

On Path Planning Methods for Automotive Collision Avoidance

David Madås*, Mohsen Nosratinia[‡], Mansour Keshavarz[†], Peter Sundström[†],
Roland Philippsen[†], Andreas Eidehall*, Karl-Magnus Dahlén[‡]

* Volvo Car Corporation, 405 31 Göteborg, Sweden
dmadas@volvocars.com
aelaidehal@volvocars.com

[†] Volvo Group Trucks Technology, 405 08 Göteborg, Sweden.
mansour.keshavarz@volvo.com,
{peter.sundstrom, roland.philippsen}@consultant.volvo.com

[‡] HiQ Göteborg, 411 06 Göteborg, Sweden
mohsen.nosratinia@hiq.se
karl-magnus.dahlen@hiq.se

Abstract—There is a strong trend for increasingly sophisticated Advanced Driver Assistance Systems (ADAS) such as Autonomous Emergency Braking (AEB) systems, Lane Keeping Aid (LKA) systems, and indeed autonomous driving. This trend generates a need for online maneuver generation, for which numerous approaches can be found in the large body of work related to path planning and obstacle avoidance.

In order to ease the challenge of choosing a method, this paper reports quantitative and qualitative insights about three different path planning methods: a state lattice planner, predictive constraint-based planning, and spline-based search tree. Each method is described, implemented and compared on two specific traffic situations.

The paper will not provide a final answer about which method is best. This depends on several factors such as computational constraints and the formulation of maneuver optimality that is appropriate for a given assistance or safety function. Instead, the conclusions will highlight qualitative merits and drawbacks for each method, in order to provide guidance for choosing a method for a specific application.

I. INTRODUCTION

The number of active safety systems being implemented in road vehicles is ever increasing. At the moment, AEB systems and LKA systems are available from a number of manufacturers [1], [2], [3]. While AEB systems reduce or mitigate collisions by braking the vehicle, LKA systems focus on keeping the vehicle within the lane by applying steering torque. Typically, objects being handled include different types of vehicles and Vulnerable Road Users (VRUs), but development is ongoing towards detecting animals and other road objects. Meanwhile, more and more traffic situations are being addressed, such as intersections. With this increasing scenario complexity comes increasing requirements on efficient and versatile threat assessment algorithms.

When carrying out an LKA steering intervention a lateral trajectory is typically planned that avoids objects and takes the vehicle back into the original lane. Similarly, when taking the decision to make an AEB intervention, the system needs to be sure that the driver does not intend to steer away at a later point. To determine this it is natural to search for escape paths and estimate their severity.

The focus of this paper is to evaluate path planning methods that are suitable for both AEB and LKA systems. The methods shall be capable of handling a large variety of traffic scenarios (e.g. highway driving, city driving, curves and crossings) and different levels of maneuver severity (i.e. both fast emergency maneuvers and slow lane keeping maneuvers).

Three different path planning methods are evaluated: state lattice, predictive constraint-based planning and spline-based search tree. These methods were chosen from the large amount of path planning methods, since they were seen to provide fundamentally different approaches to the problem of creating a collision-free path. Each method is introduced in Section III. This section also describes the translation from theory to implementation, highlighting the general logic, the assumptions made and the parameters that characterize each method. In Section IV the implemented path generation methods are applied to a small set of evaluation scenarios, followed by a discussion on the results. In Section V, the results and properties of the methods are summarized. But first, the problem is defined further.

II. DEFINITIONS AND APPROACH

In this paper, path planning is defined as a collision avoidance problem, where the host vehicle shall not hit any of the objects and be kept within the boundaries of the road. To facilitate analysis and presentation, the host vehicle is assumed to have a constant longitudinal velocity throughout the maneuver. This means that the obstacles shall be avoided by steering intervention only.

The evaluation scenarios are designed to highlight two major attributes: The vehicle dynamics feasibility and the driver comfort of the path. The former is especially important when evaluating emergency maneuvers. It also indicates whether the path is suitable for lateral control and is measured by the maximum lateral acceleration and jerk (acceleration rate). The driver comfort is more relevant for lane keeping maneuvers and is analyzed from the maximum lateral jerk and, to some extent, acceleration.

Further aspects that are considered important are computational load, flexibility (e.g. ease of introducing new

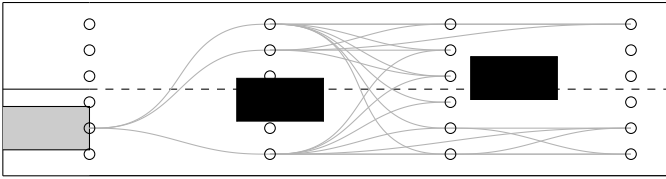


Fig. 1. State lattice overlays a regular grid on the road where the vertices are connected by cubic spirals subject to vehicle kinematics. This turns the path planning problem to a graph search problem.

scenarios or optimization criteria), ability to handle complex and dynamic situations and ease of tuning.

III. PATH PLANNING METHODS

A. STATE LATTICE

State lattices have recently appeared as a graph-based planning technique to take into account kinematic and dynamic constraints using a regular and repeatable sampling of state space [4], [5], [6]. These results come from the autonomous driving domain and tend to focus on unstructured environments. The sampled state space includes the position, orientation, and steering angle of a vehicle, and connects these in a local neighbourhood with edges that are – by construction – controllable trajectories. Given this representation, any graph-based planner can be used to find a trajectory that optimizes a given set of criteria. See Figure 1. For the application domain considered in this paper (structured on-road scenarios), the objective is to avoid collisions while minimizing acceleration and jerk levels.

Lattice edges can often be precomputed offline for real-time online planning with lookup-tables, even when computational resources are very limited. However, for structured environments with a highly varying dominant direction like streets in urban areas, discretization in heading angle can be problematic, because it can lead to oscillations between two orientation samples. Notable works which address this issue are [7], [8], [9]. A common solution is to use a Frénet frame attached to a reference curve such as the center of a lane. This results in state lattices that naturally follow curves, producing smooth and system compliant motion plans free of heading artefacts. Interestingly, heading discretization is not always necessary to achieve good paths [9]. The main drawback of bending a lattice is that system-compliant edges can no longer be easily precomputed and stored in a lookup table because bending a lattice will not result in a lattice, so the pre-calculated edges cannot be reused.

The implementation of state lattices presented here is based on the work of McNaughton [10]. The center of the lane serves as reference, specified by its curvature $\kappa(s)$ where s is the *station* (curvilinear distance along the road). The perpendicular distance to the reference is called *latitude* and written l . This results in a road-attached coordinate system denoted SL . In order to generate a lattice, nodes (s_i, l_i) are sampled at regular intervals Δs and Δl in SL coordinates. At each node, the vehicle is tangential to the reference curve, with a steering angle that corresponds to the curvature at that

point. Pairs of nodes are then connected with edges; in the simplest case a fully connected graph can be constructed (quadratic complexity), or a local neighbourhood pattern can be repeatedly applied each node (linear complexity). Edges are represented as the vehicle's steering curvature over curvilinear distance. Specifically, there exists a unique *cubic spiral*, i.e. a 3rd order polynomial $\kappa(s) = \sum_{n=0}^3 \kappa_n s^n$ that connects any (s_i, l_i) to another (s_j, l_j) . Higher-order polynomials could easily be used, to take into account additional optimality criteria such as continuity of higher-order derivatives of the curvature (see [10] and [11] for more details). Edges that would lead to collisions with other vehicles, as well as those that violate acceleration or jerk limits, are discarded from the graph. Then, each edge is annotated with its length. Graph search is used to find the shortest path from the current vehicle state to any of the nodes with the furthest *station*. In this paper we focus on constant longitudinal speed, but to accommodate the varying speed, different speed profiles are enumerated and added as a dimension in definition of each node in the lattice so the final trajectory contains the speeds in each path [10]. It would also be possible to optimize other path properties, such as minimizing acceleration, jerk, or a combination of length and jerk.

Notice that the lattice is completely characterized with three parameters: horizon length, and the number of increments in station and in latitude. The higher the number of nodes in the lattice, the closer to the optimal path the resulting trajectory will be. However, depending on the application the acceptable error margin can result in different resolutions.

B. PREDICTIVE CONSTRAINT-BASED PLANNING

The potential of the predictive control approach in autonomous/semi-autonomous applications has been investigated in several studies [12], [13], [14].

In [13] the problem of finding a steering maneuver that minimizes the steering effort for threat assessment in a collision situation using the model predictive control (MPC) approach is addressed. The steering effort in this work is defined in terms of the curvature, c , of the host vehicle trajectory and the maximum curvature and curvature rate that can be achieved are bounded to c_{max} and \dot{c}_{max} respectively. The steering maneuver is then described by a series of curvature rates, along the prediction horizon, initiated at fixed sample times t_j . The objective is to find the optimal series of \dot{c}_j while avoiding the all edges of the host vehicle to collide with the objects over the prediction horizon.

The work that is presented here in this paper is based on a work by Anderson [15]. In this work the idea is, rather than planning a certain path and controlling the vehicle along it, to design a human-like framework that is, to identify a safe field of travel - *Safe Corridor* - and convey the vehicle along that. The collision avoidance scenario is then described as a navigation problem and formulated by model predictive control (MPC) approach. In such a formulation, a constraint planner iteratively predicts, over a horizon, the

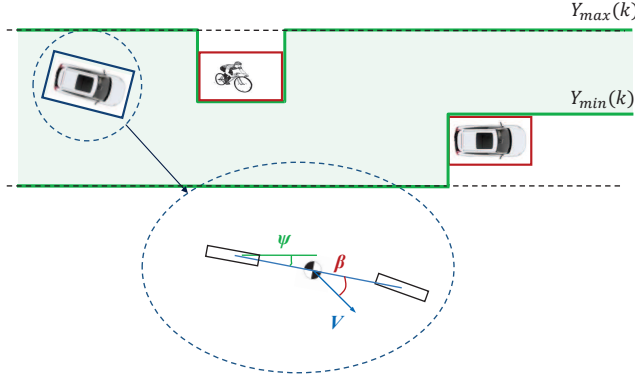


Fig. 2. Illustration of safe corridor and vehicle dynamics model to be used in MPC formulation.

Safe Corridor by use of sensor data and estimated behaviour of hazards and host vehicle. The safe corridor is presented as lateral position constraint vector and is used, together with the vehicle dynamics prediction model, to compute an optimal sequence of control inputs and the associated vehicle trajectory while optimizing the vehicle performance characteristics.

1) Problem Formulation:

The so called safe corridor is described by constraint vectors:

$$Y_{max}(k) = (y_{max}(k+1) \quad \dots \quad y_{max}(k+p))^T \quad (1a)$$

$$Y_{min}(k) = (y_{min}(k+1) \quad \dots \quad y_{min}(k+p))^T \quad (1b)$$

Where $y_{min}(k+j)$ and $y_{max}(k+j)$ represent the boundaries of the safe corridor at prediction sample j given the measurement at time k .

The vehicle model used in this work is a linear bicycle model, assuming linear tire characteristics and small angle approximation, which considers kinematics of the vehicle and also lateral and yaw dynamics. Figure 2 shows the safe corridor concept together with the vehicle dynamics model. The vehicle states, as described in (2) include the lateral position y , local vehicle lateral velocity v_y , yaw rate $\dot{\psi}$, yaw angle ψ and side slip angle β .

$$X = (y \quad v_y \quad \dot{\psi} \quad \psi \quad \beta)^T. \quad (2)$$

The MPC formulation is then given as below:

$$\begin{aligned} \min_{\Delta \mathcal{U}, \epsilon} J(x_t, \Delta \mathcal{U}, \epsilon) = & \sum_{i=1}^{H_p} \|\beta(t+i)\|_Q^2 \\ & + \sum_{i=0}^{H_c-1} \|\Delta u(t+i)\|_R^2 + \sum_{i=0}^{H_c-1} \|u(t+i)\|_S^2 + \rho \epsilon^2 \quad (3a) \end{aligned}$$

subject to

$$\begin{aligned} x_{k+1,t} &= \mathcal{A}x_{k,t} + \mathcal{B}u_{k,t}, \\ y_{k,t} &= h(x_{k,t}) \end{aligned} \quad (3b)$$

$$k = t, \dots, t + H_p - 1$$

$$u_{f,min} \leq u_{k,t} \leq u_{f,max}, \quad (3c)$$

$$\Delta u_{f,min} \leq \Delta u_{k,t} \leq \Delta u_{f,max}, \quad (3d)$$

$$y_{k,min} - \epsilon V_{k,min} \leq y_{k,t} \leq y_{k,max} - \epsilon V_{k,max} \quad (3e)$$

Where the equation (3a) is the objective function with the weighting matrices on state vector Q , control input R and control input rate S and ρ is the weight on the constraint violation variable ϵ . V represents variable constraint relaxation on vehicle lateral position. H_p and H_c are the prediction and control horizons respectively and equation (3b) shows the discrete vehicle model. The control input u is the front steering angle and inequalities (3c) and (3d) describe the bounds on the steering angle and steering angle rate. Inequality (3e) shows the constraint on the lateral position of the vehicle which is softened by ϵ . The total side slip angle β was chosen as the measure for characterizing the vehicle performance to be minimized in the object function.

C. SPLINE BASED SEARCH TREE

A fast multi-target threat assessment method was presented in [16]. The method uses a search tree to generate paths that tangent all combinations of objects, and is based on the assumption that the optimal path will be straight ahead, or touch at least one object. Each path segment from one object to the next is defined by a constant acceleration.

While effective for threat assessment, instantaneous changing of lateral acceleration and thus infinite jerk levels renders this method infeasible for vehicle control. Furthermore, the method only allows direction changes precisely when objects are passed which is not necessarily optimal.

The new solution proposed in this paper is similar to [16] in that it looks through the possible maneuvers using a search tree, but instead a smooth spline is used to describe the trajectory. Furthermore, the transition between spline segments are not fixed at the points where objects are passed. This gives a higher number of unknowns, the number of which is dependent on the number of objects and the boundary conditions chosen.

For driver comfort and feasibility of vehicle control, a minimized jerk level for the path is desired. A spline path can be formulated so it uses a minimum amount of jerk while satisfying the constraints imposed by the situation.

Figure 3 illustrates the spline-based search-tree method. The algorithm can be summarized as the following:

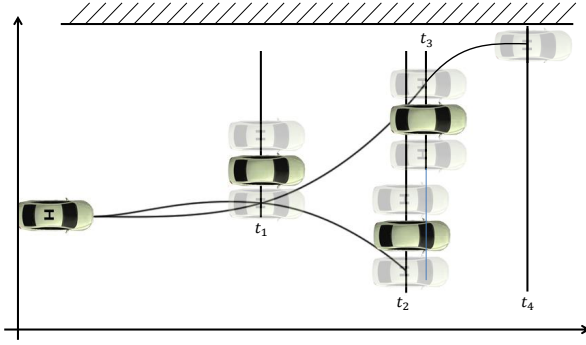


Fig. 3. Spline based method in scenario with objects and barrier.

- 1) Find all maneuvers that passes a single object to the left and the right.
- 2) Find maneuvers that consider two objects. All combinations of objects, ordered in time, are considered.
- 3) Find maneuvers that include three objects.
- 4) Carry on until the desired level of complexity has been reached.
- 5) Go through all candidate maneuvers and remove the ones with conflicts.
- 6) All remaining maneuvers are now valid escape paths. The optimal one is the one with the lowest jerk.

Two splines of different polynomial order were implemented for this paper, where the lateral position is represented by a cubic respectively quintic polynomial trajectory.

1) **CUBIC SPLINES:** The basis of the cubic spline is a piecewise constant 3rd derivative of the position, jerk, which has equal magnitude with alternating sign between segments. The 2nd derivative is then continuous and piecewise linear and thus, the position and its 1st derivative are continuous and differentiable across the entire spline. The equal magnitude of the 3rd derivative in the different segments corresponds to minimizing the jerk for the specified maneuver constraints assuming the jerk can be changed instantaneously.

The equations for segment i of the spline are:

$$\frac{d^3y}{dx} = (-1)^{i+1}k \quad (4)$$

$$\frac{d^2y}{dx} = (-1)^{i+1}kx + C_{i,1} \quad (5)$$

$$\frac{dy}{dx} = \frac{(-1)^{i+1}kx^2}{2} + C_{i,1}x + C_{i,2} \quad (6)$$

$$y(x) = \frac{(-1)^{i+1}kx^3}{6} + C_{i,1}\frac{x^2}{2} + C_{i,2}x + C_{i,3} \quad (7)$$

Where x and y denote the longitudinal and lateral position in the road-fixed coordinate system. Expressions for $C_{i,j}$ are derived from equality constraints in transitions between segments. Given maneuver constraints the transition points $x_{1,2,3,\dots}$ and the jerk level, k are determined.

2) **QUINTIC SPLINES:** To further smoothen the path, the discontinuities in jerk should be eliminated. The quintic

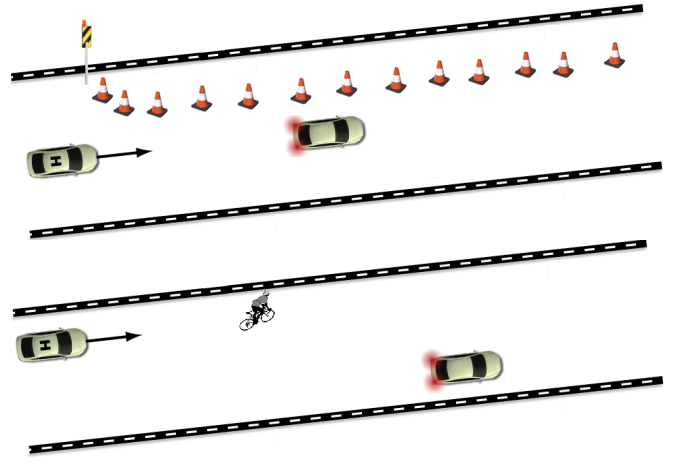


Fig. 4. Illustrations of highway driving (above) and city driving (below) evaluation scenarios.

spline will give a continuous lateral jerk curve resulting in a smoother path at the expense of a somewhat higher maximum jerk magnitude.

Since the goal is to keep the jerk level low, each segment employs the same magnitude maximum jerk, with alternating sign. Thus, a maneuver can be parametrized by

$$M = (J_{\max}, T) \quad \text{where} \quad T = \{t_1, t_2, t_3 \dots\} \quad (8)$$

where J_{\max} represents the peak magnitude of the jerk and t_i is the length of segment i . The jerk level along the path in a three segment spline can be expressed as:

$$J(t) = \begin{cases} -4J_{\max} \frac{t(t-t_1)}{t_1^2} & \text{if } t \leq t_1 \\ 4J_{\max} \frac{(t-t_2)(t-t_1)}{(t_2-t_1)^2} & \text{if } t_1 < t \leq t_2 \\ -4J_{\max} \frac{(t-t_2)(t-t_3)}{(t_2-t_3)^2} & \text{if } t_2 < t \end{cases}$$

Expressions for the lateral acceleration, lateral velocity and lateral position along the path can then be found by integrating this expression. The acceleration thereby becomes a cubic polynomial with zero derivative at the knots. A solution is found using equality constraints for the segment transitions and the boundary conditions that are implied by the scenario.

IV. RESULTS

The methods are compared over two evaluation scenarios, shown in Figure 4. Each scenario has a specified distance over which the simulations should extend. For the first scenario, the host vehicle is approaching a stationary vehicle at a velocity of 80km/h. This represents a two-lane highway scenario with one lane blocked off. The goal is for the host vehicle to maneuver around the obstacle to the right using the available space including the hard shoulder. The second scenario represents a city driving scenario, where the host vehicle must avoid both a bicyclist and, further away, a stationary vehicle. A safety margin is added to the bicyclist to accommodate unexpected swerving. The host vehicle speed is 50km/h. The simulation results for the methods are found in Figure 5 and Figure 6, displaying the lateral displacement, lateral acceleration and lateral jerk over traveled distance.

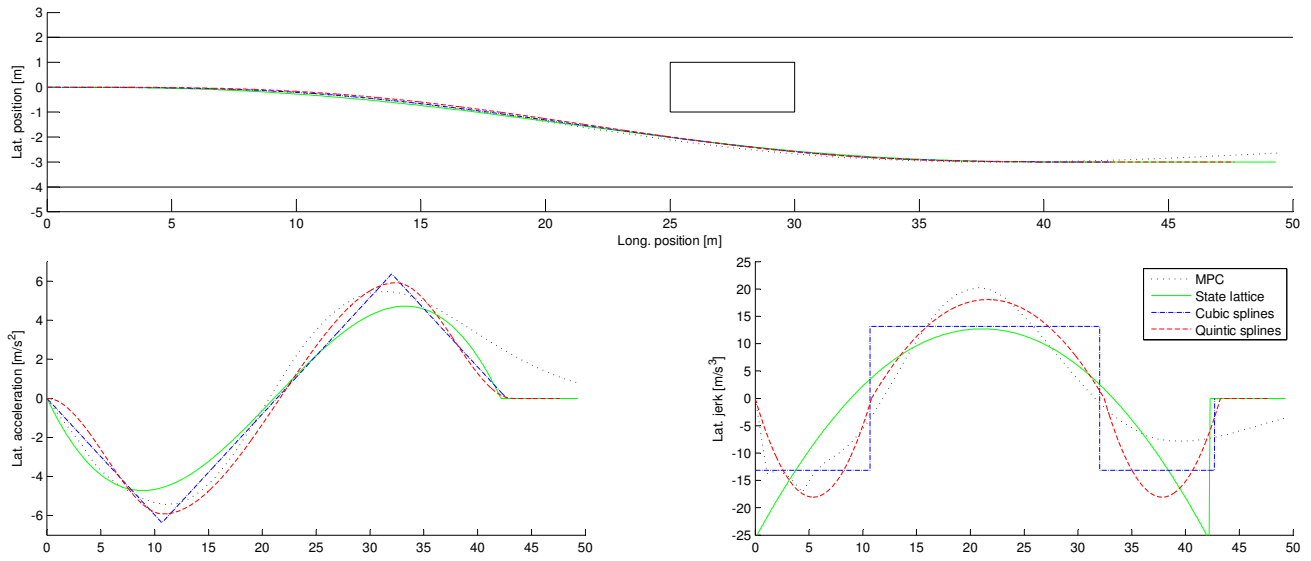


Fig. 5. Results for scenario 1, displaying lateral displacement (top), acceleration (left) and jerk (right).

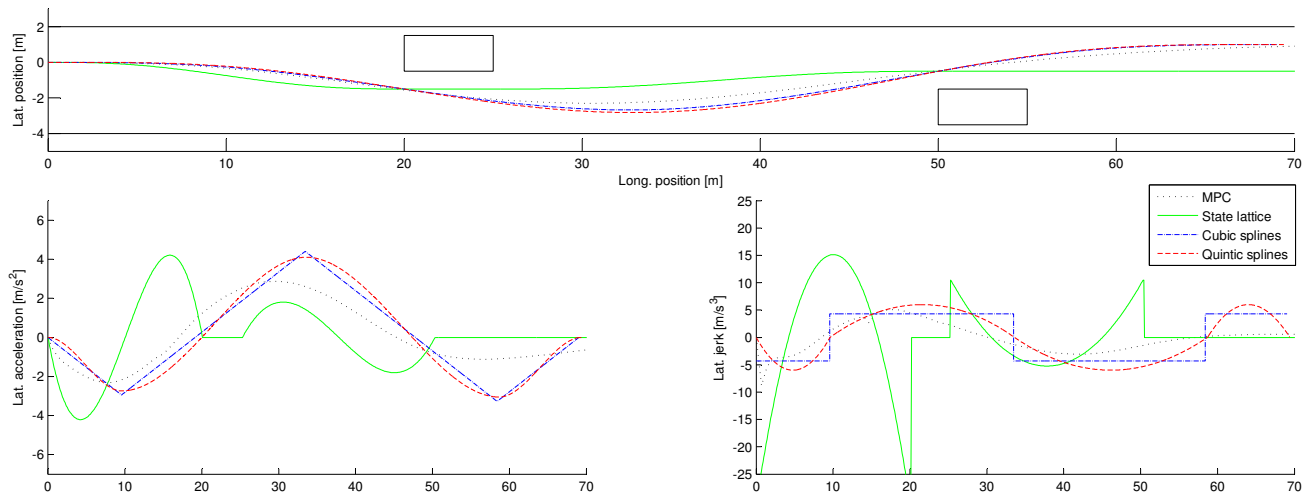


Fig. 6. Results for scenario 2, displaying lateral displacement (top), acceleration (left) and jerk (right).

	MPC with safe corridor	State Lattice	Spline-Based
Tunable Parameters	Prediction and control horizons. Weights on side slip angle, steering angle, steering angle rate and lateral position constraint violations.	Spatial resolution (2 dimensions). Prediction horizon.	Search depth. Polynomial order.
Simplifying Assumptions	No constraints on the lateral acceleration and jerk.	Curvature is a cubic polynomial, heading and curvature are zero at lattice nodes	Equal, alternating maximum jerk. Curve should tangent objects.
Model Fidelity	Bicycle kinematics including lateral and yaw dynamics.	Bicycle kinematics in a road-attached coordinate frame.	Direct expression of lateral motion.
Design Choices	Avoid high jerk maneuver by including relatively high cost on the steering angle rate in the objective function.	Make search tractable by limiting the spatial resolution.	Maximally employ jerk and acceleration limits.

TABLE I
SUMMARY OF THE DIFFERENCES BETWEEN THE STUDIED METHODS.

Looking at Figure 5, it is clear that the four trajectories align narrowly in the positional domain. The lateral jerk curves are more clearly distinguishable, implying that the paths can be close in effectiveness while providing different driver experiences. The state lattice path has the lowest maximum acceleration but on the other hand requires the largest amount of jerk. It goes without saying that it's not possible to achieve low lateral acceleration without high jerk, or vice versa, for a scenario as straight-forward as this. The preferable trade-off is a subject for the intended function design.

For Figure 6, larger deviations can be seen between the methods. The MPC method has the lowest lateral acceleration, but initially requires more jerk than the two spline-based methods. The state lattice method suffers in this case from the requirement that the host vehicle shall align with the road between curvature segments, causing a tight but jerky maneuver to be generated. The two spline-based methods are closely matched, providing consistent curves for acceleration and jerk, on expense of lateral deviation between the two obstacles.

A discussion on the less measurable attributes of the methods follows next. The main findings are summarized in Table I.

A. Output Domain vs Ease of Optimization

The output domains of the three methods are different: the state lattice produces a curvilinear path $(x(s), y(s))$, MPC produces time-parameterized controls $u(t)$, and the output of the spline methods is in the form $y(x)$. This reflects a design trade off between model fidelity and ease of optimization. The spline-based methods don't utilize a vehicle dynamics model, relying on either subsequent checks or the assumption that the path is followable. This allows the spline-based method to employ closed-form solutions for the optimal segments, but for high-curvature scenarios its trajectories will thus only be rough approximations. The state lattice uses a kinematic bicycle model, which requires more computational resources. The resulting paths thus properly model motions that follow curved roads. MPC adds lateral dynamics to the bicycle model, and it performs control signal optimization in the time domain. This produces stronger guarantees on the controllability of the computed maneuver (indeed obviating the need for a separate trajectory tracking loop, provided that MPC is run iteratively at each time step).

B. Resolution vs. Search Depth

MPC discretizes time, and state lattice discretizes space. Both optimize a discretized problem instead of a continuous one, and thus gain tractability. The spline-based method optimizes in continuous space, but has a combinatorial dependence on the number of obstacles.

C. Planning Completeness vs. Computational Complexity

The computational complexity of the three methods is affected in different ways by N_O , the number of obstacles. The state lattice places vertices (maneuver switching locations) in

a manner that is independent of N_O . The number of edges, however, is likely to *decrease* when there are more obstacles. Thus, its computational complexity does not grow with N_O (it may even run faster). But, there can be a risk that feasible paths are missed because of insufficient spatial resolution in vertex placement. The spline method creates a number of decision points that is proportional N_O . It then searches all possible avoidance combinations. Its complexity is thus exponential in N_O . However, contrary to the state lattice, there is no risk of missing a feasible path. For MPC, the safe corridor is determined by the obstacles, which is essentially a planning problem with a strong dependence on N_O . Notice that for a structured domain such as highway driving, the complexity of computing the safe corridor is only slightly influenced by N_O . Also notice that for MPC there are no switching points, it optimizes a single trajectory.

D. Solution Optimality vs. Situation Complexity

Intuitively, maneuvers that minimize the maximum jerk can be expected to be approximately divided into non-zero constant jerk segments. The spline-based methods have been designed to maximally employ acceleration and jerk limits and thus clearly exhibit this behavior. MPC, via its inclusion of Δu in the objective function, has a tendency to minimize the rate of steering angle change, which leads to a similar behavior. However, the state lattice implemented here deviates quite strongly from this expectation. It is designed to maximize the ability of handling complex situations, and ends up sacrificing solution optimality: acceleration and jerk are constrained to be zero at the start and end of each edge to allow straightforward lattice connectivity, but this pushes the overall path away from the global optimum.

E. Flexibility in Objective vs. Ease of Implementation

The spline-based method requires the equations and assumptions to be re-worked to accommodate a change in objective function (e.g. minimizing lateral acceleration). The benefit of this framework is that, once solved, the equations are easily implemented for simulation and production code generation. The MPC and state-lattice method are more versatile when it comes to the optimization objective. The constraints and objectives are directly plugged into the cost functions that will be optimized.

V. CONCLUSIONS

Three methods for generating collision-free paths have been compared: a state-lattice planner inspired by a method developed for autonomous driving, a model-predictive controller operating within the constraints of a safe corridor, and a spline-based tree search method that minimizes the maximum jerk by construction.

The spline-based method is computationally effective and predictable for short-horizon scenarios where each added object will affect the entire planned path. It is however specifically designed to minimize peak magnitude of the jerk and will need to be redesigned to take other properties into account. In addition, the search-tree methodology will

increase in complexity with the number of objects rendering this method inappropriate for scenarios with a high number of objects.

Given a reliable representation of the situation as a safe corridor, MPC provides a comprehensive take on the vehicle dynamics control problem. For complex scenarios, it is not straightforward to create a corridor or set of corridors which accurately describe the situation at hand. Both the optimization problem using a dynamic vehicle model and the corridor creation will contribute to the required computational power of the method.

The state lattice method, as opposed to the other two methods, is easily extended to handle high number of objects in complex scenarios. The planning precision is however a direct function of the lattice resolution. To increase the resolution while saving computation time, the lattice creation can utilize parallel computing to a high degree.

REFERENCES

- [1] E. Coelingh, A. Eidehall, and M. Bengtsson, "Collision warning with full auto brake and pedestrian detection - a practical example of automatic emergency braking," in *Proceedings of the IEEE Intelligent Transportation Systems*, Funchal, Portugal, 2010, pp. 155–160.
- [2] Daimler, "PRE-SAFE®-Brake," 2011. [Online]. Available: <http://www.daimler.com/dccom/0-5-1210220-1-1210348-1-0-0-0-0-1-8-7165-0-0-0-0-0-0-0.html>
- [3] Subaru, "Eyesight 2.0," 2011. [Online]. Available: <http://www.subaru.jp/legacy/eyesightsportselection/index2.html>
- [4] M. Pivtoraiko, R. A. Knepper, and A. Kelly, "Differentially constrained mobile robot motion planning in state lattices," *J. Field Robotics*, vol. 26, no. 3, pp. 308–333, 2009.
- [5] S. Pancanti, L. Pallottino, D. Salvadorini, and A. Bicchi, "Motion planning through symbols and lattices," in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, 2004.
- [6] A. Kushleyev and M. Likhachev, "Time-bounded lattice for efficient planning in dynamic environments," in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, 2009.
- [7] M. Werling, J. Ziegler, S. Kammel, and S. Thrun, "Optimal trajectory generation for dynamic street scenarios in a frenet frame," in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, 2010.
- [8] M. Ruflin and R. Siegwart, "On the design of deformable input- / state-lattice graphs," in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, 2010.
- [9] M. McNaughton, C. Urmson, J. M. Dolan, and J.-W. Lee, "Motion planning for autonomous driving with a conformal spatiotemporal lattice," in *2011 IEEE International Conference on Robotics and Automation*, vol. 1. IEEE, May 2011, pp. 4889–4895.
- [10] M. McNaughton, "Parallel Algorithms for Real-time Motion Planning," PhD, Carnegie Mellon University, 2011.
- [11] A. Kelly and B. Nagy, "Reactive Nonholonomic Trajectory Generation via Parametric Optimal Control," *The International Journal of Robotics Research*, vol. 22, no. 7-8, pp. 583–601, July 2003.
- [12] P. Falcone, M. Ali, and J. Sjöberg, "Predictive Threat Assessment via Reachability Analysis and Set Invariance Theory," *IEEE Transactions on Intelligent Transportation Systems*, vol. 12, no. 4, pp. 1352–1361, Dec. 2011. [Online]. Available: <http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=5963718>
- [13] M. Brannstrom, E. Coelingh, and J. Sjöberg, "Decision Making on when to Brake and when to Steer to Avoid a Collision," *First International Symposium on Future Active Safety Technology toward zero-traffic-accident, FASTzero11, JSAE*, Tokyo, Japan, 2011.
- [14] S. J. Anderson, S. B. Karumanchi, and K. Iagnemma, "Constraint-Based Planning and Control for Safe, Semi-Autonomous Operation of Vehicles," in *2012 Intelligent Vehicles Symposium*, vol. 1. IEEE, 2012, pp. 383–388.
- [15] S. J. Anderson, S. C. Peters, T. E. Pilutti, and K. Iagnemma, "An Optimal-Control-Based Framework for Trajectory Planning, Threat Assessment, and Semi-Autonomous Control of Passenger Vehicles in Hazard Avoidance Scenarios," *International Journal of Vehicle Autonomous Systems*, vol. 8, pp. 190–216, 2010.
- [16] A. Eidehall, "Multi-target threat assessment for automotive applications," in *Proceedings of the IEEE Intelligent Transportation Systems (ITSC)*, Washington D.C., USA, September 2011, pp. 433 – 438.