2017-01-1406

# The Robustly-Safe Automated Driving System for Enhanced Active Safety [1]

**Changliu Liu[1], Jianyu Chen[1], Trong-Duy Nguyen[2] and Masayoshi Tomizuka[1]**

[1] Department of Mechanical Engineering, University of California, Berkeley, CA 94720 USA
[2] Denso International America Inc., 24777 Denso Drive, Southfield, MI 48033 USA

## Abstract

Road safety is one of the major concerns for automated vehicles. In order for these vehicles to interact safely and efficiently with the other road participants, the behavior of the automated vehicles should be carefully designed. Liu and Tomizuka proposed the Robustly-safe Automated Driving system (ROAD) which prevents or minimizes occurrences of collisions of the automated vehicle with other road participants while maintaining efficiency. In this paper, a set of design principles are elaborated as an extension of the previous work, including robust perception and cognition algorithms for environment monitoring and high level decision making and low level control algorithms for safe maneuvering of the automated vehicle. The autonomous driving problem in mixed traffic is posed as a stochastic optimization problem, which is solved by 1) behavior classification and trajectory prediction of other road participants, and 2) a unique parallel planner architecture which addresses the efficiency goal in the long term and the safety goal in the short term separately. Moreover, a python-based high fidelity simulation system is developed and extensive simulations are performed to evaluate the effectiveness of the proposed algorithm, where both high level decision making and low level vehicle regulation are considered. Two typical scenarios are studied, driving on freeway and driving in unstructured environments such as parking lots. In the simulation, multiple moving agents representing surrounding vehicles and pedestrians are added to the environment, some of which are controlled by human subjects in order to test the real time response of the automated vehicle.

[1] This work is supported by Denso International America, Inc.

## Introduction

Automated driving is widely viewed as a promising technology for future mobility [1]. The benefits are extensive, such as to free human drivers, to ease road congestion and to lower fuel consumption. When the automated vehicle drives on public roads, safety is a big concern. While existing technologies can assure high fidelity sensing and robust control, the challenges lie in the interactions between the automated vehicle and other road participants such as manually driven vehicles and pedestrians [2]. For road safety, the driving behavior of the automated vehicle should be carefully designed.

Conservative strategies such as “braking when collision is imminent”, known as the Automatic Emergency Braking (AEB) function in existing models, may not be the best actions in many cases. Considering the dynamics and future courses of surrounding vehicles, the automated vehicle has various choices for a safe maneuver. For example, the vehicle may slow down to keep a safe headway until the headway reaches the safe limit; the vehicle may steer to the left or right to avoid a collision; and the vehicle may even speed up if it can get out a dangerous zone by doing so. For active safety, the vehicle should be able to figure out all safe actions and choose the best action among all choices.

The Robustly-Safe Automated Driving (ROAD) system [3] was proposed by Liu and Tomizuka, which is an integrated framework to design the driving behavior for automated vehicles in order to prevent or minimize occurrences of collisions among vehicles, pedestrians and obstacles while maintaining efficiency (e.g. maintaining the high speed on freeway). The ROAD system consists of three layers as shown in Fig.1. The functions in the three layers can be characterized as “observe”, “think” and “behave”, where the automated vehicle “observes” the environment through multiple sensors

in layer 1, "thinks" about the best action in layer 2 and "behaves" to realize the plan in layer 3. The effect of the ROAD system is illustrated through Fig.2a and Fig.2b, where the automated vehicle predicts the future courses of all surrounding road participants (vehicles or pedestrians) and confines its own trajectory in a safe region in adherence to the predictions for safety.

In view of active safety, challenges in ROAD system exist in the complication of the environment where interactions take place in great abundance, and the difficulty in real time computation for trajectory planning. Methods to deal with interactions are discussed in [4, 5]. The implementation of these methods also depends on real time computation, e.g. whether the automated vehicle can process the information and compute a desirable trajectory in real time with limited computation power. In literature, methods for trajectory planning in complicated environments can be divided into two categories depending on the planning horizons, namely long term (or global) planning and short term (or local) planning. According to Paden *et.al* [6], there are three kinds of long term planning methods for autonomous driving: 1) search-based methods such as hybrid A* [7]; 2) incremental methods such as Rapidly-exploring Random Tree (RRT) [8]; and 3) optimization-based methods such as Model Predictive Control (MPC) [9]. However, it is hard for long term planners to generate timely response to emergencies. To speed up the computation and address safety in real time, the ROAD system adopts a unique parallel planning architecture where one planner plans long term trajectories with low sampling frequency, and the other planner plans short term executable trajectories with high sampling frequency using the long term trajectories as references.

The contributions of this paper lie in: 1) elaboration of the design principles in the ROAD system as an extension of the previous work [3] which focuses on layer 2; 2) introduction of a python-based high fidelity simulation system; 3) evaluation of the ROAD system in multiple scenarios through extensive human-in-the-loop simulations. The test vehicle, a Lincoln MKZ, is currently under preparation as shown in Fig.3. The ROAD system will be validated on the test vehicle once it passes a significant number of human-in-the-loop simulations.

The remainder of the paper is organized as follows: in Section II, the ROAD system will be discussed in detail. The simulation system will be introduced in Section III. Multiple case studies will be presented in Section IV. Discussions are provided in Section V. Section VI concludes the paper.

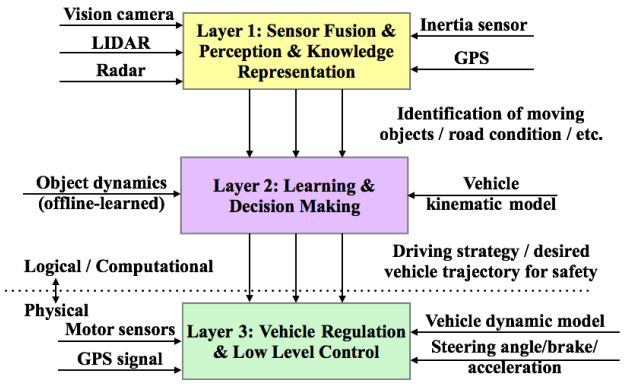
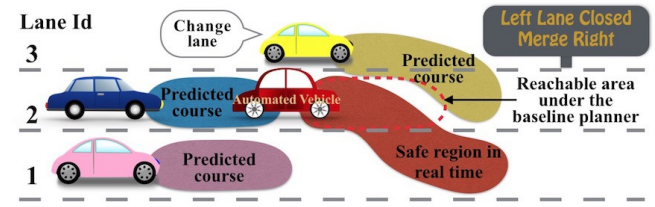
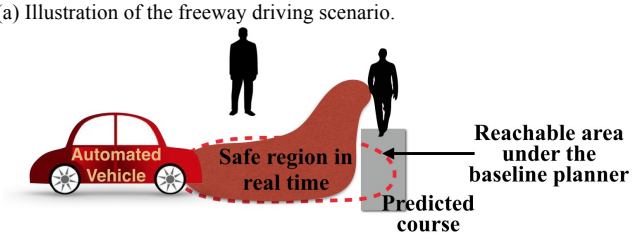



Figure 1. Architecture for the robustly-safe automated driving system.




(a) Illustration of the freeway driving scenario.




(b) Illustration of the scenario in an unstructured environment.
Figure 2. The function of the ROAD system.

## The ROAD System

In the ROAD system, the driving behavior for the automated vehicle should be designed considering the following two factors: efficiency and safety. Equations (1-5) provide the mathematical formulation. Denote the state (position $p$ and velocity $s$ of the center of the rear axis in the world frame, and vehicle heading $\theta$ and angular velocity), control input (throttle and wheel angle) and the measurements of the automated vehicle as $x_0$, $u_0$ and $y_0$ respectively. The efficiency factor requires that the objective $G_0$ of the automated vehicle (such as lane following in a constant speed or going to a desired position) be achieved in an optimal manner through minimizing a cost function $J(x_0, u_0, G_0)$. As the state $x_0$ is not directly known, the cost should be minimized in the sense of expectation given the measurement $y_0$, e.g. $E[J(x_0, u_0, G_0)|y_0]$ as shown in (1). Meanwhile, the safety factor requires that the efficiency requirement be fulfilled safely as the motion of the automated vehicle should be constrained with respect to other road participants' behaviors. Denote the state of other road participants as $x_1, \cdots, x_n$ and

the state of the environment as $x_e$. Then the constraint on the automated vehicle can be written as $x_0 \in R_S(x_1, \cdots, x_n, x_e)$ where $R_S$ is a safe set in the state space of the automated vehicle that depends on the states of other road participants and the state of the environment. As all of the states are not directly known, the constraint should be considered in the stochastic sense such that it is satisfied almost surely with probability $1-\varepsilon$ for $\varepsilon \to 0$ given the measurement $y_0$, e.g. $P(\{x_0 \in R_S(x_1, \cdots, x_n, x_e)\}|y_0) \geq 1-\varepsilon$ as shown in (5). Then the following optimal control problem can be formulated,

$$\min_{u_0} \quad E[J(x_0, u_0, G_0)|y_0] \tag{1}$$
$$\text{s.t.} \quad u_0 \in \Omega, E(x_0|y_0) \in \Gamma \tag{2}$$
$$\dot{x}_0 = f(x_0, u_0, w_0) \tag{3}$$
$$y_0 = h(x_0, x_1, \cdots, x_n, x_e, v_0) \tag{4}$$
$$P(\{x_0 \in R_S(x_1, \cdots, x_n, x_e)\}|y_0) \geq 1-\varepsilon \tag{5}$$

where $\Omega$ is the control space constraint for vehicle stability, and $\Gamma$ is the constraint regarding the speed limit and other regulations. $E(x_0|y_0)$ refers to the estimated state of the automated vehicle given the measurement $y_0$. The function $f$ in (3) and the function $h$ in (4) are the dynamic equation and the measurement equation respectively, where $w_0$ and $v_0$ are noise terms.

The ROAD system solves the above problem by 1) estimating the current states $x_i$'s using the measurement $y_0$ in layer 1 and 2) predicting the future states of all road participants and solving for the optimal control $u_0$ using the predicted states in layer 2 and 3) tracking the computed optimal control trajectory in layer 3.

### *Layer 1*

Layer 1 handles sensor fusion, perception and knowledge representation, which takes input from multiple sensors. The vehicle takes two kinds of measurements: measurements of the state of the ego vehicle $x_0$ and measurements of the environment and other road participants $x_1, \cdots, x_n, x_e$. For the first kind of measurements, the vehicle filters out the noisy data from GPS and IMU to obtain the mean and standard deviation of $x_0$. For the second kind of measurements, the vehicle relies on Lidar, vision camera and radar. For example, in the camera frame, the position $y_0^{cam,i}$ of the road participant $i$ is measured as

$$y_0^{cam,i} = R^{cam}(\theta_0)\left(p_i - p_0 - T^{cam}(\theta_0)\right) + v_0^{cam} \tag{6}$$

where $p_i$ is the position of the road participant $i$, $p_0$ is the position of the automated vehicle, $\theta_0$ is the heading of the automated vehicle, $v_0^{cam}$ is the measurement noise, $T^{cam}(\theta_0)$ is the translation vector from the rear axis to the camera in the world frame, and $R^{cam}(\theta_0)$ is the rotation matrix from the world frame to the camera frame. Equation (6) is a special case of (4). The position of the road participant $i$ can be estimated with respect to (6) by extended Kalman Filter [10]. The velocity of the road participant $i$ can then be obtained by taking finite difference of the estimated positions. If one variable is measured by multiple sensors, the final estimate of the variable will be taken as a weighted average of all estimates where the weights depend on the standard deviations of the estimates.

It is worth noting that the measurement function $h$ in (4) is determined by hardware and mounting (as indicated in (6)). The objectives in choosing the arrangement of the sensors are to 1) minimize the influence of the noise $v_0$ in the measurement $y_0$, e.g. having redundant sensors so that information can be fused for high resolution, and 2) maximize the number of observable road participants, e.g. having long range sensors. The test vehicle and the sensor configuration is shown in Fig.3. Once the measurement is obtained, the estimated states $E(x_0|y_0), \cdots, E(x_n|y_0)$ will be computed as discussed above. These estimated states will be used in solving the optimal control problem in layer 2. In the following analysis, for simplicity, the state variable $x_i$ will refer to its estimate $E(x_i|y_0)$ if there is no confusion.

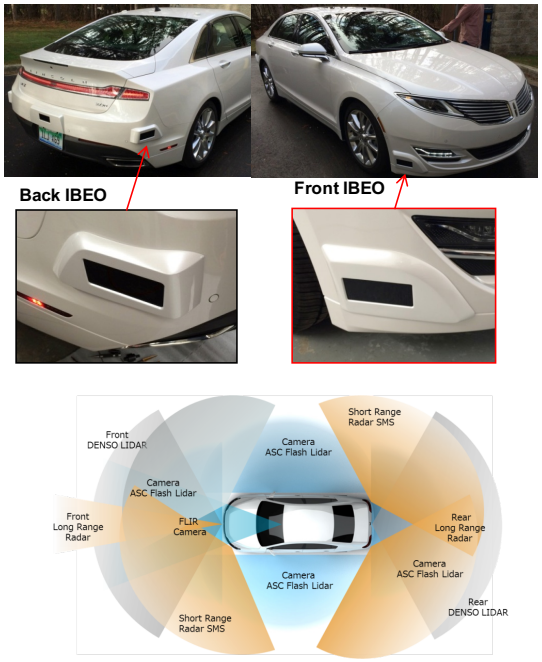



Figure 3. The test vehicle and sensor configuration.

### *Layer 2*

In layer 2, the control sequence $u_0$ is obtained by solving the optimization problem (1-5). However, the problem is in general hard to solve due to the safety constraint (5), as the dynamics of the states $x_i$'s are unknown and the set $R_S$ is non convex. To solve the problem efficiently, the behaviors of other road participants will be identified and $x_i$'s will be predicted online. A parallel planning structure will be used to solve the nonconvex optimization efficiently.

**Identification and Prediction of the Behavior of Other Road Participants**

The learning structure in Fig.4 is designed for the automated vehicle to make predictions of the surrounding road participants, in which the process is divided into two steps: 1) the discrete behavior classification, where the observed trajectory of a vehicle or a pedestrian goes through an offline trained classifier; and 2) the continuous trajectory prediction, where the future trajectory is predicted based on the identified behavior using an empirical model which contains adjustable parameters to accommodate the road participant's time-varying behavior. The classification step is needed when communications among vehicles and pedestrians are limited. Otherwise, vehicles or pedestrians can broadcast their planned behaviors.

The classifier and the empirical models are trained from data (e.g. trajectories of vehicles or pedestrians) offline using supervised learning, where the data are obtained from the human-in-the-loop simulations which will be discussed in the section "The Simulation Environment". In the future, we will use the publicly available traffic data set, such as the NGSIM data set [11], to train the classifier and to obtain better models. The online learning module uses the parameter adaptation methods. Refer to [3] for details of the offline learning and online prediction algorithms.

**Online Decision Making and Control**

Based on the predictions of the surrounding road participants, the automated vehicle needs to find a safe and efficient trajectory by solving the optimal control problem (1-5). In order for the automated vehicle to plan efficiently in the long term as well as to generate timely responses to emergencies, the decision making architecture in Fig.5 is adopted, which is designed to be a parallel combination of a baseline planner that solves the optimal control problem over a long time horizon with respect to the most likely predictions of other road participants, and a safety planner that takes care of the safety constraint (5) robustly in real time.

The baseline planner solves the optimal control problem in the long term to ensure efficiency. Methods discussed in the introduction section such as A* search or MPC can be used to solve the nonlinear and nonconvex planning problem. In this paper, to better illustrate the capacity of the safety planner, the baseline planner only solves the optimal control problem (1-4) without the safety constraint (5), which is similar to the planner in use when the automated vehicle is navigating in an open environment without interactions with other road participants. The computation in the baseline planner can be done offline. The resulting control policy will be stored for online application.

The safety planner modifies the trajectory planned by the baseline planner locally to ensure that it will lie in the safe set $R_S(x_1, \cdots, x_n, x_e)$ in real time. Using the invariant set method [12], the nonconvex state space constraint is transformed to a convex control space constraint $u_0 \in U_S$. If the baseline control input $u_0(t)$ is anticipated to violate the safety constraint, the safety controller will modify the input by mapping it to the set of safe control $U_S(t)$. Note that the set of safe control $U_S$ depends on 1) the distance between the automated vehicle and each surrounding road participant; 2) the relative motion between the automated vehicle and each surrounding road participant (which is predicted using the method shown in Fig.4). The detailed derivation was explained in [3], [12] and [13]. In this paper, the idea will be illustrated using several examples in the case studies.

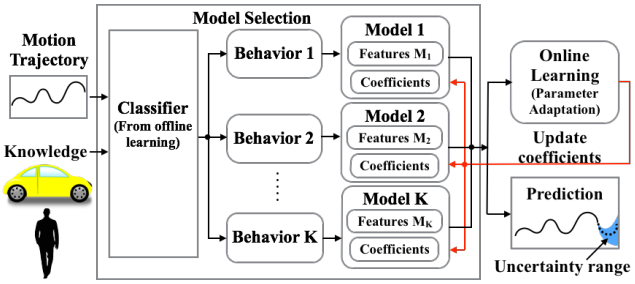



Figure 4. The structure of the learning and prediction module.

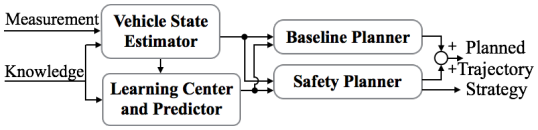



Figure 5. The structure of the decision making module.

### *Layer 3*

In layer 3, the dynamic model of the vehicle is considered (whereas only kinematic model is considered in layer 2). The low level controller includes both feedback and feedforward control. The design considerations are given to both tracking accuracy and passenger comfort. As vehicle dynamics strongly depend on road surface conditions, vehicle speed, and passenger load among others, adaptive control and gain scheduling are highly desired [14].

## The Simulation Environment

To ensure safety, a simulation environment is needed in order to evaluate the algorithms before the road test. Existing vehicle and traffic simulators often suffer from imbalanced coverage of macro traffic dynamics and micro vehicle dynamics, e.g. some either contain only high level decision making and omit the realistic vehicle dynamics, others involve too many details of the low-level vehicle control and dynamics and omit the high-level decision making and traffic. In order to simulate the real-world driving scenarios, a multi-vehicle simulator is needed to consider both interactions between vehicles, and the dynamic details of every single vehicle. For this purpose, we developed an object-oriented simulator. A physic engine [15] is embedded in the simulator

to simulate the real-world physical phenomenon (e.g. collision, friction, and gravity) as well as the vehicle dynamics to make it realistic. The simulator consists of four modules: environment, sensor, agent and vehicle, as an analogy of real-world driving, where each human driver is considered as an agent, who uses his sensors (e.g. eyes) to get information (e.g. distance from the front vehicle) from the environment, provides control inputs (e.g. turning the steering wheel) to move the vehicle so as to influence the environment.

In the simulator shown in Fig.6, multiple vehicles are running in the environment and are interacting with the environment (e.g. friction and contact with the road, collision with surrounding vehicles). The environment contains road map and all road participants. Each road participant has its own dynamics and is associated with an agent which controls the motion of the road participant. The agent may be software-controlled (e.g. agents that run the ROAD system autonomously) or human-controlled (e.g. agents that read human commands from input devices such as a wheel or a keyboard). Each software agent is associated with a sensor through which it can acquire information from the environment. For now, the sensors can access a noisy measurement of the positions and velocities of the ego vehicle and the surrounding vehicles. In the future, we will add more realistic sensor models to simulate Lidar, radar and camera. Using the sensor data, the software agent for a vehicle then makes the driving decisions and sends out the steering, throttle and brake signals to the vehicle. Other than ROAD system, we developed various classes of software agents representing different driving characteristics in order to mimic the real-world scenarios. On the other hand, a human agent observes the current configuration of the virtual environment through a visual display and controls the assigned road participant using input devices.

The trajectories of vehicles or pedestrians controlled by human agents are recorded in every simulation, which serve as the data source during offline learning in layer 2. During offline learning, the trajectories are labelled manually regarding the intended driving behavior at each time step. Then the classifier for behavior prediction and the empirical models for trajectory prediction under different driving behaviors are learned from the labelled data. The human subjects are volunteers, who also participate in the evaluation process of the ROAD system during simulation. For different driving scenarios, e.g. freeway driving or urban driving, different classifiers need to be trained as the types of road participants and their behaviors in those scenarios differ. Note that discrepancies exist between the trajectories of road participants controlled by human agents in the simulation and the trajectories of manually driven vehicles (or pedestrians) in the real world, since the human subjects may behave differently in virtual reality and in reality. In the future, real world data will be used for offline training as mentioned earlier.

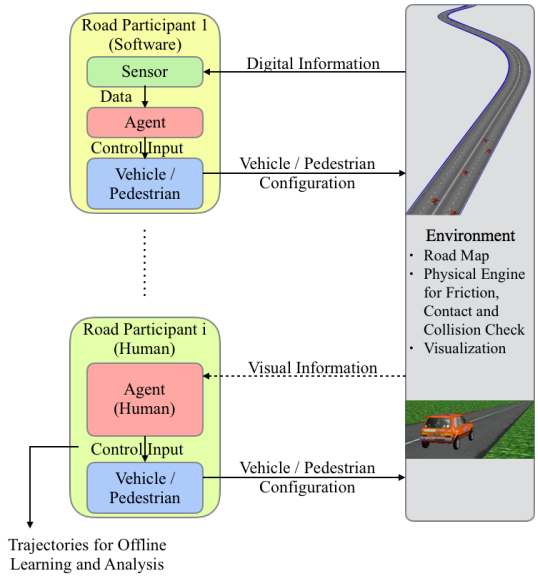



Figure 6. The simulation platform.

## Case Studies

Case studies are performed to illustrate the performance of the ROAD system in both structured driving (freeway driving with mixed traffic) and unstructured driving (driving in parking lots). The simulations are done using parameters from the Lincoln MKZ shown in Fig.3. The ROAD system will be tested on the real vehicle in the future.

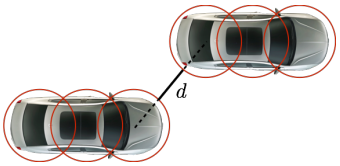
### *Freeway Driving with Mixed Traffic*

In freeway driving, there are typically two kinds of objectives $G_0$: lane following with desired speed (in any lane) and lane changing (to a target lane). The baseline planners for the two objectives are obtained offline by computing optimal control policies for problem (1-4). The safety planner checks online whether the planned trajectory is safe to execute with respect to the predicted motions of the surrounding vehicles. Three behaviors are considered for surrounding vehicles: lane following, lane change to the left, and lane change to the right. A Hidden-Markov-Model-based classifier is trained offline using labelled trajectories of human-controlled road participants in the simulator. The intended behavior of each surrounding vehicle is predicted online using the classifier. And the future motion of a surrounding vehicle is predicted using an empirical model associated with the classified behavior.

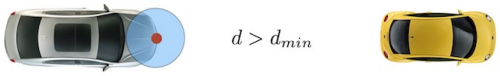
#### Safety Constraint in Different Scenarios

The constraints on the vehicle input $u_0$ in different scenarios are illustrated in Fig.7 to Fig.10. For simplicity, a constraint on planar acceleration is used to illustrate the safety constraint $U_S$ as the constraint on planar acceleration can be transformed to the constraint on throttle and wheel angles.

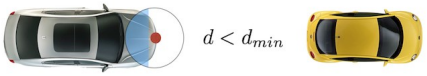
The procedures in computing $U_S$ is discussed in [12]. Refer to [13] for the robust version of this method which considers uncertainties in the predicted future motions of other road participants. For simplicity, the general method without the compensation for uncertainties is briefly reviewed here. Let all vehicles be covered by three circles as shown in Fig.7a. The diameter of the circles equals to $\sqrt{2}$ times the width of the vehicle. One circle is at the middle of the vehicle, while the other two circles cover the front and the rear of the vehicle. The relative distance $d$ between the ego vehicle and a surrounding vehicle is approximated by the smallest relative distance among the respective circles as shown in Fig.7a. Let $d_{min}$ be the minimum distance requirement. As discussed in [3], the safety constraint $U_S$ regarding a front vehicle is obtained by evaluating a safety index $\phi := d_{min}^2 - d^2 - k\dot{d}$ where $k > 0$ is a tunable parameter. The system is considered safe if the safety index is negative and $d > 0$, and unsafe otherwise. Hence $U_S$ is the whole space if $\phi < 0$, and $U_S = \{u_0 : \dot{\phi} < 0\}$ if $\phi \geq 0$. Note that $\dot{\phi} < 0$ implies that $\ddot{d} > -2d\dot{d}$ and the relative acceleration $\ddot{d}$ depends on $u_0$. Figure 7b to 7d shows the safety constraint with respect to a relatively static front vehicle, e.g. $\dot{d} = 0$ and $\phi = d_{min}^2 - d^2$. The red dot represents zero acceleration and the circle represents the boundary of maximum acceleration in any direction (which may also be in other shapes). The shaded area represents $U_S$. When the headway is far enough, all directions of acceleration are safe, e.g. all of them satisfy the safety constraint $U_S$ as $U_S$ is the whole space. When the headway is too short, only decelerations are safe. When there is a wall, turning against the wall is not safe (a similar safety index $\phi_{wall}$ can be defined with respect to the wall and the corresponding $U_S$ can be derived). Figure 8 shows the safety constraint when the front vehicle has relative motion with respect to the automated vehicle, e.g. $\dot{d} \neq 0$ and $\phi = d_{min}^2 - d^2 - k\dot{d}$. In Fig.8a, although $d > d_{min}$, $U_S$ is not the whole space since $\phi > 0$ due to $\dot{d} < 0$. Figure 9 and 10 illustrate the safety constraint when there is a vehicle in the adjacent lane. In mixed traffic, the constraint $U_S$ is the intersection of the constraints computed with respect to all surrounding vehicles. Note that above is the continuous time implementation of the method. For discrete time implementation, $U_S$ at time step $k$ is chosen to be the set $\{u_0 : \phi(k+1) < 0\}$ [12].



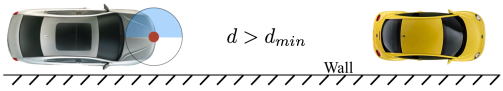

(a) Computation of the relative distance between two vehicles.




(b) $U_S$ when the distance is large enough ($\phi < 0$)

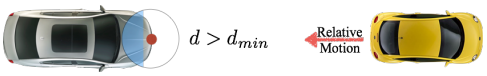



(c) $U_S$ when the distance is small ($\phi > 0$)

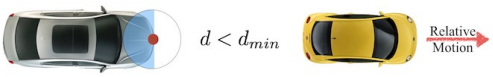



(d) $U_S$ with boundary constraint ($\phi < 0,\ \phi_{wall} = 0$)

Figure 7. The safety constraint $U_S$ with respect to a relatively static front vehicle.

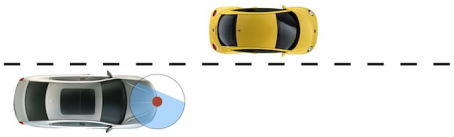



(a) $U_S$ with decreasing relative distance ($\phi > 0$)

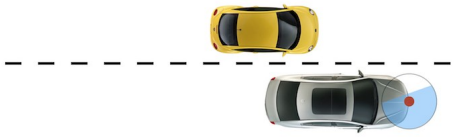



(b) $U_S$ with increasing relative distance ($\phi > 0$)

Figure 8: The safety constraint $U_S$ with respect to a relatively moving front vehicle.

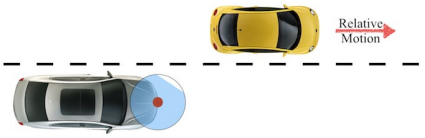


(a) $U_S$ for vehicle in the front ($\phi > 0$)

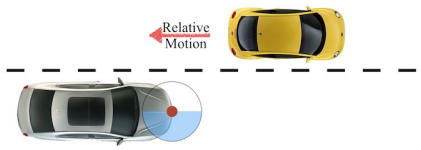


(b) $U_S$ for vehicle behind ($\phi > 0$)

Figure 9: The safety constraint $U_S$ with respect to a relatively static vehicle in the adjacent lane.



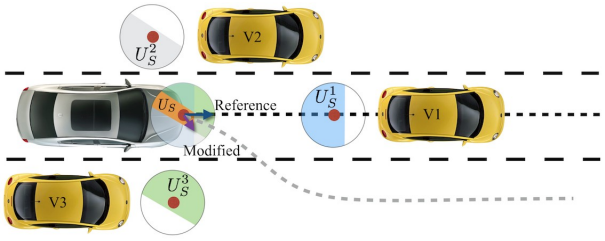

(a) $U_S$ for a vehicle that is moving away ($\phi > 0$)



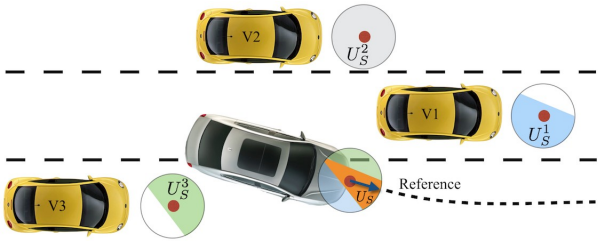

(b) $U_S$ for a vehicle that is moving closer ($\phi > 0$)
Figure 10: The safety constraint $U_S$ with respect to a relatively moving vehicle in the adjacent lane.

## Active Safety for Lane Following

Figure 11 shows the active safety measures of the ROAD system during lane following. There is heavy traffic on the left lanes. As the current speed of the automated vehicle is below the desired speed, the baseline planner generates an acceleration command. The safety constraints with respect to vehicles 1 to 3 are computed and shown as $U_S^1$, $U_S^2$ and $U_S^3$ in the figure and $U_S \coloneqq \cap\, U_S^i$. As the acceleration command generated by the baseline planner is not safe (not in $U_S$), it is modified by the safety planner. The modification signal tries to minimize the length difference between the reference acceleration vector and the modified vector as well as the angular difference between the two vectors. Then the modified signal has non trivial lateral acceleration. When the lateral velocity reaches a threshold, a lane change command will be generated if the desired lateral acceleration is not trivial. Under this new command, the vehicle changed to its right lane. The metric for the modification is a design parameter which allows the vehicle to generate diverse behaviors [12], e.g. the vehicle may have different judgements on the best way to be safe. For the current case, the priority is that the vehicle keeps the magnitude of acceleration, which then lead to the decision of lane change. On the other hand, if the metric cares more about the angular difference between the reference acceleration vector and the modified acceleration vector. The modification signal will remain in the same direction and there will be no turning command as shown in Fig.12. In this case, the automated vehicle will remain in the current lane and adapt to the motion of the front vehicle. This behavior is very similar to the one generated by adaptive cruise control (ACC).




(a)




(b)
Figure 11: Active safety for lane following.

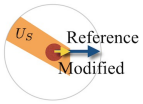
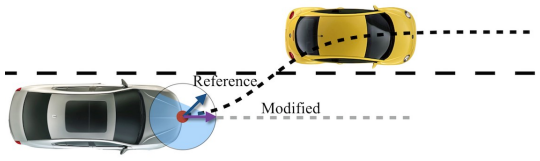


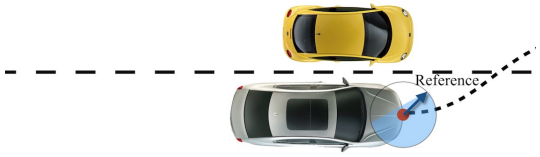

Figure 12: Different safety behavior during lane following due to different modification metric.

## Active Safety for Lane Changing

Figure 13 illustrates the active safety measures of the ROAD system during lane change on freeway. When the vehicle started to change lane, it is not safe to do so. The safety planner cancelled the turning command and increased the longitudinal acceleration. Then the automated vehicle overtook the vehicle in the target lane and finished the lane change when it was safe to do so. The behavior of the automated vehicle depends significantly on the predicted behavior of the vehicle in the target lane. If the vehicle is predicted to be moving relatively forward, it is possible that a decelerating modified signal be generated as shown in Fig.14, in which case the automated vehicle will change lane to follow the vehicle in the target lane when a safe "gap" is created by deceleration.




(a)




(b)
Figure 13: Active safety for lane change

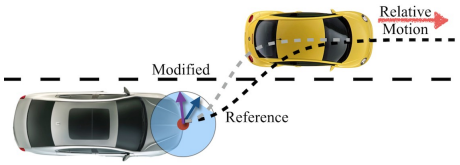



Figure 14: Different safety behavior during lane change due to different prediction.

**Active Safety for Navigation in Heavy Traffic**

Figure 15 illustrates the active safety measures of the ROAD system when traffic is heavy on a curved freeway. $d_{min}$ is chosen to be 10m in order to make the vehicle motion less conservative. The objective $G_0$ is to following any lane with a desired speed 37m/s, which is higher than the current traffic speed. To fulfill $G_0$, the automated vehicle performed several lane changes safely with the assistance of the safety planner. The scenarios discussed in Fig.11 and Fig.13 are all observed during simulation. The distance and velocity profile during the simulation is shown in Fig.16. The dark bar indicates the moment when the safety planner was active, which matches with the moment when the smallest distance to the surrounding vehicles reached a threshold (shown by the dotted line). Note that the threshold is greater than $d_{min}$, since the safety index depends on the relative motion in addition to the relative distance, and the relative distance was decreasing at the moment when the threshold was reached. The smallest distance is only computed for 1) the front vehicle in the current lane when the automated vehicle is following the lane, and 2) the front vehicle in the current lane and the surrounding vehicles in the target lane when the automated vehicle is changing lane. For example, if there is only one surrounding vehicle which is in the adjacent lane and both vehicles are going to follow the lanes, then the smallest distance is infinity. When the safety planner was on, changes on the vehicle velocity and direction were generated by the safety planner as discussed earlier. Note that the safety planner may be inactive when the minimum distance goes below $d_{min}$. That is due to the fact that the safety planner monitors the relative distance, the relative velocity as well as the relative acceleration. If the relative distance will significantly increase without any modification of the control input, the safety planner will remain inactive. In the simulation, the smallest distance was always kept over 4m which is smaller than $d_{min}$ due to 1) imperfections of the predictions with respect to the future motions of the surrounding vehicles and 2) limitations on control efforts regarding vehicle dynamics in layer 3. Hence a margin needs to be added to the safety index in order for the relative distance to be kept above $d_{min}$.

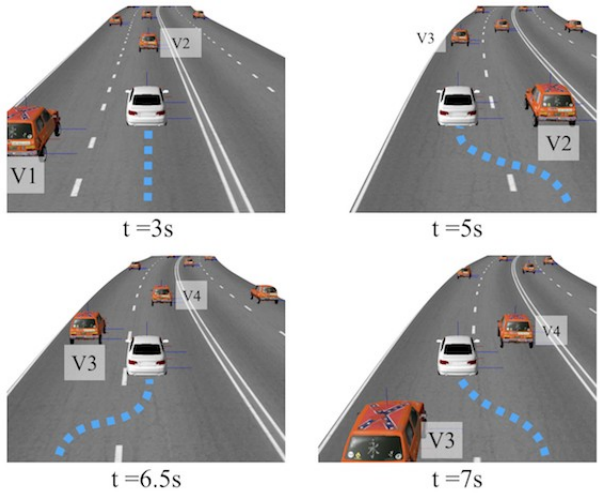



Figure 15: Active safety for heavy traffic.

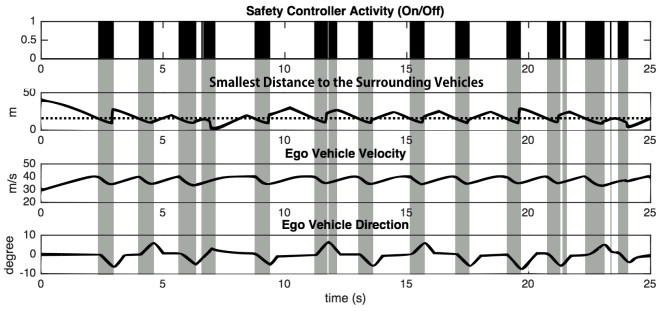



Figure 16: The distance and velocity profile for mixed traffic simulation.

***Driving in Unstructured Environments***

Driving in parking lots is a typical unstructured driving scenario. The automated vehicle needs to interact with pedestrians safely. In this case, $d_{min}$ is set to be 4m and a margin is added in the safety index. The objective $G_0$ for the automated vehicle is to navigate to a desired parking space. Similar to the freeway driving case, the baseline planner is obtained offline. The safety planner checks online whether the planned trajectory is safe to execute with respect to the predictions of the pedestrian motions. However, unlike the freeway driving case, not all directions of acceleration are feasible due to the nonholonomic nature of automobiles. For example, when the vehicle speed is high, it is possible for the vehicle to have acceleration in many directions. When the vehicle speed is low and gear shift is not allowed, the vehicle can only generate a small range of acceleration by steering and pressing the pedal as shown in Fig.17. Roughly speaking, the lateral acceleration is proportional to the vehicle speed and the turning rate. Hence, when the vehicle speed is low, not much lateral acceleration can be generated. In order for the input $u_0$ to be feasible, when modifying the trajectories in the safety planner, the nonholonomic constraint also needs to be considered as shown in Fig.18, which will be explained later.

(a) Vehicle with low speed

(b) Vehicle with high speed
Figure 17: The non-holonomic constraint.

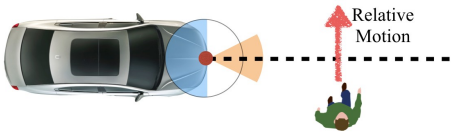



Figure 18: Safety constraint with respect to pedestrians.

**Active Safety for Interacting with Pedestrians**

Figure 19 illustrates the active safety measures of the ROAD system in a parking lot. The pedestrian in the simulation environment is modeled as a mass point with two degrees of freedom, which is controlled by a human subject in real time in order to test the response of the automated vehicle. No prior knowledge (e.g. models from offline learning) is used in predicting the pedestrian's behavior. Moreover, the system does not distinguish between different types of pedestrian behaviors in this study. It assumes a reactive behavior model for the pedestrian, i.e. the pedestrian's future trajectory depends on his or her previous movements as well as the vehicle's previous movement, while the significance of this dependency is captured by several coefficients. These coefficients are identified online by a parameter adaptation algorithm using the observed trajectory of the pedestrian [13]. Then the pedestrian's future movement is predicted by the reactive behavior model. This process uses only the online learning method as shown in the right part in Fig.4 without the model selection.

The desired parking space is marked by the white lines. In the simulation, the automated vehicle tried to go to the parking space while the pedestrian moved crossing the planned path of the automated vehicle. So the vehicle slowed down to wait for the pedestrian and accelerated only after the pedestrian passed by. The safety constraint for the automated vehicle in this scenario is shown in Fig.18. As the only action that satisfied both the safety constraint and the non-holonomic constraint was to stay still, the vehicle chose to wait for the pedestrian to pass by. The distance and velocity profile is shown in Fig.20. The safety planner was on when the distance between the automated vehicle and the pedestrian was small. The small velocity generated at the beginning was due to the imperfect predictions of the pedestrian's motion. The smallest distance was always kept over 4m.

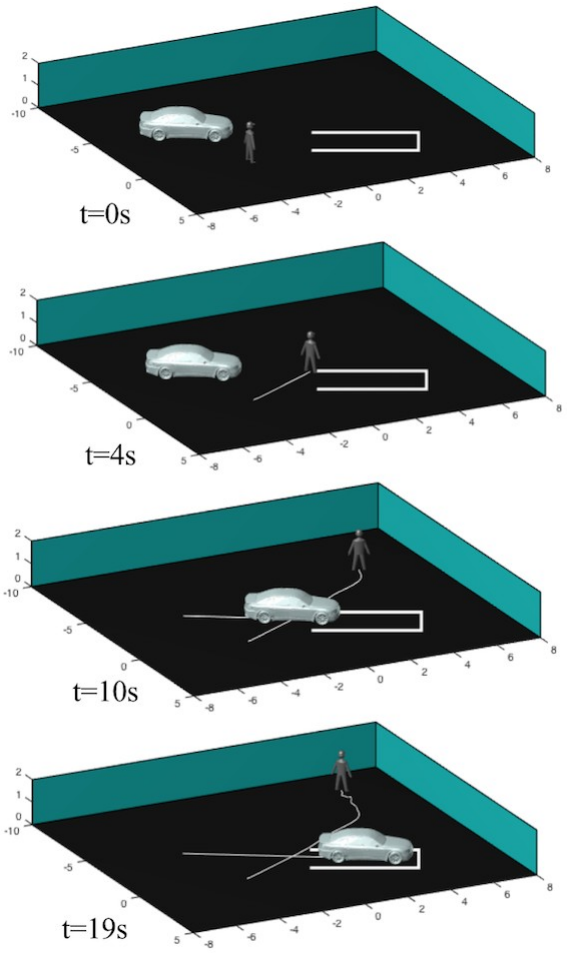



Figure 19: Active safety for unstructured driving.

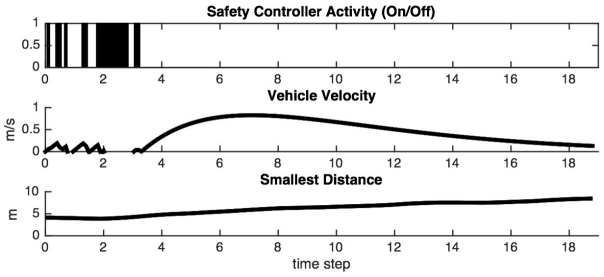



Figure 20: The distance and velocity profile for parking lot simulation.

## Discussions and Future Work

The function of the ROAD system can be divided into two parts: reasoning of other road participants' behaviors and planning the trajectory for the ego vehicle. The first part relies on offline data collection and learning. The purpose of offline learning is to let the ego vehicle make reasonable predictions of the environment so that it can behave less conservatively. Even if the environment is new and the behaviors of the road participants are never encountered before, the system can still generate safe trajectories using only the online learning

method and the parallel planners as discussed in [13]. In the beginning of the interaction, the safety planner can behave defensively by assuming the worst case scenario. During the interaction, the vehicle can fit a reactive behavior model for each road participant using the observed trajectory of that road participant, and refine the model by online adaptation, similar to the method discussed in the case study "Driving in Unstructured Environments". The safety planner then monitors the trajectory generated by the baseline planner given the predictions made by the reactive behavior models, while a larger minimum distance requirement will be chosen if the confidence level of the model is lower. The confidence level of the model can be tracked by comparing the predicted and the observed behaviors of the corresponding road participant.

Although the proposed method is mainly to address active safety for automated vehicles, it can also be applied to driving assistive systems for manually driven vehicles, if we replace the baseline planner by a function that gets human's driving command directly as shown in Fig.21. Then the safety planner can monitor human's driving commands based on the predicted motions of other road participants.

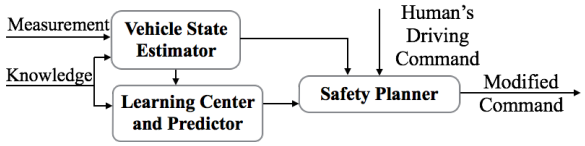



Figure 21. A Driving Assistive System Using ROAD.

In the future, the following parts will be improved. An online long-term trajectory planner will be developed in place of the current offline baseline planner. The long-term planner will consider all related issues in driving, including safety, efficiency, comfort and economy, with a relatively long preview horizon (e.g. 10s) and a relatively low sampling frequency (e.g. 4Hz). The safety controller described in this paper will be running in high frequency (e.g. 20Hz) and be prepared to interrupt the long-term planner in emergencies to guarantee safety. Moreover, to account for diverse behaviors of other road participants, the trajectory predictor will be extended to be stochastic and multimodal.

## Conclusions

In this paper, the ROAD system was discussed to address active safety. The design consideration for each layer of the ROAD system was elaborated. Two case studies were provided: the freeway driving scenario and the unstructured driving scenario. The simulation results verified the effectiveness of the method. The robustness and extendibility of the ROAD system was discussed. Future work was proposed.

## References


[1] L. D. Burns, "Sustainable mobility: a vision of our transport future," *Nature,* vol. 497, pp. 181 - 182, 2013.

[2] C. Urmson, J. Anhalt, D. Bagnell, C. Baker, R. Bittner, M. Clark, J. Dolan, D. Duggins, T. Galatali, C. Geyer and e. al, "Autonomous driving in urban environments: Boss and the urban challenge," *Jounal of Field Robotics,* vol. 25, no. 8, pp. 425 - 466, 2008.

[3] C. Liu and M. Tomizuka, "Enabling Safe Freeway Driving for Automated Vehicles," in *American Control Conference*, 2016.

[4] D. Sadigh, S. Sastry, S. A. Seshia and A. D. Dragan, "Planning for Autonomous Cars that Leverage Effects on Human Actions," in *Proceedings of the Robotics: Science and Systems Conference (RSS)*, 2016.

[5] W. Zhan, C. Liu, C.-Y. Chan and M. Tomizuka, "A non-conservatively defensive strategy for urban autonomous driving," in *Intelligent Transportation Systems Conference (ITSC)*, 2016.

[6] B. Paden, M. Cap, S. Z. Yong, D. Yershov and E. Frazzoli, "A Survey of Motion Planning and Control Techniques for Self-driving Urban Vehicles," in *arXiv preprint arXiv:1604.07446*, 2016.

[7] D. Dolgov, S. Thrun, M. Montemerlo and J. Diebel, "Practical search techniques in path planning for autonomous driving," *Ann Arbor,* vol. 1001, p. 48105, 2008.

[8] E. Frazzoli, M. A. Dahleh and E. Feron, "Real-time motion planning for agile autonomous vehicles," *Journal of Guidance, Control, and Dynamics,* vol. 25, no. 1, pp. 116 - 129, 2002.

[9] J. Levinson, J. Askeland, J. Becker, J. Dolson, D. Held, S. Kammel, J. Z. Kolter, D. Langer, O. Pink, V. Pratt and e. al, "Towards fully autonomous driving: Systems and algorithms," in *IEEE Intelligent Vehicles Symposium (IV)*, 2011.

[10] G. Rigatos and S. Tzafestas, "Extended Kalman filtering for fuzzy modelling and multi-sensor fusion," *Mathematical and computer modelling of dynamical systems,* vol. 13, no. 3, pp. 251 -266, 2007.

[11] "Next generation simulation (NGSIM) high-level data plan," [Online]. Available: http://ops.fhwa.dot.gov/trafficanalysistools/ngsim.htm..

[12] C. Liu and M. Tomizuka, "Control in a safe set: Addressing safety in human robot interactions," in *ASME Dynamics and Control Conference*, 2014.

[13] C. Liu and M. Tomizuka, "Safe Exploration: Addressing Various Uncertainty Levels in Human Robot Interactions," in *American Control Conference*, 2015.

[14] S. E. Shladover, C. A. Desoer, J. K. Hedrick, M. Tomizuka, J. Walrand, W.-B. Zhang, D. H. McMahon, H. Peng, S. Sheikholeslam and N. McKeown, "Automated vehicle control developments in the PATH program," *IEEE Transactions on vehicular technology,* vol. 40, no. 1, pp. 114 - 130, 1991.

[15] "Bullit physics library," [Online]. Available: http://www.bulletphysics.org/Bullet/phpBB3/.