A Model-Free Algorithm to Safely Approach the Handling Limit of an Autonomous Racecar

1st Alexander Wischnewski
Chair of Automatic Control
Technical University of Munich
Munich, Germany
alexander.wischnewski@tum.de

2nd Johannes Betz
Chair of Automotive Technology
Technical University of Munich
Munich, Germany
betz@ftm.mw.tum.de

3rd Boris Lohmann
Chair of Automatic Control
Technical University of Munich
Munich, Germany
lohmann@tum.de

***Abstract*—One of the key aspects in racing is the ability of the driver to find the handling limits of the vehicle to minimize the resulting lap time. Many approaches for raceline optimization assume the tire-road friction coefficient to be known. However, this neglects the fact that the ability of the system to realize such a race trajectory depends on complex interdependencies between the online trajectory planner, the control systems and the non-modelled uncertainties. In general, a high quality control system can approach the physical limit more reliable, as it applies less corrective actions. We present a model-free learning method to find the minimum achievable lap-time for a given controller using online adaption of a scale factor for the maximum longitudinal and lateral accelerations in the online trajectory planner. In contrast to existing concepts, our approach can be applied as an extension to already available planning and control algorithms instead of replacing them. We demonstrate reliable and safe operation for different vehicle setups in simulation and demonstrate that the algorithm works successfully on a full-size racecar.**

***Index Terms*—Autonomous Racing, Learning Control, Model-Free**

I. INTRODUCTION

Autonomous driving has received great interest recently in both research and public discussions. A widely acknowledged approach is the separation of the task into perception, planning and control [1]. Recently, planning and control at the handling limits is discussed by several authors [2, 3, 4, 5]. While the split between planning and control is beneficial in terms of the development and setup process, it poses challenges when the lap time shall be minimized and the vehicle is driven at the handling limits. We identified three main difficulties while operating such a system:

1) The trajectory planner does not consider the additional lateral and longitudinal accelerations applied by the feedback-controller.
2) Deviations in the velocity tracking influence the required lateral acceleration quadratically. Therefore, its control performance has a severe impact on path tracking quality.
3) The trajectory planner applies a certain safety distance to obstacles to account for tracking errors. It is not clear a-priori how to choose this, as the tracking accuracy might decrease significantly at the handling limits.

Research was supported by the basic research fund of TU Munich.

To solve those challenges, different concepts have been presented in the literature recently: Laurense et al. [4] described a control strategy utilizing the longitudinal velocity which aims at tracking a desired front axle side slip angle calculated from a friction estimate. Due to the strong correlation of side slip with tire utilization, they could reduce the lateral error significantly in comparison to a classic path tracking controller. A completely different approach is proposed in [6], using stochastic model-predictive-control (SMPC) to account for the system uncertainties in a combined planning and tracking control setup. The work was extended in [7] using a data-driven model identification method. The approach uses constraint-tightening to guarantee that the safety constraints hold in the presence of uncertainty observed according to the identified system model. With increasing model accuracy, the driving becomes less conservative and more precise. A similar approach was combined with a Safe-MPC algorithm that can guarantee the recursive feasibility of the planning and control problem in [8]. Again, the vehicle becomes faster as more data is gathered.

In contrast to the presented references, our approach is capable of minimizing the lap time in the presence of disturbances and uncertainties for a given control and planning system with only minor modifications. Furthermore, it is not necessary to specify or identify a vehicle dynamics model as the algorithm works purely on the observations of safety constraints. The mathematical background on the algorithm applied in this paper is based on the work presented in [9]. It utilizes a Bayesian Optimization strategy with Gaussian Processes for learning the cost-function and safety constraints. Our work focusses on the choice of optimization variables, safety constraints and derivation of suitable hyperparameters for the autonomous racing application. Furthermore we present applications of the algorithm in simulation and real-world testing scenarios.

The mathematical background on Bayesian Optimization and the applied algorithm is presented in Section II. The application of the algorithm to the autonomous racing task and its results are discussed in Section III and IV. Section V reviews the achievements and presents further research directions.

978-1-7281-0142-2/19/$31.00 ©2019 IEEE

II. METHODOLOGY

A. Gaussian Processes

Gaussian Processes are a machine learning technique for non-parametric function approximation. It allows one to model a non-linear function $y = f(x)$: $\mathbb{R}^d \mapsto \mathbb{R}$ using a finite set of n observations $\{(x_1, y_1), ... (x_n, y_n)\}$. The following presentation of the mathematical background is based on [10].

The Gaussian Process (GP) itself is modelled as the joint probability distribution over a finite set of Gaussian random variables. The predictions y_* can be drawn by conditioning the prediction point x_* on the observation set. The observations are stochastically related to each other based on a kernel function $k(x_i, x_j)$ which yields the covariance between x_i and x_j. This allows one to specify the joint normal probability distribution of measurement samples and predictions as

$$\begin{bmatrix} Y \\ y_* \end{bmatrix} \sim \mathcal{N}\left(m(x), \begin{bmatrix} K(X,X) + \sigma_{\mathrm{M}}^2 I & K(X, x_*) \\ K(x_*, X) & K(x_*, x_*) \end{bmatrix}\right), \quad (1)$$

where Y is the vector with measurements $\begin{bmatrix} y_0 & y_1 & \dots & y_N \end{bmatrix}^{\mathrm{T}}$, $m(x)$ the prior mean function, $K(X,X)$ the kernel matrix evaluating the kernel function $k(x_i, x_j)$ for every combination of the observation points and σ_{M}^2 the measurement variance. Note, that the input domain x might be vector valued, but the covariance function $k(\cdot)$ returns a scalar. The posterior prediction mean can be written as:

$$\begin{aligned} \mu(x_*) =& m(X) \\ &+ K(x_*, X)\left(K(X,X) + \sigma_{\mathrm{M}} I\right)^{-1}(Y - m(X)) \end{aligned} \quad (2)$$

and covariance

$$\begin{aligned} \sigma(x_*) =& K(x_*, x_*) \\ &- K(x_*, X)\left(K(X,X) + \sigma_{\mathrm{M}} I\right)^{-1} K(X, x_*) \end{aligned} \quad (3)$$

based on the observations X and Y. In the following, we will denote the mean of a GP which approximates a function $f(x)$ with $\mu_{\mathrm{f}}(x)$ and its variance with $\sigma_{\mathrm{f}}(x)$.

One of the main reasons for the great interest in GPs for optimization derives from the fact that they specify the uncertainty of their predictions. This can be used to control the exploration strategy during the optimization process.

B. Safe Bayesian Optimization

Our optimization approach utilizes the SafeOpt algorithm [9]. In the following, we will only sketch the key steps and discuss their implications with respect to the optimization process and practical application. The aim of the optimization is to solve

$$\begin{aligned} \min_{x} \quad & f(x) \\ \text{subject to} \quad & g_i(x) \leq b_i, \ i = 1, \dots, m \\ & x \in \mathcal{X}, \end{aligned} \quad (4)$$

where $f(x)$ is the scalar-valued cost function, $g_i(x)$ are the safety constraints and $\mathcal{X}$ is the set of admissible parameters.

In contrast to classic optimization approaches, the cost and constraint functions are unknown before the system starts its operation. A GP is used to model them and generate samples for each function from the measurement data. Using the confidence interval β, the upper (UCB)

$$g_i^+(x) = \mu_{g,i}(x) + \beta \sigma_{g,i}(x) \quad (5)$$

and lower confidence bounds (LCB)

$$g_i^-(x) = \mu_{g,i}(x) - \beta \sigma_{g,i}(x) \quad (6)$$

can be defined. They are used to derived three important sets for the optimization. First, the set of points x which are likely to be safe after the n-th iteration, can be defined using the UCB as

$$\mathcal{S}_n = \left\{x \in \mathcal{X} | g_i^+(x) \leq b_i, i = 1, \dots, m\right\}. \quad (7)$$

Within this set, the set of potential minimizers can be formulated as

$$\mathcal{M}_n = \left\{x \in \mathcal{S}_n | f^-(x) \leq \min_{x' \in \mathcal{S}_n} f^+(x')\right\}. \quad (8)$$

It specifies the points that have a chance to further minimize the current, conservative best value of the cost function. Finally, the set of potential expanders is defined as the set of points that have the chance to enlarge the safe set. This is formalized as follows: Let

$$h(x_{n+1}) = |\left\{x' \in \mathcal{X} \setminus \mathcal{S}_n | x' \in \mathcal{S}_{n+1}\right\}| \quad (9)$$

be the number of points that are not in the safe set for the available observations but that become safe after adding an optimistic estimate for the constraints $y_{n+1} = g_i^-(x_{n+1})$ to the GPs. The set of potential expanders can now be written as

$$\mathcal{G}_n = \left\{x \in \mathcal{S}_n | h(x) > 0\right\}. \quad (10)$$

It remains to specify an acquisition function for how to determine the next query point for a current set of measurements. Berkenkamp et al. propose to use

$$x_{n+1} = \underset{x \in \mathcal{M}_n \cup \mathcal{G}_n}{\operatorname{argmax}} \ \sigma_{\mathrm{f}}(x), \quad (11)$$

as the algorithm can be shown to converge safely to the global safe optimal value for this choice [9].

C. Trajectory Planning and Control

The trajectory planning for the racecar is split into an online and an offline part [11]. The offline part uses a sophisticated track and vehicle dynamics model to generate the time-optimal trajectory based on an optimal-control formulation. The online phase is divided into a local path generation and a velocity profile planning problem. This allows to incorporate dynamic objects and readjustments of the velocity profile according to changing acceleration limits.

The control system consists of independent lateral path and velocity tracking controllers. The latter is a P-controller combined with disturbance estimation and a feed-forward term. A gain-scheduled PD-controller accompanied with a feed-forward term is used for path tracking. More information on the controller and overall software structure can be found in [11, 12].

III. ALGORITHM

A. Choice of Optimization Variables

The overall aim in racing is to minimize one's lap time without putting the vehicle into unstable driving situations. To achieve this, we choose to apply a scale factor θ to the acceleration limits calculated from the friction settings. This can be interpreted as a safety margin with respect to the available friction level. The acceleration limits are leveraged by the trajectory planner to adjust the velocity profile while driving. It was decided not to readjust the raceline itself in order to obtain reproducible results with respect to the safety margins. One of the key advantages of this variable choice is, that the optimization problem can be formulated to maximize the scale factor instead of minimizing the lap-time. It is known from the racing literature, that both targets are equivalent [13]. This increases the applicability of the algorithm for autonomous racing, as a data driven minimization of lap time poses difficulties due to external disturbances such as other cars or overtaking scenarios. Those would prevent the algorithm from establishing a clear relation between the optimization variable and the target variable. The problem itself can be written as:

$$\begin{aligned} \min_{\theta} \quad & -\theta \\ \text{subject to} \quad & g_i(\theta) \leq b_i,\ i = 1, \dots, m \\ & \theta \in \mathcal{O}, \end{aligned} \tag{12}$$

with the additional safety constraints g_i. The overall software architecture of the algorithm is depicted in Fig. 1.

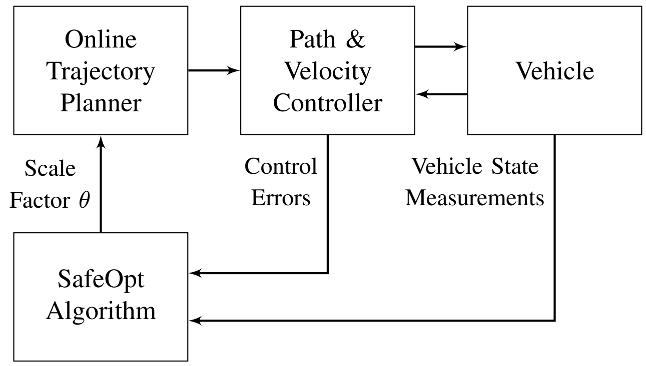

Fig. 1. Safe learning system architecture to minimize the resulting lap-time. The optimization algorithm is driven by the control errors and vehicle sensor data. It can influence the trajectory planner via a scale factor θ for the maximum longitudinal and lateral accelerations.

B. Choice of Constraints

The applied safe optimization strategy interprets safety as a set of constraints. To be safe, it has to guarantee that the statistical assumptions made during the prediction of the GPs are conservative, in the sense that at no time a point is predicted to be safe that is not safe with respect to the constraints. We will discuss how to achieve this by a proper selection of kernel parameters in the upcoming section. Furthermore, it is required that all parameters within the actual safe set lead to a valid execution of the task by the agent. In the case of autonomous racing, this means that all parameters that are predicted to be safe must lead to a completed lap without incidents. This translates to a strict stability requirement, as the vehicle would spin off the track otherwise. It is difficult to guarantee this analytically, since the dynamics model is highly nonlinear and uncertain in the racing case. While some authors apply system identification techniques to overcome this issue [7, 8], the aim of this paper is to set a baseline for model-free approaches. Therefore, we apply two different heuristic measures for vehicle stability from vehicle dynamics science [13]:

- The difference between the front and rear axle side-slip angle $\alpha_{\mathrm{F}} - \alpha_{\mathrm{R}}$. It relates the remaining potential of the front axle to the potential of the rear axle, being a key indicator for the amount of friction utilization. As the car has a tendency to understeer at the limits of friction, this value becomes large.
- The maximum wheel slip λ at one of the four wheels (denoted by the indices FL, FR, RL and RR). It indicates the tendency to lock or spin the wheel and that the vehicle operates near the friction limit.

It is usually difficult to judge how close the vehicle is to the limit or instability only based on data. While these criteria provide good indicators, they might fail to give a reliable estimate. This significantly depends on the vehicle dynamics design. Passenger cars are designed to have a broad transition region until the tire nonlinearity becomes relevant. In contrast, racecars are usually setup for maximum lateral acceleration instead of providing a cautious transition from stable to unstable regions of vehicle dynamics. We therefore have to rely on an underlying vehicle stability control system to handle these cases and prevent that the vehicle spins of the raceline at the cost of increased tracking error in these situations. In addition, we have to enforce a performance constraint on the absolute lateral tracking error e. It must stay within the safety margin specified by the planning algorithm, otherwise the vehicle will crash into the track boundaries.

All safety constraints must be fulfilled during the complete lap. The set of measurement points that belong to the lap with number i is denoted via $\mathcal{L}_i$. To decrease noise sensitivity, a moving average filter with a time window of $50\,\mathrm{ms}$ is applied to the sample time series before calculating the corresponding safety value. The time constant must be kept as small as possible, since too aggressive filtering could lead to violations of the safety specifications. The constraints can now be formalized:

$$g_1(\theta_i) = \max_{k \in \mathcal{L}_i} \quad |\alpha_{\mathrm{F}}(k) - \alpha_{\mathrm{R}}(k)| \tag{13a}$$

$$g_2(\theta_i) = \max_{k \in \mathcal{L}_i} \quad \max_{j \in \{\mathrm{FL,FR,RL,RR}\}} \quad |\lambda_j(k)| \tag{13b}$$

$$g_3(\theta_i) = \max_{k \in \mathcal{L}_i} \quad |e(k)| \tag{13c}$$

While the safety threshold for the last constraint is given by the specification of the trajectory planner, suitable thresholds for the heuristic stability criteria will be derived below.

C. Choice of Priors, Kernel Functions and Hyperparameters

The failure probability of the algorithm depends essentially on the accuracy of the statistical representation of the GP functions [14]. There are three main influence factors for this: The prior mean function $m(x)$ of the GP, the kernel $k(x_i, x_j)$ and the hyperparameters associated with the kernel function. While the algorithm provides a theoretical safety guarantee, this requires the knowledge of an upper bound of the norm of the final function in the Reproducing Kernel Hilbert Space (RKHS) in advance. This can be interpreted as a measure of function complexity [10]. As this is usually unknown in a practical setting, choosing the hyperparameters in a conservative manner is advised [14].

Following the arguments in [14], we apply a Matèrn Kernel with $\nu = 3/2$ so that the predictions are likely to have continuous first derivatives [10]. In contrast to the squared exponential kernel, it is less smooth, which is beneficial for the approximation of the uncertain safety constraints. This can be interpreted as a conservative choice. The variances are chosen to reflect the actual uncertainty about the outcome of the experiments based on previous test runs and are scaled to the physical units of the respective measurement variable. Finally, the kernel length scale parameter is chosen so that the GPs can generalize for a range of approximately 0.02-0.04. This resembles cautious steps near the friction limit based on past experience.

The priors' main effect is that they speed up the exploration process as they encode domain specific knowledge about the safe set and the cost function. At the same time, this can cause risk in case of misspecification. We choose to be conservative and use exponential functions that tend to have large values for an increase of the friction scale factor. In general, we know from manual driving experiments that realistic maximum scale factors range around 1.0. This leads to the conclusion, that factors below 1.0 are not safety critical. We therefore design the exponentials to cross the safety threshold at that value. At the same time, we know that scale factors above 1.2 are not realistic. We encode this in the priors so that they have roughly three times the value of the safety threshold at 1.2. The resulting priors are depicted in Table I. Note, that the cost function holds a different prior. This is due to the unusual construction of the optimization problem. In fact, it is not required to model this function as a GP at all, however, this is done to formulate the problem easily in the software framework at hand.

TABLE I
GP SETUP

Function	Prior	Variance	Length Scale	Unit
Cost	$-\theta$	1e−4	0.15	
Understeer	$b_1 e^{5.5(\theta-1)}$	1e−5	0.04	radians
Wheelslips	$b_2 e^{5.5(\theta-1)}$	1	0.04	percent
Control Error	$b_3 e^{5.5(\theta-1)}$	1e−2	0.04	meter

D. Choice of measurement uncertainty and safety thresholds

The standard GP framework allows one to model measurement uncertainty through the introduction of an additive measurement covariance matrix $\sigma_{\mathrm{M}}^2 I$ to the upper left block of the covariance matrix in (1). In this setting, the variance of the posterior distribution will become small once the number of samples in a specific region of the input domain becomes large. Eventually, it will converge to zero if the number of samples goes to infinity.

This is problematic for the application in a racing scenario. The controllers' main purpose is to mitigate external disturbances and uncertainties. While the latter will be comparable from lap to lap, the former can vary significantly. This could be caused e.g. by wind or dirt on the track surface. We will refer to this phenomenon as the repetition uncertainty u_{R} with variance σ_{R}^2, specified similar to the measurement uncertainty in the form of a Gaussian random variable. The GP framework would allow one to account for this using a white noise kernel, which adds uncertainty to the posterior that cannot be removed by conditioning it on the measurement data. However, this leads to difficulties in the application of the SafeOpt algorithm. They are caused by the fact that the potential minimizer set is calculated based on an optimistic prediction. If the uncertainty about the process does not converge to zero, this set will fail to shrink to size one. This results in a random choice of parameters within this set as also the acquisition function will converge to the same values for every sample.

Instead, we show that its possible to handle the repetition uncertainty by a conservative choice of the safety thresholds b_i. The actual constraint can be written as:

$$g_i(x) + u_{\mathrm{R}} \leq \bar{b}_i. \tag{14}$$

This shall hold with a probability of at least $1 - \delta$, where δ is the tolerated failure probability. Assuming that the SafeOpt algorithm can guarantee that $g_i(x) \leq b_i$ holds with the same probability, we can guarantee that the above holds if we set $b_i = \bar{b}_i - p\sigma_{\mathrm{R}}$. The safety factor p is obtained from the Gaussian normal distribution such that it covers $1 - \delta$ of the probability distribution.

Proof: The combined variance σ_{F}^2 of $g_i(x) + u_{\mathrm{R}}$ is equal to $\sigma_{g,i}^2(x) + \sigma_{\mathrm{R}}^2$, since both are Gaussian and uncorrelated. It follows that $\Pr\left(g_i(x) + u_{\mathrm{R}} \leq \bar{b}_i\right) \geq 1 - \delta$ if $\mathbb{E}\left[g_i(x) + u_{\mathrm{R}}\right] + p\sigma_{\mathrm{F}} \leq \bar{b}_i$. Since $\sigma_{\mathrm{F}} = \sqrt{\sigma_{g,i}^2(x) + \sigma_{\mathrm{R}}^2} \leq \sigma_{g,i}(x) + \sigma_{\mathrm{R}}$ for $\sigma_{\mathrm{R}}, \sigma_{g,i}(x) > 0$ and $\mathbb{E}\left[u_{\mathrm{R}}\right] = 0$, we can approximate the previous inequality conservatively as $\mu_{g,i}(x) + p\left(\sigma_{g,i}(x) + \sigma_{\mathrm{R}}\right) \leq \bar{b}_i$. After algebraic modifications, we have $\mu_{g,i}(x) + p\sigma_{g,i}(x) \leq \bar{b}_i - p\sigma_{\mathrm{R}}$. Setting $b_i = \bar{b}_i - p\sigma_{\mathrm{R}}$, we have shown the required result. ∎

Note that the approximation approaches equality if the GP variance $\sigma_{g,i}^2(x)$ becomes small. This means the approach is conservative for the beginning of the experiment, however, it eventually converges to the exact result. The final thresholds b_i are set based on simulation and track data to $b_1 = 0.06$, $b_2 = 10$ and $b_3 = 0.7$.

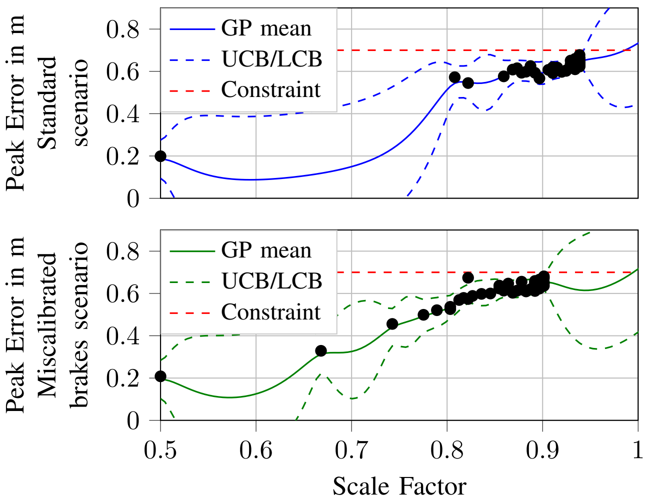

Fig. 2. The plots show the learned control error GP for the standard setup (top) and a miscalibrated brake actuator (bottom) using mean values and three sigma confidence intervals. The dots resemble measurement samples for a single lap. As the lateral error is the limiting constraint in this case, the other constraints are not visualized.

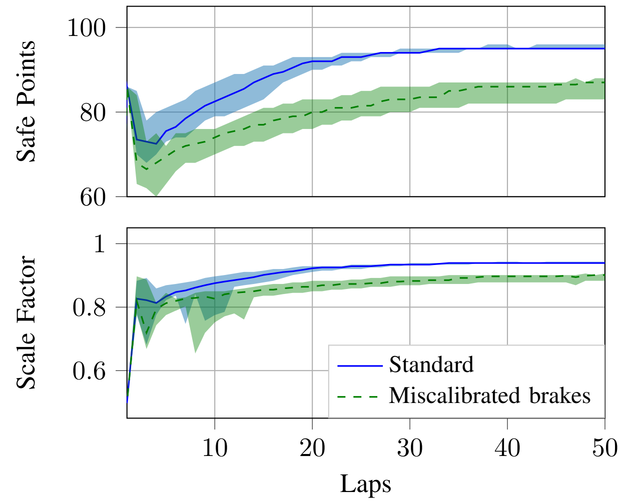

Fig. 3. The algorithm has been simulated ten times for both scenarios to analyse the variance between different runs. The uncertainty is introduced to the non-deterministic timing of many of the trajectory planner and the sensor models. The plots show the median value as bold line for each scenario. The shaded area reflects the maximum and minimum value via all iterations.

The actual choice of the measurement variance σ_{M}^2 for each GP remains to be discussed. In contrast to the repetition uncertainty σ_{R}^2, it reflects the uncertainty within the measurement, which can be reduced by consecutive measurements for the same sample points. Its proportion to the prior variance determines how strong the samples are weighted with respect to the priors. The variances are set to $\sigma_{\mathrm{M,g,1}}^2 = 1\mathrm{e}{-6}$, $\sigma_{\mathrm{M,g,2}}^2 = 4\mathrm{e}{-2}$ and $\sigma_{\mathrm{M,g,3}}^2 = 1\mathrm{e}{-3}$, which resembles a strong emphasis on the measurements.

IV. RESULTS

The algorithm is first applied to a detailed vehicle dynamics simulation. It is equipped with sensor and actuator models and a nonlinear single-track model with a Pacejka Tire Model. Since the trajectory planner, the SafeOpt algorithm and the vehicle physics run in separate processes, the outcome of the simulation is not deterministic. However, this reflects the behaviour of the car, as we cannot guarantee the timing of the trajectory planner as it is not implemented as a hard real-time application.

We use a standard simulation parametrization as a baseline and discuss the performance of the algorithm for a miscalibrated brake controller as a benchmark. It is set to a 30 % error in the brake requests and will therefore lead to a degraded velocity tracking. Fig. 2 depicts the learned GP for the peak control error function, which limits the exploration in both scenarios.

As pointed out before, the behaviour of the algorithm can not be considered deterministic, due to timing differences and different initial conditions for the sensor noise. The results of 10 experiments with 50 laps for each scenario are depicted in Fig. 3. The median is a scale factor of 0.90 for the misspecified brake actuator setting, which is below the final value of 0.94 in the standard setting. Two things should be noticed: First, the algorithm converges very slowly once it is near the friction limit, which is due to its conservative parametrization. Second, outliers in the constraints tend to force the algorithm to go back to a more conservative value for a few iterations even if it is near its final result. Again, this reflects the conservative nature of the algorithm, being focussed more on safety than on pure performance maximization.

Finally, the algorithm is applied to a full-size racecar called *DevBot 2.0*. It has a two-wheel electric drivetrain, steer- and brake-by-wire systems and is used within the Roborace Championship for autonomous vehicles. Details on the vehicle setup can be found in [15, 11]. The trials took place at the Monteblanco Circuit, Spain. The algorithm converged to an acceleration limit scale factor of 0.83, which was slightly below the factor found during manual setup of the autonomous driving system, which was 0.9. This deviation can be attributed to the conservative wheelspeed constraint value, where 10 % seems too low for the actual vehicle setup since there has not been any sign of instability noted by the safety driver or within the measurement data. The limiting constraint GP is depicted in Fig. 4.

V. CONCLUSIONS AND OUTLOOK

We have presented an algorithm that is capable of mitigating the limitations of the vehicle in the sense, that it scales the allowed accelerations in trajectory planning so that the vehicle does not violate the safety constraints. It has been proven to come reasonably close to the result achieved by manual tuning of the scale factor. It does not depend on detail knowledge about the underlying tracking controller. This facilitates easy integration in already available software stacks. Furthermore, it has been shown to be robust with respect to moderate constraint prior misspecification, which is an important feature for algorithms applied within real-world scenarios.

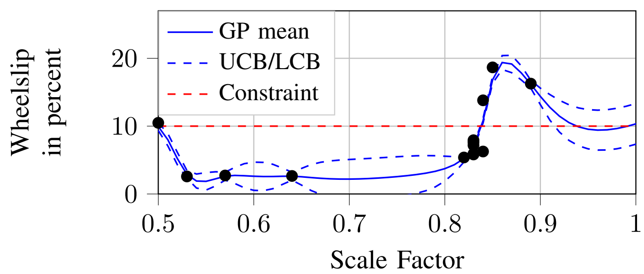

Fig. 4. At the Monteblanco Racetrack the vehicle was stopped from further minimizing the lap time by a conservative setting of the wheelslip constraint. Only the limiting constraint is depicted for a run of 14 laps. Non-safe samples were drawn due to a mismatch of the prior function and the real-world behaviour of the system. The high wheelslip sample for scale factor 0.5 is related to starting to drive from standstill. Since the prior specifies other values to be safe, this does not stop the algorithm from exploring.

One drawback of the algorithm is that the vehicle is not capable of adjusting its accelerations limitations to different circumstances on different areas on the track. Furthermore, it could leverage this information to learn faster than on a per lap basis. The difficulty with this lies in the fact that the control error depends significantly on the initial conditions present at the entry of a subsection of the lap. While this is already a problem for the algorithm running on a per lap basis, lap switching usually takes place at the start-straight where the control errors are rather low and therefore neglectable. Another difficulty lies in the fact that it is conservative with respect to outliers that exceed the constraint values. Since the algorithm does not actively retry them, they might block the exploration in case the GP length scales are set rather small, which is usually desirable to ensure safe exploration at the limits.

Future work will be dedicated to the utilization of information on a finer grid rather than a per lap basis and will leverage additional model information in the learning process. The latter promises to result in more robust guarantees in terms of control system stability and could decrease the dependence upon heuristic vehicle stability criteria.

CONTRIBUTIONS AND ACKNOWLEDGMENTS

Alexander Wischnewski initiated the idea for the paper and is responsible for the presented design and implementation. Johannes Betz and Boris Lohmann contributed equally to the conception of the overall vehicle system and the research project.

We would like to thank Roborace for the opportunity to evaluate our algorithms on their prototype as well as for their support during the testing sessions. Furthermore, we thank Johannes Strohm, Matthias Rowold, Leon Sievers, Lukas Schäfer and Tim Stahl for the many discussions on the topic and the critical revision of the manuscript.

REFERENCES

[1] S. Pendleton, H. Andersen, X. Du, X. Shen, M. Meghjani, Y. Eng, D. Rus, and M. Ang, "Perception, planning, control, and coordination for autonomous vehicles," *Machines*, vol. 5, no. 1, 2017.

[2] M. Werling, "Ein neues Konzept für die Trajektoriengenerierung und -stabilisierung in zeitkritischen Verkehrsszenarien," *KIT Scientific Publishing, Karlsruhe*, vol. 60, 2011.

[3] A. Liniger, "Path Planning and Control for Autonomous Racing," *ETH Zürich Research Collection*, 2018.

[4] V. A. Laurense, J. Y. Goh, and J. C. Gerdes, "Path-tracking for autonomous vehicles at the limit of friction," *Proceedings of the American Control Conference*, pp. 5586–5591, 2017.

[5] N. R. Kapania, "Trajectory Planning and Control for an Autonomous Race Vehicle," *Stanford Digital Repository*, 2016.

[6] A. Liniger, X. Zhang, P. Aeschbach, A. Georghiou, and J. Lygeros, "Racing miniature cars: Enhancing performance using stochastic mpc and disturbance feedback," *Proceedings of the 2017 American Control Conference*, pp. 5642–5647, 2017.

[7] L. Hewing, A. Liniger, and M. N. Zeilinger, "Cautious nmpc with gaussian process dynamics for autonomous miniature race cars," *Proceedings of the 2018 European Control Conference*, pp. pp. 1341–1348, 2018.

[8] U. Rosolia, A. Carvalho, and F. Borrelli, "Autonomous Racing using Learning Model Predictive Control," *arxiv.org*, 2016.

[9] F. Berkenkamp, A. P. Schoellig, and A. Krause, "Safe controller optimization for quadrotors with Gaussian processes," *Proceedings of the 2016 IEEE International Conference on Robotics and Automation*, pp. 493–496, 2016.

[10] C. E. Rasmussen and C. K. I. Williams, *Gaussian Processes for Machine Learning*. The MIT Press, 2005.

[11] J. Betz, A. Wischnewski, A. Heilmeier, F. Nobis, T. Stahl, L. Hermansdorfer, and M. Lienkamp, "A software architecture for an autonomous racecar," in *2019 IEEE 89th Vehicular Technology Conference*. IEEE, Apr. 2019.

[12] A. Heilmeier, A. Wischnewski, L. Hermansdorfer, J. Betz, M. Lienkamp, and B. Lohmann, "Minimum curvature trajectory planning and control for an autonomous race car," *Vehicle System Dynamics*, pp. 1–31, Jun. 2019.

[13] W. F. Milliken and D. L. Milliken, *Race Car Vehicle Dynamics*. Great Britain: Society of Automotive Engineers Inc., 1996.

[14] F. Berkenkamp, A. Krause, and A. P. Schoellig, "Bayesian optimization with safety constraints: Safe and automatic parameter tuning in robotics," *arxiv.org*, 2016.

[15] J. Betz, A. Wischnewski, A. Heilmeier, F. Nobis, T. Stahl, L. Hermansdorfer, and M. Lienkamp, "What can we learn from autonomous level-5 motorsport?" in *Proceedings*. Springer Fachmedien Wiesbaden, Sep. 2018.