Human Intention-Based Collision Avoidance for Autonomous Cars

Denis Osipychev¹, Duy Tran², Weihua Sheng³ and Girish Chowdhary⁴

Abstract—This paper considers the problem of controlling an autonomous vehicle that must share the road with human-driven cars. We present proactive collision avoidance algorithms that take into account the expressed intent of human driven cars and minimize the need for sudden braking or other purely reactive sudden actions. The presented algorithm utilizes multi-stage Gaussian Processes (GPs) in order to learn the transition model for each vehicle given the intention of the vehicle’s driver. It further updates the trajectory predictions on-line to provide an intention-based trajectory prediction and collision avoidance adapted to various driving manners and road/weather conditions. The effectiveness of this concept is demonstrated by a variety of simulations utilizing real human driving data in various scenarios including an intersection and a highway. The experiments are done in a specially developed driving simulation and a highly realistic third-party car simulator.

I. INTRODUCTION

Some of the key problems facing the inevitable introduction of autonomous cars lie in how autonomous cars will interact with human drivers. There have already been reports of accidents involving autonomous cars because they did not do what the human expected them to do. We take for granted the amount of predictive inference that human drivers do while driving everyday. When we see a car coming too close to us from the behind on the highway, we quickly predict that the driver wants to pass us. Similarly, at intersections, even the signs of a speeding driver quickly gets us to press on the brakes even before the car actually gets anywhere close to being an immediate threat. There are many examples of such proactive reasoning that humans are conditioned on and even trained by more experienced drivers. The human ability to predict (or failure thereof) and non-verbally communicate on the road can be tied very closely to safety on the road.

On the other hand, autonomous cars are often designed to be rule-based reactive control engines. More specifically, the collision avoidance, lane guidance, and even “adaptive” cruise control available on most new cars today are strictly reactive control systems based entirely on in-situ sensors. The common criticism for a purely reactive approach is that it is limited by the range of the sensor and the speed at which the system can react. On the other hand, a proactive autonomous driving controller would take into account the expected motion of the cars the autonomous vehicle is sharing the road with. To be truly effective, this method needs to utilize and analyze information about other cars that is beyond its own sensor’s range. This can be easily achieved through a vehicle-to-vehicle (V2V) network. Accordingly, the control algorithms we present are designed to assess the situation of the traffic around them and make safe driving decisions to avoid any potential conflict situations.

The main contribution of this paper is to introduce a proactive Collision Avoidance System (CAS) to enable autonomous cars to proactively avoid collisions with human cars on a shared road. In particular, this work focuses on the task of controlling an intelligent self-driven car surrounded by human-driven cars in two general cases: a highway scenario where all cars are driving the same direction on a congested highway, and an intersection scenario where human-driven cars are moving in transverse direction to the autonomous vehicle.

A. Literature Review

Systems offering assistance to the driver have attracted a significant amount of research interest in recent years. Basic reactive driving assistance systems warn the driver about difficulties on a road using in-situ sensors and may even execute an urgent action (such as applying brakes) to avoid a potential accident [1], [2]. These systems are now commonplace in automobiles, examples include reactive braking systems, and adaptive cruise control.

Improved intelligent reactive robotic systems utilize information from additional sensors, such as images from cameras
of all drivers (using in-situ sensors and V2V communication) and decision making. In the first task, we track the behavior objective into two general tasks shown in Fig. 3: prediction around it. To achieve this flexibility, we have split the vehicles, and in effect, changes in trajectories of the vehicles expressed intentions of the human drivers of surrounding ability to learn and adapt itself with respect to changes in industry due to the problems mentioned below.

The ability of the system to plan actions in advance requires the knowledge of the intentions of the human drivers. This task has been previously addressed using classification algorithms analyzing human activity [6], [7] and giving the intention or warning about possible unintended actions resulted by drowsiness or distraction [8]. Another challenging problem in developing the proactive algorithm is modeling of the human behavior which is required for prediction of the trajectory of the vehicle. This task has been solved using learning-based behavior models, for example, utilizing the Gaussian model [9], [10], grid-based estimation of the future state of the vehicle [11] or even computer vision predicting the lateral move of the car based on the position of the wheels [12]. Despite the presence of behavior modeling methods in research papers, they have not evaluated the benefits of intention awareness in context of human-robot interaction.

The next problem is the evaluation of the corrective action based on the possible outcomes which it can result. This task could be solved as an optimization task using various single-step and sequential decision-making techniques such as tree search, potential fields, Markov Decision Process (MDP). Using a tree search might be inconsistent and requires many function evaluations which can slow down the decision making [13]. The potential fields method represented the obstacles as areas repelling the agent [14] which are difficult to implement for continuously moving obstacles under uncertainty. A stochastic MDP [15], [16] and a Partially Observable MDP (POMDP) [17], where hidden intentions affect on obstacle behavior and transitions, would give an elegant solution, but they are very hard to solve, require many samples to establish the hidden links between states and are hard to check for the correctness of the solution, which is highly important in the road safety aspect.

**B. Outline**

The salient feature of the presented CAS algorithm is its ability to learn and adapt itself with respect to changes in expressed intentions of the human drivers of surrounding vehicles, and in effect, changes in trajectories of the vehicles around it. To achieve this flexibility, we have split the objective into two general tasks shown in Fig. 3: prediction and decision making. In the first task, we track the behavior of all drivers (using in-situ sensors and V2V communication) and make a prediction of their trajectories in the near future. For the purpose of this prediction, in Section II, we introduce dynamic models which store both human-driven car and autonomous car data and model their activity. In the decision making task, discussed in Section III, the proposed CAS system chooses an action in order to smoothly continue driving on the road while ensuring safety in the particular scenario by utilizing an optimization algorithm shown by a separate block in Fig. 3. Then, the desired action is applied to the car dynamics completing the feedback loop.

**II. Vehicle Behavior Modeling**

In this section, we define some relevant terms that are necessary for model development. The experience of the control algorithm will be stored in two separate behavior models: human-driven vehicle dynamic model (HDM) and autonomous vehicle dynamic model (ADM). Here, HDM represents the knowledge of the possible resulting states when the intention of a human and the current state of their car are given, while the ADM stores the future states of all the actions available to the autonomous vehicle with the given current state. These dynamic models are learned on-the-fly from human driving data and are able to predict the stochastic transition of each vehicle from one location to another in time. The estimation of the trajectories requires the knowledge of other vehicle’s location, velocity and its driver’s intention. This condition can be satisfied by establishing radio-frequency connections between all cars and transferring the data to each other using vehicle-to-vehicle (V2V) or vehicle-to-infrastructure (V2I) communication as has been explained in the work [18]. Meanwhile, the intention is assumed to be known and recognized correctly on the client-side, hence the intention classification task is beyond the scope of this work. By taking this and Markovian assumptions, we form a fully-observable system that allows to predict a next state. The vehicle’s state has been simplified to its location on the road in X, Y coordinates, angle of steering wheel, vehicle orientation, velocity, and acceleration.
of the vehicle. By knowing these states, the new state of the car can be predicted reasonably accurately over a small finite future interval (the trajectory estimation time step is defined as 1 second).

A. Gaussian Processes

This paper proposes a cost function optimization to compare the given outcomes estimated by multi-stage Gaussian Processes (GPs). The GP is a Bayesian nonparametric supervised learning method often used for mapping an input to a corresponding output [19]. The estimated trajectory represented by the displacement of the vehicle may be found as a function of time, velocity and ordinal number of the intention of the driver.

\[
f(x) \sim GP(m(x), K(x, x'))
\]

where \( m(x) \) is the mean of dataset given by \( D(x, y) \), and \( k(x, x') \) is a covariance kernel used for approximating the covariance of the dataset \( x \in X \) and other values \( x' \). Since the choice of the trajectory relays on human, the use of Radial Basis Function (RBF) kernel seems very natural. This kernel function shown in Eq. 2 is commonly used and allows to exponentially weight the error during the training. The prior of the GP is assumed to be zero, while the posterior distribution updates using Bayes law, where posterior distribution has a mean shown in Eq. 3 and a covariance in Eq. 4.

\[
m' = (K(X, X) + \omega^2 I)^{-1} y k(X,x_{k+1})
\]

\[
k' = k(x_{i+1}, x_{i+1}) - K^T(X,x_{i+1})K(X,X) + \omega^2 I)^{-1} k(X,x_{k+1})
\]

\[
\gamma = K(x_{i+1},X) - k(x_{i+1},x_{i+1})((K(X,X) + \omega^2 I)^{-1} y)
\]

In order to budget the number of kernels, we use the sparsification method proposed by Csato et al. [20] which enforces an upper bound on the cardinality of the basis vector and allocates RBFs so as to reduce the regression error; although other methods can be used. This basis vector set is updated only if the novelty of information shown in Eq. 5 for the new incoming data is above some threshold. If the threshold is not superseded, then only the weights and covariance are updated.

\[
\Delta v(t) \sim GP_t(v_0, t)
\]

\[
v(t) = v_0 + \Delta v(t)
\]

\[
(\Delta x(t), \Delta y(t)) \sim GP_t(x_0,y_0,v(t),t)
\]

\[
x(t) = x_0 + \Delta x(t)
\]

\[
y(t) = y_0 + \Delta y(t)
\]

In our system, we built 6 independent two-stage GPs - one for each intention. During the first stage shown in Eq. 6, GP estimates the change in the velocity \( \Delta V \) at each time for that particular intention \( i \). Then, the second stage shown in Eq. 8 predicts the change in the location \( \Delta x, \Delta y \) when GP is given time \( t \), velocity \( v(t) \) and intention \( i \). These GP estimators were used to develop HDM/ADM blocks shown in Fig. 4. Both blocks are similar, they store the transitions of the vehicle with respect to the action selected.

III. CAS ALGORITHM

The proposed system for controlling the vehicle has several levels shown in Fig. 5. It utilizes low level controlling signals to control the car such as steering, gas and brake, but this is not sufficiently effective for high-level decision making. For this reason, the driving task has been decomposed into two levels. At the low level, a control algorithm allows the car to follow the lane and keep the chosen speed. This task can be easily done by using Proportional-Derivative (PD) controller minimizing the error in actual lateral location and velocity of the vehicle. At the higher level, a decision-making algorithm chooses an action from the set of actions available to the autonomous vehicle defined as follows: keep driving (do nothing), speed up, slow down, change lane left, change lane right. These PD and CAS blocks are interconnected by an Interpretation Unit which translates the desired actions into PD control parameters.

The feedback is given by the observed state measurements in order to continuously update the current state of the vehicle and its probabilistic trajectory ADM model. The same update method works for the HDM model by monitoring other vehicles behavior and making changes in the model on the fly. These on-line updates allow us to build an adaptive control algorithm and adjust the models to changes in the dynamics of the vehicle and the environment.

A. Collision Probability

A collision between vehicles is defined by a condition when two or more vehicles come to the same location
at the same time. Let the probability of car $c_1$ in state $s \in S(x,y,t)$ be defined as a probability $p_{c1}(s)$. In addition, let the probability of car $c_2$ being simultaneously in state $s$ be defined as a probability $p_{c2}(s)$. Assuming that vehicles don’t intentionally collide with each other, we treat these two probabilities as independent so the probability of collision is the following joint probability:

$$p(\text{collision in } s) = p(c_1 = s, c_2 = s) = p_{c1}(s)p_{c2}(s). \quad (11)$$

For each particular time, the probability of collision $p_{\text{col}}$ will be given by

$$p_{\text{col}} = \sum_S (p_{c1}(s)p_{c2}(s)), \quad (12)$$

while the total probability of collision will be

$$p_{\text{col}} = \int_{t_0}^{t_{\text{max}}} \int_{x_{\text{min}}}^{x_{\text{max}}} \int_{y_{\text{min}}}^{y_{\text{max}}} (p_{c1}(x,y,t)p_{c2}(x,y,t)) \, dx \, dy \, dt, \quad (13)$$

where $t_{\text{max}}$ is time horizon and $(x,y)$ are the road space coordinates.

If there are multiple human-driven cars in the area surrounding the autonomous vehicle, the total risk of collision will take the following form:

$$p_{\text{col}} = \int_{t_0}^{t_{\text{max}}} \int_{x,y} (p_{c1}(x,y,t) \sum_{c_2} p_{c2}(x,y,t)) \, dx \, dy \, dt, \quad (14)$$

where $p_{c_2}$ is probability of the human-driven car $c_i \in c_2...c_n$ being in state $(x,y,t)$ assuming that there is only one human-driven car that can occupy this state at the time. In other words, we assume that there is no collision between human-driven cars.

### B. Optimization Formulation

The cost function balances the probability of collision with additional designer parameters such as preferences in actions and location on the road. The optimization problem then is

$$a = \arg \min_{a \in A} (J), \quad (15)$$

where $a$ is the preferred action and $J$ is the cost function shown below:

$$J = \sum_T \sum_{x',y'} [P_a(x',y'|x,y,v,t,a,A) \times P_b(x',y'|x,y,v,t,b,B)] + C(x',y',v',a) \quad (16)$$

The additional cost $C$ is associated with the cost of action in each particular situation such that

$$C(x',y',v',a) = \text{Cost}(a) + \text{Penalty}(v') + \text{Penalty}(x',y'), \quad (17)$$

where $\text{Cost}(a)$ is a cost of the candidate action according to the rank of the required effort, $\text{Penalty}(v')$ is a penalty for driving with a speed different from the one desired by the passenger. $\text{Penalty}(x',y')$ is a penalty for being off-road to motivate the car to follow the road, where

$$\text{Cost}(a) = \begin{cases} 
C_{a_1}, & \text{if } a = 1 \\
C_{a_2}, & \text{if } a = 2 \\
C_{a_n}, & \text{if } a = n,
\end{cases} \quad (18)$$

$$\text{Penalty}(v') = ||V_{\text{desired}} - v'||P_v, \quad (19)$$

$$\text{Penalty}(x',y') = \begin{cases} 
P_{\text{out}}, & \text{if } x',y' \notin \text{Road} \\
0, & \text{if } x',y' \in \text{Road},
\end{cases} \quad (20)$$

and where $C_{a_1}$, $P_v$, $P_{\text{out}}$ are manually defined penalty coefficients for the constrained optimization problem.

### C. Single-Step GP-based Collision Avoidance

When deciding which action to select next, we evaluate each action over the planning horizon using the ADM and HDM GPs. The planning horizon is larger than the time step so the best action is re-evaluated at each time step which is set to 0.05 sec in this work.

### IV. Simulations and Results

Since the main purpose of the proposed collision avoidance system is a cooperation with the environment and other human-driven cars, it is highly important that the drivers behave in a way similar to real humans. The use of real cars to prove the work of the system would be dangerous and also require a special enclosed area. On the other hand, driving a scaled car model on an experimental car testbed may involve dynamics and interactions which are different from the normal human driving experience. Interactive simulations have proven highly successful in training and evaluation of driving
behave[21],[22]. As such, we used the realistic Carnetsoft environment to obtain the human driving data. These data are used for training the proposed collision avoidance system for both qualitative and quantitative simulations.

A. Matlab Simulation Description

To prove the viability of the concept we have designed the MATLAB computing environment as a three lane highway with an intersection where both autonomous and human-driving vehicles were involved. This simulation environment was custom built to accommodate the wide array of vehicle dynamics, simulation resolution and driving scenarios. In this paper, we separately evaluate our CAS in both a highway and an intersection scenario.

In the highway scenario, the autonomous car, shown as a red rectangle in Fig. 6, is moving from south to north in the presence of manually controlled vehicles, shown as blue rectangles, traveling in the same direction. The simulation evaluates the performance of the autonomous vehicle in terms of the likelihood of collision and the reaction time; i.e. the reaction time is the time ahead of a possible collision at which a preventive action is taken. Intersection scenario represents the special case when the autonomous car is moving in transverse direction to the general traffic.

The simulation algorithm Alg. 1 has been built for generalized case satisfying both scenarios to utilize the dynamical equations of all vehicles and update the vehicle’s positions with a time interval of 50 ms. The short update interval guarantees the elimination of a possibility of skipping discrete states and avoids “jumping” one vehicle over another. In order to simulate the dynamics of a car, the simplified dynamical model has been described by the equations of motion based on the dynamic half car model [23].

B. Carnetsoft Simulation

The simulation algorithm discussed in Section IV-A allows to control the human-driven car with the view from the top, which is unnatural and gives the wrong feelings of driving. Even experienced driver cannot control the car without additional training; this makes worthless all the previous driving experience obtained from the real driving. Since the general idea of this work is to utilize driving experience for behavior model training - convenience of the driver is highly important for natural driving. For that purpose, RijksSchoolSimulator developed by Carnetsoft has been chosen. This simulator utilizes Logitech G27 control set, 4 monitors and software that gives the view from the cabin in front and side directions as shown in Fig. 7.

This driving simulator gives a tool to investigate driving-related scientific questions such as the effects of alcohol, distraction, drowsiness on driving and driver behavior modeling studies [24]. This software allows to prepare and perform behavioral experiments, and analyze the data. It samples the position of steering wheel, gas and brake pedals with 10 Hz frequency and utilizes its build-in dynamic

Algorithm 1: Simulation algorithm

Data: Transition model \(ADM\), Behavioral model \(HDM\), Dynamic function \(D\)

Result: Result of collision

\[
cart_n = [x_n, y_n, v_n], t = 0 ;
\]

while \(y < y_{\text{final}} \in R\) do

\[
[x'_n, y'_n, v'_n, t'_n] = D_n(x_n, y_n, v_n, t_n) \text{ for } n = [0..N_{\text{cars}}];
\]

\[
ADM.\text{update}(s_0, s'_0, a_0);
\]

\[
HDM.\text{update}(s_n, s'_n, a_n) \text{ for } n = [1..N_{\text{cars}}];
\]

if \(t < t_{\text{CAS}}\) then

\[
p_0 = ADM.\text{predict}(s_0, s'_0, a_0);
\]

\[
p_n = HDM.\text{predict}(s_n, s'_n, a_n) \text{ for } n = [1..N_{\text{cars}}];
\]

\[
a_0 = \arg \min (J(p_0(a_0), p_{1..n}));
\]

end

\[
a_{n=0..N} \rightarrow D_n ;
\]

\[
t = t + 0.05 ;
\]

end
functions to update the parameters of the cars. The graphic abilities of Carnetsoft’s simulator allows to render 3D world and reproduce night driving, rain, snow effects and sound effects. Its script language is designed to create any scenario, generate traffic and manage the data.

This realistic car simulator allows the humans to use their everyday driving skills without additional training and gets the results as close to the real driving as possible. To make this training of the behavior models possible and control the autonomous car, Carnetsoft is connected to another computer with MATLAB algorithm via UDP connection. The data required for CAS system are streamed to the computer, transformed into variables, processed and then used for training of behavior models. These data are also used in decision making process in order to find the best corrective action which is then sent back to the Carnetsoft simulator.

C. Training Behavior Models

Intention recognition using the vehicle’s states and inputs is an ongoing area of machine-learning research. Yet even with perfect knowledge of intention, collision avoidance is an important and challenging problem to solve. Therefore, we assumed that the intention would always be classified correctly (in fact we trained the drivers to utilize buttons to indicate intentions) and would not contribute to the uncertainty of the HDM training process. All intentions were extracted during driving experiments on-the-fly and formed the training vector with respect to the current zero location. This method allows us to update the behavior models on-line and adapt to changes in the environment.

We used five intentions such as changing lane to the left/right, slowing down, speeding up and keep going. The data collected during the HDM model training is shown in Fig. 8. The thin dotted lines represent the human-controlled car location data which were collected in the Carnetsoft simulator. The lines of circles show the prediction made by HDM utilizing the GP predictor in MATLAB.

The ADM model, which represents each trajectory of the autonomous vehicle, has been trained by the Monte-Carlo method where all possible actions from the action set were applied to the simulated autonomous vehicle. The uncertainty in transition was caused by the flexibility in the initial location of the autonomous car and the uncertainty in control applied to the dynamics of the vehicle as well as the uncertainty in all simplified parameters such as acceleration and steering angle. During training, a Matlab script sends control actions to the Carnetsoft simulator as the CAS system would to control the car. In response, the simulator sent back a vector of readings similar to the one used for training HDM. The data were collected and extracted from these vectors to train the GPs of ADM model. The estimated trajectory given by GP predictor is shown in Fig. 9. These ADM and HDM trajectory estimators were used by the CAS to define the probability of collision.
D. Collision Map

Single-step CAS algorithm utilized the GP predictor of the HDM to create the normal distributions of possibly occupied locations over the space. Due to the limitations of this work, we limit the human to one intention at each time instance. Future works will allow human drivers to have more than one intention at a time instance. The use of both ADM and HDM predictor gave us the two Gaussian distributions of possible locations of both autonomous and human-driven cars as shown in Fig. 10. In this figure, the prediction was made with 1 second interval for 0 to 5 seconds time-horizon. The distributions represented the probability of occupying the state $[x, y, t]$ by human-driven car (blue distribution) and autonomous car (green distribution). The intersection of two normal distributions as shown in Fig. 11 in red color represented the probability of collision with respect to the location and time and lead to a collision map required to compute the cost function.

E. Evaluation

To validate the presented CAS, we designed a qualitative experiment to investigate the real-time behavior of the autonomous vehicle cooperating with the real human driver. In that experiment different human-drivers were driving using the Carnetsoft simulator trying to reduce the distance to the autonomous car by using brake when in front of it, tail it too closely, or merge into its lane suddenly. The autonomous car reacted by changing its velocity and lane. An example of such a simulation is shown in Fig. 12 in form of trajectories resulted by the vehicles. In this example, two human driven cars were moving in the same lane as autonomous vehicle: one car in front and one behind. The last car expressed intention to speed up in dangerous proximity which lead the autonomous car to change its lane to the right as the CAS activated.

Quantitative results were obtained during a series of simulations for the parallel driving experiment on the highway shown in Fig. 6. The human-driven car shown by the blue rectangle was given an initial speed twice lower that the autonomous vehicle shown by the red rectangle. The simulation considered two speed modes: high speed (60 mph) and low speed (30 mph). For the purpose of statistically significant and consistent results, each setup was run for 100 iterations where the initial speed was varied by $\pm 10\%$ and the trajectory was held constant.

The results over 100 iterations shown in Fig. 13 and 14 compared the intention-based optimization and simple sensor-based reactive algorithms controlling the autonomous car. It should be noted that none of these algorithms got into collision in this specific setup. Proposed intention-based single-step optimization showed great improvement in the smoothness of acceleration profile used to avoid collision when compared to a sensor-based reactive approach. These result was expected since the reactive approach utilized only emergency brake, while the intention-based predictive algorithm took smooth normal action beforehand. In most of the cases, this smooth action led to a greater travel time but this difference was insignificant and could not be a metric of success of the algorithm. Maximum acceleration used to avoid a collision was defined as a parameter of comfort of the ride and registered the highest acceleration or deceleration (braking) taken by the autonomous car. As shown in Fig. 14 single-step algorithms showed good results comparing to the reactive one which was developed as “the last line of defense” and applied the maximum brakes in the very last moment before crash. It would be inaccurate to compare the reactive algorithm with others, but this comparison showed how the comfort of the passengers of the autonomous vehicle may be improved if the proactive approach is used in addition to the reactive one.

To prove the importance of utilizing the human’s intention for autonomous driving, we developed another experiment where a real human driver was driving on a highway in the Carnetsoft simulator. Their task was to drive close to the autonomous car and interact with it expressing their intention...
Fig. 15. Comparison of the average reaction time for all actions applied in freeway scenario involved continuous human driving. In intention-based mode, the intention of the driver was sent to the autonomous vehicle. Meanwhile, in no intention mode, autonomous car assumed that intention of the human is constant and performed the pseudo "reactive" driving utilizing the straight trajectory of the car.

followed by the corresponding actions in any sequence and a free manner. During this testing, we utilized two modes of the proposed CAS system. In normal mode, the intention of the driver was sent to autonomous vehicle which utilized it in order to take action. In pseudo "reactive" mode, we simulated the loss of intention such that the autonomous vehicle always predicted a straight trajectory as the intention of the human driven car. After 10 minutes of driving we compared the average between the moment when human driver expressed their intention and when autonomous car executed the action. Results shown in Fig. 15 show that the autonomous vehicle reacts faster utilizing the intention of the human driver for every given intention. In all cases when intention had not been transmitted, the autonomous vehicle reacted based on the current trajectory which produced very similar results. Meanwhile, the intention-based CAS reacted earlier, sometimes almost immediately.

V. CONCLUSION

The general concept of taking a human driver’s intention into consideration has been evaluated in quantitative simulations with different human drivers. Our simulations were performed in realistic driving conditions such as highway and intersection scenarios. They showed that the presented proactive collision avoidance system (CAS) is more effective with respect to the average reaction time and the probability of an accident when the driver intention is known. It gave a further improvement to the smoothness of vehicle motion through the ability to not only plan the action ahead, but also to start executing it as soon as another drivers’ intention is expressed or recognized.

REFERENCES


[16] Denis Osipychye, Duy Tran, Weihsia Sheng, Girish Chowdhary, and Ruili Zeng. Proactive mdp-based collision avoidance algorithm for autonomous cars. In *Cyber Technology in Automation, Control, and Intelligent Systems (CYBER), 2015 IEEE International Conference on*, pages 983–988. IEEE, 2015.


