Planning of Safe Trajectories in Dynamic Multi-Object Traffic-Scenarios

Amit Chaulwar and Michael Botsch
Technische Hochschule Ingolstadt, Ingolstadt, Germany
Email: {Amit.Chaulwar, Michael.Botsch}@thi.de

Torsten Krüger and Thomas Miehling
Automotive Safety Technologies GmbH, Ingolstadt, Germany
Email: {Torsten.Krueger, Thomas.Miehling}@astech-auto.de

***Abstract*—Active safety systems have a huge potential of increasing road traffic safety and are a necessary component of vehicles for autonomous driving. An algorithmic challenge of such systems is the prediction of collision-free trajectories in complex scenarios with multiple dynamic objects and curved roads. This work proposes an algorithm able to find safe trajectories in such an environment. The algorithm is dividing the prediction horizon into two intervals. In the first interval multiple collision-free trajectories are computed using an extension of the CL-RRT algorithm, introduced as Augmented CL-RRT, which contains a controller to consider the nonlinear dynamics of vehicles. The safety levels of the found trajectories are evaluated using the second prediction time interval, by taking into account the steering effort that is necessary to travel to a safe region on the road. The efficiency of the proposed algorithm is demonstrated exemplarily using a simulation with multiple dynamic objects in a traffic scenario with a curved road.**

***Index Terms*—trajectory planning, critical traffic scenarios, multiple dynamic objects**

I. Introduction

Active Safety Systems in vehicles like the Autonomous Emergency Braking [1], which perform autonomous interventions into the longitudinal dynamics are already on the market today. Future systems will be able to perform also autonomous interventions into the lateral dynamics, i. e., autonomous steering, in critical traffic scenarios [2]. This development will also be supported by the technological progress related to the field of autonomous driving. But it should be mentioned that autonomous intervention in the vehicle dynamics in the last few moments before a crash might occur are of high complexity, since the prediction algorithms must be able to model harsh maneuvers accurately. These interventions require the planning of safe trajectories in complex traffic scenarios with many static and dynamic objects like other vehicles, pedestrians, etc.

Trajectory planning algorithms use as input a representation of the driving environment provided by sensors like radar, video or laserscanner and have the task of finding a suitable trajectory in this environment that avoids a collision. Trajectory planning for collision avoidance can be decomposed into two tasks. The first can be named “find-goal” problem, which aims at defining a safe goal-location for the vehicle [3]. The second can be named “path-planning to a goal” problem, which has the aim to find a path from the current location to the goal-location. Algorithms for treating the first task are mainly based on expert knowledge and application specific. For the second task there exists a large number of algorithms which were mainly developed by the robotics community. Two challenges that must be addressed by trajectory-planning algorithms for traffic scenarios are the nonholonomic constraints of the vehicle motions models and the dynamic, non-deterministic vehicle environment. Deterministic trajectory planning algorithms such as A* [4] and its dynamic extensions D* [5] and D*-lite [6] cannot find a trajectory which follows the nonholonomic constraints of the vehicle. Consequently, a probabilistic approach called Rapidly-exploring Random Tree (RRT) algorithm [7] has gained a lot of popularity because of its fast runtimes and ability for planning the trajectory with nonholonomic constraints of the vehicle. The trajectory generation for nonholonomic vehicles is of great importance, especially if the goal is the development of a vehicle safety system. In [8] seventh order Bezier curves and in [9] third order B-spline basic functions are used to find smooth trajectories that do not violate the kinematic constraints of a vehicle. In [10] B-spline curves and the RRT algorithm are combined to find a collision free trajectory by taking into account interventions in the lateral and longitudinal dynamics. A restriction of this algorithm is that it can find the trajectory only among linearly moving obstacles.

Many variants of the RRT algorithm have been proposed. RRT^{x} [11] is a motion replanning algorithm for real-time navigation through a dynamic environment. After finding a shortest path for a specific configuration, the RRT^{x} algorithm replans the shortest path to a goal by continually repairing it as changes to the state space are detected. However, the algorithm needs an initial shortest

Manuscript received October 19, 2015; revised June 8, 2016.

doi: 10.18178/jtle.4.2.135-140

path and finding this path by taking into account the predicted motion of objects is not the focus of RRTx.

Closed-Loop RRT (CL-RRT) [12] runs a forward simulation using a vehicle model to compute the predicted trajectory whose feasibility is checked against vehicle and environmental constraints. It uses a drivability map which is regularly updated to check the validity of nodes and edges of the tree. The algorithm takes into account the vehicle dynamics by using a controller and the motion of objects in the vehicle environment. This algorithm has been successfully applied in the 2007 DARPA Urban Challenge. In [12] a complete stop is used as safe state at the end of a CL-RRT trajectory. In critical traffic situations with many dynamic obstacles this might not be a good choice since zero velocity does not necessarily imply a safe state.

This paper proposes an extension of the CL-RRT algorithm for finding safe trajectories in multi-object environments, while also considering nonlinear motion models for obstacle predictions and simultaneous interventions in the longitudinal and lateral dynamics of the vehicle. The algorithm also uses the state-time space concept of Time-Based-RRT (TB-RRT) [13], where a time coordinate is added to the space coordinates of the system and each node of the tree denotes a specific state in a specific time.

The paper is organized as follows: Section II formulates the problem of planning safe trajectories, Section III introduces the models and controllers that are used for vehicle motion planning in Section IV, where also the Augmented CL-RRT algorithm is introduced, Section V deals with choosing the safest trajectory from a set of computed trajectories, and Section VI presents simulation results. Throughout this work, vectors are denoted by lower case bold letters.

II. Problem Formulation

In this work the primary objective is to plan a collision-free and safe path for the EGO-vehicle [1] in critical traffic scenarios. Here, a “safe path” means that the chosen trajectory is collision-free for a longer time-interval than the one for which the trajectory is planned. A two-dimensional space is assumed for modelling a location-point in a traffic-scenario. The vehicle has nonlinear dynamics

$$\dot{\mathbf{s}}(t) = f(\mathbf{s}(t), \mathbf{u}(t)) \tag{1}$$

where $\mathbf{u}(t) \in \mathbb{R}^m$ is the control input at time t and $\mathbf{s}(t) \in \mathbb{R}^\ell$ denotes the set of points in $\mathbb{R}^2$ which represent the area that is occupied by the vehicle at time instance t. The initial state at $t = t_0$ is given by $\mathbf{s}(t_0)$. The path-planning problem implies the design of the control input $\mathbf{u}(t)$ over a finite prediction time-horizon of length τ_1, i. e. $t \in [t_0, t_0 + \tau_1]$. The set of constraints like bounds on the control input, static and dynamic obstacle avoidance or rules imposed by the road must be taken into account when computing $\mathbf{u}(t)$, so that the resulting $\mathbf{s}(t)$ is collision-free, i. e., $\mathbf{s}(t) \in S_{free}(t)$, where $S_{free}(t)$ expresses the area in $\mathbb{R}^2$ which is not occupied by other objects at prediction-time t.

[1] The vehicle in which the proposed path planning algorithm is running.

A prediction time-horizon of approximately $\tau_1 \approx 2$ s is a suitable value for performing avoidance maneuvers in most critical traffic scenarios. But the avoidance of a collision does not imply the chosen trajectory is safe since the chosen path may lead to another critical situation, e. g., the EGO-vehicle may avoid a collision by steering and changing the lane but it leads to a vehicle state that can hardly be controlled after the steering maneuver. That is why in this work a large prediction interval $[t_0, t_0 + \tau]$ is divided into the two intervals $[t_0, t_0 + \tau_1]$ and $[\tau_1, t_0 + \tau]$.

So, to avoid collisions by autonomous interventions, the problem in this work is stated as:

If a critical traffic situation is identified, find control inputs $\mathbf{u}_k(t)$ so that the resulting $\mathbf{s}_k(t)$ are collision-free, i. e., $\mathbf{s}_k(t) \in S_{free}(t)$ for $t \in [t_0, t_0 + \tau_1]$, and estimate the level of safety of the resulting trajectories in the interval $[\tau_1, t_0 + \tau]$. Choose that control input $\mathbf{u}_k(t)$, with the highest level of safety.

To find suitable control inputs $\mathbf{u}_k(t)$ an Augmented CL-RRT algorithm will be introduced in Section IV and the assessment of levels of safety is presented in Section V. The RRT algorithm is an incremental algorithm which can incorporate a vehicle dynamic model to generate realistic vehicle trajectories. Hence, a vehicle dynamics model is a base component of the trajectory planning task and will be presented in the next section.

III. Vehicle Model and Controller

Although simple kinematic vehicle dynamics models like the one-track model or the geometric model [14] are frequently used in driver assistance algorithms, in this work the EGO-vehicle is modelled with a two-track model to be able to describe the vehicle behavior accurately also for harsh de-escalation maneuvers, where values of the lateral accelerations exceed 5 m/s^2. The two track incorporates effects of individual tire forces and is described by three coupled differential equations, where the state variables are the velocity v, the sideslip angle β and yaw rate $\dot{\psi}$. The control inputs to the model are the four angles of the wheels with respect to the longitudinal axis of the vehicle $\delta_{fl}, \delta_{fr}, \delta_{rl}, \delta_{rr}$ and the four longitudinal slip values of the tires $s_{l,fl}, s_{l,fr}, s_{l,rl}, s_{l,rr}$, where the letters in the subscript stand for “front-left”, “front-right”, “rear-left”, and “rear-right”. Whereas the angles are determined by the steering wheel in a vehicle, the longitudinal slip values are determined by the acceleration or brake pedal. The angle of a wheel with respect to the longitudinal axis of the vehicle is responsible for the lateral force acting at this tire and the longitudinal slip value is responsible for the longitudinal force acting at the tire. Details about the two-track model can be found in [14]. Being able to influence the forces on the tires and thus the vehicle motion, these wheel angles and the slip values are incorporated in the input vector $\mathbf{u}(t)$ for the trajectory planning task.

$$\mathbf{u} = [\delta_{fl}, \delta_{fr}, \delta_{rl}, \delta_{rr}, s_{l,fl}, s_{l,fr}, s_{l,rl}, s_{l,rr}]^T \tag{2}$$

The inputs that are necessary to move the vehicle from its current position $\mathbf{s}(t_0)$ to a safe region are determined using a *Model Predictive Control* (MPC) approach in this work. This is achieved by computing the input $\mathbf{u}(t_0)$ based on an anticipation of future events up to the time $t_0 + \tau$. The anticipation is realized using motion models for the prediction of all dynamic objects in the time interval $[t_0, t_0 + \tau]$ and a geometric description of the road and other stationary objects. For the first part of this prediction horizon, i. e., $[t_0, t_0 + \tau_1]$ the Augmented CL-RRT algorithm is searching for a collision-free path and a lower-level controller is used to move the EGO-vehicle along the path in this prediction time, thereby generating the inputs $\mathbf{u}(t)$ up to time instance $t_0 + \tau_1$. However, from these values that lead to a collision-free path, only $\mathbf{u}(t_0)$ is finally used for the MPC-control of the vehicle. Subsequently, the input $\mathbf{u}(t_0 + \Delta t)$ is determined based on an anticipation of future events up to the time $t_0 + \Delta t + \tau$, etc.

The lower-level controller that is used as part of the MPC approach to determine the inputs $\delta_{fl}, \delta_{fr}, \delta_{rl}, \delta_{rr}$ in $\mathbf{u}(t)$ for the prediction horizon is presented in the following. The inputs $s_{l,fl}, s_{l,fr}, s_{l,rl}, s_{l,rr}$ are given by acceleration profiles as will be presented in Section IV. The goal of the lateral dynamics controller is to set the angles $\delta_{fl}, \delta_{fr}, \delta_{rl}, \delta_{rr}$ so that the vehicle will follow a reference trajectory, which is computed by the Augmented CL-RRT algorithm. Fig. 1 visualizes the location of the controller in the signal flow.

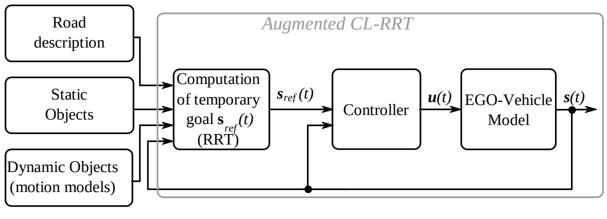

Figure 1. The controller is computing elements of $\mathbf{u}(t)$

The optimization problem of the controller is to minimize the cost function J_{lat} which is defined as

$$J_{lat}(\delta_{fl}, \delta_{fr}, \delta_{rl}, \delta_{rr}) = \sum_{t=t_0}^{t_c+\tau_1} (\mathbf{s}(t) - \mathbf{s}_{ref}(t))^2 \tag{3}$$

where $\mathbf{s}(t)$ and $\mathbf{s}_{ref}(t)$ are the actual and reference position of the EGO-vehicle. For the implementation used in this work $\mathbf{s}(t)$ is uniquely defined by the coordinates $[X, Y]$ of the center of gravity and the yaw angle ψ of the EGO-vehicle. Thus, the steering angle is adapted in the controller in such a way that the deviation between the actual location of the center of gravity and its desired location are minimized as well as the deviation between the actual yaw angle and the desired yaw angle. The desired values are computed by the Augmented CL-RRT algorithm, which will be introduced in the next section.

IV. Dynamic Motion Planning

After a brief review of the basic RRT algorithm, this section introduces the components that are used in order to realize the motion planning for safe trajectories in dynamic multi-object traffic scenarios.

A. *Rapidly-Exploring Random Tree*

The basic RRT algorithm was introduced in [15] for trajectory planning to reach a goal state from an initial state under a differential constraint. In an iterative process the algorithm samples a random point $\mathbf{s}_{rand}$ usually with some bias towards a goal S_{goal} and extends the tree by incremental motion towards $\mathbf{s}_{rand}$ from the closest $\mathbf{s}_k$ that is stored in the tree using differential constraints like in (1)

$$\mathbf{s}_{k+1} = \mathbf{s}_k + f(\mathbf{s}_k, \mathbf{u}_k)\Delta t \tag{4}$$

where Δt is the time interval for which the tree is grown. The new state $\mathbf{s}_{k+1}$ is added to the tree if $\mathbf{s}_{k+1} \in S_{free}$. The algorithm is terminated as soon as the tree contains a state in the S_{goal} region.

B. *Identification of a Critical Traffic Scenario*

As mentioned in the problem formulation in Section II, the path planning for safe trajectories is activated only if a critical traffic situation is identified. For this task the *Time-To-Collision* (*TTC*) criticality criteria is used in this work [16]. If a collision of the EGO-vehicle is detected with any other obstacle within a *TTC* of 2 s in future, the situation is said to be critical. Obstacles include static objects and dynamic objects like other vehicles, pedestrians, etc. It is assumed that the EGO-vehicle is equipped with sensors which give the information about the current state of other obstacles like their position, velocity, acceleration, yaw angle, etc. and also about the physical structure of the environment like road width, curvature, etc. Further, while predicting the trajectory for obstacles it is assumed that vehicles tend to follow their track with constant velocity and pedestrian travel linearly with constant velocity. Predictions with these assumptions generate the future positions $\mathbf{s}_{Obj,n}(t)$ for all N_{Obj} obstacles in the vicinity of the EGO-vehicle for the next τ seconds, i. e., $n = 1, ..., N_{Obj}$

and $t \in [t_0, t_0 + \tau]$. A collision between the EGO-vehicle and the $n-th$ object at time t occurs if the indicator function

$$I_c(\mathbf{s}(t), \mathbf{s}_{Obj,n}(t)) = \begin{cases} 1, & if \quad \mathbf{s}(t) \cap \mathbf{s}_{Obj,n}(t) \neq \varnothing, \\ 0, & otherwise \end{cases} \tag{5}$$

has the value 1. The difference between the first time instance t when a collision is identified and the current time instance t_0 is the TTC.

C. Full Braking

If a critical situation is identified, i. e., $TTC < 2$ s, the first step is to check if a collision can be avoided just by full breaking without any lateral intervention. If it is possible, a full braking maneuver is performed without using the Augmented CL-RRT algorithm and the resulting braking trajectory is considered a safe trajectory. This type of safe trajectory represents the Autonomous Emergency Braking that is already implemented in modern vehicles.

D. Augmented CL-RRT Algorithm

Some extensions to the RRT algorithm are introduced in this work in order to generate safe trajectories. They are presented in the following.

1) State –Time space

Each node in the RRT algorithm is not just storing the position $[X, Y]$ of the EGO center of gravity but also its yaw angle ψ, the velocity v, the steering angle δ from the two-track-model as well as the time t when the node would be reached. This is necessary since in a dynamic environment the free space $S_{free}(t)$ is also time-dependent and a representation of the predicted traffic situation must be available for all nodes in order to grow the tree. The computation of the time dependent $S_{free}(t)$ is presented next.

2) Multiple hypothesis prediction

Although it is assumed that vehicles tend to follow a road and pedestrians travel linearly with constant velocity as mentioned in IV.B for the computation of $\mathbf{s}_{Obj,n}(t)$, many possible hypothesis can be realized by introducing uncertainty in the longitudinal acceleration and lateral position of dynamic obstacles. In order to model $S_{free}(t)$, the possible location of the $n-th$ dynamic obstacle is not represented only by $\mathbf{s}_{Obj,n}(t)$ but by $\tilde{\mathbf{s}}_{Obj,n}(t)$ in the form of an ellipse that grows with increasing prediction time. The empirical rule used to grow the ellipses is to expand their axes according to the acceleration values that might occur in lateral and longitudinal direction, i. e., approx. ± 4 m/s^2. The applied rule also takes into account that obstacles move on the road, i. e., the minor axis is increased only till half of the road width. For static obstacles $\tilde{\mathbf{s}}_{Obj,n}(t) = \mathbf{s}_{Obj,n}(t) = \mathbf{s}_{Obj,n}$. Denoting the area in $\mathbb{R}^2$ that is covered by the road with S_{Road}, the free space $S_{free}(t)$ at time instance t is

$$S_{free}(t) = S_{Road} \setminus \bigcup_{n=1}^{N_{Obj}} \tilde{\mathbf{s}}_{Obj,n}(t). \tag{6}$$

3) Goal selection

As mentioned in Section I, the “find-goal” task is application specific. For the problem in this work the region $S_{goal} \subset S_{Road}$ that is used by the Augmented CL-RRT algorithm is defined by the point where the center of gravity of the EGO-vehicle will lie at time instance $t_o + \tau$ according to the prediction with the two-track-model assuming that the current velocity is constant in the prediction interval. S_{goal} is the area around this point that lies on the road and which is included in $S_{free}(t_o + \tau)$.

4) Sampling strategy

The samples $\mathbf{s}_{ref}(t)$ are randomly sampled with uniform distribution in region of the road between $\mathbf{s}(t_0)$ and S_{goal}. In case of roads having multiple tracks along the same direction the sampling is performed only in the region containing the tracks in the same direction in which the EGO-vehicle is driving, otherwise on the whole road.

5) Longitudinal acceleration profiles

To take into account not only the steering capabilities of the EGO-vehicle, in addition to the angles $\delta_{fl}, \delta_{fr}, \delta_{rl}, \delta_{rr}$ that are computed by the controller aiming to reach the sampled locations $\mathbf{s}_{ref}(t)$, also predefined acceleration profiles, which define the remaining inputs $s_{l,fl}, s_{l,fr}, s_{l,rl}, s_{l,rr}$ of $\mathbf{u}(t)$ are integrated in the Augmented CL-RRT algorithm. Using M_{acc} profiles, e. g., $M_{acc} = 3$ for “no braking”, “mean acceleration”, and “strong braking”, leads to maximally M_{acc} collision-free trajectories for the interval $[t_o, t_o + \tau_1]$. So, each resulting collision-free trajectory is the result of interventions $\mathbf{u}(t)$ in the lateral and longitudinal dynamics.

V. Safe Trajectory Criteria

The Augmented CL-RRT algorithm finds multiple trajectories (up to M_{acc}) with different acceleration profiles for the prediction time interval $[t_o, t_o + \tau_1]$. As mentioned in Section II an evasion-trajectory is considered safe if it leads to a vehicle state from where the vehicle can easily be controlled to follow the road. In order to estimate this level of safety for each of the trajectories, a further prediction for the time interval $[\tau_1, t_0 + \tau]$ is performed. To do so, for each of the initial trajectories a new goal $\tilde{S}_{goal,m_{acc}} \subset S_{free}(t_0 + \tau)$, $m_{acc} = 1 \ldots M_{acc}$ is defined. Similarly as presented above for the goal selection of S_{goal}, each goal $\tilde{S}_{goal,m_{acc}}$ is computed by predicting the area on the road where the EGO-vehicle should be at time instance $t_0+\tau$ starting from its current location $\mathbf{s}_{m_{acc}}(\tau_1)$ and assuming that the velocity at τ_1 remains constant. Then the controller for the angles $\delta_{fl}, \delta_{fr}, \delta_{rl}, \delta_{rr}$ form Section III is used again to move the vehicle on the road from $\mathbf{s}_{m_{acc}}(\tau_1)$ to $\tilde{S}_{goal,m_{acc}}$ and the corresponding steering angle values are computed. The level of safety for the $m_{acc}-th$ trajectory is given by the maximum steering angle input $\delta_{m_{acc}}$ on its collision-free path from $\mathbf{s}_{m_{acc}}(\tau_1)$ to $\tilde{S}_{goal,m_{acc}}$. Small values of $\delta_{m_{acc}}$ correspond to trajectories with a high safety level since the maneuver required to reach the goal are soft, whereas high values of $\delta_{m_{acc}}$ correspond to trajectories with a low safety level. Thus, the safest trajectory for the problem formulated in Section II is the one indexed with

$$m_{safe} = \min_{m_{acc}}(\delta_1, ..., \delta_{M_{acc}}) \tag{7}$$

VI. Simulation Results

A Matlab-based simulator is developed and used to design traffic scenarios and to evaluate the presented algorithm. It produces 2D environments with EGO and non-EGO-vehicles, pedestrians, roads, and stationary objects. The geometrical description of the environment and obstacle parameters like position, speed, etc. which are available in the simulator are used as input data for the algorithm.

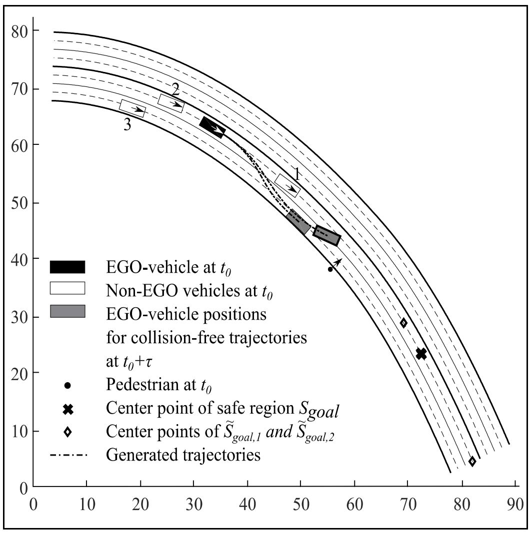

Figure 2. Scenario 1

Exemlarily Fig. 2 and Fig. 3 show two similar traffic scenarios at the time instant t_o when a collision is predicted for the EGO-vehicle with vehicle 1 which has braked sharply because of a pedestrian crossing the street. The direction of travel of all vehicles and pedestrian is shown by arrows and the non-EGO-vehicles are numbered. EGO-vehicle, vehicle 2 and vehicle 3 are traveling with same speed of $v = 50$ km/h. The only difference between two scenarios is the initial position of the vehicle 3. The Augmented CL-RRT algorithm used here finds safe trajectories for a strong deceleration, a positive acceleration and a constant velocity profile. The best trajectory is then selected according to (7). The end state of the best trajectory is highlighted with thick boundary.

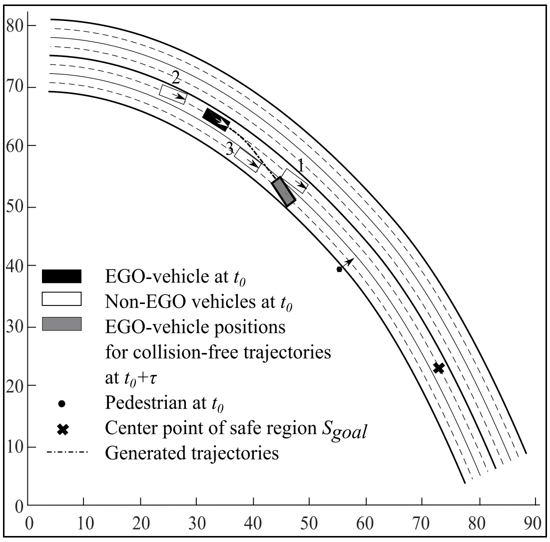

Figure 3. Scenario 2

In both scenarios, the EGO-vehicle cannot avoid the collision just by breaking because it will lead to a collision with vehicle 2 which is traveling just behind the EGO-vehicle in the same lane. Hence, a safe trajectory needs to be computed by the Augmented CL-RRT taking into account vehicle 1, vehicle 2 vehicle 3 and the pedestrian. In both scenarios 3 longitudinal acceleration profiles are used, i. e., $M_{acc} = 3$, implementing maneuvers for "no braking", "accelerate" (+4 m/s^2), and "brake" (−8 m/s^2).

In scenario 1, the Augmented CL-RRT algorithm is unable to find a collision-free trajectory for "brake" as moving to the parallel track with deceleration leads to a collision with vehicle 3. But the algorithm is able to find the trajectories for the "no braking" and "accelerate" profiles. They are shown in Fig. 2. The trajectory with the "accelerate" profile is finally chosen as the safe trajectory because it requires less steering effort to travel towards its updated goal.

In scenario 2, the Augmented CL-RRT algorithm is just able to find one collision-free trajectory, the one for the "brake" profile as moving to the parallel track with constant velocity or with acceleration will lead to a collision with vehicle 3. So, in this scenario the maneuver leading to a collision-free trajectory implies a hard intervention into the longitudinal and lateral dynamics of the EGO-vehicle.

VII. Conclusion

This work is introducing an algorithm for finding safe trajectories in critical dynamic multi-object traffic scenarios. To do so, the prediction horizon is divided into two intervals. In the first interval trajectories are computed based on extensions of the CL-RRT and a lower-level controller that takes into account the non-linear dynamic model of the EGO-vehicle, leading to collision-free paths. The extensions of the CL-RRT are introduced in this work and the resulting algorithm is named Augmented CL-RRT. In the second interval an evaluation of the found trajectories regarding a level of safety is used to choose the safest trajectory. Two examples with multiple dynamic objects in a curved-road scenario are presented to demonstrate the efficiency of the proposed algorithm.

ACKNOWLEDGMENT

This research work is funded by the German Federal Ministry of Education and Research (BMBF) under grant number 03FH034PX4.

REFERENCES

[1] N. Kaempchen, B. Schiele, and K. Dietmayer, "Situation assessment of an autonomous emergency brake for arbitrary vehicle-to-vehicle collision scenarios," *IEEE Trans. Intell. Transp. Syst.*, vol. 4, pp. 678-687, 2009.

[2] A. Eidehall, J. Pohl, F. Gustafsson, and J. Ekmark, "Toward autonomous collision avoidance by steering," *IEEE Trans. Intell. Transp. Syst.*, vol. 8, no. 1, pp. 84–94, 2007.

[3] P. Varaiya, "Smart cars on smart roads: Problems of contro," *IEEE Trans. on Automatic Control*, vol. 38, no. 2, pp. 195-207, February 1993.

[4] P. Hart, N. Nilsson, and B. Raphael, "A formal basis for the heuristic determination of minimum cost paths," *IEEE Trans. on System Science and Cybernetics*, 1968.

[5] A. Stentz, "Optimal and efficient path planning for partially-known environments," in *Proc. IEEE Int. Conf. on Robotics and Automation*, 1994.

[6] S. Koenig and M. Likhachev, "Fast replanning for navigation in unknown terrain," *Trans. on Robotics*, vol. 21, no. 3, pp. 354–363, 2005.

[7] J. Kuffner and S. LaValle, "Rrt-connect: An efficient approach to single-query path planning," in *Proc. IEEE Int. Conf. on Robotics and Automation*, 2000.

[8] A. Neto, D. Macharet, and M. Campos, "Feasible RRT-based path planning using seventh order Bezier curves," in *Proc. IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, 2010.

[9] E. Shan, B. Dai, J. Song, and Z. Sun, "A dynamic RRT path planning algorithm based on B-Spline," in *Proc. Second Int. Symp. on Computational Intelligence and Design*, 2009.

[10] K. Macek, M. Becked, and R. Siegwart, "Motion planning for car-like vehicles in dynamic urban scenarios," in *Proc. IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, 2006.

[11] Michael Otte and Emilio Frazzoli, "RRT-X: Real-Time motion planning/replanning for environments with unpredictable obstacles," in *Proc. Int. Workshop on the Algorithmic Foundations of Robotics*, Istanbul, Turkey, 2014.

[12] Y. Kuwata, J. Teo, G. Fiore, S. Karaman, E. Frazzoli, and J. How, "Real-Time motion planning with applications to autonomous urban driving," *IEEE Trans. on Control Systems Tech.*, vol. 17, no. 5, 2009.

[13] A. Sintov and A. Shapiro, "Time-Based RRT algorithm for rendezvous planning of two dynamic systems," in *Proc. IEEE Int. Conf. on Robotics & Automation*, 2014.

[14] R. Jazar, *Vehicle Dynamics Theory and Application*, 2nd ed, Springer, 2014, ch. 10.

[15] S. M. LavValle, "Rapidly-exploring random trees: A new tool for path planning," Technical Report, Computer Science Department Iowa State University, 1998.

[16] S. Lefevre, D. Vasquez, and C. Laugier, "A survey on motion prediction and risk assessment for intelligent vehicles," *ROBOMECH Journal*, 2014.

Amit Chaulwar received the B.Tech. degree in mechanical engineering from the College of Engineering, Pune (COEP), India, in 2009, and the M.Eng. degree in automotive engineering from Technische Hochschule Ingolstadt, Ingolstadt, Germany, in 2015.
He is working as a Research Assistant at Technische Hochschule Ingolstadt, since 2014. Prior starting his M.Eng. degree, he was working as System Engineer in Tata Consultancy Services, Pune, India. His research interests include path planning and machine learning techniques for collision avoidance in road traffic.

Michael Botsch received the diploma and doctoral degrees in electrical engineering, both with honors, from Technische Universität München, München, Germany, in 2005 and 2009.
He worked for five years in the automotive industry as Development Engineer at Audi AG in the field of active safety systems. In October 2013 he was appointed Professor for Vehicle Safety and Signal Processing at Technische Hochschule Ingolstadt in the Department of Electrical Engineering and Computer Science. He is the Associate Scientific Director of the vehicle safety research center CARISSMA at Technische Hochschule Ingolstadt. His research interests are in signal processing and automotive applications.
Prof. Botsch is a member of IEEE and VDE.

Torsten Krüger received the Dipl-Ing. degree in computational engineering from the department of Computer Science and Automation, Technical University of Ilmenau, Ilmenau, Germany, in 2009.
Since then he has been with Automotive Safety Technologies GmbH, Ingolstadt, Germany, where he is currently the Team Manager Advanced Development for integral safety systems. His research interests cover vehicle dynamics control and method development for functional engineering of integral safety systems.

Thomas Miehling received the diploma degree in computer science (Diplom Informatiker) from the Fachhochschule München, München, Germany, in 1997.
He started his carrier as embedded SW developer and SW project leader in TEMIC (Conti TEMIC) in Ingolstadt, Germany. In 2002, he started at Audi AG, Ingolstadt, Germany to develop embedded standard SW. Since 2008 he was at Audi Electronics Venture (AEV) the project leader for integral safety systems. In 2013 he became the department leader integral safety at the Automotive Safety Technologies, an AEV subsidiary.