Motion Planning for Mobile Robots with Noise: A Probabilistic MPC Approach

Jiawei Tang, Yik Ben Wong, Zhengyu Fu, Nachuan Yang, Sil Kwong Tse,
Winnie Suk Wai Leung and Ling Shi, *Senior Member, IEEE*

***Abstract*—This paper fills the literature gap by considering the safety-preserving motion planning problem of mobile robots with process noise. Most existing safe motion planning algorithms assume ideal system dynamics of mobile robots, which is not realistic in real-world situations. In this paper, we model the mobile robot as a nonlinear system with Gaussian noise. To satisfy the safety constraints on the mobile robot's trajectory, we propose a novel probabilistic model predictive control (MPC) approach where the MPC frequency is determined according to the safety probability of the mobile robots. We further develop an online algorithm for numerical computation which can guarantee the mobile robot's safety with a prescribed probability threshold. Due to the probabilistic framework, the computational burden for the proposed algorithm is much lower than the existing MPC-based safe motion planning algorithms. Finally, we provide several numerical simulations as well as physical experiments to verify the effectiveness of our proposed approach.**

***Index Terms*—Motion planning, mobile robots, safety.**

I. Introduction

Safe and efficient motion planning has been an essential and challenging task for mobile robots, autonomous vehicles, and unmanned aerial vehicles (UAV) [1]–[3]. A motion planner needs to generate a smooth trajectory for the mobile robot to execute based on both system's dynamics and environment information. Important applications can be found in autonomous driving [4], formation control [5], and cooperative control [6], to name just a few. Even though motion planning has been studied for decades, it is still a challenging task for practical applications [7]. The difficulties of implementing motion planning algorithms in practical tasks mainly exist in the following two parts. First, the system model for mobile robots is inaccurate and there exist process and measurement noise for real-world dynamics. Second, the motion planning has to be designed by incorporating the physical constraints, such as the actuator limits, obstacle avoidance and so on. This paper studies a further step to tackle the above practical difficulties where we consider the safety-preserving motion planning of mobile robots with process noise.

Jiawei Tang, Yik Ben Wong, Nachuan Yang, Sil Kwong Tse and Ling Shi are with the Department of Electronic and Computer Engineering, Hong Kong University of Science and Technology, Clear Water Bay, Hong Kong SAR (email: {jtangas, ybwong, nc.yang, sktse, eesling}@connect.ust.hk).

Zhengyu Fu is with the Department of Mechanical and Process Engineering, ETH Zurich, Zurich 8092, Switzerland (email: zhenfu@student.ethz.ch).

Winnie Suk Wai Leung is with the Division of Integrative Systems and Design, Hong Kong University of Science and Technology, Clear Water Bay, Hong Kong SAR (email: eewswleung@ust.hk).

The work is supported by a Hong Kong RGC General Research Fund 16210619.

In recent years, optimization-based methods have received much attention in mobile robot motion planning research. An optimization-based method formulates the motion planning problem as a mathematical optimization problem, which allows continuous planning and is compatible with various constraints. One typical category is based on the sequential quadratic programming (SQP) [8]. The work by Ziegler et al. [9] demonstrated its capability to solve the nonlinear and nonconvex motion planning problem. However, as solving a sequence of quadratic programming within each iteration is required, SQP is not suitable for real-time implementation on many hardware platforms. Differential dynamic programming (DDP) [10] and iterative linear quadratic regulator (ILQR) [11] enable us to handle the constraint-free motion planning more effectively. Many recent works have demonstrated their efficiency successfully in complex hardware platforms, including hexacopter [12] and legged robot [13].

More recently, studies on constrained optimization-based motion planning appear. The control-limited DDP [14] adds constraints to Bellman equation in value iteration, but it can only handle control input constraints. Extended LQR [15] penalizes the distance measured from the center of an obstacle to handle collision avoidance with circular objects. Chen et al. [16], [17] developed the constrained iterative linear quadratic regulator (CILQR) algorithm to handle the inequality constraints through constraint relaxation. Unfortunately, numerical stability cannot be guaranteed when incorporated with the barrier function. Aoyama et al. [18] proposed constrained DDP with penalty methods and active-set approaches. In the most recent work by Pan et al. [19], the sudden target vehicle maneuvers uncertainty on the constrained motion planning problem was addressed via adaptive CILQR through reachability analysis. However, all these motion planning algorithms assume ideal system dynamics of mobile robots, which is not realistic in real-world situations. This motivates our research.

This paper fills the literature gap by considering the safe motion planning of mobile robots with process noise. The contributions of this paper are listed as follows:

1) We propose a probabilistic framework to analyze the safety-preserving motion planning, where the safety is guaranteed subject to a probability threshold.

2) We propose a novel probabilistic model predictive control (MPC) approach where the MPC frequency is determined by the safety probability during the movement process.

3) We develop an online algorithm for numerical computation which can guarantee the mobile robot's safety with

a prescribed probability threshold. The effectiveness of our proposed algorithm and theoretical results are well verified by numerical simulations and physical experiments.

The remainder of this paper is organized in the following order. In Section II, some preliminary knowledge on ILQR and MPC is introduced, and the problem to be solved by this paper is formulated. In Section III, we propose a novel probabilistic MPC approach and the corresponding motion planning algorithm is also developed. In Section IV, we analyze the motion planning of a two-wheeled robot using our proposed framework. In Section V, the experimental results in both numerical simulations and a physical testbed are provided. In Section VI, the paper is summarized and concluded.

Notations: The notations used throughout this paper are standard. $\|\cdot\|$ denotes the Euclidean norm for vectors. The probability of event X is denoted by $\mathbb{P}(X)$. The expectation of random variable X is denoted by $\mathbb{E}[X]$. The notation $d(x, y)$ means the Euclidean distance between vector x and vector y. $A \succ 0$ (or $A \succeq 0$) means A is positive definite (or semidefinite). $A > 0$ (or $A \geq 0$) means all elements of A are positive (or nonnegative). The cumulative distribution function of a zero-mean unit variance Gaussian random variable is denoted by

$$\Phi(x) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{x} e^{-t^2/2} dt.$$

The matrices and vectors are assumed to have compatible dimensions if not explicitly mentioned.

II. Preliminary

A. Iterative Linear Quadratic Regulator

ILQR is an efficient numerical tool to solve a nonlinear optimal control problem.

Consider a discrete-time nonlinear system:

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

where $x_k \in \mathbb{R}^n$ and $u_k \in \mathbb{R}^m$ are the state and control input at time step k, $f(x, u)$ is the state transition function.

Assumption 1: $f(x, u)$ satisfies the Lipschitz condition with a Lipschitz constant $\gamma > 0$, i.e.,

$$\forall x_1, x_2 \in \mathbb{R}^n,\ u \in \mathbb{R}^m, \|f(x_1, u) - f(x_2, u)\| \leq \gamma \|x_1 - x_2\|.$$

The finite-horizon LQR performance index of the control system is defined as:

$$J = x_N^{\mathrm{T}} Q_f x_N + \sum_{k=0}^{N-1} x_k^{\mathrm{T}} Q x_k + u_k^{\mathrm{T}} R u_k \tag{2}$$

where $Q_f \succeq 0$, $Q \succeq 0$ and $R \succ 0$ are the final state cost matrix, intermediate state cost matrix and input cost matrix, respectively. ILQR aims to design the control input and state trajectory to minimize (2) subject to the nonlinear system dynamics (1). The steps for solving this nonlinear optimal control problem with ILQR are summarized as follows:

(1) Generate a feasible nominal control input and state trajectory $\bar{\tau} := \{\bar{x}_0, \bar{u}_0, \bar{x}_1, \bar{u}_1, \ldots, \bar{x}_{N-1}, \bar{u}_{N-1}, \bar{x}_N\}$.

(2) Linearize the system dynamics along the nominal trajectory with $\delta x_k = x_k - \bar{x}_k$ and $\delta u_k = u_k - \bar{u}_k$:

$$\begin{aligned} \delta x_{k+1} &= A_k \delta x_k + B_k \delta u_k \\ A_k &= \frac{\partial f(x, u)}{\partial x}|_{\bar{x}_k, \bar{u}_k} \\ B_k &= \frac{\partial f(x, u)}{\partial u}|_{\bar{x}_k, \bar{u}_k} \end{aligned} \tag{3}$$

(3) Backward optimize the control input through dynamic programming with the linearized system dynamics.

(4) Run forward pass with optimized control sequence. Adopt backtracking line search scheme to ensure the performance is improved. Update nominal trajectory $\bar{\tau}$.

(5) Iterative Step (2) to step (4) until J convergences. Return the updated trajectory $\bar{\tau}$.

B. Model Predictive Control

MPC relies on the system model to predict future trajectories and to optimize over them. Mathematically speaking, an MPC problem can be formulated as follows:

$$\begin{aligned} \min_{u} \quad & J = \sum_{k=0}^{\mathcal{P}-1} c(x_k, u_k) \\ \text{s.t.} \quad & x_{k+1} = f(x_k, u_k) \\ & g(x_k, u_k) \leq 0 \\ & h(x_k, u_k) = 0 \end{aligned}$$

where $c(\cdot)$, $g(\cdot)$ and $h(\cdot)$ denote the cost function, inequality constraint and equality constraint, respectively, while $\mathcal{P}$ denotes the predictive time horizon.

C. Problem Statement

We consider a nonlinear system with process noise

$$x_{k+1} = f(x_k, u_k) + w_k \tag{4}$$

where w_k is the i.i.d. white Gaussian noise at time step k with covariance matrix $\Sigma \succ 0$. The objective is to design the control input sequence such that the LQR performance index J is minimized. Meanwhile, the system should satisfy the safety constraints subject to a prescribed probability. Mathematically, the problem is formulated as follows:

Problem 1: (Constrained Nonlinear Optimal Control with Process Noise)

$$\min_{u} \quad J = \mathbb{E}[x_N^{\mathrm{T}} Q_f x_N + \sum_{k=0}^{N-1} x_k^{\mathrm{T}} Q x_k + u_k^{\mathrm{T}} R u_k] \tag{5}$$

$$\text{s.t.} \quad x_k = f(x_{k-1}, u_{k-1}) + w_{k-1} \tag{6}$$

$$\Lambda u_k \in \mathcal{U}_k, \quad k = 0, \ldots, N-1 \tag{7}$$

$$\mathbb{P}(\Omega x_k \in \mathcal{X}_k) \geq \omega, \quad k = 1, \ldots, N \tag{8}$$

where Λ and Ω are matrix parameters, and $\mathcal{U}_k$ and $\mathcal{X}_k$ are the safe sets associated with control input u_k and state x_k, respectively. The corresponding unsafe sets of $\mathcal{U}_k$ and $\mathcal{X}_k$ are $\mathcal{U}_k^c$ and $\mathcal{X}_k^c$. It is required that $\Omega x_k \in \mathcal{X}_k$ should hold with a probability exceeding ω. The main difficulty of the above problem is to satisfy the safety constraints as well as to achieve the optimality of the controller simultaneously. Meanwhile the control strategy needs to be computed efficiently to meet real-time requirements of a physical platform.

III. Main Result

In this section, we propose a probabilistic framework to analyze the trajectory of nonlinear systems with Gaussian noise. We first reformulate the problem in the sense of expectation and solve the nonlinear optimal control problem with ILQR. Then we analyze the bound for the mean squared error (MSE) between the real trajectory and the nominal trajectory. Finally, we propose a probabilistic MPC approach and the corresponding algorithm to solve *Problem 1*.

A. Expectation Analysis

Since w_k is i.i.d. white Gaussian, $\mathbb{E}[x_{k+1}]$ is equivalent to $\mathbb{E}[f(x_k, u_k)]$. Denote $\mathbb{E}[x_k]$ as $\hat{x}_k$. To solve *Problem 1*, we first ensure that $\hat{x}_k$ can satisfy all state constraints. Hence we consider a nonlinear optimal control problem subject to expected constraints, which is stated as follows.

Problem 2: (Nonlinear Optimal Control with Expectation Constraints)

$$\min_{x} \quad J = \hat{x}_N^{\mathrm{T}} Q_f \hat{x}_N + \sum_{k=0}^{N-1} \hat{x}_k^{\mathrm{T}} Q \hat{x}_k + u_k^{\mathrm{T}} R u_k \tag{9}$$

$$\text{s.t.} \quad \hat{x}_k = f(\hat{x}_{k-1}, u_{k-1}) \tag{10}$$

$$\Lambda u_k \in \mathcal{U}_k, \quad k = 0, \ldots, N-1 \tag{11}$$

$$\Omega \hat{x}_k \in \mathcal{X}_k, \quad k = 1, \ldots, N. \tag{12}$$

Note that *Problem 2* is a constrained nonlinear optimization problem with a nonconvex feasible region. To tackle this problem, we employ the ILQR solver to generate a feasible trajectory and optimize it. More specifically, we leverage the approach developed by [11] with a backtracking line-search scheme. By satisfying the Armijo condition [20] in the backtracking linear search iteration, the convergence of ILQR can be guaranteed [14]. As the constraints may be violated during the ILQR trajectory update, we continue to check the updated trajectory until all constraints are satisfied in the backtracking linear search iteration. Such technique is also used in [17], [19]. With a given feasible trajectory and a diminishing step size for line search along the descent direction, it can guarantee that the algorithm converges and the updated trajectory satisfies the constraints.

B. Variance Analysis

Due to the unbounded nature of Gaussian white noise, we are interested in the mean squared error (MSE) between the real trajectory and the planned trajectory. Define the difference between actual state x_k and its planned reference state $\bar{x}_k$ as e_k, i.e., $e_k = x_k - \bar{x}_k$, the MSE as $\mathbb{E}[e_k^{\mathrm{T}} e_k]$. Through analyzing the error dynamics, we have following result:

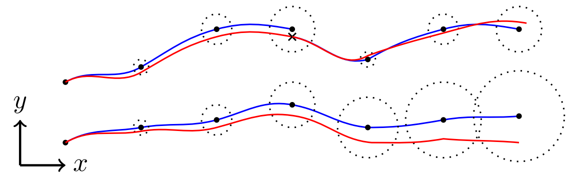

Fig. 1: The top lines represent the replanned trajectory with $\mathcal{T} = 3$, while the bottom represents the trajectory without replanning. The red line indicates the real trajectory, the blue line indicates the planned trajectory, and the area within the dotted circles is the possible position of the mobile robot due to the process noise. Notice that the MSE is increasing during the movement. Through intermittently replaning the motion, the MSE between the real trajectory and planned trajectory is bounded.

Theorem 1: For system (4) with a nominal trajectory satisfying (10) – (12), the following inequality holds

$$\mathbb{E}[e_k^{\mathrm{T}} e_k] \le \gamma^{2k} \mathbb{E}[e_0^{\mathrm{T}} e_0] + \sum_{i=0}^{k-1} \gamma^{2i} \mathrm{Tr}(\Sigma)$$

where γ is the Lipschitz constant of function $f(\cdot)$.

Proof: The error dynamics is given by

$$e(k+1) = f(x_k, u_k) - f(\bar{x}_k, u_k) + w_k$$

Denote $\Delta f_k = f(x_k, u_k) - f(\bar{x}_k, u_k)$, then the MSE is

$$\mathbb{E}[e_{k+1}^{\mathrm{T}} e_{k+1}] = \mathbb{E}[\Delta f_k^{\mathrm{T}} \Delta f_k] + \mathrm{Tr}(\Sigma)$$

Since function $f(\cdot)$ satisfies the Lipschitz condition in *Assumption 1*, we have the following inequality holds

$$\Delta f_k^{\mathrm{T}} \Delta f_k \le \gamma^2 e_k^{\mathrm{T}} e_k$$

which yields a recurrent relationship on $\mathbb{E}[e_k^{\mathrm{T}} e_k]$:

$$\mathbb{E}[e_{k+1}^{\mathrm{T}} e_{k+1}] \le \gamma^2 \mathbb{E}[e_k^{\mathrm{T}} e_k] + \mathrm{Tr}(\Sigma)$$

which further implies that

$$\mathbb{E}[e_k^{\mathrm{T}} e_k] \le \gamma^{2k} \mathbb{E}[e_0^{\mathrm{T}} e_0] + \sum_{i=0}^{k-1} \gamma^{2i} \mathrm{Tr}(\Sigma)$$

The proof is completed. □

Corollary 1: If $f(x, u)$ satisfies the constant turning rate and velocity magnitude (CTRV) motion model [7], the following inequality holds

$$\mathbb{E}[e_k^{\mathrm{T}} e_k] \le \mathbb{E}[e_0^{\mathrm{T}} e_0] + k \mathrm{Tr}(\Sigma)$$

where Σ denotes the process noise covariance.

Proof: Under the CTRV assumption, we have $\gamma = 1$ then the inequality follows immediately from *Theorem 1*. □

Notice that if $\gamma < 1$, the MSE is bounded by $\mathrm{Tr}(\Sigma)/(1-\gamma^2)$. If $\gamma \ge 1$, through replanning the trajectory during the operation, it can always ensure that the actual state would not deviate from the planned trajectory too much. The replanning cycle is denoted as $\mathcal{T} \in \mathbb{Z}$. A graphical explanation on how the replanning works is shown in Fig. 1.

C. Algorithmic Solution

Based on the expectation and variance analysis, *Problem 1* can be solved by a probabilistic MPC approach with intermittent replanning. The pseudo-code for the corresponding algorithm is shown in *Algorithm 1*. The core ingredients of the algorithm include the following three parts:

1) An **MPC-based outer loop** for replanning the trajectory to maintain safety during the operation.

2) An **ILQR-based inner loop** for producing a sequence of optimized control inputs and reference states.

3) A **line search scheme** to ensure the convergence of ILQR subject to the safety constraints.

Algorithm 1 Safety-preserving motion planning algorithm for mobile robots with process noise

Given
System dynamics: $x_{k+1} = f(x_k, u_k) + w_k$; initial state: x_0; target state: x_d Cost: $J = (x_N - x_d)^{\mathrm{T}} Q_f (x_N - x_d) + \sum_{k=0}^{N-1} (x_k - x_d)^{\mathrm{T}} Q (x_k - x_d) + u_k^{\mathrm{T}} R u_k$
Initialize $\mathcal{P} > 0$, $\xi > 0$, $\mathcal{T} > 0$, $\Delta \geq 0$, $i = 0$
`// Outer loop on MPC control`
while *not arrive at the target* **do**
  if $\mathrm{mod}(i, \mathcal{T}) == 0$ **then**
    $\bar{x}_0 \leftarrow x_i$
    Generate $\bar{\tau} := \{\bar{x}_0, \bar{u}_0, \bar{x}_1, \bar{u}_1, \ldots, \bar{x}_{N-1}, \bar{u}_{N-1}, \bar{x}_N\}$
    Calculate the cost of $\bar{\tau}$: $\bar{J}$
    `// Inner loop on ILQR control`
    while $|J - \bar{J}|/|\bar{J}| > \xi$ **do**
      `// System dynamics linearization`
      for $k = 0$ *to* $k = \mathcal{P} - 1$ **do**
        $A_k \leftarrow \frac{\partial f(x,u)}{\partial x}|_{\bar{x}_k, \bar{u}_k}$
        $B_k \leftarrow \frac{\partial f(x,u)}{\partial u}|_{\bar{x}_k, \bar{u}_k}$
      end
      `// Backward Pass`
      $P_{\mathcal{P}} \leftarrow Q_f$, $p_{\mathcal{P}} \leftarrow Q_f(\bar{x}_{\mathcal{P}} - x_d)$
      for $k = \mathcal{P} - 1$ *to* $k = 0$ **do**
        $H_k \leftarrow R + B_k^{\mathrm{T}} P_{k+1} B_k$
        $G_k \leftarrow B_k^{\mathrm{T}} P_{k+1} A_k$
        $g_k \leftarrow R u_k + B_k^{\mathrm{T}} p_{k+1}$
        $K_k \leftarrow -H^{-1} G_k$
        $l_k \leftarrow -H^{-1} g_k$
        $p_k \leftarrow Q(\bar{x}_k - x_d) + A_k^{\mathrm{T}} p_{k+1} + K_k^{\mathrm{T}} H_k l_k + K_k^T g_k + G_k^{\mathrm{T}} l_k$
        $P_k \leftarrow Q + A_k^{\mathrm{T}} P_{k+1} A_k + K_k^{\mathrm{T}} H_k K_k + K_k^{\mathrm{T}} G_k + G_k^{\mathrm{T}} K_k$
      end
      `// Line search and forward pass`
      $\alpha \leftarrow 1$, $\alpha_d > 1$
      while $J > \bar{J}$ *or constraints are violated* **do**
        Update the control: $u_k \leftarrow \bar{u}_k + \alpha l_k + K_k(x_k - \bar{x}_k)$
        Forward simulate the trajectory τ
        Calculate the cost of updated trajectory J
        $\alpha \leftarrow \alpha / \alpha_d$
      end
      $\bar{\tau} \leftarrow \tau$
    end
    $i \leftarrow 0$
  end
  Forward execute the control: $x_{i+1} = f(x_i, u_i) + w_i$
  $i \leftarrow i + 1$
end

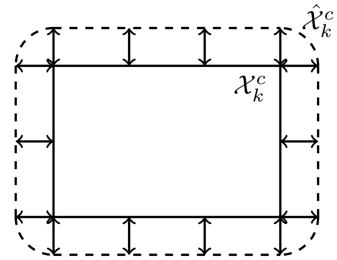

Fig. 2: An illustrated example on $\mathcal{X}_k^c$ and $\hat{\mathcal{X}}_k^c$. The region within the solid line indicates $\mathcal{X}_k^c$, and the region within the hash indicates $\hat{\mathcal{X}}_k^c$.

Due to the existence of the noise, we consider the soft unsafe set $\hat{\mathcal{X}}_k^c$ of $\mathcal{X}_k^c$, which is defined as follows:

$$\hat{\mathcal{X}}_k^c = \{x_k | d(\Omega x_k, \mathcal{X}_k^c) \leq \Delta\}$$

where $\Delta > 0$ denotes a safety margin that should be considered when doing motion planning, and $d(x, S)$ denotes the minimum Euclidean distance between x and $y \in S$, which can be calculated as follows

$$d(x, S) = \min_{y \in S} d(x, y), x \notin S$$

An example on $\mathcal{X}_k^c$ and $\hat{\mathcal{X}}_k^c$ is shown in Fig. 2.

Theorem 2: Through motion planning using *Algorithm 1* with a safety margin

$$\Delta \geq \Phi^{-1}((1 + \omega)/2)\sqrt{\mathcal{T}\mathrm{Tr}(\Sigma)},$$

it holds that $\mathbb{P}(\Omega x_k \in \mathcal{X}_k) \geq \omega$ for $k = 1, \ldots, N$.

Proof: Denoting the variance of e_k as σ, we have:

$$\mathbb{P}(\|e_k\| \leq \Delta) = 2\Phi(\frac{\Delta}{\sigma}) - 1.$$

Note that by considering $\hat{\mathcal{X}}_k^c$ in planning, if $\|e_k\| \leq \Delta$ then $d(\Omega x_k, \mathcal{X}_k^c) \geq 0$, which indicates $\Omega x_k \in \mathcal{X}_k$. Hence with

$$\Delta \geq \Phi^{-1}((1 + \omega)/2)\sigma,$$

$\mathbb{P}(\Omega x_k \in \mathcal{X}_k) \geq \omega$. Following *Corollary 1*, through planning with MPC cycle $\mathcal{T}$, $\mathbb{E}[e_k^{\mathrm{T}} e_k] \leq \mathcal{T}\mathrm{Tr}(\Sigma)$. Hence with

$$\Delta \geq \Phi^{-1}((1 + \omega)/2)\sqrt{\mathcal{T}\mathrm{Tr}(\Sigma)},$$

$\mathbb{P}(\Omega x_k \in \mathcal{X}_k) \geq \omega$. The proof is completed. □

IV. Motion Planning for Mobile Vehicle

In this section, we model a two-wheeled mobile robot with our proposed framework. We first present the system dynamics of the two-wheeled mobile robot, then we discuss the choice of the objective function and safety constraints.

A. System Dynamics

The mobile robot has a differential-wheeled locomotive which can be modeled as the kinematic unicycle model shown in Fig. 3. The state of a differential-wheel robot at time step k is $[x_k^p, y_k^p, \theta_k]^{\mathrm{T}}$, capturing its position and orientation in the global frame. The velocity vector is $[v_k, \omega_k]^{\mathrm{T}}$, containing the linear velocity and angular velocity of the mobile robot. The dynamics of the kinematic unicycle model is:

$$\dot{x}_k = \begin{bmatrix} \dot{x}_k^p \\ \dot{y}_k^p \\ \dot{\theta}_k \end{bmatrix} = \begin{bmatrix} v_k \cos\theta_k \\ v_k \sin\theta_k \\ \omega_k \end{bmatrix} = \begin{bmatrix} \cos\theta_k & 0 \\ \sin\theta_k & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} v_k \\ \omega_k \end{bmatrix}$$

The model in discrete-time form is given by:

$$x_{k+1} = \begin{bmatrix} x_{k+1}^p \\ y_{k+1}^p \\ \theta_{k+1} \end{bmatrix} = \begin{bmatrix} x_k^p \\ y_k^p \\ \theta_k \end{bmatrix} + \delta t \begin{bmatrix} \cos\theta_k & 0 \\ \sin\theta_k & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} v_k \\ \omega_k \end{bmatrix}$$

where δt is the sampling time.

The control input $u_k = [v_k^r, v_k^l]^{\mathrm{T}}$ contains the rotational speed of the left and right wheels. The transformation between u_k and $[v_k, w_k]^{\mathrm{T}}$ follows with the unicycle model kinematic constraint:

$$\begin{bmatrix} v_k \\ \omega_k \end{bmatrix} = \begin{bmatrix} R/2 & R/2 \\ R/L & -R/L \end{bmatrix} \begin{bmatrix} v_k^r \\ v_k^l \end{bmatrix}$$

where R and L are the radius of wheels and body length of robot, respectively.

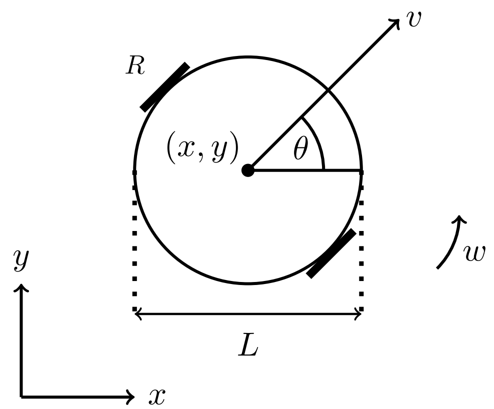

Fig. 3: Two-wheeled unicycle model.

B. Objective Function and Constraints

We consider the scenario that the mobile robot goes around obstacles and stops at the target position. The objective function is given by:

$$J = \mathbb{E}[(x_N - x_d)^{\mathrm{T}} Q_f (x_N - x_d) + \\ + \sum_{k=0}^{N-1} (x_k - x_d)^{\mathrm{T}} Q (x_k - x_d) + u_k^{\mathrm{T}} R u_k]$$

where x_d is the target position. To generate a feasible trajectory, we implemented a path planning algorithm using the potential field method [21]. As an alternative, the control framework is compatible with any controller that can generate a feasible trajectory for the mobile robot.

Two practical constraints on the state and control input which the mobile robot should not violate during the operation are considered, including the motor's rotational speed constraints and the obstacle avoidance constraints.

Motor's rotational speed constraints: The motor's rotational speed of the mobile robot should be bounded based on the motor force limit. Hence $\Lambda u_k \in \mathcal{U}_k$ can be formulated as an inequality constraint:

$$\underline{u} \leq u_k \leq \overline{u}$$

where $\underline{u}$ and $\overline{u}$ are the minimum and maximum control input.

Obstacle avoidance constraint: The mobile robot should always keep a certain distance from any obstacle. Hence $\Omega x_k \in \mathcal{X}_k$ can be formulated as:

$$d(\Omega x_k, \mathcal{X}_k^c) \geq 0 \quad \text{with} \quad \Omega = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix}$$

where $\mathcal{X}_k^c$ is the infeasible region in $\mathbb{R}^2$ at time step k.

V. Experimental Result

We conduct extensive numerical simulations and physical experiments to verify the proposed algorithm in terms of safety, effectiveness, and computation time. We compare the computation time and LQR performance of our algorithm and the CILQR algorithm in [17], which considers a similar setup for mobile robots. Then we conduct Monte Carlo simulations to evaluate the safety probability with our algorithm for different MPC replanning frequencies. Finally, to evaluate the real-time performance of our control framework, we conduct several motion planning experiments in a physical testbed.

A. Simulation Results without Uncertainty

The simulation experiments are done in a rectangle arena with several obstacles. The radius of circular obstacles is $0.5m$ while the length of square obstacles is $1m$. The starting position and the target position are randomly generated, and Δ is set to be zero. An example of the arena is shown in Fig. 4. $\underline{u}$ and $\overline{u}$ is set to $[-65, -65]^{\mathrm{T}}$ and $[65, 65]^{\mathrm{T}}$, respectively. The average time horizon is $N = 1000$. The simulator is written in Matlab and runs on a desktop with a standard CPU. To compare the computation time and LQR performance on our control framework and CILQR [17], we run 1000 Monte Carlo tests. The simulation results are shown in TABLE I:

TABLE I: Runtime and LQR Performance Comparison

Algorithm	Time (s)	J/J_{CILQR} (%)
CILQR in [17]	3.029	100%
Ours without replanning	0.256	107%
Ours with $\mathcal{T} = 50$	0.450	86%
Ours with $\mathcal{T} = 10$	2.645	69%
Ours with $\mathcal{T} = 5$	3.298	62%

Notice that without replanning by MPC, our algorithm is 10 times faster than CILQR with negligible performance loss. Through replanning the trajectory during the operation, our algorithm can achieve better LQR performance compared with CILQR while the computation time is maintained within an acceptable range for real-time implementation.

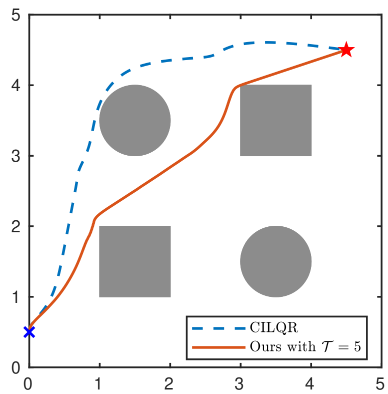

Fig. 4: Trajectories of the mobile robot running in a noiseless simulation environment. The grey areas indicate infeasible regions in the arena. The blue cross marker is the starting position, and the red star marker is the target position, which is randomly generated.

B. Simulation Results with Uncertainty

In this scenario, the simulation environment is the same as the one in Section V-A. We add zero-mean Gaussian noise to the mobile robot to evaluate the MSE and the probability of constraint satisfaction with 1000 Monte Carlo tests. The standard deviation of the noise on x_k^p and y_k^p is $1cm$, while the standard deviation of the noise on θ_k is set to $1°$. The probabilities of safety constraints satisfaction using CILQR and our algorithm are shown in TABLE II.

TABLE II: Constraint Satisfaction Probability Comparison

Algorithm	**Δ (m)**	**Safety Probability**
CILQR in [17]	0	40.2%
CILQR in [17]	0.05	58.9%
Ours with $\mathcal{T} = 15$	0.05	86.2%
Ours with $\mathcal{T} = 10$	0.05	91.3%
Ours with $\mathcal{T} = 5$	0.05	96.3%

It is easy to see that the safety probability with our algorithm is much higher than the one with CILQR.

C. Practical Results in a Physical Testbed

In this scenario, we demonstrate different control tasks in a physical testbed. In the testbed, an autonomous ground vehicle (AGV) is running in a rectangle arena with several obstacles. A server is connected to a Vicon system, which can continuously receive the poses of the AGV and the obstacles, and send commands to the AGV over the wireless network socket. Fig. 5 shows our testbed environment.

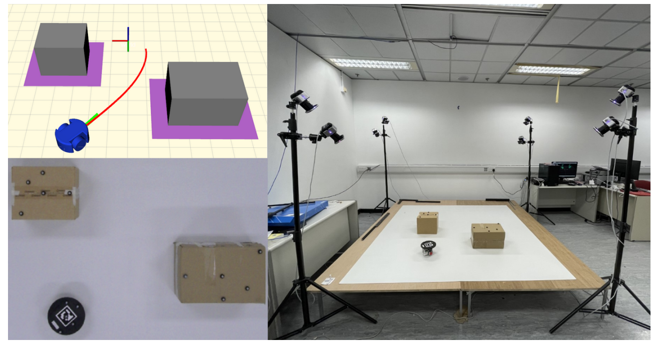

Fig. 5: Physical testbed environment.

The experiment shows that the proposed probabilistic MPC approach can be implemented online with C++, and the CPU computation time per MPC cycle is strictly less than $2ms$. Without replanning, the AGV cannot satisfy the safety constraints during the movement due to the existence of noise. Without MPC, the AGV cannot stop at the target position accurately. The proposed probabilistic MPC approach shows the capability in satisfying safety constraints and achieving optimality simultaneously. We also demonstrate the effectiveness of our algorithm with moving obstacles. The full experiment video is available at https://youtu.be/mAobl05QWyQ.

VI. Conclusion

In this paper, we studied the motion planning problem for mobile robots with process noise subject to safety constraints. The mobile robot was modeled as a nonlinear system with Gaussian process noise. To satisfy the safety constraints on the mobile robot's trajectory, a novel probabilistic MPC approach was proposed where the MPC frequency is determined according to the safety probability. We further developed an online algorithm that can guarantee the mobile robot's safety with a prescribed probability threshold. Finally, several numerical simulations, as well as physical experiments, were provided to verify the effectiveness of our proposed approach. It was shown that the probabilistic MPC approach has a lower computational burden and higher safety probability than traditional safe motion planning algorithms. The probabilistic MPC approach proposed in this paper is not limited to a single mobile robot. It can also be extended to the motion planning and formation control of multi-agent systems, which is a possible direction for future research.

References

[1] F. Liao, Y. Hu, J. Cui, Y. Tang, M. Lao, F. Lin, R. Teo, S. Lai, and J. Wang, "Motion planning of UAV platooning in unknown cluttered environment," in *2017 11th Asian Control Conference (ASCC)*. IEEE, 2017, pp. 1707–1712.

[2] L. Chen, X. Hu, W. Tian, H. Wang, D. Cao, and F.-Y. Wang, "Parallel planning: A new motion planning framework for autonomous driving," *IEEE/CAA Journal of Automatica Sinica*, vol. 6, no. 1, pp. 236–246, 2018.

[3] L. Han, F. Gao, B. Zhou, and S. Shen, "Fiesta: Fast incremental euclidean distance fields for online motion planning of aerial robots," in *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2019, pp. 4423–4430.

[4] Y. Kuwata, J. Teo, G. Fiore, S. Karaman, E. Frazzoli, and J. P. How, "Real-time motion planning with applications to autonomous urban driving," *IEEE Transactions on Control Systems Technology*, vol. 17, no. 5, pp. 1105–1118, 2009.

[5] T. D. Barfoot and C. M. Clark, "Motion planning for formations of mobile robots," *Robotics and Autonomous Systems*, vol. 46, no. 2, pp. 65–78, 2004.

[6] R. M. Murray, "Recent research in cooperative control of multivehicle systems," 2007.

[7] J.-C. Latombe, *Robot Motion Planning*. Springer Science & Business Media, 2012, vol. 124.

[8] J. W. Tolle, "Sequential quadratic programming," *Acta Numerica 1995: Volume 4*, vol. 4, pp. 1–51, 1995.

[9] J. Ziegler, P. Bender, T. Dang, and C. Stiller, "Trajectory planning for bertha — a local, continuous method," in *2014 IEEE Intelligent Vehicles Symposium Proceedings*, 2014, pp. 450–457.

[10] D. Q. Mayne, "Differential dynamic programming–a unified approach to the optimization of dynamic systems," in *Control and Dynamic Systems*. Elsevier, 1973, vol. 10, pp. 179–254.

[11] W. Li and E. Todorov, "Iterative linear quadratic regulator design for nonlinear biological movement systems." in *ICINCO (1)*. Citeseer, 2004, pp. 222–229.

[12] M. Neunert, C. de Crousaz, F. Furrer, M. Kamel, F. Farshidian, R. Siegwart, and J. Buchli, "Fast nonlinear model predictive control for unified trajectory optimization and tracking," in *2016 IEEE International Conference on Robotics and Automation (ICRA)*, 2016, pp. 1398–1404.

[13] M. Neunert, F. Farshidian, A. W. Winkler, and J. Buchli, "Trajectory optimization through contacts and automatic gait discovery for quadrupeds," *IEEE Robotics and Automation Letters*, vol. 2, no. 3, pp. 1502–1509, 2017.

[14] Y. Tassa, N. Mansard, and E. Todorov, "Control-limited differential dynamic programming," in *2014 IEEE International Conference on Robotics and Automation (ICRA)*, 2014, pp. 1168–1175.

[15] J. van den Berg, "Extended LQR: Locally-optimal feedback control for systems with non-linear dynamics and non-quadratic cost," in *Robotics Research*. Springer, 2016, pp. 39–56.

[16] J. Chen, W. Zhan, and M. Tomizuka, "Constrained iterative LQR for on-road autonomous driving motion planning," in *2017 IEEE 20th International Conference on Intelligent Transportation Systems (ITSC)*, 2017, pp. 1–7.

[17] ——, "Autonomous driving motion planning with constrained iterative LQR," *IEEE Transactions on Intelligent Vehicles*, vol. 4, no. 2, pp. 244–254, 2019.

[18] Y. Aoyama, G. Boutselis, A. Patel, and E. A. Theodorou, "Constrained differential dynamic programming revisited," *arXiv preprint arXiv:2005.00985*, 2020.

[19] Y. Pan, Q. Lin, H. Shah, and J. M. Dolan, "Safe Planning for Self-Driving Via Adaptive Constrained ILQR," in *Proc. of the IEEE International Conference on Intelligent Robots and and Systems (IROS)*, 2020.

[20] L. Armijo, "Minimization of functions having lipschitz continuous first partial derivatives," *Pacific Journal of Mathematics*, vol. 16, no. 1, pp. 1–3, 1966.

[21] Y. Koren and J. Borenstein, "Potential field methods and their inherent limitations for mobile robot navigation," in *Proceedings. 1991 IEEE International Conference on Robotics and Automation*, 1991, pp. 1398–1404 vol.2.