2018 IEEE Intelligent Vehicles Symposium (IV)
Changshu, Suzhou, China, June 26-30, 2018

Trajectory Planning for Autonomous Vehicles in Time Varying Environments Using Support Vector Machines

Mahdi Morsali[1], Jan Åslund[2], and Erik Frisk[3]

***Abstract*— A novel trajectory planning method is proposed in time varying environments for highway driving scenarios. The main objective is to ensure computational efficiency in the approach, while still ensuring collision avoidance with moving obstacles and respecting vehicle constraints such as comfort criteria and roll-over limits. The trajectory planning problem is separated into finding a collision free corridor in space-time domain using a support vector machine (SVM), which means solving a convex optimization problem. After that a time-monotonic path is found in the collision free corridor by solving a simple search problem that can be solved efficiently. The resulting path in space-time domain corresponds to the resulting planned trajectory of the vehicle. The planner is a deterministic search method associated with a cost function that keeps the trajectory kinematically feasible and close to the maximum separating surface, given by the SVM. A kinematic motion model is used to construct motion primitives in the space-time domain representing the non-holonomic behavior of the vehicle and is used to ensure physical constraints on the states of the vehicle such as acceleration, speed, jerk, steer and steer rate. The speed limits include limitations by law and also rollover speed limits. Two highway maneuvers have been used as test scenarios to illustrate the performance of the proposed algorithm.**

I. Introduction

Autonomous driving has become a demanding field and many disciplines such as computer vision, machine learning, sensor fusion, control, vehicle dynamics, and planning methods have been integrated in order to compute accurate, safe and comfortable plans for autonomous vehicles. In this study a supervised machine learning technique and a search based trajectory planning have been combined for safe path and speed planning of a vehicle. As an example, consider a take over maneuver using the oncoming lane with a vehicle approaching as shown in Fig. 1. If the oncoming vehicle is close, it is necessary to speed up in order to finish the take over maneuver before the gap closes, or detect that this is not possible and wait for the other vehicle to pass before overtaking. It is therefore necessary to find a feasible trajectory taking into account positions of other vehicles as a function of time and, e.g., limitations on acceleration and speed.

The application of intelligent methods in path planning has increased in recent years [1], [2], [3], [4]. The idea of path planning using Support Vector Machines (SVMs) was initially proposed by J. Muira [5], where the method is found to be suitable in robotics. The calculation of nonlinear safe paths for robots with relatively low cost is found to be main features of SVM path planning. However, due to the non-holonomic nature of autonomous vehicles the method might not be directly suitable for cars. X. Li et al. [6] used SVM for calculation of a safe region in a two dimensional space and then using a motion kinematics to calculate a path for a non-holonomic vehicle. In [7] particle filtering is used to estimate a region for moving obstacles in 2D space in a time horizon and then using SVM to calculate a collision free path. Q. H. Do et al. [8] used a fast marching method and SVM in order to find a path for a non-holonomic vehicle. When considering not only paths but also trajectories, i.e., a path with a velocity profile, different trajectory planning solutions have been addressed in the literature. In [9], trajectory planning is conducted by initially finding a maneuver aiming at minimizing collisions and next calculating a trajectory that optimizes travel time while respecting constraints such as traffic rules and comfort. Similarly, in [10] the trajectory planning has been divided into steering planning and speed planning with limitations imposed by vehicle mechanism and drive torque. Consideration of vehicle friction characteristics while trajectory planning is investigated in [11].

Authors are with Electrical Engineering Department, Linköping University, 581-83 SE, Linköping, Sweden
[1] mahdi.morsali@liu.se
[2] jan.aslund@liu.se
[3] erik.frisk@liu.se

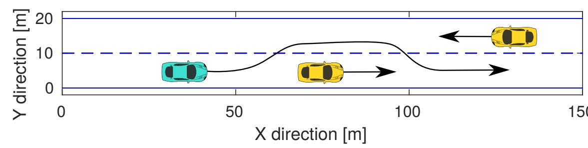

Fig. 1. A schematic of a scenario with the green vehicle overtaking the vehicle in front with oncoming traffic in the opposing lane.

The trajectory planning method proposed here extends previous work, e.g., [5], [6], to finding corridors in the space-time domain. This makes it possible to do path and velocity planning simultaneously while respecting moving obstacles and vehicle dynamic limitations, for example steer-rates and roll-over constraints. The approach relies on a convex optimization formulation that can be solved efficiently in real-time, combined with a simple search problem.

II. Trajectory Planning

A first step in solving the trajectory planning problem, illustrated in Fig. 1, is to first transform it into a path planning problem in the space-time domain. A feasible time-monotonic path, the red path in Fig. 2, is then sought with space and time as states. The other vehicles trajectories are obstacles, indicated with the red tubes in Fig. 2. The resulting path in space-time domain corresponds to the resulting planned trajectory of the vehicle. See Chapter 7 in [12] for further details. The path planning is solved by first defining two sets

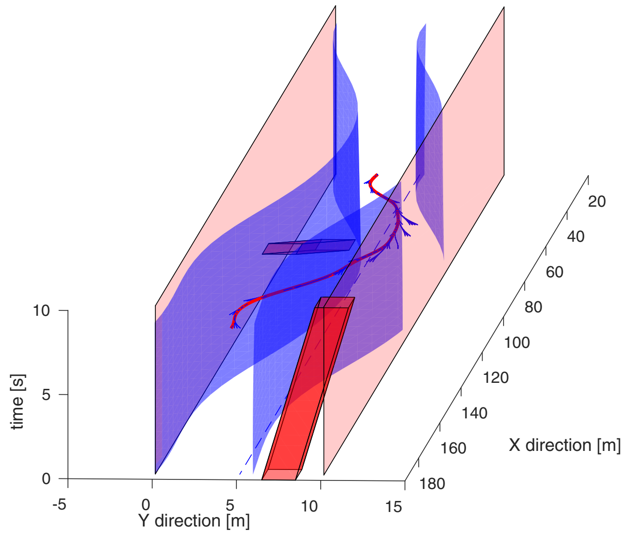

Fig. 2. Test scenario with obstacles, calculated corridor and trajectory.

in the space-time domain, each consisting of the vehicle and the road lines to the left and right, respectively. Then SVM is used to find a maximum margin classifier that separates these two sets. The margin boundaries of the classifier, the blue surfaces in Fig. 2, are used, together with a safety distance and the size of the car, to define a collision corridor in the space-time domain free of moving and stationary obstacles.

Finally the path planning problem is solved inside the collision free corridor with a search based path planning method using motion primitives. The main safety criterion considered here is rollover prevention, which is relevant mainly for heavy-duty trucks with a high center of gravity. Steer, steer-rate, acceleration, and jerk limitations provide smoothness of the trajectory and comfort for the passengers. The comfort criteria mentioned, prevent unexpected motion acceleration or instantaneous changes of the steer rate and steer angle. It is important to note that the aim is not to minimize jerk or maximize comfort and it is only constrained to prevent uncomfortable maneuvers.

The introduced planning method is suitable for real time applications since the calculation of a collision free corridor reduces complexity for the trajectory planning algorithm and therefore reduce the total computational effort. A convex optimization problem and a search based trajectory planning using a non-holonomic kinematics form the core of algorithm used in this paper. The approach outlined above requires knowledge on trajectories for moving obstacles and in a real situation, this knowledge is typically not available, or highly uncertain. To handle uncertain traffic behavior, the approach is possible to use in a model-predictive controller, where the plan is continuously updated by iterative replanning over a finite receding horizon. It is also possible to include uncertainty regions in a neighborhood of the obstacles in the space-time domain.

III. The Collision Free Corridor

An SVM approach is used to define a collision free corridor in the space-time domain. Time-monotonic paths in this corridor represents collision free trajectories for the vehicle. An important property of SVM is that it is formulated as a convex optimization problem that can be solved efficiently resulting in two surfaces representing the collision free corridor.

Before introducing the optimization problem, a test scenario is discussed and Fig. 3 is used to illustrate how the data for SVM approach is specified and classified. For each scenario, a start and destination point is allocated and guide points for these points are constructed in the space-time coordinate system. The left and right guide points, independent of their position, are labeled as -1 and $+1$ classes, respectively. As shown in the figure, the position of points representing stationary obstacles does not change with time, while points corresponding to moving obstacles change with time. Here, the objects on the left side of the road center line are labeled as -1 class and the objects on the right side of the road are labeled as $+1$ class. Similar to the stationary obstacles, if the majority of the calculated points in the moving obstacle horizon are in the left or right side of the center line it is classified as -1 or $+1$, respectively.

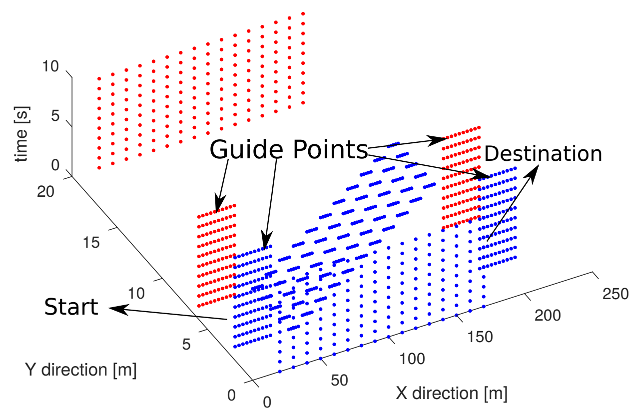

Fig. 3. Illustration of guide points, moving and stationary obstacles in space-time domain, with corresponding labels.

The SVM technique is used as a classification method that gives a decision surface that separates the two classes. The approach in [13] will be used in this paper, which gives an exact separation with no misclassified points if possible or otherwise fails, in contrast to soft margins as in [14] where some of the points are allowed to be misclassified with a penalty that increases with the distance to the boundary. In the present application this a natural choice since a misclassified point might lead to a collision. To obtain the separation of the classes, the convex problem

$$\begin{aligned} \max_{\alpha_i} \quad & \sum_{i=1}^{n} \alpha_i - \tfrac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} \alpha_i \alpha_j l_i l_j k(w_i, w_j) \\ \text{s.t.} \quad & \alpha_i \geq 0 \\ & \sum_{i=1}^{n} \alpha_i l_i = 0 \end{aligned} \tag{1}$$

is first solved, where k is the Gaussian kernel function

$$k(w_i, w_j) = \exp\left(\frac{-\|w_i - w_j\|^2}{2\sigma^2}\right) \tag{2}$$

$w_i = (x_i, y_i, t_i)$, $i = 1, \ldots, n$, are obstacle points, $l_i = \pm 1$ the corresponding labels, and α_i are Lagrangian multipliers. Then, the sign of the indicator function

$$f(w) = \sum_{i \in N_{sv}} \alpha_i l_i k(w_i, w) + b \tag{3}$$

is used as a classifier of a point w where $N_{sv} = \{i : \alpha_i > 0\}$ is the index set of the support vectors, and b is the bias term

$$b = \frac{\sum_{j \in N_{sv}} (l_j - \sum_{i=1}^{n} \alpha_i l_i k(w_j, w_i))}{|N_{sv}|}. \tag{4}$$

The set $\{w : |f(w)| < 1\}$ represents a corridor that does not contain any of the obstacle points w_i. The points that result in $+1$ or -1 represents the margin boundaries, which contain all the support vectors w_i, $i \in N_{sv}$, and the set where $f(w) = 1$ is the decision surface.

To define the collision free corridor, the size of the of the car has to be taken into account, as well as some safety distances. The longitudinal and lateral directions will be treated quite differently in the suggested approach. First of all, the longitudinal velocity is often much higher than the lateral velocity, and the corresponding safety distance have to be of a different order. Furthermore, in this case the velocity vector is almost parallel with the x-axis, and the longitudinal and later direction can be assumed to be approximated by the x- and y-direction, respectively.

The obstacle free corridor introduced above, shall now be used to define the collision free corridor. Let Δx be the sum of the longitudinal safety distance and half the length of the car and let Δy be the sum of the lateral safety distance and half the width of the car. The quantity Δx is added before the obstacle free corridor is defined using the margin boundaries. This will prevent collision in the longitudinal direction. See Fig. 4. However the same approach can not be used in the lateral direction. The reason for this is that adding Δy, in e.g., the scenario considered in Fig. 1, would result in a situation where there is no exact separation of the classes at a time point where the two yellow cars meet, and consequently the exact separation will fail in the time-space domain as well. Therefore, the quantity Δy is added after the obstacle free corridor has been defined and the collision free corridor is defined as the set of points in the obstacle free corridor whose distance to the margin boundaries is greater than Δy, see Fig. 4. Given a point $w = (x, y, t)$ it is not straightforward to calculate the distance to a margin boundary. However, if the distance is approximated by the distance to the closest point on the margin boundary in the y-direction, then it is possible to verify if the point is in the collision free corridor by taking steps of size Δ_y in the y-direction towards the two margin boundaries and then verify if the new points are inside the obstacle free corridor, i.e., if

$$|f(x, y \pm \Delta y, t)| < 1. \tag{5}$$

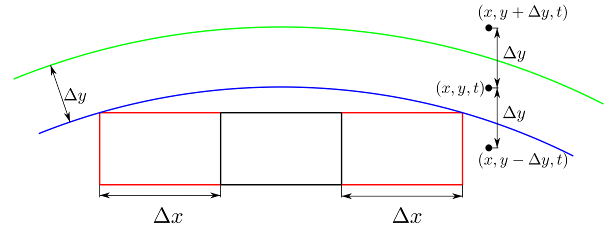

Fig. 4. Illustration of construction of the collision free corridor for a fixed time. The blue and green curves are the boundaries of the obstacle free and collision free corridor, respectively.

Note that the actual distance does not have to be calculated when using this approach. A refinement of the approximation is to, instead of moving in the y-direction, move in the direction of gradient of f with respect to the spatial coordinates x and y, which can be computed by differentiating the indicator function (3).

Consider the scenario in Fig. 1 again. If it is not possible to finish the take over maneuver before the gap closes, due to limitations on, e.g., acceleration or speed, then there is a feasible path in the collision free corridor that corresponds to waiting for the meeting vehicle to pass before overtaking, given that time horizon is long enough and possible to drive faster than the vehicle to be overtaken.

IV. Path and Speed Planning

After determining a collision free corridor for the vehicle, the next step is to find a path and speed profile for the non-holonomic vehicle. The path planning is done in the 3 dimensional space-time domain and the kinematics model used for path and speed planning [15] is

$$\begin{aligned}
\dot{X} &= v\cos(\theta) \\
\dot{Y} &= v\sin(\theta) \\
\dot{\theta} &= v\frac{\tan(\delta)}{L} \\
\dot{\delta} &= u \\
\dot{s} &= v \\
\dot{v} &= a \\
\dot{a} &= J
\end{aligned} \tag{6}$$

where X and Y are the vehicle position, θ the heading, δ the steer angle, s distance, v speed, and a acceleration. The steer rate, u, and jerk, J, are used as inputs in order to calculate motion primitives. Fig. 5, illustrates some examples of motion primitives for a vehicle with initial condition at $(x, y, t, \theta, \delta, s, v, a) = (0, 0, 0, 0, 0, 0, 0, 0)$. Three steer rates and three jerk amounts are used to integrate and construct motion primitives. The red, green and blue motion primitives correspond to maximum, zero and minimum jerk amounts, respectively. The three green motion primitives are all the same because the initial speed, acceleration and jerk are all equal to zero and therefore the vehicle's position does not change, irrespective of the steer rate.

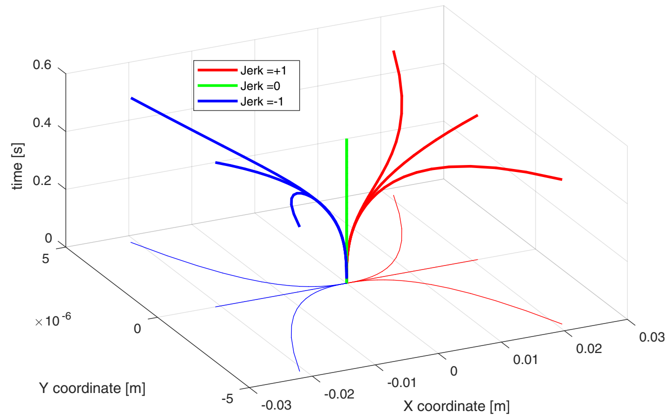

Fig. 5. Motion primitives in 3D space.

The objective function used in the search algorithm is

$$\text{cost} = t_f + \lambda \int_0^{t_f} |f|\, dt \tag{7}$$

where t_f is the final time, f is the function in (3), and λ is a weighting factor. The second penalty term is included since the objective is to keep close to $f = 0$, corresponding to the centerline of the obstacle free corridor. It is important to note that the path corresponding to $f = 0$ might not be kinematically feasible [12]. The trajectory planning algorithm used in this paper is a heuristic based search method. The heuristic is the Euclidean distance between the motion primitives end point and the maneuver destination divided by maximum allowable speed by law,

$$H = |P_{dest} - P_{pf}| / v_{max}. \tag{8}$$

The points with minimum cost are the candidates in each loop to be integrated and construct new motion primitives with inputs ranging from minimum steer rate and jerk to maximum steer rate and jerk. For instance if there are n inputs for steer rate and m inputs for jerk, a total number of nm primitives will be constructed in each loop. The new motion primitives in each loop will be added to the queuing system if all the points in the primitives

- are within the collision free corridor
- have allowable steer angle that can satisfy the servo motor properties
- have the speed below allowed speed by law
- have the speed that is below rollover limits of the vehicle
- have the acceleration within the range of acceleration that vehicle can provide

In order to prevent rollover of the vehicle, a static rollover criterion is used and the critical lateral acceleration is calculated by [16]

$$\frac{a_{ymax}}{g} = \frac{t_w}{2h_{cg}}$$

where t_w is the width of the vehicle and h_{cg} is the height of center of gravity. The maximum velocity to prevent rollover is calculated by

$$v_{max} = \sqrt{\left|\frac{a_{ymax}}{C}\right|}$$

where C is the curvature of the calculated paths and C is calculated by [17]

$$C = \frac{\left(\frac{dx}{ds}\frac{d^2y}{ds^2} - \frac{dy}{ds}\frac{d^2x}{ds^2}\right)}{\left(\left(\frac{dx}{ds}\right)^2 + \left(\frac{dy}{ds}\right)^2\right)^{3/2}}.$$

The search based trajectory planning is summarized in Algorithm 1. The state of the initial point and the coordinates of the destination is used as inputs in the algorithm (line 1). The state of the initial point includes position, heading angle, traveled distance, speed, acceleration, and steer angle. In the algorithm, P represents the position of the states. The points with minimum cost are the candidates for next iteration to construct motion primitives (line 6). Jerk and steer rate are used as control inputs where jerk is selected from a set J (line 10) ranging from minimum possible to maximum allowed jerk. Similarly, steer rate is selected from a set $\dot{\delta}$ (line 11) that ranges from minimum to maximum steer rates. By using the kinematics of the motion, inputs and point with minimum cost, new motion primitives are created (line 12). The motion primitives are examined to see if they are collision free and meet comfort and safety factors (lines 13 to 17). If the new points satisfy all limitations they are added to the queuing system (lines 19 to 22). In line 22, γ is a penalty factor in total cost function. The algorithm keeps searching until reaching the destination (lines 7 and 8).

Algorithm 1 Trajectory planning

```
1:  procedure TRAJECTORYFINDER (start_state, dest)
2:      Q = PriorityQueue
3:      cost_to_go = h(P(start_state),dest) / v_max
4:      Q.insert(cost_to_go, start_state)
5:      while Q ≠ ∅ do
6:          s = Q.GetFirst() % arc with min cost
7:          if P(s) ∩ B(dest, r) ≠ ∅ then
8:              return success
9:          s_final = s(final)%Final point in the arc (s)
10:         for j ∈ J do
11:             for steer ∈ δ̇ do
12:                 s_new = Integrate(kinematic, s_final, j, steer)
13:                 Limit_1 = P(s_new) ∈ corridor
14:                 Limit_2 = a(s_new) ∈ acceleration limits
15:                 Limit_3 = δ(s_new) ∈ steer angle limits
16:                 Limit_4 = v(s_new) ∈ speed limits
17:                 Limit = ∧{Limit_i}_{i=1,2,3,4}
18:                 s_new,final = s_new(final)
19:                 if Limit then
20:                     H = h(P(s_new,final), dest)/v_max
21:                     g = t(s_new,final) + λ ∫ |f_{s_new}| dt
22:                     Q.insert(g + γH, s_new)
23:     return failure
```

V. Results and Discussion

The only tuning parameter when calculating the obstacle free corridor using the SVM is the kernel width parameter σ in (2). Fig. 6 shows the effect on the corridor and the trajectory of using a larger and a lower value of the kernel parameter σ. In Fig. 6, the surfaces in red and blue correspond to a lower and higher value of σ respectively. The surface in black and green colors are the maximum separating surfaces ($f=0$) for the large and low value of σ respectively. The blue and red surfaces represent the corridor surfaces ($f=1$ and $f=-1$). As illustrated by the figure, the larger the σ value, the narrower the corridor and also lower curvature. It is important to note that the maximum separating surfaces are not consistent with the non-holonomic behavior of the vehicle and therefore, speed and path planning is performed after SVM corridor calculation. In order to evaluate the proposed

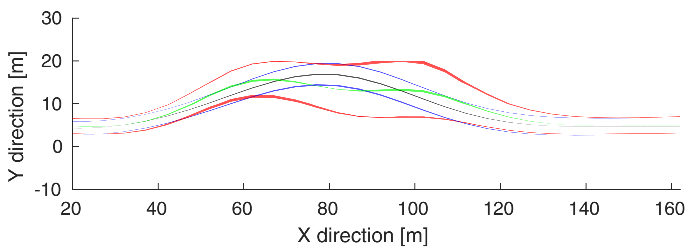

Fig. 6. The calculated corridor and maximum separating surface using two different σ values.

trajectory planning algorithm, two test scenarios are designed. The test scenarios are highway maneuvers with different configurations. In the first test scenario the ego vehicle aims to take over the vehicle in front and also there is another vehicle passing in the lane parallel to the main vehicle.

The stationary and moving obstacles (position and velocity are assumed to be known) are mapped into the (x,y,t) coordinate system. Using the SVM technique, a collision free corridor which is free of obstacles is calculated. The calculation time of collision free corridor in both scenarios is approximately 0.05 seconds using the `svmtrain` command in MATLAB. In Fig. 7 the moving obstacles are represented by the red tube, the pale red planes represent the road lines that are considered as stationary obstacles, and the red curve represents the planned trajectory. All planes and surfaces are shown in transparent colors. The corridor calculated by SVM is shown in blue.

Fig. 8, shows the path, curvature and steer rate as input. The red profile is the receding horizon of the moving obstacle which shall be avoided. In Figs. 8 and 10, the position of the vehicle and moving obstacles are cross-marked every 1 second. In Fig. 9, the speed, acceleration and jerk profiles are depicted. The dashed red line shows the imposed speed constraints by the legal and the rollover criteria. It is shown that the motion primitives and its states are within prescribed limits.

Figs, 2, 10 and 11 show the test results for the second scenario where the vehicle takes over a vehicle in front of it and another vehicle approaching in the other lane in the opposite direction to the vehicle, as described in Fig. 1.

The conducted simulations show how the collision free corridors are calculated using the SVM and results of the subsequent search based path and speed planning. In order

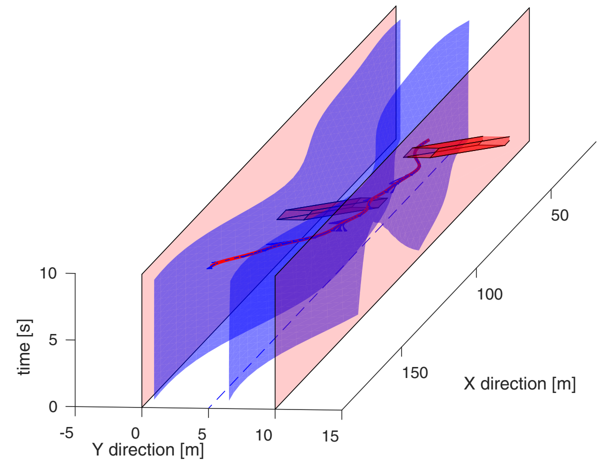

Fig. 7. First test scenario with obstacles, calculated corridor and trajectory.

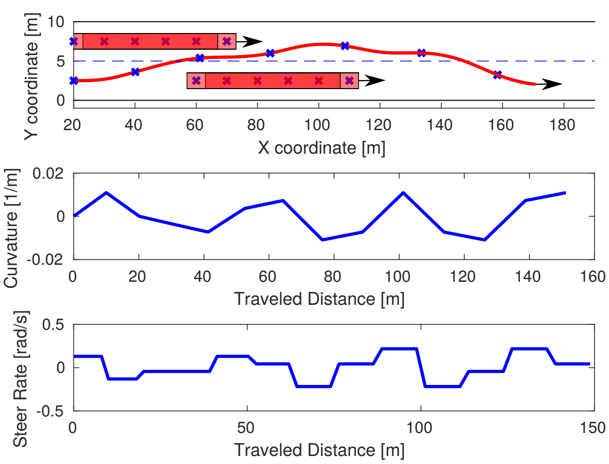

Fig. 8. First test scenario with the calculated path, curvature and steer rate.

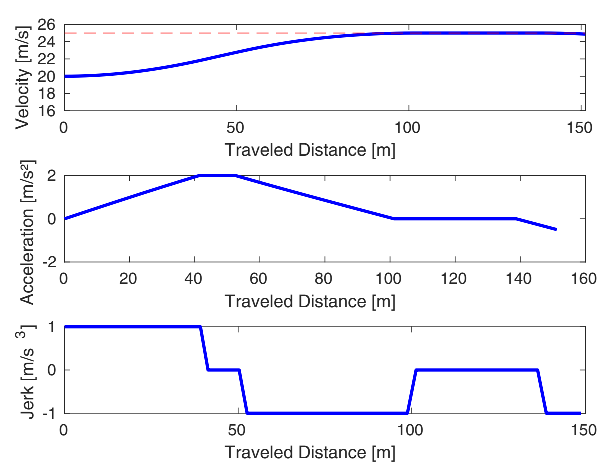

Fig. 9. First test scenario with the calculated speed profile, acceleration and jerk.

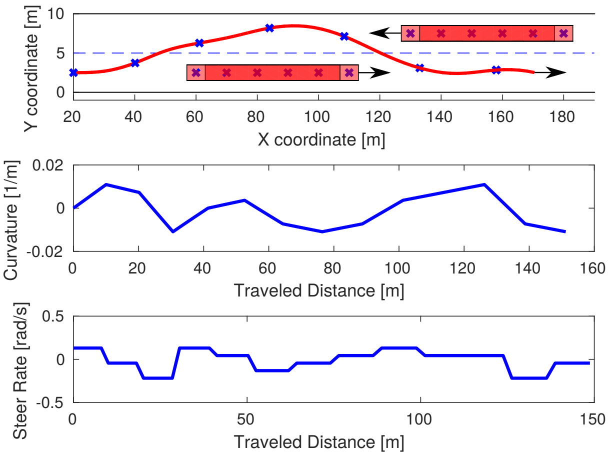

Fig. 10. Second test scenario with the calculated path, curvature and steer rate.

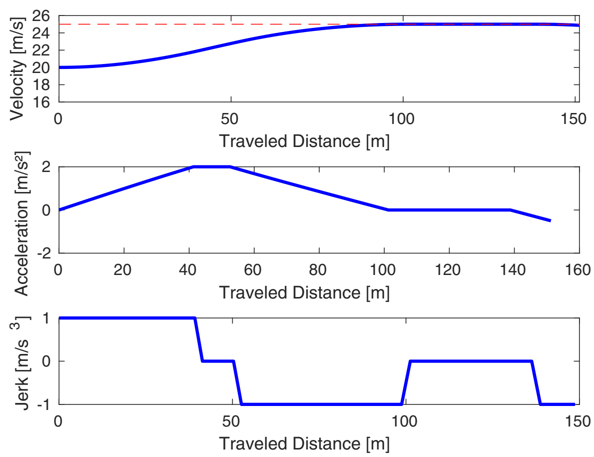

Fig. 11. Second test scenario with the calculated speed profile, acceleration and jerk.

to solve the convex optimization problem, the σ parameter must be tuned properly in order to get wide enough corridor for path planning and at the same time smoother maximum separating surface. The limitations imposed on the selection of motion primitives results in calculation of smooth paths and comfortable speed profiles. Furthermore, the safety criteria applied in speed profile guaranties prevention of rollover and unwanted speeds.

VI. Conclusions

In this paper, an SVM technique has been combined with a heuristic search technique to perform trajectory planning while avoiding stationary and moving obstacles. The SVM provides an obstacle free corridor and the combination with search based trajectory planning improves the overall computational efficiency. The collision free corridor formulation is the result of solving a convex optimization problem in space-time domain. The trajectory search is also computationally simple since a the objective is finding a kinematically feasible trajectory in an obstacle free corridor. Proper tuning of influencing parameters are crucial in the quality of the corridor and the smoothness of the calculated path by trajectory planning. The trajectory planning method proposed here uses safety and comfort criteria to eliminate unwanted motion primitives. Smooth paths are the result of steer and steer rate limitations imposed on the calculation of motion primitives. Furthermore, comfortable speed profiles are due to the jerk and acceleration limits. The trajectory planning eliminates steer rate and jerk inputs that results in unsafe speed profiles, the speed profiles that do not meet rollover and speed limit criteria. The introduced test scenarios represent the properties of the algorithm. It is shown that the vehicle avoids moving and stationary obstacles and the states of the vehicle are within the introduced bounds. The method is found to be computationally effective and can be used in real time applications.

References

[1] C. B. Xia, W. Nan, and D. Yong, “The semi-supervised support vector machine of path planning,” in *Measuring Technology and Mechatronics Automation (ICMTMA), 2013 Fifth International Conference on*. IEEE, 2013, pp. 1230–1232.

[2] F.-Y. Chou, C.-Y. Yang, and J.-S. Yang, “Support vector machine based artificial potential field for autonomous guided vehicle,” in *Proc. 4th Int. Symp. Precision Mechanical Measurements*, 2008, p. 71304J.

[3] N. Morales, J. Toledo, and L. Acosta, “Path planning using a multiclass support vector machine,” *Applied Soft Computing*, vol. 43, pp. 498–509, 2016.

[4] C. Qingyang, S. Zhenping, L. Daxue, F. Yuqiang, and L. Xiaohui, “Local path planning for an unmanned ground vehicle based on svm,” *International Journal of Advanced Robotic Systems*, vol. 9, no. 6, p. 246, 2012.

[5] J. Miura, “Support vector path planning,” in *Intelligent Robots and Systems, 2006 IEEE/RSJ International Conference on*. IEEE, 2006, pp. 2894–2899.

[6] X. Li, Z. Sun, A. Kurt, and Q. Zhu, “A sampling-based local trajectory planner for autonomous driving along a reference path,” in *Intelligent Vehicles Symposium Proceedings, 2014 IEEE*. IEEE, 2014, pp. 376–381.

[7] Q. Huy, S. Mita, H. T. N. Nejad, and L. Han, “Dynamic and safe path planning based on support vector machine among multi moving obstacles for autonomous vehicles,” *IEICE TRANSACTIONS on Information and Systems*, vol. 96, no. 2, pp. 314–328, 2013.

[8] Q. H. Do, S. Mita, and K. Yoneda, “Narrow passage path planning using fast marching method and support vector machine,” in *Intelligent Vehicles Symposium Proceedings, 2014 IEEE*. IEEE, 2014, pp. 630–635.

[9] S. Glaser, B. Vanholme, S. Mammar, D. Gruyer, and L. Nouveliere, “Maneuver-based trajectory planning for highly autonomous vehicles on real road with traffic and driver interaction,” *IEEE Transactions on Intelligent Transportation Systems*, vol. 11, no. 3, pp. 589–606, 2010.

[10] D. Wang and F. Qi, “Trajectory planning for a four-wheel-steering vehicle,” in *Robotics and Automation, 2001. Proceedings 2001 ICRA. IEEE International Conference on*, vol. 4. IEEE, 2001, pp. 3320–3325.

[11] E. Velenis and P. Tsiotras, “Optimal velocity profile generation for given acceleration limits; the half-car model case,” in *2005 IEEE International Symposium on Industrial Electronics*, 2005, pp. 355–360.

[12] S. M. LaValle, *Planning algorithms*. Cambridge university press, 2006.

[13] C. Cortes and V. Vapnik, “Support-vector networks,” *Machine learning*, vol. 20, no. 3, pp. 273–297, 1995.

[14] K. P. Bennett and O. L. Mangasarian, “Robust linear programming discrimination of two linearly inseparable sets,” *Optimization Methods and Software*, vol. 1, no. 1, pp. 23–34, 1992.

[15] R. Rajamani, *Vehicle dynamics and control*. Springer Science & Business Media, 2011.

[16] K. Lundahl, C. F. Lee, E. Frisk, and L. Nielsen, “Analyzing rollover indices for critical truck maneuvers,” *SAE International Journal of Commercial Vehicles*, vol. 8, no. 2015-01-1595, pp. 189–196, 2015.

[17] J. D. Lawrence, *A catalog of special plane curves*. Courier Corporation, 2013.