Model Predictive Control of Autonomous Driving using Unscented Kalman Filter with Sparse Spectrum Gaussian Processes

Shitian Zhang, Yunduan Cui, Naitian Deng, and Huiyun Li

***Abstract*—In this paper, a model predictive control (MPC) approach that combines sparse spectrum Gaussian processes model and unscented Kalman Filter is proposed for path tracking task in autonomous driving. To tackle the difficulty of balancing control performance and computational cost in MPC with Gaussian processes model, the proposed approach employs the sparse spectrum Gaussian processes (SSGP) to efficiently model the vehicle, and utilizes unscented Kalman filter (UKF) to naturally propagate model uncertainties during multiple step prediction of MPC. The proposed approach is evaluated in both a numerical driving simulation and a mature driving simulation CARLA. The results indicate that the proposed method achieves a robust driving performance with a significant reduction of computational complexity.**

***Index Terms*—Autonomous driving, model predictive control, Gaussian processes, Kalman filter**

I. Introduction

With many successful implementations in autonomous driving [1]–[3], model predictive control (MPC) [4] utilizes a model to optimize and predict the vehicle motion in a finite time horizon and provides an intuitive way of controlling vehicle following a designed cost function. Since the performance of MPC highly depends on the quality of model, it is crucial to accurately model the vehicle dynamics, especially in noisy environments. The conventional works usually focus on modeling vehicles by linear model [1], nonlinear model [2] and neural networks [3] that require either the prior knowledge of the vehicles or a heavily tuning of parameters without considering the uncertainties from both system and environment in a fully Bayesian formalism.

Gaussian processes (GPs) [5] naturally describes the uncertainties of dynamics and environment through data in a nonparametric form and therefore becomes one potential solution to model system dynamics with uncertainties in MPC. On the other hand, the combination of GP and MPC turns to a difficulty of balancing the control frequency in MPC and the representative capability of model due to GP has a polynomial computational complexity $\mathcal{O}(n^3)$ and $\mathcal{O}(n^2)$ for time and space with n data points [6].

There are several works tring to tackling above balancing difficulty. The sparse GP (SGP) [7] was applied with MPC in various task including autonomous driving [8]–[10] and unmanned surface vehicles [11] with a strict limitation of sample size. Sparse spectrum GP (SSGP) [6] achieves better approximation efficiency through frequency domain space while the related work in autonomous driving is limited.

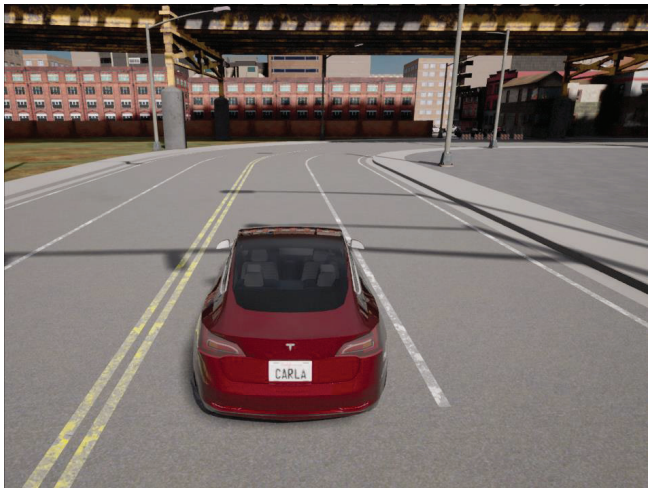

Fig. 1: The CARLA simulation environment used in this work.

In this paper, we propose SSGP-UKF integrates with MPC approach, a novel control framework specified for autonomous driving. It learns the vehicle dynamics from data by SSGP to balance control frequency and computational cost and propagates uncertainties using unscented Kalman filter (UKF) to increase the robustness against disturbances. After evaluating the control performance of path tracking task in a numerical autonomous driving simulation [12], the proposed framework is implemented and tested in a mature driving simulator CARLA [13] as shown in Figure 1. The experimental results show that SSGP-UKF significantly reduces the computational complexity with a less accuracy loss and successfully achieves path tracking task in CARLA simulator that requires controller with both high frequency and robustness. The contributions of this paper are the following:

1) Proposing a novel MPC framework for autonomous driving to propagate the uncertainties in prediction while balancing the prediction accuracy and the computational efficiency.
2) Evaluating the proposed framework on a mature driving simulation.

The remainder of this paper is organized as follows. The related works is introduced in Section II. Section III presents the proposed approach. Section IV details the experimental setting. The experimental results are presented in Section V. Conclusions follows in Section VI.

Shitian Zhang and Naitian Deng are with Shenzhen Institutes of Advanced Technology, Chinese Academy of Sciences, Shenzhen, China. University of Chinese Academy of Sciences, Beijing, China.

Yunduan Cui and Huiyun Li are with Shenzhen Institutes of Advanced Technology, Chinese Academy of Sciences, Shenzhen, China. (corresponding author: Huiyun Li, e-mail: hy.li@siat.ac.cn).

978-0-7381-1316-6/21/$31.00 © 2021 IEEE

II. Related Works

As one common solution to reduce the computational complexity of GP, SGP [7] directly search a subset, i.e., inducing points in the sample space. It was applied to several benchmark control problems including cart-pole and swing-up in [14] while ignoring the predicted variance of GP. The GP-based MPC was applied to both driving simulation and full-size autonomous race car in [8]–[10] where the extended Kalman filter (EKF) was added to propagate uncertainties of GP prediction as MPC constrains. Based on a given nonlinear vehicle model, this work employed GP to predict the model error rather than the whole dynamics. A data dictionary with limited size and sparse GP were utilized in this work to simplify the calculation. To further explore the power of uncertainty propagation in GP-based MPC, Ostafew et al. [15] utilized UKF with GP model to construct a robust MPC constrains in mobile robot path tracking. Kamthe et al. [16] first introduced analytic moment-matching [17, 18] into MPC framework. Cui et al [11] extended this framework to real-world unmanned surface vehicles control while suffering the limitation of data size and a relatively low control frequency.

Compared with conventional sparse GP approaches that search smaller inducing input set for better computational efficiency, the sparse spectrum GP (SSGP) [6] enjoys an unbiased approximation through frequency domain space using random kitchen sinks (RKS) [19] that precisely approximates a kernel function by randomly sampling features from frequency components of the Fourier transformed kernels following Bochner's theorem. Although Pan et al. have evaluated the SSGP-based MPC in an simulated vehicle drifting task [20], the investigation of SSGP-based MPC in autonomous driving still remains limited.

III. Approach

A. Model Learning

In this work, the vehicle model is learned by Gaussian Process (GP) [5] which is a collection of random variables, any finite number of which have a joint Gaussian distribution. Considering state $\boldsymbol{x} \in \mathbb{R}^D$, action $\boldsymbol{u} \in \mathbb{R}^U$, and system noise $\boldsymbol{w} \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{\Sigma}_w)$, a stochastic system dynamics of $\boldsymbol{x}$ follows:

$$\boldsymbol{x}_{t+1} = f(\boldsymbol{x}_t, \boldsymbol{u}_t) + \boldsymbol{w}. \tag{1}$$

Given the training input tuples $\tilde{\boldsymbol{x}}_t := (\boldsymbol{x}_t, \boldsymbol{u}_t)$, and targets $\boldsymbol{y}_t := \boldsymbol{x}_{t+1}$, for dimension $a = 1, ..., D$, we train a GP to model the latent function $y_t^a = f_a(\tilde{\boldsymbol{x}}_t) + w_a$ with a mean function $m_{f_a}(\cdot)$ and covariance kernel function:

$$k_a(\tilde{\boldsymbol{x}}, \tilde{\boldsymbol{x}}') = \alpha_{f_a}^2 \exp\big(-\frac{1}{2}(\tilde{\boldsymbol{x}} - \tilde{\boldsymbol{x}}')^\top \tilde{\boldsymbol{\Lambda}}_a^{-1}(\tilde{\boldsymbol{x}} - \tilde{\boldsymbol{x}}')\big), \tag{2}$$

where $\alpha_{f_a}^2$ is the variance of f_a, and $\tilde{\boldsymbol{\Lambda}}_a$ is the diagonal matrix of training inputs' length scales in kernel. They can be learned by evidence maximization [5, 21]. The prediction of a new input $\tilde{\boldsymbol{x}}_*$ follows:

$$p(f_a(\tilde{\boldsymbol{x}}_*)|\tilde{\boldsymbol{X}}, \boldsymbol{Y}^a) = \mathcal{N}(f_a(\tilde{\boldsymbol{x}}_*)|m_{f_a}(\tilde{\boldsymbol{x}}_*), \sigma_{f_a}^2(\tilde{\boldsymbol{x}}_*)), \tag{3}$$

$$m_{f_a}(\tilde{\boldsymbol{x}}_*) = \boldsymbol{k}_{a,*}^\top (\boldsymbol{K}^a + \alpha_{f_a}^2 \boldsymbol{I})^{-1} \boldsymbol{Y}^a = \boldsymbol{k}_{a,*}^\top \boldsymbol{\beta}_a, \tag{4}$$

$$\sigma_{f_a}^2(\tilde{\boldsymbol{x}}_*) = k_{a,**} - \boldsymbol{k}_{a,*}^\top (\boldsymbol{K}^a + \alpha_{f_a}^2 \boldsymbol{I})^{-1} \boldsymbol{k}_{a,*}, \tag{5}$$

where $\tilde{\boldsymbol{X}} = [\tilde{\boldsymbol{x}}_1, ..., \tilde{\boldsymbol{x}}_N]^\top$ is the training inputs set, $\boldsymbol{Y}^a = [\boldsymbol{y}_1^a, ..., \boldsymbol{y}_N^a]^\top$ is the collection of the training targets in corresponding dimension. $\boldsymbol{k}_{a,*} = k_a(\tilde{\boldsymbol{X}}, \tilde{\boldsymbol{x}}_*)$, $k_{a,**} = k_a(\tilde{\boldsymbol{x}}_*, \tilde{\boldsymbol{x}}_*)$, $K_{i,j}^a = k_a(\tilde{\boldsymbol{x}}_i, \tilde{\boldsymbol{x}}_j)$ is the corresponding element in $\boldsymbol{K}^a$, and $\boldsymbol{\beta}_a = (\boldsymbol{K}^a + \alpha_{f_a}^2 \boldsymbol{I})^{-1} \boldsymbol{Y}^a$.

To efficient calculate the learned GP model with large size of samples $(\tilde{\boldsymbol{X}}, \boldsymbol{Y})$, RKS [19] is applied in this work to efficiently approximate GP in frequency domain space. RKS proposed an unbiased approximation of a continuous shift-invariant kernel function based on Bochner's theorem. Define $\phi_\omega(\tilde{\boldsymbol{x}}) = e^{j\omega^\top \tilde{\boldsymbol{x}}}$ with a proper distribution $p(\omega)$, any continuous shift-invariant kernel function can be represented as:

$$k(\tilde{\boldsymbol{x}}_i, \tilde{\boldsymbol{x}}_j) = \int p(\omega) e^{j\omega^\top (\tilde{\boldsymbol{x}}_i - \tilde{\boldsymbol{x}}_j)} \mathrm{d}\omega. \tag{6}$$

Drawing a random sampling matrix $\boldsymbol{Z} \in \mathbb{R}^{M \times (D+U)}$ for $(D+U)$-dimensional data i.i.d. from $p(\omega) \sim \mathcal{N}(0, \boldsymbol{\Lambda}_a^{-1})$, RKS approximates Eq. 2 by a M-dimensional random features in frequency domain:

$$\begin{aligned} k_a(\tilde{\boldsymbol{x}}, \tilde{\boldsymbol{x}}') &\approx \boldsymbol{\Phi}_a(\tilde{\boldsymbol{x}})^\top \boldsymbol{\Phi}_a(\tilde{\boldsymbol{x}}'), \\ \boldsymbol{\Phi}_a(\tilde{\boldsymbol{x}}) &= \frac{\alpha_{fa}}{\sqrt{M}} \begin{bmatrix} \cos(\boldsymbol{Z}\tilde{\boldsymbol{x}}) \\ \sin(\boldsymbol{Z}\tilde{\boldsymbol{x}}) \end{bmatrix}. \end{aligned} \tag{7}$$

Following [6], the calculation of GP described in Section III-A was simplified by translating Eq. 5 to a linear Gaussian model [22] with weights $\boldsymbol{W}_a$ for each output dimension $a = 1, ..., D$:

$$\boldsymbol{W}_a \sim \mathcal{N}(\boldsymbol{\alpha}_a, \boldsymbol{\Sigma}_w \boldsymbol{A}_a^{-1}), \tag{8}$$

$$\boldsymbol{\alpha}_a = \boldsymbol{A}_a^{-1} \boldsymbol{\Psi}_a \boldsymbol{Y}^a, \tag{9}$$

$$\boldsymbol{A}_a = \boldsymbol{\Psi}_a \boldsymbol{\Psi}_a^\top + \boldsymbol{\Sigma}_w \boldsymbol{I}, \tag{10}$$

where $\boldsymbol{\Psi}_a = [\boldsymbol{\Phi}_a(\tilde{\boldsymbol{x}}_1), ..., \boldsymbol{\Phi}_a(\tilde{\boldsymbol{x}}_N)]$. The prediction of a new input $\tilde{\boldsymbol{x}}_*$ is approximated as:

$$m_{f_a}(\tilde{\boldsymbol{x}}_*) \approx \boldsymbol{\alpha}_a^\top \boldsymbol{\Phi}_a(\tilde{\boldsymbol{x}}_*), \tag{11}$$

$$\sigma_{f_a}^2(\tilde{\boldsymbol{x}}_*) \approx \boldsymbol{\Phi}_a(\tilde{\boldsymbol{x}}_*)^\top \boldsymbol{A}_a^{-1} \boldsymbol{\Phi}_a(\tilde{\boldsymbol{x}}_*). \tag{12}$$

B. Model Predictive Control with Unscented Kalman Filter

In this section, we propose SSGP-UKF, a MPC framework combining the sparse spectrum GP with UKF [23, 24] to efficiently propagate uncertainty during GP prediction.

Assume the input of GP model as a Gaussian distribution $\tilde{\boldsymbol{x}}_t \sim \mathcal{N}(\boldsymbol{\mu}_t, \boldsymbol{\Sigma}_t)$, SSGP-UKF first calculates the mean of $2D + 1$ sampling points using GP model:

$$\begin{aligned} \mathcal{X}_t &= (\boldsymbol{\mu}_t, \boldsymbol{\mu}_t + \epsilon\sqrt{\boldsymbol{\Sigma}_t}, \boldsymbol{\mu}_t - \epsilon\sqrt{\boldsymbol{\Sigma}_t}), \\ \bar{\mathcal{X}}_t^{[i]} &= \mathcal{GP}_\mu(\mathcal{X}_t^{[i]}, \boldsymbol{u}_t), \quad \text{for} \quad i = 0, ..., 2D. \end{aligned} \tag{13}$$

Algorithm 1: The process of SSGP-UKF

Input : training data set $\tilde{X}, Y$, MPC predict horizon H, cost function $l(\boldsymbol{x}, \boldsymbol{u})$, number of experiment N_{exp}, number of step N_{step}

for $i = 1, 2, ..., N_{exp}$ **do**
- $model$ = `TrainGP(`$\tilde{X}, Y$`)`
- `ResetVehicle()`
- **for** $j = 1, 2, ..., N_{step}$ **do**
  - $\boldsymbol{x}_{ref}$ = `CalculateReference()`
  - $\boldsymbol{x}'$ = `GetState()`
  - $\boldsymbol{x}^*$ = `BiasComp(`$\boldsymbol{x}'$`)`
  - $\boldsymbol{u}_j$ = `OptAct(`$\boldsymbol{x}^*, H, model, l(\boldsymbol{x}, \boldsymbol{u})$`)`
  - `OperateActions(`$\boldsymbol{u}_j$`)`

$\mathcal{GP}_\mu(\cdot)$ is the mean prediction in Eq. 11, ϵ is the parameter of UKF. The next state is then predicted as:

$$\begin{aligned} \boldsymbol{\mu}_{t+1} &= \sum_{i=0}^{2D} w_m^{[i]} \bar{\mathcal{X}}_t^{[i]}, \quad \boldsymbol{Q}_t = \mathcal{GP}_\Sigma(\mathcal{X}_t^{[i]}, \boldsymbol{u}_t), \\ \boldsymbol{\Sigma}_{t+1} &= \sum_{i=0}^{2D} w_c^{[i]} (\bar{\mathcal{X}}_t^{[i]} - \boldsymbol{\mu}_{t+1})(\bar{\mathcal{X}}_t^{[i]} - \boldsymbol{\mu}_{t+1})^\top + \boldsymbol{Q}_t. \end{aligned} \tag{14}$$

$\mathcal{GP}_\Sigma(\cdot)$ is the variance prediction in Eq. 12. The calculation of weights $w_m^{[i]}, w_c^{[i]}$ is detailed in [24].

Define a cost function $l(\cdot)$, e.g., the distance to the reference trajectory in autonomous driving, we have an optimization problem:

$$[\boldsymbol{u}_t^*, ..., \boldsymbol{u}_{t+H-1}^*] = \underset{\boldsymbol{u}_t, ..., \boldsymbol{u}_{t+H-1}}{\arg\min} \sum_{s=t}^{t+H-1} \gamma^{s-t} l(\boldsymbol{x}_s, \boldsymbol{u}_s), \tag{15}$$

where $\gamma \in [0, 1]$ is the discount factor. SSGP-UKF transfers Eq. 15 to an implicit feedback controller by re-planning the H-step open-loop control sequence at each coming state. The future states and their corresponding variances are predicted by the sparse spectrum GP (Section III-A) using UKF prediction in Eq. 13 & 14.

IV. Experimental Settings

In this section, we introduce SSGP-UKF into autonomous driving. For the training samples $(\tilde{\boldsymbol{X}}, \boldsymbol{Y})$, the input tuple includes both state and action, the output is the difference between states between the current and the next steps. The state of vehicle system is $\boldsymbol{x} = [P_x; P_y; v; \Phi]$ with the vehicle position (P_x, P_y), orientation Φ and velocity v. The action of the system is $\boldsymbol{u} = [a; \delta]$ with acceleration a and steering angle δ.

The whole process of SSGP-UKF is summarized in Algorithm 1. At the beginning of each experiment, the GP model is first trained by samples $(\tilde{\boldsymbol{X}}, \boldsymbol{Y})$. Then we initialize the vehicle and start control. The reference trajectory is generated by spline based path planner [25] with given state sequence. At each time step j, MPC optimize the action $\boldsymbol{u}_j$ based on a Bias Compensation state $\boldsymbol{x}^*$ rather than the current observation $\boldsymbol{x}'$ to reduce the effect of the previous action u_{j-1} which continuously affect the vehicle during MPC optimization in real world application and CARLA. We calculated $\boldsymbol{x}^*$ by GP one-step prediction.

The cost function is defined as the 2-norm error distance between the mean predicted state $\boldsymbol{x}_\mu$ and reference state $\boldsymbol{x}_{ref}$, with a penalty term of control signal $\boldsymbol{u}$ controlled by parameter α for better driving comfort, and with a penalty term of variance predicted state $\boldsymbol{x}_\Sigma$ controlled by parameter β for better driving safety:

$$l(\boldsymbol{x}, \boldsymbol{u}) = ||\boldsymbol{x}_\mu - \boldsymbol{x}_{ref}|| + \alpha||\boldsymbol{u}|| + \beta||\boldsymbol{x}_\Sigma||. \tag{16}$$

In this work, NLopt [26] is used to solve the MPC optimization problem, algorithm BOBYQA [27] is applied to minimize the cost function, and the number of maximum solver iterations was limited to 20 to ensure control accuracy and solve time. Additionally, during MPC optimization the system is subject to input constraints:

$$[0, -\delta_{max}] \leq [a, \delta] \leq [a_{max}, \delta_{max}], \tag{17}$$

i.e., the acceleration must be between zero and a_{max} and the steering angle is limited to a maximum angle δ_{max}.

We tested the proposed method in two simulation environments. The first one is a numerical driving simulation [12]. We pause this simulation during MPC optimization in Eq. 15 to remove the effect of computational cost in order to fairly compare the control performance between SSGP-UKF with other approaches based on original GP. 2.5% and 1% random white noises were introduced to the observation and control signal to simulate the environmental disturbances. SSGP-UKF is then evaluated in CARLA simulation [13] which is developed for simulating more complex vehicle dynamics. It runs in real time so that the computational complexity of MPC will hugely affect the control performance.

We focus on path tracking task in both two simulations. The MPC controller has a predictive horizon of $H = 3$ step with time step $\Delta t = 100$ ms, resulting in a 0.3 s look-ahead prediction. 280 and 500 samples were used to train the model in the numerical driving simulation and CARLA separately. The number of inducing points in SGP and SSGP were set to 10. We evaluate GP, SGP and SSGP with both EKF and UKF where EKF can be treat as pure GP prediction without propagation of uncertainties since its mean prediction is not affected by variance. All experiments were conducted on a PC with a 3.07GHz Intel Xeon X5675 CPU and 8GB RAM.

V. Experimental Results

A. Numerical Driving Simulation

In the numerical driving simulation, we first evaluated the driving performance of GP, SGP and SSGP with both EKF and UKF. According to the driving trajectories over 10-times trials demonstrated in Figs.2. 3 and 4, the UKF outperformed EKF in all GP settings. Since UKF naturedly considers the uncertainties during MPC, it generally enjoyed tracking behaviors with better consistency and safety compared with

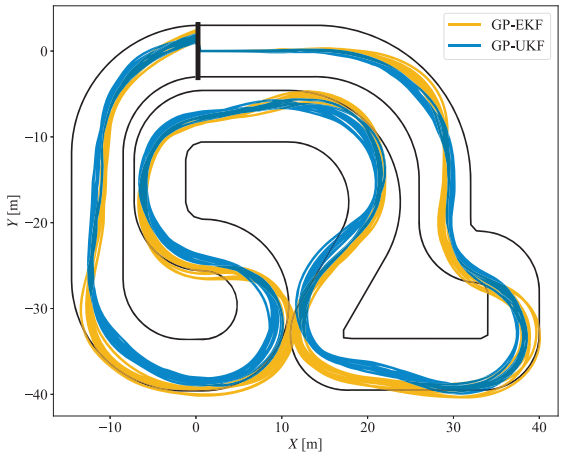

Fig. 2: Resulting trajectories of numerical driving simulation with noise using method based on GP.

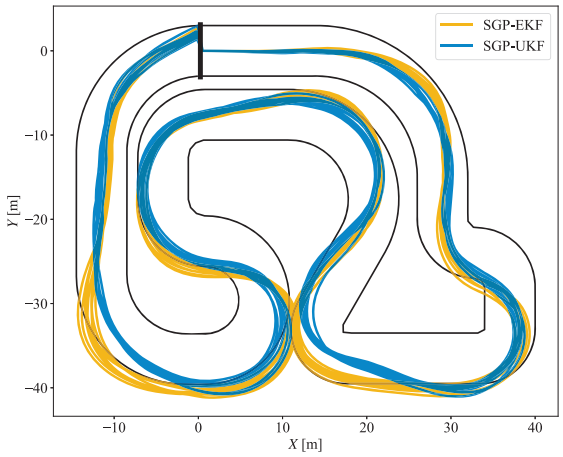

Fig. 3: Resulting trajectories of numerical driving simulation with noise using method based on SGP.

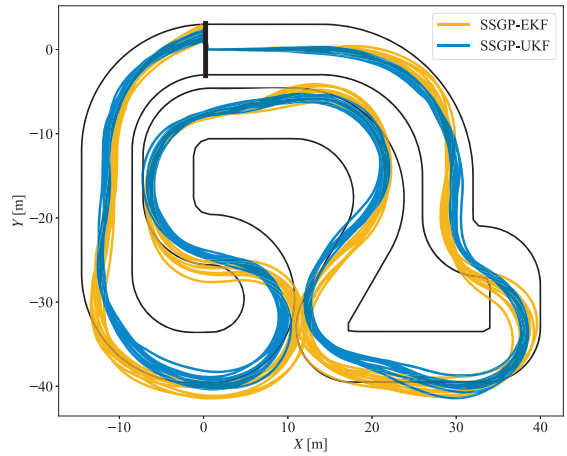

Fig. 4: Resulting trajectories of numerical driving simulation with noise using method based on SSGP.

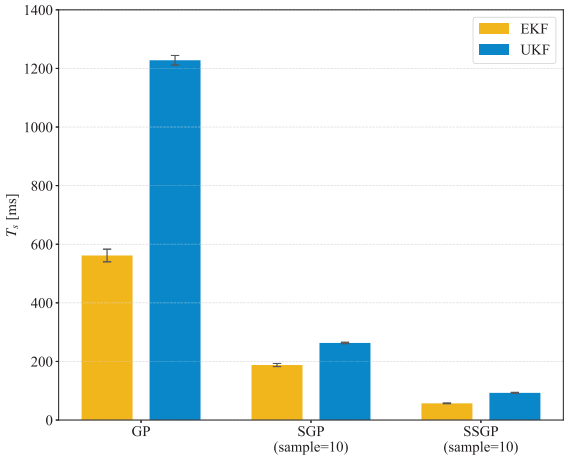

Fig. 5: The average solve time of numerical driving simulation experiment. The error bars are the standard deviation.

EKF. It is also clear that the original GP performed better than both SGP and SSGP without any approximation tricks. However, it therefore suffers a far higher computational cost as shown in Fig. 5. The UKF with original GP required about 1200 ms for one step MPC optimization which is infeasible in autonomous driving. As comparison, SGP with UKF required about 300 ms in optimization while SSGP only required near 100 ms, i.e., 10 Hz control frequency, which means a 67% reduction. The average prediction errors of vehicle position and orientation were shown in Figs. 6 and 7, where UKF had better prediction accuracy than EKF in all cases. Further more, the SSGP had a slightly lower average prediction error than SGP. Overall, it is clear that SSGP-UKF achieved the best balance between computational cost and driving performance in this experiment. It significantly reduced the optimization time using SSGP while maintaining lower accuracy loss in GP prediction by propagating uncertainties by UKF. This result indicates the potential of SSGP-UKF in real time autonomous driving.

B. CARLA Simulation

In the more complex CARLA simulator, we tackle a tracking task of a sigmoid path with approximately 200m length as shown in Fig. 8. The experiment was conducted in the map Town03 with vehicle model Tesla Model 3. All other

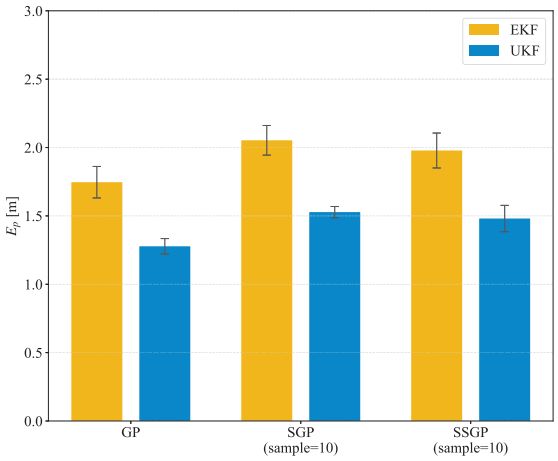

Fig. 6: The average position error of numerical driving simulation experiment. The error bars are the standard deviation.

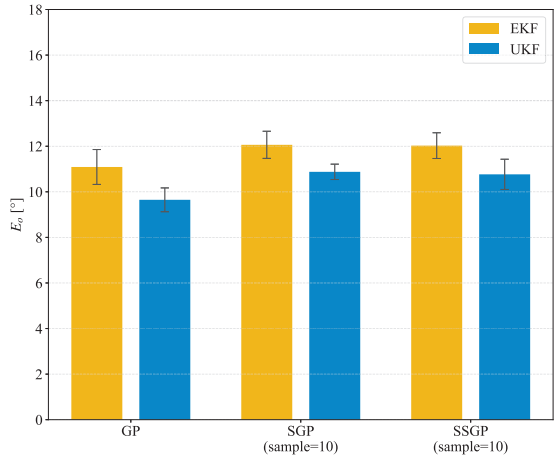

Fig. 7: The average orientation error of numerical driving simulation experiment. The error bars are the standard deviation.

settings follow Section IV. Since there is no pause during MPC optimization process, the approaches using GP and SGP could not meet the requirement of control frequency in CARLA due to the large computational cost shown in Fig. 5. Therefore, only SSGP with EKF and UKF were investigated in this experiment.

The driving performance over 10 times trials was listed in Table I where $\overline{T}_s$ is the average optimization time per step, $\overline{E}_p, \overline{E}_o$ are the average predicted error of vehicle position and orientation. T_l is the average time for finishing one lap. Although SSGP-UKF has larger average optimization time $\overline{T}_s$ than SSGP-EKF, it is still sufficient to a 10 Hz control frequency. On the other hand, SSGP-UKF enjoyed better model prediction accuracy in both $\overline{E}_p$ and $\overline{E}_o$. According to the driving trajectories shown in Fig. 9, SSGP-UKF could change its driving behavior in different trials based on its capability of propagating uncertainties and therefore outperformed SSGP-EKF.

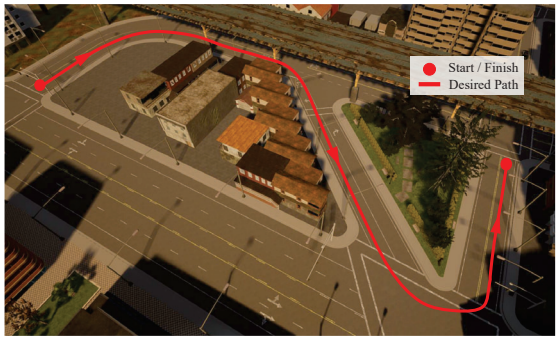

Fig. 8: Desired path in CARLA simulation

TABLE I: CARLA simulation results

Method	$\overline{T}_s$ [ms]	$\overline{E}_p$ [m]	$\overline{E}_o$ [°]	T_l [s]
SSGP-EKF	63.60±1.98	0.82±0.02	5.21±0.43	47.95±3.16
SSGP-UKF	98.85±2.90	0.75±0.07	4.74±0.83	49.51±3.82

The results in this section show that SSGP-UKF could significantly reduce the computational complexity while maintaining control safety in autonomous driving. The experiment in CARLA simulation further presents the potential of SSGP-UKF in real-time driving task with complex vehicle dynamic and its robustness against environmental disturbances.

C. Discussion

During both two experiments, the white noise was also added to the training samples to simulate the disturbances from both system and environment. We found that introducing this

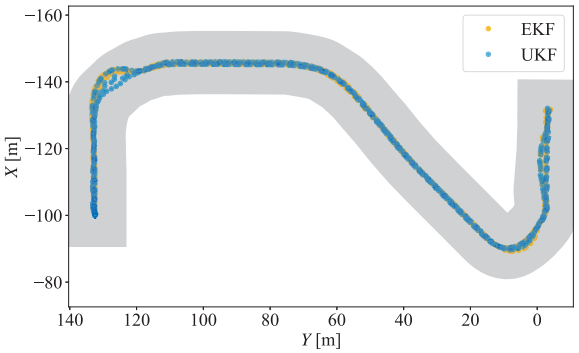

Fig. 9: Resulting trajectories of CARLA simulation using method based on SSGP.

noise could impact the performance of GP, SGP and SSGP and helps them to avoid falling into the local optimum in prediction. For a fair comparison, we fix the random seed during all experiments in this work. For the future work, we would like to extend the proposed approach to a reinforcement learning framework to automatically obtain new samples and iteratively learn driving skills. Furthermore, exploring the impacts of different optimization approaches in the SSGP-UKF is also an interesting topic.

VI. Conclusion

In this paper, we introduced SSGP-UKF integrates with MPC approach, a novel probabilistic control framework that directly tackles two challenging problems in autonomous driving: real-time control and disturbances. It models system dynamics using sparse spectrum Gaussian possesses for computational efficiency and propagates model uncertainties during multiple step prediction using unscented Kalman filter to naturally handle the disturbances from both vehicle models and environments. Evaluated in a numerical driving simulation, SSGP-UKF shows its advantages in balancing computational cost and prediction accuracy compared with other approaches including original Gaussian processes and sparse Gaussian processes with both UKF and EKF. It effectively reduces the control error in a noisy environment by considering the prediction variance in the MPC optimization. SSGP-UKF was further successful implemented in a mature driving simulation CARLA which requires high control frequency with more complex environment and vehicle model. Clearly it has potential towards a wide range of application scenarios in autonomous driving and other unmanned vehicle systems.

Acknowledgments

This work was supported by CAS Key Laboratory of Human-Machine Intelligence-Synergy Systems, Shenzhen Institutes of Advanced Technology, and Shenzhen Engineering Laboratory for Autonomous Driving Technology, the Science and Technology Development Fun, Macao S.A.R. (FDCT) No.0015/2019/AKP.

References

[1] J. Ji, A. Khajepour, W. W. Melek, and Y. Huang, "Path planning and tracking for vehicle collision avoidance based on model predictive control with multiconstraints," *IEEE Transactions on Vehicular Technology*, vol. 66, no. 2, pp. 952–964, 2016.

[2] 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.

[3] G. Williams, P. Drews, B. Goldfain, J. M. Rehg, and E. A. Theodorou, "Aggressive driving with model predictive path integral control," in *2016 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 1433–1440, IEEE, 2016.

[4] E. F. Camacho and C. B. Alba, *Model predictive control*. Springer Science & Business Media, 2013.

[5] C. E. Rasmussen and C. K. Williams, *Gaussian processes for machine learning*, vol. 1. MIT press Cambridge, 2006.

[6] M. Lázaro-Gredilla, J. Quiñonero-Candela, C. E. Rasmussen, and A. R. Figueiras-Vidal, "Sparse spectrum gaussian process regression," *The Journal of Machine Learning Research*, vol. 11, pp. 1865–1881, 2010.

[7] E. Snelson and Z. Ghahramani, "Sparse Gaussian processes using pseudo-inputs," in *Advances in neural information processing systems*, pp. 1257–1264, 2006.

[8] L. Hewing, A. Liniger, and M. N. Zeilinger, "Cautious nmpc with gaussian process dynamics for autonomous miniature race cars," in *2018 European Control Conference (ECC)*, pp. 1341–1348, 2018.

[9] J. Kabzan, L. Hewing, A. Liniger, and M. N. Zeilinger, "Learning-based model predictive control for autonomous racing," *IEEE Robotics and Automation Letters*, vol. 4, no. 4, pp. 3363–3370, 2019.

[10] L. Hewing, J. Kabzan, and M. N. Zeilinger, "Cautious model predictive control using gaussian process regression," *IEEE Transactions on Control Systems Technology*, 2019.

[11] Y. Cui, S. Osaki, and T. Matsubara, "Autonomous boat driving system using sample-efficient model predictive control-based reinforcement learning approach," *Journal of Field Robotics*.

[12] A. Sakai, D. Ingram, J. Dinius, K. Chawla, A. Raffin, and A. Paques, "Pythonrobotics: a python code collection of robotics algorithms," *arXiv preprint arXiv:1808.10703*, 2018.

[13] A. Dosovitskiy, G. Ros, F. Codevilla, A. Lopez, and V. Koltun, "CARLA: An open urban driving simulator," in *Proceedings of the 1st Annual Conference on Robot Learning*, pp. 1–16, 2017.

[14] J. Boedecker, J. T. Springenberg, J. Wülfing, and M. Riedmiller, "Approximate real-time optimal control based on sparse gaussian process models," in *Adaptive Dynamic Programming and Reinforcement Learning (ADPRL), 2014 IEEE Symposium on*, pp. 1–8, IEEE, 2014.

[15] C. J. Ostafew, A. P. Schoellig, and T. D. Barfoot, "Robust constrained learning-based nmpc enabling reliable mobile robot path tracking," *The International Journal of Robotics Research*, vol. 35, no. 13, pp. 1547–1563, 2016.

[16] S. Kamthe and M. Deisenroth, "Data-efficient reinforcement learning with probabilistic model predictive control," in *International Conference on Artificial Intelligence and Statistics*, pp. 1701–1710, 2018.

[17] A. Girard, C. E. Rasmussen, J. Q. Candela, and R. Murray-Smith, "Gaussian process priors with uncertain inputs application to multiple-step ahead time series forecasting," in *Advances in neural information processing systems*, pp. 545–552, 2003.

[18] M. P. Deisenroth, M. F. Huber, and U. D. Hanebeck, "Analytic moment-based Gaussian process filtering," in *the annual international conference on machine learning*, pp. 225–232, 2009.

[19] A. Rahimi and B. Recht, "Weighted sums of random kitchen sinks: Replacing minimization with randomization in learning," *Advances in neural information processing systems*, vol. 21, pp. 1313–1320, 2008.

[20] Y. Pan, X. Yan, E. A. Theodorou, and B. Boots, "Prediction under uncertainty in sparse spectrum gaussian processes with applications to filtering and control," in *International Conference on Machine Learning*, pp. 2760–2768, PMLR, 2017.

[21] D. J. MacKay, *Information theory, inference and learning algorithms*. Cambridge university press, 2003.

[22] C. K. Williams, "Prediction with gaussian processes: From linear regression to linear prediction and beyond," in *Learning in graphical models*, pp. 599–621, Springer, 1998.

[23] J. Ko, D. J. Klein, D. Fox, and D. Haehnel, "Gp-ukf: Unscented kalman filters with gaussian process prediction and observation models," in *2007 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 1901–1907, IEEE, 2007.

[24] S. Thrun, W. Burgard, and D. Fox, "Probabilistic robotics," 2005.

[25] K. Judd and T. McLain, "Spline based path planning for unmanned air vehicles," in *AIAA Guidance, Navigation, and Control Conference and Exhibit*, p. 4238, 2001.

[26] S. G. Johnson, "The nlopt nonlinear-optimization package [software]," *URL http://github.com/stevengj/nlopt*, 2014.

[27] M. J. Powell, "The bobyqa algorithm for bound constrained optimization without derivatives," *Cambridge NA Report NA2009/06, University of Cambridge, Cambridge*, pp. 26–46, 2009.