A Collision Avoidance System with Fuzzy Danger Level Detection

Saina Ramyar
North Carolina Agricultural and Technical State…

Syed M Salaken
Deakin University

Abdollah Homaifar
North Carolina Agricultural and Technical State…

Ali Karimoddini
North Carolina Agricultural and Technical State…

2017 IEEE Intelligent Vehicles Symposium (IV)
June 11-14, 2017, Redondo Beach, CA, USA

A Collision Avoidance System with Fuzzy Danger Level Detection*

Zihao Wang[1], Saina Ramyar[1], Syed Moshfeq Salaken[2], Abdollah Homaifar[1], Saeid Nahavandi[2], Ali Karimoddini[1]

Abstract— **Collision avoidance is an essential component in advanced driving assistance systems, as it ensures the safety of the vehicle in near crash or crash scenarios. In this study, a collision avoidance system for lane change events is proposed which plans the trajectory based on the level of danger. The danger level is computed by a fuzzy inference system developed with naturalistic driving data to better capture the real-world factors, which may cause an accident. In addition, a fault determination classifier is introduced in order to determine the responsible driver in a near crash event. This system is evaluated on simulated naturalistic near crash events and the results demonstrate good performance of the proposed system.**

I. Introduction

In the united states, traffic accidents are the cause of about 37,000 fatalities and over 230.6 billion dollars in economic loss every year. According to a statistical projection from the National Highway Traffic Safety Administration (NHTSA), motor vehicle traffic fatalities rose 10.4 percent in the first half of 2016 compared with the previous year. Most of these accidents are due to human factor errors [1]. Therefore, driver assistance systems have the potential to reduce the risk of accidents as the system will take over control of the vehicle in dangerous scenarios and prevent potential crashes [2] [3].

However, dangerous events are very complex and the level of danger cannot easily be mathematically analyzed. Therefore, researchers in [4] presented a model-based algorithm that estimates how the driver would react to avoid a collision by assessment of the threat. In [5], a situation assessment (SA) algorithm is proposed which estimates the driver's behavior and then interacts with a collision avoidance (CA) system to initiate earlier brake interventions when there is a threat.

The majority of work on collision avoidance systems is focused on SA and reaction of drivers. However, with the advances in autonomous vehicles more sophisticated systems are required not only to avoid a dangerous situation but also to modify the maneuver for further safety and comfort.

*This work is partially supported by the US Department of Transportation (USDOT), Research and Innovative Technology Administration (RITA) under University Transportation Center (UTC) Program (DTRT13-G-UTC47). The fourth and sixth author would also like to acknowledge the support from Air Force Research Laboratory and OSD for sponsoring this research under agreement number FA8750-15-2-0116.

[1]Zihao Wang, Saina Ramyar, Ali Karimoddini and Abdollah Homaifar are with the department of Electrical and Computer Engineering, North Carolina A&T State University, Greensboro,NC 27403, USA {zwang1, sramyar}@aggies.ncat.edu, {homaifar, akarimod}@ncat.edu

[2]Syed Moshfeq Salaken and Saeid Nahavandi are with the Institute of Intelligent Systems Research and Innova- tion, Deakin University, Victoria, Australia {ssalaken,saeid.nahavandi}@deakin.edu.au

Among the most dangerous driving maneuvers, lane changing events are potentially the most risky maneuvers as they occur in a short time window. Also, according to the NHTSA 75% of accidents happen during lane changes. Therefore, detecting dangerous lane change events and modifying them to avoid accidents is essential for increasing traffic safety. As an example, in [6] a one-class support vector machine (OC-SVM) trained with naturalistic driving data was used to detect anomalies in lane change events.

In this study, a fuzzy danger-level detection system based on naturalistic driving data is developed to perform a reasonable risk assessment in lane-change events. The results of this danger level detection will be used in a collision avoidance system to modify the driving performance and reduce the level of danger. In addition, a fault determination classifier is trained with the same naturalistic data in order to further increase the safety of the maneuvers.

The remainder of the paper is organized as follows. In sections II and III, the fuzzy danger level detection and collision avoidance systems are developed. Section IV includes the fault determination classifier. The simulation results and conclusions and future work are given in sections V and VI respectively.

II. Fuzzy System Design

A fuzzy danger level detection model is introduced in this study to determine the danger level of an event and adjust driving behavior accordingly. To develop a realistic system, the fuzzy system is designed based on naturalistic near-crash driving data.

A. Naturalistic Driving Data

The dataset used in this study is derived from the 100 Car naturalistic driving data, which is a public data set collected by the Virginia Tech Transportation Institute. The data collection includes 100 vehicles with 241 primary and secondary driver participants with approximately 2,000,000 vehicle miles of driving. The data was collected during a 12 to 13 months period for each vehicle. The geographical coverage included Northern Virginia and the Washington, DC metropolitan area [7].

B. Fuzzy Membership Function Design

The Fuzzy system in this work is based on Mamdani's fuzzy inference method [8] as an attempt to control a steam engine and boiler combination by synthesizing a set of linguistic control rules obtained from experienced human operators. The fuzzy logic model consists of four

978-1-5090-4803-8/17/$31.00 ©2017 IEEE

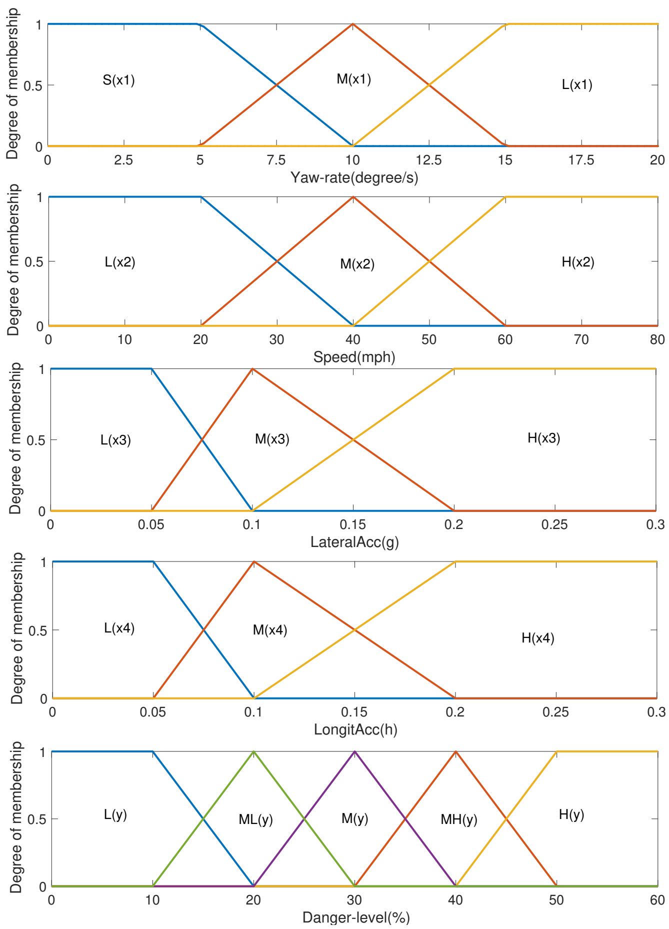

Fig. 1: Membership functions of input yaw-rate, velocity, lateral acceleration, longitudinal acceleration and output danger level

components, namely, fuzzification, a fuzzy inference engine, defuzzification, and a fuzzy rule base.

In this study, the fuzzy logic classifier adopts "Minimum" operator as the connective AND, "Maximum" as the aggregation method, and *centroid* as the defuzzification method.

The input membership functions are derived from the naturalistic driving data. The range of the measurements is used for the design of membership functions. The attributes are based on the analysis of the driving data around the time when near-crash event occurred. The membership functions are shown in Fig. 1, and the linguistic variables of all the inputs and output are listed in Tables I.

C. Fuzzy Rule Development

After the design of the membership functions for each input and output, 81 rules are developed which are the product of the number of the terms in each input linguistic variable X1, X2, X3, X4.

The $IF \dots And \dots Then$ rules for determining the event danger level are designed based on expert knowledge and are presented in the decision Table II. The danger level has five possible states which are expressed in percentage form.

Example Rule: Rule 1: IF x_1 is small AND x_2 is low AND x_3 is low AND x_4 is low THEN Danger level is Low.

TABLE I: Terms of linguistic variables

Terms of Yaw Rate(x_1)	Meaning	Membership Function
$S(x_1)$	Small Yaw Rate	$u_{11}(x_1)$
$M(x_1)$	Medium Yaw Rate	$u_{12}(x_1)$
$L(x_1)$	Large Yaw Rate	$u_{13}(x_1)$
Terms of Speed(x_2)	Meaning	Membership Function
$L(x_2)$	Low Speed	$u_{21}(x_2)$
$M(x_2)$	Medium Speed	$u_{22}(x_2)$
$H(x_2)$	High Speed	$u_{23}(x_2)$
Terms of La Acc(x_3)	Meaning	Membership Function
$S(x_3)$	Small LaAcc	$u_{31}(x_3)$
$M(x_3)$	Medium LaAcc	$u_{32}(x_3)$
$L(x_3)$	Large LaAcc	$u_{33}(x_3)$
Terms of Lo Acc(x_4)	Meaning	Membership Function
$S(x_4)$	Small LoAcc	$u_{41}(x_4)$
$M(x_4)$	Medium LoAcc	$u_{42}(x_4)$
$L(x_4)$	Large LoAcc	$u_{43}(x_4)$
Terms of Danger Level(x_5)	Meaning	Membership Function
$L(x_5)$	Low DL	$u_{51}(x_5)$
$ML(x_5)$	Medium Low DL	$u_{52}(x_5)$
$M(x_5)$	Medium DL	$u_{53}(x_5)$
$MH(x_5)$	Medium High DL	$u_{54}(x_5)$
$H(x_5)$	High DL	$u_{55}(x_5)$

TABLE II: $IF \dots And \dots Then$ rules

	Yaw Rate		Speed		LaAcc		LoAcc		Danger Level
IF	S	And	L	And	S	And	S	$Then$	L
IF	S	And	L	And	S	And	M	$Then$	ML
IF	S	And	L	And	S	And	L	$Then$	M
IF	S	And	L	And	M	And	S	$Then$	ML
IF	S	And	L	And	M	And	M	$Then$	M
IF	S	And	L	And	M	And	L	$Then$	MH
				...					
				...					
IF	L	And	H	And	L	And	S	$Then$	H
IF	L	And	H	And	L	And	M	$Then$	H
IF	L	And	H	And	L	And	L	$Then$	H

This rule implies that in a scenario where the yaw rate is small, speed of the subject vehicle is low, also the lateral and longitudinal acceleration are both low, then the danger level of subject's driving maneuver will be classified as a low risk maneuver. More specifically, this particular lane change event will be considered to reside within $0\% - 20\%$ danger level.

The proposed fuzzy danger level detection system is used to produce a basis for the crash avoidance system introduced in the next section to adjust driving behavior accordingly.

III. Collision Avoidance System

Once the danger level of an event is detected, the collision avoidance system will modify the maneuver to reduce the dangerous elements as much as possible.

The collision avoidance system proposed in this study is based on model predictive control (MPC). First, a brief explanation of MPC is provided, then the collision avoidance formulation is presented.

A. Model Predictive Control

In model predictive control an optimal control problem is solved over a finite horizon [9]. In this technique, the system is almost always formulated as a discrete state space system as in Eq. 1

$$x(k+1) = Ax(k) + Bu(u), \quad x(0) = x_0 \tag{1}$$

where $x(k) \in R^n$ and $u(k) \in R^m$ represent the system states and control inputs respectively. This model is used in a receding horizon open-loop optimization problem as follows:

$$J = min_{u(k)}\{\sum_{i=0}^{N} x(i)^T Rx(i) + u(i)^T Qu(i)\} \quad (2)$$

which is subject to a set of constraints:

$$Ex + Fu \leq \psi \quad (3)$$

where N denotes the prediction horizon, and R and Q are the optimization weights on the states and inputs respectively. Eq. 3 represent the equality and inequality constraints on states and inputs. Quadratic programming (QP) is used to solve this type of optimization problem.

B. Trajectory Planning

Each of the longitudinal and lateral dynamics of a vehicle are formulated as a double integrator system which is presented below:

$$d_{k+1} = d_k + v_k \delta t + a_k \frac{\delta t^2}{2} \quad k = 0, ..., N \quad (4a)$$

$$v_{k+1} = v_k + a_k \delta t \quad k = 0, ..., N \quad (4b)$$

where d_k, v_k, and a_k represent position, velocity and acceleration at time k respectively, and δ_t represents the sampling time.

The longitudinal maneuver can be planned using the following objective function:

$$J_x = \sum_{k=0}^{N} (v(k) - v_{des})^T R(v(k) - v_{des}) + Qa(k)^2 \quad (5)$$

where v_{des} is the desired longitudinal velocity. This objective function is designed to fix the velocity at a specific value with the smallest acceleration possible. The following are the constraints:

$$d_{max_k} = min(d^{cl}_{prec_i} - sg, d^{tl}_{prec_i} - sg) \quad t = 0..N \quad (6a)$$

$$d_{min_k} = max(d^{cl}_{front_i} - sg, d^{tl}_{front_i} - sg) \quad t = 0..N \quad (6b)$$

where d_{front} and d_{prec} denote the position of front and preceding vehicles and d^{cl} and d^{tl} denote the position in current lane and target lane respectively. The remaining constraints are on maximum and minimum values of velocity, acceleration and acceleration rate (jerking) which depend on governing rules and vehicle physical condition. The lateral maneuver can similarly be planned using the following objective function:

$$J_x = \sum_{k=0}^{N} (d(k) - d_{des})^T R(d(k) - d_{des}) + Qa(k)^2 \quad (7)$$

Where d_{des} is the desired lateral position of the vehicle, which in a lane change maneuver is on the center-line of the target lane. The position constraints for the lateral trajectory are determined according to the road boundaries. Lateral velocity, acceleration and jerking are defined based on vehicle conditions.

C. Maneuver Modification

In order to modify a maneuver, the trajectory must be re-planned with more safety considerations. For the trajectory planning, the framework proposed in [10] is utilized. In this work, longitudinal and lateral motions are planned independently and then are loosely coupled together.

Since most of the near crash lane change events happen in the presence of another vehicle, the first step in this work is to modify the minimum safety gap between the subject vehicle and surrounding vehicles. In this study, the safety gap is dependent on the danger level. Thus regardless of the subject vehicle being at fault or not, it attempts to increase the distance between itself and other vehicles to avoid a crash. The safety gap is determined as follows:

$$sg_{new} = sg_{default} \times DL \quad (8)$$

Where $sg_{default}$ is the default safety gap designed for normal scenarios, and sd_{new} is the new safety gap determined for the dangerous scenario. DL is the danger level of the scenario.

In the next step if the subject driver is at fault, in addition to the safety gap, other design parameters including velocity constraints will be adjusted so the controller can plan safer trajectories with less near crash risk for future maneuvers.

IV. Fault Determination Classifier

An additional feature in the proposed collision avoidance system is a fault determination classifier which determines whether the subject vehicle or the surrounding vehicle is responsible for the near crash/crash event.

If according to the classifier, the subject driver is responsible for most of the near crash/crash events, it means the driving performance must be improved. If an autonomous controller is driving the vehicle, its design parameters or constraints such as desired velocity must be modified. In addition, the vehicle equipment must be checked. It is possible that some of the sensors or actuators are broken and are not able to generate a well-rounded situational awareness.

On the other hand, if a human is performing most of the driving, the driver must be warned about his/her dangerous behavior, and an additional safety system such as distraction detection or emergency braking must be activated to ensure vehicle safety.

A. Pre-processing of Dataset

In order to train the classifier, first the data is pre-processed and the samples with missing or invalid data are removed. Finally, all the valid observations are combined and merged with proper class labels. The set includes 21,264 observations for the case when the subject driver is at fault, and 22,128 observations for when the other driver is at fault. Finally, the combined and processed dataset is shuffled to reduce learning difficulties at training time. In this study, a class of 0 denotes the subject driver is not at fault (i.e. the driver of the other vehicle involved in the accident is at fault) and 1 denotes that the subject driver is responsible for the accident.

B. Extreme Gradient Boosting

In this work, the problem is formulated as a classification task and Extreme Gradient Boosting algorithm [11] is used for identifying the responsible driver who is at fault. Extreme gradient boosting is a tree based boosting method which learns the task by minimizing the following regularized objective function:

$$\mathcal{L}(t) = \sum_{i}^{n} l(y_i, T_i^{(t-1)} + f_t(X_i)) + \Omega(f_t) \tag{9}$$

where t represents the number of iterations, i indicates the instance, X denotes the input vector at i-th instance, l represents a convex loss function that measures the distance between predicted output and truth, T represents the truth and

$$\Omega(f) = \gamma N + \frac{1}{2}\lambda||w||^2 \tag{10}$$

Here, N denotes the number of leaves in the tree and each f indicates one independent tree structure with leaf weights w. Naturally, $\Omega(f_t)$ will prefer a simple model over a complex one and will tend to reduce overfitting. Equation (9) is solved with the second order approximation technique [12] and exact greedy algorithm for split finding. The extreme gradient boosting algorithm also supports the approximate split finding algorithm for different computational environments [11]. Finally, the scores/decision of each tree are aggregated to determine the final output from the model.

V. Simulation and Results

In order to validate the performance of the proposed system, it is tested on an actual near-crash scenario chosen from the 100 car dataset. First the fault determination classifier is tested on part of the dataset to evaluate its performance. Then, an example based on near crash events is simulated to test the performance of the danger level detection and collision avoidance systems.

A. Fault Determination Classifier Performance

In this experiment, an extreme gradient boosting based model is used to identify if the subject driver is at fault. Once the dataset is prepared and shuffled, as described in section-IV-A, 25% of the available observations are separated as the final test set. The remaining 75% of samples are used in a 3-fold cross-validation setting to train the model. Afterwards, a grid search is conducted to find the optimal parameters for the extreme gradient boosting model based on 3 fold cross-validation. The optimal parameters and model information are shown in Table IV. The tuned model shows good performance on the test set. The confusion matrix is shown in Table -III. It can be easily seen that the positive predictive value is 0.8006. Therefore, we can say if the model classifies one incident as the fault of other driver (i.e. the subject driver is not at fault), then the likelihood that the other driver is really at fault is 80%. Similarly, a negative predictive value of 0.7947 indicates that the probability of the subject driver being at fault (given the model identify the incident at subject drivers fault) is more than 79%. This helps to conclude that the model is fairly confident on its decision and the accuracy from the model is good, considering the data includes measurement noise. Other related information regarding kappa, sensitivity and specificity is given in Table-IV.

In addition, the experiment is repeated 10 times to ensure that the model is not achieving the results by chance. In each run, the training set and testing set are sampled randomly from the available observations, the random number generator seed is shuffled and a separate extreme gradient boosting model is trained, tuned and finally tested on the test set. This process is also repeated for support vector machine, KNN classifier and generalized linear models. Fig-2 and 3 shows the accuracy and Area Under the Curve (AUC) score for the test set for each iteration in the form of a boxplot. It is clear that the extreme gradient boosting model performs better than the other models. Therefore, this model is used as an additional feature of this collision avoidance system.

TABLE III: Confusion matrix for extreme gradient boosting model with Accuracy = 0.798. Zero denotes the subject driver is not at fault.

		Reference	
		0	1
Prediction	0	4504	1122
	1	1072	4150

TABLE IV: Optimal extreme gradient boosting model

Optimal parameters		Model information	
maximum depth of tree	20	No information rate (NIR)	0.514
learning rate	0.1	P-value (Accuracy > NIR)	$< 2e^{-16}$
minimum loss reduction	1	Kappa	0.5951
subsample ratio of columns	1	Sensitivity	0.8077
minimum sum of weights	0.5	Specificity	0.7872

B. Collision Avoidance System Performance

In order to check the danger level with and without the proposed collision avoidance system, the events are simulated in MATLAB with model predictive control (MPC). The CVXGEN toolbox for convex programming [13] is used in this study for simulation of the model predictive controller. The design parameters for the collision avoidance system are given in Tables V and VI.

Near Crash Event Subject is traveling in right middle lane and begins to change into the far right lane while adjusting the radio. Subject does not signal or check the blind spot. When subject vehicle crosses the right lane line, there is a vehicle in the right lane and the subject must steer left to avoid sideswiping the other vehicle.

To replicate this event in MATLAB, the controller is designed with parameters chosen to result in a risky event. This approach can imitate the driving performance of a distracted or drowsy driver as they do not pay close attention to maintaining a safe distance with surrounding cars and keeping the speed within limits. First, the near crash maneuver is simulated with unsafe parameters, including high maximum value for speed constraint ($40m/s$ or $90mi/h$) and $32m/s$ for the desired velocity. These values are chosen based on the actual measurements of the vent.

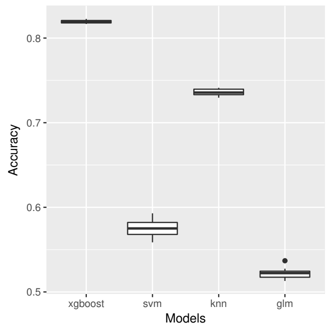

Fig. 2: Accuracy of prediction on OOB samples from 10 iterations of cross-validation. Extreme gradient boosting performs the best while the generalized linear model shows the worst performance in this experiment.

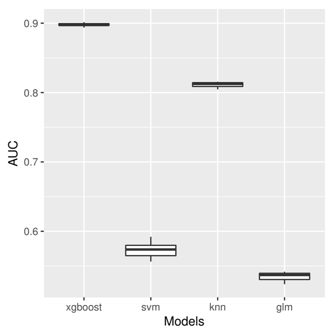

Fig. 3: Area under the curve (AUC) score of prediction on OOB samples from 10 iterations of cross-validation

TABLE V: Design parameters for the longitudinal trajectory

$v_x \in \{0, 30\}[m/s^2]$	$\Delta a_x \in \{-0.75, 0.75\}$	$N = 10$
$a_x \in \{-2, 2\}[m/s]$	$gap = 4[m]$	$\delta t = 0.5$

The longitudinal trajectory for the lane change maneuver is given in Fig. 4. It can be seen that due to its high speed, the subject vehicle accelerates and gets too close to the surrounding vehicle. As it is shown in the simulations, if the subject vehicle follows this longitudinal trajectory, it will collide with the nearby vehicle; therefore it returns to its original lane.

Since the subject vehicle immediately returns to the original lane after the attempted lane change, the lateral characteristics of the vehicle change according to the difference in its position (Fig. 5).

This near crash section of the maneuver is examined with the proposed danger level detection system and the results in Fig. 6 show the maximum danger level in this event is 70%.

Therefore, the safety gap parameter in the collision avoidance will be 1.7 times its initial value which was $4m$. In addition, the fault determination classifier with 95% accuracy determines that the subject driver is responsible for the near crash event (possibly due to the distraction), the desired speed and speed constraints are modified to represent the speed limit rules and ultimately increase safety of the maneuver.

TABLE VI: Design parameters for the lateral trajectory

$v_y \in \{-2, 2\}[m/s^2]$	$\Delta a_x \in \{-0.5, 0.5\}$	$N = 10$
$a_y \in \{-1, 1\}[m/s]$	$lanewidth = 7.5[m]$	$\delta t = 0.5$

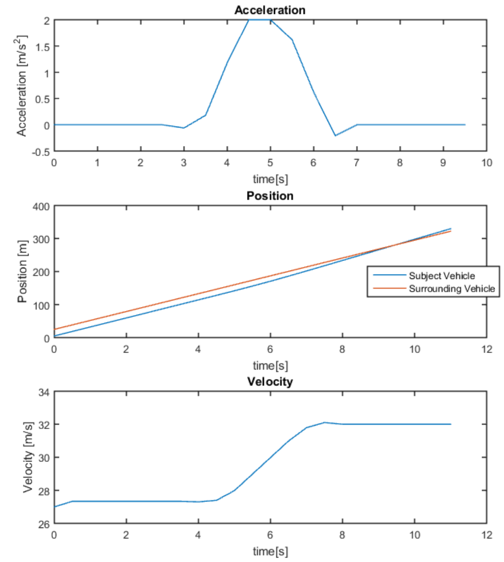

Fig. 4: Longitudinal trajectory of a near crash event

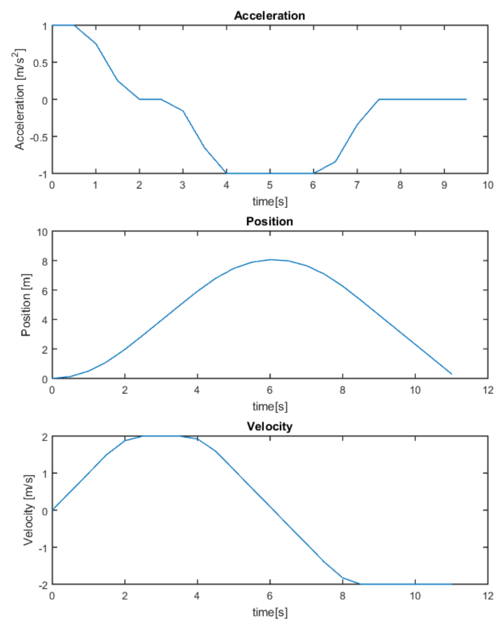

Fig. 5: Lateral trajectory of a near crash event

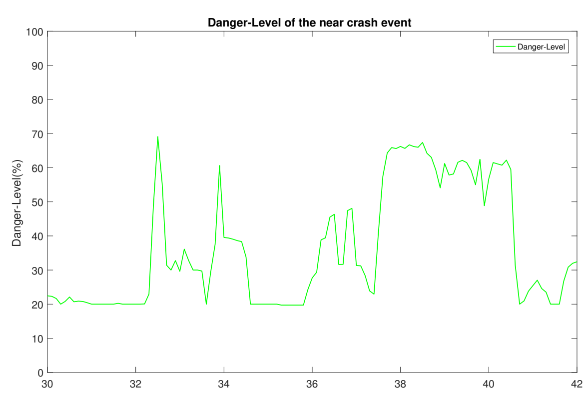

Fig. 6: Computed danger level of the near crash event

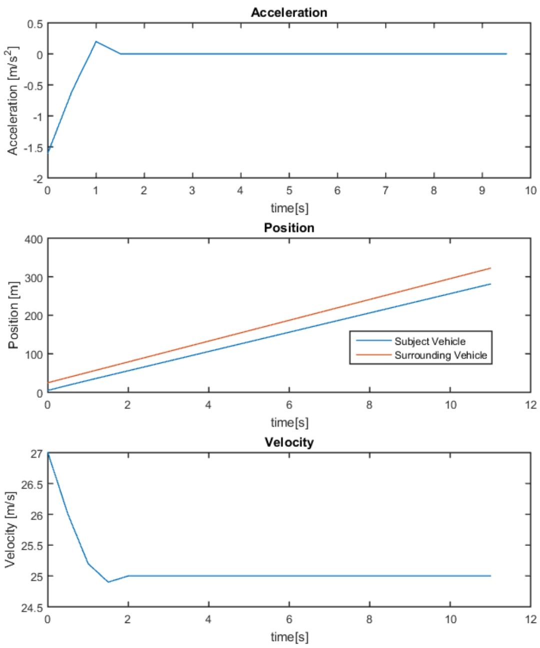

Fig. 7: Longitudinal trajectory of the modified crash event

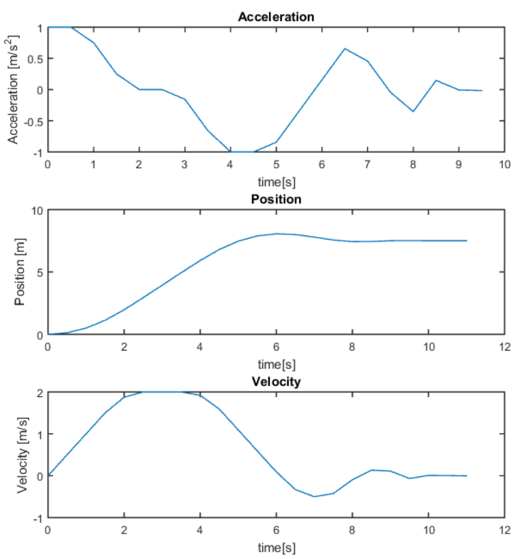

Fig. 8: Lateral trajectory of the modified near crash event

The modified longitudinal maneuver is shown in Fig. 7. It can be seen that the vehicle decelerates at the start of the maneuver to increase its distance with the preceding vehicle before moving to the adjacent lane. Consequently, the vehicle speed decreases.

The lateral trajectory planned with the proposed system is given in Fig. 8. Since the controller planned the longitudinal trajectory with respect to the surrounding vehicle, the vehicle does not need to return to its original lane. Therefore, there is less variation in the variables.

VI. Conclusions and Future Work

In this paper, a collision avoidance system for safe trajectory planning of lane change events is proposed. This collision avoidance system modifies the lane change trajectory according to the danger level of the event. A fuzzy danger level detection system is designed using naturalistic near crash events to determine a realistic risk level of a dangerous scenario. Afterwards, a model predictive controller generates safe longitudinal and lateral trajectories for the lane change maneuver based on the computed danger level. Moreover, an extreme gradient boosting algorithm is trained with the naturalistic data to identify the responsible driver in a near crash scenario. The results of this classifier can help to further adjust the performance of the proposed collision avoidance system.

In the future, both the fuzzy danger level detection system and fault determination classifier will be improved to include information about the surrounding vehicles. This should improve the performance of the system as the position and speed of surrounding vehicles has major influence on the level of danger in a near crash scenario.

References

[1] Vicki L Neale, Thomas A Dingus, Sheila G Klauer, Jeremy Sudweeks, and Michael Goodman. An overview of the 100-car naturalistic study and findings. *National Highway Traffic Safety Administration, Paper*, (05-0400), 2005.

[2] Anders Lie, Claes Tingvall, Maria Krafft, and Anders Kullgren. The effectiveness of electronic stability control (esc) in reducing real life crashes and injuries. *Traffic injury prevention*, 7(1):38–43, 2006.

[3] Ronald F Knipling and Walter W Wierwille. *Vehicle-based drowsy driver detection: Current status and future prospects*. Citeseer, 1994.

[4] Mattias Brannstrom, Erik Coelingh, and Jonas Sjoberg. Model-based threat assessment for avoiding arbitrary vehicle collisions. *IEEE Transactions on Intelligent Transportation Systems*, 11(3):658–669, 2010.

[5] Mattias Brannstrom, Jonas Sjoberg, and Erik Coelingh. A situation and threat assessment algorithm for a rear-end collision avoidance system. In *intelligent vehicles symposium, 2008 IEEE*, pages 102–107. IEEE, 2008.

[6] Saina Ramyar, Abdollah Homaifar, Ali Karimoddini, and Edward Tunstel. Identification of anomalies in lane change behavior using one-class svm. In *Systems, Man, and Cybernetics (SMC), 2016 IEEE International Conference on*. IEEE, 2016.

[7] F Kremser, M Gebhart, M Stecher, and K Bengler. The influence of a driving task on movement times of goal directed hand-arm movements. *Age (years)*, 25:2–7, 2013.

[8] Ebrahim H Mamdani and Sedrak Assilian. An experiment in linguistic synthesis with a fuzzy logic controller. *International journal of man-machine studies*, 7(1):1–13, 1975.

[9] Manfred Morari and Jay H Lee. Model predictive control: past, present and future. *Computers & Chemical Engineering*, 23(4):667–682, 1999.

[10] Julia Nilsson, Mattias Brännström, Erik Coelingh, and Jonas Fredriksson. Longitudinal and lateral control for automated lane change maneuvers. In *American Control Conference (ACC), 2015*, pages 1399–1404. IEEE, 2015.

[11] Tianqi Chen and Carlos Guestrin. Xgboost: A scalable tree boosting system. In *Proceedings of the 22Nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 785–794. ACM, 2016.

[12] Jerome Friedman, Trevor Hastie, Robert Tibshirani, et al. Additive logistic regression: a statistical view of boosting (with discussion and a rejoinder by the authors). *The annals of statistics*, 28(2):337–407, 2000.

[13] Jacob Mattingley and Stephen Boyd. Cvxgen: A code generator for embedded convex optimization. *Optimization and Engineering*, 13(1):1–27, 2012.