Moment-Based Model Predictive Control of Autonomous Systems

HanQiu Bao, Qi Kang, *Senior Member, IEEE*, XuDong Shi, *Graduate Student Member, IEEE*, MengChu Zhou, *Fellow, IEEE*, HaoJun Li, Jing An, *Member, IEEE*, and Khaled Sedraoui

***Abstract*—Great efforts have been devoted to the intelligent control of autonomous systems. Yet, most of existing methods fail to effectively handle the uncertainty of their environment and models. Uncertain locations of dynamic obstacles pose a major challenge for their optimal control and safety, while their linearization or simplified system models reduce their actual performance. To address them, this paper presents a new model predictive control framework with finite samples and a Gaussian model, resulting in a chance-constrained program. Its nominal model is combined with a Gaussian process. Its residual model uncertainty is learned. The resulting method addresses an efficiently solvable approximate formulation of a stochastic optimal control problem by using approximations for efficient computation. There is no perfect distribution knowledge of a dynamic obstacle's location uncertainty. Only finite samples from sensors or past data are available for moment estimation. We use the uncertainty propagation of a system's state and obstacles' locations to derive a general collision avoidance condition under tight concentration bounds on the error of the estimated moments. Thus, this condition is suitable for different obstacles, e.g., bounding box and ellipsoid obstacles. We provide proved guarantees on the satisfaction of the chance-constraints corresponding to the nominal yet unknown moments. Simulation examples of a vehicle's control are used to show that the proposed method can well realize autonomous control and obstacle avoidance of a vehicle, when it operates in an uncertain environment with moving obstacles. It outperforms the existing moment methods in both performance and computational time.**

***Index Terms*—Autonomous control, chance constraint, Gaussian process, intelligent vehicle, model predictive control, safety.**

Manuscript received 15 September 2022; revised 23 November 2022 and 2 January 2023; accepted 15 January 2023. Date of publication 23 January 2023; date of current version 19 May 2023. This work was supported in part by the National Natural Science Foundation of China under Grant 51775385, in part by the Strategy Research Project of Artificial Intelligence Algorithms of Ministry of Education of China under Grant 000011, in part by Shanghai Industrial Collaborative Science and Technology Innovation Project under Grant 2021-cyxt2-kj10, in part by the Innovation Program of Shanghai Municipal Education Commission under Grant 202101070007E00098, in part by the Ministry of Science and Higher Education of the Russian Federation as part of World-class Research Center program: Advanced Digital Technologies under Grant 075-15-2020-903, and in part by the Deanship of Scientific Research (DSR), King Abdulaziz University, Jeddah, Saudi Arabia under Grant RG-12-135-43. *(Corresponding author: Qi Kang.)*

HanQiu Bao, Qi Kang, and XuDong Shi are with the Department of Control Science and Engineering, Tongji University, Shanghai 201804, China (e-mail: 1910637@tongji.edu.cn; qkang@tongji.edu.cn; xdshi@tongji.edu.cn).

MengChu Zhou is with the Department of Electrical and Computer Engineering, New Jersey Institute of Technology, Newark, NJ 07102 USA, and also with Department of Cyber-Physical Systems, St. Petersburg State Marine Technical University, Lotsmanskaya St., 3, 198262 St. Petersburg, Russia (e-mail: zhou@njit.edu).

HaoJun Li is with the College of Surveying and Geo-Informatics, Tongji University, Shanghai 200092, China (e-mail: lhjch@tongji.edu.cn).

Jing An is with the School of Electrical and Electronic Engineering, Shanghai Institute of Technology, Shanghai 201418, China (e-mail: anjing@sit.edu.cn).

Khaled Sedraoui is with the ECE Department, King Abdulaziz University, 21589 Jeddah, Saudi Arabia (e-mail: sedraoui@yahoo.com).

This article has supplementary material provided by the authors and color versions of one or more figures available at https://doi.org/10.1109/TIV.2023.3238023.

Digital Object Identifier 10.1109/TIV.2023.3238023

I. Introduction

AUTONOMOUS system control technology for intelligent vehicles/robots has great significance in transportation, military, and many other application fields, e.g., self-driving cars or drones require autonomous control with safety guarantee [1]. In order to fully exploit their capabilities, we need to ensure that autonomous systems work safely in an open and dynamic environment while achieving high-level goals, such as completing their missions while reducing energy consumption [2], [3]. Model predictive control (MPC) can handle nonlinear constraints and be applied to nonlinear systems [4]. However, it relies on accurate model descriptions to enable safe and high-performance control [5], [6]. Model-identification, especially for nonlinear systems, is a time-consuming and complex, sometimes impossible, task. Therefore, an approximate model is often used, e.g., a linear model with adequate linearization, or simplified models, which, unfortunately, can degrade the control performance greatly [7]. To this end, a learning-based method of system dynamics as opposed to disturbance has first been presented [8]. The appeal of Gaussian Process (GP) regression for dynamical systems learning stems from the fact that it requires little prior process knowledge and directly provides a measure of residual model uncertainty [9], [10]. In [11], [12], a constrained tracking MPC technology for dealing with the uncertainty of the autonomous system based on GP prediction is presented.

While all these efforts can improve its control performance, an autonomous system operating in realistic scenarios must consider the environmental uncertainty on its autonomous control performance [13], [14], [15]. Hence, a predetermined reference path may not always satisfy an autonomous system's operational safety. Although a collision-free operation can be ensured, those methods tend to yield overly conservative autonomous system control while significantly reducing the autonomy and increasing the computational cost, especially as multiple obstacles appear [16], [17], [18]. Overall, the uncertainty of moving obstacles and the unmodeled part of dynamical systems represents two major challenging issue that influence the autonomous systems' performance in realistic scenarios [19].

It is substantially challenging and computationally expensive to implement a system's autonomous control when the obstacles' current and future positions are highly uncertain in dynamic environment [20], [21], [22], [23]. Due to the sensor noise and environmental uncertainty, an obstacle's position is a random variable. An autonomous system must operate in a partially known environment for most existing methods and can only perceive the environmental uncertainties online based on Gaussian stochasticity with known moments and a deterministic system model [24], [25], [26]. So, some solid assumptions need to be made to find tractable solutions, e.g., the uncertainty in scenarios is often assumed to follow Gaussian distributions [11], [12]. Generally, the obstacles are represented as a set of the constraints for autonomous system control, which produces a non-convex optimization problem [27], [28], [29], [30]. There are two main strategies to encapsulate an obstacle's space: Using a convex polyhedron [31], [32] (e.g., a cuboid) or a single differentiable surface [33], [34], [35] (e.g., an ellipsoid). It may, moreover, cause intractable problems due to the obstacle location uncertainty. To this end, using a probabilistic framework to overcome the set-bounded uncertainty is proposed in [36], [37]. However, linear chance constraints can only be reformulated as a deterministic constraint under a deterministic Gaussian moment. Nonlinear chance constraints approximated methods [34], [38], which require many samples to guarantee confidence feasibility, can lead to over-conservatism and high computational cost.

In this work, we propose a new MPC method via finite samples and moments (FMPC). It reaches a tradeoff between a data-driven approach to moment estimation and the known distribution assumption of uncertainty for autonomous system control in stochastic environments. In particular, we take on the fairly accepted model of a Gaussian distributed uncertainty, but both the dynamic obstacles with uncertain time-varying locations and the nominal system with an additive nonlinear part of the dynamics are considered during the prediction horizon. We formulate the collision avoidance condition based on uncertainty propagation as probabilistic chance constraints for an MPC problem; and further reformulate the chance constraint to a deterministic constraint by using the moment estimation. Our method can elude the over-conservatism of existing approaches while providing safety guarantees at low computational cost and greatly decreasing the risk of collision.

This work intends to make the following novel contributions:

1) It addresses an efficiently solvable approximate formulation of the stochastic MPC problem for an autonomous system to avoid obstacles with their uncertain locations and residual model uncertainty for the first time. The proposed method incorporates chance-constraints and considers an additive GP model. This paper also derives a deterministic approximation of a stochastic optimal control problem suitable for numerical optimization;
2) We derive a collision avoidance condition under tight concentration bounds on the error of the estimated moments and prove its generality for different shapes of obstacles under the risk allocation proposed in this paper.

We introduce the problem formulation in Section II. Then, we present a moment-based MPC framework and build our theoretical results in Section III. Next our approach is evaluated through two benchmarks and a simulation of a vehicle's autonomous control in Section IV and Section V. Finally, conclusions and future work are provided in Section VI.

II. Preliminaries

This section introduces the problem formulation and briefly reviews Gaussian Process Regression.

A. Problem Formulation

We consider the problem of autonomous system control with non-cooperative moving obstacles with uncertain localization, model, and disturbances in the form of additive Gaussian noise. For dynamic obstacles, as in [31], [34], we assume a time-varying model with Gaussian noise and known geometry of the scene.

Assumption 1: The dynamics of an obstacle and measurements of its state are available in each control step.

Thus, the dynamics of both an autonomous system and obstacles can be described as discrete-time and stochastic models:

$$x_{k+1} = \bar{f}\ (x_k, u_k) + B_G \left(\tilde{f}\,(x_k, u_k) + w_k \right) \tag{1a}$$

$$\tilde{x}^j_{k+1} = E_k\ \tilde{x}^j_k + F_k + \tilde{w}^j_k \tag{1b}$$

where $u_k \in \mathbb{R}^{n_u}$ and $x_k \in \mathbb{R}^{n_x}$ are the system input and system state sampled at time k, respectively. $\tilde{x}^j_k \in \mathbb{R}^{n_{\tilde{x}^j}}$ is the j-th obstacle state. $E_k \in \mathbb{R}^{n_{\tilde{x}^j} \times n_{\tilde{x}^j}}$ and $F_k \in \mathbb{R}^{n_{\tilde{x}^j}}$ are the obstacle dynamics matrices. $w_k \sim \mathcal{N}(0, \sigma_w^2 I)$ and $\tilde{w}^j_k \sim \mathcal{N}(0, \sigma^2_{\tilde{w}^j} I)$ are the unknown disturbances with i.i.d. Gaussian probability, which are spatially uncorrelated. $\bar{f}$ is a known nominal or priori model. $\tilde{f}$ is an initially unknown dynamics that can be learned from data and is assumed to lie in the subspace spanned by B_G. Given measured obstacle's position data, we estimate and predict their future positions and uncertainties with a linear Kalman filter [39].

We define $p_k \in \mathbb{R}^3$ and $\tilde{p}^j_k \in \mathbb{R}^3$ as the position of the autonomous system and j-th obstacles at time k, respectively. Notably, p_k and $\tilde{p}^j_k$ are the part of the state variable x_k and $\tilde{x}^j_k$, respectively. For system control, we need ensure an autonomous system operating in a safe area, which requires the autonomous system's position p_k subject to a safety set $\mathbb{x}^s_k$.

1) Ellipsoid Obstacle Region: Each obstacle is modeled as a non-rotating enclosing ellipsoid with semi-principal axes (a_o, b_o, c_o) and rotation matrix R_o. The autonomous system is modeled as an enclosing rigid sphere $\mathcal{S}_x$ with radius r_x. Thus, the safety set $\mathbb{x}^s_k$ is defined as the space outside the ellipsoid set, i.e.,

$$\mathbb{x}^s_k \triangleq \left\{ p_k \in \mathbb{R}^3 | \bigwedge_{j=1}^{N_o} \|\Theta \left(p_k - \tilde{p}^j_k \right) \| > r^j_s \right\} \tag{2}$$

The spherical collision region $\mathbb{x}^s_k$ can be enlarge into a half space $\widetilde{\mathbb{x}}^s_k$ [34], i.e.,

$$\tilde{\mathbb{x}}_k^s \triangleq \left\{ p_k \in \mathbb{R}^3 | \bigwedge_{j=1}^{N_o} a_{j,o}^{\mathrm{T}} \Theta \left(p_k - \tilde{p}_k^j \right) > r_s^j \right\} \tag{3}$$

where $r_s^j = 1$, $a_{j,o} = (p_k - \tilde{p}_k^j)/\|p_k - \tilde{p}_k^j\|$ and $\Theta = R_o^{\mathrm{T}} \operatorname{diag}\left(\frac{1}{r_x+a_o}, \frac{1}{r_x+b_o}, \frac{1}{r_x+c_o}\right)$, respectively. $\|\cdot\|$ and $\wedge$ denote the Euclidean norm and logical AND, respectively. N_o is the number of obstacles indexed by *j*.

2) Bounding Box Obstacle Region: Each obstacle is modeled as a bounding box with side length (l_{o1}, l_{o2}, l_{o3}). The autonomous system is modeled as an enclosing rigid sphere $\mathcal{S}_x$ with radius r_x. Thus, the safety set $\mathbb{x}_k^s$ is defined outside the bounding box set, i.e.,

$$\mathbb{x}_k^s \triangleq \left\{ p_k \in \mathbb{R}^3 | \bigwedge_{j=1}^{N_o} \bigvee_{i=1}^{3} \left| p_k^{(i)} - \tilde{p}_k^{j\,(i)} \right| > l_i^j \right\} \tag{4}$$

and this set can be safely approximated by its minimum volume enclosing ellipsoid [19], i.e.,

$$\tilde{\mathbb{x}}_k^s \triangleq \left\{ p_k \in \mathbb{R}^3 | \bigwedge_{j=1}^{N_o} a_{j,o}^{\mathrm{T}} \Theta \left(p_k - \tilde{p}_k^j \right) > r_s^j \right\} \tag{5}$$

where $l_i^j = \frac{l_{oi}+2r_x}{2}$, $r_s^j = 3$, $a_{j,o} = (p_k - \tilde{p}_k^j)/||p_k - \tilde{p}_k^j||$, $\Theta = R_o^{\mathrm{T}} \operatorname{diag}\left(\frac{1}{l_1^j}, \frac{1}{l_2^j}, \frac{1}{l_3^j}\right)$ and rotation angle is zero, and the Cartesian coordinates are iterated through index *i*. $|\cdot|$ and $\vee$ denote the absolute value and logical OR, respectively. N_o is the number of obstacles, with each obstacle indexed by *j*.

Due to the stochasticity of obstacles, we can define a safety chance constraint over the horizon length $N+1$ within a risk level $1-\alpha$ as follows:

$$\Pr\left(\bigwedge_{k=1}^{N+1} p_k \in \tilde{\mathbb{X}}_k^s \right) \geq 1 - \alpha \tag{6}$$

which can enforce the autonomous system to stay within the safe space in an acceptable safety level $1-\alpha$. Thus, autonomous system control can be formulated as a chance-constrained MPC problem:

$$J^* = \min_{\boldsymbol{U}} \sum_{t=1}^{N} l\left(x_{k,t}, u_{k,t}\right) + l_f\left(x_{k,N+1}\right) \tag{7a}$$

$$\text{s.t. } x_{k,t+1} = \bar{f}\ \left(x_{k,t}, u_{k,t}\right) + B_G\left(\tilde{f}\left(x_{k,t}, u_{k,t}\right) + w\left(k\right)\right) \tag{7b}$$

$$\boldsymbol{U} = \left[u_{k,1}, \ldots, u_{k,N}\right] \in \mathcal{U} \tag{7c}$$

$$\boldsymbol{X} = \left[x_{k,1}, \ldots, x_{k,N+1}\right] \in \mathcal{X} \tag{7d}$$

$$\Pr\left(\bigwedge_{t=1}^{N+1} p_{k,t} \in \tilde{\mathbb{x}}_k^s \right) \geq 1 - \alpha \tag{7e}$$

$$x_{k,1} = x_k\ ,\ x_{k,N+1} \in \mathcal{X}_f \tag{7f}$$

where $\boldsymbol{U}$ is a control vector. $x_{k,t}$ and $u_{k,t}$ are a predictive state and input at time step $k+t-1$, respectively. $l(\cdot)$ and $l_f(\cdot)$ are the stage cost function and terminal cost function at time *k*, respectively. $p_{k,t}$ and $\tilde{p}_{k,t}^j$ are the part of the state variable $x_{k,t}$ and $\tilde{x}_{k,t}^j$, at time step $k+t-1$, respectively. Usually, it is a weighted quadratic cost $||x||_Q^2 = x^{\mathrm{T}}Qx$ suitable for tracking tasks, i.e., $l(x_{k,t}, u_{k,t}) = ||x_{k,t}||_Q^2 + ||u_{k,t}||_R^2$ and $l_f\left(x_{k,N+1}\right) = ||x_{k,N+1}||_P^2$, where the Q, R and P are weighting matrices, and $Q \succ 0$, $R \succ 0$ and $P \succ 0$, respectively. N is the prediction horizon at time k. $\mathcal{U}$ and $\mathcal{X}$ are convex sets containing the origin. x_k is the system's current state at time *k*. The terminal constraint region $\mathcal{X}_f$ is a polytope. The MPC law can be obtained by solving (7), resulting in

$$u^*\ (x_k, k)\ = u_{k,1}^* \tag{8a}$$

$$\boldsymbol{U}_k^* = \left[u_{k,1}^*,\ u_{k,2}^*, \ldots, u_{k+N}^*\right] \tag{8b}$$

$$\boldsymbol{X}_k^* = \left[x_{k,1}^*,\ x_{k,2}^*, , \ldots, x_{k,N+1}^*\right] \tag{8c}$$

where $u_{k,1}^*$ is the first element of the computed optimal control sequence $\boldsymbol{U}_k^*$ applied to the system at time step *k*. $\boldsymbol{X}_k^*$ is the optimal trajectory sequence at time step *k*.

B. Gaussian Process Regression (GPR)

Due to its flexibility and inherent ability to describe uncertainty in function estimation, GPR is increasingly used for the modeling of nonlinear dynamical systems from data to improve a controller's performance. Thus, we use it to assess the assessment of residual model uncertainty with additive Gaussian noise, and generate the training dataset as follows:

$$\begin{aligned} \Omega\ &= \left\{ \boldsymbol{y} = [y_0, \ldots, y_n]^{\mathrm{T}} \in \mathbb{R}^{n\times 1}, \boldsymbol{z} \right. \\ &= \left. [z_0, \ldots, z_n]^{\mathrm{T}} \in \mathbb{R}^{n\times n_g} \right\} \\ z_i &= \left[x_i^{\mathrm{T}}, u_i^{\mathrm{T}}\right]^{\mathrm{T}} \\ y_i &= \tilde{f}\ (z_i) + \ w_i = B_G^{\dagger}\ \left(f\left(z_i\right) - \bar{f}\left(z_i\right)\right) \end{aligned} \tag{9}$$

where $w_i \sim \mathcal{N}(0, \sigma_w^2 I)$ is an i.i.d. Gaussian noise, and $z_i \in \mathbb{R}^{n_g}$ is a relevant feature to be used in the regression. $B_G^{\dagger}$ is the Moore–Penrose pseudoinverse. f is the true dynamical system function. y_i and z_i are the output and input data, respectively. We assume each y_i to be independent.

We model the initially unknown dynamics of a system as a Gaussian process:

$$\tilde{f}|z \sim \mathcal{N}\left(m\left(z\right), k\left(z, z'\right)\right)$$

where $m(z)$ and $k(z, z')$ are any valid functions that evaluate the mean and covariance of $\tilde{f}$, respectively.

The posterior distribution of $\tilde{f}$ evaluated at a test point z is also Gaussian [11] and given as:

$$f_G\left(z\right) \triangleq \left\{ \tilde{f}|z, \boldsymbol{z}, \boldsymbol{y} \sim \mathcal{N}\left(\mu^G\left(z\right), \Sigma^G\left(z\right)\right) \right\} \tag{10}$$

where

$$\begin{aligned} \mu^G\left(z\right) &= m\left(z\right) + K\left(z, \boldsymbol{z}\right)\left(K\left(\boldsymbol{z}, \boldsymbol{z}\right) + \sigma_w^2 I\right)^{-1} \\ &\quad \times \left(\boldsymbol{y} - m\left(\boldsymbol{z}\right)\right) \\ \Sigma^G\left(z\right) &= K\left(z, z\right) - K\left(z, \boldsymbol{z}\right)\left(K\left(\boldsymbol{z}, \boldsymbol{z}\right) + \sigma_w^2 I\right)^{-1} \\ &\quad \times K\left(\boldsymbol{z}, z\right) \\ \left[\ K\left(\mathbf{z}, \mathbf{z}'\right)\right]_{ij} &= k\left(z_i, z_j'\right) \end{aligned}$$

μ^G and Σ^G are the mean and variance of the prior f_G, respectively.

The choice of kernel function $K(\cdot)$ and its parameterization are the determining factors for the inferred distribution of $\tilde{f}$ and are typically specified by using prior process knowledge and optimization, e.g., by optimizing the likelihood of the observed data points. In this paper, we use the Square Exponential Kernel (SE) function for better hyperparameter tuning:

$$k\ (z, z') = \sigma_f^2 \exp\left(-0.5(z - z')^{\mathrm{T}} M^{\mathrm{T}} (z - z')\right)$$

where $\sigma_f^2 \in \mathbb{R}^{n_g}$ and $M = \mathrm{diag}(l_1, l_2, \ldots, l_{ng}) \in \mathbb{R}^{n_g \times n_g}$ are the selected system's variable variance and positive diagonal length-scale matrix, respectively. Thus, we can express all hyperparameters as:

$$\theta = \left[l_1, l_2, \ldots, l_{ng}, \sigma_f^2, \sigma_n^2\right]$$

We have described GP regression for one-dimensional output. In this paper, we treat each output dimension as an independent GP (with possibly different hyper-parameters). The computational complexity of GP regression strongly depends on the number of data points n. A further significant reduction in computational complexity can be achieved by employing sparse GP approaches [5].

C. Uncertainty Propagation

As stated in this paper, unmodeled dynamics might harm the performance of MPC, because the predictions might deviate from the true dynamics of the system. Since GPR's flexibility and inherent ability to describe uncertainty in function estimation and a GP is always used to evaluate the residual model uncertainty [9], [10], [40], [41], [42], [43]. Among all probabilistic distributions, Gaussian is the most commonly used one because of the central limit theory and its convenience for implementation. To this end, a GP can be used to learn residual unknown dynamics to enhance predictions of the MPC. In fact, GP is a probability distribution over functions, such that most finite samples of function value are jointly Gaussian distributed. Note that, we assumed that $f_G(z_k)$ and z_k are independent. in this paper. Although this assumption might eventually deteriorate the prediction quality, this is a simple and computationally cheap approach, allowing a tractable MPC formulation as in [5]. Therefore, we use GPR to learn the moment of the unmodeled dynamics $\tilde{f}$ and uncertainty, and improve prediction accuracy for MPC:

$$x_{k+1} = \bar{f}\ (x_k, u_k) + B_G\left(f_G\left(z_k\right) + w_k\right) \tag{11}$$

where $z_k = [B_{z_x} x_k; B_{z_u} u_k]$ is selected from the state and input as the regression feature that affects the prediction accuracy. Disturbances and model uncertainty are modeled as a GP (Section II.B), and thus $f_G(\cdot)$is normally distributed $f_G(\cdot) \sim \mathcal{N}(\mu_G(\cdot), \Sigma_G(\cdot))$. Then, we consider the design of an MPC controller for system (1a) using the GP-based model (11), and the resulting stochastic control problem can be formulated as:

$$J^* = \min_{\boldsymbol{U}} \sum_{t=1}^{N} l\left(x_{k,t}, u_{k,t}\right) + l_f\left(x_{k,N+1}\right) \tag{12a}$$

$$\text{s.t. } x_{k,t+1} = \bar{f}\ (x_{k,t}, u_{k,t}) + B_G\left(f_G\left(z_k\right) + w_k\right) \tag{12b}$$

$$\boldsymbol{U} = [u_{k,1}, \cdots, u_{k,N}] \in \mathcal{U} \tag{12c}$$

$$\boldsymbol{X} = [x_{k,1}, \cdots, x_{k,N+1}] \in \mathcal{X} \tag{12d}$$

$$\Pr\left(\bigwedge_{t=1}^{N+1} p_{k,t} \in \tilde{\mathbb{X}}_k^s\right) \geq 1 - \alpha \tag{12e}$$

$$x_{k,1} = x_k\ ,\ x_{k,N+1} \in \mathcal{X}_f \tag{12f}$$

By using GPR to evaluate unmodeled dynamical systems, the predictions become random variables, which results in non-Gaussian distribution. In order to obtain practical approximations, we use a commonly-used approach, which can approximate a state, control input and nonlinear disturbance as joint Gaussian distribution at time step k. In order to quickly and accurately get the mean $\mathbb{E}(x_{k+1}) \triangleq \mu_{k+1}^x$ and covariance $\mathbb{V}\mathrm{ar}[x_{k+1}] \triangleq \Sigma_{k+1}^x$ w.r.t. the random variable of the system state prediction x_{k+1}. We use unscented transformation to approximate them, similar to Cubature Kalman filtering (CKF). This permits simple update equations for the state mean and variance and provides an approximation accuracy up to third-order Taylor expansion of high dimension nonlinear systems (11); while other approximation methods [44], [45] cannot provide a good tradeoff between approximation accuracy and computational complexity for high dimension nonlinear systems.

To predict $x_{k,t},\ t = 1, \ldots, N+1$, we define a state $\gamma_{k,t} = (\mu_{k,t}^x, \mu_G(z_{k,t}))$ representing the mean state and disturbance at time step k with uncertainty, $P_{k,t} = \mathrm{diag}(\Sigma_{k,t}^x, \Sigma_G(z_{k,t}))$. We compute $2n$ volume points, $\Gamma_{k,t,j} = (\chi_{k,t,j}, \psi_{k,t,j})$, where n is the freedom of the system state and $\chi_{k,t,j}$ and $\psi_{k,t,j}$ are the volume points of $x_{k,t}$ and $\mu_G(z_{k,t})$, respectively.

$$\Gamma_{k,t,j} = \gamma_{k,t}\ + \xi_j S_{k,t}\ j = 1, \ldots, 2n \tag{13}$$

where $S_{k,t}\ S_{k,t}^{\mathrm{T}} = P_{k,t}$ with $S_{k,t} \in \mathbb{R}^{2n \times 2n}$ derived from the Cholesky decomposition of $P_{k,t}$. The point where the sphere of the unit sphere intersects with its coordinate system can be chosen as volume point ξ_j. The volume points are then passed through the nonlinear model:

$$\chi_{k,t+1,j} = \bar{f}\ (\chi_{k,t,j}, u_{k,t}) + B_G \psi_{k,t,j} \tag{14}$$

where the $\bar{f}(\cdot)$ is our a priori model. We recombine the volume points into the predicted mean and uncertainty, i.e.,

$$\mu_{k,t+1}^x = \frac{1}{2n} \sum_{j=1}^{2n} \chi_{k,t+1,j} \tag{15a}$$

$$\Sigma_{k,t+1}^x = \frac{1}{2n} \sum_{j=1}^{2n} \chi_{k,t+1,j}^{\mathrm{T}} \chi_{k,t+1,j} - {\mu_{k,t+1}^x}^{\mathrm{T}} \mu_{k,t+1}^x + B_G \Sigma_w {B_G}^{\mathrm{T}} \tag{15b}$$

Now, we can reformulate (12) as follows:

$$J^* = \min_{\boldsymbol{U}} \sum_{t=1}^{N} l\left(\mu_{k,t}^x, u_{k,t}\right) + l_f\left(\mu_{k,N+1}^x\right) \tag{16a}$$

$$\text{s.t.} \quad (13\text{–}15) \tag{16b}$$

$$\boldsymbol{U} = [u_{k,1}, \cdots, u_{k,N}] \in \mathcal{U} \tag{16c}$$

$$\boldsymbol{X} = \left[\mu_{k,1}^x, \cdots, \mu_{k,N+1}^x\right] \in \mathcal{X} \tag{16d}$$

$$\Pr\left(\bigwedge_{t=1}^{N+1} \mu_{k,t}^p \in \tilde{\mathbb{X}}_k^s\right) \geq 1-\alpha \tag{16e}$$

$$\mu_{k,N+1}^x \in \mathcal{X}_f \tag{16f}$$

$$\mu_{k,1}^x = x_k,\ \Sigma_{k,1}^x = 0 \tag{16g}$$

where x_k is the current state. $p_{k,t} \sim \mathcal{N}(\mu_{k,t}^p, \Sigma_{k,t}^p)$ and $\mu_{k,t}^p$, $\Sigma_{k,t}^p$ are the part of the state variable $\mu_{k,t}^x$ and $\Sigma_{k,t}^x$, respectively. Note that, we also assume that at each time-step the measurement is perfect and therefore $\mu_{k,1}^x = x_k$, $\Sigma_{k,1}^x = 0$.

III. Moment-Based MPC Controller Design

This section introduces a framework of safety MPC. Due to the chance constraints, (16) is computationally intractable. Next, we present techniques for deriving an efficiently solvable approximation.

A. Chance Constraint Reformulation

Solving (16) is not tractable due to the chance constraints and random variables. Thus, we need to approximate the problem as a deterministic disjunctive program, which is then cast into a nonlinear program to be solved efficiently by existing solvers.

We need to transform non-convex disjunction (6) into a convex conjunction by rewriting it as:

$$\Pr\left(\bigwedge_{t=1}^{N+1}\bigwedge_{j=1}^{N_o} ||\Theta\left(p_{k,t} - \tilde{p}_{k,t}^j\right)|| > r_s^j\right) \geq 1-\alpha \tag{17}$$

It is easy to decompose the conjunctions of (17) into single-chance constraints by using Boole's Inequality Approach [31], [32]:

$$\bigwedge_{t=1}^{N+1}\bigwedge_{j=1}^{N_o} \Pr\left(\left\|\Theta\left(p_{k,t} - \tilde{p}_{k,t}^j\right)\right\| > r_s^j\right) \geq 1 - \alpha_j^{k,t} \tag{18}$$

We define the single chance constraints' probabilistic variables as $\alpha_j^{k,t} \in [0,1]$ for all t, j, and k that satisfy

$$\sum_{t=1}^{N+1}\sum_{j=1}^{N_o} \alpha_j^{k,t} \leq \alpha \tag{19}$$

In this paper, we design an adaptive risk allocation method (20) by using the nature of MPC.

$$\alpha_j^{k,t} = \begin{cases} \frac{\alpha/N_o}{2^{N-t_c+t}}, & t \leq t_c \\ \frac{\alpha/N_o}{2^{N-t+1}}, & t_c < t \leq N+1 \end{cases} \tag{20}$$

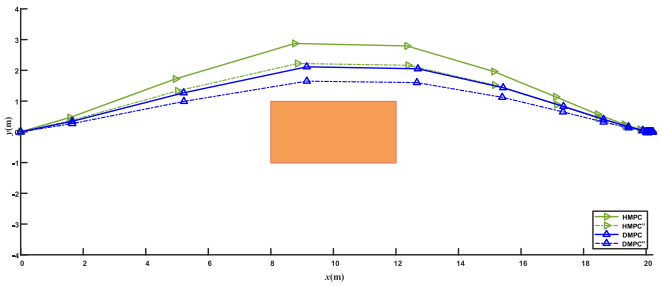

Fig. 1. The box obstacle has a width of 1 meter and length of 2 meters with uncertain location covariance $\Sigma_o = \text{diag}(0.2, 0.2)$. The solid line represents the adaptive probabilistic allocation, and the dotted line represents the uniform allocation. The horizon time is 8 s with the number of prediction steps $N = 20$ and $t_c = 4$.

where $t_c = \arg\min_t ||p_{k,t} - \tilde{p}_{k,t}^j||$ for $t = 1, \ldots, N+1$.

For comparison, we run the same simulation with risk uniform allocation by using two methods, i.e., hybrid-chance constraints using (20) (HMPC for short), the disjunctive-chance-constraints using (20) (DMPC), hybrid-chance constraints using uniform allocation (HMPC^u) and the disjunctive-chance-constraints using uniform allocation (DMPC^u) [19], [31]. The results are illustrated in Fig. 1. The simulations demonstrate that the adaptive risk allocation is less conservative than uniform allocation while improving safety distance to an obstacle.

Remark 1: The easiest way is to allocate risk uniformly [31]. It must be noted, however, that such uniform risk allocation can be overly conservative (or even lead to an infeasible optimization problem) in many cases. An autonomous system has different levels of risks along a trajectory, e.g., a vehicle has a higher risk when it is close to an obstacle and a lower risk when it is far away from an obstacle. Thus, an approach that uses our proposed risk allocation would retrieve a more optimal or/and less conservative solution. Our first result below shows that, for the problem at hand, one can significantly improve uniform risk allocation without needing additional optimization.

Lemma 1: Using the risk allocation (20), any feasible point for problem (18) (19) (20) is also feasible for problem (17).

Proof: We have:

$$\begin{aligned} \sum_{t=1}^{N+1}\sum_{j=1}^{N_o} \alpha_j^{k,t} &= \alpha\left(\frac{1}{2} + \cdots + \frac{1}{2^{N+1}}\right) \\ &\leq \alpha\left(\frac{1}{2} + \cdots + \frac{1}{2^{\infty}}\right) \\ &= \alpha\frac{\frac{1}{2} - \frac{1}{2}\times\frac{1}{2^{\infty}}}{1-\frac{1}{2}} \\ &\leq \alpha \end{aligned}$$

from which the statement of the lemma readily follows.

Lemma 2 [34]: Suppose that $X \sim \mathcal{N}(\mu, \Sigma)$, where $\mathcal{N}$ denotes a multivariate Gaussian distribution. $\forall \alpha \in (0,1]$ and $\forall (a,b) \in \mathbb{R}^{n_X}$, then

$$\Pr\left(a^{\mathrm{T}}X + b > 0\right) \geq 1-\alpha \tag{21}$$

can be written as a deterministic equivalent version of the following form:

$$a^{\mathrm{T}}\mu + b - \Psi^{-1}(1-2\alpha)\sqrt{2a^{\mathrm{T}}\Sigma a} \geq 0 \tag{22}$$

where Ψ is a standard error function:

$$\Psi(\epsilon) = \frac{2}{\sqrt{\pi}}\int_{-\infty}^{\epsilon}\exp\left(-\varsigma^2\right)d\varsigma \tag{23}$$

Gaussian distribution is the most commonly used one because of the central limit theory and its convenience for implementation. Given positions and uncertainty covariances of an autonomous system and obstacle $p_{k,t} \sim \mathcal{N}(\mu^p_{k,t}, \Sigma^p_{k,t}), \tilde{p}^j_k \sim \mathcal{N}(\tilde{\mu}^j_{k,t}, \tilde{\Sigma}^j_{k,t})$. Following the Lemma 2 and (17) can be transformed into a deterministic constraint as follows:

$$\begin{aligned}&\bigwedge_{t=1}^{N+1}\bigwedge_{j=1}^{N_o} {a_{j,o}}^{\mathrm{T}}\Theta\left(\mu^p_{k,t} - \tilde{\mu}^j_{k,t}\right) - r^j_s \\ &- \Psi^{-1}\left(1-2\,\alpha^{k,t}_j\right) \\ &\sqrt{2a_{j,o}{}^{\mathrm{T}}\Theta\left(\Sigma^p_{k,t} + \tilde{\Sigma}^j_{k,t}\right)\Theta^{\mathrm{T}}a_{j,o}} \geq 0\end{aligned} \tag{24}$$

where $a_{j,o} = (\mu^p_{k,t} - \tilde{\mu}^j_{k,t})/||\mu^p_{k,t} - \tilde{\mu}^j_{k,t}||$.

B. Moments-Based Chance Constraint Reformulation

In Section III.A, a hypothesis that a multivariate Gaussian random variable's moments are known for use to establish a deterministic equivalent version in (24). In fact, we cannot predict the moments of random variables in advance, but we can extract sample data of variables by sensors and estimate the moments from the data. Next, we derive the tight bounds of these uncertainties and a conservative reformulation of the chance constraints accordingly.

Lemma 3 [31]: Suppose that N_s i.i.d. samples data $\{\tilde{p}^j_{k,1}, \tilde{p}^j_{k,2}, \ldots, \tilde{p}^j_{k,N_s}\}$ are extracted from $\tilde{p}^j_k \sim \mathcal{N}(\tilde{\mu}^j_k, \tilde{\Sigma}^j_k)$, and

$$\hat{\mu}^j_k = \frac{\sum_{i=1}^{N_s}\tilde{p}^j_{k,i}}{N_s} \tag{25a}$$

$$\hat{\Sigma}^j_k = \frac{\sum_{i=1}^{N_s}\left(\tilde{p}^j_{k,i} - \hat{\mu}^j_k\right)\left(\tilde{p}^j_{k,i} - \hat{\mu}^j_k\right)^{\mathrm{T}}}{N_s - 1} \tag{25b}$$

where $\hat{\mu}^j_k$ and matrix $\hat{\Sigma}^j_k \succ 0$ are the estimated mean and covariance of $\tilde{x}^j_k$. Then, $\forall\,\varepsilon \in (0,1)$,

$$||\tilde{\mu}^j_k - \hat{\mu}^j_k|| \leq r^j_1 \tag{25c}$$

$$r^j_1 \triangleq \sqrt{\frac{T^2_{n,N_s-1}(1-\varepsilon)}{N_s\lambda_{\min}\left(\hat{\Sigma}^{j\,-1}_k\right)}} \tag{25d}$$

$$\left|x^{\mathrm{T}}\left(\tilde{\Sigma}^j_k - \hat{\Sigma}^j_k\right)x\right| \leq x^{\mathrm{T}}\hat{\Sigma}^j_k x r^j_2 \tag{25e}$$

$$r^j_2 \triangleq \max\left\{\left|1 - \frac{(N_s-1)}{\chi^2_{N_s-1,1-\frac{\varepsilon}{2}}}\right|, \left|1 - \frac{(N_s-1)}{\chi^2_{N_s-1,\frac{\varepsilon}{2}}}\right|\right\} \tag{25f}$$

where $T^2_{n,N_s-1}(1-\varepsilon)$ represents $(1-\varepsilon)$-th quantile of Hotelling's T-squared distribution with n, N_s-1. n is the dimension of $\hat{\Sigma}$. $\lambda_{\min}(\cdot)$ denotes the minimum eigenvalue of $\hat{\Sigma}^{-1}$. $\chi^2_{N_s-1,\frac{\varepsilon}{2}}$ represents $(\frac{\varepsilon}{2})$-th quantile of Chi-square distribution with N_s-1 degrees of freedom.

Lemma 4: Suppose that $\tilde{p}^j_{k,t} \sim \mathcal{N}(\tilde{\mu}^j_{k,t}, \tilde{\Sigma}^j_{k,t})$ and its samples data can be extracted. Then

$$\begin{aligned}&{a_{j,o}}^{\mathrm{T}}\Theta\left(\mu^p_{k,t} - \tilde{\mu}^j_{k,t}\right) - r^j_s \\ &- \Psi^{-1}\left(1-2\,\alpha^{k,t}_j\right)\sqrt{2a^{\mathrm{T}}_{j,o}\Theta\left(\Sigma^p_{k,t} + \tilde{\Sigma}^j_{k,t}\right)\Theta^{\mathrm{T}}a_{j,o}} \geq 0\end{aligned}$$

can be written as a deterministic equivalent version of the following form as (26) and it holds with a probability of at least $1-2\varepsilon$:

$$\begin{aligned}&\left(\left\|\mu^p_{k,t} - \hat{\mu}^j_{k,t}\right\| + r^j_{1,t}\right)\Psi^{-1}\left(1-2\,\alpha^{k,t}_j\right) \\ &\times\sqrt{2\left\|\Theta\left(\Sigma^p_{k,t} + \left(I + r^j_2\right)\hat{\Sigma}^j_{k,t}\right)\Theta^{\mathrm{T}}\right\|_F} \\ &+\left(\left\|\mu^p_{k,t}{}^{(1)} - \hat{\mu}^j_{k,t}{}^{(1)}\right\| + r^j_{1,t}{}^{(1)}\right) \\ &\left(\left\|\mu^p_{k,t}{}^{(2)} - \hat{\mu}^j_{k,t}{}^{(2)}\right\| + r^j_{1,t}{}^{(2)}\right)\|\Theta_{12} - \Theta_{21}\| \\ &-\sum_{i=1}^{3}\left(\left\|\mu^p_{k,t}{}^{(i)} - \hat{\mu}^j_{k,t}{}^{(i)}\right\| - r^j_{1,t}{}^{(i)}\right)^2\Theta_{ii} \\ &+ r^j_s\left(\left\|\mu^p_{k,t} - \hat{\mu}^j_{k,t}\right\| + r^j_{1,t}\right) \leq 0\end{aligned} \tag{26}$$

Proof: See Supplementary File.

Corollary 1: The bound of (26) can be tightened to:

$$\begin{aligned}&\lambda_{\min}(\Theta)\left\|\mu^p_{k,t} - \hat{\mu}^j_{k,t}\right\| - \left(\lambda_{\min}(\Theta)\,r^j_{1,t} + r^j_s\right) \geq \\ &\Psi^{-1}\left(1-2\,\alpha^{k,t}_j\right) \\ &\sqrt{2\left\|\Theta\left(\Sigma^p_{k,t} + \left(I + r^j_2\right)\hat{\Sigma}^j_{k,t}\right)\Theta^{\mathrm{T}}\right\|_F}\end{aligned} \tag{27}$$

where $\lambda_{\min}(\cdot)$ denotes the minimum eigenvalue.

Remark 2: Although the bound (26) derived in Lemma 4 holds for a general matrix Θ, Corollary 1 provides a tighter bound. Applying the Rayleigh quotient produces a strict lower bound, and it leads to a tighter bound (27) and a less conservative solution when the obstacles are modeled as an ellipsoid. It is easy to find that as the obstacles are modeled as a sphere, the bound (27) is equivalent to (26).

Then, we can reformulate (16) as:

$$J^* = \min_{\boldsymbol{U}}\sum_{t=1}^{N} l\left(\mu^x_{k,t}, u_{k,t}\right) + l_f\left(\mu^x_{k,N+1}\right) \tag{28a}$$

$$\text{s.t.}\ \ (13-15) \tag{28b}$$

$$\boldsymbol{U} = [u_{k,1}, \cdots, u_{k,N}] \in \mathcal{U} \tag{28c}$$

$$\boldsymbol{X} = \left[\mu^x_{k,1}, \cdots, \mu^x_{k,N+1}\right] \in \mathcal{X} \tag{28d}$$

$$\bigwedge_{t=1}^{N+1}\bigwedge_{j=1}^{N_o}(26) \tag{28e}$$

$$(19-20) \tag{28f}$$

$$\mu^x_{k,N+1}\in\mathcal{X}_f \tag{28g}$$

$$\mu^x_{k,1}=x_k\ ,\ \Sigma^x_{k,1}=\ 0 \tag{28h}$$

where $\boldsymbol{U}$ is a control vector. $x_{k,t}$ and $u_{k,t}$ are a predictive state and input at time step $k+t-1$, respectively. $l(\cdot)$ and $l_f(\cdot)$ are the stage cost function and terminal cost function at time k, respectively. $p_{k,t}$ and $\tilde{p}^j_{k,t}$ are the part of the state variable $x_{k,t}$ and $\tilde{x}^j_{k,t}$ at time step $k+t-1$, respectively. Usually, it is a weighted quadratic cost $||x||^2_Q=x^{\mathrm{T}}Qx$ suitable for tracking tasks, i.e., $l\ (x_{k,t},u_{k,t})=||x_{k,t}||^2_Q\ +||u_{k,t}||^2_R$ and $l_f\ (x_{k,N+1})=||x_{k,N+1}||^2_P$, where the Q,R and P are weighting matrices, and $Q\succ 0,\ R\succ 0$ and $P\succ 0$, respectively. N is the prediction horizon at time k. $\mathcal{U}$ and $\mathcal{X}$ are convex sets containing the origin. x_k is the system's current state at time k. The terminal constraint region $\mathcal{X}_f$ is a polytope. x_k is the current state. $p_{k,t}\sim\mathcal{N}(\ \mu^p_{k,t},\Sigma^p_{k,t})$ and $\ \mu^p_{k,t},\ \Sigma^p_{k,t}$ are the part of the state variable $\ \mu^x_{k,t}$ and $\Sigma^x_{k,t}$, respectively.

Theorem 1: Consider the sample estimates (25a) and (25b) under risk allocation (20), a solution of Problem (28) is a feasible solution to Problem (12) with a probability of at least $1-\ 2\varepsilon NN_o,\ \forall\,\varepsilon\in(0,\frac{1}{2})$ and $1-\ \varepsilon NN_o,\ \forall\,\varepsilon\in(\frac{1}{2},1)$. Given constraint (4) instead of (28e), if $Ns\to\infty$ and $t\to\infty$, then the solution of Problem (28) asymptotically converges to the solution of the exact moment problem.

Proof: See Supplementary File.

We have therefore shown that an MPC controller based on formulation (28) satisfies closed-loop chance constraints and is a feasible solution to the stochastic optimal control problem (12) in a probabilistic sense.

Remark 3: To safely operate an autonomous system, we need to predict the future positions of obstacles with quantifiable confidence. According to Assumption 1, we can estimate and predict their future positions and uncertainties with a linear Kalman Filter. But in practice, the moments of $\tilde{w}^j_k$ are not known exactly and thus we cannot quantify the probability of constraint violation. To this end, we utilize past sample data to get a confidence bound on the predicted trajectory of an obstacle.

The presented MPC optimization problem requires the cost evaluation of (12a). Depending on a specific form, such evaluation can be computationally expensive. For some cost functions, e.g., linear or quadratic one, however, this can be done efficiently based on the moments of the predicted state x_k, to be outlined next, together with a resulting quadratic cost bound.

C. Convergence Property

The following assumption and definitions are needed to provide a sufficient condition to prove the stability of the proposed methodology.

Assumption 2: If the initial problem (12) at time 0 is feasible, then problem (12) at time k is feasible for all $k>0$.

Definition 1: A continuous function $V(t,x)$ is a locally positive-definite function, if $V\ (t,0)=\ 0$ and

$$V\ (t,x)\geq\rho\left(|x_k|\right),\ \forall x\in\mathbb{B}_d,t\geq 0 \tag{29}$$

where $\mathbb{B}_d$ is a ball centered in the origin with radius d, and the function $\rho(\cdot)$ is continuous and strictly increasing with $\rho\ (0)=0$.

Consider the system (1a) with a terminal constraint $\mathcal{X}_f=\ 0$ and the value function is denoted as

$$V_N\ (k,x_k,u_k)=\sum_{t=1}^{N}l\ (x_{k,t},u_{k,t})\ +l_f\ (x_{k,N+1}) \tag{30}$$

where $\ l(x_{k,t},u_{k,t})=||x_{k,t}||^2_Q+||u_{k,t}||^2_R\ $ and $\ l_f(x_{k,N+1})=||x_{k,N+1}||^2_P$, where the Q,R and P are weighting matrices, and $Q\succ 0,\ R\succ 0$ and $P\succ 0$, respectively.

Remark 4: Assumption 2 is reasonable since for actual physical systems, an acceptable solution can usually be found for problem (7). It may be noted that the recursive feasibility is a standard result in MPC formulation. In this section, we mainly focus on the conditions for the uniform asymptotic stability of the origin of system (1a) under value function (30). For the more general stability conditions of an autonomous system, we consider for simplicity the desired set point as the terminal set for each state, i.e., a new value function $l(x_{k,t}-x_{des,k,t},u_{k,t})$ and $l_f(x_{k,N+1}-x_{des,k,N+1})$. x_{des} is the desired trajectory that has been planned offline.

Theorem 2: For the value function (30), there exists a function $\epsilon(\cdot)$ such that $V_N(k,x_k,u_k)\leq\epsilon(|x_k|)$,where function $\epsilon(\cdot)$ is continuous and strictly increasing with $\epsilon\ (0)=\ 0$.

Proof: It is easy to verify that the components of u_k and x_k are bounded by $\mathcal{U}$ and $\mathcal{X}$. Since the weight matrix $Q\succ 0,\ \mathrm{R}\succ 0$ and $P\succ 0$, we have: $V_N(k,x_k,u_k)\geq 0$ and it is bounded. Then we can always find a positive-definite function $\epsilon(\cdot)$, such that $V_N(k,x_k,u_k)\leq\epsilon(|x_k|),\ \forall x_k\in\mathcal{X}$.

In the following, we use $V_N(k,x_k,u_k)$ as a Lyapunov function to find a sufficient condition to prove the stability.

Theorem 3: Consider the system (11) with a terminal constraint $\mathcal{X}_f=\ 0$ and value function (30). If

$$l_f\ (x_{k+1,N+1})-l\ \left(x^*_{k,1},u^*_{k,1}\right)+\sum_{t=1}^{N}\left(l\left(x_{k+1,t}-x^*_{k,t+1},u_{k+1,t}-u^*_{k,t+1}\right)\right)\leq 0 \tag{31}$$

the origin of the closed-loop system is uniformly, locally asymptotically stable with the feedback control law.

Proof: See Supplementary File.

Remark 5: Condition (31) is established for the stated nonlinear system and problem (7), and it directly leads to an additional convex constraint to be embedded and incorporated in the required MPC design and algorithm. For problem (28), we just need to change $x_{k+1,t}$ into $\ \mu^x_{k,t}$ and we have:

$$l_f\ \left(\mu^x_{k+1,N+1}\right)-l\left(\mu^{x,*}_{k,1},u^*_{k,1}\right)+\sum_{t\ =\ 1}^{N}\left(l\left(\mu^x_{k+1,t}-\mu^{x,*}_{k,t+1},u_{k+1,t}-u^*_{k,t+1}\right)\right)\leq 0 \tag{32}$$

Algorithm 1: MPC of Autonomous Systems via Finite Samples.

Initialize the training dataset Ω of the autonomous system and train a GP model using (10); Initialize state vector of the autonomous system x_0; Initialize the desired trajectory x_{des}; Given samples $\{\tilde{w}^{j}_{k,t}{}^{(i)}\}_{i=1}^{N_s}$ for $k = 0, \ldots, \infty, t = 1, \ldots, N$ from past data or generative model of the j-th obstacle.
For $k = 0, \ldots, \infty$ **do**
  For $t = 1, \ldots, N$ **do**
    if $t = 1$ then
      Initialize $\hat{\mu}^{j}_{k,1}, \hat{\Sigma}^{j}_{k,1}$ to $\tilde{\mu}^{j}_{k,1}, \tilde{\Sigma}^{j}_{k,1}$.
    else
    Propagate samples $\left\{\tilde{x}^{j}_{k,t}{}^{(i)}\right\}_{i=1}^{N_s}$ using $\left\{\tilde{w}^{j}_{k,t}{}^{(i)}\right\}_{i=1}^{N_s}$ and (1b) to get $\left\{\tilde{x}^{j}_{k,t}{}^{(i)}\right\}_{i=1}^{N_s}$.
    Calculate sample moments $\hat{\mu}^{j}_{k,t}, \hat{\Sigma}^{j}_{k,t}$ using (25a) and (25b).
    Compute $\mu^{x}_{k,t+1}, \Sigma^{x}_{k,t+1}$ based on $\mu^{x}_{k,t}, \Sigma^{x}_{k,t}$ using (13)-(15).
    Generate the constraints of collision avoidance to moving obstacles by (26) and (20).
    Generate the additional constraint based on the stability condition (31).
    Compute the cost function by (30) and Remark 4.
    end
  end
  Solve Problem (28) and **return** control sequence $\boldsymbol{U}^{*}_{k}$.
  Apply **first input** $u^{*}_{k,1}$ of control sequence $\boldsymbol{U}^{*}_{k}$.
  Measure current system state to **update** as x_k.
end

Lemma 5: Any feasible solution of Problem (28) with the convex constraint (31) embedded is a feasible solution to Problem (12) with the convex constraint (31) embedded, and the optimal cost of Problem 28 is an upper bound on the optimal cost of Problem 12.

Proof: By comparing their constraints, we see that Problem 28 tightens Problem 12, from which the proof follows.

Remark 6: GPR has the flexibility and ability to describe consistent uncertainty estimates. It allows GP to estimate the residual dynamics through online learning from data, which can improve the prediction accuracy. Since the noise is assumed to be zero mean and it implies the prediction means is only infected by the GP-model's accuracy. Because of stochastic process noise and the additional GP model, future predicted states result in stochastic distributions. As the controller satisfies the boundary conditions on the stability, the solution of problem (28) can drive the states of the GP model to converge to a distribution. Its mean is the desired state and variance is the result of the uncertainty propagation.

Algorithm 1 is developed to obtain the optimal control inputs at time k and, thus, the optimal control sequence can be obtained. Next, we show our method's performance via a vehicle's autonomous control, obstacle avoidance and tracking control without online trajectory planning.

IV. Fixed Obstacle Avoidance

In this section, fixed obstacle avoidance problem is provided to verify the performance of the proposed avoidance method in Section III. All computations are carried out on an AMD Ryzen 7 5800H CPU at 3.20 GHz with 16 GB of memory using optimization framework Casadi.

A. Fixed Obstacle Benchmark With Deterministic Moment

In order to illustrate the advantages and feasibilities of our proposed algorithm of obstacle avoidance, we consider a vehicle modeled as a simple bicycle. The system states are $z = [x, y]^{\mathrm{T}}$ and system inputs are $u = [\varphi, v]$. Its dynamics is described as a differential equation:

$$\dot{z} = \begin{bmatrix} \dot{x} \\ \dot{y} \end{bmatrix} = \begin{bmatrix} \cos\varphi \\ \sin\varphi \end{bmatrix} v \tag{33}$$

where, x and y are the inertial coordinate system; φ is yaw angle; v is the vehicle speed. Δt is the discretization step.

In this section, we assume that the true system model (33) is known without disturbance, i.e., $\tilde{f}(z_k, u_k) = 0$ and $w_k = 0$. In practice, the discrete system model can be established using Euler's method as follows:

$$z_{k+1} = \bar{f}(z_k, u_k) \tag{34a}$$

$$\bar{f}(z_k, u_k) = z_k + \begin{bmatrix} \cos\varphi_k \\ \sin\varphi_k \end{bmatrix} v_k \times \Delta t \tag{34b}$$

We use a quadratic stage cost as in (28), with weight matrices $Q = I_2$, $R = 2I_2$, $P = I_2$ and horizon $N = 20$. The sample time $\Delta t = 0.4$ seconds.

We have compared it with three state-of-the-art methods, i.e., hybrid-chance constraints (HMPC for short) [19], robust-constraints (RMPC for short) [33] and the disjunctive-chance-constraints (DMPC for short) [31] on stochastic optimal collision avoidance for obstacle avoidance. The existing methods RMPC, however, cannot realize collision avoidance with indeterministic moments. So, we simulate with the deterministic moments for other methods and FMPC in this section.

Case 1: A vehicle and obstacle are placed at $[X_R, Y_R] = [0, 0]$ and $[X^1_o, Y^1_o] = [10, 0]$, respectively. The box obstacle has a width of 1 meter and length of 2 meters with uncertain location covariance $\Sigma^1_o = \mathrm{diag}(0.2, 0.2)$. The joint chance constraints are imposed with $\alpha = 0.01$ and $\varepsilon = 0.001$ over the whole horizon length. The horizon time is 8 s with the number of prediction steps $N = 20$ and $t_c = 4$.

Case 2: A vehicle and obstacle are placed at $[X_R, Y_R] = [0, 0]$ and $[X^1_o, Y^1_o] = [10, 0]$, respectively. The box obstacle has a width of 2 meters and length of 2 meters with uncertain location covariance $\Sigma^1_o = \mathrm{diag}(0.2, 0.2)$. The joint chance constraints are imposed with $\alpha = 0.01$ and $\varepsilon = 0.001$ over the whole horizon

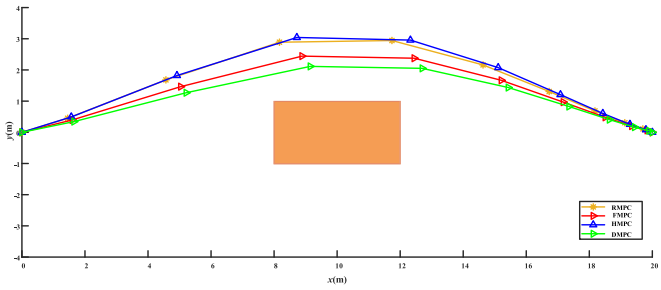

Fig. 2. Fixed box obstacle benchmark contrast for case 1.

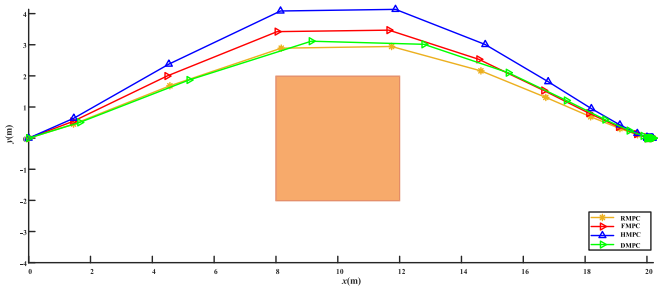

Fig. 3. Fixed box obstacle benchmark contrast for case 2.

TABLE I
THE COMPARISON OF CASE 1 AND CASE 2

	Case 1		Case 2	
Approaches	CPU Time(s)	Objective	CPU Time(s)	Objective
FMPC	1	1	1	1
RMPC[33]	1.3636	1.0191	1.1204	0.9816
DMPC[31]	102.5212	0.9918	94.9984	0.9811
HMPC[19]	1.1688	1.0123	1.2066	1.0187

length. The horizon time is 8 s with the number of prediction steps $N = 20$ and $t_c = 4$.

Case 3: A vehicle and obstacle are placed at $[X_R, Y_R] = [0, 0]$ and $[X_o^1, Y_o^1] = [10, 0]$, respectively. The ellipse obstacle's semi-principal axes size is $(2, 1)$ with uncertain location covariance $\Sigma_o^1 = \mathrm{diag}(0.2, 0.2)$. The joint chance constraints are imposed with $\alpha = 0.01$ and $\varepsilon = 0.001$ over the whole horizon length. The horizon time is 8 s with the number of prediction steps $N = 20$ and $t_c = 4$.

Case 4: A vehicle and obstacle are placed at $[X_R, Y_R] = [0, 0]$ and $[X_o^1, Y_o^1] = [10, 0]$, respectively. The ellipse obstacle's semi-principal axes size is $(2, 2)$ with uncertain location covariance $\Sigma_o^1 = \mathrm{diag}(0.2, 0.2)$. The joint chance constraints are imposed with $\alpha = 0.01$ and $\varepsilon = 0.001$ over the whole horizon length. The horizon time is 8 s with the number of prediction steps $N = 20$ and $t_c = 4$.

From Figs. 2 and 3, DMPC is the most conservative in Case 1 and RMPC is so in Case 2, but they are too close to the obstacle, which may lead to a collision. Although HMPC is the least conservative, FMPC has a better tradeoff between conservativeness and safety, which is less conservative than DMPC in Case 1 and less conservative than RMPC and DMPC in Case 2. Their control performances are shown in Table I. Our approach provides a conservative approximation to DMPC over

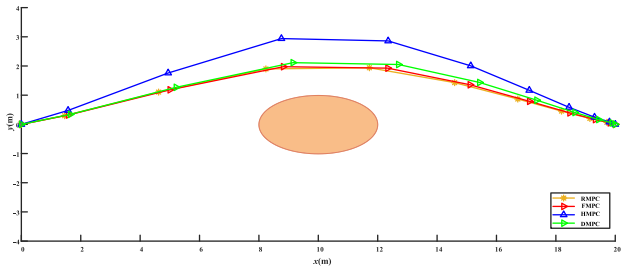

Fig. 4. Fixed ellipse obstacle benchmark contrast for case3.

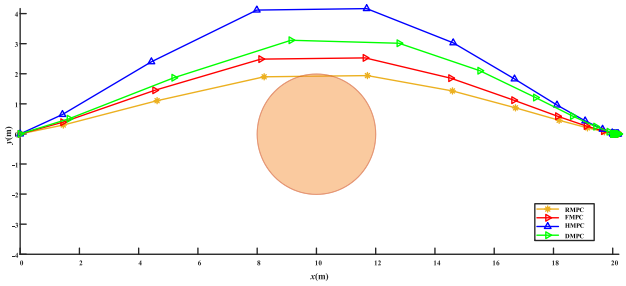

Fig. 5. Fixed ellipse obstacle benchmark contrast for case 4.

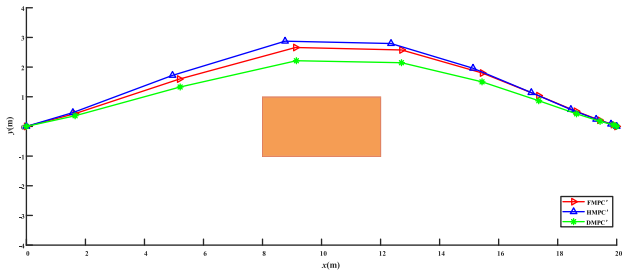

Fig. 6. Fixed ellipse obstacle benchmark contrast for case 5.

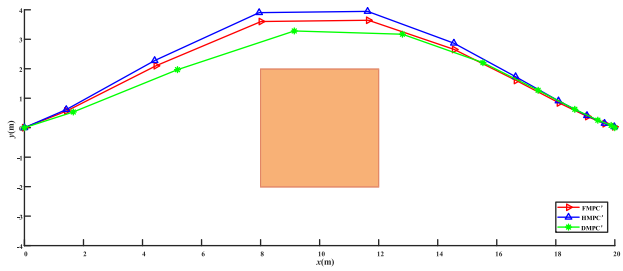

Fig. 7. Fixed ellipse obstacle benchmark contrast for case 6.

102.52 and 94.99 times faster at the expense of 0.82% and 1.89% off the optimality in Cases 1 and Case 2, respectively. It takes less time and cost than RMPC and HMPC in Cases 1 and 2.

From Figs. 4 and 5, FMPC is the most conservative in Case 3 and RMPC is the most conservative in Case 4. But the result of RMPC in Case 4 is too close to the obstacle, which may lead to a collision. FMPC has a better tradeoff between conservativeness and safety in Cases 3 and 4. Their control performances are shown in Table II. Our approach provides a

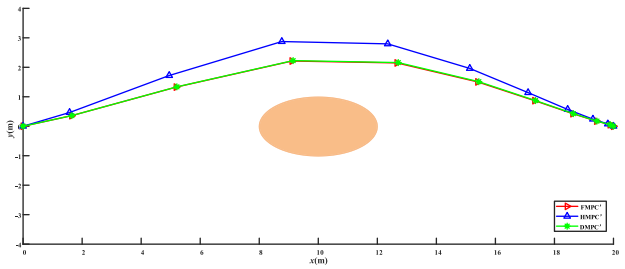

Fig. 8. Fixed ellipse obstacle benchmark contrast for case 7.

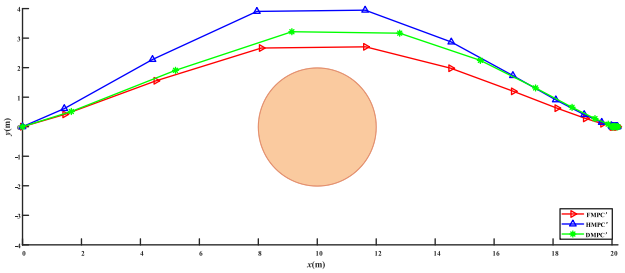

Fig. 9. Fixed ellipse obstacle benchmark contrast for case 8.

TABLE II
THE COMPARISON OF CASE 3 AND CASE 4

	Case 3		Case 4	
Approaches	CPU Time(s)	Objective	CPU Time(s)	Objective
FMPC	1	1	1	1
RMPC[33]	1.3697	1.0043	1.3166	0.9853
DMPC[31]	94.8251	1.0021	102.9641	1.0097
HMPC[19]	1.0650	1.0228	1.1754	1.0484

conservative approximation to RMPC over 1.31 times faster at the expense of 1.47% off the optimality in Case 4. It takes less time and cost than RMPC, DMPC and HMPC in Case 3 and DMPC and HMPC in Case 2, respectively. In general, FMPC is safer for a vehicle and can prevent its collision with an obstacle, and keep a reasonable safe distance from the obstacle, which is different from its three peers. By comparing their experimental results, we conclude that FMPC represents the best solution for safety and efficiency in real applications, dealing with different shapes of obstacles.

B. Fixed Obstacle Benchmark With Indeterministic Moment

The previous section has proved the effectiveness and feasibility of the method proposed in this article. The existing methods [34], however, cannot realize collision avoidance with indeterministic moments. So, we simulate the cases with indeterministic moments (HMPC', DMPC' and FMPC' for short) for comparison to show the generality of the proposed method.

In order to ensure autonomous system's operational safety in uncertain environments, we need to predict the future positions of an obstacle with quantifiable confidence. According to Assumption 1, this prediction can be done by propagating estimates means and covariance using a Kalman filter and (34). However, in practice, the moments of $\tilde{w}_k^j$ is usually unknown and

TABLE III
THE COMPARISON OF CASE 5 AND CASE 6

	Case 5		Case 6	
Approaches	CPU Time(s)	Objective	CPU Time(s)	Objective
FMPC'	1	1	1	1
DMPC'[31]	112.9254	0.9889	94.7505	0.9799
HMPC'[19]	1.2968	1.0124	0.9174	1.0161

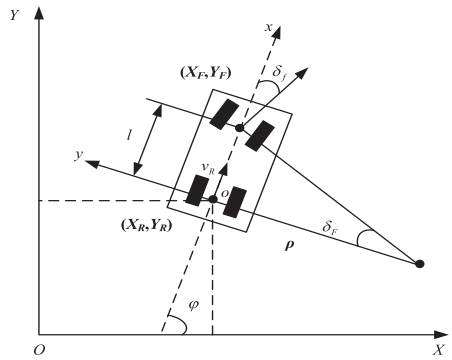

Fig. 10. Schematic diagram of particle model.

thus we cannot quantify the probability of constraint violation. Therefore, we use past sample data to get a confidence bound of the predicted trajectory of an obstacle. According to Lemma 4, we assume that we have N_s samples of an obstacle available for the next N steps of the prediction horizon, i.e., samples $\left\{\tilde{w}_{k,t}^{j\ (i)}\right\}_{i=1}^{N_s}$ for $k = 0, \ldots, \infty, t = 1, \ldots, N$ from the past data or generative model of the j-th obstacle. The availability of these samples is secured either through the use of a generative model or by considering that the obstacle maneuvers are usually extracted through a clustering algorithm based on previously collected sample data [31]. In both cases, a set of trajectory data $\left\{\tilde{x}_{k,t}^{j\ (i)}\right\}_{i=1}^{N_s}$ is available, from which the model uncertainty $\left\{\tilde{w}_{k,t}^{j\ (i)}\right\}_{i=1}^{N_s}$ can be calculated by comparison with the nominal response of linear model (34a). We use these samples directly to simulate the process model (34a) for N steps and to estimate the required uncertainty moments. Consequently, we can form the sample estimates (25a) and (25b) of the obstacle's predicted state.

Case 5: As same as Case 1
Case 6: As same as Case 2
Case 7: As same as Case 3
Case 8: As same as Case 4

From Figs. 6 and 7 and Table III, we have that FMPC' provides a conservative approximation of DMPC' over 112.92 and 94.75 times faster at the expense of 1.11% and 2.01% off the optimality in Cases 5 and 6. From Figs. 8 and 9 and Tables IV, FMPC' takes less time and cost than DMPC' and HMPC' in Case 7 and Case 8. By comparing experimental results of different methods, we conclude that FMPC has the ability to deal with uncertainty and different shapes of obstacles.

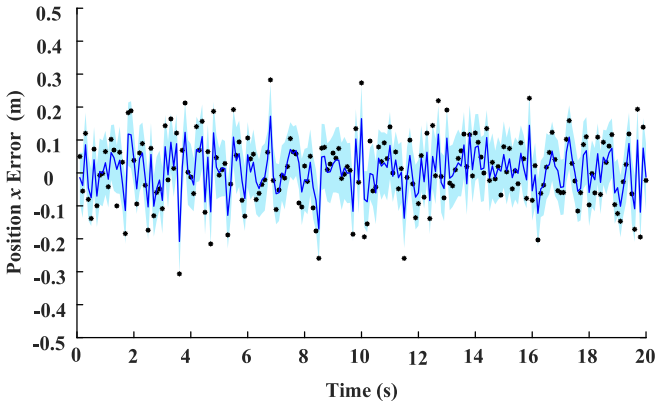

Fig. 11. Dynamic GP compensation of the position x error with 200 historical data collected offline. The black dots show the measured error on the position x under process noise as encountered at each time step, while the blue line shows the error predicted by the GP. The blue shaded region is the $2-\sigma$ confidence interval, including noise.

TABLE IV
THE COMPARISON OF CASE 7 AND CASE 8

	Case 7		Case 8	
Approaches	CPU Time(s)	Objective	CPU Time(s)	Objective
FMPC'	1	1	1	1
DMPC'[31]	101.0092	1.0003	109.7288	1.0081
HMPC'[19]	1.0686	1.0215	1.1225	1.0512

V. DYNAMICALLY CHANGING OBSTACLE AVOIDANCE

In this section, the vehicle model and obstacle model are described first. Dynamically changing obstacle avoidance problem is next provided to verify the performance of the proposed control method. All computations are carried out on an AMD Ryzen 7 5800H CPU at 3.20 GHz with 16 GB of memory using optimization framework Yalmip.

A. Vehicle Dynamics

We model a vehicle as a particle model which can provide a good trade-off between simplicity for real-time implementation and accuracy for high-performance control of an autonomous race car [20]. The model is based on a particle model illustrated in Fig. 10, with the system states $\xi = [X_R, Y_R, \varphi]^{\mathrm{T}}$ and system input $u = [v_R, \omega]$. Its dynamics is described as a differential equation:

$$\begin{bmatrix} \dot{X}_R \\ \dot{Y}_R \\ \dot{\varphi} \end{bmatrix} = \begin{bmatrix} \cos\varphi \\ \sin\varphi \\ 0 \end{bmatrix} v_R + \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \omega \tag{35}$$

where X, O and Y are the inertial coordinate system; x, o and y are the vehicle's coordinate system; φ and σ_F are its yaw angle and front wheel yaw angle, respectively; (X_R, Y_R) and (X_F, Y_F) are the front wheel center coordinate and rear wheel center coordinate, respectively; $l = 1$ meter is the wheelbase; ρ is the instantaneous turning radius of the rear wheel center; ω is the vehicle yaw rate; v_R is the vehicle speed. We just use a nominal model as the prediction model and using GPR to learn the unmodeled dynamics from the data generated by true model in the simulation.

B. Dynamic Obstacle Model

In this paper, the obstacles are modeled as nominal dynamics:

$$\tilde{x}^j_{k+1} = E_k\, \tilde{x}^j_k + F_k + \tilde{w}^j_k \tag{36a}$$

$$\tilde{y}^j_k = H_k\, \tilde{x}^{j,o}_k + \tilde{v}^j_k \tag{36b}$$

where $\tilde{x}^j_k \in \mathbb{R}^{n_{\tilde{x}^j}}$ is the j-th obstacle state and $\tilde{x}^j_0 \sim \mathcal{N}(0, \sigma^2_{\tilde{x}^j_o} I)$ is the initial state, $\tilde{y}^j_k \in \mathbb{R}^{n_{\tilde{y}^j}}$ is the measurement sampled at time k, respectively. $E_k \in \mathbb{R}^{n_{\tilde{x}^j} \times n_{\tilde{x}^j}}$ and $F_k \in \mathbb{R}^{n_{\tilde{x}^j}}$ the obstacle dynamics' matrices. $\tilde{v}^j_k \sim \mathcal{N}(0, \sigma^2_{\tilde{v}^j} I)$ and $\tilde{w}^j_k \sim \mathcal{N}(0, \sigma^2_{\tilde{w}^j} I)$ are white noise signals. We use a Kalman filter to estimate the position of an obstacle.

C. GP-Based Dynamic Obstacle Avoidance

In this example, we treat the case of model learning, that is we start with 200 data about the unknown model $\tilde{f}(\xi_k, u_k)$, and we can also collect measurement data during operation and enhance performance online. The discrete system model can also be established using Euler's method as follows,

$$\xi_{k+1} = \bar{f}\ (\xi_k, u_k) + \tilde{f}\ (\xi_k, u_k) + \begin{bmatrix} 0.1 \\ 0 \\ 0 \end{bmatrix} w_k \tag{37a}$$

$$\bar{f}\ (\xi_k, u_k) = \xi_k\ + \begin{bmatrix} 0.5 * \cos\varphi_k \\ \sin\varphi_k \\ 0 \end{bmatrix} v_k \Delta t + \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \omega_k \Delta t \tag{37b}$$

$$\tilde{f}\ (\xi_k, u_k) = \begin{bmatrix} 0.5 * \cos\varphi_k \\ 0 \\ 0 \end{bmatrix} v_k \Delta t \tag{37c}$$

where w is an i.i.d. zero-mean Gaussian process noise with variance $\Sigma_w = 1$ and it only affects the velocity states of X_R.

GP data $\Omega = \{\boldsymbol{y} = [y_0, \ldots, y_n]^{\mathrm{T}} \in \mathbb{R}^{n\times 1}, \quad \boldsymbol{z} = [z_0, \ldots, z_n]^{\mathrm{T}} \in \mathbb{R}^{n\times n_g}\}$ is generated by calculating the deviation of the true model from the measured states of nominal model as described in Section II-B where the initial state and input are generated randomly. The input data is selected as $z_k = [v_{X_k}, v_{Y_k}, \omega_k]^{\mathrm{T}}$. We employ a squared exponential kernel for output dimension with fixed hyperparameters $M = \mathrm{diag}(96.49, 0.1417, 9994)$, and variances $\sigma_f^2 = 0.0872$ and $\sigma_n^2 = 0.0084$. We use a quadratic stage cost as in (28), with weight matrices $Q = \mathrm{diag}(1, 10, 1)$, $R = 0.1 I_2$, $P = 20 I_3$, horizon $N = 10$, and I denotes identity matrix. The sample time $\Delta t = 0.2$ seconds. The constraint of input and state are $\mathcal{U} = \{u | [-20, -\pi/3]^{\mathrm{T}} \le u \le [20, \pi/3]^{\mathrm{T}}\}$ and $\mathcal{X} = \{x | [-\infty,\ 0,\ -\pi/2]^{\mathrm{T}} \le x \le [\infty,\ 4,\ \pi/2]^{\mathrm{T}}\}$, respectively.

Data is continuously updated during the closed-loop run. Specifically, since the computational complexity of GP regression strongly depends on the number of data points n, we have performed the following procedure: i) Set a size limit for the dictionary; ii) If the dictionary is full, select the so-called inducing points [5]; and iii) Choose the closest point in the Euclidean space to the new point to be replaced or add new

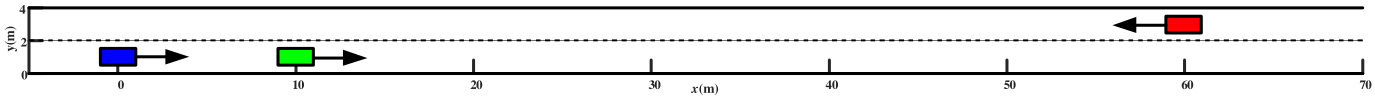

Fig. 12. Dynamic obstacle avoidance and speed scale on the right, we define left as negative and right as positive.

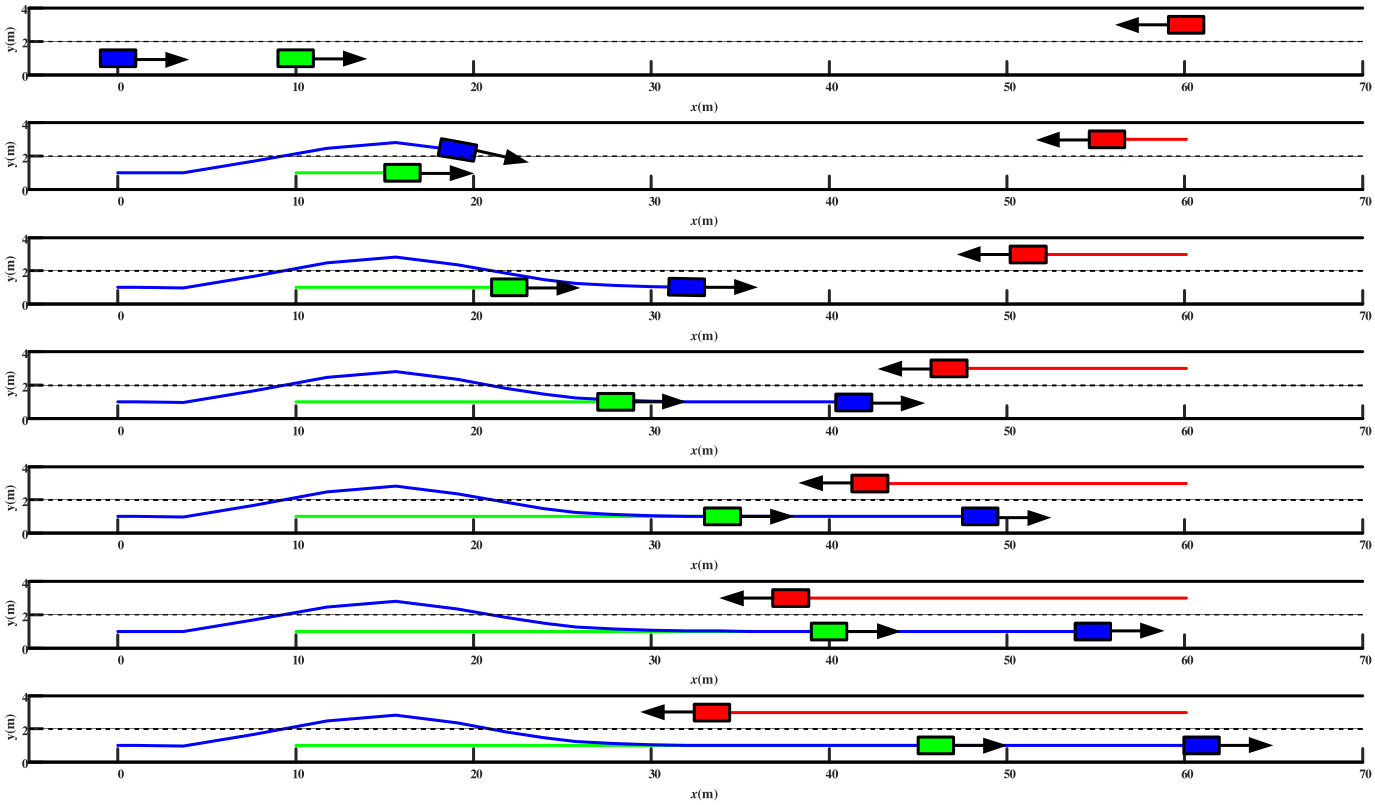

Fig. 13. Simulation result for Case 9. Pictured are time steps $t = 0.0$, 1.2, 3.6, 4.8, 6.0 and 7.2 s.

points, update model and remove the points with the lowest covariances.

Fig. 11 shows the residual model error during the closed-loop simulation applying the GP-based controller, as well as the predicted 2-σ residual error bound of the GP which overall matching the resulting residual errors well.

The experiment is illustrated in Fig. 12, consisting of three vehicles and two lanes of the width 2 m going in opposite directions. We control the blue vehicle in the lower lane and moving to the right. There are two vehicle obstacles, i.e., the green one is 10 m in front of the blue vehicle in the lower lane and moving to the right with constant velocities. The red one is 60 m in front of the blue vehicle in the upper lane and moving left with constant velocities. The obstacles do not change their vertical positions. Our task is to control the blue vehicle to move forward faster, and overtake the green vehicle while not colliding with the black vehicle or the two lanes boundaries.

In the absence of prior knowledge of adversary obstacles' dynamics, we assume a Gaussian distribution of each state $\tilde{x}_k^j$ but with uncertain moments. The prediction steps of the future positions of an obstacle are similar to the one in Section IV.B. In addition, we design the desired trajectory in advance for the blue vehicle. We verify the effectiveness and feasibility of the proposed algorithm in the following four cases.

Case 9: The blue vehicle and obstacles are placed at $[X_c^b, Y_c^b] = [0,\ 1]$, $[X_o^g, Y_o^g] = [10,\ 1]$ and $[X_o^r, Y_o^r] = [60,\ 3]$, respectively. The red obstacle has a width of 1 meter and length of 2 meters with uncertain location covariance $\Sigma_o^r = \mathrm{diag}(4, 0, 0)$, and its input has an indeterministic moment with mean (-5,0,0) and covariance $\Sigma_o^r = \mathrm{diag}(1, 0, 0)$. The green obstacle has a width of 1 meter and length of 2 meters with uncertain location covariance $\Sigma_o^g = \mathrm{diag}(1, 0.01, 0)$, and its input has an indeterministic moment with mean [5,0,0] and covariance $\Sigma_o^r = \mathrm{diag}(1, 0, 0)$. The joint chance constraint is imposed with $\alpha = 0.01$ and $\varepsilon = 0.001$ over the whole horizon length. The number of prediction steps are $N = 10$.

Case 10: As same as Case 9.

The vehicle has external process noise with zero mean and covariance $\Sigma_w = 1$ as simulation conditions to verify the proposed algorithm's effectiveness and feasibility under the following

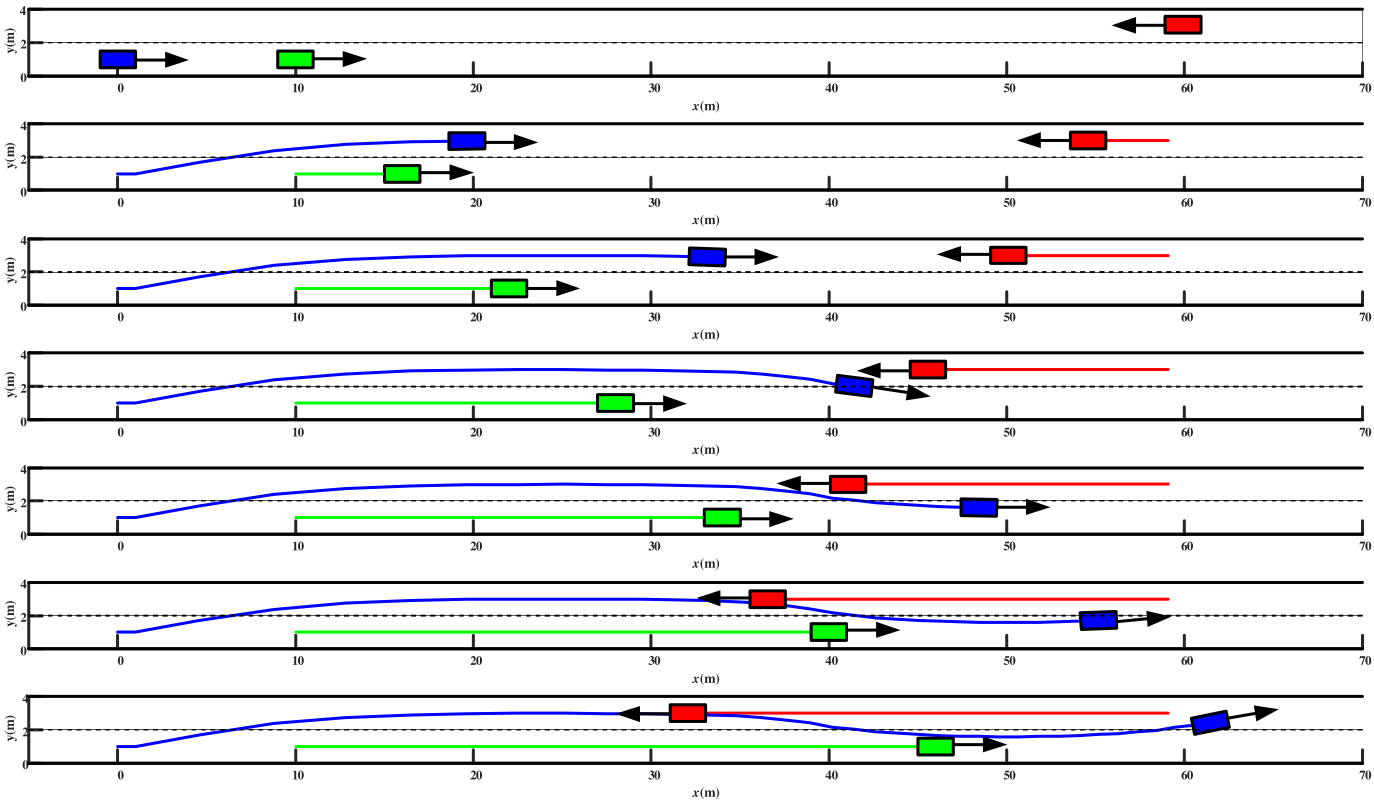

Fig. 14. Simulation result for Case 10. Pictured are time steps $t = 0.0$, 1.2, 3.6, 4.8, 6.0 and 7.2 s.

conditions, i.e., blue vehicle overtaking the green obstacle and moving in lower lane for Cases 9, and overtaking the green obstacle and moving in upper lane for Cases 10.

As shown in Fig. 13, the blue vehicle can straightly overtake the green obstacle, because of the sufficient safety distance between the blue and red obstacles. From Fig. 14, the blue vehicle can straightly overtake the green obstacle because of the sufficient safety distance between the blue vehicle and the red obstacle. Then, the blue vehicle moves to the upper lane. For Cases 9-10, we can achieve autonomous obstacle avoidance without online trajectory planning and keep vehicles moving inside a lane with model error. It can be seen from the experimental results that the method proposed in this paper has highly desired autonomy and feasibility for the vehicle operations dealing with the model error and uncertain environments.

VI. Conclusion

This work proposes a safety-guarantee MPC algorithm for autonomous system control, dealing with uncertain models and uncertain environments. The proposed method can effectively and efficiently solve autonomous system control problems when operating with moving obstacles. Moreover, it incorporates the existence of the environmental uncertainty, nonlinear dynamics and external noise into the scope of control. The method can effectively realize the vehicle 's autonomous collision avoidance for different shapes of obstacles. Compared with existing methods, this method can reduce substantially the conservativeness, computational time and computational cost while guaranteeing the safety of vehicles.

Our future work aims to tune the distribution model of uncertainty to specific problems and exploring ways to reduce the conservativeness introduced by Boole's inequality. We plan to perform some simulation with more complex and highly nonlinear model [46], [47], [48] to demonstrate the effectiveness of the proposed method. The mean square asymptotic stability [49], [50], [51] of the original system is an important but highly challenging issue. It deserves our further study. In addition, new practical applications should be targeted, including mobile robotic platforms [52], [53] and other applications [54], [55].

Acknowledgment

The authors are grateful for the efforts from our colleagues in the Sino-German Center of Intelligent Systems, Tongji University.

References

[1] P. M. Kebria, A. Khosravi, S. M. Salaken, and S. Nahavandi, "Deep imitation learning for autonomous vehicles based on convolutional neural networks," *IEEE/CAA J. Automatica Sinica*, vol. 7, no. 1, pp. 82–95, Jan. 2020.

[2] T. Bai, S. Li, and Y. Zou, "Distributed MPC for reconfigurable architecture systems via alternating direction method of multipliers," *IEEE/CAA J. Automatica Sinica*, vol. 8, no. 7, pp. 1336–1344, Jul. 2021.

[3] G.-P. Liu, "Coordination of networked nonlinear multi-agents using a high-order fully actuated predictive control strategy," *IEEE/CAA J. Automatica Sinica*, vol. 9, no. 4, pp. 615–623, Apr. 2022.

[4] S. Ju, J. Wang, and L. Dou, "MPC-based cooperative enclosing for nonholonomic mobile agents under input constraint and unknown disturbance," *IEEE Trans. Cybern.*, vol. 53, no. 2, pp. 845–858, Feb. 2023.

[5] L. Hewing, J. Kabzan, and M. N. Zeilinger, "Cautious model predictive control using Gaussian process regression," *IEEE Trans. Control Syst. Technol.*, vol. 28, no. 6, pp. 2736–2743, Nov. 2020,.

[6] H. Bao, J. An, M. Zhou, and Q. Kang, "A performance-driven MPC algorithm for underactuated bridge cranes," *Machines*, vol. 9, no. 8, pp. 177–194, 2021.

[7] J. Köhler, M. A. Müller, and F. Allgöwer, "Nonlinear reference tracking: An economic model predictive control perspective," *IEEE Trans. Autom. Control*, vol. 64, no. 1, pp. 254–269, Jan. 2019.

[8] J. Kocijan, R. Murray-Smith, C. E. Rasmussen, and A. Girard, "Gaussian process model based predictive control," in *Proc. 2004 Amer. Control Conf.*, 2004, vol. 3, pp. 2214–2219.

[9] X. Shi, Q. Kang, M. Zhou, J. An, and A. Abusorrah, "Novel L1 regularized extreme learning machine for soft-sensing of an industrial process," *IEEE Trans. Ind. Informat.*, vol. 18, no. 2, pp. 1009–1017, Feb. 2022.

[10] X. Shi, Q. Kang, M. Zhou, A. Abusorrah, and J. An, "Soft sensing of nonlinear and multimode processes based on semi-supervised weighted Gaussian regression," *IEEE Sensors J.*, vol. 20, no. 21, pp. 12950–12960, Nov. 2020.

[11] C. J. Ostafew, A. P. Schoellig, and T. D. Barfoot, "Robust constrained learning-based NMPC enabling reliable mobile robot path tracking," *Int. J. Robot. Res.*, vol. 35, no. 13, pp. 1547–1563, May 2016.

[12] C. D. McKinnon and A. P. Schoellig, "Experience-based model selection to enable long-term, safe control for repetitive tasks under changing conditions," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2018, pp. 2977–2984.

[13] M. Lorenzen, F. Dabbene, R. Tempo, and F. Allgöwer, "Constraint-Tightening and Stability in Stochastic Model Predictive Control," *IEEE Trans. Autom. Control*, vol. 62, no. 7, pp. 3165–3177, Jul. 2017.

[14] Y. Li, X.-Y. Lu, J. Wang, and K. Li, "Pedestrian trajectory prediction combining probabilistic reasoning and sequence learning," *IEEE Trans. Intell. Veh.*, vol. 5, no. 3, pp. 461–474, Sep. 2020.

[15] Z. Zuo et al., "MPC-based cooperative control strategy of path planning and trajectory tracking for intelligent vehicles," *IEEE Trans. Intell. Veh.*, vol. 6, no. 3, pp. 513–522, Sep. 2021.

[16] P. F. Lima, M. Nilsson, M. Trincavelli, J. Mårtensson, and B. Wahlberg, "Spatial model predictive control for smooth and accurate steering of an autonomous truck," *IEEE Trans. Intell. Veh.*, vol. 2, no. 4, pp. 238–250, Dec. 2017.

[17] Y. Chen, S. Chen, H. Ren, Z. Gao, and Z. Liu, "Path tracking and handling stability control strategy with collision avoidance for the autonomous vehicle under extreme conditions," *IEEE Trans. Veh. Technol.*, vol. 69, no. 12, pp. 14602–14617, Dec. 2020.

[18] I.-L. G. Borlaug, K. Y. Pettersen, and J. T. Gravdahl, "Tracking control of an articulated intervention autonomous underwater vehicle in 6DOF using generalized super-twisting: Theory and experiments," *IEEE Trans. Control Syst. Technol.*, vol. 29, no. 1, pp. 353–369, Jan. 2021.

[19] M. Castillo-Lopez, P. Ludivig, S. A. Sajadi-Alamdari, J. L. Sanchez-Lopez, M. A. Olivares-Mendez, and H. Voos, "A real-time approach for chance-constrained motion planning with dynamic obstacles," *IEEE Robot. Automat. Lett.*, vol. 5, no. 2, pp. 3620–3625, Apr. 2020.

[20] T. Weiskircher, Q. Wang, and B. Ayalew, "Predictive guidance and control framework for (semi-)autonomous vehicles in public traffic," *IEEE Trans. Control Syst. Technol.*, vol. 25, no. 6, pp. 2034–2046, Nov. 2017.

[21] A. Liniger and J. Lygeros, "Real-time control for autonomous racing based on viability theory," *IEEE Trans. Control Syst. Technol.*, vol. 27, no. 2, pp. 464–478, Mar. 2019.

[22] J. Lim, S. Pyo, N. Kim, J. Lee, and J. Lee, "Obstacle magnification for 2-D collision and occlusion avoidance of autonomous multirotor aerial vehicles," *IEEE/ASME Trans. Mechatron.*, vol. 25, no. 5, pp. 2428–2436, Oct. 2020.

[23] W. Cai, Y. Wu, and M. Zhang, "Three-dimensional obstacle avoidance for autonomous underwater robot," *IEEE Sensors Lett.*, vol. 4, no. 11, Nov. 2020, Art. no. 7004004.

[24] B. Lindqvist, S. S. Mansouri, A. Agha-mohammadi, and G. Nikolakopoulos, "Nonlinear MPC for collision avoidance and control of UAVs with dynamic obstacles," *IEEE Robot. Automat. Lett.*, vol. 5, no. 4, pp. 6001–6008, Oct. 2020.

[25] A. Britzelmeier and M. Gerdts, "A nonsmooth newton method for linear model-predictive control in tracking tasks for a mobile robot with obstacle avoidance," *IEEE Control Syst. Lett.*, vol. 4, no. 4, pp. 886–891, Oct. 2020.

[26] Y. Huang, H. Wang, A. Khajepour, H. Ding, K. Yuan, and Y. Qin, "A Novel local motion planning framework for autonomous vehicles based on resistance network and model predictive control," *IEEE Trans. Veh. Technol.*, vol. 69, no. 1, pp. 55–66, Jan. 2020.

[27] J. Lim, S. Pyo, N. Kim, J. Lee, and J. Lee, "Obstacle magnification for 2-D collision and occlusion avoidance of autonomous multirotor aerial vehicles," *IEEE/ASME Trans. Mechatron.*, vol. 25, no. 5, pp. 2428–2436, Oct. 2020.

[28] X. Wang, Q. Kang, M. Z., S. Yao, and A. Abusorrah, "Domain adaptation multitask optimization," *IEEE Trans. Cybern.*, early access, Nov. 2022, doi: 10.1109/TCYB.2022.3222101.

[29] Q. Deng, Q. Kang, L. Zhang, M. Zhou, and J. An, "Objective space-based population generation to accelerate evolutionary algorithms for large-scale many-objective optimization," *IEEE Trans. Evol. Comput.*, early access, Apr. 2022, doi: 10.1109/TEVC.2022.3166815.

[30] C. Premachandra, M. Otsuka, R. Gohara, T. Ninomiya, and K. Kato, "A study on development of a hybrid aerial /terrestrial robot system for avoiding ground obstacles by flight," *IEEE/CAA J. Automatica Sinica*, vol. 6, no. 1, pp. 327–336, Jan. 2019.

[31] V. Lefkopoulos and M. Kamgarpour, "Trajectory planning under environmental uncertainty with finite-sample safety guarantees," *Automatica*, vol. 131, 2021, Art no. 109754.

[32] V. Lefkopoulos and M. Kamgarpour, "Using uncertainty data in chance constrained trajectory planning," in *Proc. IEEE 18th Eur. Control Conf.*, 2019, pp. 2264–2269.

[33] M. Kamel, J. Alonso-Mora, R. Siegwart, and J. Nieto, "Robust collision avoidance for multiple micro aerial vehicles using nonlinear model predictive control," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2017, pp. 236–243.

[34] H. Zhu and J. Alonso-Mora, "Chance-constrained collision avoidance for MAVs in dynamic environments," *IEEE Robot. Automat. Lett.*, vol. 4, no. 2, pp. 776–783, Apr. 2019.

[35] M. Castillo-Lopez, S. A. Sajadi-Alamdari, J. L. Sanchez-Lopez, M. A. Olivares-Mendez, and H. Voos, "Model predictive control for aerial collision avoidance in dynamic environments," in *Proc. IEEE 26th Mediterranean Conf. Control Automat.*, 2018, pp. 1–6.

[36] A. Wang, A. Jasour, and B. C. Williams, "Non-Gaussian chance-constrained trajectory planning for autonomous vehicles under agent uncertainty," *IEEE Robot. Automat. Lett.*, vol. 5, no. 4, pp. 6041–6048, Oct. 2020.

[37] A. Mesbah, "Stochastic model predictive control: An overview and perspectives for future research," *IEEE Control Syst. Mag.*, vol. 36, no. 6, pp. 30–44, Dec. 2016.

[38] L. Blackmore, M. Ono, A. Bektassov, and B. C. Williams, "A probabilistic particle-control approximation of chance-constrained stochastic predictive control," *IEEE Trans. Robot.*, vol. 26, no. 3, pp. 502–517, Jun. 2010.

[39] H. Fang, N. Tian, Y. Wang, M. Zhou, and M. A. Haile, "Nonlinear bayesian estimation: From Kalman filtering to a broader horizon," *IEEE/CAA J. Automatica Sinica*, vol. 5, no. 2, pp. 401–417, Mar. 2018.

[40] Q. Kang, S. Yao, M. Zhou, K. Zhang, and A. Abusorrah, "Effective visual domain adaptation via generative adversarial distribution matching," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 32, no. 9, pp. 3919–3929, Sep. 2021.

[41] S. Yao, Q. Kang, M. Zhou, M. Rawa, and A. Albeshri, "Discriminative manifold distribution alignment for domain adaptation," *IEEE Trans. Syst., Man, Cybern.: Syst.*, vol. 53, no. 2, pp. 1183–1197, Feb. 2023.

[42] S. Yao, Q. Kang, M. Zhou, M. Rawa, and A. Abusorrah, "A survey of transfer learning for machinery diagnostics and prognostics," *Artif. Intell. Rev.*, early access, Aug. 2022 ,doi: 10.1007/s10462-022-10230-4.

[43] X. Wang, Q. Kang, M. Zhou, L. Pan, and A. Abusorrah, "Multiscale drift detection test to enable fast learning in non-stationary environments," *IEEE Trans. Cybern.*, vol. 51, no. 7, pp. 3483–3495, Jul. 2021.

[44] A. K. Singh, "Exponentially fitted cubature kalman filter with application to oscillatory dynamical systems," *IEEE Trans. Circuits Syst. I: Regular Papers*, vol. 67, no. 8, pp. 2739–2752, Aug. 2020.

[45] W. Ye, J. Cheng, L. Chen, Y. Liu, B. Wang, and R. Hu, "Iterative noise estimation-based cubature Kalman filtering for distributed pos in aerial earth observation imaging," *IEEE Sensors J.*, vol. 21, no. 24, pp. 27718–27727, Dec. 2021.

[46] S. Bi, M. Deng, and Y. Xiao, "Robust stability and tracking for operator-based nonlinear uncertain systems," *IEEE Trans. Automat. Sci. Eng.*, vol. 12, no. 3, pp. 1059–1066, Jul. 2015.

[47] M. Deng and T. Kawashima, "Adaptive nonlinear sensorless control for an uncertain miniature pneumatic curling rubber actuator using passivity and robust right coprime factorization," *IEEE Trans. Control Syst. Technol.*, vol. 24, no. 1, pp. 318–324, Jan. 2016.

[48] M. Zhou, Z. Cao, M. C. Zhou, J. Wang, and Z. Wang, "Zonotoptic fault estimation for discrete-time LPV systems with bounded parametric uncertainty," *IEEE Trans. Intell. Transp. Syst.*, vol. 21, no. 2, pp. 690–700, Feb. 2020.

[49] D. J. Higham, "Mean-square and asymptotic stability of numerical methods for stochastic ordinary differential equations," *SIAM J. Numer. Anal.*, vol. 38, no. 3, pp. 753–769, 2000.
[50] C. Huang, "Exponential mean square stability of numerical methods for systems of stochastic differential equations," *J. Comput. Appl. Math.*, vol. 236, no. 16, pp. 4016–4026, 2012.
[51] X. Jiang, S. Tian, T. Zhang, and W. Zhang, "Stability and stabilization of nonlinear discrete-time stochastic systems," *Int. J. Robust Nonlinear Control*, vol. 29, no. 18, pp. 6419–6437, 2019.
[52] H. Xia, M. A. Khan, Z. Li, and M. Zhou, "Wearable robots for human underwater movement ability enhancement: A survey," *IEEE/CAA J. Autom. Sinica*, vol. 9, no. 6, pp. 967–977, Jun. 2022.
[53] L. Jin, S. Liang, X. Luo, and M. Zhou, "Distributed and time-delayed -winner-take-all network for competitive coordination of multiple robots," *IEEE Trans. Cybern.*, vol. 53, no. 1, pp. 641–652, Jan. 2023.
[54] Z. Cao, Q. Xiao, R. Huang, and M. Zhou, "Robust neuro-optimal control of underactuated snake robots with experience replay," *IEEE Trans. Cybern. Neural Netw. Learn. Syst.*, vol. 29, no. 1, pp. 208–217, Jan. 2018.
[55] L. Huang, M. Zhou, and K. HAO, "Non-dominated immune-endocrine short feedback algorithm for multi-robot maritime patrolling," *IEEE Trans. Intell. Transp. Syst.*, vol. 21, no. 1, pp. 362–373, Jan. 2020.

HanQiu Bao received the B.S. degree in electrical engineering and its automation from the Nanjing University of Aeronautics and Astronautics, Nanjing, China, in 2016, and the M.S. degree in control theory and control engineering from the Shanghai Institute of Technology, Shanghai, China, in 2019. He is currently working toward the Ph.D. degree in control theory and control engineering with Tongji University, Shanghai, China. His research interests include MPC, underactuated system control, and nonlinear systems modeling.

Qi Kang (Senior Member, IEEE) received the B.S. degree in automatic control and the M.S. and Ph.D. degrees in control theory and control engineering from Tongji University, Shanghai, China, in 2002, 2005, and 2009, respectively.

From 2007 to 2008, he was a Research Associate with the University of Illinois at Chicago, Chicago, IL, USA. From 2014 to 2015, he was a Visiting Scholar with the New Jersey Institute of Technology, Newark, NJ, USA. He is currently a Professor with the Department of Control Science and Engineering and the Shanghai Institute of Intelligent Science and Technology, Tongji University. His research interests include swarm intelligence, evolutionary computation, machine learning, and intelligent control and optimization.

Dr. Kang was the General Chair of the 19th IEEE International Conference on Networking, Sensing and Control (ICNSC 2022). He is also an Associate Editor for IEEE TRANSACTIONS ON INTELLIGENT TRANSPORTATION SYSTEMS. He is the Secretary General with Shanghai Association for Systems Simulation.

XuDong Shi (Graduate Student Member, IEEE) received the M. S. degree in control theory and control engineering from Jiangnan University, Wuxi, China, in 2019. He is currently working toward the Ph.D. degree in control theory and control engineering with Tongji University, Shanghai, China. His research interests include machine learning, industrial process monitoring, and soft sensor development.

MengChu Zhou (Fellow, IEEE) received the B.S. degree from Nanjing University of Science and Technology, Nanjing, China in 1983, the M.S. degree from Beijing Institute of Technology, Beijing, China in 1986, and the Ph.D. degree from Rensselaer Polytechnic Institute, Troy, NY, USA, in 1990, He then joined the New Jersey Institute of Technology, Newark, NJ, USA, where he is currently a Distinguished Professor. His research interests are in Petri nets, automation, Internet of Things, cloud/edge computing, and AI. He has more than 1000 publications including 14 books, more than 700 journal papers (more than 600 in IEEE transactions), 31 patents and 32 book-chapters. He is a recipient of Excellence in Research Prize and Medal from NJIT, Humboldt Research Award for US Senior Scientists from Alexander von Humboldt Foundation, and Franklin V. Taylor Memorial Award and the Norbert Wiener Award from IEEE Systems, Man, and Cybernetics Society, and Edison Patent Award from the Research & Development Council of New Jersey. He is a life member of Chinese Association for Science and Technology-USA and served as its President in 1999. He is Fellow of IEEE, International Federation of Automatic Control (IFAC), American Association for the Advancement of Science (AAAS), Chinese Association of Automation (CAA) and National Academy of Inventors (NAI).

HaoJun Li received the Ph.D. degree in geodesy with Tongji University, Shanghai, China. He is currently a Professor of geodesy and surveying engineering with Tongji University. His research interests include GNSS precise point positioning, GNSS satellite clock, multi-GNSS, and multi-frequency biases.

Jing An (Member, IEEE) received the Ph.D. degree in control theory and control engineering from Tongji University, Shanghai, China, in 2013. She is currently an Associate Professor with the School of electrical and electronic engineering, Shanghai Institute of Technology, Shanghai, China. Her research interests include computational intelligence, multi-objective optimization, and intelligent information processing.

Khaled Sedraoui received the B.S. and Ph.D. degrees in electrical engineering from the Institute of Technology and Sciences, Tunis, Tunisia. Since 1997, he has been a Faculty Member with the College of Technology, Jeddah, Saudi Arabia. He is currently an Associate Professor with King Abdulaziz University, Jeddah, Saudi Arabia. His research interests include renewable energy, design and optimization of complex systems, smart grid and security.