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

Integrated Obstacle Detection and Avoidance in Motion Planning and Predictive Control of Autonomous Vehicles

Quirynen, Rien; Berntorp, Karl; Kambam, Karthik; Di Cairano, Stefano

TR2020-096 July 03, 2020

Abstract

This paper presents a novel approach for obstacle avoidance in autonomous driving systems, based on a hierarchical software architecture that involves both a lowrate, long-term motion planning algorithm and a high-rate, highly reactive predictive controller. More specifically, an integrated framework of a particle-filter based motion planner is proposed in combination with a trajectory-tracking algorithm using nonlinear model predictive control (NMPC). The motion planner computes a reference trajectory to be tracked, and its corresponding covariance is used for automatically tuning the time-varying tracking cost in the NMPC problem formulation. Preliminary experimental results, based on a test platform of small-scale autonomous vehicles, illustrate that the propose approach can enable safe obstacle avoidance and reliable driving behavior in relatively complex scenarios.

American Control Conference (ACC) 2020

Mitsubishi Electric Research Laboratories, Inc.
201 Broadway, Cambridge, Massachusetts 02139

Integrated Obstacle Detection and Avoidance in Motion Planning and Predictive Control of Autonomous Vehicles

Rien Quirynen[1], Karl Berntorp[1], Karthik Kambam[1], Stefano Di Cairano[1]

Abstract— **This paper presents a novel approach for obstacle avoidance in autonomous driving systems, based on a hierarchical software architecture that involves both a low-rate, long-term motion planning algorithm and a high-rate, highly reactive predictive controller. More specifically, an integrated framework of a particle-filter based motion planner is proposed in combination with a trajectory-tracking algorithm using nonlinear model predictive control (NMPC). The motion planner computes a reference trajectory to be tracked, and its corresponding covariance is used for automatically tuning the time-varying tracking cost in the NMPC problem formulation. Preliminary experimental results, based on a test platform of small-scale autonomous vehicles, illustrate that the proposed approach can enable safe obstacle avoidance and reliable driving behavior in relatively complex scenarios.**

I. INTRODUCTION

Autonomous vehicles are complex decision-making systems that require the integration of advanced and interconnected sensing and control components. At the highest level, a sequence of destinations is computed through the road network by a route planner. A discrete decision-making layer is responsible for determining the local driving goal of the vehicle. Each decision could be any of turn right, stay in lane, turn left, or come to full stop in a particular lane at an intersection. A 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.

An implementation of high-level mapping and decision functionalities is not discussed further in the present paper, but more information can be found, e.g., in [1]. Here, we focus on the integration of the motion planning algorithm and the vehicle controller, see Figure 1a. The motion planner is responsible for determining a safe, desirable and dynamically feasible trajectory that the vehicle should follow based on the outputs from the sensing and mapping module. A vehicle control algorithm then aims to track this reference motion, at a relatively high sampling frequency, by issuing commands, e.g., steering angle, wheel torque and brake force. Finally, an actuator control layer regulates the actuators to achieve these requested commands.

The motion-planning problem in autonomous vehicles shares many similarities with the standard robotics setup [2], and optimal solutions are in most cases intractable due to non-convexity of the problem. In order to deal with this non-convexity, motion planning is often performed using sampling-based methods such as rapidly-exploring random trees (RRTs) [2], or graph-search methods such as A*, D* and other variations [3]. In previous work [4], [5], we developed and demonstrated a probabilistic method for motion planning, using particle filtering for approximating the involved probability density functions (PDFs). The driving requirements, such as staying on the road, right-hand traffic, and obstacle avoidance, are formulated as measurements for the nonlinear filtering problem.

Any technique can be used in the vehicle control layer to track the reference motion that is computed by the motion planner. However, as discussed also in [6], the particular future information in the long-term motion plan can be effectively used in a model predictive control (MPC) algorithm for reference tracking. Many recent publications have shown the potential benefits of using nonlinear MPC (NMPC) in autonomous driving systems, e.g. [7], [8] based on simulation results and [9], [10] based on real-world experiments. A popular approach to implement NMPC in a computationally efficient manner, is based on the real-time iteration (RTI) scheme [11], [12], which typically combines a direct multiple shooting type optimal control discretization with an online variant of sequential quadratic programming (SQP).

In this paper, we propose a tailored integration between the particle-filtering based motion planner and NMPC-based vehicle control layer in order to enable a reasonable sharing in the burden of ensuring safe obstacle avoidance and reliable driving behavior in relatively complex scenarios. The planning algorithm computes a reference trajectory to be tracked, and its corresponding covariance matrices are used for automatically tuning the time-varying trade off in the cost function of the NMPC problem formulation. In fact, the covariance associated to the trajectory indicates how much the path planner believes that its computed trajectory is effective. We aim to allow more deviations when such belief is low, and less deviations when such belief is relatively high. We experimentally validate our approach using a test platform of small-scale autonomous vehicles [13], in which the ego vehicle performs advanced in-lane swaying maneuvers for safe and reliable obstacle avoidance.

The paper is organized as follows. Section II introduces the overall problem formulation and modeling choices. Sections III and IV briefly summarize the particle filtering based motion planner and NMPC based tracking algorithm, and their implementation of obstacle avoidance requirements. The automatic tuning of the reference tracking cost is described in Section V, followed by the experimental validation in Section VI. Section VII concludes the paper.

[1]Control and Dynamical Systems, Mitsubishi Electric Research Laboratories, Cambridge, MA, 02139, USA. quirynen@merl.com

II. MODELING AND EXPERIMENTAL SETUP

Let us briefly describe the vehicle dynamics and problem formulation, before presenting our proposed algorithms and the corresponding experimental results in the next sections.

A. Dynamic Model for Vehicle Planning and Control

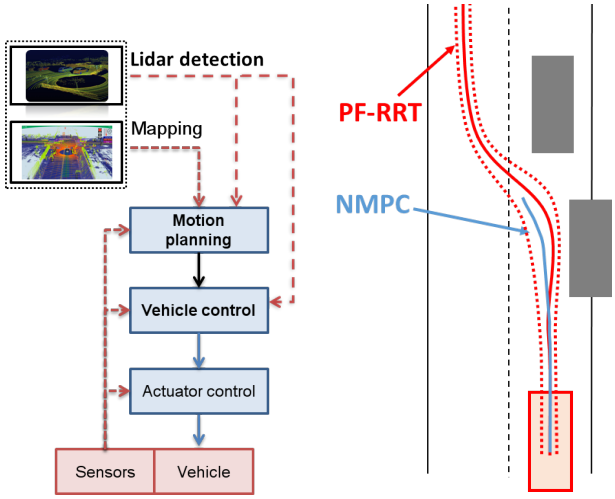
Under the assumption of normal driving conditions, i.e., not at-the-limit maneuvers, the modeling can be based on a single-track model in which the two wheels on each axle are lumped together. Although a dynamic vehicle model based on force-torque balances is generally more accurate than a kinematic model, differences are small for regular driving [14], and model errors are corrected for by feedback action of the controller. Furthermore, a dynamic model depends on several more states and parameters, such as wheel radii, tire stiffness, vehicle mass and inertia, slip angles and slip ratios, that are challenging to precisely estimate in real-time with automotive grade sensors. Hence, for simplicity, we use the kinematic single-track model

$$\dot{x} = \begin{bmatrix} \dot{p}_\mathrm{X} \\ \dot{p}_\mathrm{Y} \\ \dot{\psi} \\ \dot{\delta}_\mathrm{f} \\ \dot{v}_\mathrm{x} \\ \dot{\delta} \end{bmatrix} = \begin{bmatrix} v_\mathrm{x}\cos(\psi+\beta) \\ v_\mathrm{x}\sin(\psi+\beta) \\ \frac{v_\mathrm{x}}{L}\tan(\delta_\mathrm{f})\cos(\beta) \\ 1/t_\mathrm{d}(\delta+\delta_0-\delta_\mathrm{f}) \\ u_1 \\ u_2 \end{bmatrix}, \tag{1}$$

where $p_\mathrm{X}, p_\mathrm{Y}$ is the longitudinal and lateral position in the world frame, ψ is the heading angle and $\dot{\psi}$ the heading rate of the vehicle, v_x is the longitudinal velocity of the vehicle, δ and δ_f are, respectively, the desired and actual front wheel steering angle, $L := l_f + l_r$ is the wheel base, and $\beta := \arctan\left(l_r \tan(\delta_\mathrm{f})/L\right)$ is the body-slip angle. A first order front steering equation is included in Eq. (1) which models that, due to the steering mechanism, the wheel angle response is not immediate. In addition, we estimate the offset value δ_0 for the steering angle online. The inputs u_1, u_2 are the acceleration and steering rate, respectively. This choice of control inputs results in smooth velocity and steering profiles and allows constraining the rate of change of the vehicle velocity and front steering wheel angle.

B. Problem Formulation and Research Contributions

In prior work [5], [13], we demonstrated how a particle-filtering based motion planner and an NMPC algorithm for trajectory tracking can be integrated successfully into a control and estimation software architecture for autonomous driving capabilities. The present paper aims to improve the integration of the motion planning and vehicle control layer of Figure 1a, by leveraging implementation details from both the motion planning and vehicle control layer. In particular, our motion planning produces both a first and a second moment of the statistics for the trajectory. We use the first moment as reference for the controller, similar to [5], [13], and the second moment to determine the uncertainty of such reference. Thus, if the second moment shows a small uncertainty, the trajectory must be tracked more closely than if such uncertainty is large (see Figure 1b).



(a) Hierarchical autonomous driving software that involves mapping, sensing, planning, decision making, estimation and control.

(b) Motion planning algorithm computes covariance (dashed line) for automatic tuning of tracking cost in NMPC controller.

Fig. 1. Integrated software design for autonomous driving systems.

We exploit this to enable an automatic tuning mechanism in the NMPC controller for trading off competing objectives, e.g., achieving high tracking performance while satisfying safe obstacle avoidance requirements.

The effectiveness of our proposed approach is illustrated in Section VI based on experimental results in a scenario where the autonomous vehicle needs to perform an in-lane swaying maneuver for obstacle avoidance. The latter can be necessary when it is impossible to safely stay in the middle of the current lane, because of either static or dynamic obstacles on the side of that lane (e.g., bicycles or parked cars), and impossible or undesirable to change to another lane, e.g., when the other lanes may be currently blocked. We first describe the particle-filtering based motion planner and the NMPC tracking algorithm, including their respective approaches to deal with obstacle avoidance constraints.

III. OBSTACLE AVOIDANCE IN MOTION PLANNING WITH PARTICLE FILTERING

In [4], we developed a motion planner based on particle filtering for generating complex and dynamically feasible trajectories. Particle filtering is a sampling-based technique for solving the nonlinear filtering problem. Based on a probabilistic system model, the particle filter numerically approximates the PDF of the variables of interest given the measurement history, by generating N random trajectories and assigning a weight q^i to each trajectory i according to how well it predicts the observations. The planner relies on the fact that driving requirements, such as staying on the road, right-hand traffic, and avoid obstacles, are known ahead of planning. The driving requirements are modeled as measurements generated by a system subject to uncertainty. An interpretation is that the particle-filter based motion planner finds the trajectory that best achieves a trade off of the driving requirements, when there is both modeling and

sensing uncertainty. In each planning phase, the particle filter approximates the joint PDF of the state trajectory conditioned on the decision and driving requirements.

The motion planner considers the following discrete-time model for the ego vehicle (EV)

$$x_{k+1} = f(x_k) + g(x_k)u_k, \tag{2}$$

with state $x_k \in \mathbb{R}^{n_x}$ and input $u_k \in \mathbb{R}^{n_u}$, where k is the time index corresponding to time t_k. This includes many of the common vehicle models, such as the kinematic single-track model in (1) or more advanced dynamic models [4]. In our approach, the input vector u_k is modeled as an input disturbance distributed according to p_u. The EV driving requirements y_k are modeled by

$$y_k = h(x_k) + e_k, \tag{3}$$

where $h(\cdot)$ is a function relating the driving requirements to the vehicle state. The term e_k is regarded as a sample from some noise distribution p_e, $e_k \sim p_e(x)$, where $p_e(\cdot)$ is in general state dependent, and the interpretation is that p_e is the tolerated deviation from the requirements considering the uncertainties. Possible driving requirements can include a nominal velocity profile, mid-lane tracking and ensuring a safety distance to obstacle vehicles (OV) [4].

In a Bayesian setting, (2) and (3) can be formulated as

$$x_{k+1} \sim p(x_{k+1}|x_k), \qquad y_k \sim p(y_k|x_k), \tag{4}$$

where x_{k+1} and y_k are regarded as samples. Modeling the vehicle dynamics and driving requirements probabilistically may be important for several reasons, e.g., to account for uncertainties from state and map estimation algorithms, and to avoid infeasibility in trying to fulfill all driving requirements exactly. By considering the planning problem in a probabilistic framework, we naturally integrate the inherent uncertainties into the motion-planning problem. In this paper, we model the input disturbance u_k and tolerated deviation from driving requirements e_k as zero-mean Gaussian distributed, that is, $u_k \sim p_u = \mathcal{N}(0, Q_k)$ and $e_k \sim p_e = \mathcal{N}(0, R_k)$. Using the Bayesian formulation (4), the motion planner [4] constructs the posterior density function $p(x_{0:T}|y_{0:T})$ of the state trajectory over the T planning steps, given the driving requirements $y_{0:T}$ as a weighted set of N state trajectories,

$$p(x_{0:T}|y_{0:T}) \approx \sum_{i=1}^{N} q_T^i \delta(x_{0:T} - x_{0:T}^i), \tag{5}$$

where q_T^i is the importance weight for the ith particle and $\delta(\cdot)$ is the Dirac delta mass. From the density function (5), the motion plan can be extracted, for instance, as the minimum mean-square estimate

$$x_{0:T} \approx \frac{1}{N} \sum_{i=1}^{N} q_T^i x_{0:T}^i. \tag{6}$$

The motion planner in [4] naturally integrates safe obstacle avoidance. By leveraging a tailored proposal density in the particle filter, the particles will tend to obey the driving requirements, and if Q_k and R_k are chosen wisely, the risk of generating a trajectory that will lead to collision is kept to a minimum. The reason is that the weights q_k^i corresponding to the particles that violate, or are close to violating, the constraints, will become small and therefore will be rejected in the resampling procedure of the particle filter.

Since the particle-filter based motion planner computes the PDF of the state trajectory, higher-order moments than the weighted mean (6) can be determined. For instance, by computing the covariance (i.e., the second moment) along the weighted mean, we can determine a Gaussian approximation $\mathcal{N}(x_k, P_k)$ at each time step k of (5) in which

$$x_k \approx \frac{1}{N} \sum_{i=1}^{N} q_k^i x_k^i, \tag{7a}$$

$$P_k \approx \sum_{i=1}^{N} q_k^i (x_k - x_k^i)(x_k - x_k^i)^T. \tag{7b}$$

This approximation is exploited for integrating the motion planner and NMPC algorithm in Section V.

IV. OBSTACLE AVOIDANCE IN NONLINEAR MODEL PREDICTIVE CONTROLLER

The NMPC tracking objective is formulated based on a smooth approximation of the reference motion plan at each control time step, parameterized with respect to a time-dependent path variable τ for which the change rate $\frac{\mathrm{d}\tau}{\mathrm{d}t} = \dot{\tau}$ is an additional control input [15]. In continuous time, the NMPC cost function therefore consists of the following terms

$$\int_0^T \left(\|F(x(t)) - y^{\mathrm{ref}}(\tau, d)\|_{W(t)}^2 \; + \; \|u(t)\|_R^2 \; + \; r_{\mathrm{s}}\, s(t) \right) \mathrm{d}t, \tag{8}$$

including a term for tracking the reference trajectory and regularization terms for penalizing control variables, where $W(t) \in \mathbb{R}^{n_{\mathrm{y}} \times n_{\mathrm{y}}}$ and $R \in \mathbb{R}^{n_{\mathrm{u}} \times n_{\mathrm{u}}}$ are the corresponding weighting matrices. In Eq. (8), $W(t)$ for all $t \in [0, T]$ is a time-varying tracking weight matrix, as opposed to the constant weights used in [5], [13], that allows for auto-tuning based on the second moment information provided by the motion planner (see Section V). The definition of a positive slack variable $s(t) \geq 0$ allows for an easy implementation of an exact L1 soft constraint penalty in the objective [16], i.e., $|s(t)| = s(t)$. Note that $r_{\mathrm{s}} > 0$ in (8) denotes the weight value for the slack penalty. The reference motion plan $x_{0:T}$ is approximated by a smooth function $y^{\mathrm{ref}}(\tau, d)$ that depends on the path variable τ as well as on additional parameters d. Let us define the tracking function in the objective based on a polynomial approximation of the reference trajectories

$$F(x(t)) - y^{\mathrm{ref}}(\tau, d) = \begin{bmatrix} p_X(t) - p_X^{\mathrm{ref}}(\tau, d_X) \\ p_Y(t) - p_Y^{\mathrm{ref}}(\tau, d_Y) \\ \psi(t) - \psi^{\mathrm{ref}}(\tau, d_\psi) \\ v_{\mathrm{x}}(t) - v_{\mathrm{x}}^{\mathrm{ref}}(\tau, d_v) \end{bmatrix} = \begin{bmatrix} p_X(t) - \sum_{i=0}^{n_X} d_X^i \tau^i \\ p_Y(t) - \sum_{i=0}^{n_Y} d_Y^i \tau^i \\ \psi(t) - \sum_{i=0}^{n_\psi} d_\psi^i \tau^i \\ v_{\mathrm{x}}(t) - \sum_{i=0}^{n_v} d_v^i \tau^i \end{bmatrix}. \tag{9}$$

The inequality constraints in the NMPC formulation include hard bounds on the control inputs and soft constraints for

limiting the distance to the parameterized reference trajectory, the vehicle velocity and the front steering wheel angle:

$$\underline{\dot{\tau}} \leq \dot{\tau} \leq \bar{\dot{\tau}}, \qquad -\bar{\dot{\delta}} \leq \dot{\delta} \leq \bar{\dot{\delta}}, \qquad -\bar{\dot{v}}_{\mathrm{x}} \leq \dot{v}_{\mathrm{x}} \leq \bar{\dot{v}}_{\mathrm{x}}, \tag{10a}$$

$$-\bar{e}_Y \leq e_Y + s, \quad -\bar{\delta}_f \leq \delta_f + s, \quad -\bar{v}_{\mathrm{x}} \leq v_{\mathrm{x}} + s, \tag{10b}$$

$$e_Y \leq \bar{e}_Y + s, \qquad \delta_f \leq \bar{\delta}_f + s, \qquad v_{\mathrm{x}} \leq \bar{v}_{\mathrm{x}} + s. \tag{10c}$$

where $e_Y = \cos(\psi^{\mathrm{ref}})\left(p_Y - p_Y^{\mathrm{ref}}\right) - \sin(\psi^{\mathrm{ref}})\left(p_X - p_X^{\mathrm{ref}}\right)$ is the path tracking error, which is defined as the orthogonal distance to the parameterized reference trajectory and $s(t) \geq 0$ denotes the slack variable. The obstacles are modeled as ellipsoidal sets, including uncertainty around the spatial extent of the (rectangular) OVs. Therefore, the obstacle avoidance constraints are formulated as

$$1 \leq \left(\frac{\delta_{x,j}(t)}{a_{x,j}}\right)^2 + \left(\frac{\delta_{y,j}(t)}{a_{y,j}}\right)^2, \tag{11}$$

where $\begin{bmatrix}\delta_{x,j}\\ \delta_{y,j}\end{bmatrix} = R(e_{\psi,j})^\top \begin{bmatrix}p_{\mathrm{X}} - e_{x,j}\\ p_{\mathrm{Y}} - e_{y,j}\end{bmatrix}$ is the rotated distance, $(e_{x,j}, e_{y,j}, e_{\psi,j})$ denotes the pose of the obstacle, and $(a_{x,j}, a_{y,j})$ denotes the lengths of the principal semi-axes of the ellipsoid that defines the safety margin around each obstacle. The time-varying ellipsoidal inequality constraint in (11) is defined for the $j = 1, \ldots, M$ nearest static or dynamic obstacles. This information can be obtained in real time, e.g., from an (extended) Kalman filter that is based on LIDAR measurements.

The resulting nonlinear optimal control problem (OCP) formulation includes $n_{\mathrm{x}} = 7$ differential states, $n_{\mathrm{u}} = 4$ control inputs and $N = 80$ control intervals with a sampling period of $T_{\mathrm{s}} = 25$ ms over a $T = 2$ s horizon length. The NMPC controller is implemented with a sampling frequency of 40 Hz, using the direct multiple shooting discretization method in combination with the RTI algorithm in the `ACADO` code generation tool and the `PRESAS` QP solver that was recently proposed in [17], [18].

V. REAL-TIME AUTOMATIC TUNING OF NMPC BASED REFERENCE TRACKING

It is important to note that the particle-filtering based motion planner in Section III computes a long-term, highly predictive reference trajectory but it typically needs to run at a relatively low sampling frequency, i.e., it has a relatively slow update rate (every $\sim$1 s) and therefore rather low reactivity. Instead, NMPC in Section III typically uses a much shorter prediction horizon but it runs at a much higher sampling frequency (every $\sim$25 ms), such that the controller can be highly reactive to local deviations, e.g., due to uncertainties in the pose estimation for the ego vehicle as well as for the surrounding obstacles. It is therefore important to share the responsibility between the planning and control layer in Figure 1a for ensuring safe and reliable driving behavior, especially in order to satisfy real-time obstacle avoidance requirements under uncertainty.

Our proposed approach involves an integrated software framework where the motion planner provides a reference trajectory (6) as well as its time-varying covariance (7b) to the NMPC controller. As illustrated in Figure 1b, this allows the reference tracking algorithm to automatically adapt the trade off that exists between competing control objectives such as, e.g., achieving high tracking performance while satisfying safe obstacle avoidance requirements. More specifically, one could expect the uncertainty of the reference motion plan to increase when the vehicle is predicted to become relatively close to surrounding obstacles, such that the penalization of deviations from the reference trajectories should decrease, and vice versa. This in turn allows larger, or lower, deviations of the NMPC trajectory from its reference, i.e., the motion planning trajectory.

This inverse proportional relation between the uncertainty of the motion planner and the tracking cost in the NMPC problem, can be used to explicitly compute a time-varying sequence of weighting matrices W_k for $k = 0, \ldots, N$ in Eq. (8), where N denotes the number of control intervals in the discrete time OCP formulation. Let us define the covariance matrices P_k in (7b) that correspond to the reference trajectory x_k in (7a) at a sequence of time points t_k for $k = 0, \ldots, N$. The time-varying tracking cost is defined as

$$W_k = P_k^{-1/2}\, Q_k\, P_k^{-1/2}, \tag{12}$$

using the square root of the inverse of the covariance matrix $P_k \succ 0$, given a symmetric nominal scaling matrix $Q_k \succ 0$. The above expression becomes computationally much cheaper to evaluate when ignoring off-diagonal elements in the covariance matrices, i.e.,

$$W_k(i,i) = \frac{Q_k(i,i)}{\max(\epsilon, P_k(i,i))}, \quad \text{for } i = 1, \ldots, n_x, \tag{13}$$

where the nominal scaling matrix $Q_k \succ 0$ has additionally been assumed to be diagonal and a regularization parameter $\epsilon > 0$ has been included for ensuring good numerical conditioning. As the motion planner operates at a lower rate than the controller, the reference trajectories, in combination with the corresponding covariance matrices, are shifted from one control time step to the next, until a new reference trajectory is computed by the motion planner. The resulting integrated planning and control strategy for autonomous driving is summarized in Algorithm 1.

Algorithm 1 Automatic Tuning for NMPC Reference Tracking Cost using Particle Filtering based Motion Planner.

Input: Time-varying trajectories of reference values $\left\{y_k^{\mathrm{ref}}\right\}_{k=0,\ldots}$ and covariance matrices $\{P_k\}_{k=0,\ldots}$.

```
while true do                                          ▷ Control loop
    if new reference motion plan then
        Reset trajectories: {y_k^ref}_{k=0,...} and {P_k}_{k=0,...}.
    Smooth approximation y^ref(·) of reference motion.
    for k = 0, ..., N do
        Automatic tuning: W_k, given P_k in (12) or (13).
    Wait for next time step after T_s = 25 ms.
end while
```

Fig. 2. The Ackermann-steered Hamster robot used in experiments. The markers are used to track the robot via an OptiTrack motion-capture system.

VI. EXPERIMENTAL RESULTS

We use a test platform of small-scale autonomous vehicles, based on the Hamster robots as illustrated in Figure 2, for testing and validating our complete planning, control and estimation software stack before deploying in full-scale vehicle experiments [5], [13].

A. Scaled Vehicle Experimental Platform

The Hamster is a 25×20 cm mobile robot for research and prototype development, see Figure 2. It is equipped with scaled versions of sensors commonly available on full-scale research vehicles, such as a 6 m range mechanically rotating 360 deg LIDAR, an inertial measurement unit (IMU), GPS receiver, HD camera, and motor encoders. It uses two Raspberry Pi 3 computing platforms, each with an ARM Cortex-A53 processor. The robot has Ackermann steering and is therefore kinematically equivalent to a full-scale vehicle, and its dynamic behavior, such as the suspension system, resembles that of a vehicle. It has built-in mapping and localization capabilities, and object detection and tracking can be done with the onboard LIDAR and/or camera.

Hence, the platform is a good test setup for verifying the dynamic feasibility and performance of vehicle control and estimation software in a realistic setting, with a sensor setup similar to the one expected in full-scale autonomous vehicles. The Hamster communicates and connects to external algorithms using the robot operating system (ROS). In order to be able to accurately evaluate the control and estimation algorithms in terms of performance in a controlled environment, we use an OptiTrack motion capture system. The OptiTrack system is a flexible camera-based six degrees-of-freedom tracking system that can be used for tracking drones, ground vehicles, and industrial robots.

B. Experimental Validation Results

We use three Hamsters in the experimental validation, one acts as the ego vehicle (EV) and two act as obstacles. For simplicity of illustrating the performance of our proposed algorithmic framework, both obstacle vehicles (OVs) are restricted to be static in these experiments, even though results with dynamic obstacles have been presented in [4], [5], and our modified approach also works in those scenarios. Our objective is to avoid the obstacles while circulating a two-lane closed circuit in the counter-clockwise direction, with the inner lane as preferred lane, see Figure 3. The OVs in this scenario could represent, e.g., two parked vehicles on the side of each of the two lanes on the road. The position of the EV is obtained from the OptiTrack system. As for the OVs, their position is estimated in real-time using the onboard LIDAR measurements.

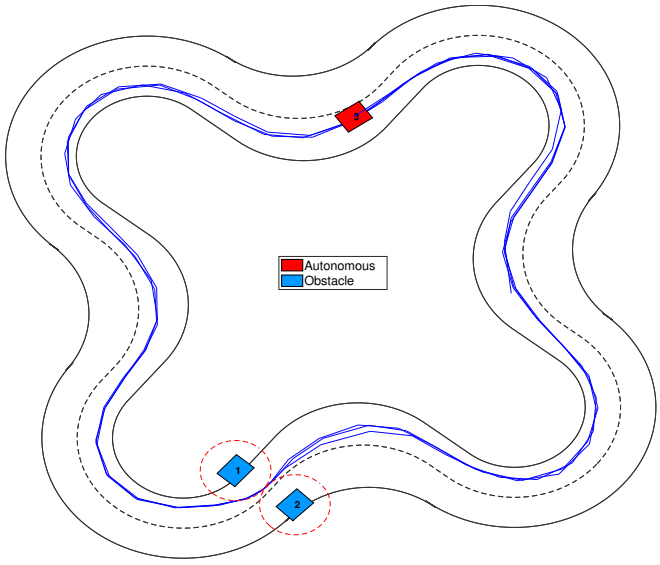


Fig. 3. Illustration of EV circulating a two-lane closed circuit in counter-clockwise direction (blue trace), while safely avoiding two static OVs.

Figure 4 shows results of three separate experiments in which the integration between the particle-filtering based motion planner and the NMPC is based on either the auto-tuning method in Algorithm 1 or a fixed high (low) tracking cost weight $W(t) = W_{\text{high}}$, or $W(t) = W_{\text{low}}$, for all t in Eq. (8). The closed-loop time trajectories show the lateral tracking error and the average weighting matrix value on tracking of the vehicle position, in order to illustrate the corresponding driving behaviors. The EV passes in between the static obstacles along the two-lane closed circuit (see Figure 3) at $t = 75$ and $t = 120$ s, corresponding to the highlighted time windows in Figure 4. As expected, it can be observed that the auto-tuning based implementation results in an automatic balancing of the trade off between high tracking performance, similar to the results when using a fixed but relatively high tracking cost, and the larger safety margins for obstacle avoidance that are achieved when lowering the tracking cost weight. More specifically, in Figure 4, one can observe that the tracking cost becomes consistently lower for the auto-tuning method at $t = 75$ and $t = 120$ s, due to an increase in uncertainty of the motion planner, resulting in an increased lateral error to ensure safe obstacle avoidance.

Figure 5 shows in more detail the satisfaction of safe obstacle avoidance requirements that is achieved when lowering the tracking cost weight, which is performed automatically in the proposed auto-tuning implementation due to the increased covariance of the motion planner trajectory when the distance to surrounding obstacles is relatively small. The figure shows a snapshot of the computed particles and

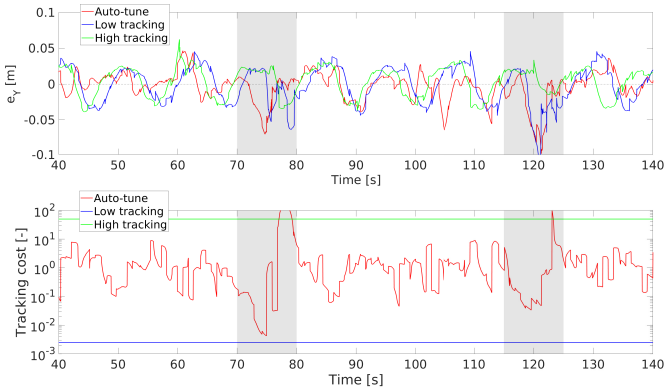
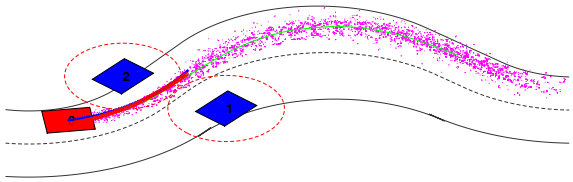
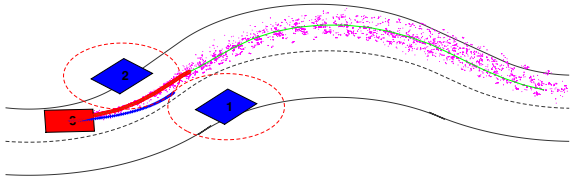


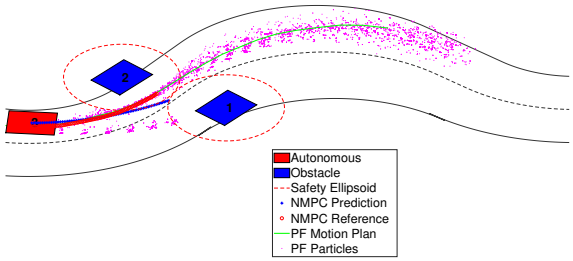
Fig. 4. Closed-loop time trajectories showing lateral tracking error and position tracking cost for three separate experiments: NMPC with auto-tuning, NMPC with fixed low and fixed high tracking cost. The highlighted time windows correspond to the EV passing in between the two OVs.



(a) Motion planning and NMPC: high tracking cost weight.



(b) Motion planning and NMPC: low tracking cost weight.



(c) Motion planning and NMPC: auto-tuning (see Alg. 1).

Fig. 5. Snapshots that illustrate the swaying behavior for the vehicle experimental results at around 120 s in Figure 4.

the reference trajectory from the motion planner and the predicted trajectory from the NMPC controller at around 120 s in each of the three separate vehicle experiments. Indeed, one can observe that the automatic spreading of the particles when the ego vehicle is relatively close to surrounding obstacles, leads to an increased uncertainty and therefore an increased flexibility for the NMPC controller to deviate from the reference motion plan.

VII. CONCLUSIONS

The present paper proposed a tighter integration of a particle filtering based motion planner and a nonlinear model predictive controller for real-time tracking, based on the motion planner providing not only the trajectory but also information about the second statistical moments. The planning algorithm computes a reference trajectory to be tracked and the corresponding covariance matrices, which are used for automatically tuning the time-varying tracking cost in the NMPC formulation. Experimental results based on a test platform of small scale vehicles are used to illustrate the more strongly integrated hierarchical planning and control strategy for autonomous driving systems.

REFERENCES

[1] H. Ahn, K. Berntorp, and S. D. Cairano, "Reachability-based decision making for city driving," in *Amer. Control Conf.*, June 2018.

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

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

[4] K. Berntorp, T. Hoang, and S. Di Cairano, "Motion planning of autonomous vehicles by particle filtering," *IEEE Trans. Intell. Veh.*, vol. 4, no. 2, pp. 197–210, 2019.

[5] K. Berntorp, P. Inani, R. Quirynen, and S. Di Cairano, "Motion planning of autonomous road vehicles by particle filtering: Implementation and validation," in *Proc. Amer. Control Conf. (ACC)*, 2019.

[6] D. Hrovat, S. Di Cairano, H. E. Tseng, and I. V. Kolmanovsky, "The development of model predictive control in automotive industry: A survey," in *Int. Conf. Control Applications*, Dubrovnik, Croatia, 2012.

[7] F. Borrelli, P. Falcone, T. Keviczky, and J. Asgari, "MPC-based approach to active steering for autonomous vehicle systems," *Intern. J. of Vehicle Autonomous Systems*, vol. 3, no. 2, pp. 265–291, 2005.

[8] J. V. Frasch, A. J. Gray, M. Zanon, H. J. Ferreau, S. Sager, F. Borrelli, and M. Diehl, "An auto-generated nonlinear MPC algorithm for real-time obstacle avoidance of ground vehicles," in *Proc. European Control Conf. (ECC)*, 2013, pp. 4136–4141.

[9] 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.

[10] R. Verschueren, S. D. Bruyne, M. Zanon, J. V. Frasch, and M. Diehl, "Towards time-optimal race car driving using nonlinear MPC in real-time," in *Proc. IEEE Conf. Dec. Control (CDC)*, 2014, pp. 2505–2510.

[11] M. Diehl, H. G. Bock, and J. P. Schlöder, "A real-time iteration scheme for nonlinear optimization in optimal feedback control," *SIAM J. Control and Optim.*, vol. 43, no. 5, pp. 1714–1736, 2005.

[12] S. Gros, M. Zanon, R. Quirynen, A. Bemporad, and M. Diehl, "From linear to nonlinear MPC: bridging the gap via the real-time iteration," *International Journal of Control*, 2016.

[13] K. Berntorp, T. Hoang, R. Quirynen, and S. Di Cairano, "Control architecture design for autonomous vehicles," in *Proc. IEEE Conf. on Control Techn. and Appl. (CCTA)*, 2018.

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

[15] T. Faulwasser and R. Findeisen, "Nonlinear model predictive control for constrained output path following," *IEEE Trans. Automatic Control*, vol. 61, no. 4, pp. 1026–1039, 2015.

[16] R. Fletcher, *Practical Methods of Optimization*, 2nd ed. Chichester: Wiley, 1987.

[17] R. Quirynen, K. Berntorp, and S. Di Cairano, "Embedded optimization algorithms for steering in autonomous vehicles based on nonlinear model predictive control," in *Proc. Amer. Control Conf. (ACC)*, 2018.

[18] R. Quirynen and S. D. Cairano, "PRESAS: Block-structured preconditioning of iterative solvers within a primal active-set method for fast MPC," *arXiv preprint arXiv:1912.02122*, 2019.