots, \bar{N}\}$.
 - 3: **for** Each time step **do**
 - 4: update $\varphi_{\bar{N}}$ and $\theta_{\bar{N}}$, and set $x(0)$ equal to the vehicle's current state.
 - 5: **if** $g(\alpha, \mathcal{O}, \mathcal{I}) == 1$ and $\delta u \neq \emptyset$ **then**
 - 6: apply $\delta u = \lambda(\Pi(\Delta(\alpha, k, \mathcal{O}, \mathcal{I})), \mathcal{C}(k, \mathcal{I}))$ to the vehicle.
 - 7: **end if**
 - 8: **end for**
-

C. Metrics to Evaluate Performance

To evaluate the performance of our VPM and the VIF, we utilize four metrics first described in [13], which we briefly summarize in the following. The most informative set prescribed by the VPM would predict a single vehicle trajectory that was always accurate. Since this would demand perfectly deciphering the intent of the driver, we instead forecast a set of potential trajectories. During our experiment in Section IV, we measure the informativeness of our prediction by first determining if the observed vehicle behavior for the next $\bar{N} + 1$ time steps was inside of the predicted set at each of the next $\bar{N} + 1$ time steps; we call this the accuracy of the VPM (AVPM). Explicitly, fixing $\alpha \in [0, 1]$, suppose that we are given M distinct observed vehicle trajectories for $\bar{N} + 1$ time steps beginning at time step i , denoted by $\{x_{i,\bar{N}} : \{0, \dots, \bar{N}\} \rightarrow \mathbb{R}^n\}_{i=1}^M$, then

$$\begin{aligned} \text{AVPM} &= \frac{1}{M} \sum_{i=1}^M \prod_{k=0}^{\bar{N}} \mathbb{1} \\ &\times \left\{ (x_{i,\bar{N}}(k) - x_{i,\bar{N}}(0)) \in \Delta(\alpha, k, \mathcal{O}, \mathcal{I}_i) \right\} \end{aligned} \quad (5)$$

where $\mathbb{1}$ is the indicator function, and the current information \mathcal{I}_i is indexed by the time step i since the VPM is a function of the initial condition of the vehicle $x_{i,\bar{N}}(0)$. We also measure the precision of the predicted set, which is called the precision of the VPM (PVPM), by computing one minus the area of the predicted set for the next $\bar{N} + 1$ time steps over the area of the set of reachable states for the next $\bar{N} + 1$ time steps, as

$$\text{PVPM} = 1 - \frac{1}{M} \sum_{i=1}^M \frac{\left| \bigcup_{k=0}^{\bar{N}} \Delta(\alpha, k, \mathcal{O}, \mathcal{I}_i) \right|}{\left| \bigcup_{k=0}^{\bar{N}} \mathcal{R}(k, \mathcal{I}_i) \right|} \quad (6)$$

where $\mathcal{R}(k, \mathcal{I}_i)$ is the set of reachable states at time step k beginning at $x_{i, \overline{N}}(0)$. We normalize by the set of reachable states since it is always accurate, but imprecise because it assumes that the vehicle behaves arbitrarily.

Similarly, the most useful VIF would only act if and when the vehicle was actually going to get into an accident. Unfortunately, the determination of this most useful intervention procedure is difficult since it demands being clairvoyant. To understand these concepts in this context, we borrow several definitions from the binary classification literature [33]. We define an observed $\overline{N} + 1$ time step vehicle trajectory to be “true” if an accident is actually observed at any instance in the next $\overline{N} + 1$ time steps, and we define an observed $\overline{N} + 1$ time step vehicle trajectory to be “false” if an accident is not observed in any of the next $\overline{N} + 1$ time steps. We define an observed vehicle trajectory to be “positive” if the VIF (i.e., the classifier in our context) returns 1, and we define an observed vehicle trajectory to be “negative” if the VIF returns 0.

The recall of the VIF (RVIF) is then defined as the true-positive observations divided by the sum of the true-positive observations and false-negative observations (note that, in the binary classification literature, this is when the observation is true, but the classifier returns negative) and measures the sensitivity of the vehicle detection function. *If the RVIF is 1, then the semiautonomous vehicle framework intervened whenever an accident occurred.* Notice that if, instead of relying on the VPM, the VIF relied on the set of reachable states, then there would be no false-negative observations since an accident could have only occurred if there was a nontrivial intersection between the reachable set (RS) and an obstacle. Therefore, this choice of VIF would always be perfect, with respect to this metric.

The precision of the VIF (PVIF) is defined as the true-positive observations divided by the sum of the true-positive observations and false-positive observations (note that, in the binary classification literature, this is when the observation is false, but the classifier returns positive). *If the PVIF is much less than 1, then the semiautonomous vehicle framework intervened far more than was actually required.* If this is the case, the vehicle begins to behave more like an autonomous vehicle. Suppose again that, instead of relying on the VPM, the VIF relied on the set of reachable states, then there would most likely be a large number of false-positive events since the RS would intersect with some obstacle in the neighboring lanes that the driver, in all likelihood, would never have an accident with. Our goal in the remainder of this paper is to illustrate the informativeness and utility of considering prior observations before the current time step, by using these four metrics.

III. SYSTEM ARCHITECTURE

Here, we describe the vehicle model and specific instantiations of the semiautonomous architecture proposed in the previous section used in the experiment, which we describe in the next section.

A. Vehicle Modeling

Here, we present the vehicle dynamics illustrated in Fig. 1 used during control design. We use the following set of differ-

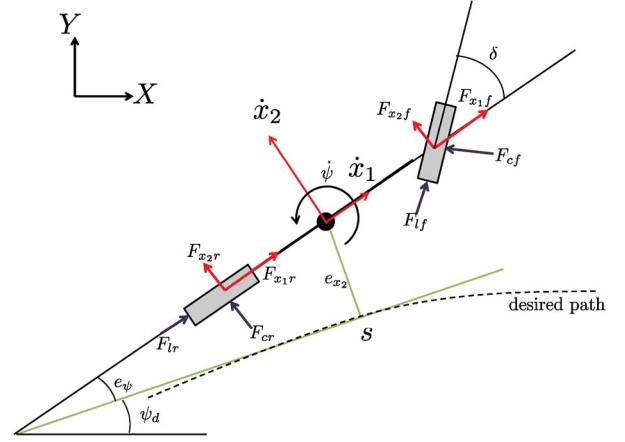


Fig. 1. Illustration depicting the forces modeled in the vehicle body-fixed frame (F_{x1*} and F_{x2*}), the forces in the tire-fixed frame (F_{l*} and F_{c*}), and the rotational and translational velocities used in the vehicle model described in Section III-A. The relative coordinates e_{x2} and e_ψ , in addition to the road tangent ψ_d , are also pictured.

ential equations based on the bicycle model described in [34] to describe the vehicle motion within the lane:

$$m\ddot{x}_1 = m\dot{x}_2\dot{\psi} + 2F_{x1f} + 2F_{x1r} \quad (7a)$$

$$m\ddot{x}_2 = -m\dot{x}_1\dot{\psi} + 2F_{x2f} + 2F_{x2r} \quad (7b)$$

$$I_z\ddot{\psi} = 2aF_{x2f} - 2bF_{x2r} \quad (7c)$$

$$\dot{e}_\psi = \dot{\psi} - \dot{\psi}_d \quad (7d)$$

$$\dot{e}_{x2} = \dot{x}_2 \cos(e_\psi) + \dot{x}_1 \sin(e_\psi) \quad (7e)$$

$$\dot{s} = \dot{x}_1 \cos(e_\psi) - \dot{x}_2 \sin(e_\psi) \quad (7f)$$

where m and I_z denote the vehicle mass and yaw inertia, respectively; a and b denote the distances from the vehicle's center of gravity to the front and rear axles, respectively. \dot{x}_1 and \dot{x}_2 denote the vehicle's longitudinal and lateral velocities, respectively, and $\dot{\psi}$ denotes the vehicle's turning rate around a vertical axis at the vehicle's center of gravity. e_ψ and e_{x2} in Fig. 1 denote the vehicle orientation and lateral position in a road-aligned coordinate frame, respectively. ψ_d is the angle of the tangent to the road centerline in a fix coordinate frame. s is the vehicle longitudinal position along the lane center. F_{x2f} and F_{x2r} are the front and rear tire forces acting along the vehicle's lateral axis, and F_{x1f} and F_{x1r} are the corresponding forces acting along the vehicle's longitudinal axis.

The longitudinal and lateral tire force components in the vehicle body frame are modeled as

$$F_{x1*} = F_{l*} \cos(\delta_*) - F_{c*} \sin(\delta_*) \quad (8a)$$

$$F_{x2*} = F_{l*} \sin(\delta_*) + F_{c*} \cos(\delta_*) \quad (8b)$$

where $*$ denotes either f or r for front and rear tire, and δ_* is the steering angle at the wheel. We introduce the following assumption on the steering angles.

Assumption 2: Only the steering angle at the front wheels can be controlled (i.e., $\delta_f = \delta$ and $\delta_r = 0$). In addition, an actuator δ_c , which corrects the driver-commanded steering angle δ_d , is available.

The longitudinal force in the tire frame, i.e., $f_{l\star}$, is calculated from

$$f_{l\star} = \beta_r \mu F_{z\star} \quad (9)$$

where $\beta_r \in [-1, 1]$ is referred to as the braking ratio. $\beta_r = -1$ corresponds to full braking, and $\beta_r = 1$ corresponds to full throttle. $f_{c\star}$ is computed using a modified nonlinear Fiala tire model [35]

$$f_{c\star} = \begin{cases} -C_\alpha \tan(\alpha_\star) \\ -\frac{C_\alpha^3}{27\eta^2 \mu^2 F_{z\star}} \tan^3(\alpha_\star) \\ +\frac{C_\alpha^2}{2\eta \mu F_{z\star}} |\tan(\alpha_\star)| \tan(\alpha_\star), & \text{if } |\alpha_\star| < \alpha_{sl} \\ -\eta \mu F_{z\star} \text{sgn}(\alpha_\star), & \text{if } |\alpha_\star| \geq \alpha_{sl} \end{cases} \quad (10)$$

where α_\star denotes the tire slip angle, μ denotes the friction coefficient, and $F_{z\star}$ denotes the vertical load at each wheel. C_α is the tire cornering stiffness, and $\eta = \sqrt{\mu^2 F_z^2 - f_l^2} / \mu F_z$, which can be written as $\eta = \sqrt{1 - \beta_r^2}$. $\alpha_{sl} = \tan^{-1}(3\eta \mu F_z / C_\alpha)$ denotes the saturation point of the slip ratio. μ is assumed to be a known constant and is the same for all wheels. The vertical forces $F_{z\star}$ are also assumed to be constants and are determined by the steady-state weight distribution of the vehicle with no accelerations at the center of the gravity. The slip ratios α_\star are approximated as

$$\alpha_f = \frac{\dot{x}_2 + a\dot{\psi}}{\dot{x}_1} - \delta \quad \alpha_r = \frac{\dot{x}_2 - b\dot{\psi}}{\dot{x}_1}. \quad (11)$$

We write the model described in (7)–(11) in the form

$$\dot{x}(t) = f_c(x(t), u(t)) \quad (12)$$

where $x = [\dot{x}_1, \dot{x}_2, \dot{\psi}, e_\psi, e_{x_2}, s]$ and $u = [\delta, \beta_r]$ are the state and input vectors, respectively. We discretize the system in (12) with a fixed sampling time T_s to obtain

$$x(k+1) = f(x(k), u(k)). \quad (13)$$

B. Safety Constraints

The overall objective of the proposed safety system is to keep the vehicle on the road while avoiding obstacles. Here, we describe the constraints on the vehicle state and input. To reduce the complexity, we constrain the center of gravity of the vehicle for road-keeping and obstacle-avoidance constraints. The requirement that the vehicle remain in the free portion of the road is expressed as

$$e_{x_2, \min} + w \leq e_{x_2} \leq e_{x_2, \max} - w \quad (14)$$

where $e_{x_2, \min}$ and $e_{x_2, \max}$ are the bounds determined by the road width and obstacle position at each prediction step. w is the vehicle width. Angle deviation constraints are not incorporated because, at high speed, large angle deviation results in either an inability to keep the vehicle on the road or a high cost from the large/fast steering input to keep the vehicle on the road. Thus, the lateral deviation constraints implicitly limit angle deviations. To ensure that the vehicle operates in a state space where a normal driver can easily handle the vehicle, we

constrain the tire slip angles to remain within the linear region. Hence

$$-\alpha_{sl} \leq \alpha_\star \leq \alpha_{sl}. \quad (15)$$

The constraints (14)–(15) are compactly written as

$$h(x, u) \leq \mathbf{0} \quad (16)$$

where

$$h(x, u) = \begin{bmatrix} -e_{x_2} + e_{x_2, \min} + w \\ e_{x_2} - e_{x_2, \max} - w \\ -\alpha_f(x, u) - \alpha_{sl} \\ -\alpha_r(x) - \alpha_{sl} \\ \alpha_f(x, u) - \alpha_{sl} \\ \alpha_r(x) - \alpha_{sl} \end{bmatrix} \quad (17)$$

and $\mathbf{0}$ is the zero vector with appropriate dimension.

C. VPM and VIF

Since determining the exact level of distraction of the driver is an ill-posed problem, we instead hypothesize that the current configuration of the driver and the environment along with prior observations is correlated with the future state of the vehicle over a short time horizon. We further hypothesize that the marginal distribution of these current and prior observations, with respect to the future state of the vehicle, i.e., $P((X(k) - x(0)) \subset \mathcal{O} | \mathcal{O}, \mathcal{I})$, is a multimodal distribution.

This marginal distribution, which allows us to determine the VPM, is constructed by first clustering the data using a k-means algorithm [36] and then constructing an empirical distribution within each cluster separately. As a result, we do not explicitly determine whether the subject is attentive or distracted or how the current state of the environment (e.g., road curvature or the state of other vehicles) deterministically influences driver behavior.

For each cluster \mathcal{M} , there is a set $\mathcal{O}_{\mathcal{M}} = \{o_i\}_{i \in \{1, \dots, |\mathcal{M}|\}} \subset \mathcal{O}$ of observations associated with each cluster, where $|\mathcal{M}|$ denotes the number of elements in mode \mathcal{M} . Each o_i is a vector composed of observations, which we describe in Section V. Fixing $\bar{N} \in \mathbb{N}$ with respect to each observation o_i , we can associate the actual observed vehicle input and trajectory for the next $\bar{N} + 1$ time steps beginning at time step i , which we denote by $u_{i, \bar{N}} : \{0, \dots, \bar{N}\} \rightarrow \mathbb{R}^m$ and $x_{i, \bar{N}} : \{0, \dots, \bar{N}\} \rightarrow \mathbb{R}^n$, respectively.

As Algorithm 1 runs at each time step, we are given current information \mathcal{I} , which determines that cluster to which the driver belongs at the time step in question. Letting \mathcal{M} denote the mode to which \mathcal{I} belongs, for some $z_1, z_2 \in \mathbb{R}^n$, let $\Omega = \{x \in \mathbb{R}^n | z_1 \leq x \leq z_2\}$, and we define the probability that the difference between the future trajectory at time step k , denoted by $x(k)$, and the vehicle's initial condition $x(0)$ belongs to Ω (between z_1 and z_2) as

$$\begin{aligned} &P(z_1 \leq x - x(0) \leq z_2 | \mathcal{O}, \mathcal{I}) \\ &= \frac{1}{|\mathcal{M}|} \sum_{i \in \mathcal{M}} \mathbb{1}\{z_1(k) \leq (x_{i, \bar{N}}(k) - x_{i, \bar{N}}(0)) \leq z_2(k)\}. \end{aligned} \quad (18)$$

During our experiment in Section IV, we find the minimum z_1 (most counterclockwise turn) and maximum z_2 (most clockwise turn) such that the probability in (18) is 1 (i.e., we let $\alpha = 1$).

Given the VPM, the VIF must determine whether intervention is required. In our prior work, the function Π in (3) was chosen to be the identity map since our focus was just the determination of when to intervene rather than how to intervene [13]. Our goal, in contrast, within this paper is the construction of a real-time intervention procedure. Since the state of the art in our chosen strategy for real-time control intervention, which we describe in the following, is only able to guarantee the safety of a single trajectory after autonomous intervention, we construct Π to pick the expected value $\Delta(1, k, \mathcal{O}, \mathcal{I})$ for each $k \in \{0, \dots, \bar{N}\}$. For convenience, we denote the user's input corresponding to this expected trajectory as $u^{\text{driver}}(k)$.

D. Predictive Control Problem

Here, we formulate the lane-keeping and obstacle-avoidance problem as a model predictive control (MPC) problem [37]. At each sampling time instant, an optimal input sequence is computed by solving a constrained finite-time optimal control problem. The computed optimal control input sequence is only applied to the system during the following sampling interval. At the next time step, the optimal control problem is solved again, using new measurements.

Using the discretized system described in (13), we formulate the optimization problem being solved at each time instant as

$$\min_{\mathbf{u}, \varepsilon} \lambda \varepsilon + \sum_{k=0}^{\bar{N}} \left((u^c(k))^T R (u^c(k)) + (\delta u^c(k))^T S (\delta u^c(k)) \right) \quad (19a)$$

$$\text{s.t. } x(k+1) = f(x(k), u(k)) \quad (19b)$$

$$u(k) = u^c(k) + u^{\text{driver}}(k) \quad (19c)$$

$$u^c(k) = u^c(k-1) + \delta u^c(k) \quad (19d)$$

$$h(x(k), u(k)) \leq 1\varepsilon, \quad \varepsilon \geq 0 \quad (19e)$$

$$u(k) \in \mathcal{U}_{\bar{N}} \quad (19f)$$

$$\delta u^c(k) \in \mathcal{U}_{\bar{N}} \quad (19g)$$

$$\forall k \in \{0, \dots, \bar{N} - 1\} \quad (19h)$$

$$x(0) = x_0. \quad (19i)$$

$x(k)$ denotes the predicted state at time k from the current time obtained by applying the control sequence $\mathbf{u} = \{u(0), \dots, u(k-1)\}$ to the system in (13) with $x(0) = x_0$. \bar{N} denotes the prediction horizon. Equation (19e) describes the state constraints imposed, in order to ensure lane keeping and obstacle avoidance, as described in Section III-B, and have been imposed as soft constraints, by introducing the slack variable ε in (19a) and (19e). $u^{\text{driver}}(k)$ is the predicted driver steering input at time step k , provided by the VPM. $u^c(k)$ is the correction to the driver's input. $\delta u^c(k)$ is the change in correction to the driver's input, which is initialized to be $\mathbf{0}$. R , S , and λ are weights penalizing the control correction, the change rate of control correction, and the violation of the soft constraints,

respectively. In the experiment described in Section V, the values of R , S , and λ are chosen to be $\begin{bmatrix} 1 & 0 \\ 0 & 0.1 \end{bmatrix}$, $\begin{bmatrix} 0.1 & 0 \\ 0 & 0.01 \end{bmatrix}$, and 1000, respectively. \mathbf{u} and ε are initialized as zero at the beginning of each optimization.

It is clear that u^c is zero at the optimal solution to the optimization problem as long as u^{driver} is able to keep the states inside the safe region. The MPC controller runs on a dSpace Microautobox, which runs at 800 MHz with 8-MB memory. The nonlinear optimization problem in (19) is a nonlinear and nonconvex receding horizon optimization problem and is solved by NPSOL at a sampling rate of 200 ms. The computational time varies under different scenarios. In practice, we found that the solver converges to a solution in the given sampling time to the required accuracy.

IV. EXPERIMENTAL SETUP

This section describes an experiment, which is inspired by the survey in [4], conducted to analyze our proposed framework.

A. Hardware Setup

The experimental setup, as illustrated in Fig. 2, and the driver view, as illustrated in Fig. 3, consist of a driving simulator, a Microsoft Kinect, and a smartphone to simulate distraction. The Kinect is placed approximately 1 m in front of the subject, 0.25 m to the right of the subject, and 0.25 m above the subject's eye level. Due to the existing limitations of the Kinect and since our distraction mechanism relies upon the driver answering a cell phone, we focused our attention on the shoulder, elbow, and wrist joints on each arm. A steering wheel and pedals (Logitech G-25) are used to interface with the driving simulator through Simulink. The driving simulator, i.e., CarSim 8.02, communicates with a real-time dSpace DS1006 hardware-in-the-loop over a fiber-optic link for the physics and model computations, at 50 frames per second, and a real-time dSpace DS1401 MicroAutoBox over an ethernet link for the MPC controller. CarSim's vehicle model is proprietary, but CarSim outputs information at each time step about the state of the vehicle and environment. This includes information at each time step about the global coordinates of the car, the path of the center of the road in global coordinates, the radius of curvature of the road, the longitudinal and lateral velocities of the car at each time step, the location of all of the obstacles in global coordinates, the longitudinal and lateral velocities of all of the obstacles, and a radar sensor telling us which obstacles are observable from the vehicle. All of this information is incorporated during the safety constraint formulation used within the MPC optimization procedure, as described in Section III-D.

The dSpace DS1006 box sends the vehicle state via the Controller Area Network bus (CANbus) to the dSpace DS1401 MicroAutoBox, which runs the MPC controller. CANbus is the industry standard that allows microcontrollers and devices to communicate with each other in vehicles. The MPC optimization problem sends the controlled vehicle state via CANbus back to the DS1006 box. A Microsoft Kinect is used to obtain joint locations in 3-D space for the upper extremities

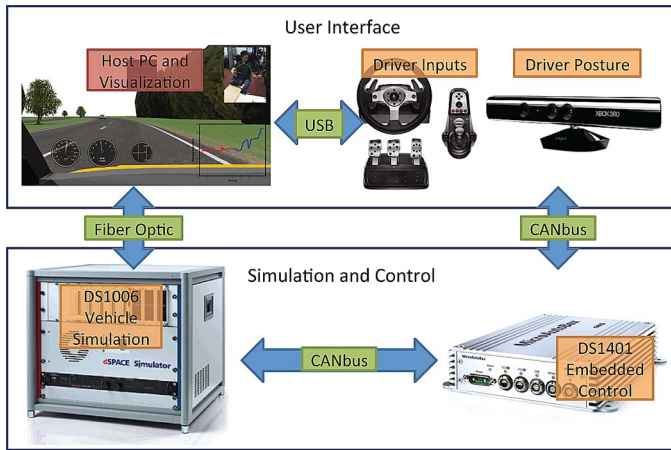


Fig. 2. Flowchart depicting the experimental setup. The top box represents the input and visual output modules of the system. The bottom box represents the simulation and control modules of the system. The blue arrows represent data transfer between the modules, and green boxes represent the method of data transfer. The orange boxes represent the hardware components, and the red box represents the software component of the system.

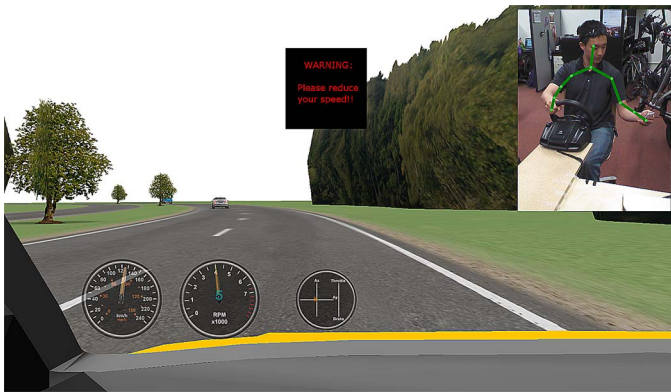


Fig. 3. Simulated environment from the driver's view used in our experiment. On the top right is the view from the Kinect. On the bottom are the speedometer, tachometer, and steering/acceleration inputs.

at 20 Hz. The DS1401 box sends the vehicle state via CANbus to the main computer, which predicts future driving behavior in under 10 ms, and returns it to the DS1401 for the MPC optimization procedure. The DS1401 box also sends data via an ethernet link to the main driving simulator machine for recording.

B. Experimental Design

The experiment was conducted over a single 2-h session. The first hour was for data collection, and the second hour was for evaluation. The subject was asked to keep the vehicle's speed between 105 and 110 km/h, to drive safely, and to stay in the lane unless an emergency-forced lane departure. There was traffic in the opposite lane and also in the driver's lane. Programming the behavior of any traffic element required describing either its location relative to the road or the subject's vehicle at any time. To maintain the tenability of the experiment, the traffic in the same lane (except during unexpected events described in the following) traveled at the same longitudinal velocity as the subject's vehicle at any time.

As a result, the subject could never reach other vehicles in the same lane. Nevertheless, in certain circumstances, which we describe in the following, a vehicle in the same lane travels at a speed different than the subject's vehicle.

The VPM for each subject was built over four 10-min courses¹ Scenario 1 consisted of a two-way two-lane roadway with many turns and several roadblocks in the way. Scenario 2 consisted of a one-way single-lane course with 13 left and right turns. Scenario 3 consisted of a two-way single-lane roadway. Scenario 4 consisted of a straight two-lane highway with occasional turns. If an accident ever occurred with another vehicle or if the driver drove off the course, then the scenario continued as if nothing occurred.

In order to test the effect of distraction, the subject was distracted by requiring them to read and respond to a text message on a phone. To keep the text messages consistent, an Android smartphone was provided to each participant. A software application described in [13] was used to display questions and relay responses. There were two types of messages sent to the driver. The first type required the driver to respond to a question with a short answer. Example questions of the first type are "what did you have for lunch today?" and "what did you do last weekend?" The second type required the driver to repeat the sentence that appeared. Example sentences are "I'm doing fine, how about you?" or "Oh no, I forgot my tie."

Training scenario 1 is as follows: From the start, there are three cars in the left lane driving at 100 km/h. Whenever a vehicle trails the subject's car by 50 m, the vehicle reappears 300 m in front of the subject's vehicle. Approximately 3 min into the drive, a cow appears on the right lane of the road, which the driver must avoid. Then, 4.5 min into the drive, a lead car appears, which the driver can choose to overtake. After 6 min into the drive, another lead car appears, which the driver can choose to overtake. Approximately every 40 s, the driver is notified to respond to a text message, totaling 13 text messages.

Training scenario 2 is as follows: From the start, there are no other cars or traffic. The driver must navigate through 13 turns, while responding to a text message every 40 s. The driver is asked to respond to a text message in different types of situations (straightaway, preturn, turn, and postturn), totaling 13 text messages.

Training scenario 3 is as follows: From the start, there are three cars in the left lane driving at 100 km/h and three cars in the right lane driving at 100 km/h. Approximately 6 min into the drive, a lead car in the right lane appears driving at the same speed as the subject. Then, 1.5 min later, the lead car slows down to 36 km/h. Approximately every 40 s, the driver is notified to respond to a text message, totaling 13 text messages.

Training scenario 4 is as follows: From the start, there are three cars in the left lane driving at 100 km/h. There is also a lead car, which slows down every 800 m, causing the driver to do a lane change or stay behind the lead car at 35 km/h. Every time the lead car trails the subject's vehicle by 100 m, it reappears 100 m in front of the subject. Approximately every 40 s, the driver is notified to respond to a text message. The

¹The roadway and scene information can be viewed at www.purl.org/vshia/experiment_route.

driver is asked to respond to a text message in different types of situations (straightaway, preturn, turn, and postturn), totaling ten text messages.

During the second hour, there were four testing scenarios. During the first three testing scenarios the VPM, the VIF, and the semiautonomous controller are turned on. During the final testing scenario, only the VPM and the VIF are turned on. This was done in order to compute the RVIF and PVIF metrics without the semiautonomous controller's intervention, which is purposefully constructed to prevent accidents. Subjects were made aware when the semiautonomous controller was active during the testing scenario and when it was not active.

In order to test the semiautonomous controller on one's daily commute, the first three testing scenarios for the evaluation of our autonomous framework are nearly identical to those used during the training scenario. The only major difference being that the set of text messages sent to the driver is different. Testing scenario 1 is the same course as training scenario 1. Testing scenario 2 is a similar course as training scenario 4. The difference between testing scenario 2 and training scenario 4 is that they contain different turns. Testing scenario 3 is the same course as training scenario 2.

Testing scenario 4 is as follows: From the start, there are three cars in the left lane driving at 100 km/h. Approximately 3 min into the drive, an animal suddenly appears in the roadway. Then, 6 min later, another obstacle appears. One minute later, a vehicle appears driving at the same speed as the subject. Approximately every 40 s, the driver is notified to respond to a text message. The driver is asked to respond to a text message in different types of situations (straightaway, preturn, turn, and postturn), totaling eight text messages.

V. EVALUATION

We ran the experiment on 24 individuals. Nineteen subjects were between 20 and 30 years old, and five subjects were over 30 years old. There were 15 male subjects and nine female subjects. Fifteen subjects drove at least once a week, and nine subjects drove less frequently. The average driving experience was 8.2 years, with a standard deviation of 7.8 years. The least experienced driver had one year of experience, and the most had 40 years of experience. For each subject, we allowed the k-means algorithm to train using data from the four training scenarios and tested our autonomous controller performance on the data from the three testing scenarios. The data used to construct the VPM corresponded to the environmental bounds generated by CarSim for the next 4 s and the locations in \mathbb{R}^3 of the left and right shoulders, elbows, and wrists for the previous 2 s. For each cluster, we associate the predicted set of vehicle inputs and the associated vehicle trajectory for the following 1.2 s. For this model, we fixed the number of clusters chosen by the k-means algorithm to 100, $\bar{N} = 40$, and $\underline{N} = 40$. During the three testing scenarios, the VPM was run in the loop with the MPC controller.

The advice given to motorists who are learning to drive is to maintain a 3-s distance between themselves and the car in front of them while traveling at 60 mi/h, or approximately

96.6 km/h, in order to be able to stop safely [38]. Safe driving, as a result, requires the driver to plan over a 3-s horizon. Since, in our context, the driver was traveling at a maximum of 110 km/h, we assumed that the driver was planning over a 4-s horizon, which we used during the clustering procedure to determine the VPM.

Ideally, the entire previous history of the joint state up to the current state would be used during the clustering procedure to determine the VPM. However, computational limitations make this untenable. We chose the previous 2-s joint localization since it was the longest time horizon that we were able to choose while guaranteeing the real-time performance of our system. Similarly, the future vehicle trajectory was computed over a 1.2-s horizon since it was the longest time horizon over which we could plan while ensuring the real-time performance of the MPC optimization. We plan on investigating techniques to extend these horizons in the future. $\underline{N} = 40$ and $\bar{N} = 40$ were chosen to provide a consistent sampling rate between the different sensors over the time horizons that we fixed.

A. Probability Distributions

Using our VPM, we generate probability distributions for a driver's future steering and future set of trajectories. Each predicted set of trajectories, as illustrated in Figs. 4 and 5, describes the distribution of how a particular driver behaved in the past when in a similar scenario. For a single driver, the distribution of driver behavior follows the road when attentive on a curved road. When the driver is distracted on a turn, there were instances, albeit rare, when the driver failed to make the turn in time. The wide distribution in the straight-road-and-attentive probability distributions arises because our current VPM fails to distinguish intent. For example, a future left lane change and future straight driving are clustered together, making the set larger. When a driver is distracted on a straight road, he drives fairly straight, as expected.

B. Qualitative Evaluation

This section presents the qualitative evaluation of our experimental setup drawn from surveying the participants. The presurvey was completed after the four training trials and focused on the subject's experience with the simulator and an evaluation of their own performance. The postsurvey was completed after all eight trials and asked the subject to evaluate the semiautonomous controller. The results are summarized in Table I.

In the precontroller survey, 54% of subjects admitted responding to the text messages until it was safe, potentially causing a problem for the training data since the distinction between distraction could no longer be determined primarily by the position of the hand holding the phone. In the post-controller survey, a majority of subjects said that, due to the semiautonomous controller, they changed their behavior either by letting the vehicle drift closer to the lane boundary, by driving at higher speeds, or by checking their phone more often. The unintended consequences of this type of semiautonomous controller on driver behavior has yet to be studied and may be critical to the future implementation of intervening controllers.

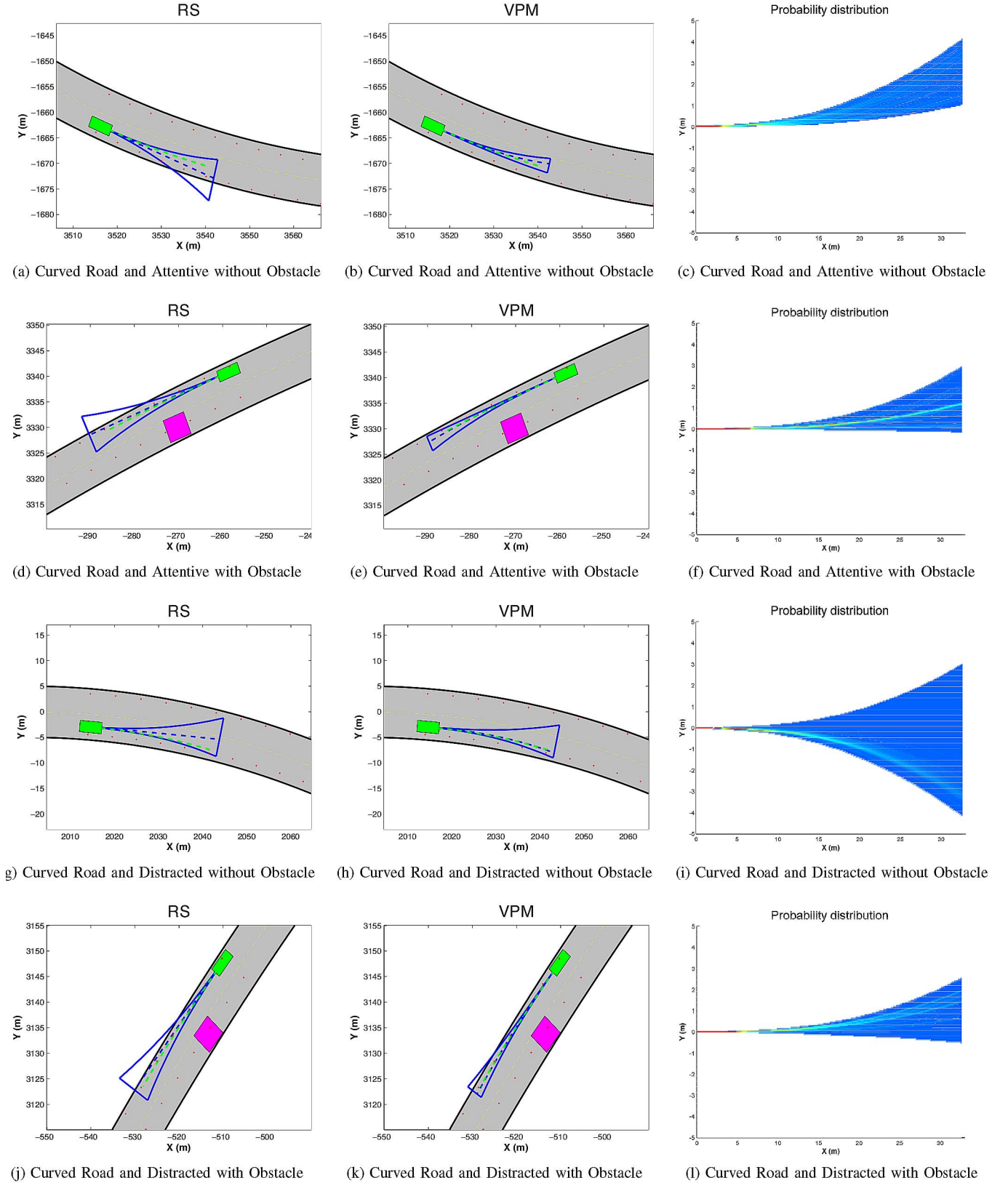


Fig. 4. Sample prediction sets at the same instance in time for four scenarios (each row) for the RS model (column 1), the VPM model (column 2), and the corresponding probability distribution used for the VPM model (column 3). The predicted sets are drawn in blue beginning at the subject's vehicle, which is drawn as a green box, in an environment with bounds drawn as red dots. The dotted blue line is the expected value of the trajectory from the VPM used in the VIF, and the dotted green line is the actual trajectory of the vehicle. Other obstacles are illustrated as magenta boxes. These sets are taken from scenarios 1 and 2 in the testing set. In column 3, the red color corresponds to a probability of 1 and blue corresponds to a probability of 0. Colors range from blue to red and represent a probability from 0 to 1, respectively.

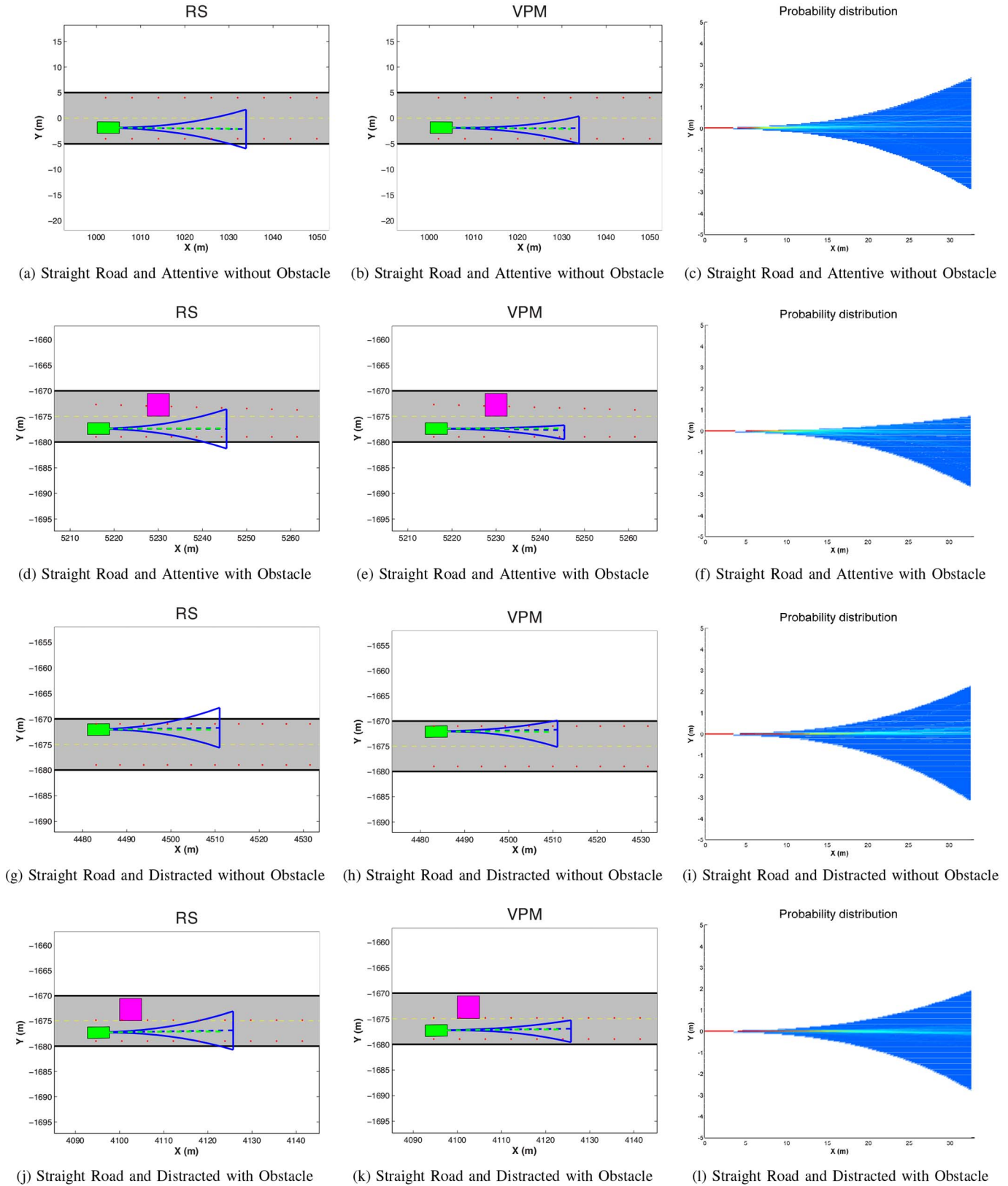


Fig. 5. Sample prediction sets at the same instance in time for four scenarios (each row) for the RS model (column 1), the VPM model (column 2), and the corresponding probability distribution used for the VPM model (column 3). The predicted sets are drawn in blue beginning at the subject's vehicle, which is drawn as a green box, in an environment with bounds drawn as red dots. The dotted blue line is the expected value of the trajectory from the VPM used in the VIF, and the dotted green line is the actual trajectory of the vehicle. Other obstacles are illustrated as magenta boxes. These sets are taken from scenarios 1 and 2 in the testing set. In column 3, the red color corresponds to a probability of 1, and blue corresponds to a probability of 0. Colors range from blue to red and represent a probability from 0 to 1, respectively.

TABLE I
SURVEY AND RESPONSES BY PARTICIPANTS IN THE EXPERIMENT DESCRIBED IN SECTION IV. THE PRECONTROLLER SURVEY (PRE) WAS CONDUCTED AFTER PERFORMING THE TRAINING SCENARIOS BUT BEFORE THE TESTING SCENARIOS. THE POSTCONTROLLER SURVEY (POST) WAS CONDUCTED AFTER THE ENTIRE EXPERIMENT

Survey	Question	Yes (%)	No (%)	Neutral (%)
Pre	Realistic driving environment?	37	46	17
Pre	Improved performance over trials?	79	4	17
Pre	Realistic simulated distraction?	54	29	17
Pre	Texting response realistic?	38	54	8
Pre	Responded to text only when safe?	54	29	17
Post	Controller effective in preventing collisions?	75	4	21
Post	Controller too aggressive?	50	29	21
Post	Controller changed driving behavior?	58	28	14

TABLE II
MEAN/STANDARD DEVIATION PERFORMANCE OF THE EMPIRICAL VPM COMPARED TO THE RS WITH RESPECT TO THE FOUR METRICS DEFINED IN SECTION II-C FOR ALL SUBJECTS. THE METRICS AVPM AND PVPM WERE TESTED USING THE FOUR TESTING SCENARIOS WITH THE SEMIAUTONOMOUS CONTROLLER ACTIVE. IN ORDER TO COMPUTE RVIF AND PVIF ON VEHICLE TRAJECTORIES UNCHANGED BY THE SEMIAUTONOMOUS CONTROLLER, WE USED TESTING SCENARIO 4

Model - # clusters	AVPM	PVPM	RVIF	PVIF
<i>RS</i>	100/0	0	100	9.84/9.83
<i>VPM</i> - 50	91.95/4.11	58.97/15.88	92.86/0.0516	71.19/18.65
<i>VPM</i> - 100	87.27/4.14	50.07/15.05	92.66/0.0454	71.20/18.79
<i>VPM</i> - 200	79.19/4.34	41.26/13.80	89.52/0.0826	66.50/19.84

Subjects also noted that the semiautonomous controller was effective in preventing the vehicle from going off the road and preventing collisions. In summary, although the driving simulator lacked speed and force feedback, the participants were still able to adapt to the system. Participants complained that the monitor was too small and should have side screens to provide peripheral information.

C. Evaluation of the Semiautonomous Controller

We compare the performance of our VPM to one that ignores all prior observations and defines the VPM as the RS, using the four metrics presented in Section II-C and shown in Table II. This serves as an illustration of the baseline performance of existing active safety systems. It is clear that, by incorporating information about the driver, we can reliably predict the behavior of the vehicle. Figs. 4 and 5 illustrate the VPM and the actual observed trajectory of the driver. For these experiments, 100 clusters were chosen to balance the four metrics. Depending on the application and desired accuracy or precision, the desired number of clusters is subject to tuning. We also plan to investigate the optimal number of clusters for each driver.

The metrics AVPM and PVPM were tested using the four testing scenarios with the semiautonomous controller active. While the AVPM of the VPM is not as accurate as the RS, the PVPM of the VPM indicates that the set that the VPM outputs is 50% smaller than its RS counterpart. In addition, the RVIF indicates that the semiautonomous architecture intervenes during 92% of the instances where the driver was in danger. More importantly, the PVIF illustrates that, during 71% of the times that the architecture relying on the VPM decided to intervene, the driver was going to be in danger in the near future. Compare this to the architecture that relied upon the RS, which intervened far more often than was necessary. Once again, the RVIF and PVIF was calculated using only testing scenario 4, which did not have the semiautonomous controller active.

During the testing scenarios with the autonomous controller active, the autonomous controller was always able to keep

the vehicle safe. Out of the 734 848 number of total opportunities that the autonomous controller could have chosen to intervene, it intervened approximately 94 996 number of times, or approximately 12.93% of the time. Fig. 6 illustrates some instances when the controller chose to intervene. Since our scenarios were simple, there were no circumstances in which the autonomous controller could not find a trajectory to keep the vehicle safe.

VI. CONCLUSION

In this paper, we have presented and implemented an architecture for a provably safe semiautonomous controller that predicts potential vehicle behavior and intervenes when the vehicle is deemed unsafe. Using four metrics that we define, we show that incorporating a driver model during the construction of the semiautonomous controller allows for better performance than a semiautonomous controller that treats the driver as a disturbance. This semiautonomous framework is modular and easily allows the use of different driver models and controllers. Future work includes developing and implementing a better unsupervised learning algorithm, such as hierarchical clustering or active learning using convex programming [39], [40].

Before extending our proposed setup to a real-world system, several issues must be considered. First, the driver model must be able to update itself in real time as observations arrive. A method to perform this operation was proposed in [41]. Second, different types of driver distraction, such as those considered in [42], may require the inclusion of additional joints such as head position or attitude and implementing articulated tracking systems such as those proposed in [21], [43], and [44]. Third, incorporating strategies to predict future states of other vehicles or obstacles in the environment, such as those proposed in [45], [46], will be critical. Finally, taking advantage of the entire set of predicted future trajectories for the vehicle, rather than just the expected trajectory, by extending stochastic MPC [47] to the nonlinear vehicle model case will be essential to improving the overall performance of the proposed architecture.

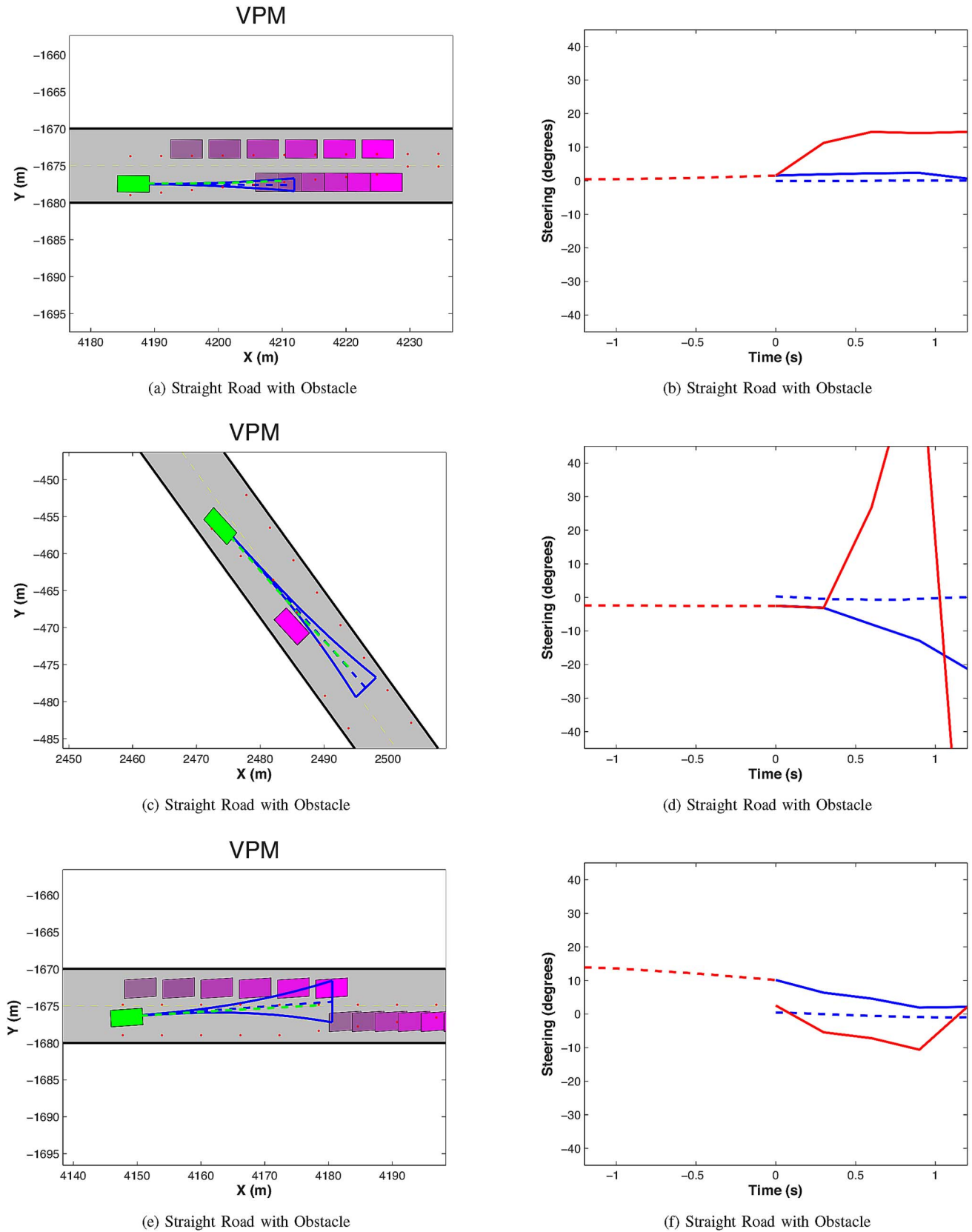


Fig. 6. Sample prediction sets with instances when the controller provided a change to the driver's steering input. The color scheme in the first column is the same as in Fig. 5, except for the gradient of magenta. The gradient from dark to bright magenta shows the position of the obstacle from 0 to 1.2 s in the future. In the second column, the dotted red line denotes the past steering wheel angle, the solid blue line represents the future steering wheel angle, the dotted blue line denotes the expected steering from the VPM, and the red line represents the future controller steering. The dotted red line and solid blue line are continuous because they represent the steering wheel angle.

These experiments were not without shortcomings. Common subject complaints involved the system lacking tactile feedback and that the controller was too “jerky”, which is a limitation of current hardware. The nonlinear MPC controller has a sampling time of 200 ms, limiting the agility of maneuvers and responding time. Recent work in [48] has reduced the sampling time 100 ms. Additional methods to improve the speed may include using a linear time-varying vehicle model or parallelizing the nonlinear program. When subjects were driving with the phone in their hand, they often looked up at the road, effectively deceiving the driver model. This combined with the limitations of the Kinect requires an improved method of measuring driver distraction.

In contrast to fully autonomous systems, whose widespread adoption has been forestalled due to the inability of existing verification tools to guarantee their safety in arbitrary situations, semiautonomous systems must address the problem of transitioning control between the driver and an autonomous controller. Currently, research groups such as the Revs Program at Stanford and the Center for Automotive Research are studying this problem [49]. If the vehicle hands over control to a nonattentive driver, the entire system may be just as unsafe as a distracted driver [50]. More troublingly, many of the participants in our experiment reported that the controller changed their own behavior. The unintended consequences of this type of semiautonomous controller on driver behavior needs to be more fully studied.

REFERENCES

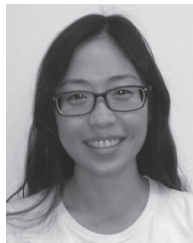
- [1] National Highway Traffic Safety Administration, Fatality Analysis Reporting System, Washington, DC, USA, 2011.
- [2] S. G. Klauer, “The impact of driver inattention on near-crash/crash risk: An analysis using the 100-car naturalistic driving study data,” Virginia Tech Transp. Inst., Blacksburg, VA, USA, Apr. 2006, National Highway Traffic Safety Administration.
- [3] J. C. Stutts and W. W. Hunter, “Driver inattention, driver distraction and traffic crashes,” *Inst. Transp. Eng.*, vol. 73, no. 7, pp. 34–45, Jul. 2003.
- [4] D. Basacik, N. Reed, and R. Robbins, “Smartphone use while driving: A simulator study,” Transp. Res. Lab., Berkshire, U.K., Tech. Rep. 1, Mar. 2012.
- [5] D. Ascone, T. Lindsey, and C. Varghese, “An examination of driver distraction as recorded in NHTSA databases,” Transp. Res. Board, Washington, DC, USA, Sep. 2009.
- [6] H. Kress-Gazit, D. C. Conner, H. Choset, A. A. Rizzi, and G. J. Pappas, “Courteous cars,” *IEEE Robot. Autom. Mag.*, vol. 15, no. 1, pp. 30–38, Mar. 2008.
- [7] S. Thrun *et al.*, “Stanley: The robot that won the DARPA Grand Challenge,” in *The 2005 DARPA Grand Challenge*. Berlin, Germany: Springer-Verlag, 2007.
- [8] C. Urmson *et al.*, “Autonomous driving in urban environments: Boss and the urban challenge,” *J. Field Robot.*, vol. 25, no. 8, pp. 425–466, Aug. 2008.
- [9] P. Varaiya, “Smart cars on smart roads: Problems of control,” *IEEE Trans. Autom. Control*, vol. 38, no. 2, pp. 195–207, Feb. 1993.
- [10] E. Coelingh, L. Jakobsson, H. Lind, and M. Lindman, “Collision warning with auto brake—A real-life safety perspective,” Nat. Highway Traffic Safety Admin., Washington, DC, USA, Apr. 2007.
- [11] S. J. Anderson, S. C. Peters, T. E. Pilutti, and K. D. Iagnemma, “An optimal-control-based framework for trajectory planning, threat assessment, semi-autonomous control of passenger vehicles in hazard avoidance scenarios,” *Int. J. Veh. Auton. Syst.*, vol. 8, no. 2, pp. 190–216, Jan. 2010.
- [12] F. Borrelli, *Constrained Optimal Control of Linear and Hybrid Systems*, vol. 290. Berlin, Germany: Springer-Verlag, 2003, ser. Lecture Notes in Control and Information Sciences.
- [13] R. Vasudevan *et al.*, “Safe semi-autonomous control with enhanced driver modeling,” in *Proc. Amer. Control Conf.*, Jun. 2012, pp. 2896–2903.
- [14] D. Basacik and A. Stevens, *Scoping Study of Driver Distraction*, vol. 95, *Road Safety Research Report*. Berkshire, U.K.: Transport Research Laboratory, Feb. 2008.
- [15] M. Regan, J. D. Lee, and K. L. Young, *Driver Distraction: Theory, Effects, Mitigation*. Boca Raton, FL, USA: CRC Press, 2008.
- [16] A. Sathyanarayana, S. Nageswaren, H. Ghasemzadeh, R. Jafari, and J. H. L. Hansen, “Body sensor networks for driver distraction identification,” in *Proc. IEEE Int. Conf. Veh. Electron. Safety*, Sep. 2008, pp. 120–125.
- [17] E. Wahlstrom, O. Masoud, and N. Papanikolopoulos, “Vision-based methods for driver monitoring,” in *Proc. IEEE Intell. Transp. Syst.*, Oct. 2003, vol. 2, pp. 903–908.
- [18] L. M. Bergasa, J. Nuevo, M. A. Sotelo, R. Barea, and M. E. Lopez, “Real-time system for monitoring driver vigilance,” *IEEE Trans. Intell. Transp. Syst.*, vol. 7, no. 1, pp. 63–77, Mar. 2006.
- [19] M. Jabon, J. Bailenson, E. Pontikakis, L. Takayama, and C. Nass, “Facial expression analysis for predicting unsafe driving behavior,” *IEEE Pervasive Comput.*, vol. 10, no. 4, pp. 84–95, Apr. 2011.
- [20] L. Fletcher, N. Apostoloff, L. Petersson, and A. Zelinsky, “Vision in and out of vehicles,” *IEEE Intell. Syst.*, vol. 18, no. 3, pp. 12–17, May/Jun. 2003.
- [21] A. Doshi, B. T. Morris, and M. M. Trivedi, “On-road prediction of driver’s intent with multimodal sensory cues,” *IEEE Pervasive Comput.*, vol. 10, no. 3, pp. 22–34, Jul.–Sep. 2011.
- [22] C. Tran and M. M. Trivedi, “Towards a vision-based system exploring 3D driver posture dynamics for driver assistance: Issues and possibilities,” in *Proc. IEEE Intell. Veh. Symp.*, Jun. 2010, pp. 179–184.
- [23] A. Doshi and M. M. Trivedi, “Tactical driver behavior prediction and intent inference: A review,” in *Proc. IEEE Conf. Intell. Transp. Syst.*, Oct. 2011, pp. 1892–1897.
- [24] R. Liu, H. Yu, R. McGee, and Y. L. Murphey, “Driving course prediction for vehicle handling maneuvers,” in *Proc. Amer. Control Conf.*, Jun. 2012, pp. 2096–2101.
- [25] M. Li and T. Zhou, “Trajectory generation based vehicle action tracking for intersection driving safe,” in *Proc. IEEE Intell. Veh. Symp.*, Jun. 2011, pp. 302–307.
- [26] T. Gindele, S. Brechtel, and R. Dillmann, “A probabilistic model for estimating driver behaviors and vehicle trajectories in traffic environments,” in *Proc. IEEE Conf. Intell. Transp. Syst.*, Sep. 2010, pp. 1625–1631.
- [27] B. Morris and M. M. Trivedi, “Trajectory learning for activity understanding: Unsupervised, multilevel, long-term adaptive approach,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 33, no. 11, pp. 2287–2301, Nov. 2011.
- [28] J. Wiest, M. Hoffken, U. Kresel, and K. Dietmayer, “Probabilistic trajectory prediction with Gaussian mixture models,” in *Proc. IEEE Intell. Veh. Symp.*, Jun. 2012, pp. 141–146.
- [29] N. Kuge, T. Yamamura, O. Shimoyama, and A. Liu, *A Driver Behavior Recognition Method Based on a Driver Model Framework*. Warrendale, PA, USA: SAE, Jan. 2000.
- [30] N. Oliver and A. P. Pentland, “Graphical models for driver behavior recognition in a smartcar,” in *Proc. IEEE Intell. Veh. Symp.*, Oct. 2000, pp. 7–12.
- [31] S. Peeta and J. W. Yu, “Adaptability of a hybrid route choice model to incorporating driver behavior dynamics under information provision,” *IEEE Trans. Syst., Man, Cybern. A, Syst., Humans*, vol. 34, no. 2, pp. 243–256, Mar. 2004.
- [32] J. Shotton *et al.*, “Real-time human pose recognition in parts from single depth images,” in *Proc. IEEE Int. Conf. Comput. Vis. Pattern Recognit.*, Jun. 2011, pp. 1297–1304.
- [33] N. Cristianini and J. Shawe-Taylor, *An Introduction to Support Vector Machines and Other Kernel-Based Learning Methods*. Cambridge, U.K.: Cambridge Univ. Press, 2000.
- [34] R. Rajamani, *Vehicle Dynamics and Control*, 1st, New York, NY, USA: Springer-Verlag, Oct. 2005.
- [35] R. Y. Hindiyyeh and J. C. Gerdes, “Equilibrium analysis of drifting vehicles for control design,” in *Proc. ASME Dyn. Syst. Control Conf.*, Oct. 2009, pp. 181–188.
- [36] J. A. Hartigan and M. A. Wong, “A K-means clustering algorithm,” *J. R. Stat. Soc. C, Appl. Stat.*, vol. 28, no. 1, pp. 100–108, 1979.
- [37] A. Bemporad, F. Borrelli, and M. Morari, “Model predictive control based on linear programming—The explicit solution,” *IEEE Trans. Autom. Control*, vol. 47, no. 12, pp. 1974–1985, Dec. 2002.
- [38] E. Brown and B. Kelly, *California Driver Handbook—Safe Driving Practices*. Sacramento, CA, USA: Department of Motor Vehicles, 2013.
- [39] W. Zhang, X. Wang, D. Zhao, and X. Tang, “Graph degree linkage: Agglomerative clustering on a directed graph,” in *Proc. ECCV*, 2012, pp. 428–441.

- [40] E. Elhamifar, G. Sapiro, A. Yang, and S. S. Sastry, "A convex optimization framework for active learning," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2013, pp. 209–216.
- [41] J. Wang, S. Zhu, Y. Gong, and S. Member, "Driving safety monitoring using semisupervised learning on time series data," *IEEE Trans. Intell. Transp. Syst.*, vol. 11, no. 3, pp. 728–737, Sep. 2010.
- [42] P. Jimenez, L. M. Bergasa, J. Nuevo, N. Hernandez, and I. G. Daza, "Gaze fixation system for the evaluation of driver distractions induced by IVIS," *IEEE Trans. Intell. Transp. Syst.*, vol. 13, no. 3, pp. 1167–1178, Sep. 2012.
- [43] E. Ohn-Bar and M. M. Trivedi, "In-vehicle hand activity recognition using integration of regions," in *Proc. IEEE Intell. Veh. Symp.*, Jun. 2013, pp. 1034–1039.
- [44] C. Tran, A. Doshi, and M. M. Trivedi, "Modeling and prediction of driver behavior by foot gesture analysis," *Comput. Vis. Image Understand.*, vol. 116, no. 3, pp. 435–445, Mar. 2012.
- [45] C. Hermes, C. Wahler, K. Schenk, and F. Kummert, "Long-term vehicle motion prediction," in *Proc. IEEE Intell. Veh. Symp.*, 2009, pp. 652–657.
- [46] G. Aoude, J. Joseph, N. Roy, and J. How, "Mobile agent trajectory prediction using Bayesian nonparametric reachability trees," in *Proc. AIAA Infotech Aerosp.*, Mar. 2011, pp. 1–17.
- [47] A. Gray, Y. Gao, J. Hedrick, and F. Borrelli, "Robust predictive control for semi-autonomous vehicles with an uncertain driver model," in *Proc. IEEE Intell. Veh. Symp.*, 2013, pp. 208–213.
- [48] Y. Gao, A. Gray, H. E. Tseng, and F. Borrelli, "A tube-based robust nonlinear predictive control approach to semiautonomous ground vehicles," *Veh. Syst. Dyn.*, vol. 52, no. 6, pp. 802–823, 2014.
- [49] S. Beiker, CARS—Stanford's Automotive Affiliates Program 2012.
- [50] W. Knight, "Driverless cars are further away than you think," MIT Technology Review 2013.



Katherine Driggs Campbell received the B.S.E. degree (with honors) in electrical engineering from Arizona State University, Tempe, AZ, USA, in 2012. She is currently working toward the Ph.D. degree in electrical engineering and computer science at University of California at Berkeley, Berkeley, CA, USA.

She was an Undergraduate Student Researcher with Arizona State University, where her research focused on sensor development. She is currently a Graduate Student Researcher with University of California at Berkeley, where she is working on various control applications. Her research interests include human-in-the-loop control, driver modeling, and automation.



Theresa Lin received the B.E. degree in mechatronics engineering from University of Auckland, Auckland, New Zealand, in 2008. She is currently working toward the Ph.D. degree in the Model Predictive Control Laboratory, University of California at Berkeley, Berkeley, CA, USA.

She is engaged in researching human modeling, particularly the application of control theory to model human decisions and actions in a dynamic environment, as well as the integration of the models into control algorithms. She is also interested in modeling the learning aspect of human behavior.



Francesco Borrelli received the "Laurea" degree in computer science engineering in 1998 from University of Naples "Federico" II, Naples, Italy, and the Ph.D. degree in 2002 from the Swiss Federal Institute of Technology (ETH Zurich), Zurich, Switzerland, working in the Automatic Control Laboratory.

He is an Associate Professor with the Department of Mechanical Engineering, University of California at Berkeley, Berkeley, CA, USA. He is the author of more than 50 publications and in the field of predictive control. He is author of the book *Constrained Optimal Control of Linear and Hybrid Systems* (Springer Verlag), which won the Innovation Prize 2004 from the ElectroSwiss Foundation and won the 2009 the National Science Foundation Faculty Early Career Development (CAREER) award. His research interests include constrained optimal control and model predictive control, and its application to advanced automotive control and energy-efficient building operation.

Prof. Borrelli was appointed as the Chair of the IEEE Technical Committee on Automotive Control in 2008.



Ruzena Bajcsy (M'81–SM'88–F'92) received the M.S. and Ph.D. degrees in electrical engineering from Slovak Technical University, Bratislava, Slovakia, in 1957 and 1967, respectively, and the Ph.D. degree in computer science from Stanford University, Stanford, CA, USA, in 1972.

She is currently a Professor of electrical engineering and computer sciences with University of California at Berkeley (UC Berkeley), Berkeley, CA, USA, where she is the Director Emeritus of the Center for Information Technology Research in the Interest of Science (CITRIS). Prior to joining UC Berkeley, she headed the Computer and Information Science and Engineering Directorate at the National Science Foundation. As a former faculty member of University of Pennsylvania, Philadelphia, PA, USA, she also served as the Director of the University's General Robotics and Active Sensory Perception Laboratory, which she founded in 1978, and chaired the Computer and Information Science Department from 1985 to 1990.

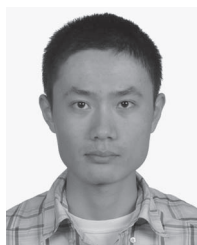
Dr. Bajcsy is a member of the National Academy of Engineering and the National Academy of Science Institute of Medicine, as well as a Fellow of the Association for Computing Machinery (ACM), the Institute of Electronic and Electrical Engineers, and the American Association for Artificial Intelligence. In 2001 she received the ACM/Association for the Advancement of Artificial Intelligence Allen Newell Award, and she was named as one of "The 50 Most Important Women in Science" in the November 2002 issue of *Discover* magazine. In 2008 she received the Benjamin Franklin Medal for Computer and Cognitive Sciences. In 2010 she received the IEEE Robotics and Automation Pioneer award.



Victor A. Shia received the B.S. degree in electrical engineering and computer science and the M.S. degree from the University of California at Berkeley, Berkeley, CA, USA, in 2010 and 2013, respectively, where he is also currently working toward the Ph.D. degree in electrical engineering.

He is a Graduate Student Researcher with the Teleimmersion Laboratory headed by Prof. Ruzena Bajcsy. His research interests include verification of hybrid systems, human-in-the-loop control, semiautonomous vehicles, and robotics.

Mr. Shia received the National Science Foundation Center for Interdisciplinary Bio-Inspiration in Education and Research Integrative Graduate Education and Research Traineeship (CiBER-IGERT).



Yiqi Gao received the joint B.S. degree in mechanical engineering from University of Michigan, Ann Arbor, MI, USA, and Shanghai Jiao Tong University, Shanghai, China, in 2009. He is currently working toward the Ph.D. degree in controls at the University of California at Berkeley, Berkeley, CA, USA.

He is with the Model Predictive Control Laboratory, University of California at Berkeley, where he is researching methods in advanced active safety systems for semiautonomous ground vehicles. His

research interests include theory and implementation of model predictive control on fast and uncertain systems.



Ramanarayan Vasudevan (S'10–M'12) received the B.S. degree in electrical engineering and computer sciences and an honors degree in physics, and the M.S. and Ph.D. degrees in electrical engineering from University of California at Berkeley, Berkeley, CA, USA, in 2006, 2009, and 2012, respectively.

He is a Postdoctoral Associate with Massachusetts Institute of Technology, Cambridge, MA, USA. His research interests include sensor networks, computer vision, hybrid systems, and optimal control.

Dr. Vasudevan received the Regent and Chancellor's Scholarship and was a corecipient of the Innovations in Networking Award presented by the Corporation for Education Network Initiatives in California.