A Gaussian Process Model for Opponent Prediction in Autonomous Racing

Edward L. Zhu, Finn Lukas Busch, Jake Johnson, and Francesco Borrelli

Abstract—In head-to-head racing, performing tightly constrained, but highly rewarding maneuvers, such as overtaking, require an accurate model of interactive behavior of the opposing target vehicle (TV). We propose to construct a prediction model given data of the TV from previous races. In particular, a one-step Gaussian process (GP) model is trained on closed-loop interaction data to learn the behavior of a TV driven by an unknown policy. Predictions of the nominal trajectory and associated uncertainty are rolled out via a sampling-based approach and are used in a model predictive control (MPC) policy for the ego vehicle in order to intelligently trade-off between safety and performance when racing against a TV. In a Monte Carlo study, we compare the GP-based predictor in closed-loop with the MPC policy against several predictors from literature and observe that the GP-based predictor achieves similar win rates while maintaining safety in up to 3x more races. Through experiments, we demonstrate the approach in real-time on a 1/10th scale racecar platform operating at speeds of around 2.8 m/s, and show a significant level of improvement when using the GP-based predictor over a baseline MPC predictor. Videos of the experiments can be found at https://youtu.be/KMSs4ofDfIs.

I. INTRODUCTION

A major challenge in automated vehicles is the modeling of interactions between agents in highly dynamic constrained environments. Knowledge of such interactions is crucial for short-term decision making and can have a significant impact on the vehicle’s safety and performance. This is especially relevant in situations where information is not shared between agents and inference must be performed to obtain a prediction of future plans of the other agents in the environment. In this work, we consider the context of racing where agents are driven by competitive or possibly adversarial policies. In such scenarios, obtaining exact knowledge of an opponent’s dynamics and its racing policy is unlikely. However, it is certainly possible that data about them is available from past races. We therefore propose a Gaussian process (GP) based prediction model which learns the one-step closed-loop behavior of an opponent, or target vehicle (TV), from a database of interactions and obtains trajectory predictions and the associated uncertainties via sampling. To show the benefits of the proposed approach, we formulate a model predictive control (MPC) policy for the ego vehicle (EV) which leverages the time-varying uncertainties from the GP-based predictor by constructing uncertainty-expanded collision avoidance constraints to balance performance and safety in head-to-head races. We demonstrate through a Monte Carlo simulation study that the GP-based predictor achieves similar win rates while maintaining safety in up to 3x more races when compared to predictors in literature.

We expand upon a previous workshop paper by additionally demonstrating the approach in experiments on a 1/10th scale racecar platform operating at high speeds of around 2.8 m/s and show significant improvement over a baseline predictor. Related Work: Recent research in the construction of driver prediction models has focused on using learning-based methods for short-term trajectory prediction. Notably, methods such as [1], [2], and [3] infer high level human intent, predicting agent trajectories based on goal states given a semantic map. With respect to prediction under uncertainty, [1] models uncertainty in other road agents with a bivariate Gaussian over every future position, whilst considering short-term agent history. Much of the above work takes a model-free approach to trajectory forecasting by embedding interactions into a surrounding “scene context”. Alternatively, [4] proposes a graph-based prediction method that incorporates agent dynamic constraints and [5] explicitly embeds kinematic feasibility into the last layer of the prediction model. While these methods can certainly be applied in the context of racing, their use of semantic images as model inputs leads to the requirement of large motion datasets for training. In contrast, our approach leverages closed-loop state trajectories and can achieve good performance after training on a relatively small dataset of 5000 data points. There also exists much work which incorporates learned prediction models in closed-loop with optimal control. [6] maps features of the environment to strategy states through a GP, which are then used to construct MPC terminal constraints which drive the system to high performing regions of the state space given a static environment. [7] and [8] propose learning a mapping from vehicle state history to a finite set of behavioral parameters with a hidden Markov model and GP respectively. In [7] these parameters are used to modify an MPC reference whereas in [8] they are used to construct a measurement model which allows for trajectory predictions of other agents using an extended Kalman filter. Compared to these approaches, we not only condition the TV predictions on the current state of all agents, but also on the future plan of the EV. [9] is the most similar to our work, where a GP-based state transition model for the TV is learned and the distribution over the predicted trajectory is used in the construction of tightened half-space collision avoidance constraints for a stochastic MPC. The approach is demonstrated on straight track segments for a linearized kinematic vehicle model. Additionally, a “one-move” rule is assumed for the TV, which restricts its competitiveness. In contrast, we do not make the one-move assumption and formulate the approach for a nonlinear dynamic vehicle model and ellipsoidal collision avoidance constraints.
II. PROBLEM FORMULATION

A. Vehicle Model

We model the racing agents using the dynamic bicycle model [10], where the vehicle state and input vectors are defined as \( z = [p_x, p_y, \phi, v_x, v_y, \omega]^\top \) and \( u = [F_x, \delta]^\top \). \( p_x \) and \( p_y \) are the Cartesian coordinates of the vehicle’s center of gravity (CoG), \( \phi \) is the vehicle’s heading, \( v_x \) and \( v_y \) are the longitudinal and lateral velocity of the CoG respectively, and \( \omega \) is the yaw rate. The control inputs to the vehicle are the rear longitudinal tire force \( F_x \) and the steering angle \( \delta \). We discretize the nonlinear dynamics using the 4th order Runge-Kutta method with time step \( T_s \). We additionally make use of the vehicle pose in a curvilinear reference frame w.r.t. the centerline of a track [11], which is presented via the twice continuously differentiable mapping \( r : [0, L] \rightarrow \mathbb{R}^2 \), where \( L \) is the arc length of the track and \( \tau(0) = \tau(L) \). We denote the first and second derivatives of \( r \) w.r.t. the track progress \( s \) as \( \tau', \tau'' : [0, L] \rightarrow \mathbb{R}^2 \). The curvilinear pose can be computed from the global position \( u = [p_x, p_y]^\top \) and heading angle \( \phi \), and consists of the vehicle’s track progress \( s(p) = \arg \min_s \| \tau(s) - p \|_2 \), its lateral deviation from the centerline \( e_y(p) = \min_s \| \tau(s) - p \|_2 \), and its heading deviation from the centerline tangent angle \( e_y(p, \phi) = \phi - \arctan(\tau'(s(p))) \).

We can additionally define the signed curvature of the track at some \( s \in [0, L] \) as \( |\kappa(s)| = \| \tau''(s) \|_2 \), where the sign of \( \kappa \) depends on the sign of the swept angle rate for the track at \( s \). We distinguish between quantities for the EV and TV by adding a superscript, e.g. \( z(1) \) and \( z(2) \).

B. Model Predictive Contouring Control (MPCC)

Suppose that at time step \( k \), a TV trajectory prediction \( \bar{z}(k) = \{ z(2)_k, \ldots, z(k+N) \} \) is given. We define a baseline optimal control problem for the EV adapted from MPCC [12] to obtain the inputs \( u^{(1)}, u^{(2)} \) and \( u^{(1)}(1) \):

\[
\min_{u,v} \sum_{t=0}^{N-1} q_c^{(1)} e_c(p_t, \bar{s}_t) + q_e^{(1)} e(p_t, \bar{v}_t)^2 + u_t^T R u_t + \Delta u_t^T R_d^{(1)} \Delta u_t - q_u^{(1)} u_N^2 \quad (1a)
\]

s.t. \( z_0 = z^{(1)}_k \), \( \bar{s}_0 = s(p_0) \) \( (1b) \)
\( z_{t+1} = f^{(1)}(z_t, u_t), \quad t = 0, \ldots, N-1 \) \( (1c) \)
\( \bar{s}_{t+1} = \bar{s}_t + T_T \bar{v}_t, \quad t = 0, \ldots, N-1 \) \( (1d) \)
\( z_{t+1} \in Z, \quad u_t \in U, \quad t = 0, \ldots, N \) \( (1e) \)
\( h(z_t, u_{t+1}) \leq t, \quad t = 0, \ldots, N \) \( (1f) \)

where \( u = \{ u_0, \ldots, u_{N-1} \}, \bar{v} = \{ \bar{v}_0, \ldots, \bar{v}_{N-1} \}, \Delta u_t = u_t - u_{t-1}, \) and \( u_{-1} = u_0 \). The objective (1a) quantifies the goal of maximizing progress along a given track via the terms \( q_c, q_e, u \), which are approximations of \( s, e_y, \) and \( s \), respectively [12], and are weighted by parameters \( q_c(1), q_e(1), u_0(1) > 0 \). We additionally penalize amplitude and rate via \( R(1), R_d^{(1)} \geq 0 \). Vehicle state and approximate track progress evolution are governed by the dynamics in (1b)-(1d) and we impose track boundary and input constraints in (1e) via \( Z = \{ z | \ -W/2 \leq e_y(p) \leq W/2 \} \) and \( U = \{ u | \ -u \leq u \leq u_a \} \), where \( W \) is the width of the track. Constraint (1f) describes coupling between predicted EV and TV trajectories which, in this work, represent collision avoidance constraints. Using the optimal open-loop sequence from (1), we can define the EV feedback policy:

\[
\pi(1)(z^{(1)}_k, u^{(1)}(1)) = u_0^{(1)}(1) * (z^{(1)}_k, u^{(1)}(1)). \quad (2)
\]
Algorithm 1: TV trajectory prediction

Input: $N$, $M$, $z(2)$, $k$, $z(1)$, $k$

1. $z_{2,[i]}^{k} \leftarrow (z(2), z(1), k)\forall i = 1, \ldots, M$;
2. Sample $\Delta z_{k+1}^{(2),[i]} \sim \mathcal{N}(\mu_{z_{k+1}^{(2),[i]}}, \text{Var}_{z_{k+1}^{(2),[i]}})$;
3. $z_{k+1}^{(2),[i]} \leftarrow z_{k+1}^{(2),[i]} + \Delta z_{k+1}^{(2),[i]}$;

end

for $t = 0, 1, \ldots, N - 1$ do
4. $z_{k+1}^{(2),[i]} \leftarrow z_{k+1}^{(2),[i]} + e_{y,t}^{(2),[i]}(1)$
5. $\Sigma_{k+1}^{(2)} \leftarrow \Sigma_{k}^{(2)} + M^{-1} \sum_{i=1}^{M} (z_{k+1}^{(2),[i]} - z_{k+1}^{(2)}) (z_{k+1}^{(2),[i]} - z_{k+1}^{(2)})^{T}$;

end

Output: $\hat{z}_{k}^{(2)}, \Sigma_{k}^{(2)}, \ldots, \Sigma_{k+N}$

the EV’s state at time step $k$. During these data generation rollouts, the predictions $\hat{z}_{k}^{(2)}$ in the EV policy are replaced with the open-loop solution to (1) for the TV.

The regression features are constructed using the resulting closed-loop trajectories as $x_{i}[k] = [\Delta s_{k}, \Delta y_{k}, e_{x,k}, e_{y,k}, \phi_{k}, \psi_{k}, \sigma_{z_{k},k}, \kappa_{k}]^{T}$, where $\Delta s_{k} = s(p_{k}^{(2)}(t)) - s(p_{k}^{(1)}(t))$, $\Delta y_{k} = e_{y}^{(2)}(t) - e_{y}^{(1)}(t)$, and $\kappa_{k}$ is a vector of track curvatures at the look-ahead points $s(p_{k}^{(2)}(t)) + i\delta$, $i = 1, \ldots, V$ for some chosen $V$ and $\delta > 0$. The regression targets are chosen as the one-step TV state difference, which are transformed into the curvilinear frame via $y_{i}^{[k]} = C(z_{k+1}^{(2)}) - C(z_{k}^{(2)})$, where $C: \mathbb{R}^{6} \rightarrow \mathbb{R}^{6}$ denotes the invertible operator which transforms the vehicle’s kinematic pose in the global frame to the curvilinear frame. Evaluation of the predictor can be done with $c_{z_{k+1}^{(2)}} = C^{-1}(C(z_{k}^{(2)}) + \mu(z_{k}, z_{k}^{(1)}))$, where $\mu$ is the vector-valued function of the GP posterior mean.

C. Trajectory Prediction

After training the GP prediction model, we can now construct a trajectory prediction for the TV and the associated uncertainty over a finite horizon of length $N$, as conditioned on a given EV plan. We propose a straight-forward sampling-based approach which estimates a nominal TV state and approximately propagates the uncertainty based on $M$ rollouts of the one-step GP model. The procedure is described in Algorithm 1. As the rollouts are performed independently, our procedure is amenable to parallelization. We train a variational approximate GP [14] with independent output dimensions using an inducing point method with GPyTorch [15] on simulation rollouts with $q(2) = 200$. Executing Algorithm 1 with $M = 10$ rollouts yields an average total run time of 55 ms for a horizon of length $N = 10$ on an NVIDIA GeForce RTX 3080.

IV. DESIGN OF MPCC RACING POLICY

In this section, we introduce modifications to (1) for the EV, which allow it to leverage the uncertainties provided by Algorithm 1 to intelligently trade off between safety and performance. We represent the EV at time step $k$ as a set of four circles with radii $r_{i}$ and centers $c_{x,k}^{(1),i} = p_{x,k}^{(1)} + r_{i} \cos(\phi_{k}^{(1)})$ and $c_{y,k}^{(1),i} = p_{y,k}^{(1)} + r_{i} \sin(\phi_{k}^{(1)})$ for user-defined coverage constants $r_{i}$ and $i = 1, \ldots, 4$ such that the extents of the EV are contained within the union of these circles [16].

Fig. 1: Overview of the EV obstacle avoidance constraints with uncertainty-expanded ellipses pictured in red and vehicle extent ellipses pictured in blue. The EV is represented by 4 discs of radius $r_{d}$.

The TV is represented as the minimum covering ellipse of the vehicle extents, with semi-axis lengths $a$ and $b$. These are illustrated in Fig. 1 by the green circles and blue ellipses respectively. The obstacle avoidance constraint can be written for $j = 1, \ldots, 4$ and $t = k, \ldots, k + N$ as

\[
\begin{align*}
    &h_{j}(z_{k}^{(1),j}, z_{k+1}^{(2),j}) = 1 - (c_{x,t}^{(1),j} - p_{x,t}^{(2)})^{2}/a^{2} - ((c_{y,t}^{(1),j} - p_{y,t}^{(2)})^{2}/b^{2}.
\end{align*}
\]

We propose an uncertainty-expanded safety ellipse whose area is proportional to the position prediction variance from Algorithm 1. These are illustrated by the red ellipses in Fig. 1. Let $\text{Var}(s(t)^{(2)})$ and $\text{Var}(\hat{c}(t)^{(2)})$ be the prediction variances for track progress and lateral deviation on the diagonal of $\Sigma_{k+1}^{(2)}$. To derive the tightened constraint, we transform these variances from the track-aligned curvilinear frame to the body-aligned frame of the TV, where $\text{Var}(a(t)) = \text{cos}^{2}(\hat{c}(t))\text{Var}(\hat{c}(t)) + \sin^{2}(\hat{c}(t))\text{Var}(\hat{c}(t))$ and $\text{Var}(b(t)) = \text{sin}^{2}(\hat{c}(t))\text{Var}(\hat{c}(t)) + \text{cos}^{2}(\hat{c}(t))\text{Var}(\hat{c}(t))$ correspond to the variance in the TV’s longitudinal and lateral directions respectively. Note that we neglect off-diagonal terms in $\Sigma_{k+1}^{(2)}$ and the uncertainty in the heading. The semi-axis lengths of the expanded ellipse $a_{t}^{\ell}$ and $b_{t}^{\ell}$ are then computed as

\[
\begin{align*}
    a_{t}^{\ell} = \gamma \sqrt{\text{Var}(a(t))}, \quad b_{t}^{\ell} = \gamma \sqrt{\text{Var}(b(t))} (1 - \epsilon_{t}) + \frac{a}{b}.
\end{align*}
\]

where $\gamma > 0$ is chosen to be the number of standard deviations we would like to account for in our safety bound, and the constraint function (1f) is defined by replacing $a$ and $b$ in (5) with the semi-axis lengths computed in (6). We allow for a certain amount of constraint violation through the slack variables $\epsilon_{t} \in [0, 1]$. Note that when $\epsilon_{t} = 1$, the original constraint (5) is recovered. We add the quadratic term $\frac{1}{2} \epsilon_{t} Q_{t} \epsilon_{t} + q_{t} \epsilon_{t}$ to the cost function in (1a), where $Q_{t} \geq 0$ and $q_{t} \geq 0$. This disincentivizes the use of slack variables, i.e. violation of the uncertainty-expanded safety bound, but will allow for a level of risk to be taken when a significant improvement in performance can be achieved.

The optimal control problem is formulated using CasADi [17] and solved using sequential quadratic approximations with the QP solver hpipm [18]. Average solution times are 30 ms with a 2.6 GHz 9th-Gen Intel Core i7 CPU.

V. RESULTS AND DISCUSSION

Our approach is evaluated in simulation and hardware races, which match the EV, using the MPCC policy from Sec. IV, against a TV, using the blocking policy defined in Sec. III-B. Both policies use a horizon length of $N = 10$ and sample time of $T_{s} = 0.1$ s. In both simulation and hardware races, the EV starts the race behind the TV and is responsible...
model mismatch is evaluated. To simulate model mismatch, we vary the blocking aggressiveness \( q_y^{(2)} \) of the TV policy while keeping the predictors unchanged, i.e. no retraining or additional tuning is done. As a performance baseline, we include race results where the open-loop solutions of the TV policy are used in lieu of a prediction when computing the control action of the EV. This corresponds to the case where exact knowledge of the TV’s future plans are known to the EV, we call this the ground truth (GT) case. For each class of predictor, we additionally vary the size of the safety bound to investigate its effect on closed-loop race performance. For the GP, we pick safety bounds with \( \gamma = 0.5, 1, 2 \). For the CAV and NL predictors, circular safety bounds of radius 0.025, and 0.1 m are added. These are chosen such that the safety bound size on average is similar to the safety bounds induced by the GP’s uncertainties in the one and two standard deviation cases. The performance of the EV policy in closed-loop with each of these predictors against the TV blocking policy with various aggressiveness factor settings are depicted in Fig. 3.

In terms of win rate (upper left), GP clearly outperforms the other predictors for the TV policy that it was trained on, i.e. \( q_y^{(2)} = 200 \). For the other settings of \( q_y^{(2)} \), while GP does not impart a clear advantage in win rate, it remains competitive with the other predictors, including the GT case when we use the TV open-loop solutions directly instead of a predictor. In addition, GP seems to be less sensitive against variations in the size of the safety bound, likely because it provides consistently accurate predictions and does not have to rely on large safety bounds to keep the vehicle safe.

As for the crash rate (upper right), GP is able to maintain a crash rate below 10% for all values of \( q_y^{(2)} \), whereas the other predictors cannot keep the EV safe when the TV exhibits more aggressive blocking behavior. Interestingly, in terms of safety, GP is essentially able to achieve identical performance to GT where the TV’s future plans are known to the EV. Generally, we observe that GP outperforms the other predictors in terms of safety as it is able to accurately anticipate the TV’s blocking attempts and react accordingly by braking or steering out of the way, ultimately reducing the number of crashes.

While the ultimate objective of a race is to win, it is also worthwhile to look at the the outcome of races where the EV loses to the TV. In particular, we are interested in whether the EV was able to finish the race safely despite the loss or if the loss was caused by a crash. The reason for this concerns the practical aspects of car racing where crashes can be catastrophic and expensive. As such, consider two predictors A and B where Predictor A has a win rate of 40% and a crash rate of 5% and Predictor B has a win rate of 50% and a crash rate of 20%. Despite the lower win rate, it would be prudent to prefer Predictor A over Predictor B due to its significantly lower crash rate. In order to quantify this preference, we introduce the metric wins/crash, where a high score for this metric requires that the predictor not only lead to more wins but also few crashes. The results for this metric are shown in the lower left, where it is clear that GP outperforms the other predictors across all TV policies.

We finally examine the largest deceleration (averaged over all races) experienced by the EV under each predictor.
In this work, we presented a learning-based method for predicting opponent behavior in the context of head-to-head racing. The method uses a machine learning approach to predict the actions of the opposing vehicle (EV) based on its previous actions and the state of the race. The predictions are made using a Monte Carlo closed-loop simulation, which allows for the evaluation of the performance of the prediction models under various conditions.

The predictions are based on a set of features that include the longitudinal velocity of the EV, the longitudinal velocity of the TV (the vehicle being overtaken), and the relative position of the EV in relation to the TV. The predictions are then used to generate a policy for the EV to follow, which is designed to avoid collisions and maximize its chances of overtaking the TV.

The performance of the prediction models is evaluated using a variety of metrics, including win rate, crash rate, and the predicted behavior of the TV. The results show that the proposed method is effective in predicting the behavior of the TV, and that it is able to generate policies that lead to successful overtakes.

The results also indicate that the proposed method is able to adapt to the behavior of the TV, and that it is able to handle a wide range of scenarios, including those in which the TV is aggressive or passive.

In conclusion, the proposed method is a promising approach for predicting opponent behavior in the context of head-to-head racing, and it has the potential to be used in a variety of applications, including autonomous racing and driver assistance systems.
head competitive autonomous racing. In particular, Gaussian processes were trained on data from past races to generate trajectory predictions and their corresponding uncertainties, which are then used in closed-loop with an MPCC policy. We show through both simulation and hardware experiments that our approach outperforms non-data-driven predictors in terms of safety while maintaining a high win rate. Future work will focus on an extension to an arbitrary number of opponents and online fine-tuning of the predictor, which would allow for adaptation to previously unseen behavior.

REFERENCES

[1] J. Hong, B. Sapp, and J. Philbin, “Rules of the road: Predicting driving behavior with a convolutional model of semantic interactions,” 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp. 8446–8454, 2019.

[2] N. Djuric, V. Radosavljevic, H. Cui, T. Nguyen, F.-C. Chou, T.-H. Lin, N. Singh, and J. G. Schneider, “Uncertainty-aware short-term motion prediction of traffic actors for autonomous driving,” 2020 IEEE Winter Conference on Applications of Computer Vision (WACV), pp. 2084–2093, 2020.

[3] H. Zhao, J. Gao, T. Lan, C. Sun, B. Sapp, B. Varadarajan, Y. Shen, Y. Shen, Y. Chai, C. Schmid, C. Li, and D. Anguelov, “Tnt: Target-driven trajectory prediction,” in CoRL, 2020.

[4] T. Salzmann, B. Ivanovic, P. Chakravarty, and M. Pavone, “Trajectron++: Dynamically-feasible trajectory forecasting with heterogeneous data,” in ECCV, 2020.

[5] H. Cui, T. Nguyen, F.-C. Chou, T.-H. Lin, J. G. Schneider, D. Bradley, and N. Djuric, “Deep kinematic models for kinematically feasible vehicle trajectory predictions,” 2020 IEEE International Conference on Robotics and Automation (ICRA), pp. 10563–10569, 2020.

[6] C. Vallon and F. Borrelli, “Data-driven strategies for hierarchical predictive control in unknown environments,” ArXiv, vol. abs/2105.06005, 2022.

[7] S. Leffèvre, A. Carvalho, Y. Gao, H. E. Tseng, and F. Borrelli, “Driver models for personalised driving assistance,” Vehicle System Dynamics, vol. 53, no. 12, pp. 1705–1720, 2015.

[8] J. Lee, H. E. Tseng, P. Chakravarty, and D. F. B. Borrelli, “An optimization-based model predictive control framework for safe autonomous racing,” 2020 IEEE International Conference on Robotics and Automation (ICRA), pp. 10563–10569, 2020.

[9] T. Brüdigam, A. Capone, S. Hirche, D. Wollherr, and M. Leibold, “Gaussian process-based stochastic model predictive control for overtaking in autonomous racing,” ArXiv, vol. abs/2105.12236, 2021.

[10] J. Kong, M. Pfeiffer, G. Schildbach, and F. Borrelli, “Kinematic and dynamic vehicle models for autonomous driving control design,” IEEE Intelligent Vehicles Symposium, Proceedings, vol. 2015-August, pp. 1094–1099, 2015.

[11] A. Micaelli and C. Samson, “Trajectory tracking for unicycle-type and two-steering-wheels mobile robots,” Ph.D. dissertation, INRIA, 1993.

[12] A. Liniger, A. Domahidi, and M. Morari, “Optimization-based autonomous racing of 1:43 scale rc cars,” Optimal Control Applications and Methods, vol. 36, no. 5, pp. 628–647, 2015.

[13] C. K. Williams and C. E. Rasmussen, Gaussian processes for machine learning. MIT press Cambridge, MA, 2006.

[14] J. Hensman, A. G. de G. Matthews, and Z. Ghahramani, “Scalable variational gaussian process classification,” in AISTATS, 2015.

[15] J. Gardner, G. Pleiss, K. Q. Weinberger, D. Bindel, and A. G. Wilson, “GPytorch: Blackbox-matrix-matrix gaussian process inference with gpu acceleration,” Advances in neural information processing systems, vol. 31, 2018.

[16] W. Schwarting, J. Alonso-Mora, L. Paull, S. Karaman, and D. Rus, “Parallel autonomy in automated vehicles: Safe motion generation with minimal intervention,” 2017 IEEE International Conference on Robotics and Automation (ICRA), pp. 1928–1935, 2017.

[17] J. A. E. Andersson, J. Gillis, G. Horn, J. B. Rawlings, and M. Diehl, “CasADi – A software framework for nonlinear optimization and optimal control,” Mathematical Programming Computation, vol. 11, no. 1, pp. 1–36, 2019.

[18] G. Frison and M. Diehl, “Hpipm: a high-performance quadratic programming framework for model predictive control,” IFAC-PapersOnLine, vol. 53, no. 2, pp. 6563–6569, 2020.