Minimizing Driver Interference Under a Probablistic Safety Constraint in Emergency Collision Avoidance Systems

Jeff Johnson Yajia Zhang Kris Hauser
School of Informatics and Computing, Indiana University at Bloomington
{jj56,zhangyaj,hauserk}@indiana.edu

***Abstract*— Automated emergency maneuvering systems can avoid or reduce the severity of collisions by taking control of a vehicle away from the driver during high-risk situations. To do so, these systems must be capable of making certain decisions: there is the choice of *when* to interfere in the driver's control, which is made challenging in the presence of dynamic obstacles and uncertain information (such as that provided by imperfect sensors), and there is the choice of *how* to interfere, since, in general, controls that mitigate collisions will not be unique. We address both of these questions with a probabilistic decision threshold framework that overrides the driver's control only when safety drops beneath a problem-specific threshold, and minimizes the magnitude of the interference. We demonstrate its application to two scenarios: collision-imminent braking for obstacles traveling along the vehicle's path, and braking and accelerating for unprotected lane crossings.**

I. Introduction

Over 6.3 million automobile accidents occurred in the U.S. in 2007, including 1.8 million injury crashes and 37,435 fatalities at a cost of hundreds of billions of dollars [1]. Although the numbers of injuries and fatalities per traveled mile have decreased significantly over the last four decades due to advances in vehicle safety equipment and control systems, these numbers have leveled off over the last two decades. Further gains in safety could be had by reasoning not only the state of the vehicle itself, but the other vehicles on the road. Semiautonomous active safety systems that incorporate information about other vehicles in order to detect emergency situations are one promising approach.

These systems must distinguish between emergency scenarios and assess collateral impact of collision-mitigating or collision-avoidance strategies when deciding whether to interfere with the driver's control. Additionally, the effect these systems have on the driver's habits must be considered. In the literature for in-vehicle collision avoidance warning systems (IVCAWS), it has been observed that a system considered to be a nuisance might simply be disabled by the driver [2], [3]; on the other hand, too much automation can lead to inattentive or risk-seeking behavior behind the wheel [4]. Worse, automated collision avoidance systems that brake harshly can startle the driver and may cause them to lose control [5]. Hence, it is important to design semiautonomous safety systems to *minimize interference* to the driver in normal driving conditions. Moreover, semiautonomous systems must deal with the major challenge of uncertainty in the driving environment. Uncertainty arises due to noisy sensor readings from lidar, radar, or vision; unknown behavior of other vehicles and pedestrians; errors in speedometer readings due to tire wear and environmental factors; and unknown stopping time due to brake wear and road surface characteristics. In this paper we introduce a framework that addresses both driver interference and environmental uncertainty in a unified fashion. The *safety-constrained interference minimization principle* (SCIMP) formulates the problem as a probabilistically-constrained optimization to interfere minimally with driver control while maintaining at least an specified level of safety α. In other words, safety is treated as a hard probabilistic constraint and driver interference is minimized as a soft constraint.

We demonstrate the application of SCIMP to collision avoidance in two scenarios: rear-end collisions during single-lane driving, and transverse collisions during unprotected intersection crossings. Our framework can be applied to general control problems, but we demonstrate it here with strictly longitudinal controls. The intersection case poses a unique challenge for emergency maneuvering systems because acceleration may need to be employed in addition to braking to clear the lane of an oncoming vehicle. In both cases we demonstrate that the SCIMP control can be calculated quickly using a tractable approximation. The α parameter allows the system to be tuned to trade off between two performance metrics — collision severity vs. driver interference. Experiments suggest that SCIMP is safer than systems that do not consider uncertainty in their decision-making, and can moreover be tuned easily to provide desired levels of safety-interference tradeoff.

II. Related Work

Active safety systems take control of the vehicle only during an emergency or when a potential accident is foreseen in order to mitigate or avoid the consequences of an accident. A longitudinal collision-mitigation braking strategy was described by Hillenbrand et al (2006) that gradually applies stronger braking as the collision boundary is approached, which smoothes the control output and copes some uncertainty [6]. Anderson et al (2009) present a 2D hazard avoidance scheme based on model predictive control which allows varying levels of autonomy based on risk assessed by control magnitudes [7]. Our approach introduces the additional considerations of uncertainty which provides a more natural definition of risk. Karlsson et al. (2004)

introduced a statistical decision rule that applies the brake if the probability of impact is greater than some threshold α_{safe} [8]. This approach is advantageous in the presence of uncertainty. Similar thresholding techniques were applied to autonomous driving in environments mapped using 2D range finders [9]. The SCIMP framework presented in this work generalizes the probabilistic threshold approach to treat driver inputs and many types of environmental uncertainty in a unified manner. Our work deals with both acceleration and braking, and we are able to handle multiple moving obstacles under uncertainty by employing a method that we explored in prior work [10] as a subroutine.

III. Safety-Constrained Interference Minimization Principle

At every time step t the vehicle is given the user's desired control u_t^d and sensor input z_t. Using z_t it infers a joint belief distribution over hypothetical agent and obstacle system states $bel(x_t)$. Although our model can be generalized to two-dimensional motion with steering and velocity control, here we will only consider the longitudinal control problem in which the vehicle travels in a one dimensional space along a known path, which may be curved or straight. The output of the system is a continuous control $u \in \mathcal{U} = [-1, 1]$, where $u = 0$ indicates no control, $u = -1$ indicates maximum braking, and $u = 1$ indicates maximum acceleration. SCIMP can be extended in a straightforward way to handle steering as a second control variable. This leads to a more computationally challenging optimization, but the underlying principle remains the same.

We define the *safety-constrained interference minimization* control $u_t^\star$ at probability $\alpha_{\text{safe}} \in [0, 1]$ as the result of the following optimization:

$$\begin{aligned} u_t^\star &= \arg\min_{u \in \mathcal{U}} |u - u_t^d| \\ \text{s.t.} \quad & P(\text{safe}|u_t = u) \geq \alpha_{\text{safe}} \end{aligned} \tag{1}$$

where $P(\text{safe}|u_t = u)$ is the probability that the system remains safe under $bel(x_t)$ given the choice of u at the current time step t and the *safest* policy thereafter. If no such u meets the α_{safe} threshold, we set

$$u_t^\star = \arg\max P(\text{safe}|u_t = u). \tag{2}$$

Under this formulation both safety and driver interference can be tuned in a problem-specific way using a single parameter α_{safe}.

In this framework, the resulting control $u_t^\star$ satisfies the following properties:

- The user's control will be replicated *exactly* ($u_t^\star - u_t^d$) if there exists a future sequence of controls that is safe with probability α_{safe}.
- If the user's control is such that no sequence exists, then $u_t^\star$ will be the closest control to u_t^d such that the α_{safe} threshold is achieved.
- If no policy can meet the α_{safe} threshold for whatever reason, the safest control is chosen regardless of the user's input.

A. Implementation of SCIMP

The major challenge in implementing SCIMP is evaluating $P(\text{safe}|u_t = u)$ because it requires solving a stochastic optimal control problem. In general, these types of problems are intractable [11], a property which can be attributed to the fact that the belief space that must be searched typically grows exponentially in the number of states. To address the intractability of the problem we make the assumption that the probability can be approximated by integrating over the optimal hypotheses evaluated under $bel(x_t)$ by *assuming that the underlying state hypothesis is true*. In other words $P(\text{safe}|u_t = u) \approx \tilde{P}(\text{safe}|u_t = u)$ where:

$$\tilde{P}(\text{safe}|u_t = u) = \int_x S(x, u) bel(x_t = x) dx \tag{3}$$

where $S(x, u)$ is an indicator function that evaluates whether the system can remain safe under known state x and initial control u. This approximation is useful because it becomes better as uncertainty decreases and/or sensing provides more accurate information. Furthermore, $S(x, u)$ is a deterministic problem that can be solved using optimal control or analytical techniques, as seen fit for the given scenario. This formulation allows us to use a sampling-based approach to the problem, and such approaches are have been shown to often work very well in practice [12].

The integral in (3) can now be estimated using Monte-Carlo integration by sampling n state hypotheses $x^{(1)}, \ldots, x^{(n)}$ independently at random according to $bel(x_t = x)$ and evaluating $S(x^{(i)}, u)$. Using a Bayesian interpretation we rigorously determine the value of n required to determine the probability that $\tilde{P}(\text{safe}|u_t = u) \geq \alpha_{safe}$. Suppose k samples are found to satisfy $S(x^{(i)}, u) = 1$. The outcomes of each test are viewed as coin flips S_i for $i = 1, \ldots, n$ from a Bernoulli distribution with underlying probability of success $\theta \equiv \tilde{P}(\text{safe}|u_t = u)$. Let the results of these flips be the data D.

We assume the prior over θ is given as a Beta distribution $Beta(\theta; a, b)$ where a and b are hyperparameters that indicate the prior belief that a control is safe. (These can be tuned to reflect varying degrees of optimism with $a > b$ or pessimism with $a < b$, but we use the uninformed prior $a = b = 1$). Given the information D that k of n samples are safe, the posterior on θ is $Beta(\theta; a + k, b + n - k)$. Then, the probability that the true state x_t is safe is the expectation of θ, namely $(a + k)/(a + b + n)$. A convenient way to reduce the number of free variables is to choose $k = n$, so that all samples must be safe. The necessary value of n can then be determined to be $n = \lceil \frac{\alpha(a+b)-a}{1-\alpha} \rceil$.

The next implementation issue is how to optimize (1) subject to the probabilistic constraint. One simple method would be to sample many u's (for example, in a grid), sort them in increasing order of $|u - u_t^d|$, then return the first that satisfies $\tilde{P}(\text{safe}|u) \geq \alpha_{\text{safe}}$. This is potentially computationally expensive, especially for large n. Instead, for certain scenarios it may be possible to extend the deterministic optimal control calculation so that it yields the *set* of safe controls $\mathcal{U}_{\text{safe}}(x) = \{u \mid S(x, u) = 1\}$. Given such a procedure, the SCIMP control can be found by sampling n hypothetical states and solving the problem

$$\begin{aligned} u_t^\star &= \arg\min_{u \in \mathcal{U}} |u - u_t^d| \\ \text{s.t. } u &\in \cap_{i=1}^{n} \mathcal{U}_{\text{safe}}(x^{(i)}) \end{aligned} \tag{4}$$

In particular, if the set of feasible controls is convex, then the feasible region is convex as well and convex optimization approaches can be employed.

IV. Application Scenarios

We apply SCIMP to two scenarios: 1) rear-end collisions along a single-lane, and 2) transverse collisions during lane crossing. These are two of the most significant sources of automobile accidents involving elderly drivers [13]. In both cases we present the implementation of the SCIMP subroutine $S(x, u)$ for evaluating the safety of a control beginning at u and behaving optimally for all future controls, starting in a deterministic state x. Furthermore we present subroutines to calculate the set of safe controls $\mathcal{U}_{\text{safe}}(x)$. The rear-end collision scenario in particular lends itself to convenient optimization because $\mathcal{U}_{\text{safe}}(x)$ has a simple form. The lane crossing scenario is more challenging because acceleration may be needed in addition to braking, and multiple dynamic obstacles may need to be tracked. We employ a significant prior development that furnishes an optimal, exact, polynomial-time planner that is used to compute $S(x, u)$ and $\mathcal{U}_{\text{safe}}(x)$ in these scenarios [10].

A. Collision Imminent Braking

We assume the vehicle is moving in the same direction as the obstacle and the obstacle does not move in reverse. We assume that the vehicle is equipped with a speedometer and a range finder (e.g., radar or lidar). The behavior of an obstacle is considered as a black box, and the vehicle needs to infer whether an obstacle is still, accelerating, or braking through the information received through its sensors. In order to synthesize this information, we suppose the car runs an Extended Kalman Filter (EKF), which acts as its perception algorithm, to track a belief distribution over the system state [14]. The belief distribution produced by the EKF is then incorporated into the probabilistic decision rule.

1) Stochastic Dynamics Model: The state of the car can be described using car's position p_c, velocity v_c and the maximum deceleration $\underline{a}$ that the car currently can apply. Additionally, when an obstacle is present, the obstacle position p_o, velocity v_o, and acceleration a_o are modeled as part of the system state. The vehicle receives a noisy speedometer v_s and range reading d. Decisions are made at a time step of Δt (0.1 s in our implementation). The dynamics and sensing are stochastic, with conservative noise parameters in Table I.

2) Extended Kalman Filter: The vehicle is assumed to employ an extended Kalman filter (EKF) in order to estimate the state from the stochastic dynamics and observations. An EKF is a version of the Kalman filter that addresses nonlinear systems by linearizing about the estimated mean and covariance [15]. The dynamics at time step t can be written in the following form:

$$x_{t+1} = f(x_t, u_t) + w_t \tag{5}$$

$$z_t = h(x_t) + v_t \tag{6}$$

$$w_t \sim \mathcal{N}(0, Q_t) \tag{7}$$

$$v_t \sim \mathcal{N}(0, R_t) \tag{8}$$

Here, x_t denotes the state $(p_c, v_c, a_{c_{max}}, p_o, v_o, a_o)$, u_t denotes the braking control input, z_t is the observation (d, v_s) at time step k. w_t is the process error term with Q_t as its covariance matrix. v_t is the measurement noise term with R_t as its covariance matrix.

TABLE I
DYNAMICS AND OBSERVATION MODELS

Road Surface $\dot{a}_c$	Max. applicable acceleration is a random walk $\dot{a}_c \sim \mathcal{N}(0, \Delta t)$
Actuation Errors $\dot{v}_c$	Proportional to control and max. deceleration, $\dot{v}_c = ua_c(1 + e_u)$, with $\epsilon_u \sim \mathcal{N}(0, 0.01^2)$
Object Behavior $\dot{a}_o$	Random noise with 99.99% within $[-5.0\,m/s^2, 5.0\,m/s^2]$, $\dot{a}_o \sim \mathcal{N}(0, 1.25^2)$
Speedometer v_s	Multiplicative noise on actual velocity, $v_s = v_c(1 + \epsilon_s), \epsilon_s \sim \mathcal{N}(0, 0.025^2)$ (99.99% within 10% of current velocity)
Range Reading d	Combined linear and multiplicative noise $d = n_{dL} + (p_o - p_c)(1 + n_{dM})$, with $n_{dL}, n_{dM} \sim \mathcal{N}(0, 0.0125^2)$ (99.99% within by $5\,cm$ + 5% of true distance)

At each step, the EKF maintains a state estimate $\hat{x}_t$ and covariance matrix P_t. Upon reading the observation z_t from the vehicle's sensors, the EKF performs a Kalman update using the system linearized about $\hat{x}_t$ to obtain a new state estimate $\hat{x}_{t+1}$ and covariance P_{t+1}.

Because obstacles may appear and disappear from the range sensor reading, the obstacle state and distance measurements are included in the EKF update only when an obstacle is detected. When an obstacle appears for the first time, its position estimate is initialized to the raw range sensor estimate $\mathcal{N}(d, (0.0125d)^2)$. Its velocity is initialized to a broad distribution $\mathcal{N}(\hat{v}_c/2, (\hat{v}_c/2)^2)$, and its acceleration is initialized to $\mathcal{N}(0, (2.5\,m/s^2)^2)$.

While the EKF suffers from problems in highly nonlinear systems, in our case the system is close to linear and the EKF seems to provide sufficiently accurate performance. Regardless our decision-making algorithms still apply to more general state estimators, like particle filters.

3) Known-State Braking Policy: Given perfect state information, the optimal braking policy is essentially trivial. The optimal braking policy $\pi_D(x)$ is a bang-bang control given in Algorithm 1.

Here, C is a constant that is used for indicating a nominal safety margin, which we set to 1 m. p'_c is the estimated stopping position of the car if it initiates maximum braking. p'_o defines the estimate position of the obstacle when the car stops, and it will either stop after the car does (first conditional branch) or before (second branch). If $p'_c > p'_o$,

Algorithm 1 Bang-Bang Braking Policy

$p_c{}' \leftarrow p_c + v_c{}^2/(-2a_{c_{max}}) + C$
$t' \leftarrow v_c/(-a_{c_{max}})$
if $v_o + a_o t' \geq 0$ **then**
  $p_o{}' \leftarrow p_o + v_o t' + 1/2 a_o t'^2$
else
  $p_o{}' \leftarrow p_o + v_o{}^2/(-2a_o)$
end if
return $u = -1$ if $p_c' > p_o'$, otherwise **return** $u = 0$

there will be a collision between car and obstacle, otherwise, no collision.

Using this policy we can implement $S(x, u)$ and (4) in a straightforward manner. Note that stronger braking is always guaranteed to be safer, so for each state sample $x^{(k)}$ all controls lower than $\pi_D(x)$ are safe. So, the SCIMP optimization is reduced to 1) testing if the user's control is safe, and if not, 2) finding the weakest braking control that keeps the system safe with probability α. In other words, we find $\max_{i=1,\ldots,n} \pi_D(x^{(k)})$.

B. Intersection Crossing

Intersection crossing requires consideration of both braking as well as acceleration in order to avoid crossing too slowly. It also requires considering the behavior of multiple obstacles which makes optimal decision boundaries more complex even in the known-state case. Nevertheless, SCIMP applies directly once we have furnished $S(x, u)$ and $\mathcal{U}_{safe}(x)$. Here we employ a planning algorithm that analytically computes the safety of a given control $S(x, u)$ in the presence of multiple-lane intersection crossings. Because the complete description of the algorithm is rather complex, we only summarize the technique here.

1) Assumptions and Problem Statement: In this problem the vehicle is asssumed to travel along a known path, which can be straight as in street crossings or curved as in unprotected lefthand turns. Acceleration and velocity are assumed bounded. Thus, the vehicle's state is defined as the time t, arc-length parameterized position p, and tangential velocity v along the path, and we assume the following velocity- and acceleration- bounded dynamics:

$$\dot{p} = v \tag{9}$$

$$v \in [\underline{v}, \overline{v}] \tag{10}$$

$$\dot{v} \in [\underline{a}, \overline{a}] \tag{11}$$

with $\underline{v} \geq 0$ and $\underline{a} < 0 < \overline{a}$. The car begins at state (p_t, v_t) and ends a lane-crossing maneuver at final position p_T with a range of admissible final velocities $[\underline{v_T}, \overline{v_T}]$.

The joint state x contains both the (p, v, t) variables of the vehicle as well as the position, velocity, and acceleration of each sensed moving obstacle O_i, $i = 1, \ldots, n$. Each O_i that crosses the vehicle's path can be interpreted as generating a *forbidden region* PO_i in the path-time space (p, t). These PT-obstacles can be constructed by examining the set of (p, t) points that would cause the vehicle to overlap O_i [16]. This requires that the length and width of O_i are known and that future behavior of O_i is known. We currently assume that each O_i travels along a given path with known velocities and accelerations. With these assumptions each PT-obstacle PO_i can be approximated by a rectangular bound in (p, t)-space. The problem is now one of finding a trajectory in (p, v, t) space that satisfies the dynamic constraints, completes the lane-crossing maneuver, and avoids all PT obstacles.

2) Visibility graph formulation: Kant and Zucker (1986) showed how the version of this problem with velocity bounds but without acceleration constraints can be addressed by a visibility graph formulation in the two-dimensional path/time (PT) space [17]. Our recent work in Johnson and Hauser (2012) extended this visibility graph formulation to consider acceleration constraints as well in the path/velocity/time (PVT) space [10]. Once the visibility graph is constructed, an optimal trajectory can be found traversing the graph backwards from the goal region to the initial state. If no such trajectory is found, then no feasible path exists. So, we can calculate the value of $S(x, u)$ simply by estimating the state of the world x' at time $t+\Delta t$ should the vehicle execute the control u, and run the planner starting from x'.

3) Efficient SCIMP optimization: The visibility graph data structure is useful because we can propagate visibility backwards from the goal to the origin to compute the set of velocities at the origin that also admit a path to connect to the goal. We can also compute the set of velocities at time $t+\Delta t$ that are simultaneously reachable from the current state and can connect to the goal. This allows us to incrementally compute the SCIMP control as more hypothetical state samples $x^{(k)}$ are drawn using (4). We describe this procedure in more detail below.

Recall that we wish to compute the control that is closes to u^d and is safe for all state samples $x^{(k)}$. If no such control exists, we wish to compute the control that is maximally safe. So, as we consider more samples, we incrementally grow the PT-obstacles PO_i, $i = 1, \ldots, n$ to contain the corresponding obstacles from all previous samples *for which a safe control was found*. A safe control found for this set is then guaranteed to be safe for all prior state samples. So, we maintain the closest u to u_d that is safe throughout the optimization. First, we initialize $u = u_d$. Then we grow the PT obstacles, compute a visibility graph, and then compute the set F of feasible, reachable velocities at time $t + \Delta t$. If F is nonempty, then we find the closest control u' in F to u and set u to u'. This process is repeated for all samples and the final control u is sent to the vehicle.

V. Evaluation

A good emergency safety system should be able to achieve low collision risk and low driver interference. To evaluate performance we consider the characteristics of collision velocity (CV) and an *interference index* that combines several aspects of driver interference. The interference index (II) attempts to measure deviation from the driver's desired behavior and is a function of the following metrics:

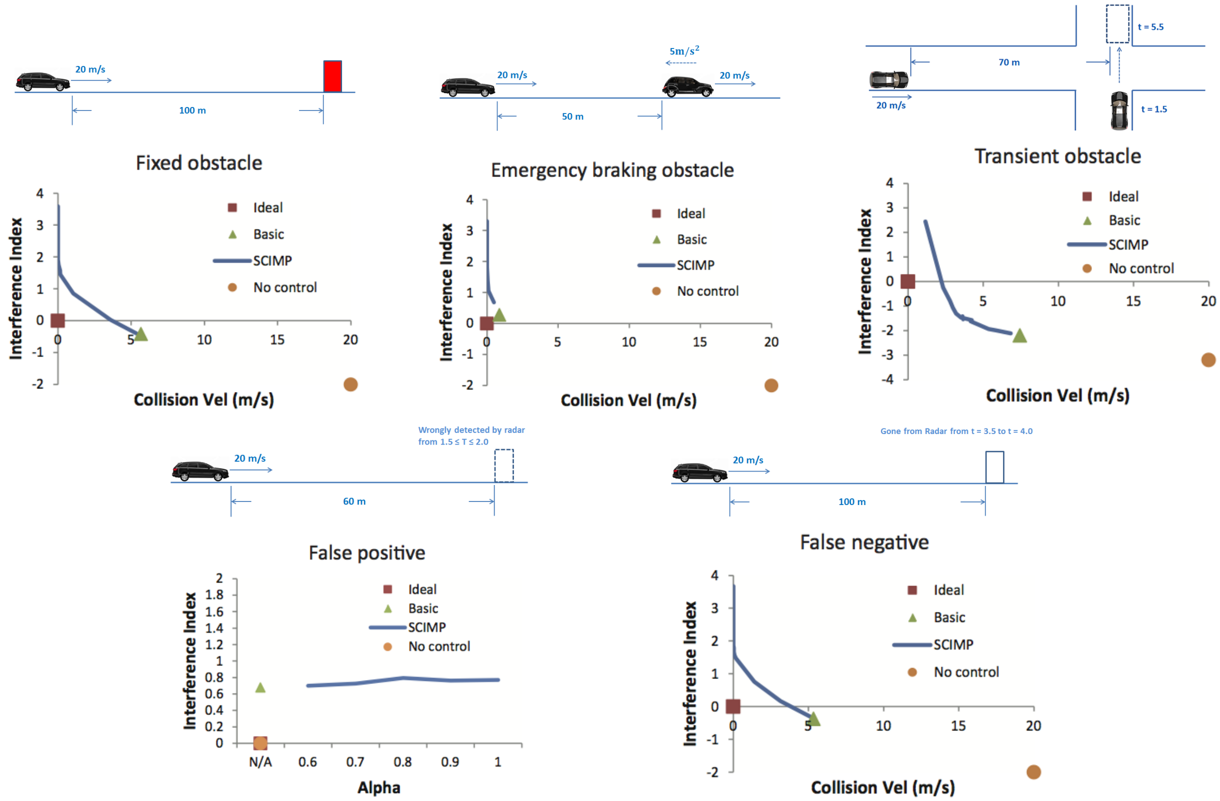

Fig. 1. Five braking scenarios. Results are averaged over 100 trials

1) *Discontinuity Time* (DT): measures the amount of jerk experienced by passengers. We integrate the time over which the acceleration at two time subsequent time steps is greater than a threshold, which we set to $4\,\mathrm{m/s}^2$.
2) *Excess Time* (ET): measures the amount of time consumed by excessive interference in a scenario. A scenario is considered completed if the car reaches a goal, collides with an obstacle, or, in the braking case, comes to a stop. ET is computed by measuring the policy completion time and subtracting the completion time for an optimal collision-free policy with perfect state information. Because some policies can complete a scenario faster than the optimal policy (e.g., by colliding with an obstacle), ET may be negative.
3) *Stopping Distance* (SD): This metric measures the distance to the obstacle after the vehicle stops, or 0.0 if the scenario is completed in any other manner. SD only applies to the braking scenarios 1–3 and 5.

II is computed as follows:

$$II = c_1 DT + c_2 ET + c_3 SD$$

where c_1, c_2, and c_3 are proportionality constants. These are set to $10\,s^{-1}$, $1\,s^{-1}$, and $1/2\,m^{-1}$ respectively based on some amount of tuning. In future work we hope to use human subjects experiments to determine weights that yield an interference index that is more perceptually meaningful to human drivers.

We designed five test scenarios for the braking condition and three for the intersection crossing condition. The scenarios simulate actual environments that an emergency safety system might face in practice, and the behavior of obstacles and simulation constants are *not* known in advance to the vehicle. Rather, it infer them through sensor readings.

The braking scenarios are illustrated in Figure 1. They include fixed obstacles, hard-braking obstacles, transient lane-crossing obstacles, and false positives and negatives. In all cases the vehicle starts at $20\,m/s$ and the driver's control maintains a steady velocity. The intersection crossing scenarios are illustrated in Figure 2. In all cases the driver's control is a constant acceleration of $4\,m/s^2$. In the *between obstacles* scenario, the car is $22\,m$ from crosing a two-lane road intersection and has an initial velocity of $4\,m/s$. Two obstacles are approaching from either direction at $8\,m/s$ with constant velocity. The two *side-impact* scenarios have the following initial conditions: 1) the vehicle collides with the obstacle unless it brakes, and 2) the obstacle collides head-on with the vehicle unless the vehicle accelerates. In condition 1, the car's initial velocity is $4\,m/s$ and the obstacle approaches with velocity $7.5m/s$, while in condition 2 the car has initial velocity $10m/s$ and the obstacle approaches with velocity $7m/s$.

The plots above each scenario in Figures 1 and 2 compare the SCIMP policies with α = 60, 70, 80, 90, 95, and 99, to the ideal omniscient controller (Ideal), which has perfect information about the future behavior of the environment; the

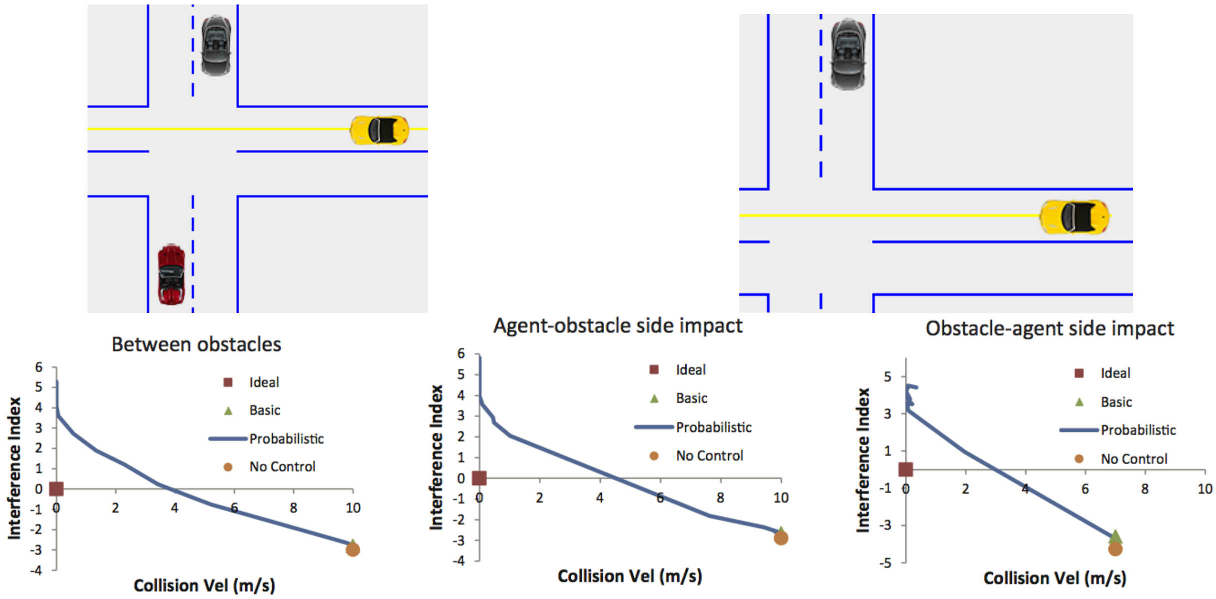

Fig. 2. Three intersection scenarios. Results are averaged over 100 trials.

controller that runs deterministic optimal control on the most-likely environment state (Basic), and the raw driver's control (No control). We performed a Monte-Carlo evaluation by running each policy 100 times under the stochastic sensing and dynamics models.

VI. Conclusion

We present a generic SCIMP framework for conducting semiautonomous collision avoidance and two concrete implementations that can be used under that framework. First, a probabilistic based collision-avoidance braking strategy in terms of behavior in the presence of uncertainty in vehicle dynamics, sensor noise, and unpredictable obstacle behavior. Second, a technique for determining safe trajectories for intersection crossings in the presence of state uncertainty is presented. A number of Monte-Carlo simulations demonstrate that SCIMP achieve low collision risk and driver interference in a variety of scenarios. Our current work is now testing these algorithms with human drivers in commercial driving simulator to observe how they perform and respond to assisted driving.

Acknowledgement

This work is partially supported by the Indiana University Collaborative Research Grant fund of the Office of the Vice President for Research.

References

[1] National Highway and Traffic Safety Administration, "Fatality analysis reporting system general estimates system: 2008 data summary. national highway and traffic safety administration report no.: Dot hs 811 171."

[2] H. Alm, O. Svidén, and Y. Waern, *Cognitive ITS: on cognitive integration of ITS functions around the driver's task*. Hillsdale, NJ, USA: L. Erlbaum Associates Inc., 1997, pp. 231–237.

[3] A. Ben-Yaacov, M. Maltz, and D. Shinar, "Effects of an in-vehicle collision avoidance warning system on short- and long-term driving performance," *Human Factors The Journal of the Human Factors and Ergonomics Society*, vol. 44, no. 2, pp. 335–342, 2002.

[4] M. Lehto, "An experimental comparison of conservative versus optimal collision avoidance warning system thresholds," *Safety Science*, vol. 36, no. 3, pp. 185–209, 2000.

[5] Y. Fujita, K. Akuzawa, and M. Sato, "RADAR BRAKE SYSTEM," in *Intelligent Transportation: Serving the User Through Deployment. Proceedings of the 1995 Annual Meeting of ITS America.*, 1995.

[6] J. Hillenbrand, A. Spieker, and K. Kroschel, "Efficient decision making for a multi-level collision mitigation system," in *Intelligent Vehicles Symposium*, 2006.

[7] S. .Anderson, S. Peters, K. Iagnemma, and T. Pilutti, "A unified approach to semi-autonomous control of passenger vehicles in hazard avoidance scenarios," in *IEEE International Conference on Systems, Man and Cybernetics (SMC)*, 2009.

[8] R. Karlsson, J. Jannson, and F. Gustafsson, "Model-based statistical tracking and decision making for collision avoidance application," in *American Control Conference*, 2004.

[9] M. Althoff, O. Stursberg, and M. Buss, "Model-based probabilistic collision detection in autonomous driving," in *Intelligent Transportation Systems*, 2009.

[10] J. Johnson and K. Hauser, "Optimal acceleration-bounded trajectory planning in dynamic environments along a specified path," in *International Conference on Robotics and Automation (ICRA)*, St. Paul, USA, May 2012.

[11] O. Madani, S. Hanks, and A. Condon, "On the undecidability of probabilistic planning and infinite-horizon partially observable markov decision problems," 1999.

[12] D. Hsu, W. Sun, and L. N. Rong, "What makes some pomdp problems easy to approximate?"

[13] NHTSA, "Identifying behaviors and situations associated with increased crash risk for older drivers," National Highway and Safety Administration, Tech. Rep. DOT HS 811 093, 2009.

[14] F. Gustafsson, F. Gunnarsson, N. Bergman, U. Forssell, J. Jansson, R. Karlsson, and P.-J. Nordlund, "Particle filters for positioning, navigation, and tracking," in *Transactions on Signal Processing*, 2002.

[15] G. Welch and G. Bishop, "An introduction to the kalman filter," University of North Carolina at Chapel Hill, Tech. Rep., 2006.

[16] T. Fraichard, "Dynamic trajectory planning with dynamic constraints: a 'state-time space' approach," 1993.

[17] K. Kant and S. W. Zucker, "Toward efficient trajectory planning: the path-velocity decomposition," *Int. J. Rob. Res.*, vol. 5, pp. 72–89, September 1986.