Nonlinear Dyn (2023) 111:21779–21796
https://doi.org/10.1007/s11071-023-08987-6

ORIGINAL PAPER

Gaussian process-based cautious model predictive control for visual servoing of mobile robots

Huazhong Zhu · Zhehao Jin · Andong Liu · Hongjie Ni

Received: 5 April 2022 / Accepted: 6 October 2023 / Published online: 8 November 2023

Abstract Mobile robots equipped with visual sensors are widely used in challenging unstructured environment due to their flexibility. However, dynamic properties of actuators are generally neglected when modeling mobile robots, which may reduce the performance of the servo controller. In this paper, we present a cautious model predictive control method for visual servoing of mobile robots with unknown actuator dynamic properties. Firstly, an enhanced model constructed by a nominal and an additive Gaussian process (GP) model is learned on-line, where the GP model is stochastic and captures dynamic properties of actuators by using the training data. Furthermore, a stochastic model predictive control (SMPC) formulation is presented for cautious control where the chance constraints of predictive states are considered to ensure visibility of the feature point. For solving the SMPC problem, an augmented deterministic model (ADM) that represents the uncertainty propagation of the stochastic state is presented to transform the SMPC formulation to a deterministic model predictive control (DMPC) formulation. Then, the DMPC problem is solved by employing a modified iterative linear quadratic regulator (iLQR) with a Lorentzian ρ-function introduced in the terminal cost function. Finally, the validity of the proposed method is validated by several examples.

H. Zhu · Z. Jin · A. Liu (✉) · H. Ni
College of Information Engineering, Zhejiang University of Technology, Hangzhou 310023, China
e-mail: lad@zjut.edu.cn

Keywords Mobile robots · Cautious model predictive control · Gaussian process · Visual servoing · ILQR

1 Introduction

In recent years, mobile robots have always been a research hotspot in the field of robotics to their flexibility and controllability [1–5]. Visual servoing control that integrates feedback control with visual information is widely used in mobile robots to control their orientation and position. By using the rich visual information, the intelligence of mobile robots can be improved and their application can be broadened [6–10]. Therefore, visual servoing control for mobile robots has received widespread attention from many scholars.

Image-based visual servoing (IBVS) controls the robot directly at the two-dimensional image plane, and its concise structure does not require 3D pose estimation in Cartesian coordinate and is insensitive to camera parameter calibration errors. Due to its advantages, many control approaches have been proposed, such as sliding mode control (SMC) [11,12], PID control [13] and adaptive control methods [14–16]. However, actual visual servoing robot systems often have constraints on velocity, position and field of view (FOV). Due to the ability to deal with visibility constraints, model predictive control (MPC) has received widespread attention [17–20]. In [17] and [18], a robust tube-based predictive control method is proposed to compensate for the effects of bounded uncertainties with the constraint state variables, such as servoing error, velocity and acceleration to stabilize the nominal visual servoing system. In [19], a MPC controller with constraints on roll and pitch angles is proposed for IBVS of a quadrotor to guarantee the visibility of feature points. In [21], a model predictive path integral (MPPI) control framework is proposed by using path integral (PI) control theory [22], which does not require the calculation of gradients and second-order approximations. In [23,24], a real-time inversion-free control method based on MPPI is proposed for both IBVS, 3D point (3DVS) and position-based visual servoing (PBVS), which has been validated on a 6-DoF Cartesian robot (namely, Gantry robot) with an eye-in-hand camera. However, due to only considering the kinematic model and ignoring the dynamic characteristics of the actuators, their performance may decrease. For example, it is assumed that the actual velocity can immediately track the desired velocity of the servo controller output, which is impractical. Thus, it is necessary to establish a velocity deviation model and incorporate it into the design of the servo controller to accurately reflect the difference between the desired velocity and the actual velocity.

In general, there are two main approaches to deal with the modeling problem. One is mechanism modeling method. For instance, a linear model has been established in [25] for a visual servo task of tendon-driven continuum robots by using the depth camera data. In [26], based on the MPC and IBVS framework, a full model of the continuum robot is derived to improve the control robustness of the system uncertainties, perceived noise and modeling error. But it is difficult to write accurate mathematical expressions for objects with complex mechanisms. The another one is black box modeling method witch uses the input–output data to construct a sufficiently accurate model for prediction, such as neural network model [27,28] or a fuzzy model [29]. However, these methods are difficult to evaluate the quality of the model on-line. The Gaussian process [30] (GP) is a nonparametric modeling approach based on a Bayesian framework [31]. Due to its limited prior knowledge [32] and the ability to directly provides model uncertainty [33], it is used to model various systems, such as race cars [34], manipulators [35] and quadcopters [36]. GP models is usually included into the MPC framework, which is termed as GP-based model predictive control (GPMPC). Since the velocity model of a vehicle is built by GP, a GPMPC method is proposed in [37] for path tracking problem of vehicles, and the performance is better than MPC method. In [38], a GPMPC method is employed in a time-varying system to deal with prediction uncertainty where the uncertainty is converted into constraints for safe operation. However, the performance index used in [37] and [38] are deterministic even though the GPR-based model is stochastic, which does not fully utilize the model information. In [32], the performance index of GPMPC is set as the expected value of the accumulated quadratic stage cost. This kind of performance index leads to a cautious control where the state area with less prediction uncertainty is more preferred. In [39], GPMPC incorporated with risk-sensitive cost is presented. Different with cautious control proposed in [32], the controller is encouraged to explore the unknown state area at a reasonable level to learn a better model, which improves the control performance. The stability of GPMPC is proved in [40].

Summarized by the above discussion, a GPMPC method is proposed for a wheeled mobile robot (WMR) to deal with the IBVS task under unknown actuator dynamic properties. Moreover, the orientation angle control is also considered to track the target pose of the WMR. The main contributions are shown as follows:

1) A GP-enhanced model instead of a pure GP model is learned on-line by constructing a nominal and a GP model. The nominal model improves the control performance in the state area which stays away from the training data set, and the GP model captures actuators' dynamic properties which can lead to differences between the nominal model and the real model.
2) To guarantee the visibility of the feature point, the chance constraints of the feature point's image coordinates are proposed and combined into the stochastic GPMPC formulation.
3) To solve the stochastic GPMPC problem, an augmented deterministic model (ADM) that represents the uncertainty propagation of the state is proposed to transform the stochastic MPC (SMPC) formulation to a deterministic model predictive control (DMPC) formulation which is solved by iterative linear quadratic regulator (iLQR).
4) Due to the static servo error that exists when using iLQR, a Lorentzian ρ-function is introduced into the terminal cost to replace the common quadratic terminal cost.

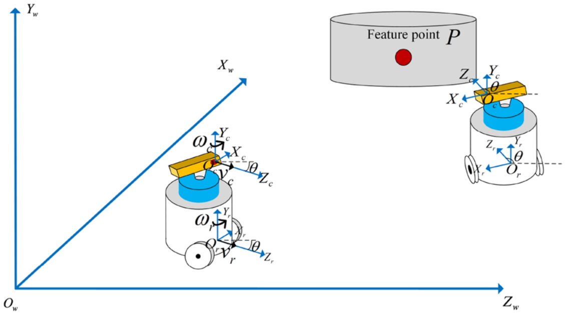


Fig. 1 Visual servoing system for a WMR

2 Model description

2.1 GP-enhanced model

The considered visual servoing system for a WMR is shown in Fig. 1, where $O_w X_w Y_w Z_w$, $O_c X_c Y_c Z_c$ and $O_r X_r Y_r Z_r$ represent the world coordinate, camera coordinate and robot coordinate, respectively. $\begin{bmatrix} v_r & w_r \end{bmatrix}^T$ denote the linear velocity and angular velocity of the WMR, and $\begin{bmatrix} v_c & w_c \end{bmatrix}^T$ denote the linear velocity and angular velocity of the camera. An identity matrix is created by setting the rotation matrix between the robot coordinates and the camera coordinates; we can obtain

$$\begin{bmatrix} v_r & w_r \end{bmatrix}^T = \begin{bmatrix} v_c & w_c \end{bmatrix}^T \tag{1}$$

The camera coordinates and image coordinates are shown in Fig. 2 where (x_i, y_i) and (x_c, y_c, z_c) are the positions of the feature point in the image coordinates and camera coordinates, respectively. The relationship between these two coordinates can be described by the camera projection model as follows

$$\begin{aligned} x_i &= x_c / z_c \\ y_i &= y_c / z_c \end{aligned} \tag{2}$$

$$\begin{bmatrix} \dot{x}_i \\ \dot{y}_i \end{bmatrix} = \begin{bmatrix} x_i / z_c & -\left(1 + x_i^2\right) \\ y_i / z_c & -x_i y_i \end{bmatrix} \begin{bmatrix} v_c \\ w_c \end{bmatrix} \tag{3}$$

In this paper, y_c is a constant since the height of the camera and the feature point are fixed. Thus, (3) can be rewritten as (4) to remove the depth information

$$\begin{bmatrix} \dot{x}_i \\ \dot{y}_i \end{bmatrix} = \begin{bmatrix} x_i y_i / y_c & -\left(1 + x_i^2\right) \\ y_i^2 / y_c & -x_i y_i \end{bmatrix} \begin{bmatrix} v_c \\ w_c \end{bmatrix} \tag{4}$$

By substituting (1) into (4), we obtain

$$\begin{bmatrix} \dot{x}_i \\ \dot{y}_i \end{bmatrix} = \begin{bmatrix} x_i y_i / y_c & -\left(1 + x_i^2\right) \\ y_i^2 / y_c & -x_i y_i \end{bmatrix} \begin{bmatrix} v_r \\ w_r \end{bmatrix} \tag{5}$$

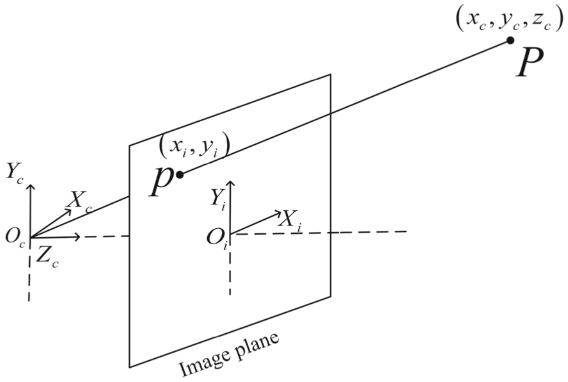


Fig. 2 Projection relations of the image frame and the camera frame

Moreover, the orientation angle $\dot{\theta} = w_r$ is also introduced into (5) as follows

$$\begin{bmatrix} \dot{x}_i \\ \dot{y}_i \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} x_i y_i / y_c & -\left(1 + x_i^2\right) \\ y_i^2 / y_c & -x_i y_i \\ 0 & 1 \end{bmatrix} \begin{bmatrix} v_r \\ w_r \end{bmatrix} \tag{6}$$

To use MPC method, (6) is discretized as follows

$$\begin{bmatrix} x_{i,k+1} \\ y_{i,k+1} \\ \theta_{k+1} \end{bmatrix} = \begin{bmatrix} x_{i,k} \\ y_{i,k} \\ \theta_k \end{bmatrix} + T \begin{bmatrix} x_{i,k} y_{i,k} / y_c & -\left(1 + x_{i,k}^2\right) \\ y_{i,k}^2 / y_c & -x_{i,k} y_{i,k} \\ 0 & 1 \end{bmatrix} \times \begin{bmatrix} v_{r,k} \\ w_{r,k} \end{bmatrix} \tag{7}$$

where T and k are the sampling time and the time index.

However, the desired velocity $\begin{bmatrix} v_{r,d} & w_{r,d} \end{bmatrix}^T$ is the output of the visual servo controller in practical and the real velocity $\begin{bmatrix} v_r & w_r \end{bmatrix}^T$ is not directly controllable. Thus, the velocity model that represents the relationship between $\begin{bmatrix} v_{r,d} & w_{r,d} \end{bmatrix}^T$ and $\begin{bmatrix} v_r & w_r \end{bmatrix}^T$ should also be considered to design an efficient visual servo controller. Generally, the velocity model is

$$\begin{bmatrix} v_{r,k} & w_{r,k} \end{bmatrix}^T = f\left(v_{r,k-1}, w_{r,k-1}, v_{r,d,k}, w_{r,d,k}\right) \tag{8}$$

In this paper, (8) is written as follows

$$\begin{bmatrix} v_{r,k} \\ w_{r,k} \end{bmatrix} = \begin{bmatrix} v_{r,d,k} \\ w_{r,d,k} \end{bmatrix} + g\left(v_{r,k-1}, w_{r,k-1}, v_{r,d,k}, w_{r,d,k}\right) \tag{9}$$

By combining (9) and (7), we have the following augmented model

$$\left[x_{i,k+1}\ y_{i,k+1}\ \theta_{k+1}\ v_{r,k}\ w_{r,k}\right]^T = m\left(x_{i,k}, y_{i,k}, \theta_k, v_{r,d,k}, w_{r,d,k}\right) + h\left(x_{i,k}, y_{i,k}, \theta_k, v_{r,k-1}, w_{r,k-1}, v_{r,d,k}, w_{r,d,k}\right) + \varepsilon_k \tag{10}$$

where

$$m\left(x_{i,k}, y_{i,k}, \theta_k, v_{r,d,k}, w_{r,d,k}\right) = \begin{bmatrix} x_{i,k} \\ y_{i,k} \\ \theta_k \\ 0 \\ 0 \end{bmatrix} + T\begin{bmatrix} x_{i,k}y_{i,k}/y_c & -\left(1+x_{i,k}^2\right) \\ y_{i,k}^2/y_c & -x_{i,k}y_{i,k} \\ 0 & 1 \\ 1/T & 0 \\ 0 & 1/T \end{bmatrix}\begin{bmatrix} v_{r,d,k} \\ w_{r,d,k} \end{bmatrix} \tag{11}$$

is a nominal model which is usually directly used to design a servo controller in most works, and

$$h\left(x_{i,k}, y_{i,k}, \theta_k, v_{r,k}, w_{r,k}, v_{r,d,k}, w_{r,d,k}\right) = T\begin{bmatrix} x_{i,k}y_{i,k}/y_c & -\left(1+x_{i,k}^2\right) \\ y_{i,k}^2/y_c & -x_{i,k}y_{i,k} \\ 0 & 1 \\ 1/T & 0 \\ 0 & 1/T \end{bmatrix} g\left(v_{r,k-1}, w_{r,k-1}, v_{r,d,k}, w_{r,d,k}\right) \tag{12}$$

is an additive model which contain the information about dynamic properties of WMR's actuators. Moreover, we also consider the i.i.d process noise $\varepsilon_k \sim N\left(0, \Sigma_\varepsilon\right)$ with $\Sigma_\varepsilon = diag\left(\left(\sigma_\varepsilon^1\right)^2, \ldots, \left(\sigma_\varepsilon^5\right)^2\right)$. The GP method is employed in this paper to approximate the function h, and thus, (10) is named as the GP-enhanced model. For convenience, we define $o_k = \left[x_{i,k+1}\ y_{i,k+1}\ \theta_{k+1}\ v_{r,k}\ w_{r,k}\right]^T$ and $u_k = \left[v_{r,d,k}\ w_{r,d,k}\right]^T$ as the state and input of the GP-enhanced model at time index k, respectively.

2.2 GP modeling

The Gaussian process regression (GPR) method can use the previously collected data set to describe the additive model h. More concretely, five independent GP models are built where each model is corresponding to an output dimension of h.

At time index k, the state o_k has been observed and the value $h\left(o_k, u_k\right)$ is expected to be inferred. Labels of the data set for i^{th} model are set as follows

$$\overrightarrow{y}_{k-1}^{i} = \left[\left(o_2 - m_1\right)^i \ \ldots \ \left(o_{j+1} - m_j\right)^i \ \ldots \ \left(o_k - m_{k-1}\right)^i\right]^T \tag{13}$$

where o_j and m_j denote the value of the state and output of the nominal model at time index j, respectively. $\left(o_{j+1} - m_j\right)^i$ represents the i^{th} dimension of the $\left(o_{j+1} - m_j\right)$. Features for each GP model are the same and defined as

$$Z_{k-1} = \left[z_1\ \ldots\ z_j\ \ldots\ z_{k-1}\right]^T \tag{14}$$

where $z_j = \left[o_j^T\ u_j^T\right]^T$. The relationship between labels and features is as follows

$$y_j^i = \left(o_{j+1} - m_j\right)^i = h^i\left(z_j\right) + \varepsilon_j^i \tag{15}$$

where ε_j^i represents i^{th} dimension of the ε_j, and $\varepsilon_j^i \sim N\left(0, \left(\sigma_\varepsilon^i\right)^2\right)$. By using the GPR method, $\overset{\rightharpoonup}{h}^i = \left[h^i\left(z_1\right) \ldots h^i\left(z_{k-1}\right)\right]^T$ is assumed to satisfy a multivariate Gaussian distribution as

$$\overset{\rightharpoonup}{h}^i \sim N\left(\begin{bmatrix} \beta^i\left(z_1\right) \\ \vdots \\ \beta^i\left(z_{k-1}\right) \end{bmatrix}, \begin{bmatrix} \varphi^i\left(z_1, z_1\right) & \ldots & \varphi^i\left(z_1, z_1\right) \\ \vdots & \ddots & \vdots \\ \varphi^i\left(z_{k-1}, z_1\right) & \ldots & \varphi^i\left(z_{k-1}, z_{k-1}\right) \end{bmatrix}\right) \tag{16}$$

where $\varphi^i\left(\cdot, \cdot\right)$ is the kernel function and $\beta^i\left(\cdot\right)$ is the mean function of the GPR method. The mean function can be arbitrarily set. For convenience, we set $\beta^i\left(\cdot\right) = 0$. The kernel function should be designed such that $\Phi_{k-1}^i = \begin{bmatrix} \varphi^i\left(z_1, z_1\right) & \ldots & \varphi^i\left(z_1, z_1\right) \\ \vdots & \ddots & \vdots \\ \varphi^i\left(z_{k-1}, z_1\right) & \ldots & \varphi^i\left(z_{k-1}, z_{k-1}\right) \end{bmatrix}$ is positive semidefinite or positive definite. Here, the kernel function is set as

$$\varphi^i\left(z_a, z_b\right) = \left(\sigma_s^i\right)^2 e^{-\frac{1}{2}\left(z_a - z_b\right)^T \Lambda^i \left(z_a - z_b\right)} \tag{17}$$

where Λ^i and $\left(\left(\sigma_s^i\right)^2, \Lambda^i\right)$ are the diagonal matrix and the hyper parameters of the GPR [30], respectively. That can be decided by maximizing the log-likelihood function given by

$$\begin{aligned}&\log p\left(\overrightarrow{y}^{i}_{k-1}|Z_{k-1},\left(\sigma^i_s\right)^2,\Lambda^i\right)\\&=-\frac{1}{2}\left(\overrightarrow{y}^{i}_{k-1}\right)^T\left(\Phi^i_{k-1}+\left(\sigma^i_\varepsilon\right)^2 I\right)^{-1}\overrightarrow{y}^{i}_{k-1}\\&\quad-\frac{1}{2}\log\left|\left(\Phi^i_{k-1}+\left(\sigma^i_\varepsilon\right)^2 I\right)^{-1}\right|-\frac{7}{2}\log\left(2\pi\right)\end{aligned}\tag{18}$$

After determining the hyperparameters, the joint distribution of $h^i\left(z_k\right)$ and $\overrightarrow{y}^{i}_{k-1}$ is shown as follows

$$\begin{aligned}&\left[\begin{array}{c}\overrightarrow{y}^{i}_{k-1}\\h^i\left(z_k\right)\end{array}\right]\Bigg|Z_{k-1},\left(\sigma^i_s\right)^2,\Lambda^i\sim\\&N\left(\overrightarrow{0},\left[\begin{array}{cc}\Phi^i_{k-1}+\left(\sigma^i_\varepsilon\right)^2 I&\overrightarrow{\varphi}^{i}\\\left(\overrightarrow{\varphi}^{i}\right)^T&\varphi^i\left(z_k,z_k\right)\end{array}\right]\right)\end{aligned}\tag{19}$$

where $\overrightarrow{\varphi}^{i}=\left[\varphi^i\left(z_1,z_k\right)\ldots\varphi^i\left(z_{k-1},z_k\right)\right]^T$. Applying the conditional Gaussian rules [30], the posterior distribution of $h^i\left(z_k\right)$ can be obtained as

$$\begin{aligned}&h^i\left(z_k\right)\Bigg|Z_{k-1},\left(\sigma^i_s\right)^2,\Lambda^i,\overrightarrow{y}^{i}_{k-1}\sim\\&N\left(\mu^i_{h,k},\left(\sigma^i_{h,k}\right)^2\right)\end{aligned}\tag{20}$$

where

$$\mu^i_{h,k}=\left(\overrightarrow{\varphi}^{i}\right)^T\left(\Phi^i_{k-1}+\left(\sigma^i_\varepsilon\right)^2 I\right)^{-1}\overrightarrow{y}^{i}_{k-1}\tag{21}$$

$$\begin{aligned}\left(\sigma^i_{h,k}\right)^2&=\varphi^i\left(z_k,z_k\right)\\&\quad-\left(\overrightarrow{\varphi}^{i}\right)^T\left(\Phi^i_{k-1}+\left(\sigma^i_\varepsilon\right)^2 I\right)^{-1}\overrightarrow{\varphi}^{i}\end{aligned}\tag{22}$$

Then, the posterior distribution of $h\left(z_k\right)$ which is expected to be inferred can be computed as follows [30]

$$\begin{aligned}&h\left(z_k\right)\Bigg|Z_{k-1},\left(\sigma^i_s\right)^2,\Lambda^i,\overrightarrow{y}^{i}_{k-1,i=1,\ldots,5}\sim\\&N\left(\mu_{h,k},\Sigma_{h,k}\right)\end{aligned}\tag{23}$$

where

$$\begin{aligned}&\mu_{h,k}=\left[\mu^1_{h,k}\ \mu^2_{h,k}\ \mu^3_{h,k}\ \mu^4_{h,k}\ \mu^5_{h,k}\right],\Sigma_{h,k}\\&=diag\left(\left(\sigma^1_{h,k}\right)^2,\left(\sigma^2_{h,k}\right)^2,\left(\sigma^3_{h,k}\right)^2,\left(\sigma^4_{h,k}\right)^2,\left(\sigma^5_{h,k}\right)^2\right).\end{aligned}$$

Based on the posterior distribution of $h\left(z_k\right)$, the distribution of o_{k+1} can be easily derived as

$$o_{k+1}\sim N\left(\mu_{o,k+1},\Sigma_{o,k+1}\right)\tag{24}$$

where

$$\mu_{o,k+1}=m_k+\mu_{h,k}\tag{25}$$

$$\Sigma_{o,k+1}=\Sigma_{h,k}+\Sigma_\varepsilon\tag{26}$$

3 MPC controller design

With the GP-enhanced model, the MPC problem is exactly nonlinear and stochastic, which is given as follows

$$\min_{U_k} J=E\left[\sum_{t=k}^{k+M-1}l\left(o_t,u_t\right)+l_f\left(o_{k+M}\right)\right]\tag{27a}$$

$$\begin{aligned}&s.t.\quad\forall i=1,\ldots,5,\ \forall j=1,2\,and\,\forall t=k,\ldots,\\&k+M-1:\end{aligned}$$

$$p\left(o_{t+1}|o_k,u_k\right)\sim N\left(\mu_{o,t+1},\Sigma_{o,t+1}\right)\tag{27b}$$

$$u^j_t\geq u^j_{\min}\tag{27c}$$

$$u^j_t\leq u^j_{\max}\tag{27d}$$

$$p\left(o^i_{t+1}\geq o^i_{\min}|o_k,u_k\right)\geq c\tag{27e}$$

$$p\left(o^i_{t+1}\leq o^i_{\max}|o_k,u_k\right)\geq c\tag{27f}$$

where $U_k=\left[u_k\ \ldots\ u_{k+M-1}\right]^T$ is the control sequence which is need to be searched, the stage cost function $l\left(o_t,u_t\right)$ is defined as a quadratic function $l\left(o_t,u_t\right)=\left\|o_t-o^*\right\|_Q+\left\|u_t\right\|_R$ and the terminal cost function is defined as follows

$$l_f\left(o_{k+M}\right)=\eta d^2+\lambda\log\left(d^2+\alpha\right)\tag{28}$$

where $d^2=\left\|o_{k+M}-o^*\right\|_Q$ and o^* is the target state, and the Lorentzian ρ-function $\log\left(d^2+\alpha\right)$ is introduced into the terminal cost function to encourage accuracy placement into the o^*, which is shown in Fig. 3 (Assuming $\eta=\lambda=1$).

It should be noted that the performance index (27a) is an expected value, which is different with traditional MPC problem. Thus, the expected values of the $l\left(o_t,u_t\right)$ and $l_f\left(o_{k+M}\right)$ need to be derived. However, accurate closed forms of these expected values cannot be obtained. Thus, the second-order Taylor expansion

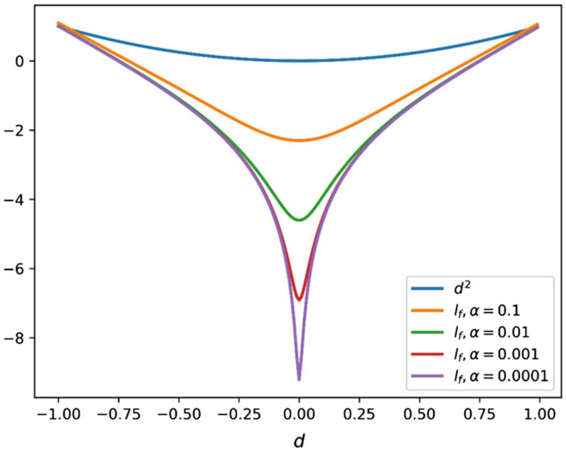


Fig. 3 Shape of the terminal cost function

technique can be used to approximate the expected values as

$$\begin{aligned}
&E\left[l\left(o_t, u_t\right)\right] = \int p\left(o_t\right) l\left(o_t, u_t\right) do_t \\
&\approx \int p\left(o_t\right)\left(\left\|\mu_{o,t} - o^*\right\|_Q + \left(o_t - \mu_{o,t}\right)^T \tfrac{\partial l(o_t, u_t)}{\partial o_t}\Big|_{o_t=\mu_{o,t}}\right. \\
&\quad + \tfrac{1}{2}\left(o_t - \mu_{o,t}\right)^T \tfrac{\partial^2 l(o_t, u_t)}{\partial {o_t}^2}\Big|_{o_t=\mu_{o,t}} \\
&\quad \left.\times\left(o_t - \mu_{o,t}\right)\right) do_t \qquad (29)\\
&\quad + \left\|u_t\right\|_R = \left\|\mu_{o,t} - o^*\right\|_Q + 0 + tr\left(\tfrac{1}{2}\int p\left(o_t\right)\right. \\
&\quad \left.\times\left(o_t - \mu_{o,t}\right)\left(o_t - \mu_{o,t}\right)^T do_t \tfrac{\partial^2 l(o_t, u_t)}{\partial {o_t}^2}\Big|_{o_t=\mu_{o,t}}\right) + \left\|u_t\right\|_R \\
&= \left\|\mu_{o,t} - o^*\right\|_Q + tr\left(\Sigma_{o,t} Q\right) + \left\|u_t\right\|_R
\end{aligned}$$

$$\begin{aligned}
&E\left[l_f\left(o_{k+M}\right)\right] = \int p\left(o_{k+M}\right) l_f\left(o_{k+M}\right) do_{k+M} \\
&\approx \int p\left(o_{k+M}\right)\left(\left\|\mu_{o,k+M} - o^*\right\|_Q + \left(o_{k+M} - \mu_{o,k+M}\right)^T\right. \\
&\quad \times \tfrac{\partial l_f(o_{k+M})}{\partial o_{k+M}}\Big|_{o_{k+M}=\mu_{o,k+M}} \qquad (30)\\
&\quad + \tfrac{1}{2}\left(o_{k+M} - \mu_{o,k+M}\right)^T \tfrac{\partial^2 l(o_{k+M})}{\partial {o_{k+M}}^2}\Big|_{o_{k+M}=\mu_{o,k+M}} \\
&\quad \left.\times\left(o_{k+M} - \mu_{o,k+M}\right)\right) do_{k+M} \\
&= \eta\mu_{d^2,k+M} + \lambda \log\left(\mu_{d^2,k+M} + \alpha\right) + tr\left(\Sigma_{o,t} B\right)
\end{aligned}$$

where

$$\mu_{d^2,k+M} = \left\|\mu_{o,k+M} - o^*\right\|_Q \qquad (31)$$

$$\begin{aligned}
B = \eta Q + \lambda &\left(\frac{1}{\mu_{d^2,k+M} + \alpha} Q - \frac{2}{\left(\mu_{d^2,k+M} + \alpha\right)^2}\right. \\
&\left. Q\left(\mu_{o,k+M} - o^*\right)\left(\mu_{o,k+M} - o^*\right)^T Q\right) \qquad (32)
\end{aligned}$$

when Q is a weighted matrix with $Q \geq 0$, and the second part of (29) can be represented as $tr\left(\Sigma_{o,t} Q\right) = \Sigma_{o,t}^{(1,1)} Q^{(1,1)} + \ldots + \Sigma_{o,t}^{(5,5)} Q^{(5,5)}$, which penalizes the predicted state with large variances. Thus, the GPMPC controller with such stage cost function prefers the predicted state area with less uncertainty and behaves cautiously.

However, since the model (27b) and constraints (27e) and (27f) are stochastic, the optimization problem (27) is computationally intractable. Thus, several techniques are presented in the remainder of this section to transform (27) into a tractable optimization problem.

3.1 Uncertainty propagation

In this subsection, the specific form of $\mu_{o,t+1}$ and $\Sigma_{o,t+1}$ in the stochastic model (27b) is derived. However, for long-term prediction, $\left(\mu_{o,k+2}, \Sigma_{o,k+2}\right), \ldots, \left(\mu_{o,k+M}, \Sigma_{o,k+M}\right)$ should also be computed, which are more complex compared with computation for $\left(\mu_{o,k+1}, \Sigma_{o,k+1}\right)$ since features $z_{k+1}, \ldots, z_{k+M-1}$ for prediction are stochastic variables.

Only the derivation of $\left(\mu_{o,k+2}, \Sigma_{o,k+2}\right)$ is presented here since the method to derive other mean–variance pairs is similar. The feature z_{k+1} is a variable which satisfies a Gaussian distribution as

$$\begin{aligned}
&z_{k+1} \sim N\left(\mu_{z,k+1}, \Sigma_{z,k+1}\right) \\
&= N\left(\begin{bmatrix} \mu_{o,k+1} \\ u_{k+1} \end{bmatrix}, \begin{bmatrix} \Sigma_{o,k+1} & 0_{5\times 2} \\ 0_{2\times 5} & 0_{2\times 2} \end{bmatrix}\right) \qquad (33)
\end{aligned}$$

where $\left(\mu_{o,k+1}, \Sigma_{o,k+1}\right)$ can be computed based on (25) and (26). Under the circumstance of the uncertain input, $\mu_{o,k+2}$ can be computed as

$$\begin{aligned}
&\mu_{o,k+2} = E_{z_{k+1}, h_{k+1}}\left[o_{k+2}\right] \\
&= \iint p\left(z_{k+1}\right) p\left(h_{k+1} | z_{k+1}\right)\left(m_{k+1} + h_{k+1} + \varepsilon_{k+1}\right) \\
&dz_{k+1} dh_{k+1} \\
&= \int p\left(z_{k+1}\right) m_{k+1} dz_{k+1} + \iint p\left(z_{k+1}\right) p\left(h_{k+1} | z_{k+1}\right) \\
&h_{k+1} dz_{k+1} dh_{k+1} \\
&= \int p\left(z_{k+1}\right) m_{k+1} dz_{k+1} + \int p\left(z_{k+1}\right) \mu_{h,k+1} dz_{k+1} \\
&\approx m_{k+1}\left(\mu_{z,k+1}\right) + \mu_{h,k+1}\left(\mu_{z,k+1}\right)
\end{aligned} \qquad (34)$$

where the final approximate equation is obtained by using the first-order Taylor expansion of m_{k+1} and $\mu_{h,k+1}$ around $\mu_{z,k+1}$. Although an accurate solution of $\int p\left(z_{k+1}\right) \mu_{h,k+1} dz_{k+1}$ can be computed as in [41], the Taylor expansion approximation is used to simplify calculation. Covariance matrix can be obtained based on the above approximated mean value as

$$\begin{aligned}
&\Sigma_{z,k+1} \\
&= E_{z_{k+1}, h_{k+1}}\left[\left(o_{k+2} - \mu_{o,k+2}\right)\left(o_{k+2} - \mu_{o,k+2}\right)^T\right]
\end{aligned}$$

$$
\begin{aligned}
&= E_{z_{k+1},h_{k+1}}\Bigg[\Bigg(\underbrace{\left(\frac{\partial m_{k+1}}{\partial z_{k+1}}\,|_{z_{k+1}=\mu_{z,k+1}}\right)\left(z_{k+1}-\mu_{z,k+1}\right)}_{\Delta m_{k+1}} \\
&\quad + \underbrace{h_{k+1}-\mu_{h,k+1}\left(\mu_{z,k+1}\right)}_{\Delta h_{k+1}} + \varepsilon_{k+1}\Bigg) \\
&\quad \left(\Delta m_{k+1}+\Delta h_{k+1}+\varepsilon_{k+1}\right)^T\Bigg] \\
&= \int \Delta m_{k+1}\Delta m_{k+1}^T p\left(z_{k+1}\right) dz_{k+1} \\
&\quad + \iint \Delta m_{k+1}\Delta h_{k+1}{}^T p\left(z_{k+1}\right) p\left(h_{k+1}|z_{k+1}\right) \\
&\quad \times dz_{k+1}dh_{k+1} \\
&\quad + \iint \Delta h_{k+1}\Delta m_{k+1}^T p\left(z_{k+1}\right) p\left(h_{k+1}|z_{k+1}\right) \\
&\quad \times dz_{k+1}dh_{k+1} \\
&\quad + \iint \Delta h_{k+1}\Delta h_{k+1}{}^T p\left(z_{k+1}\right) p\left(h_{k+1}|z_{k+1}\right) \\
&dz_{k+1}dh_{k+1} + \Sigma_\varepsilon
\end{aligned}
\tag{35}
$$

The remaining work is to compute integrals. The first integral can be computed as

$$
\begin{aligned}
&\int \Delta m_{k+1}\Delta m_{k+1}^T p\left(z_{k+1}\right) dz_{k+1} \\
&= \left(\frac{\partial m_{k+1}}{\partial z_{k+1}}\,|_{z_{k+1}=\mu_{z,k+1}}\right)\int p\left(z_{k+1}\right) \\
&\left(z_{k+1}-\mu_{z,k+1}\right)\left(z_{k+1}-\mu_{z,k+1}\right)^T \\
&dz_{k+1}\left(\frac{\partial m_{k+1}}{\partial z_{k+1}}\,|_{z_{k+1}=\mu_{z,k+1}}\right)^T \\
&= \left(\frac{\partial m_{k+1}}{\partial z_{k+1}}\,|_{z_{k+1}=\mu_{z,k+1}}\right)\Sigma_{z,k+1}\left(\frac{\partial m_{k+1}}{\partial z_{k+1}}\,|_{z_{k+1}=\mu_{z,k+1}}\right)^T
\end{aligned}
\tag{36}
$$

The value of the second integral can be obtained as

$$
\begin{aligned}
&\iint \Delta m_{k+1}\Delta h_{k+1}{}^T p\left(z_{k+1}\right) p\left(h_{k+1}|z_{k+1}\right) dz_{k+1}dh_{k+1} \\
&= \int p\left(z_{k+1}\right)\Delta m_{k+1}\int p\left(h_{k+1}|z_{k+1}\right) \\
&\left(h_{k+1}-\mu_{h,k+1}\left(\mu_{z,k+1}\right)\right)^T dh_{k+1}dz_{k+1} \\
&= \int p\left(z_{k+1}\right)\Delta m_{k+1}\left(\mu_{h,k+1}-\mu_{h,k+1}\left(\mu_{z,k+1}\right)\right) dz_{k+1} \\
&\approx 0
\end{aligned}
\tag{37}
$$

where the final approximate equation is obtained by using the first-order Taylor expansion of $\Delta m_{k+1}\left(\mu_{h,k+1}-\mu_{h,k+1}\left(\mu_{z,k+1}\right)\right)$ around $\mu_{z,k+1}$. The value of the third integral which can be derived by using the same method as the second integral is also zero. The fourth integral can be computed as

$$
\begin{aligned}
&\iint \Delta h_{k+1}\Delta h_{k+1}{}^T p\left(z_{k+1}\right) p\left(h_{k+1}|z_{k+1}\right) dz_{k+1}dh_{k+1} \\
&= \int p\left(z_{k+1}\right)\int p\left(h_{k+1}|z_{k+1}\right)\Delta h_{k+1}\Delta h_{k+1}{}^T dh_{k+1}dz_{k+1} \\
&\approx \int p\left(z_{k+1}\right)\int p\left(h_{k+1}|z_{k+1}\right) h_{k+1}h_{k+1}^T dh_{k+1}dz_{k+1} \\
&\quad -\mu_{h,k+1}\left(\mu_{z,k+1}\right)\mu_{h,k+1}^T\left(\mu_{z,k+1}\right) \\
&= \int p\left(z_{k+1}\right)\left(\Sigma_{h,k+1}+\mu_{h,k+1}\mu_{h,k+1}^T\right) dz_{k+1} \\
&\quad -\mu_{h,k+1}\left(\mu_{z,k+1}\right)\mu_{h,k+1}^T\left(\mu_{z,k+1}\right) \\
&\approx \Sigma_{h,k+1}\left(\mu_{z,k+1}\right)
\end{aligned}
\tag{38}
$$

where the second approximate equation and the final approximate equation are obtained by using the first-order Taylor expansion of $\mu_{h,k+1}$ and $\Sigma_{h,k+1}+\mu_{h,k+1}\mu_{h,k+1}^T$ around $\mu_{z,k+1}$, respectively. Thus, based on (35)-(38), the approximated covariance matrix is as follows

$$
\begin{aligned}
&\Sigma_{o,k+2} \approx \left(\frac{\partial m_{k+1}}{\partial z_{k+1}}\,|_{z_{k+1}=\mu_{z,k+1}}\right) \\
&\Sigma_{z,k+1}\left(\frac{\partial m_{k+1}}{\partial z_{k+1}}\,|_{z_{k+1}=\mu_{z,k+1}}\right)^T + \Sigma_{h,k+1}\left(\mu_{z,k+1}\right) \\
&+\Sigma_\varepsilon
\end{aligned}
\tag{39}
$$

where

$$
\begin{aligned}
\mu_{z,k+1} &= \begin{bmatrix}\mu_{o,k+1} & u_{k+1}\end{bmatrix}^T \\
\Sigma_{z,k+1} &= \begin{bmatrix}\Sigma_{o,k+1} & 0_{5\times 2} \\ 0_{2\times 5} & 0_{2\times 2}\end{bmatrix}
\end{aligned}
\tag{40}
$$

It can be found that (25) and (26) are special forms for (34) and (39), respectively, by setting $z_k = \mu_{z,k}$ and $\Sigma_{z,k} = 0_{5\times 5}$.

Although the distribution of o_{k+2} is not exactly Gaussian, it is approximated into a Gaussian distribution by using the derived $\left(\mu_{o,k+2}, \Sigma_{o,k+2}\right)$. Then, the chance constraints (27e) and (27f) can be transformed into a more tractable formulation by using the quantile function of Gaussian distribution. In this paper, c is set as 0.95, which lead to the following constraints

$$
\begin{aligned}
o_{t+1}^i - 2\Sigma_{o,t+1}^{(i,i)} &\geq o_{\min}^i \\
o_{t+1}^i + 2\Sigma_{o,t+1}^{(i,i)} &\leq o_{\max}^i
\end{aligned}
\tag{41}
$$

for all $t = k, \ldots, k+M-1$ and $i = 1, 2, 3, 4, 5$.

3.2 Tractable MPC design

In this subsection, the intractable SMPC formulation is transformed to a tractable DMPC formulation by designing an ADM based on the derivations in subsection 3.1, and a modified iLQG approach is proposed to deal with the DMPC problem.

The state of the ADM at time index t is defined as

$$s_t = \left[\mu_{o,t}^T \; vec\left(L_{o,t}\right)^T \right]^T \tag{42}$$

where $L_{o,t} = \left[l_1 \dots l_5 \right]$ is a lower triangular matrix which is obtained by using Cholesky decomposition of $\Sigma_{o,t}$, and $vec\left(L_{o,t}\right) = \left[l_1^T \dots l_5^T \right]^T$.

Remark 1 We set $L_{o,k} = \overrightarrow{0}$ since $\Sigma_{o,k} = 0_{5\times5}$, and for $t = k+1, \dots k+M$, $L_{o,t}$ can be obtained by using Cholesky decomposition of $\Sigma_{o,t}$ since $\Sigma_{o,t}$ is a positive definite matrix.

The ADM is represented by $s_{t+1} = F\left(s_t, u_t\right)$ which can be easily derived by using (34) and (39). Under the DMPC formulation, the expected value of the $l_s\left(o_t, u_t\right)$ and $l_f\left(o_{k+M}\right)$ can be transformed as

$$l\left(s_t, u_t\right) = E\left[l\left(o_t, u_t\right)\right] == \left\| s_t - s^* \right\|_{Q_1} + \left\| u_t \right\|_R \tag{43}$$

and

$$\begin{aligned} l_f\left(s_{k+M}\right) &= E\left[l_f\left(o_{k+M}\right)\right] = \left\| s_{k+M} - s^* \right\|_{Q_2} \\ &+ \lambda \log\left(\left\| s_{k+M} - s^* \right\|_{Q_3} + \alpha \right) \end{aligned} \tag{44}$$

,

where $s^* = \left[\left(o^*\right)^T, \overrightarrow{0}^T \right]^T \in R^{30}$, $Q_1 = diag\left(\underbrace{Q, \dots, Q}_{6} \right) \in R^{30\times30}$,

$Q_2 = diag\left(Q, \underbrace{B, \dots B}_{5} \right)$,

$Q_3 = dig\left(Q, \underbrace{0_{5\times5}, \dots 0_{5\times5}}_{5} \right)$. Constraints in (41) are equivalent to

$$\begin{aligned} & s_{t+1}^T e^i - 2\left(L_{o,t} L_{o,t}^T\right)^{(i,i)} \geq s_{\min}^i \\ & s_{t+1}^T e^i + 2\left(L_{o,t} L_{o,t}^T\right)^{(i,i)} \leq s_{\max}^i \end{aligned} \tag{45}$$

for $i = 1, \dots, 5$, where e^i is the i^{th} column vector of an identity matrix. Then, the DMPC problem is given as follows

$$\min_{U_k} J_k = \sum_{t=k}^{k+M-1} l\left(s_t, u_t\right) + l_f\left(s_{k+M}\right) \tag{46a}$$

$$s.t. \quad \forall i = 1, \dots, 5, \forall j = 1, 2 and \forall t = k, \dots, k + M - 1$$

$$s_{t+1} = F\left(s_t, u_t\right) \tag{46b}$$

$$u_t^j \geq u_{\min}^j \tag{46c}$$

$$u_t^j \leq u_{\max}^j \tag{46d}$$

$$s_{t+1}^T e^i - 2\left(L_{o,t} L_{o,t}^T\right)^{(i,i)} \geq s_{\min}^i \tag{46e}$$

$$s_{t+1}^T e^i + 2\left(L_{o,t} L_{o,t}^T\right)^{(i,i)} \leq s_{\max}^i \tag{46f}$$

To simplify the DMPC problem, constraints (46c) - (46f) are softened the performance index by introducing several barrier functions as

$$J_k' = J_k + \sum_{t=k}^{k+M-1} l_{bs}\left(s_{t+1}\right) + l_{bu}\left(u_t\right) \tag{47}$$

where

$$\begin{aligned} l_{bs}\left(s_{t+1}\right) = & \sum_{i=1}^{5} b_1 r\left(s_{t+1}^T e^i + 2\left(L_{o,t} L_{o,t}^T\right)^{(i,i)} - s_{\max}^i \right. \\ & \left. + b_2 r\left(s_{\min}^i - s_{t+1}^T e^i + 2\left(L_{o,t} L_{o,t}^T\right)^{(i,i)} \right) \right) \end{aligned} \tag{48}$$

$$l_{bu}\left(u_t\right) = \sum_{j=1}^{2} b_3 r\left(u_t^j - u_{\max}^j\right) + b_4 r\left(u_{\min}^j - u_t^j\right) \tag{49}$$

$$r(*) = \begin{cases} 0, * \leq 0 \\ *, * \geq 0 \end{cases} \tag{50}$$

To facilitate the subsequent introduction of iLQR method, J_k' is rewritten as

$$J_k' = l'\left(s_t, u_t\right) + l_f'\left(s_{k+M}\right) \tag{51}$$

where

$$l'\left(s_t, u_t\right) = l\left(s_t, u_t\right) + l_{bs}\left(s_t\right) + l_{bu}\left(u_t\right) \tag{52}$$

$$l_f'\left(s_{k+M}\right) = l_f\left(s_{k+M}\right) + l_{bs}\left(s_{k+M}\right) \tag{53}$$

With the above settings, the considered DMPC problem is simplified as

$$\min_{U_k} J_k' = \sum_{t=k}^{k+M-1} l'\left(s_t, u_t\right) + l'_f\left(s_{k+M}\right) \tag{54a}$$

$$s.t. \quad s_{t+1} = F\left(s_t, u_t\right) \tag{54b}$$

The iLQR [42] approach is employed to address the above DMPC issue through trajectory optimization. The value function of x_t is defined as cost-to-go, and written as

$$V\left(x_t\right) = \min_{U_t} \sum_{t=k}^{k+M-1} l'\left(s_t, u_t\right) + l'_f\left(s_{k+M}\right) \tag{55}$$

where $U_t = \left[u_t \dots u_{t+M-1} \right]$. The value function of x_{k+M} is set as $V\left(x_{k+M}\right) = l_f'\left(s_{k+M}\right)$. Then, the iLQR can perform minimization sequentially on a single control unit rather than minimizing over the entire control sequence by proceeding backward in time as

$$\begin{aligned} V(x_t) &= \min_{u_t} l'(s_t, u_t) + V(s_{t+1}) = \min_{u_t} l'(s_t, u_t) \\ &\quad + V(F(s_t, u_t)) \end{aligned} \tag{56}$$

To solve the above minimization problem, the Q function is defined as a perturbation function around the current state–input pair as

$$\begin{aligned} &Q(\delta s_t, \delta u_t) = l'(s_t + \delta s_t, u_t + \delta u_t) - l'(s_t, u_t) \\ &+ V(F(s_t + \delta s_t, u_t + \delta u_t)) - V(F(s_t, u_t)) \end{aligned} \tag{57}$$

The minimization problem (56) is then transformed to find an optimal δu_t^* that minimizes $Q(\delta s_t, \delta u_t)$. Function $Q(\delta s_t, \delta u_t)$ can expand to a second order as

$$Q(\delta s_t, \delta u_t) \approx \frac{1}{2} \begin{bmatrix} 1 \\ \delta s_t \\ \delta u_t \end{bmatrix}^T \begin{bmatrix} 0 & Q_{s_t}^T & Q_{u_t}^T \\ Q_{s_t} & Q_{s_t s_t} & Q_{s_t u_t} \\ Q_{u_t} & Q_{u_t s_t} & Q_{u_t u_t} \end{bmatrix} \begin{bmatrix} 1 \\ \delta s_t \\ \delta u_t \end{bmatrix} \tag{58}$$

where

$$Q_{s_t} = l'_{s_t} + F_{s_t}^T V_{s_{t+1}} \tag{59a}$$

$$Q_{u_t} = l'_{u_t} + F_{u_t}^T V_{s_{t+1}} \tag{59b}$$

$$Q_{s_t s_t} = l'_{s_t s_t} + F_{s_t}^T V_{s_{t+1} s_{t+1}} F_{s_t} + V_{s_{t+1}} F_{s_t s_t} \tag{59c}$$

$$Q_{u_t u_t} = l'_{u_t u_t} + F_{u_t}^T V_{s_{t+1} s_{t+1}} F_{u_t} + V_{s_{t+1}} F_{u_t u_t} \tag{59d}$$

$$Q_{u_t s_t} = l'_{u_t s_t} + F_{u_t}^T V_{s_{t+1} s_{t+1}} F_{s_t} + V_{s_{t+1}} F_{u_t s_t} \tag{59e}$$

The last terms in (59c), (59d) and (59e) are ignored in iLQR for reducing calculation burden. Minimizing (58) with respected to δu_t, we obtain

$$\delta u_t^* = -Q_{u_t u_t}^{-1} \left(Q_{u_t} + Q_{u_t s_t} \delta s_t \right) = k_t + K_t \delta s_t \tag{60}$$

where $k_t = -Q_{u_t u_t}^{-1} Q_{u_t}$ and $K_t = -Q_{u_t u_t}^{-1} Q_{u_t s_t}$. Substituting (60) into (58), we have

$$V_{s_t} = Q_{s_t} - Q_{u_t} Q_{u_t u_t}^{-1} Q_{u_t s_t} \tag{61}$$

$$V_{s_t s_t} = Q_{s_t s_t} - Q_{s_t u_t} Q_{u_t u_t}^{-1} Q_{u_t s_t} \tag{62}$$

which can be used to compute k_{t-1} and K_{t-1}. Recursively computing $((k_{k+M-1}, K_{k+M-1}), \ldots, (k_k, K_k))$ constitutes a backward pass. Once the backward pass is completed, a forward pass is used to obtain a new trajectory as

$$\begin{aligned} &\hat{s}_k = s_k \\ &\hat{u}_t = u_t + k_t + K_t \left(\hat{s}_t - s_t \right) \\ &\hat{s}_{t+1} = F\left(\hat{s}_t, \hat{u}_t \right) \end{aligned} \tag{63}$$

By iteratively performing backward pass and forward pass, an optimal $U_k^* = \left[u_k^* \ldots u_{k+M-1}^* \right]^T$ can be obtained, and the first control unit u_k^* is applied to the WMR. The specific algorithm for designing GPMPC controller is summarized as algorithm 1.

Algorithm 1: GPMPC controller design

```
for $k = 1, 2, \cdots$ do
    Observing the state $s_k$;
    if $k \neq 1$ then
        Updating the GP model by minimizing (18);
    end
    Initializing a preliminary control sequence $u_k$;
    Initializing a trajectory $(s_k, u_k, \cdots, s_{k+M})$;
    for $c_f = 1, \cdots, C$ do
        Executing a backward pass according to (59)- (62);
        Executing a forward pass according to (63);
    end
    Computing the $U_k^* = \left[ u_k^* \ldots u_{k+M-1}^* \right]^T$ ;
    Applying the $u_k^*$ to the WMR
end
```

4 Simulation

In this section, the validity of the proposed GPMPC approach will be verified by numerical simulations. The simulation example runs in Python environment. The computer used is configured with AMD-R7 4800 H CPU, 2.90 GHz and 16.0 GB running memory. Specifically, two comparison simulations are performed to verify the efficacy of the GP-enhanced model and the Lorentzian ρ-function, respectively. The velocity model in simulations is set as

$$\begin{aligned} v_{r,k} &= v_{r,k-1} + 0.5 \times \left(v_{r,d,k} - v_{r,k-1} \right) \\ w_{r,k} &= w_{r,k-1} + 0.5 \times \left(w_{r,d,k} - w_{r,k-1} \right) \end{aligned} \tag{64}$$

It means that the real velocity of the robot only steps forward to the desired velocity but cannot arrive at the desired velocity. Other parameters are shown in Table 1.

Only the first two dimensions of the state are constrained since visibility constraints are considered. The input constraints are handled by applying a saturation function to the U_k^* that derived by iLQR method, and thus, b_3 and b_4 are set as 0.

To highlight the importance of model learning, we first compare the simulation results of the proposed GPMPC and the traditional MPC (T-MPC) that only

Table 1 Parameters used in the simulation

Parameters	Values
o_0	$\left[0.5\,[\text{pixels}]\ 0.1\,[\text{pixels}]\ 0\ 0\ 0\right]^T$
o^*	$\left[-0.5\,[\text{pixels}]\ 0.35\,[\text{pixels}]\ \pi/6\,[\text{rad}]\ 0\ 0\right]^T$
Q	$\text{diag}\,(1, 3, 0.5, 0, 0)$
R	$\text{diag}\,(0.01, 0.01)$
T	$0.05\,[s]$
M	20
iLQR iterative times C	2
Σ_ε	$\text{diag}\,(0.001, 0.001, 0.001, 0.001, 0.001)$
(η, λ, α)	$(1, 1, 0.001)$
(b_1, b_2, b_3, b_4)	$(100, 100, 0, 0)$
$o_{\max}$	$\left[0.58\,[\text{pixels}]\ 0.38\,[\text{pixels}]\ \sim\ \sim\ \sim\right]^T$
$o_{\min}$	$\left[-0.58\,[\text{pixels}]\ 0\,[\text{pixels}]\ \sim\ \sim\ \sim\right]^T$
$u_{\max}$	$\left[1\,[\text{m / sec}]\ 0.5\,[\text{rad / sec}]\right]^T$
$u_{\min}$	$\left[-1\,[m/sec]\ -0.5\,[\text{rad / sec}]\right]^T$

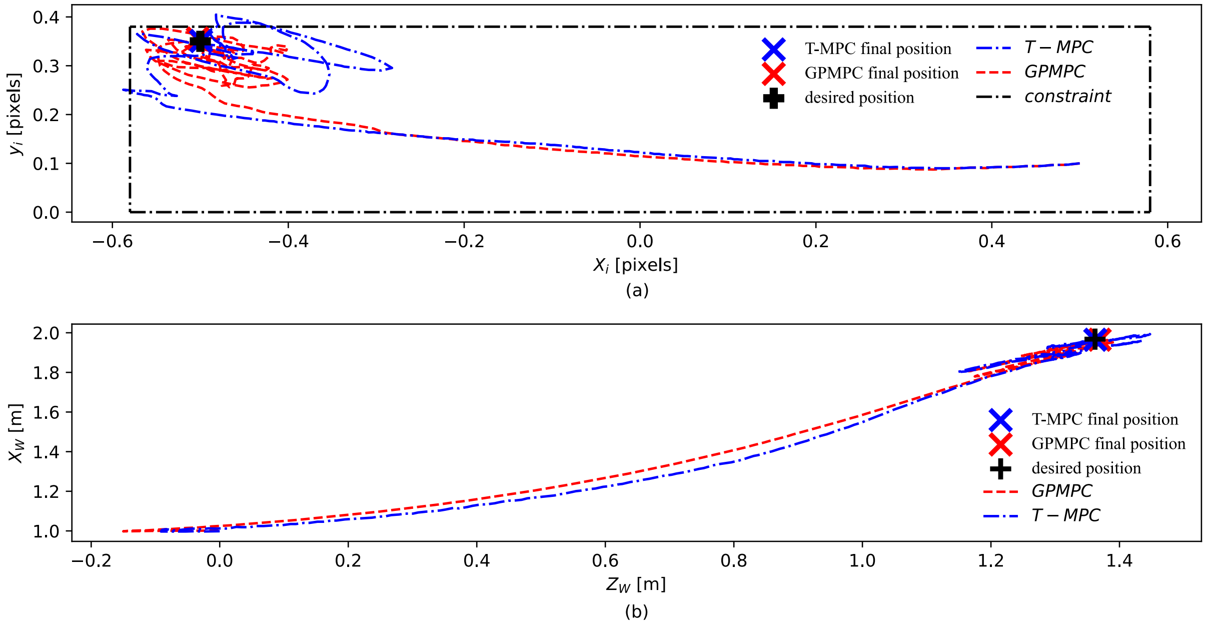


Fig. 4 Comparison results of the GPMPC and T-MPC. **a** Moving trajectories of the feature point in the image frame. **b** Moving trajectory of the camera in the world frame

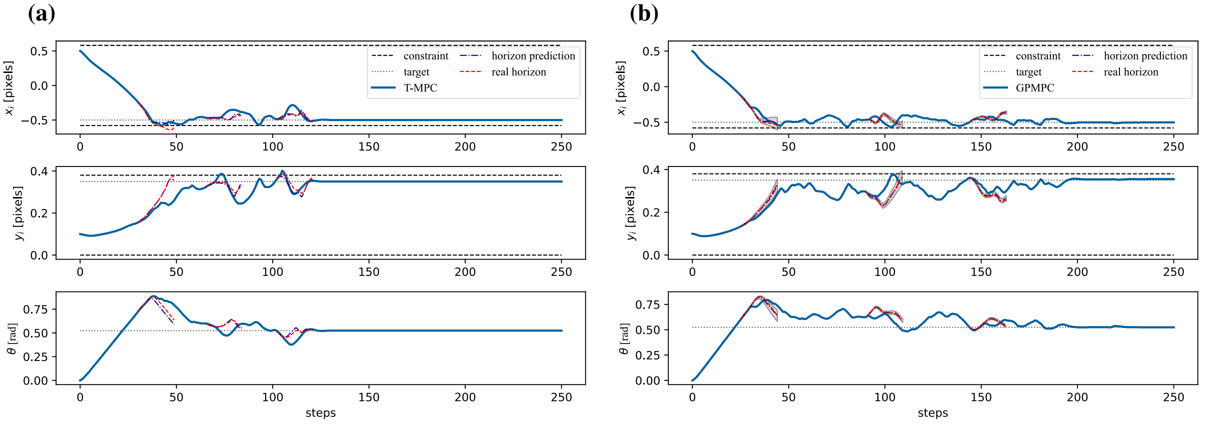


Fig. 5 Computing details of the T-MPC controller and the GPMPC controller. **a** Computing details of the T-MPC controller. **b** Computing details of the GPMPC controller

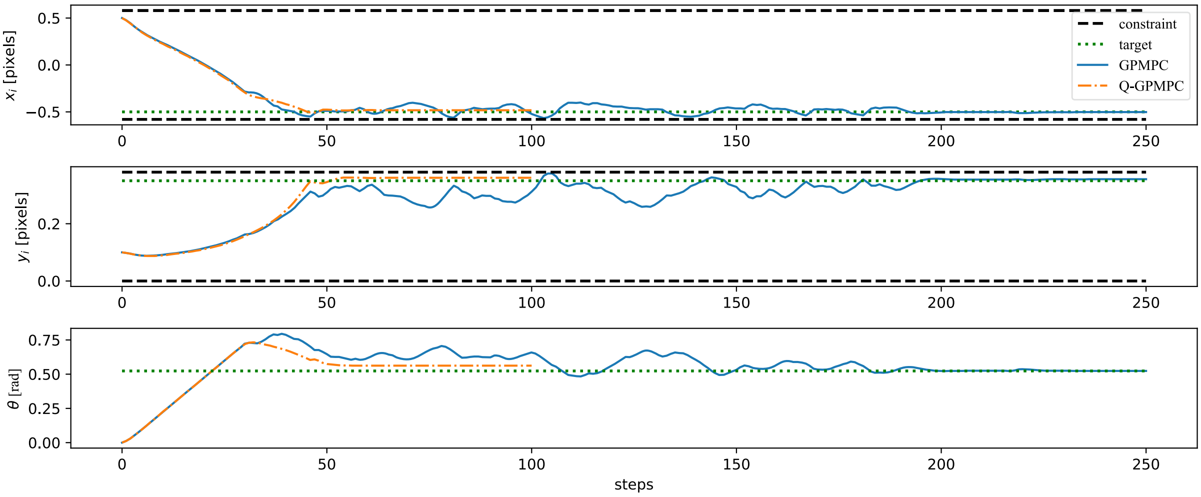


Fig. 6 Comparison results of the GPMPC controller and Q-GPMPC

uses the nominal model (11), which is shown in Fig. 4. Solid circles in Fig. 4a and b represent the end position of the feature point and the camera. The symbol '+' represents the desired position and the dashed box denotes the visibility constraints. It is evident that with the control of GPMPC and the T-MPC, the feature point can all arrive the desired position. However, the FOV of the feature point will be lost when the WMR is controlled by T-MPC, which is reflected in Fig. 4a that the blue trajectory is sometimes outside the dashed box. In real applications, this issue may cause the failure of the visual servoing task.

To further analyze reasons for this issue, we record the computing details of the GPMPC controller and the T-MPC controller, as shown in Fig. 5. The prediction horizon lines (blue dashed lines) represent the horizon state trajectories (HSJs) predicted by the optimal control sequence and current model (the nominal model for T-MPC and the GP-enhanced model for GPMPC), and the real horizon lines (red dashed lines) represent the real horizon state trajectories (RHSJs) predicted by the optimal control sequence and the real model. Figure 5a shows that visibility constraints are sometimes violated, especially in the Y-axis of the image frame under the control of the T-MPC. This is caused by the model error between the nominal model and the real model. At step 102, although HSJs do not violate visibility constraints, the RHSJs do because the

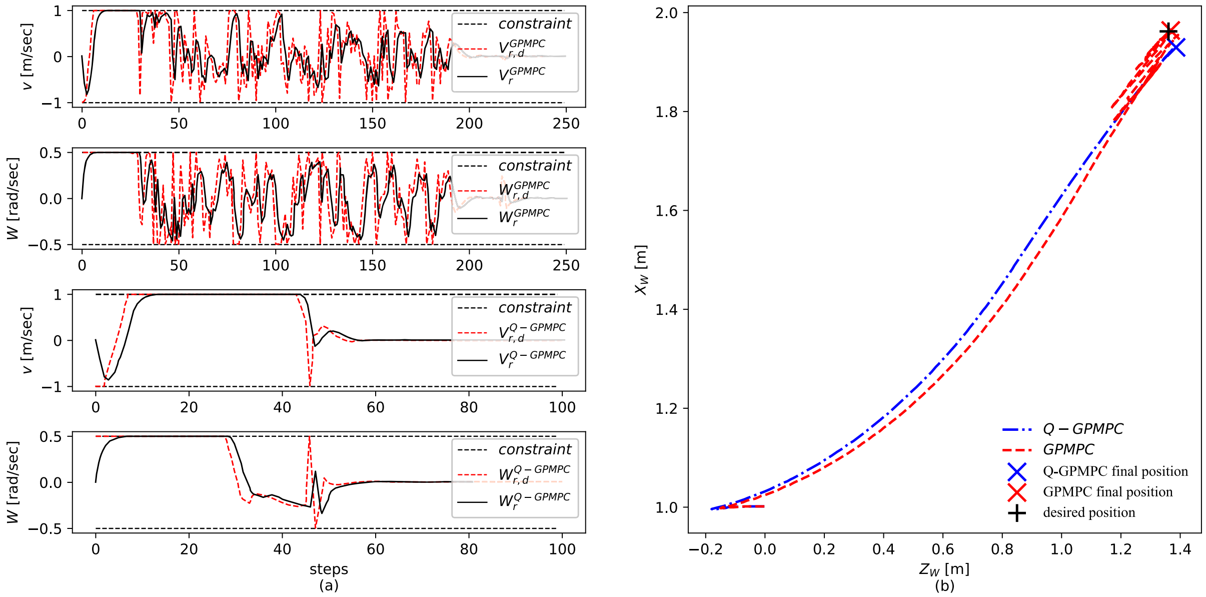
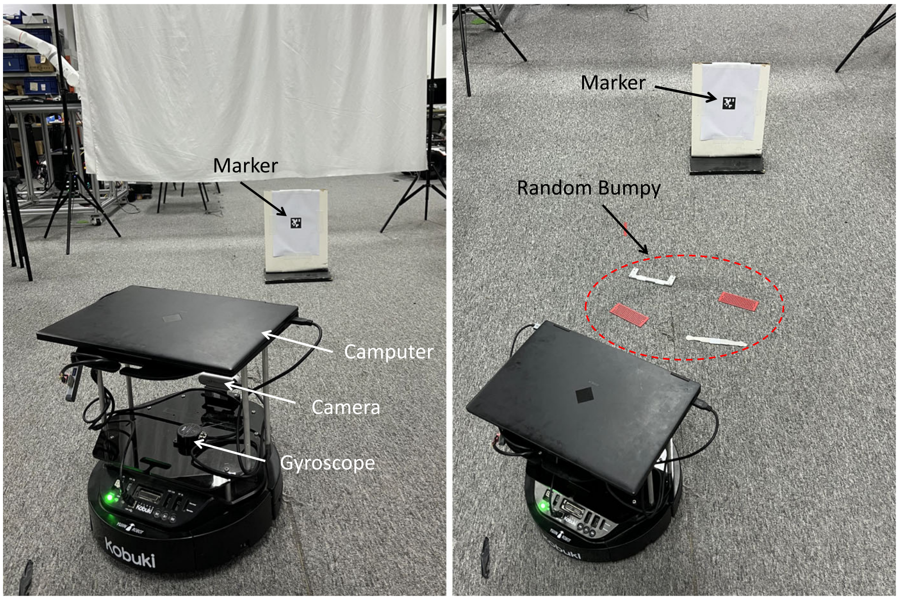


Fig. 7 Detailed control information of the Q-GPMPC and GPMPC

Fig. 8 Environment of the experiment



optimal control lists are derived based on the nominal model. Therefore, with the model error, visibility constraints are not guaranteed to be satisfied. For GPMPC, the additive GP model can effectively capture differences between the nominal model and the real model, and thus, the GP-enhanced model is accurate enough to derive proper optimal control lists. HSJs for GPMPC are stochastic, shown in Fig. 5b, where shadow areas represent the 95% predictive confidence region. It can be seen that RHSJs are close to the mean trajectories, which reflects the fact that the GP-enhanced model resembles the real model. Moreover, the prediction variance that represents the prediction uncertainty gradually decreases with the duration of on-line model learning. Thus, the visibility constraints are satisfied by using the GPMPC method.

In what follows, the effectiveness of the Lorentzian ρ-function is shown by comparing the control results

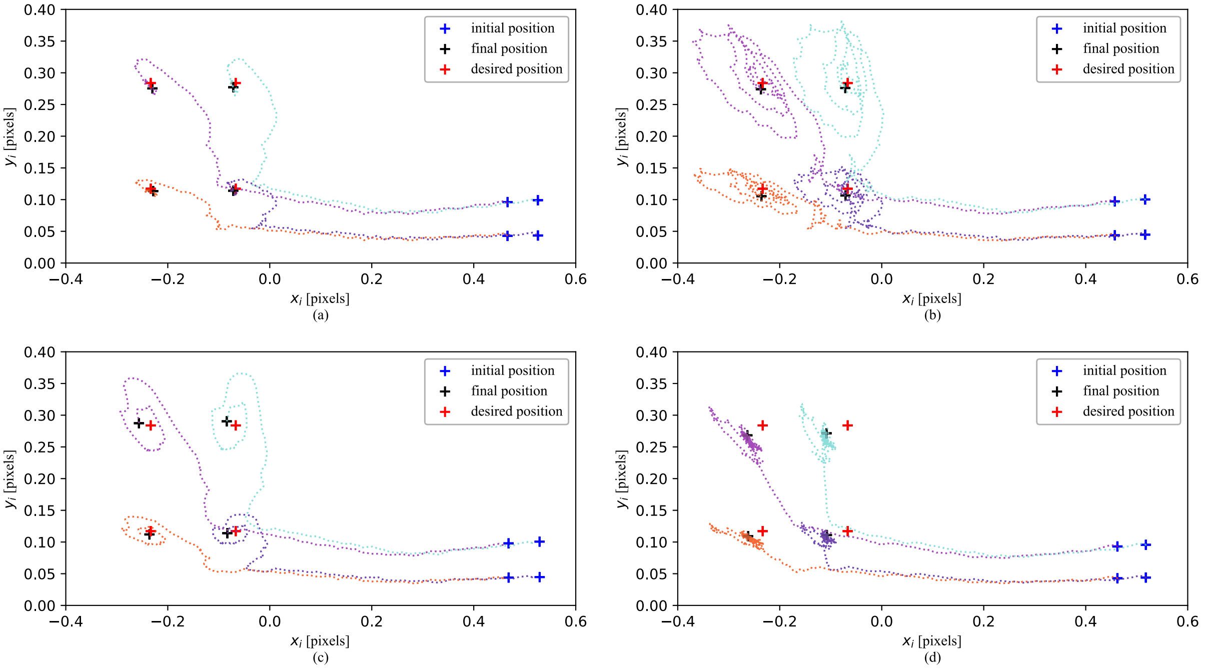


Fig. 9 Motion trajectories of the feature point in the image frame. **a** motion trajectories of GPMPC. **b** Motion trajectories of O-GPMPC. **c** Motion trajectories of T-MPC. **d** Motion trajectories of Q-GPMPC

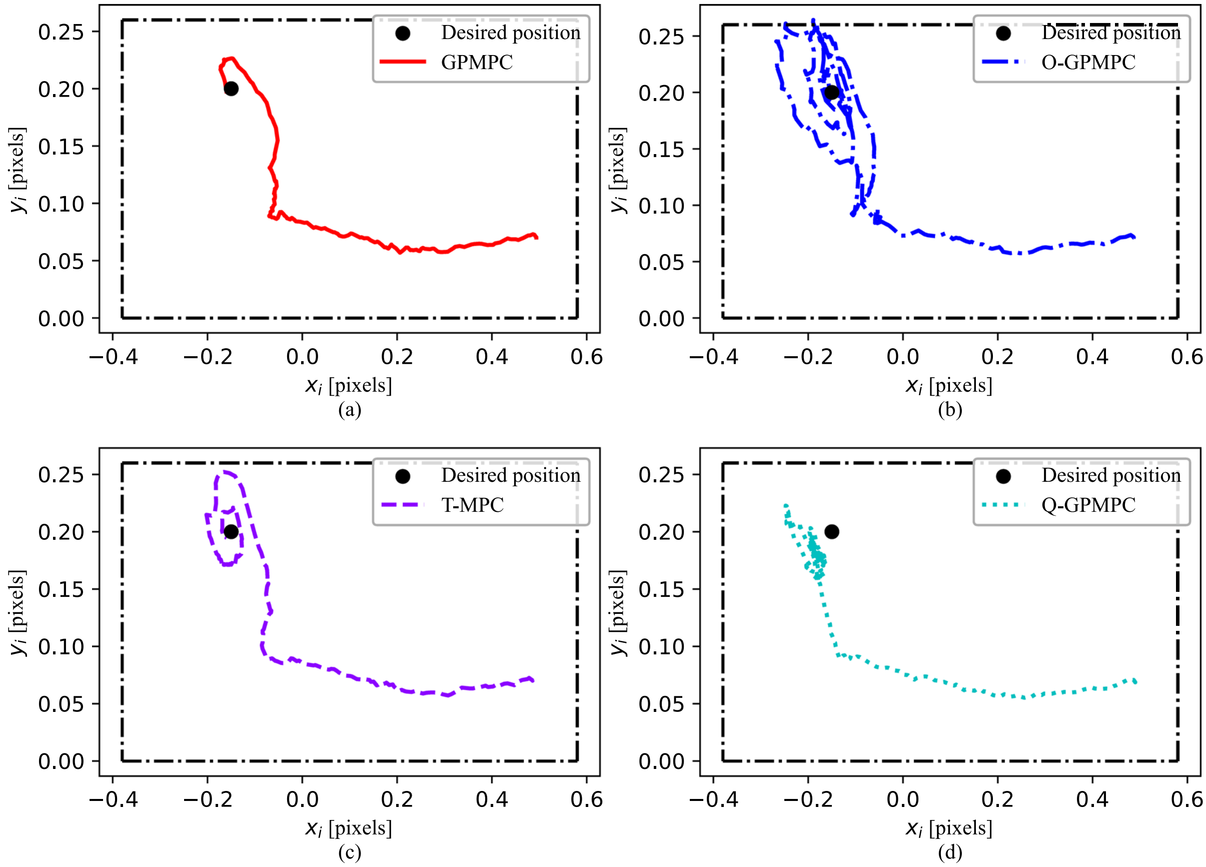
of the GPMPC and the quadratic GPMPC (Q-GPMPC) that uses the quadratic terminal cost function designed as $l_f\left(o_{k+M}\right) = d^2$. Figure 6 illustrates the comparison results with the control of the Q-GPMPC. It can be seen that Q-GPMPC has relatively large static errors when the system is stable in the θ, x_i and y_i directions, which are 0.038 [rad], 0.022 [$pixels$] and 0.01 [$pixels$], respectively. In contrast, the static errors of GPMPC are much minor, which are 0.002 [rad], 0.002 [pixels], and 0.004 [pixels], respectively. This issue is caused by the shape of the quadratic cost function, i.e., the gradient is nearly zero when d is near 0 (when o is near o^*). Thus, the optimizer outputs the near-zero control list when the error is small, leading to the static error. In contrast, the Lorentzian ρ-function has a concave shape as d is close to 0, which encourages precise control.

Figure 7 shows the detailed control information. From Fig. 7a, the control inputs to the Q-GPMPC disappear at step 60 even though the servo error still exists. Figure 7b shows that the WMR finally stops near the desired position. Different with the Q-GPMPC, the control inputs of GPMPC do not disappear when WMR is near the desired position. However, the control inputs become more complex, gradually driving the WMR to the desired position by alternating forward and backward movements. The basic condition for producing these alternating motions is that the value function has obvious differences in the region near the desired position, which is consistent with the shape of the Lorentzian ρ-function.

The average control periods of the T-MPC and GPMPC are 0.046 [sec] and 0.080 [sec], respectively. Thus, as the above analyses, the proposed GPMPC method is effective to address the visual servoing task for the WMR with unknown velocity model and visibility constraints.

5 Experiments

In this section, a corresponding experiment of WMR visual servoing point stabilization is designed to confirm the effectiveness of the GPMPC method in practical applications. The experimental equipment includes a turtlebot2 mobile robot, a Realsense D435 USB camera, a high-precision gyroscope and a computer with an i7-12700 H CPU and RTX-3060 GPU. The camera frame rate of the camera is set to a frame rate



(a)

(b)

(c)

(d)

Fig. 10 Motion trajectories in the image frame. **a** Motion trajectory of GPMPC. **b** Motion trajectory of O-GPMPC. **c** Motion trajectory of T-MPC. **d** Motion trajectory of Q-GPMPC

of 30 (FPS), and the resolution is set 640x480 pixels, with the internal parameters $(fx, fy, c_u, c_v) = (-607.5, -606.2, 325.5, 243.8)$. The gyroscope angle accuracy is 0.1 degrees. The experiment is conducted in the environment shown in Fig. 8. In the experiment, four corner points of the AprilTag marker were selected as visual servoing image features used by the Visual Servoing Platform (ViSP) [43]. A tolerance area is set ($\left|e_{x_i}\right| < 0.01\,[\text{pixels}]$, $\left|e_{y_i}\right| < 0.01\,[\text{pixels}]$, $|e_\theta| < 0.035\,[rad]$), which means that the servo task is assumed to be completed only when all three conditions are met simultaneously. It should be noted that some experimental parameters are different from simulation parameters are set as $o_0 = \left[0.5\,[pixels]\ 0.07\,[pixels]\ -\pi/6\,[rad]\ 0\ 0\right]^T$, $o^* = \left[-0.15\,[pixels]\ 0.2\,[pixels]\ 0\ 0\ 0\right]^T$, $Q = diag\ (0.1, 0.4, 0.05, 0, 0)$, $R = diag\ (0.003, 0.003)$, $o_{\max} = \left[0.58\,[pixels]\ 0.26\,[pixels]\ \sim\sim\sim\right]^T$, $o_{\min} = \left[-0.38\,[pixels]\ 0\,[pixels]\ \sim\sim\sim\right]^T$, $u_{\max} = \left[0.2\,[m/\sec]\ 0.2\,[rad/\sec]\right]^T$, $u_{\min} = \left[-0.2\,[m/\sec]\ -0.2\,[rad/\sec]\right]^T$.

To clearly show the model uncertainty, we have added experiment tests on a road with random bumps to verify the effectiveness of the proposed method under environmental uncertainty. For the sake of clarity, we use O-GPMPC to represent GPMPC experiments conducted in a randomly bumpy environment. The experiment results of GPMPC, O-GPMPC, T-MPC and Q-GPMPC are shown in Figs. 9, 10, 11 and 12. In Fig. 9, the blue symbol '+' represents the initial position, the red symbol '+' represents the desired position, and the black symbol '+' represents the final position. In Fig. 10, the solid circles represent the desired positions of the corresponding trajectories, and the dashed box denotes the visibility constraints. To keep it brief, only the root mean square error of the camera features is

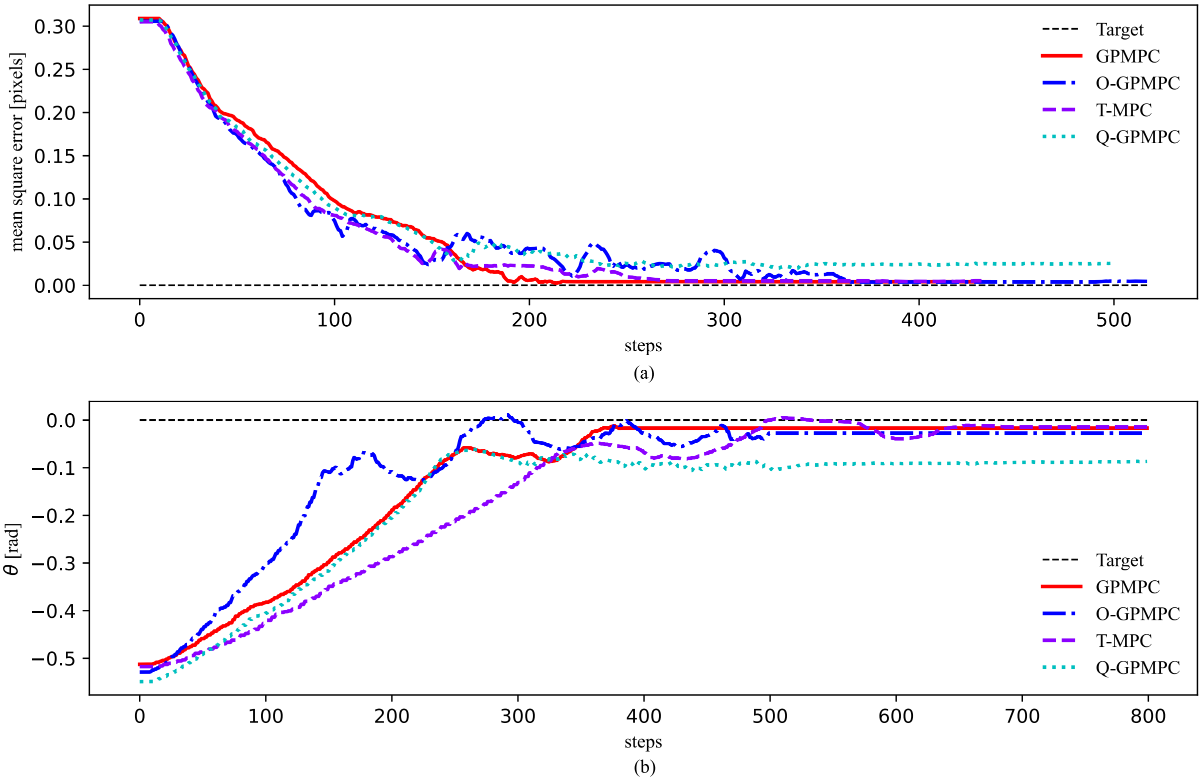


Fig. 11 Comparison results of GPMPC, O-GPMPC, T-MPC and Q-GPMPC

displayed in Fig. 11 instead of showing all errors of the camera features [26]. When the system stabilize and reach the desired position, it can be clearly seen that the motion trajectory of T-MPC in subgraph (c) of Figs. 9, 10 is closer to the constraint boundary than GPMPC in subgraph (a) of Fig. 9, 10. It can be seen that the constraint performance of GPMPC is better than T-MPC. The results of GPMPC are similar to the simulations. Figure 11 shows that Q-GPMPC has a relatively large static error. In contrast, the GPMPC control method with the addition of the Lorentzian ρ-function has a smaller static error. Specifically, the θ and mean square error values of Q-GPMPC are 0.025 $[pixels]$ and -0.086 $[rad]$. On the other hand, GPMPC shows better performance with θ and mean square error values of 0.004 $[pixels]$ and -0.016 $[rad]$. It can be indicated that Q-GPMPC has poorer performance compared to GPMPC. Under the bumpy environment, feature points undergo significant fluctuations in the FOV. The subgraph (b) results of Fig. 9, 10 and Fig. 11 demonstrate that the proposed GPMPC controller can successfully move the WMR to the desired position in a bumpy environment. However, there is oscillation present during the movement. It infers that the proposed method is robust to the uncertainty of the environment. Figure 12 shows the commanded and actual velocities for those control methods. The Lorentzian ρ-function can reduce static errors effectively. However, it can also cause oscillations in WMR when approaching the desired posture. This result is consistent with the simulation results. It can be seen that subgraphs (g) and (h) in Fig. 12 shows persistent small oscillations of the commanded and actual velocity around the desired point. These oscillations are caused by the large static error of Q-GPMPC. Therefore, we only analyzed the first 800 steps of data using the Q-GPMPC method.

The experiment shows that the proposed GPMPC method effectively handles model uncertainty, reduces static errors, and overcomes visibility constraints in WMR visual servoing tasks. Furthermore, it demonstrates robustness to environmental uncertainties.

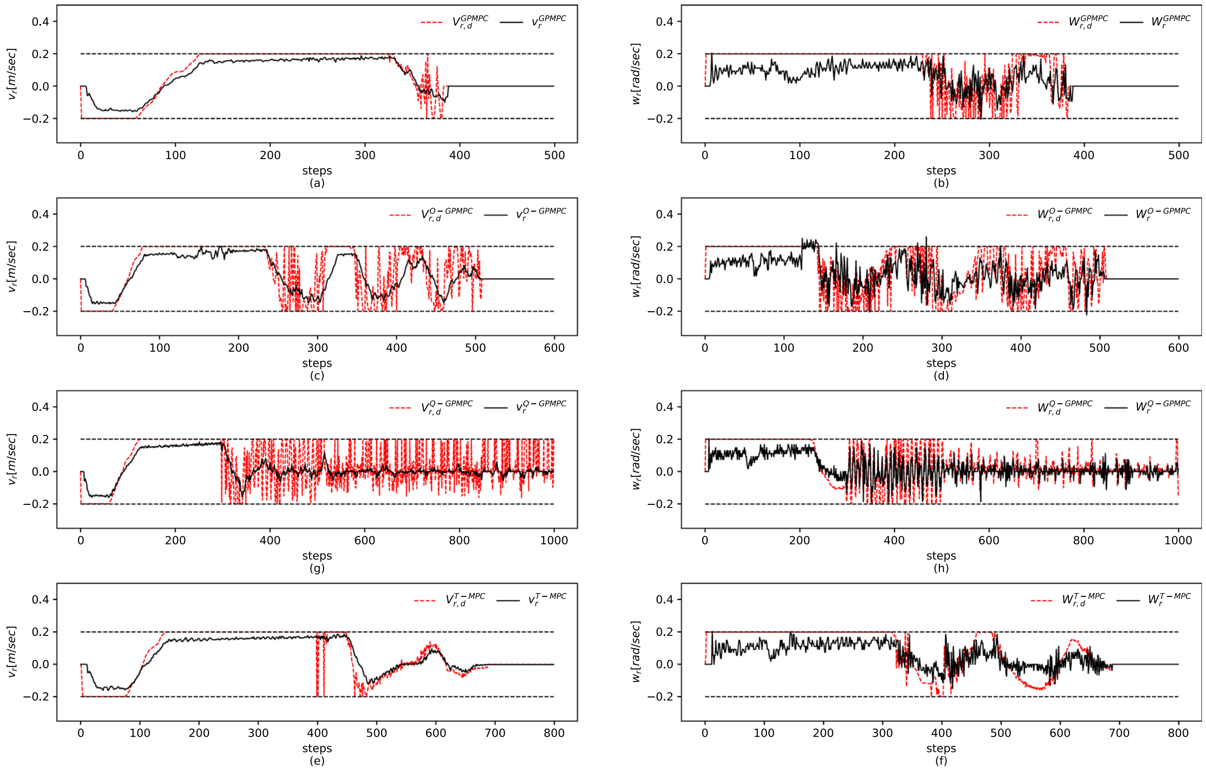


Fig. 12 Command and real velocities in experiments

6 Conclusion and future works

This paper proposes a GPMPC approach to deal with the issue of IBVS in WMR with unknown actuator dynamics. Firstly, a stochastic GP-enhanced model is learned on-line to approximate the real model using the GP method. Then, an SMPC formulation is presented where chance constraints of the state is considered to ensure visibility of the feature point. Moreover, the Taylor expansion techniques is utilized to approximate the uncertainty propagation when performing the multistep forward state prediction. Based on the approximation results, the SMPC problem is transformed as a DMPC problem which is solved by iLQG. Finally, we give two comparison simulations and experiments to verify the validity of the proposed GPMPC method. Notice that iLQG-based MPC requires derivative information and Taylor expansion approximation. In contrast, the MPPI control framework, which does not require the calculation of gradients and second-order approximations, can be easily applied to the real system in real time. In future work, we will use the MPPI strategy to improve the model predictive control algorithm and verify it effectiveness on the actual robot.

Acknowledgements The authors thank the anonymous reviewers for their valuable suggestions to improve the quality of this paper.

Funding This research was funded by the Key R&D Foundation of Zhejiang, China (Grant No. 2023C01224), and the Natural Science Foundation of China (Grant No. 61973275).

Data availability The data generated during the current study are available from the corresponding author on reasonable request.

Declarations

Conflict of interest The authors declare that they have no conflict of interest.

References

1. Liu, A., Zhang, W.A., Yu, L.: Robust predictive tracking control for mobile robots with intermittent measurement and quantization. IEEE Trans. Ind. Electron. **68**(1), 509–518 (2021)

2. Liu, A., Zhang, W.A., Yu, L., Yan, H., Zhang, R.: Formation control of multiple mobile robots incorporating an extended state observer and distributed model predictive approach. IEEE Trans. Syst. Man Cybern. Syst. **50**(11), 4587–4597 (2020)
3. Sun, F., Li, H., Zhu, W., Kurths, J.: Fixed-time formation tracking for multiple nonholonomic wheeled mobile robots based on distributed observer. Nonlin. Dyn. **106**(4), 3331–3349 (2021)
4. Mohamed, I.S., Allibert, G., Martinet, P.: Model predictive path integral control framework for partially observable navigation: A quadrotor case study. In: 2020 16th International Conference on Control, pp. 196–203. Automation, Robotics and Vision (ICARCV), IEEE (2020)
5. Mohamed, I.S., Yin, K., Liu, L.: Autonomous navigation of agvs in unknown cluttered environments: log-mppi control strategy. IEEE Robot. Autom. Lett. **7**(4), 10240–10247 (2022)
6. Touil, D.E., Terki, N., Aouina, A., Ajgou, R.: Intelligent image-based-visual servoing for quadrotor air vehicle. In: 2018 International Conference on Communications and Electrical Engineering (ICCEE), pp 1–7 (2018)
7. Dirik, M., Kocamaz, A.F., Dönmez, E.: Visual servoing based path planning for wheeled mobile robot in obstacle environments. In: 2017 International Artificial Intelligence and Data Processing Symposium (IDAP), pp 1–5 (2017)
8. Ou, M., Sun, H., Zhang, Z., Gu, S.: Fixed-time trajectory tracking control for nonholonomic mobile robot based on visual servoing. Nonlinear Dyn. **108**(1), 251–263 (2022)
9. Xu, F., Wang, H., Wang, J., Au, K.W.S., Chen, W.: Underwater dynamic visual servoing for a soft robot arm with online distortion correction. IEEE/ASME Trans. Mechatron. **24**(3), 979–989 (2019)
10. Jin, Z., Wu, J., Liu, A., Zhang, W.A., Yu, L.: Policy-based deep reinforcement learning for visual servoing control of mobile robots with visibility constraints. IEEE Trans. Ind. Electron. **69**(2), 1898–1908 (2022)
11. Kim, J.k., Kim, D.w., Choi, S.j., Won, S.c.: Image-based visual servoing using sliding mode control. In: 2006 SICE-ICASE International Joint Conference, pp 4996–5001 (2006)
12. Yüksel, T.: Ibvs with fuzzy sliding mode for robot manipulators. In: 2015 International Workshop on Recent Advances in Sliding Modes (RASM), pp 1–6 (2015)
13. Dong, J., Hu, Y., Peng, K.: Robot visual servo control based on fuzzy adaptive pid. In: 2012 International Conference on Systems and Informatics (ICSAI2012), pp 1337–1341 (2012)
14. Xu, F., Wang, H., Liu, Z., Chen, W.: Adaptive visual servoing for an underwater soft robot considering refraction effects. IEEE Trans. Ind. Electron. **67**(12), 10575–10586 (2020)
15. Ghasemi, A., Xie, W.F.: Adaptive image-based visual servoing of 6 dof robots using switch approach*. In: 2018 IEEE International Conference on Information and Automation (ICIA), pp 1210–1215 (2018)
16. Zhang, X., Fang, Y., Zhang, X., Jiang, J., Chen, X.: Dynamic image-based output feedback control for visual servoing of multirotors. IEEE Trans. Ind. Inf. **16**(12), 7624–7636 (2020)
17. Ke, F., Li, Z.: Visual servoing of constrained differential-drive mobile robots using robust tube-based predictive control. In: 2017 13th IEEE Conference on Automation Science and Engineering (CASE), pp 1073–1078 (2017)
18. Ke, F., Li, Z., Yang, C.: Robust tube-based predictive control for visual servoing of constrained differential-drive mobile robots. IEEE Trans. Ind. Electron. **65**(4), 3437–3446 (2018)
19. Sheng, H., Shi, E., Zhang, K.: Image-based visual servoing of a quadrotor with improved visibility using model predictive control. In: 2019 IEEE 28th International Symposium on Industrial Electronics (ISIE), pp 551–556 (2019)
20. Ke, F., Li, Z., Xiao, H., Zhang, X.: Visual servoing of constrained mobile robots based on model predictive control. IEEE Trans. Syst. Man Cybern. Syst. **47**(7), 1428–1438 (2017)
21. Williams, G., Aldrich, A., Theodorou, E.A.: Model predictive path integral control: from theory to parallel computation. J. Guid. Control Dyn. **40**(2), 344–357 (2017)
22. Kappen, H.J.: Path integrals and symmetry breaking for optimal control theory. J. Statist. Mech. Theory Exp. **2005**(11), P11011 (2005)
23. Mohamed, I.S.: Mppi-vs: Sampling-based model predictive control strategy for constrained image-based and position-based visual servoing. arXiv preprint arXiv:2104.04925 (2021)
24. Mohamed, I.S., Allibert, G., Martinet, P.: Sampling-based mpc for constrained vision based control. In: 2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), IEEE, pp 3753–3758 (2021)
25. Fallah, M.M.H., Norouzi-Ghazbi, S., Mehrkish, A., Janabi-Sharifi, F.: Depth-based visual predictive control of tendon-driven continuum robots. In: 2020 IEEE/ASME International Conference on Advanced Intelligent Mechatronics (AIM), pp 488–494 (2020)
26. Somayeh Norouzi-Ghazbi MMFFJS Ali Mehrkish: Constrained visual predictive control of tendon-driven continuum robots. Robot. Auton. Syst. **145**, 103856 (2021)
27. Voos, H.: Nonlinear and neural network-based control of a small four-rotor aerial robot. In: 2007 IEEE/ASME international conference on advanced intelligent mechatronics, pp 1–6 (2007)
28. Dierks, T., Jagannathan, S.: Output feedback control of a quadrotor uav using neural networks. IEEE Trans. Neural Netw. **21**(1), 50–66 (2010)
29. Han, F., Feng, G., Wang, Y., Zhou, F.: Fuzzy modeling and control for a nonlinear quadrotor under network environment. In: The 4th Annual IEEE International Conference on Cyber Technology in Automation, Control and Intelligent, pp 395–400 (2014)
30. Williams, C.K., Rasmussen, C.E.: Gaussian processes for machine learning, vol. 2. MIT press Cambridge, MA (2006)
31. Jin, Z., Wu, J., Liu, A., Zhang, W.A., Yu, L.: Gaussian process-based nonlinear predictive control for visual servoing of constrained mobile robots with unknown dynamics. Robot. Auton. Syst. **136**, 103712 (2021)
32. Hewing, L., Kabzan, J., Zeilinger, M.N.: Cautious model predictive control using gaussian process regression. IEEE Trans. Control Syst. Technol. **28**(6), 2736–2743 (2020)
33. Hashimoto, K., Yoshimura, Y., Ushio, T.: Learning self-triggered controllers with gaussian processes. IEEE Trans. Cybern. **51**(12), 6294–6304 (2021)

34. Kabzan, J., Hewing, L., Liniger, A., Zeilinger, M.N.: Learning-based model predictive control for autonomous racing. IEEE Robot. Autom. Lett. **4**(4), 3363–3370 (2019)
35. Nguyen-Tuong, D., Peters, J.: Using model knowledge for learning inverse dynamics. In: 2010 IEEE International Conference on Robotics and Automation, pp 2677–2682 (2010)
36. Cao, G., Lai, E.M.K., Alam, F.: Gaussian process model predictive control of unmanned quadrotors. In: 2016 2nd International Conference on Control, Automation and Robotics (ICCAR), pp 200–206 (2016)
37. Kim, T., Kim, W., Choi, S., Kim, H.J.: Path tracking for a skid-steer vehicle using model predictive control with on-line sparse gaussian process. IFAC-PapersOnLine **50**(1), 5755–5760 (2017)
38. Zhou, M., Guo, Z.Q., Li, X.: Design of model predictive control for time-varying nonlinear system based on gaussian process regression modeling. In: 2016 IEEE 21st International Conference on Emerging Technologies and Factory Automation (ETFA), pp 1–6 (2016)
39. Yang, X., Maciejowski, J.: Risk-sensitive model predictive control with gaussian process models. IFAC-PapersOnLine **48**(28), 374–379 (2015)
40. Maiworm, M., Limon, D., Manzano, J.M., Findeisen, R.: Stability of gaussian process learning based output feedback model predictive control. IFAC-PapersOnLine **51**(20), 455–461 (2018)
41. Deisenroth, M.P.: Efficient reinforcement learning using Gaussian processes, vol 9. KIT Scientific Publishing (2010)
42. Tassa, Y., Erez, T., Todorov, E.: Synthesis and stabilization of complex behaviors through online trajectory optimization. In: 2012 IEEE/RSJ International Conference on Intelligent Robots and Systems, pp 4906–4913 (2012)
43. Marchand, E., Spindler, F., Chaumette, F.: Visp for visual servoing: a generic software platform with a wide class of robot control skills. IEEE Robot. Autom. Mag. **12**(4), 40–52 (2005)

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.