MITSUBISHI ELECTRIC RESEARCH LABORATORIES
http://www.merl.com

Path Planning and Integrated Collision Avoidance for Autonomous Vehicles

Berntorp, K.

TR2017-072 May 2017

Abstract

This paper discusses some of the current state-ofthe-art and remaining challenges in path planning and vehicle control for enabling autonomous vehicles. Reliable path planning is fundamental for the proper operation of an autonomous vehicle. Typically, the path planner relies on an incomplete model of the surroundings to generate a reference trajectory, used as input to a vehicle controller that tracks this reference trajectory. Depending on how much complexity is put into the path-planning block, the path planning and vehicle-control blocks can be viewed as independent of each other, connected to each other, or merged into one block. There are several types of path-planning techniques developed over the last decades, each with its own set of benefits and drawbacks. We review different techniques for path planning and trajectory generation, and give examples of its use in relation to autonomous vehicles. We report on our own recent findings and give an outlook on potential research directions.

American Control Conference (ACC)

Path Planning and Integrated Collision Avoidance for Autonomous Vehicles

Karl Berntorp[1]

***Abstract*— This paper discusses some of the current state-of-the-art and remaining challenges in path planning and vehicle control for enabling autonomous vehicles. Reliable path planning is fundamental for the proper operation of an autonomous vehicle. Typically, the path planner relies on an incomplete model of the surroundings to generate a reference trajectory, used as input to a vehicle controller that tracks this reference trajectory. Depending on how much complexity is put into the path-planning block, the path planning and vehicle-control blocks can be viewed as independent of each other, connected to each other, or merged into one block. There are several types of path-planning techniques developed over the last decades, each with its own set of benefits and drawbacks. We review different techniques for path planning and trajectory generation, and give examples of its use in relation to autonomous vehicles. We report on our own recent findings and give an outlook on potential research directions.**

I. INTRODUCTION

One of several key problems to solve before autonomous vehicles become a reality is how to determine a collision-free trajectory of the autonomous vehicle, when the sensor information provides an incomplete, and occasionally even wrong, picture of the surroundings. Fig. 1 provides a high-level system schematics for an autonomous vehicle. The sensing and mapping module uses various sensor information, such as Radar, Lidar, camera, and global positioning system (GPS) information, together with prior map information, to estimate the parts of the surroundings relevant to the driving scenario. The outputs from the sensing and mapping block provides the basis for the path planner, which produces a desired trajectory that the vehicle should follow. Autonomous vehicle control is commonly divided into trajectory generation (path planning) and trajectory tracking (vehicle control). However, the two modules should not be seen as isolated parts of the autonomous vehicle, but should ideally work together, and aid each other. Loosely speaking, given a system that is subject to a set of differential constraints (the dynamics), an initial state, a final state, a set of obstacles, a set of environmental constraints, and a goal region, the path-planning problem is to find a trajectory that drives the system from its initial state to the goal region. For general environments the path-planning problem is known to be computationally difficult [1], and there exist many different approaches for solving the path-planning problem. There has been extensive developments in graph-search methods, for example, A* [2], D* [3], [4], and D*-lite [5], with several applications to autonomy [6], [7]. The use of probabilistic planners based on rapidly-exploring random trees (RRTs) has been an active area of research over the last two decades [8]–[14], and has also been used successfully in autonomous vehicles [15]–[20]. RRT relies on random exploration of the state space. Randomized planners are widely employed because they provide a path whenever one exists, and optimal variants of the suboptimal RRT have been developed recently. Oftentimes structure and prior knowledge of the particular problem instance can be imposed. In such cases, deterministic trajectory generation based on optimal control can be used [21]–[23], although nonconvexity of the underlying planning problem is still oftentimes an issue.

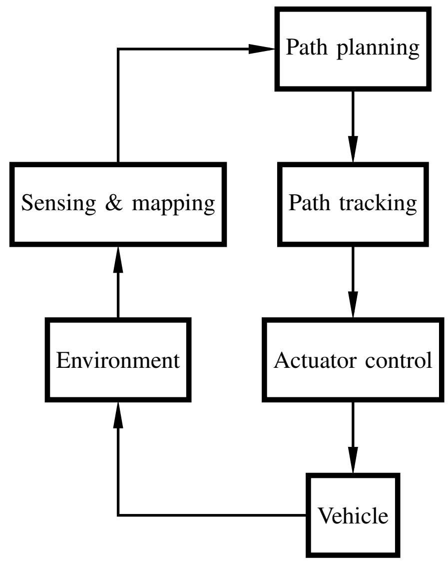


Fig. 1. A high-level system architecture of an autonomous vehicle.

For proper overall system performance, the path-planning and vehicle-control blocks should exchange information with each other. For example, the vehicle control should be able to closely track the path computed by the path planner. Hence, limitations on the type of paths that can be tracked should be transferred to the path planner. Similarly, depending on the type of path (e.g., lane change, overtaking, braking) the specific type of vehicle control that is being used may change and should be propagated to the vehicle-control module. That the environment is typically changing, also implies that both path planning and vehicle control need to consider prediction of the environment.

Autonomous vehicles must react to emergency situations, for example, emergency turning to avoid suddenly appearing

[1]Karl Berntorp is with Mitsubishi Electric Research Laboratories (MERL), 02139 Cambridge, MA, USA, Email:{karl.o.berntorp,dicairano}@ieee.org

pedestrians and/or emergency braking to avoid collision with rapidly approaching vehicles. This is traditionally handled in the vehicle control module. Methods based on constrained control, such as model-predictive control (MPC) [24]–[26] have been proven well suited for this problem, because of its ability to efficiently handle constraints and that the trajectory tracking problem is often convex, or at least well approximated by a convex problem.

In this paper, we review techniques related to integrating path planning and collision avoidance for autonomous vehicles, and talk about some of the main challenges associated with autonomous vehicles. We give examples from our own research and end with a research outlook.

II. VEHICLE MODELS

The detail level of the employed models vary with the intended purpose of the model. It is important that the path-planning module at some point considers the constraints imposed by the kinematics of the vehicle, which implies that the models need to consider dynamic feasibility. The vehicle control should also cover the interaction between road and vehicle, and in case of emergency maneuvers the models need to be complex enough to cover, or at least be able to predict, vehicle behavior in the nonlinear operating regime of the vehicle. The models can broadly be characterized into kinematic and dynamic models, respectively. The kinematic single-track model is a model that mimics the vehicle dynamics well under mild driving conditions where wheel slip can be neglected. Fig. 2 provides a schematic of the model. The dynamic equations are:

$$\dot{x} = \begin{bmatrix} \dot{p}_X \\ \dot{p}_Y \\ \dot{\psi} \\ \dot{v}_x \end{bmatrix} = f(x, u) = \begin{bmatrix} v_x \cos(\psi + \beta)/\cos(\beta) \\ v_x \sin(\psi + \beta)/\cos(\beta) \\ v_x \tan(\delta)/L \\ a_x \end{bmatrix},$$

where $\dot{\psi}$ is the yaw rate; v_x is the longitudinal velocity at the center of mass; l_f, l_r are the distances from the center of mass to the front and rear wheel base $L := l_f + l_r$, the body slip β is defined as

$$\beta = \arctan\left(\frac{l_r \tan(\delta)}{L}\right),$$

and where the inputs u are given by the steering angle δ and the acceleration a_x. The dynamic single-track model has the same kinematics as in Fig. 2 but the differential equations are instead given by [22]

$$\begin{aligned} \dot{v}_x - v_y\dot{\psi} &= \frac{1}{m}(F_{x,f}\cos(\delta) + F_{x,r} - F_{y,f}\sin(\delta)), \\ \dot{v}_y + v_x\dot{\psi} &= \frac{1}{m}(F_{y,f}\cos(\delta) + F_{y,r} + F_{x,f}\sin(\delta)), \\ I_{zz}\ddot{\psi} &= l_f F_{y,f}\cos(\delta) - l_r F_{y,r} + l_f F_{x,f}\sin(\delta), \end{aligned} \tag{1}$$

where m is the vehicle mass; I_{zz} is the vehicle inertia about the z-axis; and $\{F_{x,i}, F_{y,i}\}_{i=f,r}$ are the longitudinal and lateral tire forces acting at the front and rear wheels, respectively. The forces are in general modeled as nonlinear

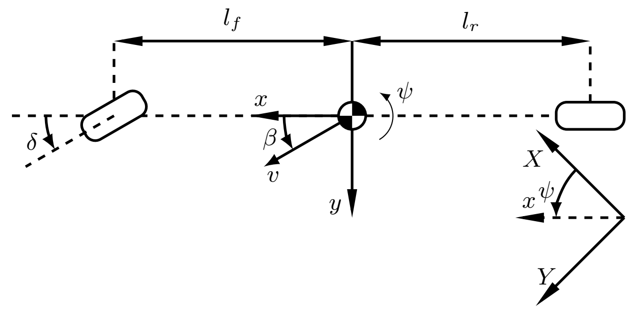


Fig. 2. Schematics of the single-track model. The capital letters denote the inertial frame.

functions of the wheel slip λ and wheel slip angle α, where the slip angles α_f, α_r are described by [27]

$$\alpha_f = \delta - \arctan\left(\frac{v_y + l_f\dot{\psi}}{v_x}\right), \tag{2}$$

$$\alpha_r = -\arctan\left(\frac{v_y - l_r\dot{\psi}}{v_x}\right). \tag{3}$$

For moderate driving, small-angle approximations and linear tire-force expressions can be used, resulting in a linear dynamic model of the vehicle.

For the path planning problem, the models provided here are sufficiently complex, but the vehicle control may, depending on how aggressive maneuvering should be captured, need to consider even more complex models. Considering even simpler models can be possible depending on how low-level the path plan should be. For instance, curvature-bounded point-mass models may be sufficient in certain scenarios. However, ideally the path planning and vehicle control modules should at least share common characteristics, to be able to provide guarantees on performance of the overall system.

III. PATH PLANNING

Let $\mathcal{X} \subseteq \mathbb{R}^n$, $\mathcal{Y} \subseteq \mathbb{R}^p$, and $\mathcal{U} \subseteq \mathbb{R}^m$, and assume that the vehicle dynamics can be expressed by a nonlinear differential equation on the form

$$\begin{aligned} \dot{x}(t) &= f(x(t), u(t)), x(0) = x_0 \\ y(t) &= h(x(t), u(t)), \end{aligned} \tag{4}$$

where the state $x(t) \in \mathcal{X}$, the ouput $y(t) \in \mathcal{Y}$, and control input $u(t) \in \mathcal{U}$. Further, let $\mathcal{X}_{\text{obs}}$ denote the, potentially dynamic, obstacle space, $\mathcal{X}_{\text{goal}}$ the goal region, and let $\mathcal{X}_{\text{free}} = \mathcal{X}/\mathcal{X}_{\text{obs}}$ denote the drivable space. A basic formulation of the path-planning problem we consider is to solve the following problem,

$$\begin{aligned} \underset{x(t)}{\text{minimize}} \quad & \int_0^T g(x(t), u(t))\,\mathrm{d}t \\ \text{subject to} \quad & x(t) \in \mathcal{X}_{\text{free}} \\ & \dot{x}(t) = f(x(t), u(t)), x(0) = x_0 \\ & u(t) \in \mathcal{U} \\ & x(T) \in \mathcal{X}_{\text{goal}}, \end{aligned} \tag{5}$$

with possibly free final time T. In autonomous-driving applications where the environment changes with time and the differential constraints of the vehicle need to be enforced, exact solutions to (5) are unrealistic. Instead, numerical solutions are sought that provide feasible solutions in short enough time. Because of the dynamic environment, (5) needs to be solved sequentially, either at a fixed update rate or whenever the sensing module provides an environment that does not match what was used in previous time steps.

Path-planning techniques can broadly be divided into variational methods, graph-search methods, and sampling-based methods. Variational, or optimal-control based, methods can be efficient when the problem size is sufficiently small such that computations of new solutions can be done fast, and/or when the environment is nearly static, as shown in the DARPA Urban Challenge [6]. However, oftentimes the non-convexity of the underlying nonlinear optimization problem can cause convergence to local minima, slow convergence, or even lack of convergence.

A. Determining the drivable space

An important part of a trajectory-generation system is to determine the drivable space $\mathcal{X}_{\mathrm{free}}$, or equivalently to determine the obstacle space $\mathcal{X}_{\mathrm{obs}}$. Determining the obstacle space essentially boils down to predicting the motion of the moving obstacles. Without motion predictions, the path planner has to assume a static, or perfectly known, environment, which is unrealistic because the environment is typically dynamic and highly uncertain. If precise prediction of the environment is available, path planners can compute safer, less conservative, and more robust trajectories. Motion prediction is a fairly well researched field, but there are still challenges remaining, such as how to seamlessly integrate the motion prediction with the path planner.

Motion prediction, which potentially includes driver-intention recognition, can be approached in several ways. For example, deterministic methods predict a single future trajectory, while stochastic methods represent the future trajectories with probability density functions (PDFs), which are estimated using statistical methods such as Monte-Carlo sampling [28]. Another common approach is to base the prediction on Markov chains [26], [29] for reachable set computations [30]. In [31], the reachable set computation was done by means of Monte Carlo and the driver-intention recognition was implicitly performed by a biased driver-preference distribution, while we have investigated how to predict individual vehicles by the notion of similarity with other vehicles [32].

B. Graph-Search Methods

Graph-search methods approach the path-planning problem by representing the state space $\mathcal{X}_{\mathrm{free}}$ as a discretized graph $\mathcal{G}$ of vertices (states) $\mathcal{V}$ and edges (paths/trajectories) $\mathcal{E}$ connecting the vertices. The initial state x_0 is a vertex of the graph and there is at least one vertex associated with the goal region $\mathcal{X}_{\mathrm{goal}}$. Graph-search methods have been rather commonly used in autonomous driving [6], [33], [34], especially as a high-level path planning method on road networks. There exist various methods to construct the graph as surveyed in [35], such as geometric methods and sampling-based methods [36]. Looking at the particular application of autonomous driving on structured road networks, it might also be feasible to discretize the preferred locations in each lane, for example, the middle lanes in each lane. By modeling the path-planning problem in the road-aligned frame, as is commonly done in vehicle control and estimation [31], [37], [38] the dimensionality of the graph can be heavily reduced, which is one reason why graph-based methods may be a preferred choice for path planning. Given a static graph, Dijkstra's algorithm [39] or A* [2] can be used for computing the shortest path. As in any path-planning method, however, the possibility of replanning with the path plan using the previous plan is important. Thus, when the environment changes, or perhaps more importantly, when the environment changes in a way that is inconsistent with the predictions of the drivable space, a new plan has to be computed. To this end, D* and its related methods are efficient replanning methods [4], [40].

When using graph-search methods for trajectory planning (i.e., path with time information), it becomes important to consider the differential constraints imposed by the vehicle dynamics/kinematics. Some work on this can be found in [41]. Traditional methods have been based on investigating approximate solutions (e.g., [42]), because in general exact solutions are infeasible to obtain. In this context, feedback motion planning [14], [43] can be useful because the edges are computed by feedback control, and are therefore dynamically feasible by construction. Feedback-based path planning may also have the potential to better integrate with vehicle control, because then the two modules share common theory. In addition, certain guarantees on robustness and invariance may also be possible.

C. Sampling-Based Methods

Sampling-based methods are focused toward incrementally build a feasible path, or a sequence of feasible paths that converge to an optimal path, given enough computation time. RRT is an important instance of sampling-based methods, which has found various applications [41]. RRT-type algorithms incrementally build a tree by selecting a random sample and expanding the tree towards that sample. Checking if the sample and the corresponding edge is in $\mathcal{X}_{\mathrm{free}}$ amounts to pointwise comparison. RRT does therefore naturally integrate with some of the methods for determing the drivable space, since it does not require a geometric expression for $\mathcal{X}_{\mathrm{free}}$, as opposed to some methods for constructing the graph in graph-search methods. RRT provides a path whenever one exists [44], but is suboptimal [45]. The quality of the path in RRT can vary heavily between any two planning instants, which is highly undesirable for autonomous driving. There exist various heuristic techniques to provide good performance [41], one of which was developed for the DARPA Urban challenge [15]. Optimal variants of RRT have recently been provided, such as RRT* [45], [46] and RRT$^{\#}$

[13], [47], under the assumption of known obstacles. RRT^X [48] is an extension that allows for optimality in uncertain environments. A difficulty in RRT-type algorithms is to allow for differential constraints, which are important to consider in vehicle applications, where the vehicle dynamics limits the drivable region. Work towards alleviating this problem can, for example, be found in [15], [17], [43], [49], [50]. One underlying assumption is the availability of a steering function that connects two vertices. However, this amounts to solving a two-point boundary value problem, which is computationally difficult to solve in general. Local linearization [51] or focusing on linear dynamics [49] simplify the solution to the boundary-value problem, and differential flatness has been utilized for autonomous high-speed driving [17].

Avoiding the need for exact steering is clearly desired for autonomous-driving applications. One approach toward this is the Stable sparse tree (SST) [52], which is based on generating random controls and propagating them through the dynamic model, whereas we in [50] developed an extension to the closed-loop prediction in [15], thus providing certain optimality without imposing a steering function. Feedback-based sampling methods seem like a fruitful approach for increasing robustness and computational efficiency. Using feedback-based planning bridges the gap between the path-planning and vehicle-control modules, and can potentially offer a more integrated approach to path planning and collision avoidance in autonomous driving.

IV. VEHICLE CONTROL AND COLLISION AVOIDANCE

Predictive control, specifically MPC [24], [25], has recently evolved as an important approach in the research literature for automotive control in general and vehicle-dynamics control in particular. In its general form, MPC admits nonlinear dynamics as in (4). MPC solves at each time step a finite-horizon optimal-control problem and applies the first of the optimal inputs to the system. There exist various formulations of the MPC problem, depending on the nature of the models, constraints, computational resources, and performance guarantees. A rather basic discrete-time formulation is

$$\begin{aligned} \underset{u(t)}{\text{minimize}} \quad & F(x(N_p)) + \sum_{k=0}^{N_p-1} L(x(k), y(k), u(k)) \\ \text{s.t.} \quad & x(k+1) = f(x(k), u(k)), \\ & x_{\min} \leq x(k) \leq x_{\max}, \quad k \in \{1, \dots, N_p\}, \\ & u_{\min} \leq u(k) \leq u_{\max}, \quad k \in \{0, \dots, N_c\}, \\ & x(0) = x_0, \end{aligned} \tag{6}$$

where N_p is the prediction horizon and N_c the control horizon. Today there exist efficient software and implementations for real-time solution of MPC, especially for convex MPC formulations. Furthermore, various MPC approaches have been proposed in relation to autonomous-driving applications. For example, [53] suggests MPC for trajectory generation of an active front steering vehicle. MPC can also be used in a hierarchical framework, where a high-level

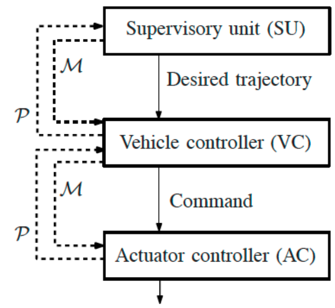


Fig. 3. Example of planning and control stack for driver assistance and automated driving features.

MPC provides trajectory generation to a low-level trajectory-tracking MPC [54]–[56]. Here, robust control invariant (RCI) sets can be exploited to provide the desired guarantess. For instance, [57] uses robust positive invariant sets to bound the maximum deviation of the trajectory from a nominal trajectory and provide collision avoidance, and [58] uses RCIs for vehicle stabilization.

There are also methods for unifying path planning and vehicle control/collision avoidance into one module, for example, by utilizing the structured environment of one-way roads [59]. Unifying the planning and control modules has potential, but typically also makes for a more complex and difficult problem to solve [60], [61]. Completely integrated path planning and vehicle control may therefore be out of reach in the general autonomous-driving scenario, but certainly has potential for several situations (e.g., lane-change assist).

V. EXAMPLE: INTEGRATING MPC WITH PATH PLANNING

For proper system performance, the different modules should exchange information with each other. Investigating Fig. 1 in a bit more detail in Fig. 3, we can interconnect the vehicle control with the actuator control and path planner, respectively. For instance, the actuator controller needs to guarantee that it can achieve the commands selected by the vehicle controller. The vehicle controller must guarantee a tracking error bound along the desired trajectory, so that the path planner can plan a robust trajectory accounting for such potential tracking error. In general the tracking error will depend on the reference trajectory, and hence the error bounds will depend on the classes of desired trajectories, according to "agreements" such as: "The vehicle controller will ensure performance measure $\mathcal{M}$ as long as the path planner generates reference trajectories satisfying property $\mathcal{P}$". In this example we investigate MPC for tracking control of a motion plan generated by a path planner, for example an RRT, that incorporates obstacle prediction. Hence, the explicit collision avoidance is done at the planning level with given robustness margins, whereas an MPC ensures that the generated reference trajectory is tracked with a preassigned tolerance.

When considering vehicle control through steering, the shape of the reference trajectory is related to the shape of the road. Road segments are often similar to clothoids [37], in which the curvature changes at a constant rate.The road is therefore well represented by a piecewise-clothoidal (PWCL) curve, and here we consider PWCL trajectories with bounded curvature *and* curvature rate of change. As a consequence, the trajectories are subject to state-dependent constraints, i.e., the currently allowed range of curvature rate depends on the current curvature.

In [38] we designed a steering controller for PWCL trajectories ensuring a preassigned bound on the tracking error. For PWCL trajectories, algorithms for maximal RCI sets generally results in nonconvex sets, which makes it hard to do real-time control for PWCL trajectories. Based on a recently developed method [62] for constructing convex RCI sets, in [38] we provided a guarantee of the vehicle controller performance measure ($\mathcal{M}$), when the trajectories generated by the path planner are in the class of trajectories satisfying the PWCL curvature and curvature rate bounds ($\mathcal{P}$). The method can be used to determine which curvature rate bounds are acceptable, and hence determines at design time the property $\mathcal{P}$ and the control algorithm that enforces the performance measure $\mathcal{M}$ in the "agreement" between the path planner and vehicle controller. Actuator constraints are accounted for, to avoid negative interactions between the vehicle and actuator control modules.

The reference trajectory from the path planner is modeled as generated by a particle moving at constant speed v_x along a curve with curvature $\kappa = 1/R_T$, R_T being the turn radius. This results in the model for the desired vehicle yaw, $\psi_{\rm des}$, and yaw rate, $\dot{\psi}_{\rm des}$,

$$\ddot{\psi}_{\rm des} = v_x \dot{\kappa}. \tag{7}$$

Sampling (7) with sampling period T_s yields

$$\dot{\psi}_{\rm des}(t+1) = \dot{\psi}_{\rm des}(t) + \gamma(t), \tag{8}$$

where $\gamma(t) = v_x T_s \dot{\kappa}(t)$ is an exogenous variable that describes the change of desired yaw rate. Eq. (8) models PWCL trajectories where the change of curvature is constant in time periods of length (at least) T_s. The model we use for vehicle control is described by (1)–(3), but with respect to a trajectory with yaw rate $\dot{\psi}_{\rm des}$. The state $x_e = [e_y\ \dot{e}_y\ e_\psi\ \dot{e}_\psi]'$, e_y, e_ψ is the lateral and yaw rate tracking errors, respectively, the input is steering angle $v_e = \delta$, the disturbance is $d = \dot{\psi}_{\rm des}$. For ensuring that the difference between the reference trajectory and the vehicle motion is bounded in a preassigned range, which allows the SU to plan with appropriate safety margins, we enforce the constraints

$$e_{y_{\min}} \leq e_y \leq e_{y_{\max}}. \tag{9}$$

Often, additional constraints have to be imposed on the vehicle system. In particular, the steering angle and angular rate are bounded, for both physical limitations and for safety reasons, as

$$\begin{aligned} \delta_{\min} \leq & v \leq \delta_{\max}, && \text{(10a)}\\ \Delta\delta_{\min} \leq & u \leq \Delta\delta_{\max}. && \text{(10b)} \end{aligned}$$

Further constraints can be imposed on the state variables such that together with (9) they result in the state bounds

$$x_{e\min} \leq x_e \leq x_{e\max}. \tag{11}$$

Thus, (10a), (11) describe the performance measure $\mathcal{M}$ that the vehicle control has to guarantee in the "agreement" with respect to the path planner.

A. Path Planning

The path planner can be of any type, as long as it can generate constrained PWCL reference trajectories according to the parameters determined determined together with the steering control law. We have previously designed [19], [20] an RRT-type path planner that instead of sampling the state-space (as in RRT) we sample the input space, which makes the method computationally efficient. We formulate the tree expansion in the RRT as a nonlinear, possibly multimodal, estimation problem, which we solve using particle filtering. Enforcing constraints on the PWCL trajectories amounts to constraining $\dot{\psi}_{\rm des}$ in the path planner, which is straightforward.

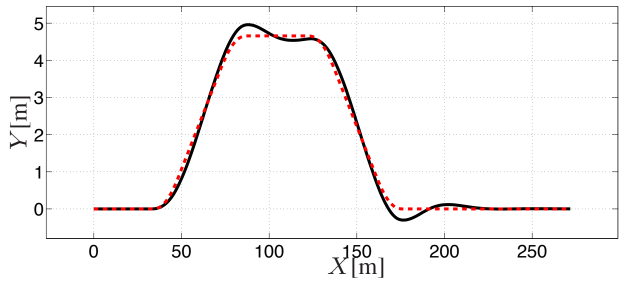
B. Results

We use as simulation vehicle a mid-size SUV on a dry asphalt road. The vehicle speed is 80km/h and the single-track model is discretized with sampling-time $T_s = 0.05$s. The bounds on steering and maximum tolerable lateral deviation are $\delta_{\min} = -\delta_{\min} = 0.165$[rad], $\Delta\delta_{\max} = -\Delta\delta_{\min} = 0.420$[rad/s], and $e_{y_{\max}} = -e_{y_{\min}} n = 0.3$m. For the values used in simulations, from $d_{\max}$ we obtain that the minimum turn radius at 80km/h is 44.44m. The simulation reported in Fig.4 shows the results for a double lane change with maximum rate of change $\gamma_{\max}$, $-\gamma_{\max}$ and where $N = 4$. Indeed, it is shown in Fig.4 that the tracking constraints, as well as the steering and steering rate constraints are satisfied, despite the short horizon.

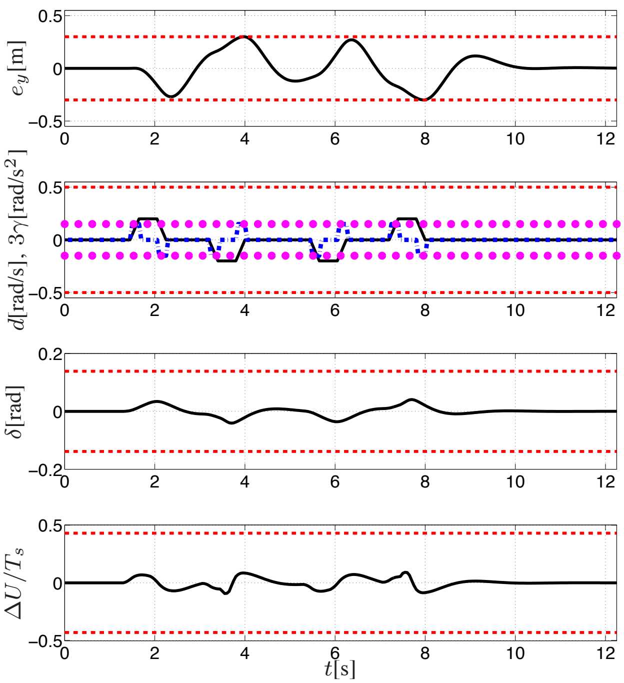
VI. CONCLUSION

There has been many advancements over the last decades for enabling truly autonomous systems. Over the next decade, the capabilities of production vehicles, eventually resulting in fully autonomous vehicles, is expected to increase as a result of advancements in algorithms, computing and sensing capabilities. Autonomous vehicles are complex decision-making systems and include a number of intricate subsystems, where the output of one system depends on the output of the previous. Thus, a research challenge is to integrate the different modules to ensure performance and safety of the vehicles. The performance of the complete system is limited by the weakest link in the chain, and a sophisticated vehicle controller alone cannot guarantee performance of the vehicle. Integrating MPC with path planning is one challenge. We presented one possible approach to this, but there are still things to be done.

Furthermore, achieving trajectory generation in the path planner that is robust to sensing and prediction errors is far from trivial. There are approaches targeting this problem, but



(a) Spatial path. Desired (red, dash), and actual (black, solid).



(b) Time trajectories of lateral error, desired yaw rate, steering, input (black, solid), and constraints (red, dash). Second plot from top: yaw rate change (blue, dash-dot) and constraints (magenta, dot) scaled by 3.

Fig. 4. Double lane change with maximum curvature rate of change.

challenges remain. In particular, there are several methods for determining the safe space, some based on statistical methods and other on geometric methods. Still, how to efficiently incorporate this at the path-planning stage is not a fully solved problem.

REFERENCES

[1] J. H. Reif, "Complexity of the movers problem and generalizations," in *Proc. IEEE Conf. Foundations of Computer Science*, 1979, pp. 421–427.

[2] N. J. Nilsson, *Principles of Artificial Intelligence*. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 1980.

[3] A. Stentz, "Optimal and efficient path planning for partially-known environments," in *IEEE Int. Conf. Robotics and Automation*, San Diego, CA, 1994.

[4] ——, "The focussed D* algorithm for real-time replanning," in *Proc. Int. Joint Conf. on Artificial Intelligence - Volume 2*, Montreal, Quebec, Canada, 1995, pp. 1652–1659.

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

[6] C. Urmson, J. Anhalt, D. Bagnell, C. Baker, R. Bittner, M. Clark, J. Dolan, D. Duggins, T. Galatali, C. Geyer, *et al.*, "Autonomous driving in urban environments: Boss and the urban challenge," *J. Field Robotics*, vol. 25, no. 8, pp. 425–466, 2008.

[7] M. Montemerlo, J. Becker, S. Bhat, H. Dahlkamp, D. Dolgov, S. Ettinger, D. Haehnel, T. Hilden, G. Hoffmann, B. Huhnke, *et al.*, "Junior: The stanford entry in the urban challenge," *J. Field Robotics*, vol. 25, no. 9, pp. 569–597, 2008.

[8] B. Donald, P. Xavier, J. Canny, and J. Reif, "Kinodynamic motion planning," *J. Association for Computing Machinery*, vol. 40, no. 5, pp. 1048–1066, November 1993.

[9] L. E. Kavraki and J.-C. Latombe, "Randomized preprocessing of configuration space for fast path planning," Dept. Computer Science, Stanford University, Stanford, CA, Tech. Rep. STAN-CS-93-1490, 1993.

[10] S. M. LaValle and J. J. Kuffner, "Rapidly-exploring random trees: Progress and prospects," in *Algorithmic and Computational Robotics: New Directions*, B. R. Donald, K. M. Lynch, and D. Rus, Eds. A. K. Peters, Wellesley, 2001, pp. 293–308.

[11] S. M. LaValle, *Planning Algorithms*. Cambridge, UK: Cambridge University Press, 2006.

[12] S. Karaman and E. Frazzoli, "Sampling-based algorithms for optimal motion planning," *Int. J. Robotics Research*, vol. 30, no. 7, pp. 846–894, 2011.

[13] O. Arslan and P. Tsiotras, "Use of relaxation methods in sampling-based algorithms for optimal motion planning," in *IEEE Int. Conf. Robotics and Automation*, Karlsrühe, Germany, May 6–10 2013, pp. 2413–2420.

[14] R. Tedrake, I. R. Manchester, M. Tobenkin, and J. W. Roberts, "Lqr-trees: Feedback motion planning via sums-of-squares verification," *Int. J. Robot. Res.*, vol. 29, no. 8, pp. 1038–1052, 2010.

[15] Y. Kuwata, S. Karaman, J. Teo, E. Frazzoli, J. How, and G. Fiore, "Real-time motion planning with applications to autonomous urban driving," *IEEE Trans. Control Syst. Technol.*, vol. 17, no. 5, pp. 1105–1118, 2009.

[16] E. Frazzoli, M. A. Dahleh, and E. Feron, "Real-time motion planning for agile autonomous vehicles," *J. Guidance, Control, and Dynamics*, vol. 25, no. 1, pp. 116–129, 2002.

[17] J. Hwan Jeon, R. V. Cowlagi, S. C. Peters, S. Karaman, E. Frazzoli, P. Tsiotras, and K. Iagnemma, "Optimal motion planning with the half-car dynamical model for autonomous high-speed driving," in *Am. Control Conf.*, Washington, DC, June 2013.

[18] Y. Kuwata, J. Teo, S. Karaman, G. Fiore, E. Frazzoli, and J. P. How, "Motion planning in complex environments using closed-loop prediction," in *AIAA Guidance, Navigation, and Control Conf.*, 2008.

[19] K. Berntorp and S. D. Cairano, "Particle filtering for online motion planning with task specifications," in *Amer. Control Conf.*, Boston, MA, July 2016.

[20] ——, "Joint decision making and motion planning for road vehicles using particle filtering," in *IFAC Symp. Advances in Automotive Control*, Kolmården, Sweden, June 2016.

[21] P. Dingle and L. Guzzella, "Optimal emergency maneuvers on highways for passenger vehicles with two- and four-wheel active steering," in *Amer. Control Conf.*, Baltimore, Maryland, 2010.

[22] K. Berntorp, B. Olofsson, K. Lundahl, and L. Nielsen, "Models and methodology for optimal trajectory generation in safety-critical road–vehicle manoeuvres," *Veh. Syst. Dyn.*, vol. 52, no. 10, pp. 1304–1332, 2014.

[23] S. Anderson, S. Peters, T. 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," *Int. J. Vehicle Autonomous Systems*, vol. 8, no. 2/3/4, pp. 190–216, 2010.

[24] C. E. Garcia, D. M. Prett, and M. Morari, "Model predictive control: theory and practice–a survey," *Automatica*, vol. 25, no. 3, pp. 335–348, 1989.

[25] D. Hrovat, S. Di Cairano, H. E. Tseng, and I. V. Kolmanovsky, "The development of model predictive control in automotive industry: A survey," in *IEEE Int. Conf. Control Applications*. IEEE, 2012, pp. 295–302.

[26] A. Carvalho, S. Lefévre, G. Schildbach, J. Kong, and F. Borrelli, "Automated driving: The role of forecasts and uncertainty—A control perspective," *European J. Control*, vol. 24, pp. 14–32, 2015.

[27] H. B. Pacejka, *Tire and Vehicle Dynamics*, 2nd ed. Oxford, United Kingdom: Butterworth-Heinemann, 2006.

[28] A. Broadhurst, S. Baker, and T. Kanade, "Monte Carlo road safety reasoning," in *IEEE Intell. Veh. Symp.* IEEE, 2005, pp. 319–324.

[29] S. Lefèvre, D. Vasquez, and C. Laugier, "A survey on motion prediction and risk assessment for intelligent vehicles," *Robomech Journal*, vol. 1, no. 1, p. 1, 2014.

[30] M. Althoff, O. Stursberg, and M. Buss, "Model-based probabilistic collision detection in autonomous driving," *IEEE Trans. Intell. Transp. Syst.*, vol. 10, no. 2, pp. 299–310, 2009.

[31] A. Eidehall and L. Petersson, "Statistical threat assessment for general road scenes using Monte Carlo sampling," *IEEE Trans. Intell. Transp. Syst.*, vol. 9, no. 1, pp. 137–147, Mar. 2008.

[32] K. Okamoto, K. Berntorp, and S. Di Cairano, "Similarity-based vehicle-motion prediction," in *Amer. Control Conf.*, Seattle, WA, May 2017.

[33] A. Bacha, C. Bauman, R. Faruque, M. Fleming, C. Terwelp, C. Reinholtz, D. Hong, A. Wicks, T. Alberi, D. Anderson, *et al.*, "Odin: Team victorTango's entry in the DARPA Urban challenge," *Journal of Field Robotics*, vol. 25, no. 8, pp. 467–492, 2008.

[34] D. Dolgov, S. Thrun, M. Montemerlo, and J. Diebel, "Practical search techniques in path planning for autonomous driving," *Ann Arbor*, vol. 1001, p. 48105, 2008.

[35] B. Paden, M. p, S. Z. Yong, D. Yershov, and E. Frazzoli, "A survey of motion planning and control techniques for self-driving urban vehicles," *IEEE Trans. Intell. Veh.*, vol. 1, no. 1, pp. 33–55, Mar. 2016.

[36] L. E. Kavraki, P. Švestka, J.-C. Latombe, and M. H. Overmars, "Probabilistic roadmaps for path planning in high-dimensional configuration spaces," *IEEE Trans. Robot. Autom.*, vol. 12, no. 4, pp. 566–580, 1996.

[37] R. Rajamani, *Vehicle Dynamics and Control*. Berlin Heidelberg: Springer-Verlag, 2006.

[38] S. D. Cairano, U. V. Kalabi, and K. Berntorp, "Vehicle tracking control on piecewise-clothoidal trajectories by MPC with guaranteed error bounds," in *IEEE Conf. Decision and Control*, Las Vegas, NV, Dec. 2016.

[39] E. W. Dijkstra, "A note on two problems in connexion with graphs," *Numerische mathematik*, vol. 1, no. 1, pp. 269–271, 1959.

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

[41] S. M. LaValle, *Planning Algorithms*. Cambridge University Press, 2006.

[42] P. Jacobs and J. Canny, "Planning smooth paths for mobile robots," in *Nonholonomic Motion Planning*. Springer, 1993, pp. 271–342.

[43] C. Danielson, A. Weiss, K. Berntorp, and S. D. Cairano, "Path planning using positive invariant sets," in *IEEE Conf. Decision and Control*, Las Vegas, NV, Dec. 2016.

[44] S. M. LaValle and J. J. Kuffner, Jr., "Randomized kinodynamic planning," *Int. J. Robotics Research*, vol. 20, no. 5, pp. 378–400, May 2001.

[45] S. Karaman and E. Frazzoli, "Sampling-based algorithms for optimal motion planning," *Int. J. Robotics Research*, vol. 30, no. 7, pp. 846–894, 2011.

[46] S. Karaman, M. R. Walter, A. Perez, E. Frazzoli, and S. Teller, "Anytime motion planning using the RRT*," in *IEEE Int. Conf. Robotics and Automation*, Shanghai, China, May 2011.

[47] O. Arslan and P. Tsiotras, "Dynamic programming guided exploration for sampling-based motion planning algorithms," in *IEEE Int. Conf. Robotics and Automation*, Seattle, Washington, May 26–30 2015.

[48] M. Otte and E. Frazzoli, "RRT^X: Real-time motion planning/replanning for environments with unpredictable obstacles," in *Algorithmic Foundations of Robotics XI*. Springer, 2015, pp. 461–478.

[49] D. J. Webb and J. van den Berg, "Kinodynamic rrt*: Asymptotically optimal motion planning for robots with linear dynamics," in *IEEE Int. Conf. Robotics and Automation*. IEEE, 2013, pp. 5054–5061.

[50] O. Arslan, K. Berntorp, and P. Tsiotras, "Sampling-based Algorithms for Optimal Motion Planning Using Closed-loop Prediction," *ArXiv e-print: 1601.06326*, Jan. 2016.

[51] A. Perez, R. Platt, G. Konidaris, L. Kaelbling, and T. Lozano-Perez, "LQR-RRT*: Optimal sampling-based motion planning with automatically derived extension heuristics," in *IEEE Int. Conf. Robotics and Automation*. IEEE, 2012, pp. 2537–2542.

[52] Y. Li, Z. Littlefield, and K. E. Bekris, "Sparse methods for efficient asymptotically optimal kinodynamic planning," in *Algorithmic foundations of robotics XI*. Springer, 2015, pp. 263–282.

[53] P. Falcone, F. Borrelli, J. Asgari, H. E. Tseng, and D. Hrovat, "Predictive active steering control for autonomous vehicle systems," *IEEE Trans. Control Syst. Technol.*, vol. 15, no. 3, pp. 566–580, 2007.

[54] P. Falcone, F. Borrelli, H. E. Tseng, J. Asgari, and D. Hrovat, "A hierarchical model predictive control framework for autonomous ground vehicles," in *American Control Conference, 2008*. IEEE, 2008, pp. 3719–3724.

[55] Y. Gao, T. Lin, F. Borrelli, E. Tseng, and D. Hrovat, "Predictive control of autonomous ground vehicles with obstacle avoidance on slippery roads," in *ASME 2010 dynamic systems and control conference*. American Society of Mechanical Engineers, 2010, pp. 265–272.

[56] K. Berntorp and F. Magnusson, "Hierarchical predictive control for ground-vehicle maneuvering," in *Amer. Control Conf.*, Chicago, Il, July 2015.

[57] Y. Gao, A. Gray, H. E. Tseng, and F. Borrelli, "A tube-based robust nonlinear predictive control approach to semiautonomous ground vehicles," *Veh. Syst. Dyn.*, vol. 52, no. 6, pp. 802–823, 2014.

[58] G. Palmieri, M. Barić, L. Glielmo, and F. Borrelli, "Robust vehicle lateral stabilisation via set-based methods for uncertain piecewise affine systems," *Veh. Syst. Dyn.*, vol. 50, no. 6, pp. 861–882, 2012.

[59] J. Nilsson, P. Falcone, M. Ali, and J. Sjöberg, "Receding horizon maneuver generation for automated highway driving," *Control Engineering Practice*, vol. 41, pp. 124–133, 2015.

[60] J. Funke, M. Brown, S. M. Erlien, and J. C. Gerdes, "Collision avoidance and stabilization for autonomous vehicles in emergency scenarios," *IEEE Trans. Control Syst. Technol.*, vol. PP, no. 99, pp. 1–13, 2016.

[61] V. Turri, A. Carvalho, H. E. Tseng, K. H. Johansson, and F. Borrelli, "Linear model predictive control for lane keeping and obstacle avoidance on low curvature roads," in *IEEE Int. Conf. Intell. Transp. Syst.*, The Hague, Netherlands, 2013.

[62] S. Di Cairano and F. Borrelli, "Reference tracking with guaranteed error bound for constrained linear systems," *IEEE Transactions on Automatic Control*, vol. 61, no. 8, pp. 2245–2250, 2016.