Proceedings of the 2006 IEEE/RSJ
International Conference on Intelligent Robots and Systems
October 9 - 15, 2006, Beijing, China

Motion Planning for Car-Like Vehicles in Dynamic Urban Scenarios

Kristijan Maček*, Marcelo Becker†, Roland Siegwart*

*Swiss Federal Institute of Technology Zurich, Switzerland
email: {kristijan.macek, roland.siegwart}@mavt.ethz.ch

†Pontifical Catholic University of Minas Gerais, Brazil
email: marcelo.becker@epfl.ch

Abstract— **Abstract - This paper focuses on development of a motion planning strategy for car-like vehicles in dynamic urban-like scenarios. The strategy can be summarized as a search for a collision-free trajectory among linearly moving obstacles applying Rapidly-exploring Random Trees (RRT) and B-splines. Collision avoidance is based on geometric search in transformed state space of chained form kinematic model decomposition. The time criterion for avoiding obstacles is based on relative robot to obstacle motion and is checked iteratively for possible collisions within the RRT exploration phase. The line segment geometric path is interpolated with a B-spline curve in order to generate a feasible trajectory that takes into account non-holonomic constraints. The exploration strategy aims at finding an optimal steering and longitudinal control of the vehicle in minimum time and steering activity sense. In order to test the strategy a MatLab based simulator was developed. This simulator reproduces a simple 2D urban-like environment with parked and moving cars, buses, trucks, people, buildings, streets, and trees. The test vehicle, a modified Smart Car equipped with several sensors was kinematically modeled. The sensor data are extracted from the environment based on its geometrical description and used as input data for the motion planning strategy which was verified in a dynamic urban scenario simulation.**

I. Introduction

The improvement of general traffic safety by applying intelligent systems on-board passenger and commercial vehicles is a challenge in many ways. Since the 1990s many research groups around the world are working to expand the use of technologies and algorithms developed in the field of mobile robotics into passenger and commercial vehicles. Disabled people, the elderly and those who deal with a very hectic life would welcome a vehicle whose design is grounded on the concept of full autonomous behavior. There is also the case of those who would appreciate an intelligent system able to provide them intelligent driving assistance. This could be translated into a vehicle provided with various on-board sensors that are used to detect environment features (e.g. lanes, trees, guard-rails, people, other vehicles, etc.) where some interference would occur in peril of accidents. A fully autonomous behavior may allow the driver/passenger to execute other more relevant or relaxing tasks during the travel.

As the urban city scenario is inherently a dynamic environment, motion of moving obstacles has to be explicitly included into the planning state. The motion planning problem for nonholonomic platforms, which include passenger and commercial vehicles, in dynamic environments has seen its first approaches in the configuration-time space of [1], the hierarchical static and velocity planning in [2] and the state-time approach of [3]. The constant velocity obstacles were analyzed later in [4] and extended to arbitrarily moving ones in [5]. A randomized kinodynamic motion planner was used in [6], whereas a behavior dynamics based planning was recently introduced in [7]. The concept of inevitable collision states was first introduced in the [8] and elaborated further in [9].

In [10] a closed-form solution for motion planning among linearly moving obstacles was proposed based on polynomial trajectories. Based on the state transformation and the collision avoidance criterion of [10], a solution for the obstacle-free smooth trajectory was introduced in [11] where the aim was to improve the local refinement and controllability of curve generation by using B-splines as trajectories, providing continuous kinematic level controls and motion. The search for feasible trajectories was performed by geometric Random-Rapidly Exploring Trees (RRT) expansion [12]. In order to solve the collision check for all the obstacles for future steps in a single sweep, both [10] and [11] rely on the constraint that the motion of the vehicle along the transformed state axes which is controlled explicitly, is constant. This renders the techniques essentially as steering methods, although the longitudinal velocity is varying as a consequence of the nonlinear state transform. To obtain full controllability of the vehicle, where the desired longitudinal vehicle velocity at each instant is searched for in the planning phase, this paper proposes an extension of [11] such that the motion (velocity) along the explicitly controlled state axes is a trapezoidal profile one according to the acceleration limits of the vehicle. The duration of acceleration/deceleration phase of the desired velocity profile is locally randomly explored by RRT search expansion, therefore the RRT leafs are randomized both in orientation and length. This implies that the collision criterion for obstacle avoidance must be incrementally solved along the single RRT tree leafs, i.e. path increments for all obstacles. The advantage of the newly proposed motion planning solution is that the vehicle velocity profile can be search for in the minimum-time sense with the natural acceleration/breaking phases occurring

1-4244-0259-X/06/$20.00 ©2006 IEEE

automatically according to the local obstacle configuration along the path. Thus, the vehicle travels at maximum allowed speed in free space, or slows down in confined areas while having the full steer capability. Additionally, since the chained form transformed space is used for the RRT search defining the functional dependency of the states a-priori, a purely geometric search can be used that includes no state numerical integration for collision check of the vehicle trajectory with obstacles.

Sec.II gives a short overview on standard car-like kinematic modeling and chained form state-space transformation. Sec. III recapitulates the collision avoidance criterion for moving obstacles presented first in [10]. In Sec. IV it is explained how the vehicle trajectory is generated with the RRT expansion and B-splines for full vehicle control. The urban scenario simulation is presented in Sec. V followed by conclusions in Sec. VI.

II. Kinematic modeling and chained form state-space transformation

Car-like kinematic model can be described by configuration coordinates $q = [x\, y\, \theta\, \phi]$, where the Cartesian coordinates (x, y) correspond to the midpoint between front and rear axle centers which are a separated by a length l, vehicle orientation θ and ϕ being the steering angle. Denoting ρ the wheel radius, u_1 the angular velocity of the rear driving wheels, u_2 the steering rate of the front wheels, then the kinematic car-like model is:

$$\begin{pmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \\ \dot{\phi} \end{pmatrix} = \begin{pmatrix} \rho\cos(\theta) - \frac{\rho}{2}\tan(\phi)\sin(\theta) & 0 \\ \rho\sin(\theta) + \frac{\rho}{2}\tan(\phi)\cos(\theta) & 0 \\ \frac{\rho}{l}\tan(\phi) & 0 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} u_1 \\ u_2 \end{pmatrix}. \tag{1}$$

According to [13] and [10] the kinematic model of Eq. 1 can be transformed into the canonical chained form where derivatives of transformed states depend only on lower states in the chain and the inputs:

$$\begin{aligned} \dot{z}_1 &= v_{c1}\,, & \dot{z}_3 &= z_2 v_{c1}\,, \\ \dot{z}_2 &= v_{c2}\,, & \dot{z}_4 &= z_3 v_{c1}\,, \end{aligned} \tag{2}$$

where the configuration coordinates $q = [x\, y\, \theta\, \phi]$ and inputs u_1, u_2 are transformed to new state space of $z = [z_1, z_2, z_3, z_4]$ and inputs v_{c1}, v_{c2}:

$$\begin{aligned} &z_1 = x - \tfrac{l}{2}\cos(\theta)\,, \quad && z_2 = \tfrac{\tan(\phi)}{l\cos^3(\theta)}\,, \\ &z_3 = \tan(\theta)\,, && z_4 = y - \tfrac{l}{2}\sin(\theta)\,, \\ &u_1 = \tfrac{v_{c1}}{\rho\cos(\theta)}\,, \\ &u_2 = -\tfrac{3\sin(\theta)}{l\cos^2(\theta)} sin^2(\phi) v_{c1} + l\cos^3(\theta)\cos^2(\phi) v_{c2}\,. \end{aligned} \tag{3}$$

The state and control transformation of Eq. 3 is valid for angles $\theta \neq \pm\pi/2$. This singularity is not a hard limitation, since the $x-y$ plane can be pre-rotated for guaranteeing $\theta(t) \in (-\pi/2\,, \pi/2)$ or by introducing an intermediate goal position.

The transformed states z_2 and z_3 can be further expressed as:

$$z_3 = \tfrac{\mathrm{d}z_4}{\mathrm{d}z_1}\,, \qquad z_2 = \tfrac{\mathrm{d}^2 z_4}{\mathrm{d}z_1^2}\,. \tag{4}$$

III. Collision avoidance criterion for moving obstacles

The criterion for collision avoidance between the vehicle and a single moving obstacle is adopted from [10] and used afterwards in [11]. For clarity, the collision criterion is exposed here as follows:

- The physical limits of the vehicle are denoted by a 2D circle of radius R at the center $O(t) = (x(t)\,, y(t))$. The choice of the reference point being on the midpoint of the connecting line between the axes renders the R size less conservative. The translational vehicle velocity to be determined is $v_r = [\dot{x}(t)\, \dot{y}(t)]^\top$.
- A set of $i = 1, \ldots, n_o$ objects is represented by circles at points $O_i(t)$ and radius r_i denoted by $B_i\,(O_i(t)\,, r_i)$ and moving at linear velocities $v_i(t)$.

The state of the environment in terms of obstacle positions and directions of movement is resampled at a rate T_s. Starting at sample instant k, the *i*th obstacle's description in the interval $t \in [kT_s\,, (k+1)T_s]$ is assumed to be evolving linearly from position $O_i = \left(x_i^k\,, y_i^k\right)$ with velocity $v_i^k = \left[v_{i,x}^k\, v_{i,y}^k\right]^\top$.

The relative vehicle velocity to the *i*th obstacle is defined as:

$$v_{r,i}^k = v_r - v_i^k = \begin{bmatrix} v_{r,i,x}^k \\ v_{r,i,y}^k \end{bmatrix} = \begin{bmatrix} \dot{x} - v_{i,x}^k \\ \dot{y} - v_{i,y}^k \end{bmatrix}. \tag{5}$$

Taking into account the physical sizes of the vehicle and the *i*th obstacle, the avoidance criterion to a single obstacle can be formulated as:

$$\left(x_i' - x_i^k\right)^2 + \left(y_i' - y_i^k\right)^2 \geq (r_i + R)^2\,, \tag{6}$$

where $x_i' = x - v_{i,x}^k \tau$ and $y_i' = y - v_{i,y}^k \tau$, with τ being $\tau = t - kT_s$.

The avoidance relation according to Eq. 6 is based on relative vehicle motion to a virtually static *i*th obstacle. The minimum distance between the two is a radius of $(r_i + R)$ whenever the obstacle's x_i^k coordinate is in the following interval in the $x - y$ plane:

$$x_i^k \in [x_i' - r_i - R, x_i' + r_i + R]\,. \tag{7}$$

In order to obtain the avoidance criterion for the transformed state space, the relations from Eq. 3 combining variable x with z_1 and y with z_4 are used. Therefore, the avoidance criterion of Eq. 6 can be described in the transformed plane $z_4 - z_1$ as:

$$\begin{aligned} &\left(z_1 + \frac{l}{2}\cos\theta - v_{i,x}^k\tau - x_i^k\right)^2 + \\ &+ \left(z_4 + \frac{l}{2}\sin\theta - v_{i,y}^k\tau - y_i^k\right)^2 \geq (r_i + R)^2\,, \end{aligned} \tag{8}$$

or conversely in the relative motion plane $z_{4,i}' - z_{1,i}'$ as:

$$\begin{aligned} &\left(z_{1,i}' + \frac{l}{2}\cos\theta - x_i^k\right)^2 + \left(z_{4,i}' + \frac{l}{2}\sin\theta - y_i^k\right)^2 \\ &\qquad \geq (r_i + R)^2\,, \end{aligned} \tag{9}$$

where:

$$z_{1,i}' = z_1 - v_{i,x}^k\tau\,,\ z_{4,i}' = z_4 - v_{i,y}^k\tau\,. \tag{10}$$

The criterion of Eq. 9 represents a snapshot at time t (or τ) and sample interval k of the relative position of the vehicle with transformed coordinates $\left(z'_{1,i}\,, z'_{4,i}\right)$ to a virtually static obstacle at position $O_i^k = \left(x_i^k\,, y_i^k\right)$ in the interval:

$$x_i^k \in \left[z'_{1,i} + \frac{l}{2}\cos\theta - r_i - R\,, z'_{1,i} + \frac{l}{2}\cos\theta + r_i + R\right] \tag{11}$$

This criterion includes dependence on the current vehicle orientation θ which can be undesirable if the orientation of the vehicle in the planning stage is not known beforehand. By further geometric inspection an orientation independent criterion was developed in [10] only in terms of relative transformed variables $z'_{1,i}$, $z'_{4,i}$, vehicle and obstacles sizes.

Firstly, as the relative motion variables x'_i and y'_i can be expressed with the relative transformed variables $z'_{1,i}$, $z'_{4,i}$ as:

$$x'_i = z'_{1,i} + \tfrac{l}{2}\cos(\theta)\,, \quad y'_i = z'_{4,i} + \tfrac{l}{2}\sin(\theta) \tag{12}$$

and the fact that the vehicle orientation angle θ can only take values $\theta \in (-\pi/2\,, \pi/2)$, then it follows that the points $(x'_i\,, y'_i)$ can only be located at the right semicircle located at $\left(z'_{1,i}\,, z'_{4,i}\right)$ with radius $\frac{l}{2}$ in the $z'_{4,i} - z'_{1,i}$ plane.

Furthermore, the Eq. 6 gives the condition on minimum distance of $(r_i + R)$ between the possible relative vehicle point $(x'_i\,, y'_i)$ and the *i*th obstacle's virtual static position $\left(x_i^k\,, y_i^k\right)$. Therefore, by drawing circles of radius $(r_i + R)$ at all possible $(x'_i\,, y'_i)$ loci, a circular prohibitive region Υ_i for i*th* obstacle center $O_i^k = \left(x_i^k\,, y_i^k\right)$ is defined and is of radius $\tilde{R} = \left(r_i + R + \frac{l}{2}\right)$, giving the final vehicle to single *i*th obstacle collision avoidance criterion:

$$\left(z'_{1,i} - x_i^k\right)^2 + \left(z'_{4,i} - y_i^k\right)^2 \geq \left(r_i + R + \frac{l}{2}\right)^2 \tag{13}$$

with no vehicle orientation θ dependence. Because of the limited θ range, the potential collision region for the $z'_{1,i}$ axis is defined as:

$$x_i^k \in \left[z'_{1,i} - r_i - R\,, z'_{1,i} + \frac{l}{2} + r_i + R\right]. \tag{14}$$

A criterion for the $z'_{4,i}$ axis that considers any possible orientation θ, results in a region Ξ_i for an *i*-th obstacle:

$$y_i^k \in \left[z'_{4,i} - \frac{l}{2} - r_i - R\,, z'_{4,i} + \frac{l}{2} + r_i + R\right]. \tag{15}$$

IV. Vehicle trajectory generation

Using the chained form transformation of Eq.2 for the vehicle kinematic model allows to describe the variable z_4 in a direct functional relation to z_1. The family of curves that are used here are B-splines[14]:

$$z_4(t) = b(z_1(t)) = \sum_{j=0}^{m} B_j N_{j,d}(z_1(t))\,, \tag{16}$$

where B_j are spline control points, N_j^d denote the basis functions of degree d defined by a knot sequence $z_{1_0} \leq z_{1_1} \leq \cdots \leq z_{1_{m+d+1}}$. The variable z_1 represents the spline parameter that can take any real value in the range $\left[z_{1_0}, z_{1_{m+d+1}}\right]$. The B-splines are used to their local controllability/flexibility which may be important in confined environments, so problems like large detours can be avoided. Additionally, they enable a tight interpolation of the linear-segment based RRT search tree that is the base for control points generation.

The motion planning problem can be stated as problem of moving the vehicle from an initial configuration $q_0 = \left[x_0, y_0, \theta_0, \phi_0, \dot{\phi}_0\right]$ to a final configuration $q_f = \left[x_f, y_f, \theta_f, \phi_f, \dot{\phi}_f\right]$ while avoiding the linearly moving obstacles $B_i\,(O_i(t)\,, r_i)$, $i = 1, \ldots, n_o$, as described in Sec. III. The configuration boundary conditions define the set of transformed state boundary conditions $\left[z_1^k, \dot{z}_2^k, z_2^k, z_3^k, z_4^k\right]$, $k = 0, \ldots, N_s$ global replanning sample intervals.

The task of finding the vehicle trajectory is divided into three phases:

- defining potential collision areas with obstacles;
- performing a RRT search expansion within the obstacle free regions of the transformed space;
- smoothing of the linear segment based path with B-splines to obtain the final smooth trajectory.

The search for a feasible path with RRT expansion in the transformed state space spanned by $z_4 - z_1$ plane can be considered as an analogy to a $y - x$ plane search. A tree leaf of the RRT is a line segment that represents a path increment, where p, q are connected tree nodes indices. Due to dynamic obstacles, the collision regions in $z_4 - z_1$ plane where the tree leafs cannot be expanded, depend directly on the vehicle motion and therefore control inputs beforehand, so that this space-time combination results in a $3D$ search problem.

Nevertheless, the evolution of the state z_1 can be analyzed independently from the rest of the vehicle configuration, due to chained transform (Eq. 2) where $\dot{z}_1$ is an input control variable. If the evolution of $z_1(t)$ is known by some defined profile, the time component is embedded in z_1 and a pure $2D$ geometric search can be performed with moving obstacle traces in $z_4 - z_1$ plane related to their respective velocities and z_1 component.

A. Potential collision areas with obstacles

Let the RRT tree in the replanning phase $t \in [kT_s\,, (k+1)T_s]$ be $\mathcal{T}^k$ and let tree nodes $\mathcal{T}_p^k$, $\mathcal{T}_q^k$ define a connected tree leaf $\mathcal{T}_{p,q}^k = \left\{\left(z_{1,p}^k, z_{4,p}^k\right)\left(z_{1,q}^k, z_{4,q}^k\right)\right\}$, where $p, q \in [0 \ldots N_{\mathcal{T}^k}]$ and $N_{\mathcal{T}^k}$ total number of connected leafs. The proposed $\dot{z}_{1,p,q}^k$ velocity profile can be expressed as:

$$\dot{z}_{1,p,q}^k(\tau) = \dot{z}_{1,p}^k + \ddot{z}_{1,p,q}^k \cdot \tau\,, \tag{17}$$

as linear acceleration phase in $\tau \in [T_{s,p}^k, T_{s,p}^k + T_{a,p,q}^k]$ with $\ddot{z}_{1,p,q}^k$ and then moving forward with a constant speed of:

$$\dot{z}_{1,p,q}^k(\tau) = \dot{z}_{1,q}^k\,, \tag{18}$$

in the interval $\tau \in [T_{s,p}^k + T_{a,p,q}^k, T_{s,p}^k + T_{d,p,q}^k]$. For full control search capabilities the acceleration/deceleration $\ddot{z}_{1,p,q}^k$ is randomized enabling full longitudinal velocity control. The acceleration period $T_{a,p,q}^k$ is also randomized changing the

width of the trapezoidal velocity shape. The length of the $\mathcal{T}^k_{p,q}$ leaf depends also on the randomized time interval $T^k_{d,p,q}$ on which the velocity profile is applied. This is particularly interesting if the vehicle has to circumvent obstacles and a finer, i.e. smaller leaf length is needed for tree search. It is then evident that the total time $T^k_{s,p}$ up from the tree root to the node $\mathcal{T}^k_p$ depends on the history of profiles along the previous connected nodes. In terms of achieving a goal position, the final time τ^k_f in *k*-th resample period is therefore left as a free parameter in the search.

The velocity $\dot{z}_1$ can be expressed also as:

$$\dot{z}_1(\tau) = \dot{x}(\tau) + \frac{l}{2}\sin(\theta(\tau)) \cdot \dot{\theta}(\tau)\,. \tag{19}$$

By approximating the motion along leaf $\mathcal{T}^k_{p,q}$ as straight-line motion, the acceleration $\ddot{z}^k_{1,p,q}$ is bound by:

$$\|\ddot{z}^k_{1,p,q}\| = acc_{max} \cdot \cos\theta^k_{p,q}\,, \tag{20}$$

where acc_{max} denotes the maximal vehicle acceleration. $\theta^k_{p,q}$ is a constant vehicle orientation along the $\mathcal{T}^k_{p,q}$ leaf, according to Eq. 2 and Eq. 4 where $z_{3,p,q}(\tau) = \tan(\theta_{p,q}(\tau))$, representing the slope of a RRT line segment. Constant angle approximation is valid if the leaf length and therefore the interval sample time $T^k_{d,p,q}$ is small enough.

To calculate possible prohibited collision regions for a leaf $\mathcal{T}^k_{p,q}$ with obstacles $i = 1, \ldots, n^k_o$, time intervals $\tau \in \left[\tau^k_{i,p,q,1}, \tau^k_{i,p,q,2}\right]$ must be calculated. If any of $\tau^k_{i,p,q,1}$ or $\tau^k_{i,p,q,2}$ lies in interval $\tau \in [T^k_{s,p}, T^k_{s,p}+T^k_{d,p,q}]$ the *i*-th obstacle has a potential collision course with the vehicle.

By combining Eq. 10 with velocity profiles of Eq. 17 and Eq. 18 the following four $z^k_{1,p,q,1}$, $z^k_{1,p,q,2}$, $z^k_{1,p,q,3}$, $z^k_{1,p,q,4}$ potential bound positions are calculated for each *i*-th obstacle. On the linear acceleration part of the velocity profile it is obtained:

$$\begin{aligned} z^k_{1,p,q,\{1,2\}}(\tilde{\tau}^k_{i,p,q,\{1,2\}}) = \quad & z^k_{1,p} + \dot{z}^k_{1,p} \cdot \tilde{\tau}^k_{i,p,q,\{1,2\}} \\ & + \ddot{z}^k_{1,p,q} \cdot \frac{\left(\tilde{\tau}^k_{i,p,q,\{1,2\}}\right)^2}{2} \end{aligned} \tag{21}$$

and on the constant part of the velocity profile:

$$\begin{aligned} & z^k_{1,p,q,\{3,4\}}(\tilde{\tau}^k_{i,p,q,\{3,4\}}) = z^k_{1,p} + \dot{z}^k_{1,p} \cdot T^k_{a,p,q} + \\ & \ddot{z}^k_{1,p,q} \cdot \frac{\left(T_{a,p,q}\right)^2}{2} + \left(\tilde{\tau}^k_{i,p,q,\{3,4\}} - T^k_{a,p,q}\right) \cdot \dot{z}^k_{1,q}\,. \end{aligned} \tag{22}$$

Calculating the collision avoidance interval bounds from Eq. 14 yields:

$$\begin{aligned} x^k_i &= z^k_{1,p,q,\{1,3\}}(\tilde{\tau}^k_{i,p,q,\{1,3\}}) - r_i - R \\ x^k_i &= z^k_{1,p,q,\{2,4\}}(\tilde{\tau}^k_{i,p,q,\{2,4\}}) + \frac{l}{2} + r_i + R\,. \end{aligned} \tag{23}$$

By solving Eq. 21 and 22 with Eq. 23, the collision time bounds for *i*-th obstacle are obtained as:

$$\begin{aligned} \tau^k_{i,p,q,1} &= min\left\{\tilde{\tau}^k_{i,p,q,1}, \tilde{\tau}^k_{i,p,q,2}\right\} \\ \tau^k_{i,p,q,2} &= max\left\{\tilde{\tau}^k_{i,p,q,3}, \tilde{\tau}^k_{i,p,q,4}\right\}\,, \end{aligned} \tag{24}$$

if any of the time bounds is within the interval $\tau \in [T^k_{s,p}, T^k_{s,p}+T^k_{d,p,q}]$. Additionally, if entry into collision with *i*-th obstacle was detected in an adjacent previous leaf or before and no change is detected on the current leaf, the time bounds are set to $\tau^k_{i,p,q,1} = T^k_{s,p}$ and $\tau^k_{i,p,q,2} = T^k_{s,p} + T^k_{d,p,q}$.

As mentioned earlier, the vehicle angle $\theta^k_{p,q}$ can be considered constant for small leaf lengths (profiles). Since the leaf orientation is known in the search phase, the collision interval bounds for z_4 of Eq. 15 where any possible orientation was assumed, can be less strict. If the *i*-th obstacle can potentially collide with the vehicle on the interval, it's trace in the $z_4 - z_1$ plane is a parallelogram:

$$\begin{aligned} & z^k_{1,i,p,q,1} = z_1(\tau^k_{i,p,q,1})\,, \qquad z^k_{1,i,p,q,2} = z_1(\tau^k_{i,p,q,2}) \\ & z^k_{4,i,p,q,1} = y^k_i + v^k_{i,y}\tau^k_{i,p,q,1} - \tfrac{l}{2}\sin(\theta^k_{p,q}) - r_i - R\,, \\ & z^k_{4,i,p,q,2} = y^k_i + v^k_{i,y}\tau^k_{i,p,q,1} + \tfrac{l}{2}\sin(\theta^k_{p,q}) + r_i + R\,, \\ & z^k_{4,i,p,q,3} = y^k_i + v^k_{i,y}\tau^k_{i,p,q,2} - \tfrac{l}{2}\sin(\theta^k_{p,q}) - r_i - R\,, \\ & z^k_{4,i,p,q,4} = y^k_i + v^k_{i,y}\tau^k_{i,p,q,2} + \tfrac{l}{2}\sin(\theta^k_{p,q}) + r_i + R\,, \end{aligned} \tag{25}$$

where $z^k_{1,i,p,q,1}$ and $z^k_{1,i,p,q,2}$ are calculated depending on the velocity profile of Eq. 21 or 22.

B. RRT expansion search

In Sec. IV-A it was stated that four different parameters of each tree leaf $\mathcal{T}^k_{p,q}$ are randomized, namely, velocity profile's acceleration $\ddot{z}^k_{1,p,q}$, acceleration duration $T^k_{a,p,q}$, total leaf sample time $T^k_{d,p,q}$ and the orientation angle $\theta^k_{p,q}$. The first three parameters are related to the longitudinal control search, whereas the orientation angle is related to the steering control search. At each replanning sample instant k a tree $\mathcal{T}^k$ is grown from the current vehicle position $z_{init} = (z^k_1(0), z^k_4(0))$ to a goal region $\mathcal{U}(z^k_{goal}, z^k_{1,goal_{max}}, z^k_{4,goal_{max}}) = \left[z^k_1(\tau^k_f) \pm z_{1,goal_{max}}, z^k_4(\tau^k_f) \pm z_{4,goal_{max}}\right]$. The subgoal position z^k_{goal} is defined by higher level meta-planner based on global scenario waypoints. $z^k_{1,goal_{max}}$ and $z^k_{4,goal_{max}}$ are subgoal region bounds dependent on current vehicle velocity. The subgoal region is not bound to a single point, because it might be too restrictive in some cases, e.g. when a vehicle in front is just positioned "on the subgoal", therefore more maneuvering flexibility is given to the vehicle on intermediate goals, whereas the final goal $z_{goal}(T_f)$ is a point in state space. The time to reach subgoal τ^k_f and the final time T_f for the whole run are free parameters. Note that replanning must be started either at sample rate T_s from the *k*-th interval or even before, if $\tau^k_f < T_s$.

The trees are expanded through the EXTEND function in a random z_{rand} direction from the nearest z_{near} point on the tree obtained by NEAREST_NEIGHBOR function. An increment from z_{near} along the z_1 axis is based on the randomized velocity profile (Sec. IV-A). The possible new state is tested for collision against all obstacles in the NEW_STATE.

A list $\mathcal{L}$ of final nodes that reached the goal region is kept separately. The node with minimum cost and its backpointers

RRT_SEARCH($z_{init}, \mathcal{U}(z_{goal}, z_{1,goal_{max}}, z_{4,goal_{max}})$)
1 $\mathcal{T}$.init(z_{init});
2 **for** $n = 1$ **to** N **do**
3 $z_{new} \leftarrow$ EXTEND($\mathcal{T}$, z_{rand});
4 **if not** ($z_{new} = \varnothing$) **then**
5 **if**($z_{new} \in \mathcal{U}$) **then**
6 $\mathcal{L} \leftarrow \mathcal{L} \cup (z_{new}, cost)$;
7 **end**
8 **if not** $\mathcal{L} = \varnothing$ **then**
9 **return** MIN_PATH($\mathcal{T}$);
10 **else return** *Failure*;

EXTEND($\mathcal{T}$, z)
1 $z_{near} \leftarrow$ NEAREST_NEIGHBOR(z,$\mathcal{T}$);
2 $z_{new} \leftarrow$ NEW_STATE(z, z_{near});
3 **if not** ($z_{new} = \varnothing$) **then**
4 $\mathcal{T} \leftarrow \mathcal{T} \cup z_{new}$;
5 **return** z_{new};
6 **else return** $\varnothing$;

TABLE I
RRT SEARCH IN THE $z_4 - z_1$ SPACE.

form the final RRT path, which is a polyline. The cost of a single node $\mathcal{T}_p$ on the list $\mathcal{L}$ is pondered as:

$$\Gamma(\mathcal{T}_p) = \omega_{time} \cdot \Gamma_{time}(\mathcal{T}_p) + \omega_{\theta_{diff}} \cdot \Gamma_{\theta_{diff}}(\mathcal{T}_p) \qquad (26)$$

where Γ_{time} is the total time cost from the k-th resample time and $\Gamma_{\theta_{diff}}$ the total absolute orientation change of the vehicle along the path to $\mathcal{T}_p$ that should be minimized.

C. B spline interpolation

The RRT polyline polygon is oversampled at control points $z_4 = \mathcal{B}(z_1)$ with $\mathcal{B} = \{B_0, \ldots, B_m\}$:

$$b(z^*_{1_j}) = l(z^*_{1_j}) = B_j\,,\ j = 0, \ldots, m\,. \qquad (27)$$

at the Greville abscissae [14]:

$$z^*_{1_j} = \sum_{i=j+1}^{j+d} \frac{z_{1_i}}{d}\,. \qquad (28)$$

In order to calculate the state variables z_2 and z_3, the B-spline curve of degree $d = 4$ is derived as:

$$\begin{aligned} z_3 &= \tfrac{\mathrm{d}z_4}{\mathrm{d}z_1} = b'(z_1) = \textstyle\sum_{j=0}^{m} B_j N'_{j,d}(z_1)\,, \\ z_2 &= \tfrac{\mathrm{d}^2 z_4}{\mathrm{d}{z_1}^2} = b''(z_1) = \textstyle\sum_{j=0}^{m} B_j N''_{j,d}(z_1)\,, \\ \dot{z}_2 &= \tfrac{\mathrm{d}^3 z_4}{\mathrm{d}{z_1}^3} \cdot \dot{z}_1 = b'''(z_1) \cdot \dot{z}_1 = \textstyle\sum_{j=0}^{m} B_j N'''_{j,d}(z_1) \cdot \dot{z}_1\,. \end{aligned} \qquad (29)$$

The recursion relations for the basis functions and derivatives [14] at boundary conditions give the necessary starting and end control point positions.

V. URBAN SCENARIO SIMULATOR AND MOTION PLANNING RESULTS

The Smart Car simulator was developed in MatLab. It reproduces a simple 2D urban-like environment (approximately 800 m x 800 m) with parked and moving cars, buses, trucks, and people, buildings, walls, streets, and trees. Using the simulator one may reproduce the 2D kinematical behavior of a car-like vehicle. The real modified Smart Car used at the Autonomous System Lab is based on an ordinary smart fortwo passenger car that was equipped with several sensors and actuators for controlling the vehicle steering, accelerating, and breaking behaviors. Exteroceptive sensors with laser range finders and monocular cameras were installed. In simulator all environment static and dynamic features are represented by lines and/or arcs. The sensor data are extracted from the environment based on its geometrical description and used as input data for the algorithms. The simulator uses the global position of the Smart Car in the environment for selecting a feature-window that contains all lines and/or arcs close to the vehicle according to the exteroceptive sensor measurement. The Dijkstra algorithm [15] is used for calculating the shortest global path by taking into account the right-hand traffic convention. A sequence of intermediate positions from the vehicle's initial position to the goal is thus generated.

Sequence in Fig. (1a), (2a), (3a) shows different scenarios with intersection crossing of the vehicle (waiting for other vehicles to pass by), speeding up in free space area and slowing down in front of slowly moving vehicles. The RRT search is shown in Fig. (1b), (2b), (3b) with potential collision obstacle traces and the B-spline final trajectory which is obstacle free. The speed and the steering angle of the vehicle are shown in Fig. (1c), (2c), (3c). Specifications of the car are: max. speed $v_{max} = 10m/sec$, max. steering angle $\phi_{max} = 0.4rad$, max. acceleration $acc_{max} = 1.5m/sec^2$, the RRT leaf randomized time interval $T_{d,p,q} \in [1, 2]\,sec$ and global planning sample rate of $T_s = 2sec$. The RRT structure is built incrementally, so the obstacle traces drawn are valid only for one path variant. In Fig. (1b) it can be seen that the only obstacle trace within the search area (defined by the sensor range) is the one of the upmost vehicle moving down, whereas the other moving vehicles have already passed by the time the ego-vehicle reaches the collision zone. In the particular case, there was no solution found to drive into the intersection before the other vehicles due to the acceleration limits. In Fig. (3b) there is no trace of the obstacle in the front shown, since it is already gone from the search area by the time the ego-vehicle reaches the subgoal position. However, the ego-vehicle is still slowed down as a result of search for a feasible trajectory.

VI. CONCLUSION

The paper presents a motion planning algorithm for a nonholonomic car-like platform in dynamic environments with linearly moving obstacles. The potential collision check is performed in transformed space where time component is resolved in a criterion based on relative vehicle to obstacle motion. In consequence, generating a feasible path is a geometric search which is performed here using RRT expansion. Final smooth trajectory is obtained using B-spline interpolation. The approach enables full control of a vehicle both in terms of longitudinal and rotational velocity (i.e. speed and steering).

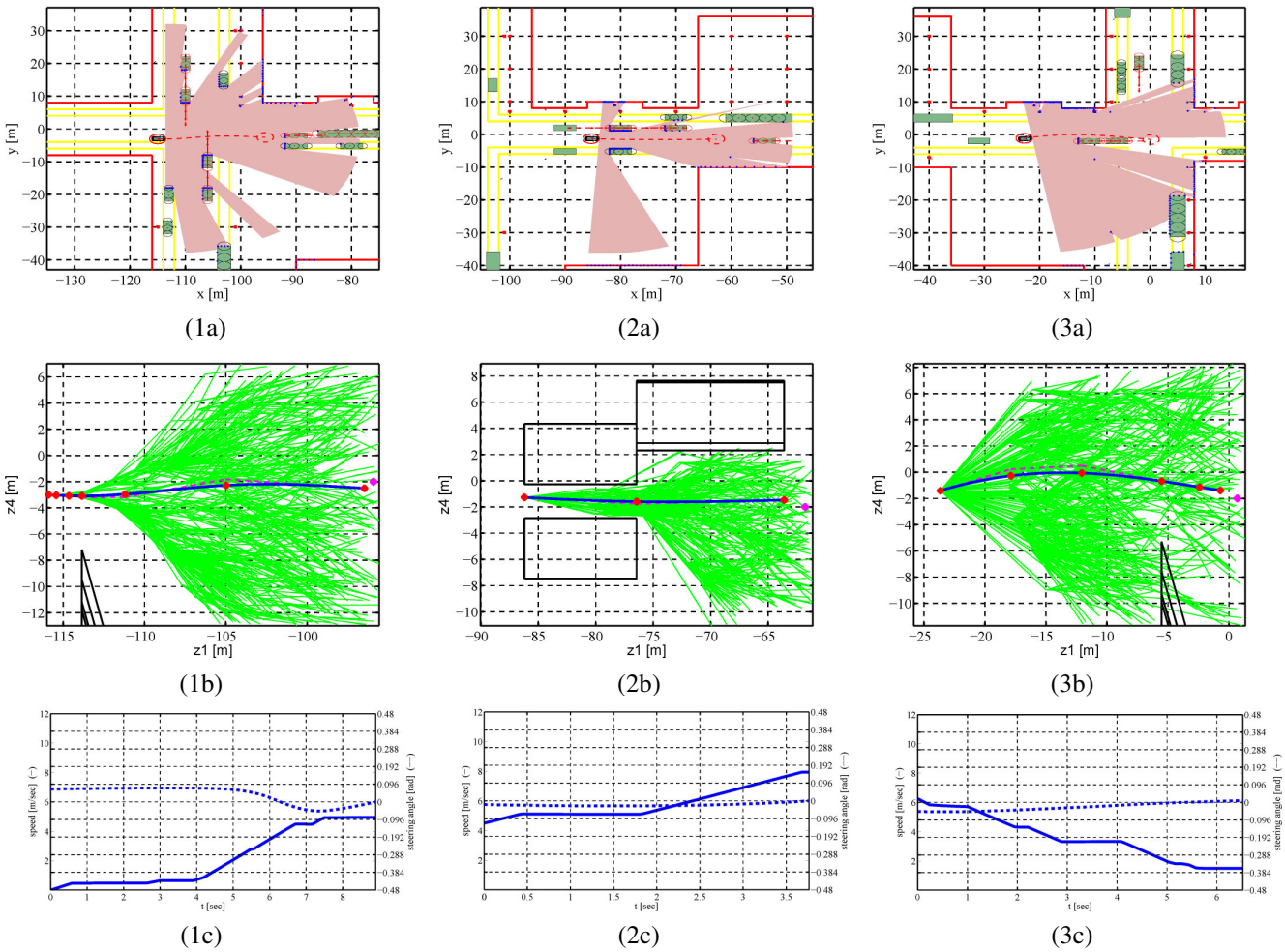

Fig. 1. Traffic simulation of ego-vehicle in a dynamic urban scenario. (1a) Intersection crossing (waiting for the other vehicles to pass by). (2a) Accelerating to nominal speed in free area. (3a) Slowing down in front of a slowly moving vehicle ahead. (1b), (2b), (3b) RRT search in transformed space, potential collision obstacle traces (polygons) and B-spline feasible trajectory. (1c), (2c), (3c) Longitudinal velocity $u_1(t)$ and steering angle $\phi(t)$ of the vehicle (full and dashed line, respectively).

Simulation results are provided for a urban traffic scenario. Future work would include extensive testing with respect to computational effort where real-time requirements should be met before implementing the strategy on a real vehicle.

REFERENCES

[1] M. Erdmann and T. Lozano-Perez, "On multiple moving objects," in *Proc. IEEE Int. Conf. Robotics and Automation*, San Francisco,CA, Apr. 1986, pp. 1419–1424.

[2] K. Kant and S. W. Zucker, "Planning collision free trajectories in time-varying environment: A two level hierarchy," in *Proc. IEEE Int. Conf. Robotics and Automation*, Raleigh, NC, 1988, pp. 1644–1649.

[3] T. Fraichard, "Dynamic trajectory planning with dynamic constraints: a state-time approach," in *IEEE/RSJ Int. Conf. on Intell. Robots and Sys.*, 1993, pp. 1393–1400.

[4] P. Fiorini and Z. Shiller, "Motion planning in dynamic environments using velocity obstacles," *Int. J. Robot. Res.*, vol. 17, pp. 760–772, 1998.

[5] Z. Shiller, F. Large, and S. Sekhavat, "Motion planning in dynamic environments: Obstacles moving along arbitrary trajectories," in *Proc. IEEE Int. Conf. Robotics and Automation*, Seoul, Korea, May 2001, pp. 3716–3721.

[6] D. Hsu, R. Kindel, J.-C. Latombe, and S. Rock, "Randomized kinodynamic motion planning with moving obstacles," *Int. J. Robot. Res.*, vol. 21, pp. 233–255, 2002.

[7] Xing-Jian Jing, "Behavior dynamics based motion planning of mobile robots in uncertain dynamic environments," *Robotics and autonomous systems*, vol. 53, pp. 99–123, 2005.

[8] T.Fraichard and H.Asama, "Inevitable collision states. a step towards safer robots?," *Advanced Robotics*, vol. 10, pp. 1001–1024, 2004.

[9] S.Petti and T.Fraichard, "Safe motion planning in dynamic environments," in *IEEE/RSJ IROS*, Edmonton,CA, 2005, pp. 2210–2215.

[10] Z. Qu, J. Wang, and C.E. Plaisted, "A new analytical solution to mobile robot trajectory generation in the presence of moving obstacles," *IEEE Transactions on Robotics*, vol. 20, pp. 978–993, 2004.

[11] K.Macek and R.Siegwart, "Motion planning in the presence of moving obstacles using rrt search and b-splines," in *In Proceeding of the 8th International IFAC Symposium on Robot Control (SYROCO)*, Bologna,Italy, 2006.

[12] S. M. LaValle and J. J. Kuffner, *Algorithmic and Computational Robotics: New Directions*, chapter Rapidly-exploring random trees: Progress and prospects, pp. 293–308, Wellesley,MA, 2001.

[13] J.P. Laumond, *Robot motion planning and control*, Lecture notes in control and information sciences 229. Springer, 1998.

[14] C. de Boor, *A practical guide to splines*, Applied mathematical sciences 27. Springer, 2001.

[15] E. W. Dijkstra, "A note on two problems in connexion with graphs," *Numerische Mathematik*, vol. 1, pp. 269–271, 1959.