2012 American Control Conference
Fairmont Queen Elizabeth, Montréal, Canada
June 27-June 29, 2012

Safe Semi-Autonomous Control with Enhanced Driver Modeling

Ram Vasudevan, Victor Shia, Yiqi Gao, Ricardo Cervera-Navarro, Ruzena Bajcsy, and Francesco Borrelli

Abstract— **During semi-autonomous driving, threat assessment is used to determine when controller intervention that overwrites or corrects the driver's input is required. Since today's semi-autonomous systems perform threat assessment by predicting the vehicle's future state while treating the driver's input as a disturbance, controller intervention is limited to just emergency maneuvers. In order to improve vehicle safety and reduce the aggressiveness of maneuvers, threat assessment must occur over longer prediction horizons where driver's behavior cannot be neglected. We propose a framework that divides the problem of semi-autonomous control into two components. The first component reliably predicts the vehicle's potential behavior by using empirical observations of the driver's pose. The second component determines when the semi-autonomous controller should intervene. To quantitatively measure the performance of the proposed approach, we define metrics to evaluate the informativeness of the prediction and the utility of the intervention procedure. A multi-subject driving experiment illustrates the usefulness, with respect to these metrics, of incorporating the driver's pose while designing a semi-autonomous system.**

I. Introduction

Despite the improvement of vehicle safety features in recent years, the number of traffic crashes and fatalities remains high [4]. Though it is difficult to quantify why these accidents occurred, it is agreed that in-vehicle information systems (e.g. vehicle navigation systems) and other devices such as cell phones make the driver prone to distraction which degrades driving performance. Using data from vehicles instrumented with cameras, several studies have estimated that driver distraction contributes to between $22 - 50\%$ of all crashes [19], [26]. Alarmingly, a recent report indicates that the percentage of fatal crashes due to distraction is rising [8].

A potential resolution to the driver distraction problem is the autonomous vehicle [20], [28], [29], [31]. Unfortunately the deployment of these systems has been hindered principally due to the lack of formal methods to verify 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 moved to design active safety systems that avoid accidents by performing situation specific threat assessment or driver modeling [7]. Volvo, for example, has developed a system that detects potential collisions by treating the driver's input as a disturbance, computing the reach set of the car over a short time horizon (approximately 500 milliseconds), and immediately decelerating the car if it detects an intersection of the reach set with an obstacle [13]. Confidence in this autonomous breaking maneuver was achieved via extensive testing over several million kilometers.

As these types of semi-autonomous systems become more elaborate and are applied in higher speed situations, threat assessment must occur over a much longer time horizon, which means that the reach set becomes large. In this context, we move from a scenario where the driver can simply be treated as a disturbance, to a scenario where the driver behavior becomes fundamental to the design of a provably safe semi-autonomous vehicle otherwise the semi-autonomous system begins to behave as a fully autonomous system. Importantly, a single universal driver model for all semi-autonomous vehicle systems is insufficient, and we must instead rely on a framework that adapts to different drivers by exploiting real-time empirical observations of the driver. Once this shift occurs, empirical validation becomes difficult due to the reliance of the system on the driver model itself. Fortunately recent advances in numerical optimization techniques have shown that provably safe controllers can be designed rapidly and robustly given a particular circumstance [6], [11]. In this paper, we leverage these latest advances in numerical optimization to design a controller methodology for a semi-autonomous vehicle that only employs a provably safe controller if one exists and if the driver's behavior in closed loop with the vehicle dynamics suggests a potential threat.

Modeling the driver's behavior is critical to our architecture and has been considered in a variety of communities. The human factors community, as described in the surveys found in [10], [23], has focused on quantifying the amount of attention required to accomplish a variety of tasks, and the level of distraction due to a variety of common preoccupations that arise during driving. These studies do not provide a method to predict the driver's behavior given a particular level of distraction and a particular task complexity.

Other communities have attempted to use body sensors on the driver [24], eye trackers [32], or facial tracking [12], [18], [21] to predict the driver's behavior. Unfortunately these approaches do not provide a means to employ the prediction during controller design or a method to quantify the utility of the given prediction in preventing accidents. In this paper, we quantify driver distraction, predict future vehicle state, and couple this prediction with controller

R. Vasudevan, V. Shia, R. Cervera-Navarro, and R. Bajcsy are with the Department of Electrical Engineering and Computer Sciences, University of California at Berkeley, Berkeley, CA 94720 {ramv,vshia,rcervera,bajcsy}@eecs.berkeley.edu

Y. Gao and F. Borrelli are with the Department of Mechanical Engineering, University of California at Berkeley, Berkeley, CA 94720 {yiqigao,fborrelli}@me.berkeley.edu

This work was supported by the National Science Foundation under grants 0903711 and 0931437.

978-1-4577-1096-4/12/$26.00 ©2012 AACC

design to construct a provably safe semi-autonomous vehicle framework and illustrate its utility in preventing accidents during an experiment. Given the complexity of the task at hand, we describe only the portion of the semi-autonomous vehicle framework up to the application of our provably safe controller and do not describe how we eventually return full control of the vehicle back to the driver.

Our contributions are three-fold: first, in Section II we present a novel architecture that divides the task of constructing a provably safe semi-autonomous controller into a component that involves predicting potential vehicle behavior by using empirical observations of the driver's pose and another component that determines when the provably safe controller constructed by the numerical optimization scheme should intervene; second, in Section II-C we present four metrics to quantify the performance of the two aforementioned components; third, in Section IV we construct an experiment and illustrate the utility of incorporating empirical observations of the driver's pose in the control loop. The rest of this paper is organized as follows: Section III describes the implementation of the architecture and Section V concludes the paper.

II. Methodology

In this section, we present the semi-autonomous vehicle framework. A specific example of the various framework components is described in subsequent sections.

A. Semi-Autonomous Vehicle Framework

We are interested in the control of a vehicle whose evolution is given as the solution of the *vehicle dynamics*:

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

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, connected set containing the origin, and $x(0)$ is assumed given. For a fixed horizon $\overline{N} \in \mathbb{N}$ the *constraint function*, $\varphi_{\overline{N}} : \mathbb{R}^n \times \{0, \dots, \overline{N}\} \to \mathbb{R}$, describes the safe part of the environment. That is, the state of the car remains outside of obstacles in the environment at a time k if $\varphi_{\overline{N}}(x(k), k) \leq 0$. 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 into the constraint function, $\varphi_{\overline{N}}$.

The focus of this paper is on the development of a provably safe semi-autonomous vehicle framework by employing driver modeling. To construct such a provably safe controller, we require a safe controller and information about the driver's state. Since there has already been considerable work done on the robust, real-time construction of such a controller using numerical optimization, we prescribe the existence of such a numerical optimization scheme by assumption and describe a specific algorithm that satisfies this assumption in Section IV.

Assumption 1: *Given* $x(0) \in \mathbb{R}^n$, $\overline{N} \in \mathbb{N}$, *and a cost function* $J : \mathbb{R}^n \times \mathcal{U}_{\overline{N}} \times \mathbb{N} \to \mathbb{R}$ *where* $\mathcal{U}_{\overline{N}} = \left\{ \{u(k)\}_{k=0}^{\overline{N}} \mid u(k) \in U, \forall k \in \{0, \dots, \overline{N}\} \right\}$ *assume there exists an algorithm,* $\mathcal{A} : \mathbb{R}^n \times \mathcal{U}_{\overline{N}} \times \mathbb{N} \to \mathcal{U}_{\overline{N}}$, *that computes:*

$$\begin{aligned} &\underset{u \in \mathcal{U}_{\overline{N}}}{\operatorname{argmin}} \; J(x(0), u, \overline{N}) \\ &\text{s.t.} \quad \varphi_{\overline{N}}(x(k), k) \leq 0, \quad \forall k \in \{0, \dots, \overline{N}\} \end{aligned} \tag{2}$$

where $x(k)$ *is a solution to Equation* (1) *using the initial condition* $x(0)$ *and input* u. *If the problem is infeasible, then* $\mathcal{A}(x(0), u, \overline{N})$ *returns* $\emptyset$.

A particular choice of cost function J can penalize control effort or deviation from a specific trajectory.

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 [25]. 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 2: *Given* $\underline{N} \in \mathbb{N}$ *and* $N > 0$ *joints, we assume there exists a function* $\theta_{\underline{N}} : \{-\underline{N}, \dots, 0\} \to SO(3)^N$ *called the* driver state function *which describes the evolution of the joints of the driver where* $SO(3)$ *is the special orthogonal group in dimension* 3.

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 semi-autonomous vehicle framework divides the problem of determining when to intervene into two components. We begin by considering the "simplest" semi-autonomous vehicle architecture which treats the driver's input into the vehicle as a disturbance. The simplest semi-autonomous vehicle architecture, in this instance, would construct a reach set for the vehicle and would apply the output of the numerical optimization scheme if a solution existed and whenever the reach set of the vehicle intersected with an obstacle.

Unfortunately this framework is too conservative since by treating the driver as a disturbance the simplest architecture predicts a large potential set of behavior and therefore acts too aggressively. To illustrate this shortcoming, suppose, for example, that the semi-autonomous controller employed a breaking maneuver and did not want to decelerate the vehicle quickly. In this instance it needs to act over a long time horizon (e.g. more than 2 seconds). The set of reachable states at highway speeds for a 2 second time horizon is large and in all likelihood would intersect with an obstacle all the time (e.g. a vehicle in a neighboring lane) though the driver would most likely never have an accident with this obstacle.

The first component of the proposed two component semi-autonomous vehicle framework addresses this shortcoming, by incorporating the prior observations denoted $\mathcal{O}$ (this may include past states of the driver, past vehicle trajectories, past generated optimal vehicle trajectories from the numerical optimization algorithm) and the current information $\mathcal{I} = \{\overline{N}, \underline{N}, f, x(0), \varphi_{\overline{N}}, \mathcal{U}_{\overline{N}}, J, \mathcal{A}, \theta_{\underline{N}}\}$ in order to predict a smaller and more useful set of potential states for the vehicle. The component can defined formally as:

Component 1: *Fixing $\overline{N}, \underline{N} \in \mathbb{N}$ and choosing $\alpha \in [0,1]$ and $k \in \{0,\ldots,\overline{N}\}$, construct the* vehicle prediction multifunction *denoted $\Delta(\alpha, k, \mathcal{O}, \mathcal{I})$ as the solution to:*

$$\begin{aligned} &\underset{\Delta \subset \mathbb{R}^n}{\operatorname{argmin}} \; |\Delta| \\ \text{s.t.} \quad & P\left((X(k) - x(0)) \subset \Delta \mid \mathcal{O}, \mathcal{I}\right) \geq \alpha, \end{aligned} \tag{3}$$

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

The vehicle prediction multifunction 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. In Section III, we describe how we construct the required conditional distribution and how we construct the vehicle prediction multifunction.

Using the vehicle prediction multifunction and the constraint function, the second component of the semi-autonomous vehicle framework is a binary-valued function that can defined formally as follows:

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

$$g(\alpha, \mathcal{O}, \mathcal{I}) = \begin{cases} 1 & \textit{if } \cup_{k=0}^{\overline{N}} \left(\Delta(\alpha, k, \mathcal{O}, \mathcal{I}) \cap \mathcal{C}(k, \mathcal{I})\right) \neq \emptyset \\ 0 & \textit{otherwise,} \end{cases} \tag{4}$$

where $\mathcal{C}(k, \mathcal{I}) = \{z \in \mathbb{R}^n \mid \varphi_{\underline{N}}(z - x(0), k) > 0\}$.

The vehicle intervention function is 1 if any α probable vehicle trajectory intersects an obstacle in the next $\overline{N}+1$ time steps given our past observations and current information.

Algorithm 1 describes our semi-autonomous vehicle framework. We make two observations. First, notice we do not describe how to apply the intervention procedure which requires not only some form of aiding the driver (i.e. one could just apply the output of the numerical optimization scheme or could restrict control inputs that may directly lead to danger), but also some procedure for returning control to the driver. This is a non-trivial problem, but has been considered by several human factors researchers [15], [16]. Second, notice that since we do not demand the vehicle prediction multifunction to perfectly predict the potential behavior of the vehicle, the driver may get into an accident since the semi-autonomous vehicle framework may not intervene. This means the safety of the semi-autonomous vehicle framework is guaranteed in the sense that when the semi-autonomous controller intervenes it is able to do so in a provably safe manner under the environment model prescribed by $\varphi_{\overline{N}}$.

Algorithm 1 Semi-Autonomous Vehicle Framework

```
Data: $\overline{N}, \underline{N} \in \mathbb{N}$ and $x(0) \in \mathbb{R}^n$
Let $u(k) = 0, \forall k \in \{0,\ldots,\overline{N}\}$.
for Each time step do
    update $\varphi_{\overline{N}}$ and $\theta_{\overline{N}}$, and set $x(0)$ equal to the vehicle's
    current state.
    if $g(\alpha, \mathcal{O}, \mathcal{I}) == 1$ and $\mathcal{A}(x(0), u, \overline{N}) \neq \emptyset$ then
        apply intervention procedure.
    end if
    if $\mathcal{A}(x(0), u, \overline{N}) \neq \emptyset$ then
        set $u = \mathcal{A}(x(0), u, \overline{N})$.
    end if
end for
```

C. Metrics to Evaluate Performance

In this subsection, we define four separate metrics: the *Accuracy of the Vehicle Prediction Multifunction* (AVPM), the *Precision of the Vehicle Prediction Multifunction* (PVPM), the *Recall of the Vehicle Intervention Function* (RVIF), and the *Precision of the Vehicle Intervention Function* (PVIF). The AVPM and the PVPM measure the informativeness of our vehicle prediction multifunction and the RVIF and the PVIF measure the utility of the vehicle intervention function.

The most informative set prescribed by the vehicle prediction multifunction 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 $\overline{N}+1$ time steps was inside of the predicted set at each of the next $\overline{N}+1$ time steps. Explicitly, fixing $\alpha \in [0,1]$ suppose we are given M distinct observed vehicle trajectories for $\overline{N}+1$ time steps denoted $\{x_{i,\overline{N}} : \{0,\ldots,\overline{N}\} \to \mathbb{R}^n\}_{i=1}^{M}$ then:

$$\begin{aligned} \text{AVPM} = \frac{1}{M} \sum_{i=1}^{M} \prod_{k=0}^{\overline{N}} \mathbb{1}\{(x_{i,\overline{N}}(k) + {} \\ - x_{i,\overline{N}}(0)) \in \Delta(\alpha, k, \mathcal{O}, \mathcal{I}_i)\}, \end{aligned} \tag{5}$$

where $\mathbb{1}$ is the indicator function and the current information, $\mathcal{I}_i$, is indexed by observation i since the vehicle prediction multifunction is a function of the initial condition of the vehicle, $x_{i,\overline{N}}(0)$. We also measure the informativeness of the predicted set by computing one minus the area of the predicted set for the next $\overline{N}+1$ time steps over the area of the set of reachable states for the next $\overline{N}+1$ time steps:

$$\text{PVPM} = 1 - \frac{1}{M} \sum_{i=1}^{M} \frac{|\cup_{k=0}^{\overline{N}} \Delta(\alpha, k, \mathcal{O}, \mathcal{I}_i)|}{|\cup_{k=0}^{\overline{N}} \mathcal{R}(k, \mathcal{I}_i)|}, \tag{6}$$

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 vehicle intervention function 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 [14]. 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 vehicle intervention function (i.e. the classifier in our context) returns 1, and we define an observed vehicle trajectory to be "negative" if the vehicle intervention function returns 0.

The 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 semi-autonomous vehicle framework intervened whenever an accident occurred.* Notice that if instead of relying on the vehicle prediction multifunction the vehicle intervention function 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 non-trivial intersection between the reachable set and an obstacle. Therefore, this choice of vehicle intervention function would always be perfect with respect to this metric.

The 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 semi-autonomous 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 vehicle prediction multifunction the vehicle intervention function relied on the set of reachable states, then there would most likely be a large number of false-positive events since the reachable set 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, the driver state function and the vehicle trajectory generated by the numerical optimization algorithm for the current time step in the semi-autonomous vehicle framework by using these four metrics.

III. Vehicle Prediction Multifunction

In this section, we describe how we construct the vehicle prediction multifunction. In order to accomplish this task, we need to construct $P\left((X(k)-x(0)) \subset \Delta \mid \mathcal{O}, \mathcal{I}\right)$. During the experiment considered in Section IV, we want to illustrate the utility of considering various types of prior observations and current information. Therefore, in this section we consider the construction of the required distribution in generality.

We could perform the construction of the required distribution by presuming a certain type of distribution. Unfortunately it is not immediately apparent what the appropriate choice of distribution should be. Another option is to take all of the observed data and construct the empirical cumulative distribution, in which case we make no assumptions about the underlying distribution (see Chapter 19 of [30]). Instead, in this paper, we hypothesize that given a level of distraction for the driver and a specific environment, the distribution of the potential vehicle trajectories at a particular time step has distinct modes. To recover these distinct modes, we apply a k-means clustering algorithm on the prior observations and construct a distinct probability distribution for each cluster using the empirical cumulative distribution defined on the observations within that cluster [17], [27].

Suppose we cluster a set of observations $\mathcal{O} = \{p_i\}_i$. Fixing $\overline{N} \in \mathbb{N}$ with respect to each observation, we can associate the actual observed vehicle behavior for the next $\overline{N}+1$ time steps beginning at time step i which we denote by $x_{i,\overline{N}} : \{0,\dots,\overline{N}\} \to \mathbb{R}^n$. As Algorithm 1 runs at each time step, we are given current information, $\mathcal{I}$, which includes the vehicles current initial condition which we denote by $x(0)$. Letting $\mathcal{M}$ denote the cluster to which $\mathcal{I}$ belongs, for some $z \in \mathbb{R}^n$ we define the probability that the difference between the vehicle trajectory at time step k, denoted $x(k)$, and the vehicle's initial condition $x(0)$ is less than z as:

$$\begin{aligned} P((X(k)-x(0)) \leq z \mid \mathcal{O}, \mathcal{I}) = \\ = \frac{1}{|\mathcal{M}|} \sum_{\forall i \in \mathcal{M}} \mathbb{1}\{(x_{i,\overline{N}}(k) - x_{i,\overline{N}}(0)) \leq z\}, \quad (7) \end{aligned}$$

where $|\mathcal{M}|$ denotes the number of elements in cluster $\mathcal{M}$. Note that when we write $a \leq b$ for $a, b \in \mathbb{R}^n$, we mean that each of the coordinates of a is less than or equal to each of the coordinates of b. During our experiment in Section IV, we fix $\alpha = 1$. Therefore, the vehicle prediction multifunction at time step k is:

$$\begin{aligned} \Delta(1,k,\mathcal{O},\mathcal{I}) = \{z \in \mathbb{R}^n \mid \forall j \in \{1,\dots,n\}, z^j \in \\ [\underline{\delta}^j(k,\mathcal{O},\mathcal{I}), \overline{\delta}^j(k,\mathcal{O},\mathcal{I})]\}, \quad (8) \end{aligned}$$

where

$$\underline{\delta}^j(k,\mathcal{O},\mathcal{I}) = \min_{i \in \mathcal{M}} \{x^j_{i,\overline{N}}(k) - x^j_{i,\overline{N}}(0)\} \quad (9)$$

$$\overline{\delta}^j(k,\mathcal{O},\mathcal{I}) = \max_{i \in \mathcal{M}} \{x^j_{i,\overline{N}}(k) - x^j_{i,\overline{N}}(0)\}, \quad (10)$$

where for $a \in \mathbb{R}^n$ the jth cooordinate of a is denoted by a^j.

IV. Experiment

In this section, we describe our experiment that illustrates the utility of considering the driver pose and the optimal vehicle trajectory generated by the optimal control algorithm while designing a semi-autonomous vehicle framework. We begin by describing our experiment setup. Then, we describe

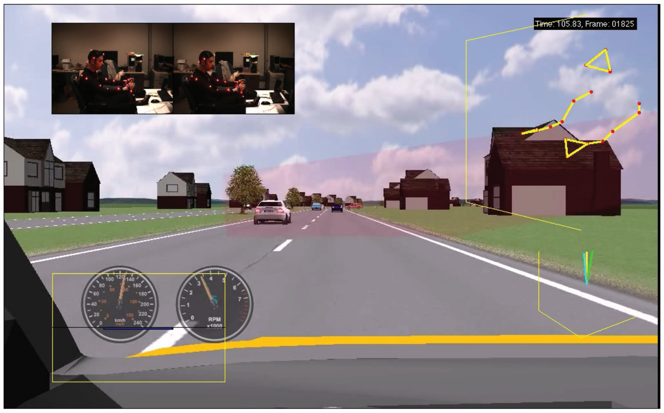

Fig. 1: Simulated environment from the driver's view used in our experiment with data from sensors and stereo camera superimposed. Top-left: Subject and motion capture setup from a stereo camera (left camera and right camera views). Top-right: Human skeleton–head, arms and hands–and markers resulting from the motion capture data. The triangle-like shape on the top is the subject's head. The triangles at the end of the arms represent the driver's hands. Bottom-left: Steering angle (almost zero in this picture).

the vehicle model, the constraint function, and the optimal control algorithm. Finally, we evaluate the performance of our system with respect to the performance metrics described in Section II-C.

A. Experimental Setup

The experimental setup, as illustrated in Figure 1, consists of a driving simulator, an optical motion-capture system, and smartphone to simulate distraction. We use a high end gaming steering wheel and pedals (Logitech G-25) which interfaces with the driving simulator through Simulink. The driving simulator, CarSim 8.02 [1], communicates with a real-time dSpace (DS1006) hardware-in-the-loop over a fiber-optic link for the physics and model computations [2] and works at 50 frames per second. 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 about the global coordinates of the car at each time step, 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 at each time step, the longitudinal and lateral velocities of all of the obstacles at each time step, and a radar sensor telling us which obstacles are observable from the vehicle. The optical motion capture system from PhaseSpace consists of a suit equipped with 20 LED markers and 8 cameras [5] which compute the 3D location of the LED markers at 50 frames per second and is synced with CarSim using a Network Timing Protocol server [3]. For each experiment, the subject wore 4 markers on the head, 5 on each arm, 2 on each hand, and 2 on the right foot. We extract the driver state function from the joint angles constructed from the motion capture data. To simulate a common distraction while driving, we use an Android smartphone with a texting app.

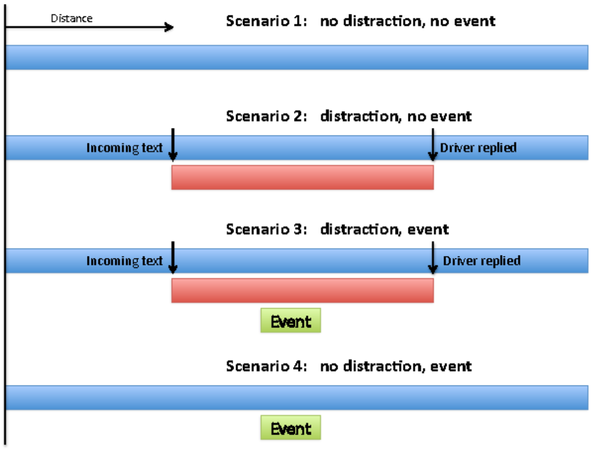
Distance
Scenario 1: no distraction, no event
Scenario 2: distraction, no event
Incoming text
Driver replied
Scenario 3: distraction, event
Incoming text
Driver replied
Event
Scenario 4: no distraction, event
Event

Fig. 2: Simplified diagram illustrating the differences between the 4 different driving scenarios considered in our experiment.

As illustrated in Figure 2, we construct an experiment that consisted of 4, 5-minute scenarios. The purpose of the first 5-minute session (scenario 1) is to get the subject comfortable with the setup. Scenario 1 consists of a 2-lane highway with some traffic. The subject is only asked to keep the vehicle's speed between 65 and 70 miles per hour and to drive safely. The subject is not asked to perform any task in this scenario and no unexpected events occur. Scenarios 2, 3 and 4 occur on a two-way, one lane road. The subject is asked to stay in the lane unless an emergency forces him to depart from the lane. There is traffic in the opposite lane and also in the driver's lane. The traffic in the same lane (except during unexpected events) travels at the same longitudinal velocity as the subject's vehicle at any time, so the subject can never reach those vehicles. This is done to limit the variability of potential interactions.

In scenario 2 no unexpected event occurs. Near the middle of the scenario, a message appears on the screen that instructs the subject to look at the cell phone since there is an incoming message. The subject is expected to pick up the phone, look at the question in the message, answer the question while driving, and return the phone back to its original location. In scenario 3, the subject is instructed to look at the cell phone moments before a low visibility turn. While the subject is typing on the phone and turning, the subject finds a fallen tree in the middle of the road that he has to try to avoid while performing the texting task. In scenario 4, the subject does not have to perform any task, but there is a car traveling 80 meters ahead of the subject's vehicle in the same lane that at about one-third of the scenario's length suddenly reduces its speed to 20 miles per hour. The subject has to try and avoid the car while continuing to drive. At about two-thirds of the scenario length, a cow suddenly appears in the middle of the road, 80 meters ahead of the subject's vehicle. The subject has to try to avoid hitting the cow. In any of the scenarios if the subject crashes, the accident is recorded but the experiment

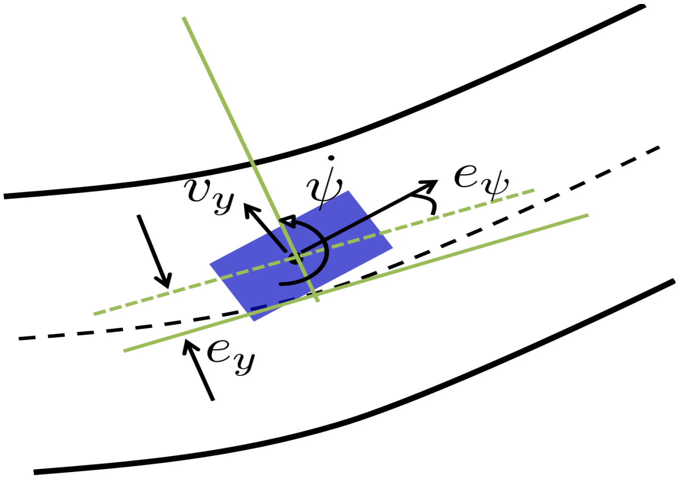

Fig. 3: Illustration of the states of the vehicle model described in Equation (12).

proceeds as if nothing occurred.

B. Vehicle Model, Constraint Function, and Optimal Control

In Algorithm 1, we use an oversimplified vehicle model adapted from [22]. The dynamics of the vehicle within the lane are described using the tracking error with respect to the lane center line. Letting e_y be the lateral position error with respect to the lane center line, e_ψ be the yaw error with respect to the lane center line, $\dot{\psi}$ be the yaw rate of the vehicle with respect to an inertial reference frame, and v_y be the lateral velocity of the vehicle with respect to a coordinate frame attached to the vehicle, the vehicle dynamics, as illustrated in Figure 3, are:

$$\begin{aligned} \dot{x}(t) &= Ax(t) + Bu(t) + h, \\ |u(t)| &\leq u_{max} \end{aligned} \tag{11}$$

where

$$x(t) = \begin{bmatrix} e_y(t) \\ e_\psi(t) \\ v_y(t) \\ \dot{\psi}(t) \end{bmatrix} \quad B = \begin{bmatrix} 0 \\ 0 \\ -\frac{2C_t}{m} \\ \frac{2aC_t}{I} \end{bmatrix} \quad h = \begin{bmatrix} 0 \\ -\frac{v_x}{R} \\ 0 \\ 0 \end{bmatrix} \tag{12}$$

$$A = \begin{bmatrix} 0 & v_x & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & \frac{4C_t}{mv_x} & \frac{2C_t(a-b)}{mv_x} - v_x \\ 0 & 0 & \frac{2C_t(a-b)}{Iv_x} & \frac{2C_t(a^2+b^2)}{Iv_x} \end{bmatrix}.$$

I is the rotational inertia of the vehicle, m is the mass of the vehicle, a is the distance between the vehicle's center of gravity and its front, b is the distance between the vehicle's rear wheels, C_t is the tire stiffness, v_x is the longitudinal velocity, and R is the radius of curvature of the road. The input into the dynamics is the front steering angle denoted u and is bounded by u_{max}. The specific parameter values used in the model during our experiment are summarized in Table I. We then discretize the differential equation in order to construct a difference equation.

In order to construct an algorithm, $\mathcal{A}$, that satisfies Assumption 1, we employ Model Predictive Control (MPC) [9]. In particular, fixing $\overline{N} > 0$, during every iteration of the for loop in Algorithm 1, we choose a quadratic cost function:

$$J(x_0, u, \overline{N}) = \sum_{k=0}^{\overline{N}} \left(\|x(k)\|_2^2 + \|u(k)\|_2^2 \right) \tag{13}$$

that encodes the lane keeping maneuver where $x(k)$ is the solution of the discretized version of Equation (11) beginning at $x(0)$ and where we fix v_x and R as the longitudinal velocity and the road's curvature, respectively, to the outputs of the CarSim model. Observe that the cost function penalizes deviation from the lane center line and control effort.

In order to construct the constraint function, $\varphi_{\overline{N}}$, we consider the obstacles detected from the vehicle by the radar. Assuming that the longitudinal velocity of these visible obstacles stays fixed, we describe their evolution for the next $\overline{N}+1$ time steps. At each time step, we define the constraint function as being less than or equal to 0 when outside of the obstacle's location (which is treated a point) at this time step. Since we satisfy the assumptions required to apply the MPC formalism (see Section 2.1 of [9] and observe that the feasible set can be encoded as a partition of polyhedral sets), we can construct a real-time, robust algorithm, $\mathcal{A}$, that satisfies Assumption 1. In our experiment, when we set $\overline{N} = 100$, corresponding to a 2 second horizon, each iteration of our unoptimized MATLAB implementation of the RHC took approximately 100 milliseconds on a dual-core 2.2 GHz, 8 GB machine.

C. Evaluation

In this subsection, we describe how we analyze the experimental data. We ran the experiment on 6 individuals. For each subject, we allowed the k-means algorithm to train using data from scenario 3 and tested our performance on the data from scenarios 2 and 4.

In order to illustrate the utility of our approach, we considered 3 different implementations of the vehicle prediction multifunction. In *Model 1*, we ignored all prior observations and defined the vehicle prediction multifunction as the reachable set. This serves as an illustration of the baseline performance of existing active safety systems with respect to the AVPM and PVPM. In *Model 2*, we constructed the vehicle prediction multifunction by clustering the data corresponding to the vehicle's trajectory for the previous $\underline{N}+1$ time steps ending at time step i. That is, in Equation (8) we set $p_i = x_{i,\underline{N}} : \{-\underline{N}, \ldots, 0\} \rightarrow \mathbb{R}^n$. This is meant to illustrate the utility of building a vehicle prediction multifunction using prior observations without incorporating any observation of the current driver state or the current optimal trajectory.

Parameter	a	b	m	I	C_t	u_{max}
Value	1.432	1.472	2050	3344	-53400	20
Units	m	m	kg	$kg \cdot m^2$	N	$^\circ$

TABLE I: Model parameters for the vehicle dynamics described in Equation (12).

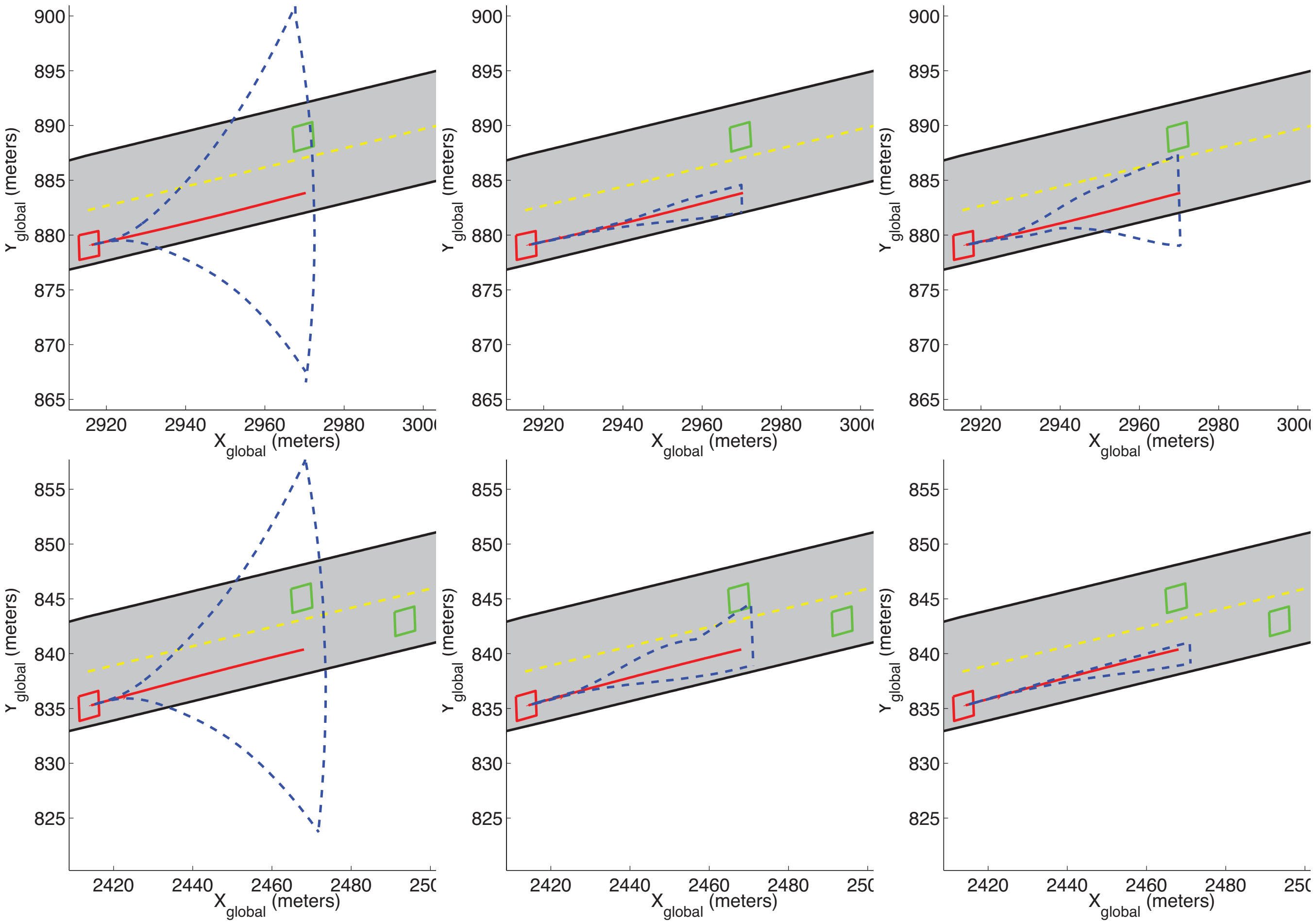

Fig. 4: Sample prediction sets at the same instant in time for two different scenarios (each row) for Model 1 (left column), Model 2 (middle column), and Model 3 (right column) the limits of which are drawn in dotted blue around a subject's vehicle drawn in red in an environment with other vehicles drawn in green. The scenario in the top row is taken from scenario 2 and the driver in the time step depicted is responding to a text. The scenario in the bottom row is taken from scenario 4 where the driver is not distracted.

	AVPM	PVPM	RVIF	PVIF
Model 1	1	0	1	0.0031
Model 2	0.78	0.80	1	0.52
Model 3	0.81	0.82	1	0.68

TABLE II: The performance of the 3 model semi-autonomous vehicle frameworks with respect to the 4 metrics

Finally in *Model 3*, we constructed the vehicle prediction multifunction by clustering the data corresponding to the optimal vehicle trajectory generated for the next $\overline{N}+1$ time steps beginning at time step i and the driver state function from the previous $\underline{N}+1$ time steps ending at time step step i. That is, in Equation (8) we set $p_i = (x^*_{i,\overline{N}} : \{0,\dots,\overline{N}\} \to \mathbb{R}^n, \theta_{i,\underline{N}} : \{-\underline{N},\dots,0\} \to SO^N(3))$. This is meant to illustrate the utility of incorporating the prior observations, the driver state function, and the optimal trajectory in the semi-autonomous vehicle framework. For each model we fixed the number of clusters chosen by the k-means algorithm to 10, set $\overline{N} = 100$, and set $\underline{N} = 100$. The total number of observations marked as "true" (i.e. the observed vehicle trajectory in $\overline{N+1}$ time steps actually got into an accident) for scenarios 2 and 4 combined for all participants was 37.

Figure 4 illustrates the vehicle prediction multifunction and the actual observed trajectory taken by the driver for each of the 3 models at two instances in time. The performance of each of the model semi-autonomous vehicle frameworks with respect to our four metrics is illustrated in Table II. It is immediately clear that by incorporating some information about the driver (i.e. Models 2 and 3), we can construct a precise vehicle prediction multifunction without degrading the accuracy considerably.

To illustrate the baseline performance with respect the RVIF and PVIF, suppose we constructed a vehicle intervention function that acted randomly and with probability one-half chose to intervene and with probability one-half chose to not intervene. This random vehicle intervention function's expected RVIF in this experiment would be 0.5, and its expected PVIF in this experiment would be 0.00010. Model 1 performs better than this random vehicle intervention function, but it still intervenes on the normal operation of the vehicle far more often than necessary. Models 2 and 3 perform considerably better than Model 1 and the random chance model since they almost only ever intervened when necessary. This demonstrates in our experiment the utility of incorporating prior observations, the driver state function, and the optimal vehicle trajectory in preventing accidents while not taking over control of the vehicle more often than necessary.

V. Conclusion

In this paper, we present a novel architecture for a provably safe semi-autonomous controller that predicts potential vehicle behavior and intervenes when the vehicle is deemed in trouble. Using four metrics that we define, we show on a multi-subject experiment that incorporating human pose during the construction of the semi-autonomous controller allows for better performance than a semi-autonomous controller that treats the driver as a disturbance.

In order to realize this framework in practical situations, we must relax our environmental assumptions on future positions of objects, employ a more sophisticated model for the vehicle, construct a method to actually apply the intervention, and implement a system in a vehicle that can perform real-time robust articulated human tracking with guarantees on its accuracy and speed. In practice if we are only confident of the location of certain obstacles with a given probability then we can apply the model predictive control algorithm on the probable obstacle location and modify our vehicle intervention function to only be 1 if the probable location of obstacles intersects with our prediction. We are working on performing these extensions and applying our semi-autonomous framework on longer more elaborate experiments with more subjects.

References

[1] CarSim Mechanical Simulation. http://www.carsim.com.

[2] dSPACE Inc. http://www.dspaceinc.com.

[3] Network Time Protocol. http://www.ntp.org.

[4] NHTSA Data Website. http://www.fars.nhtsa.dot.gov/Main/index.aspx.

[5] PhaseSpace. http://www.phasespace.com.

[6] S. Anderson, S. Peters, T. Pilutti, and K. Iagnemma. An optimal-control-based framework for trajectory planning, threat assessment, and semi-autonomous control of passenger vehicles in hazard avoidance scenarios. *International Journal of Vehicle Autonomous Systems*, 8(2):190–216, 2010.

[7] G. Aoude, B. Luders, K. Lee, D. Levine, and J. How. Threat assessment design for driver assistance system at intersections. In *13th International IEEE Conference on Intelligent Transportation Systems*, pages 1855–1862, 2010.

[8] D. Ascone, T. Lindsey, and C. Varghese. An examination of driver distraction as recorded in NHTSA databases. *Online, September*, 2009.

[9] M. Baotic, F. Borrelli, A. Bemporad, and M. Morari. Efficient on-line computation of constrained optimal control. *SIAM Journal on Control and Optimization*, 47(5):2470–2489, 2008.

[10] D. Basacik and A. Stevens. Scoping study of driver distraction. *Transport Research Laboratory. Road Safety Research Report*, 95, 2008.

[11] A. Bemporad, F. Borrelli, and M. Morari. Model predictive control based on linear programming – the explicit solution. *IEEE Transactions on Automatic Control*, 47(12):1974–1985, 2002.

[12] L. Bergasa, J. Nuevo, M. Sotelo, R. Barea, and M. Lopez. Real-time system for monitoring driver vigilance. *IEEE Transactions on Intelligent Transportation Systems*, 7(1):63–77, 2006.

[13] E. Coelingh, L. Jakobsson, H. Lind, and M. Lindman. Collision warning with auto brake: a real-life safety perspective. *Innovations for Safety: Opportunities and Challenges*, 2007.

[14] N. Cristianini and J. Shawe-Taylor. *An introduction to Support Vector Machines and other kernel-based learning methods*. Cambridge Univ Press, 2000.

[15] B. Donmez, L. Boyle, and J. Lee. The impact of distraction mitigation strategies on driving performance. *Human Factors: The Journal of the Human Factors and Ergonomics Society*, 48(4):785–804, 2006.

[16] B. Donmez, L. Boyle, and J. Lee. Safety implications of providing real-time feedback to distracted drivers. *Accident Analysis & Prevention*, 39(3):581–590, 2007.

[17] J. Hartigan and M. Wong. Algorithm as 136: A k-means clustering algorithm. *Journal of the Royal Statistical Society. Series C (Applied Statistics)*, 28(1):100–108, 1979.

[18] M. Jabon, J. Bailenson, E. Pontikakis, L. Takayama, and C. Nass. Facial expression analysis for predicting unsafe driving behavior. *IEEE Transactions on Pervasive Computing*, 2010.

[19] S. Klauer. *The Impact of Driver Inattention on Near-crash/crash Risk: An Analysis Using the 100-Car Naturalistic Driving Study*. National Highway Traffic Safety Administration, 2006.

[20] H. Kress-Gazit, D. Conner, H. Choset, A. Rizzi, and G. Pappas. Courteous cars. *IEEE Robotics & Automation Magazine*, 15(1):30–38, 2008.

[21] N. Oliver and A. Pentland. Graphical models for driver behavior recognition in a smartcar. In *Proceedings of the IEEE Intelligent Vehicles Symposium*, pages 7–12, 2000.

[22] R. Rajamani. *Vehicle Dynamics Control*. Birkhauser, 2006.

[23] M. Regan, J. Lee, and K. Young. *Driver distraction: Theory, effects, and mitigation*. CRC Press, 2008.

[24] A. Sathyanarayana, S. Nageswaren, H. Ghasemzadeh, R. Jafari, and J. Hansen. Body sensor networks for driver distraction identification. In *IEEE International Conference on Vehicular Electronics and Safety*, pages 120–125, 2008.

[25] J. Shotton, A. Fitzgibbon, M. Cook, T. Sharp, M. Finocchio, R. Moore, A. Kipman, and A. Blake. Real-time human pose recognition in parts from single depth images. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2011.

[26] J. Stutts and W. Hunter. Driver inattention, driver distraction and traffic crashes. *ITE Journal*, 73(7):34–45, 2003.

[27] R. Subbarao and P. Meer. Nonlinear mean shift over riemannian manifolds. *International Journal of Computer Vision*, 84(1):1–20, 2009.

[28] S. Thrun, M. Montemerlo, H. Dahlkamp, D. Stavens, A. Aron, J. Diebel, P. Fong, J. Gale, M. Halpenny, G. Hoffmann, et al. Stanley: The robot that won the darpa grand challenge. *The 2005 DARPA Grand Challenge*, pages 1–43, 2007.

[29] C. Urmson, J. Anhalt, D. Bagnell, C. Baker, R. Bittner, M. Clark, J. Dolan, D. Duggins, T. Galatali, C. Geyer, et al. Autonomous driving in urban environments: Boss and the urban challenge. *Journal of Field Robotics*, 25(8):425–466, 2008.

[30] A. Van der Vaart. *Asymptotic statistics*. Number 3. Cambridge University Press, 2000.

[31] P. Varaiya. Smart cars on smart roads: problems of control. *IEEE Transactions on Automatic Control*, 38(2):195–207, 1993.

[32] E. Wahlstrom, O. Masoud, and N. Papanikolopoulos. Vision-based methods for driver monitoring. In *IEEE Proceedings on Intelligent Transportation Systems*, volume 2, pages 903–908, 2003.