<http://www.diva-portal.org>

Postprint

This is the accepted version of a paper presented at *2019 IEEE International Conference on Vehicular Electronics and Safety (ICVES)*, 4-6 Sept. 2019, Cairo, Egypt, Egypt.

Citation for the original published paper:

Parseh, M., Asplund, F., Nybacka, M., Svensson, L., Törngren, M. (2019)
Pre-Crash Vehicle Control and Manoeuvre Planning: A Step Towards Minimizing
Collision Severity for Highly Automated Vehicles
In: *2019 IEEE International Conference of Vehicular Electronics and Safety (ICVES)*
Institute of Electrical and Electronics Engineers (IEEE)
<https://doi.org/10.1109/ICVES.2019.8906431>

N.B. When citing this work, cite the original published paper.

Permanent link to this version:

<http://urn.kb.se/resolve?urn=urn:nbn:se:kth:diva-264115>

Pre-Crash Vehicle Control and Manoeuvre Planning: A Step Towards Minimizing Collision Severity for Highly Automated Vehicles

Masoumeh Parseh*, Fredrik Asplund*, Mikael Nybacka[†], Lars Svensson*, and Martin Törngren*

*Department of Machine Design, ITM

[†]Department of Aeronautical and Vehicle Engineering, SCI

KTH Royal Institute of Technology

Stockholm, Sweden

Email: {mparseh, fasplund, mnybacka, larsvens, martint}@kth.se

Abstract—This paper addresses the control of a highly automated vehicle in a traffic scenario, where colliding with other traffic agents is unavoidable. Such a critical situation could be the result of a fault in the vehicle, late obstacle detection or the presence of an aggressive driver. We provide an approach that allows the vehicle’s control system to choose the manoeuvre that is likely to lead to the least severe injuries to vehicle occupants.

The approach involves the off-line solving of an optimal control problem to create a set of trajectories based on controlling the steering angle rate and the braking rate at the vehicle’s limits. Occupant injury severity prediction, based on accident data with the focus on impact location, is used by a real-time collision control algorithm to choose a trajectory from the pre-computed optimal set. A simulation set-up is presented to illustrate the idea of the collision control algorithm in a simple scenario involving dynamic traffic agents.

Index Terms—Trajectory planing, Collision control, Occupant protection, Automated vehicles

I. INTRODUCTION

Vehicle safety is of vital importance to the development of autonomous vehicles. Assuring safety for high levels of automation remains challenging despite the technological advances in recent years. When a highly automated vehicle cannot guarantee safety due to a failure, it must be able to fall back to a minimal risk condition [1]. The minimal risk condition can vary with the type and extent of failures, but in general a highly automated vehicle should have the ability to protect vehicle occupants in many known pre-crash scenarios [2], [3]. Fortunately, new opportunities present themselves at higher levels of automation, where vehicles should be able to perform all driving functions under certain conditions [4]. For example, highly automated vehicles have the ability to perform complex avoidance manoeuvres that are beyond a typical driver’s capabilities.

Solutions for these complex manoeuvres are mostly focused on collision avoidance, with much associated research dedicated to Threat Assessment (TA) techniques. Dahl et al. performed a literature review on TA methods, categorizing them into two high-level groups [5]: physical model-based and data driven-based methods. However, current practices cannot *guarantee* collision avoidance due to for instance oversimplification of

vehicle models, incomplete sensor data, environmental issues, the presence of aggressive drivers and calculation delays. As an example, oversimplifying a vehicle model influences the number and shape of possible trajectories and as a result decreases the number of evasive manoeuvres available in critical scenarios. Situations where a collision is unavoidable are thus likely to continue to exist and require the attention of vehicle manufacturers throughout the foreseeable future.

Indeed, based on available guidelines [1], [3] there is an opportunity to do more for occupant safety in such situations. For example, the functionality of collision mitigation system could be extended from pure collision avoidance to choosing manoeuvres that lead to the least harm in the event of a collision. This study considers obstacle handling, with the focus on passenger cars, in critical situations from this perspective, defining as a part of the collision mitigation functionality of a highly automated vehicle. The contribution of the study is an independent real-time collision control algorithm running in parallel to nominal planning, only activating as a last resort. It selects a trajectory among a pre-computed set based on accident data analysis. The occupant injuries in a situation, where a collision is unavoidable can thus be minimized. In the proposed algorithm we combine an optimization method with a data-driven approach: trajectory generation is optimization based, while trajectory selection is data-driven.

The paper is organized as follows: section II is a literature study that provides background information to motivate the necessity for the concept of collision control; section III introduces the optimal control formulation for trajectory set generation; section IV presents a pre-crash problem; section V presents the collision control algorithm; the paper ends with presenting, discussing and eliciting conclusions from the results.

II. LITERATURE STUDY

We provide a short study on trajectory planning and accident analysis. Planning techniques are introduced and an argument is made for the chosen method. Results of literature on injury

severity prediction based on accident data are briefly discussed, and later used as a cost function for trajectory selection.

A. Trajectory Planning

Trajectory planning plays an essential role in automated vehicles and mobile robotics research. According to [6], planning techniques are classified into four categories: *graph search, sampling, interpolating, and numerical optimization*. Among which, interpolating and graph search are the most common algorithms for real implementation applications.

A comprehensive study on sampling-based motion planning was performed in [7]. In this method, the way of planning is by sampling the configuration space. According to this paper, sampling based methods are fast, but they do not guarantee a solution and when a solution exists it might not be optimal.

In optimization based approaches, a cost function is either minimized or maximized and subjected to vehicle dynamics and or environmental constraints. The successful use of optimization method for trajectory planning is presented in work done by [8], where a vehicle completed a route of 103km autonomously. Trajectory planning using Model Predictive Control (MPC), where an optimal control problem is formulated, has shown significant advantages in the field. MPC is an online optimization method, where the optimal trajectory is calculated at each time step based on the vehicle's current position and the changes in the environment such as dynamic obstacles. An example of using MPC for trajectory planning for an autonomous vehicle operating on rough terrains is shown in [9], where MPC is preferred over, sampling method.

Even though optimization methods may require high computational power, they allow for the off-line computation of trajectories. The use of pre-computed trajectory set can be seen in research done in robotics. An example is path planning for Mars exploration rover, where the robot uses a fixed set of pre-computed paths and is equipped with on-board stereo vision [10]. The robot builds a goodness map of the surrounding terrain, where each trajectory is associated with a score. When trajectory selection is activated, the robot evaluates each trajectory in the set and follows the trajectory that has the highest score, meaning safer and easier to drive. Similar work in avionics for trajectory planning of an unmanned autonomous helicopter is performed in [11], where the pre-computed set is used as an emergency manoeuvre library. The set is calculated in a way to increase the probability of having at least one collision-free trajectory to select during the run-time. Trajectory planning based on a pre-computed set for safe stop planning is also presented in the work done by [12].

Collision mitigation demands the vehicle to drive on limits of handling. This and a short time to collision requires a more comprehensive vehicle model to produce more realistic trajectories. This in return results in a high computational burden. The use of pre-computed trajectory set provides the opportunity for a more detailed vehicle model without compromising the run-time operation of the vehicle. This also allows checking for possible manoeuvres at an eminent crash quickly. To be able to consider and incorporate the vehicle

constraints, an optimization approach is used.

Studies on vehicle post-impacts to reduce the possibility of a second impact can be seen in work by [13], but pre-crash studies are more focused on collision avoidance.

B. Accident Analysis

Although active and passive vehicle safety technologies continue to develop, accidents that lead to severe injuries and fatalities still happen. Malliaris identified more than twenty factors that contribute to injury severity of a crash by assessing the National Automotive Sampling System Crash-worthiness Data System (NASS/CDS) during years 1988-1995 [14]. The most influential factors were used to build an algorithm based on a regression model to predict injury risk of vehicle occupants. Probability estimation of injury severity was based on the Maximum Abbreviated Injury Scale (MAIS). High casualty is associated with MAIS 3+ and lower casualty with MAIS 2+. Based on the results, if we assume that the vehicle body is a clock, and plot the direction of force as a function of car crash severity, the 8-10 and 2-4 o'clock impacts are dominant in terms of seriousness of the casualty. 11-1 o'clock impacts are second and 5-7 o'clock are the last. The front of the vehicle is equipped with energy absorption materials and also the use of seat belt and presence of airbag make this part to be more resistant to crash damage [15]. This study also identified that front collisions were usually involved with a side crash of another vehicle, and the occupants involved in the side crash were seriously injured as opposed to occupants of a frontal crash. Bahouth et al. also showed that only 2% of people involved in a rear crash were seriously injured.

Another study based on NASS-CDS was performed during years 1999-2008 [16]. The goal was to develop a model to predict the probability that a vehicle involved in a crash, contains occupants with serious injuries. In this study, serious injury was defined as the Injury Severity Score (ISS) greater or equal to 15. Crash factors or the inputs for this model were identified as Delta-V, i.e. total change in the vehicle's velocity, crash direction, vehicle type, single or multiple impacts, belt use, age and gender. The crash direction was defined as a 1-12 o'clock and similar to [15]. Based on the analysis of this paper, the highest injury rates were associated with right and left impacts with 6.4% and 8.5%, respectively.

Nishimoto et al. performed a study based on comprehensive police data during years 2000-2011 in Japan to create an injury prediction algorithm. This study also showed that the probability of severe injuries as a function of Delta-V is higher for side impacts in comparison to other impacts [17]. A study in Sweden was performed during years 2003-2013 based on the Swedish Traffic Accident data Acquisition (STRADA) with the aim to develop an injury severity prediction algorithm for vehicle occupants [18]. High severity injury was associated with head-on or side outside an intersection collision, containing 12% of accidents with ISS>8 and rear-end collisions with 1.1%.

In our work, the results of these studies are used as input information to build a cost function. This cost function determines a strategy to execute a trajectory in a pre-crash

scenario in order to reduce harm to vehicle occupants.

III. TRAJECTORY LIBRARY OPTIMAL CONTROL FORMULATION

This section introduces an optimal control problem to create a trajectory library. The trajectories are created ahead of run-time and are stored in a set. The set then goes through an online process to choose one trajectory based on a cost function.

A. Optimal Control Problem Formulation

The optimal control problem is formulated as,

$$\begin{aligned} \min_{x,u} \quad & \int_0^{t_f} ((X_g - X(t))^2 + (Y_g - Y(t))^2) dt + v_x(t_f) \\ \text{s.t.} \quad & \dot{\mathbf{x}}(t) = \mathbf{f}(\mathbf{x}(t), \mathbf{u}(t)), \\ & G(\mathbf{x}(t), \mathbf{u}(t)) \leq 0, \\ & \mathbf{x}(0) = \mathbf{x}_0, \\ & \mathbf{x}(t_f) = \mathbf{x}_T, \end{aligned} \quad (1)$$

where $\mathbf{x}$ is the state vector, $\mathbf{u}$ is the control vector, $\mathbf{f}$ is the vehicle model, $\mathbf{x}_0$ and $\mathbf{x}_T$ are the initial and final states of the vehicle, respectively. X_g and Y_g are the goal positions of the vehicle in the global coordinates, G contains the path constraints and t_f is the final time.

B. Vehicle Model

A dynamic bicycle model is used to present the ego vehicle's motion. The longitudinal behaviour is controlled through individual braking of front and rear axles and the lateral motion through steering angle of the front axle. Braking and steering are controlled through the rate of change of braking forces and rate of change of steering angle. The system dynamics are described as follows,

$$\begin{aligned} \dot{X} &= v_x \cos(\psi) - v_y \sin(\psi), \\ \dot{Y} &= v_x \sin(\psi) + v_y \cos(\psi), \\ \dot{\psi} &= r, \\ \dot{v}_x &= \frac{1}{m}(F_{x_f} \cos(\delta) + F_{x_r} - F_{y_f} \sin(\delta)) + \dot{\psi} v_y, \\ \dot{v}_y &= \frac{1}{m}(F_{x_f} \sin(\delta) + F_{y_r} + F_{y_f} \cos(\delta)) - \dot{\psi} v_x, \\ \dot{\psi} &= \frac{1}{L_z}(L_f F_{x_f} \sin(\delta) + L_f F_{y_f} \cos(\delta)) - L_r F_{y_r}, \\ \dot{\delta} &= \zeta, \\ \dot{F}_{x_f} &= dF_{x_f}, \\ \dot{F}_{x_r} &= dF_{x_r}, \end{aligned} \quad (2)$$

where X , Y and ψ are the global coordinates and heading of the vehicle body. v_x , v_y and r are the longitudinal, lateral and rotational velocity of the vehicle body. F_{x_f} and F_{x_r} are the longitudinal forces on front and rear axles, respectively. δ is steering angle, and ζ is its rate of change. dF_{x_f} and dF_{x_r} are the rate of change of braking for front and rear axles, respectively. Vehicle's parameters are introduced in Table I.

Table I
EGO VEHICLE PARAMETERS

Parameter	Description	Value
m	Mass of ego vehicle	2150 kg
I_z	Moment of Inertia	3075 kgm ²
L_f	Distance between front axle and center of gravity	1.4m
L_r	Distance between rear axle and center of gravity	1.45m
h	Center of gravity to ground height	0.738m
C_f	Front axle tyre stiffness	22
C_r	Rear axle tyre stiffness	22
μ	Friction coefficient	0.763

C. Tyre Model

A simplified version of the nonlinear Pacejka Magic Formula tyre model is used in this paper [19]. The lateral tyre forces on the front and rear axles are defined as follows, respectively,

$$\begin{aligned} F_{y_f} &= -\sin(\tan^{-1}(C_f \alpha_f)) \sqrt{(\mu F_{z_f})^2 - F_{x_f}^2}, \\ F_{y_r} &= -\sin(\tan^{-1}(C_r \alpha_r)) \sqrt{(\mu F_{z_r})^2 - F_{x_r}^2}, \end{aligned} \quad (3)$$

where C_f is the front and C_r is the rear axle tyre stiffness. The slip angles for front and rear axles are defined as,

$$\begin{aligned} \alpha_f &= \tan^{-1} \left(\frac{v_y + L_f \dot{\psi}}{|v_x|} \right) - \delta, \\ \alpha_r &= \tan^{-1} \left(\frac{v_y - L_r \dot{\psi}}{|v_x|} \right). \end{aligned} \quad (4)$$

Load transfer between axles is considered. Therefore, the normal loads on the front and rear axles are as follows,

$$\begin{aligned} F_{z_f} &= (mgL_r - h_{CG}ma_x)/L_f + L_r, \\ F_{z_r} &= (mgL_f + h_{CG}ma_x)/L_f + L_r, \end{aligned} \quad (5)$$

where acceleration, a_x , is defined as (9),

$$a_x = (F_{x_f} + F_{x_r})/m. \quad (6)$$

D. Optimization Method and Tool

The optimization toolbox used in this paper is CasADi; an open source tool for nonlinear optimization and optimal control [20]. By using the direct multiple shooting method, the optimization problem is discretized in time. In this method, both states and controls are decision variables in NLP. The control interval is divided into N sub-intervals, where it is piece-wise constant. In each control interval, there exists a starting state and a fixed control. An approximate solution of the ordinary differential equations is solved using the numerical initial value problem methods. The resulting NLP problem is solved by Interior point method with IPOPT. To reduce the number of infeasible problems, each optimization problem is solved with the last successful solution as the initial guess for the next step.

E. Trajectory Set Generation

A fine-grained grid of candidate final positions is defined as goal states of the ego vehicle. For each goal position an optimal control problem, as defined in section III, is solved to create an optimal trajectory between the initial position of the vehicle and goal positions given the constraints and a cost. As a result, a set containing optimal trajectories is created off-line for different initial velocities. During run-time, a subset of trajectories is selected corresponding to the current velocity.

Therefore, reachable goal positions are accessible in a short amount of time. All derived trajectories are combined braking and steering; designed to enable maximum braking potential of the vehicle, while keeping steerability. As an example, a subset of trajectories for $v_x = 15m/s$ and $v_y = 0$ is presented in Fig. 1, for a fixed X_g and a range of Y_g .

IV. PROBLEM SETUP

In this section, a scenario at a basic intersection, involving an ego vehicle and two dynamic traffic agents, is presented as a pre-crash scenario. Other traffic agents are modelled with constant acceleration and kinematic bicycle model, and are assumed to be passenger cars in the following scenarios.

A. Assumptions

It is assumed that the ego vehicle can detect position, velocity and heading angle of other traffic agents. Post-crash effects and secondary impacts are ignored. We assume that there exists accident data comprehensive enough to build a relationship between vehicle parameters and occupants injury severity. The intersection is assumed to be known and coded in the vehicle's coordination system. We assume that other traffic agents have pre-defined paths.

B. Pre-crash Scenario Setup

We present a scenario, where a collision is possible between the ego vehicle and other dynamic agents. The ego vehicle is plotted by red boxes at each time step, while other traffic agents are in magenta and blue. At the time of the collision detection, colliding vehicles are presented with the same color. Collision detection is performed through robotics occupancy grid [21]. The area is meshed into small grids, and at each time step all cells overlapped by physical objects are considered occupied. Fig. 2 is a scenario where a collision is first detected at $t = 1.0716s$ between the ego vehicle and the traffic agent (2), given that the vehicle follows either of the trajectories represented by the light blue boxes. These trajectories are removed from the set for the next time step. At the next time step, $t = 1.1112s$, a subset of collision free trajectories from the previous step collides with the same traffic agent. These are shown with black boxes and then removed from the set. Fig. 3 presents a later time step, $t = 1.1509s$, where collision free trajectories in the set now collide with traffic agent (1).

Following different trajectories leads to different impact locations of traffic agents. For example, Fig. 2 shows side impacts with traffic agent (2) and Fig. 3 shows front impacts with traffic agent (1) at a later time step. These figures are zoomed in the middle of intersection for the purpose of clarification. Thin black boxes in Fig. 3 show collision at previous time steps. The red boxes are collision free trajectories.

C. Desired Outcome

The final manoeuvre of the ego vehicle consists of two steps. As the first step, the ego vehicle is braking along its longitudinal path, which is defined as a hard optimization constraint and the velocity is minimized at goal positions as a soft constraint. This step is achieved off-line. The planning horizon is small,

Fig. 1. Trajectory subset for initial velocities of $v_x = 15m/s$ and $v_y = 0$ at $X_g = 20$ and $Y_g = [-10 \ 10]$.

where the longest trajectory has a final time less than $2s$ and all trajectories are interpolated to have the same length. In the second step, one trajectory that leads to the least injury severity for occupants is chosen for execution based on a cost, when no collision-free trajectory exists in the pre-computed set or not enough time is available to check all possible trajectories.

V. PRE-CRASH COLLISION CONTROL

Based on the results of accident data literature, side impacts are associated with the highest injury severity, while frontal and rear crashes are usually less severe. Therefore, the trajectories leading to side impacts receive the highest cost, front impacts the second and the rear impacts have the lowest cost. In this section, we present the algorithm for final trajectory selection.

A. Trajectory Selection Cost Function

The ultimate aim is to allocate a cost to each colliding trajectory based on impact location and crash direction. In this study the cost function is only based on the type of collision -side, rear or front. Achieving the ultimate aim would require access to more detailed accident data.

B. Type of Collision

Determining the type of collision starts with calculating all the vertices of all vehicles, assuming that the ego vehicle is equipped with measurement tools. Vehicles' bodies are then sectioned into points, where the distance between each point on the ego vehicle's body and all points on the other traffic agents are calculated. The smallest and the second smallest distances and the relative yaw angle are used to decide the type of collision. If each pair of points for the smallest and second smallest distances belong to the front of both vehicles, then the collision is a front collision. If they belong to the front of one vehicle and the side of the other, the collision is a side one, and if they belong to the front of one vehicle and back of the other, it is a rear impact. Due to occupancy grid calculations, the smallest and the second smallest distances might be associated with different types of collision. This could happen if some parts of an occupied cell is outside the traffic agents' boxes.

Fig. 2. Collision detection at time steps $t = 1.0716s$ and $t = 1.1112s$ with traffic agent (2).

In this case, a conservative decision is made and the collision is identified as the most severe of the two types of collision. Here, we present the collision control algorithm.

Algorithm 1 Collision Control

```

V = vehicleCurrentVelocity
TrajectorySubset = TrajectorySet(V)
for i = 0 : Δt : tf do
    calculateTrafficAgentInfo(X', Y', ψ')
    for j = 1 : length(TrajectorySubset) do
        collision = occgrid.checkOccupancy
        if collision == 1 then
            saveTrajectorySubset(j).(X, Y, ψ)
            saveTrafficAgentInfo(X', Y', ψ')
            calculateTypeOfCollision(front, rear, side)
            cost = evaluateCostOfTrajectorySubset(j)
            AssignCostToTrajectorySubset(j)
            RemoveTrajectorySubset(j)
            AddTrajectorySubset(j) to collisionSet
        end
    end
end
if collision == 0 then
    FollowNominalPlanner
else checkcollisionSet;
    selectLowestCostOfTrajectorySubset(j)
end
return : trajNumberOfLowestCost

```

VI. RESULTS AND DISCUSSIONS

The collision control algorithm enables decision making on how vehicles should collide with each other to increase occupant safety if a collision becomes unavoidable. Fig. 4 visualizes what would be the results from applying this algorithm on the scenario presented in Fig. 2 and Fig. 3. A collision between the ego vehicle and traffic agent (2) is detected, first at $t = 1.0716s$ and then at $t = 1.1112s$. At the next time step, $t = 1.1509s$, another collision happens with traffic agent (1). After collision detection, considering an unavoidable collision, type of collision needs to be determined

Fig. 3. Collision detection at time $t = 1.1509s$ with traffic agent (1).

for final manoeuvre execution. Based on the proposed method the impacts with traffic agent (2) are categorized as side impacts and collisions with traffic agent (1) are categorized as frontal crashes. Therefore, the choice is between a side collision, with higher cost, at an earlier time or a later frontal crash with a lower cost. Thus, in order to execute a manoeuvre with the lowest cost, the decision is to follow one of the trajectories that lead to a frontal crash at $t = 1.1509s$. Among the lowest cost subset, the first trajectory in the set is executed due to time-sensitiveness of the situation. This chosen trajectory is shown in Fig. 4 with magenta colors and the associated decision variables of the optimal control, defined in section III, are presented in Fig. 5. The studied collision control algorithm thus allows a highly automated vehicle to choose between crashes that are likely to be of different severity with the data available today.

One should note that there is a trade-off between the choice of vehicle model, computation time and the number of available pre-computed trajectories in the set. A more complex vehicle model with constraints at limits means more accurate trajectories, but it is more computationally expensive. In the work done by [19], the dynamic bicycle model with nonlinear tyres and load transfer is compared to a full dynamics CarMaker model. The results showed that the overall behavior of the CarMaker model is similar to the simplified bicycle model. Therefore, the model presented here is a good representation of the vehicle motions in terms of both accuracy and computational cost. The number of trajectories in the set also increases as the number of final positions increase. More trajectories are associated with more options for trajectory selection, but also requires more time when computing the trajectories off-line and when checking them during run-time.

Other factors influencing crash severity such as velocity, direction of force and occupant position could be included in the cost function. To do so would be complex, and each factor deserves its own study to ensure that the refinement of the cost function is correct.

Fig. 4. Final executed trajectory of ego vehicle leading to a front collision with traffic agent (1).

Fig. 5. Decision variables for the executed final trajectory.

VII. CONCLUSION AND FUTURE WORK

This paper contributes to active safety studies by developing an optimal control based algorithm that quantifies the concept of occupant protection, while following the guidelines mentioned in [1], [2] and [3]. This algorithm can be used to control a highly automated vehicle in a pre-crash scenario, executing a last resort command to reduce crash severity. The proposed collision control algorithm can take into account different body parts of the impacted vehicle, relative heading angle between the vehicles and their longitudinal and lateral position at the time of collision. This is possible due capabilities of highly automated vehicles. Therefore, the final chosen trajectory from the set has the potential to reduce harm to vehicle occupants.

Exploring alternative techniques for trajectory generation such as run-time MPC and evaluating the algorithm in different scenarios are parts of the future studies. More detailed accident data, velocity and occupant position will also be used to build a more detailed cost function for run-time selection process.

ACKNOWLEDGMENT

Support from ECSEL PRYSTINE with grant agreement No. 783190, TRENOP (Transport Research Environment with

Novel Perspectives) at KTH and AutoDrive H2020-ECSEL is gratefully acknowledged.

REFERENCES

- [1] Federal automated vehicle policy, US Department of Transportation, National Highway Traffic Safety Administration (NHTSA), Sep 2016.
- [2] Benefits estimation framework for automated vehicles operations, Prepared for: Intelligent Transportation System Joint Program Office, U.S. Department of Transportation, Final Report, August 2015.
- [3] Automated driving systems, A vision for safety 2.0, NHTSA, U.S. Department of Transportation, September, 2017.
- [4] SAE, J3016, Taxonomy and definitions for terms related to driving automation systems for on-road motor vehicles, June 2018.
- [5] J. Dahl, G. Rodrigues de Campos, C. Olsson, and J. Fredriksson, "Collision avoidance: a literature review on threat-assessment techniques", IEEE Transactions on Intelligent Vehicles, vol.4, no.1, March 2019.
- [6] D. González, J. Pérez, V. Milanés, and F. Nashashibi, "A review of motion planning techniques for automated vehicles", IEEE Transaction on Intelligent Transportation System, Vol.17, No. 4, 2016.
- [7] M. Elbhanawi, and M. Simic. "Sampling-based robot motion planning: a review", IEEE Access, practical innovations: open solutions, 2014".
- [8] J. Ziegler, P. Bender, T. dang, and Ch. Stiller, "Trajectory planning for BERTHA- a local, continuous method", IEEE Intelligent Vehicles Symposium (IV), June 8-11, 2014, Dearborn, Michigan, U.S.A.
- [9] L. Bascetta, D. Cucci, G. Magnani, M. Matteucci, D. Osmankovic, and A. tahirovic, "Towards the implementation of a MPC-based planner on an autonomous All-Terrain vehicle", Workshop on Robot Motion Planning: online, reactive and in real time, IROS, Portugal, 2012.
- [10] A. Rusu, S. Moreno, Y. Watanabe, M. Rognant, and M. Devy, "State lattice generation and nonholonomic path planning for a planetary exploration rove", 65th Astronautical Congress, Canada. pp.953, 2014.
- [11] S. Arora, S. Choudhury, D. Althoff, and S. Schere, "Emergency maneuver library-ensuring safe navigation in partially known environments", IEEE International conference on Robotic and Automation, May 2015.
- [12] L. Svensson, L. Masson, N. Mohan, E. Ward, A.P. Brenden, L. Feng, and M. Törngren, "Safe stop trajectory planning for highly automated vehicles: an optimal control problem formulation", IEEE Intelligent Vehicle Symposium, China, 2018.
- [13] D. Yang, B. Jacobson, M. Jonasson, and T. Gordon, "Minimizing vehicle post impact path lateral deviation using optimized braking and steering sequences", International Journal of Automotive Technology, Vol. 15, No. 1, pp.7-17, 2014.
- [14] A. C. Malliaris, K. H. Digges, and J. H. DeBlois, "Relationships between crash casualties and crash attributes", SAE Technical Paper series, International Congress and Exposition Detroit, Michigan, 1997.
- [15] G. Bahouth, K. Digges and C. Schulman, "Influence of injury risk thresholds on the performance of an algorithm to predict crashes with serious injuries", 56th AAAM Annual Conference, 2012.
- [16] D. W. Kononen, C.A.C. Flanagan, and S. Wang, "Identification and validation of a logistic regression model for predicting serious injuries associated with motor vehicle crashes", Accident Analysis and Prevention, Vol.43, pp.112-122, 2011.
- [17] T. Nishimoto, K. Mukaigawa, Sh. Tominaga, N. Lubbe, T. Kichi, T. Motomura, and H. Matsumoto, "Serious injury prediction algorithm based on large-scale data and under-triage control", Accident Analysis and Prevention, Vol.98, pp.266-276, 2017.
- [18] R. Buendia, S. candefjord, H. Fagerlind, A. Bálint, and B.A. Sjöqvist, "On scene injury severity prediction (OSISP) algorithm for car occupants", Accident Analysis and Prevention, Vol.81, pp.211-217, 2015.
- [19] J. Mårtensson, M. Nybacka, J. Jerrelind, and L. Drugge, "Evaluation of safety distance in vehicle platoons by combined braking and steering", AVEC12.
- [20] J. A.E. Andersson, J. Gillis, G. Horn, J.B. Rawlings, and M. Diehl, "CasADi- a software framework for nonlinear optimization and optimal control", Mathematical Programming Computation, In Press, 2018.
- [21] A. Elfes, "Using occupancy grids for mobile robot perception and navigation", COMPUTER, vol.22, No. 6, pp.46-57, 1989.