

Fuzzy Adaptive Model Predictive Control for Image-Based Visual Servoing of Robot Manipulators with Kinematic Constraints

Tianqi Zhu, Jianliang Mao*, Linyan Han, and Chuanlin Zhang

Abstract: This paper presents a novel image-based visual servoing (IBVS) controller for a six-degree-of-freedom (6-DoF) robot manipulator by employing a fuzzy adaptive model predictive control (FAMPC) approach. The control strategy allows the robot to track the desired feature points adaptively and fulfill kinematic constraints appearing in a vision-guided task with different initial Cartesian poses. To this aim, the successive linearization method is firstly employed to transform the nonlinear IBVS model to the linear time-invariant (LTI) one at each sampling instant. The nonlinear optimization problem is therefore degraded into a convex quadratic programming (QP) problem. Subsequently, a fuzzy logic is exploited to tune the weighting coefficients in the cost function on the basis of image pixels changes at each step, endowing the reliable adaptation capabilities to different working environments. Experimental comparison tests performed on a 6-DoF robot manipulator with an eye-in-hand configuration are provided to demonstrate the efficacy of the proposed controller.

Keywords: robot manipulator, image-based visual servoing, fuzzy adaptive model predictive control, successive linearization

1. INTRODUCTION

In recent years, great breakthroughs have been achieved in related fields such as computer vision and deep learning, and meanwhile the accuracy of image recognition along with the speed of recognition has been greatly improved [1]. This allows more and more visual information to be widely applied in robot vision servoing systems, such as automatic drilling [2], cargo transportation [3], autonomous driving [4] and multirobot systems [5]. Traditional visual servoing methods are divided into position-based visual servoing (PBVS), image-based visual servoing and hybrid visual servoing (HVS). Detailed information about their advantages can be referred in [6]. Although PBVS can directly handle tracking tasks in three-dimension (3D) space, it is sensitive to calibration errors of camera as well as the recognition errors of object geometry. Since IBVS constitutes a closed-loop control in the image feature space, there is no need to reconstruct the image in three dimensions.

To improve the dynamic tracking performance of the IBVS, numerous of promising methods have been studied in recent years, such as fuzzy proportional-derivative control [7], sliding mode control (SMC) [8] and sampled-

data control [9], etc. However, visibility and actuator constraints are not explicitly considered in these methods. If visibility constraints are not considered, the image feature points may be out of the field range of the camera during visual servo control [10], which may lead to failure when the robot manipulator works in a certain restricted operating environment. Current common methods to handle visibility are via path planning [11]. However, the path planning method requires environment model and knowledge of the camera calibration parameters [12]. Moreover, planning-based algorithms require more memory and computation time, which poses challenges for real-time tracking. Besides the above issues, the smooth movement of the robot is essential for high quality operation. A sudden change in the Cartesian space velocity will inevitably lead to the jitter of the robot.

It is well acknowledged in the robot and automation society that model predictive control (MPC) provides an integrated solution for controlling systems with interacting variables, complex dynamics and constraints [13]. In [14], the authors highlight the potential advantages of the proposed nonlinear predictive control in terms of image prediction and constraint handling. To deal with the visibility constraint, a robust nonlinear model predictive con-

Manuscript received xxx. This work was supported in part by the National Natural Science Foundation of China under Grant 62203292 and Grant 62173221, and in part by the Fundamental Research Funds for the Central Universities under Grant 2242022k30038.

Tianqi Zhu, Jianliang Mao and Chuanlin Zhang are with the Intelligent Autonomous Systems Laboratory, College of Automation Engineering, Shanghai University of Electric Power, Shanghai 200090, PR China (e-mail: zhutianqi@mail.shiep.edu.cn; jl_mao@shiep.edu.cn; clzhang@shiep.edu.cn).

Linyan Han is with the School of Automation, Key Laboratory of Measurement and Control of Complex Systems of Engineering, Ministry of Education, Southeast University, Nanjing 210096, PR China (e-mail: ly.han@seu.edu.cn).

* Corresponding author.

trol (NMPC) scheme is proposed in [15], and the prediction accuracy is sufficiently guaranteed with a nonlinear optimization manner. However, due to the direct invoking of nonlinear models for prediction, the solution problem is no longer a QP problem, but a nonlinear optimization problem instead, which may place a larger computational burden on the controller and hence, it is less suitable for systems that require fast responsive performance under a high sampling frequency. To reduce the computational burden caused by NMPC, an alternative feasible solution is to convert the nonlinear model into a linear one. In [16], by employing the feedback linearization method, a robust MPC is directly developed based on a linear time-invariant (LTI) model. However, the constrained optimization problem is still nonlinear in this case. Moreover, a novel controller based on an efficient tensor product (TP) model transformation method is put forward in [17]. However, the optimization problem is still relatively complex since multiple LTI models are introduced for the optimization procedure. In addition, the controller weights of the above methods are fixed throughout the dynamic optimization process, which may cause the controller to be too aggressive or conservative when the working environments are switched.

The difficulties posed by the change of working conditions can be solved by introducing adaptive mechanism [18, 19]. In [20], the fuzzy logic is incorporated in dealing with multi-input multi-output (MIMO) robot systems with simultaneous output and velocity constraints. In [21], the fuzzy adaptive parameters are designed to regulate the learning rate of the system under different stages of reinforcement learning, which allows the agent to make more or less use of prior experience according to the unstructured environment. To compensate the unknown nonlinear terms within uncertain dynamics, an adaptive fuzzy logic based disturbance estimator is constructed for the disturbance estimation in [22]. In addition, neural networks based control methods have also achieved positive effects in complicated underactuated systems [23]. However, due to the complexity of setting up the environment and training, the application is also relatively complex, and is not widely used in industry.

Based on the observations mentioned above, aiming to adaptively track the desired feature points as well as fulfilling the kinematic constraints for a vision-guided robot manipulator, a novel fuzzy adaptive model predictive control (FAMPC) is proposed in this work. To be specific, firstly, since the kinematic model of IBVS system is essentially a nonlinear one, successive linearization is employed to transform the model into the LTI one at each sampling moment. Secondly, a convex QP problem is formulated considering the practical kinematic constraints. Finally, in order to improve the adaptability under different working environments, a fuzzy logic is designed to change the weight coefficient in the cost function accord-

ing to the tracking errors of image pixels. By comparing the proposed method with existing related works, the main distinguishable features can be summarized as the following aspects:

- 1) A LTI approximation of IBVS model obtained at each sampling instant is derived to predict the future outputs of the system, which leads to the reduction of the computational burden by avoiding nonlinear programming problems while ensuring the acceptable accuracy of the model.
- 2) The kinematic visibility and actuator saturation of robot manipulators are constrained by solving an online QP problem, which guarantees the smoother movements as well as keeping the target in sight of the camera.
- 3) The adaptive adjustment of the weight matrix using fuzzy logic allows the controller to adapt to changes under different operating conditions, thus avoids the necessity to re-tune the parameters which generally relies on artificial experiences.

The paper is organized as follows. In Section 2, the standard nonlinear IBVS model is discussed. Then, the proposed FAMPC for IBVS is explained in detail in Section 3. In Section 4, experiments on a 6-DoF robot manipulator with eye-in-hand configuration are performed to illustrate the capabilities of the proposed approach. Finally, conclusions are given in Section 5.

2. PROBLEM FORMULATION

2.1. IBVS of Robot Manipulators

Generally, the typical control framework of a vision-guided robot manipulator is usually composed of two feedback control loops, i.e., visual-servoing loop and speed loop [24], as shown in Fig. 1. The outer loop aims to track the target by using visual signals and generates the Cartesian velocity for the desired joint velocity, whereas the inner loop aims to realize the robot's trajectory following despite of the uncertain robot dynamics. In this paper, we assume that the inner speed loop performs well, i.e., the dynamic control of the robot can be viewed as a proportional element with a scale factor of 1. In this work, the joint saturation and the visibility constraints will be taken into account for the visual servoing control of robot manipulators.

2.2. Modeling of IBVS System

The core idea of IBVS relies on the tracking of image feature points rather than 3D pose estimation [25]. To design the IBVS controller, kinematics modeling of the vision system should be firstly carried out.

Assume that there is a 3D position $\mathbf{P} = [x, y, z]^T$ under the camera coordinate system, and the projected coordinate in the image space of \mathbf{P} is denoted by $\mathbf{p} = [u, v]^T$.

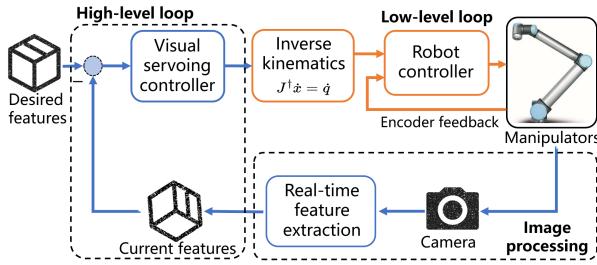


Fig. 1. Structure of vision-guided robot manipulators.

For an eye-in-hand camera, defining the camera velocity as $\mathbf{V}_c = [v_{cx}, v_{cy}, v_{cz}, w_{cx}, w_{cy}, w_{cz}]^T$. The relationship between the camera velocity and the velocity of the image point through image Jacobian matrix \mathbf{L}_p can be expressed as [25]

$$\dot{\mathbf{p}} = \mathbf{L}_p(\mathbf{p}, z)\mathbf{V}_c \quad (1)$$

where

$$\mathbf{L}_p = \begin{bmatrix} -\frac{f}{\rho_u z} & 0 & \bar{u} & \frac{\rho_v \bar{u} \bar{v}}{f} & -\frac{f^2 + \rho_u^2 \bar{u}^2}{\rho_u f} & \frac{\rho_u \bar{v}}{\rho_u} \\ 0 & -\frac{f}{\rho_v z} & \bar{v} & \frac{f^2 + \rho_v^2 \bar{v}^2}{\rho_v f} & -\frac{\rho_v \bar{u} \bar{v}}{f} & -\frac{\rho_u \bar{u}}{\rho_v} \end{bmatrix} \quad (2)$$

in which f is the focal length of the camera, $\bar{u} = u - u_0$ and $\bar{v} = v - v_0$ are pixel points relative to the principle point (u_0, v_0) , ρ_u and ρ_v are the width and height of each pixel, respectively.

In order to be able to indirectly acquire the 6-DoF pose of the object, at least three feature points need to be marked [26]. It is due to that if we only use three feature points, there will exist some configurations that lead to the singularity of the image Jacobian matrix. Furthermore, when the error is eliminated, there will be four global minima, and it is impossible to differentiate them, which is the reason that four feature points is used in most IBVS systems to generate image Jacobian matrix \mathbf{L}_{p4} , whose expression is given by

$$\mathbf{L}_p(\mathbf{p}_m, z) = \begin{bmatrix} \mathbf{L}_p(\mathbf{p}_1, z_1) \\ \mathbf{L}_p(\mathbf{p}_2, z_2) \\ \mathbf{L}_p(\mathbf{p}_3, z_3) \\ \mathbf{L}_p(\mathbf{p}_4, z_4) \end{bmatrix} \in \mathbb{R}^{8 \times 6} \quad (3)$$

where \mathbf{p}_i and z_i are the pixel coordinate and depth of the i -th feature point, respectively, and $\mathbf{p}_m = [\mathbf{p}_1^T, \dots, \mathbf{p}_4^T]^T$, $z = [z_1, \dots, z_4]^T$. In this paper, we consider the case that the depth is measurable at each sampling time by using 3D reconstruction methods.

2.3. Kinematic Constraints

In a vision-guided robotic system, constraints usually include two aspects, i.e., input and output constraints.

Input constraints are generally implemented in the system by constraining the Cartesian velocity \mathbf{V}_c and the

change of the Cartesian velocity $\Delta\mathbf{V}_c$, which is possible to indirectly avoid violent jerks caused by fast speed of the joint motors. The constraints of the input can be represented as

$$\mathbf{V}_{\min} \leq \mathbf{V}_c \leq \mathbf{V}_{\max}, \Delta\mathbf{V}_{\min} \leq \Delta\mathbf{V}_c \leq \Delta\mathbf{V}_{\max}. \quad (4)$$

The output of the system is the position of four feature points in the image plane, which is defined as $\mathbf{p}_m = [\mathbf{p}_1^T, \mathbf{p}_2^T, \mathbf{p}_3^T, \mathbf{p}_4^T]^T \in \mathbb{R}^8$. If they are constrained with the predefined boundary, the tracked object can be well prevented from falling out of the camera view, i.e.,

$$\mathbf{p}_{\min} \leq \mathbf{p}_i \leq \mathbf{p}_{\max}, i = 1, \dots, 4. \quad (5)$$

2.4. Control Objective

The basic control task of the IBVS system is to make the current feature points \mathbf{p}_m always follow the desired ones \mathbf{p}_m^* subject to kinematic constraints. The error signal e represents the difference between the desired feature points and the current ones, i.e.,

$$\lim_{k \rightarrow \infty} e(k) = \lim_{x \rightarrow \infty} (\mathbf{p}_m(k) - \mathbf{p}_m^*(k)) = 0. \quad (6)$$

Generally, the desired feature points \mathbf{p}_m^* artificially given to be knowable, and it could be static or dynamic in the real applications. In this paper, we set \mathbf{p}_m^* as the center of the image.

3. CONTROLLER DESIGN

In this section, a novel fuzzy adaptive model predictive control for IBVS is proposed. The proposed control strategy is capable of guaranteeing input, output constraints and adjusting the cost function adaptively according to different working environments.

3.1. Discrete-Time Model for IBVS

Numerous literature is dedicated to employ MPC to the problem of prediction and optimization in nonlinear model. However, the problem of directly applying MPC with nonlinear prediction model is time consuming and leading to the computational burden. In this sense, if we can get the LTI model of IBVS at each sampling moment, the prediction problem of the model will be simplified.

First, the original nonlinear IBVS model (1) can be overwritten as

$$\dot{\mathbf{p}}_m = \mathbf{F}(\mathbf{p}_m, \mathbf{u}) \quad (7)$$

where $\mathbf{u} = \mathbf{V}_c$ is the control input vector.

To obtain the approximate LTI model at each time instant, successive linearization is adopted to cope with the nonlinear system (7). For the current operating point $(\mathbf{p}_{op}, \mathbf{u}_{op})$, the resulting linearized model can be described as

$$\dot{\mathbf{p}}_m(t) = \dot{\mathbf{p}}_{op} + \mathbf{A}_J(\mathbf{p}_m(t) - \mathbf{p}_{op}) + \mathbf{B}_J(\mathbf{u}(t) - \mathbf{u}_{op}) \quad (8)$$

where $\mathbf{A}_J = \frac{\partial \mathbf{F}}{\partial \mathbf{p}_m}(\mathbf{p}_{op}, \mathbf{u}_{op})$, $\mathbf{B}_J = \frac{\partial \mathbf{F}}{\partial \mathbf{u}}(\mathbf{p}_{op}, \mathbf{u}_{op})$, $\dot{\mathbf{p}}_{op} = \mathbf{F}(\mathbf{p}_{op}, \mathbf{u}_{op})$.

By organizing the constant terms in (8) and summarizing them into one term, it yields

$$\dot{\mathbf{p}}_m(t) = \mathbf{A}_J \mathbf{p}_m(t) + \mathbf{B}_J \mathbf{u}(t) + \mathbf{K} \quad (9)$$

where $\mathbf{K} = \dot{\mathbf{p}}_{op} - \mathbf{A}_J \mathbf{p}_{op} - \mathbf{B}_J \mathbf{u}_{op}$.

Suppose that the sampling time interval is T_s and a ZOH is adopted for the digital processor, then one obtains $\mathbf{u}(t) = \mathbf{u}(kT_s), \forall t \in [kT_s, (k+1)T_s]$, in which $\mathbf{u}(kT_s)$ denotes the control input at the kT_s instant. Similarly, $\mathbf{p}_m(t)$ can be denoted as $\mathbf{p}_m(kT_s)$. Then, according to (9), the discrete state space becomes

$$\begin{cases} \mathbf{p}_m(k+1) = \mathbf{A}_k \mathbf{p}_m(k) + \mathbf{B}_k \mathbf{u}(k) + \mathbf{M}_k \\ \mathbf{y}(k) = \mathbf{p}_m(k) \end{cases} \quad (10)$$

where $\mathbf{A}_k = e^{\mathbf{A}_J T_s}$, $\mathbf{B}_k = \int_0^{T_s} e^{\mathbf{A}_J \tau} d\tau \mathbf{B}_J$, $\mathbf{M}_k = \int_0^{T_s} e^{\mathbf{A}_J \tau} d\tau \mathbf{K}$.

Note that the original nonlinear kinematics model is reduced to a locally linearized one. The system matrices \mathbf{A}_k and \mathbf{B}_k are updated at each time instant.

3.2. Formulation of Output Prediction

In order to derive the future output, we define the control sequence \mathbf{U} and the output sequence \mathbf{Y} as follows

$$\begin{aligned} \mathbf{U} &= [\Delta \mathbf{u}(k)^T, \Delta \mathbf{u}(k+1)^T, \dots, \Delta \mathbf{u}(k+N_c-1)^T]^T, \\ \mathbf{Y} &= [\mathbf{y}(k+1|k)^T, \mathbf{y}(k+2|k)^T, \dots, \mathbf{y}(k+N_p|k)^T]^T \end{aligned} \quad (11)$$

where $\mathbf{U} \in \mathbb{R}^{6N_c}$, $\mathbf{Y} \in \mathbb{R}^{8N_p}$, N_c and N_p represent the control horizon and the prediction horizon, respectively, satisfying $N_p \geq N_c$ [27].

Assume that control inputs remain unchanged outside the control horizon, i.e.,

$$\Delta \mathbf{u}(k+i) = 0, \quad i = N_c, \dots, N_p - 1. \quad (12)$$

Therefore, based on this assumption, the predicted output variables can be expressed by

$$\begin{aligned} \mathbf{y}(k+i|k) &= \mathbf{A}_k^i \mathbf{p}_m(k) + \sum_{j=0}^{i-1} \mathbf{A}_k^{i-j-1} \mathbf{B}_k \mathbf{u}(k-1) \\ &\quad + \sum_{n=0}^{i-1} \left(\sum_{j=n}^{i-1} \mathbf{A}_k^{i-j-1} \mathbf{B}_k \right) \Delta \mathbf{u}(k+n) \\ &\quad + \sum_{j=1}^i \mathbf{A}_k^{i-j} \mathbf{D}_k. \end{aligned} \quad (13)$$

For the sake of simplicity, the output prediction sequence can be rewritten as

$$\mathbf{Y} = \mathbf{F}_k \mathbf{p}_m(k) + \mathbf{G}_k \mathbf{u}(k-1) + \Phi_k \mathbf{U} + \Gamma_k \quad (14)$$

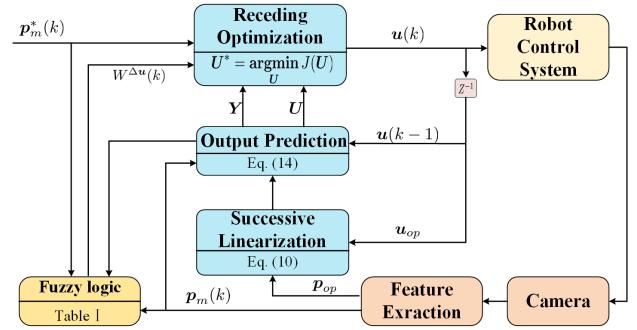


Fig. 2. Implementation diagram of the proposed method

where

$$\begin{aligned} \mathbf{F}_k &= \begin{bmatrix} \mathbf{A}_k \\ \mathbf{A}_k^2 \\ \vdots \\ \mathbf{A}_k^{N_p} \end{bmatrix}, \quad \mathbf{G}_k = \begin{bmatrix} \mathbf{B}_k \\ \mathbf{B}_k + \mathbf{A}_k \mathbf{B}_k \\ \vdots \\ \sum_{j=0}^{N_p-1} \mathbf{A}_k^{N_p-j-1} \mathbf{B}_k \end{bmatrix}, \\ \Phi_k &= \begin{bmatrix} \Theta(1,0) & 0 & \cdots & 0 \\ \Theta(2,0) & \Theta(2,1) & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ \Theta(N_p,0) & \Theta(N_p,1) & \cdots & \Theta(N_p, N_c-1) \end{bmatrix}, \\ \Gamma_k &= \begin{bmatrix} \mathbf{D}_k & 0 & \cdots & 0 \\ \mathbf{A}_k \mathbf{D}_k & \mathbf{D}_k & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{A}_k^{N_p-1} \mathbf{D}_k & \mathbf{A}_k^{N_p-2} \mathbf{D}_k & \cdots & \mathbf{D}_k \end{bmatrix}, \\ \Theta(i,n) &= \sum_{j=n}^{i-1} \mathbf{A}_k^{i-j-1} \mathbf{B}_k. \end{aligned}$$

3.3. Constrained Optimization Design

The core part of MPC is to find the feasible solution that minimizes cost function J with respect to a control sequence \mathbf{U} over horizon N_p . The mathematical formulation of the proposed FAMPC strategy can be written in a convex optimization problem with QP as follows:

$$\mathbf{U}^* = \underset{\mathbf{U}}{\operatorname{argmin}} J(\mathbf{U}) \quad (15)$$

with

$$\begin{aligned} J(\mathbf{U}) &= \sum_{i=1}^{N_p} \| \mathbf{p}_m^*(k+i) - \mathbf{y}(k+i|k) \|^2 \\ &\quad + \sum_{j=0}^{N_c-1} \| W^{\Delta u}(k) \Delta \mathbf{u}(k+j) \|^2 \end{aligned} \quad (16)$$

subject to

$$\begin{cases} \mathbf{Y} = \mathbf{F}_k \mathbf{p}_m(k) + \mathbf{G}_k \mathbf{u}(k-1) + \Phi_k \mathbf{U} + \boldsymbol{\Gamma}_k, \\ \mathbf{y}(k) = \mathbf{p}_m(k), \\ \mathbf{p}_{\min} \leq \mathbf{y}(k+i|k) \leq \mathbf{p}_{\max}, \\ \mathbf{V}_{\min} \leq \mathbf{u}(k+j) \leq \mathbf{V}_{\max}, \\ \Delta \mathbf{V}_{\min} \leq \Delta \mathbf{u}(k+j) \leq \Delta \mathbf{V}_{\max} \end{cases} \quad (17)$$

where $W^{\Delta \mathbf{u}}(k) = \alpha I_{6 \times 6}$ is the time-varying weight matrix on the control inputs.

It is worth noting that the smaller parameter α the more aggressive the controller. In order to achieve the desired control effect under different operating environments, the weight matrix should be updated in real-time according to the tracking effects. To this end, the self-adaption parameter α is adjusted by the fuzzy logic rule in the next subsection.

Since the first action of the optimal solution is applied to the control system, the resulting state feedback control law is designed as $\mathbf{u}(k) = \mathbf{u}(k-1) + \Delta \mathbf{u}(k)^*$, where $\Delta \mathbf{u}(k)^*$ is the first element of \mathbf{U}^* . To be more intuitive, the entire control structure of the proposed approach is illustrated in Fig. 2.

Remark 1: Since the optimal controller takes Cartesian space velocity as input, the singularity avoidance problem for inverse kinematics should be considered in the low-level motion controller design. Here, we will briefly present an alternative design manner.

Inspired by [28], the manipulability ellipsoid is firstly introduced as

$$M(\boldsymbol{\theta}) = \sqrt{\det(J_{ee}(\boldsymbol{\theta})J_{ee}^\top(\boldsymbol{\theta}))} \quad (18)$$

where $J_{ee}(\boldsymbol{\theta})$ denotes the jacobian matrix of the robot manipulator and $\boldsymbol{\theta}$ represents the joint position. Furthermore, by incorporating the derivative of the manipulability ellipsoid $J_m(\boldsymbol{\theta})$ into the optimization problem, the cost function can be defined as

$$f_o(\dot{\boldsymbol{\theta}}) = \frac{1}{2} (\mathbf{J}_{ee}(\boldsymbol{\theta})\dot{\boldsymbol{\theta}} - \mathbf{v})^\top \lambda_\theta \mathbf{I} (\mathbf{J}_{ee}(\boldsymbol{\theta})\dot{\boldsymbol{\theta}} - \mathbf{v}) - \mathbf{J}_m(\boldsymbol{\theta})\dot{\boldsymbol{\theta}} \quad (19)$$

where λ_θ is the gain for velocity-norm minimisation, $\dot{\boldsymbol{\theta}}$ is the joint velocity, and \mathbf{v} denotes the input Cartesian space velocity.

Based on the above declaration, the optimization problem for singularity-free inverse kinematics can be expressed as

$$\min_{\boldsymbol{\theta}} f_o(\dot{\boldsymbol{\theta}}) \quad (20)$$

subject to

$$\dot{\boldsymbol{\theta}}_{\min} \leq \dot{\boldsymbol{\theta}} \leq \dot{\boldsymbol{\theta}}_{\max} \quad (21)$$

where $\dot{\boldsymbol{\theta}}_{\min}$ and $\dot{\boldsymbol{\theta}}_{\max}$ represent the minimum and maximum joint velocities, respectively. So the resulting optimization result will allow the manipulator close to the specified speed while staying away from singularities.

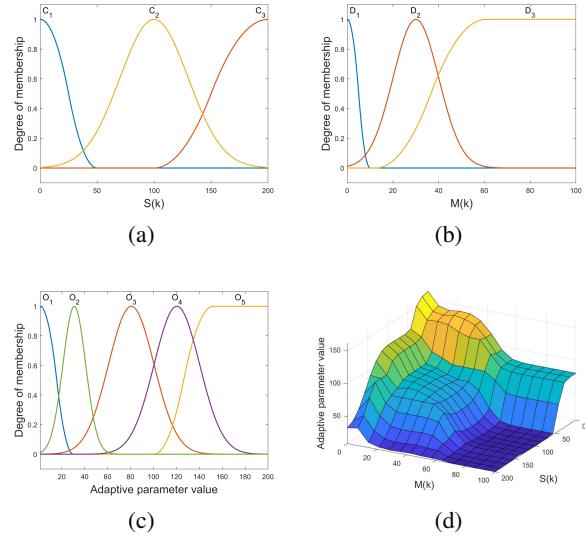


Fig. 3. Affiliation function. (a) The input affiliation function of $S(k)$. (b) The input affiliation function of $M(k)$. (c) The output affiliation function. (d) Fuzzy input and output stereogram.

3.4. Fuzzy Logic Design

Since the frames per second of ordinary cameras generally do not exceed 30 Hz, it often leads to untimely model updates after approaching the object, resulting in excessive system energy and prolonged oscillation near the target. In order to maintain a fast response in the case of tracking moving objects and reducing large deviations, we design the adaptive parameters α by using fuzzy logic to automatically adjust according to the sum of the absolute values of the errors $S(k)$ and the absolute value of change in mass $M(k)$, i.e.,

$$\begin{cases} S(k) = \frac{1}{N_p} \sum_{i=1}^{N_p} \|\mathbf{p}_m^*(k+i) - \mathbf{y}(k+i|k)\|_1, \\ M(k) = \frac{1}{T_s} |S(k) - S(k-1)|. \end{cases} \quad (22)$$

From the cost function, one obtains that the system prefers to have a small control signal when the parameter α becomes larger, and prefers to have a fast response when the parameter α becomes smaller, that is, $S(k)$ and $M(k)$ are inversely proportional to α .

Up to now, we assume that the range of the parameter α is $[0, 200]$, the range of the parameter $S(k)$ is $[0, \max\{S(k)\}]$ and the range of the parameter $M(k)$ is $[0, \max\{M(k)\}]$. Then, the fuzzy set of α can be chosen as O_1, O_2, O_3, O_4, O_5 , the fuzzy set of $S(k)$ can be chosen as C_1, C_2, C_3 , and the fuzzy set of $M(k)$ can be chosen as D_1, D_2, D_3 . The fuzzy inference rules are given as follows: if $S(k) = C_i$ and $M(k) = D_j$, then $\alpha = O_m$.

Table 1. Fuzzy logic table

$S(k)$	C_1	C_2	C_3
$M(k)$			
D_1	O_5	O_4	O_2
D_2	O_4	O_3	O_1
D_3	O_3	O_1	O_1

In this paper, we use Mamdani-Type fuzzy engine [29] and α is obtained by using the center of gravity method. The fuzzy logic rules are shown in Table 1.

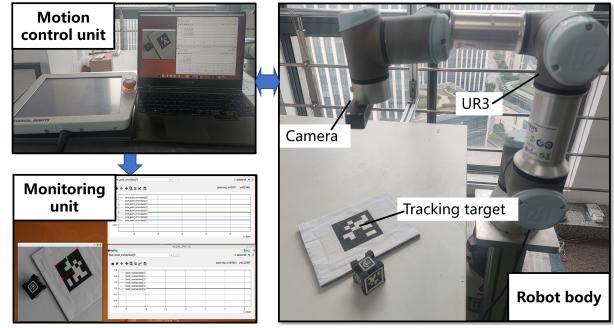
As described above, the input affiliation function and the output affiliation function are designed, as shown in Fig. 3(a)-(c). The relationship between the output and the two inputs are shown in Fig. 3(d). One can observe that when the pixel deviation and the speed of pixel change are relatively large, the adaptive parameters will become smaller at this time, and meanwhile the optimization of the loss function also tends to reduce the error, rather than considering the amount of change in the actuator.

Remark 2: The computational complexities of the proposed control method can be theoretical analyzed. Assuming that the number of state variables is n_x and the number of control variables is n_u . The optimization problem solved by traditional linear MPC is usually quadratic programming, e.g., Riccati based solutions, which has a computational complexity of $O(N_p(n_u + n_x)^3)$ [30]. However, for the general NMPC method, since the nonlinear programming (NLP) needs to be condensed before solving, the computational complexity naturally consists of two parts, i.e., condensing procedure $O(\frac{N_p(N_p+1)(N_p+2)n_u^2n_x}{6})$ and solving process $O(N_p n_u^3)$ [31]. For the proposed control approach, in addition to the computational complexity required by the linear MPC, additional computational cost will be caused by successive linearization and lookup table in fuzzy logic, whose computational complexity is $O(n_x^3 + n_u^2 n_x)$ and $O(n_x + n_u)$, respectively. We will further conduct an experimental verification for the computational burden of different methods in the following section.

4. EXPERIMENTAL RESULTS

In this section, the experimental results of the proposed method are presented. The experimental setup of the IBVS system is shown in Fig. 4. The hardware platform for experiments are configured with a Lenovo laptop with an i7-10750H CPU and 32GB RAM. The robot chosen for experiments are conducted on Universal Robots 3 (UR3) with eye-in-hand configuration. The process of calculating the solution is done by ACADO [32]. The image processing of the visual servoing target is selected VISP [33].

The communication period between the robot and the

**Fig. 4.** Experimental setup**Table 2.** Camera parameters

Parameters	symbol	Values
Sampling time	T_s	0.04 s
Principal point	$(u_0 v_0)$	(320 240)
Camera resolution	-	640×480
Focal length in pixels	f/ρ_u	383.57
Focal length in pixels	f/ρ_v	383.57

controller is about 30 ms. The camera characteristics are listed in Table 2. Moreover, the depth information of feature points is measured by a RGB-D camera.

4.1. Overall Experimental Design

Three experiments are performed with the proposed FAMPC strategy. The first two experiments aim to control the manipulator move from initial points to desired points with the large and small initial deviations, respectively. The third experiment is constructed to track a moving target. The aim of the fourth experiment is to compare the computational burdens of NMPC and FAMPC. We have made a video presentation about the experimental comparisons on the website ¹. For the constraints, the linear velocity of the camera in Cartesian space is limited to 0.3 m/s and the linear acceleration is limited to 0.6 m/s². The angular velocity of the camera in Cartesian space is limited to 0.3 rad/s and the angular acceleration is limited to 0.6 rad/s². The prediction horizon is $N_p = 10$ and the control horizon is $N_c = 1$. The output constraint is defined as

$$\begin{bmatrix} u_{\min} = 40 \\ v_{\min} = 30 \end{bmatrix} \leq p(k) \leq \begin{bmatrix} u_{\max} = 600 \\ v_{\max} = 450 \end{bmatrix}. \quad (23)$$

4.2. Visual Servoing for a Set-Point Tracking

In this case, the task is to implement a set-point tracking for vision-guided robotic system. In order to better show the advantages of the proposed method, two tests

¹<https://www.bilibili.com/video/BV19K411S7Bv>.

Table 3. Initial and desired location of feature points in pixel

Feature point position				
	$(u_1 \ v_1)$	$(u_2 \ v_2)$	$(u_3 \ v_3)$	$(u_4 \ v_4)$
Test 1	Initial (210 370)	(194 292)	(141 313)	(152 387)
	Desired (380 300)	(380 180)	(260 180)	(260 300)
Test 2	Initial (316 338)	(303 220)	(185 236)	(198 352)
	Desired (380 300)	(380 180)	(260 180)	(260 300)

Table 4. Initial poses of the camera in Cartesian space

Initial Cartesian poses					
	x (m)	y (m)	z (m)	ϕ (rad)	θ (rad)
Test 1	0.468	0.070	0.474	-3.016	0.445
Test 2	0.555	0.240	0.430	3.124	0.041
					-1.648

including different level of initial deviations are made for comparison. The initial and desired locations of the features are given in Table 3, and the corresponding initial Cartesian poses of the camera are listed in Table 4. The comparative control methods are the two MPCs with fixed weight matrix, i.e., MPC with the low input weight (MPC-L) and MPC with the high input weight (MPC-H) in the cost function. The test results are given in Figs. 5-12.

The trajectory of features in the image plane can be observed from Fig. 5 and Fig. 9, where the multiplication symbol ("*") represents the initial position of the visual feature points, and the circle symbol ("○") represents the desired feature points. It demonstrates that the proposed method in visual servoing has been successfully completed in both experiments. In a brief analysis, the MPC-L weighting design tends to offset the error as quickly as possible, but may cause system oscillation due to excessive control energy generated by system noise and model misalignment, while the MPC-H weighting design tends to complete the visual servo task smoothly, but with the consequent sacrifice of response speed. The experimental results reflected by the following pictures are also basically consistent with the above analysis. In Figs. 6-8, the response time of MPC-L and FAMPC is much faster than MPC-H. The jitters of FAMPC is significantly smaller than that of MPC-L when it is near the target point, and this phenomenon is especially evident in the plot of angular velocity. In Figs. 10-12, at this operating condition with small starting deviation, the FAMPC accomplishes the servo task smoothly, similar to the MPC-H, and does not produce long periods of oscillations.

In order to compare the advantages of each controller more intuitively, we use the response time (RT), integrated

time and absolute error (ITAE) in the first 10 seconds and (mean squared error) MSE in the 5 seconds after the response time as performance indices in Table 5.

4.3. Visual Servoing for a Dynamic Target

In this case, the task is to track a moving target with displacement and rotation. The object motion consists of small lifting and falling, linear motion and small rotation. The histogram of the pixel average error in the tracking process is shown in Fig. 13(b), and the vertical coordinate represents the total number of corresponding errors occurring during tracking. The results also show that the proposed method also exhibits certain robustness in the dynamic target following scenario.

4.4. Comparison of Computational Burden

Table 5. Performance indexes

Performance				
	Controllers	RT(s)	ITAE	MSE
Test1	MPC-L	3.519	1014.523	6.336
	MPC-H	7.711	1651.133	45.040
	FAMPC	3.564	983.760	4.803
Test2	MPC-L	2.201	352.029	8.674
	MPC-H	2.338	344.336	5.616
	FAMPC	2.230	337.199	5.825

The proposed method is optimised by obtaining a LTI model at each sampling instant by using successive linearization, which is much less computationally demanding than applying the nonlinear model directly. To demonstrate this point, we solved the same problem 5000 times using FAMPC and NMPC, respectively, as shown in Fig. 14. FAMPC average time for a single calculation is 17.32ms, and the calculation time of the NMPC is 48.84ms. As seen in the histogram figure, the calculation time of FAMPC does not fluctuate very much, the maximum calculation time consumed does not exceed 40ms. However, the NMPC's calculation time fluctuates very sharply, reaching a maximum of 120ms.

5. CONCLUSIONS

In this paper, a novel fuzzy adaptive model predictive control for IBVS has been proposed for a vision-guided robot manipulator. The purpose of this method is to improve system adaptation ability while inheriting the key features of the general MPC. Constraint handling and response features were focused on analysis. The performance of the proposed control strategy were evaluated and validated via two experimental tests on a 6-DoF manipulator with eye-in-hand configuration. The reported results

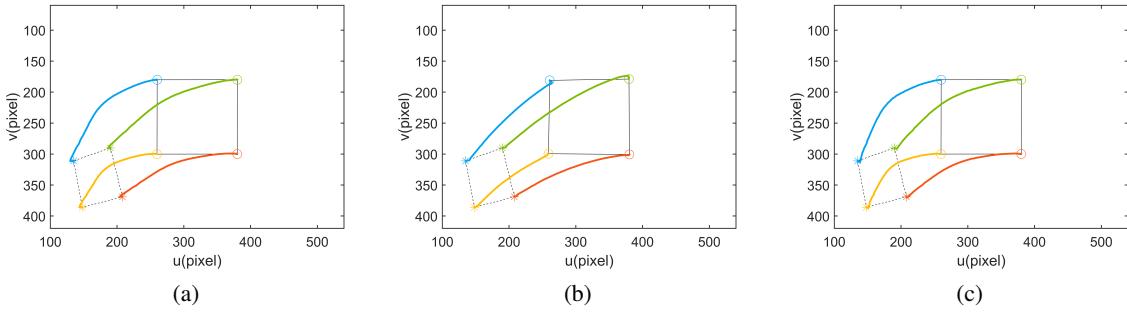


Fig. 5. Trajectory of the reference point tracking in the pixel plane with large initial deviations in Test 1. (a) MPC-L, (b) MPC-H, and (c) FAMPC.

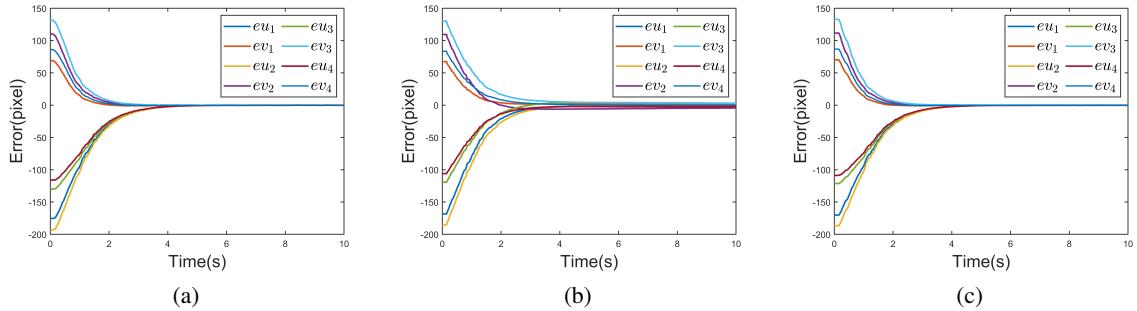


Fig. 6. Tracking errors with large initial deviations in Test 1. (a) MPC-L, (b) MPC-H, and (c) FAMPC.

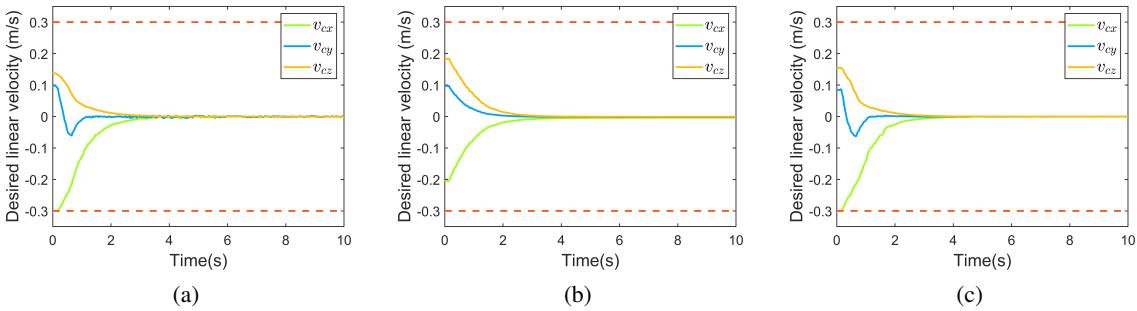


Fig. 7. Desired linear velocity of the camera in Cartesian space with large initial deviations in Test 1. (a) MPC-L, (b) MPC-H, and (c) FAMPC.

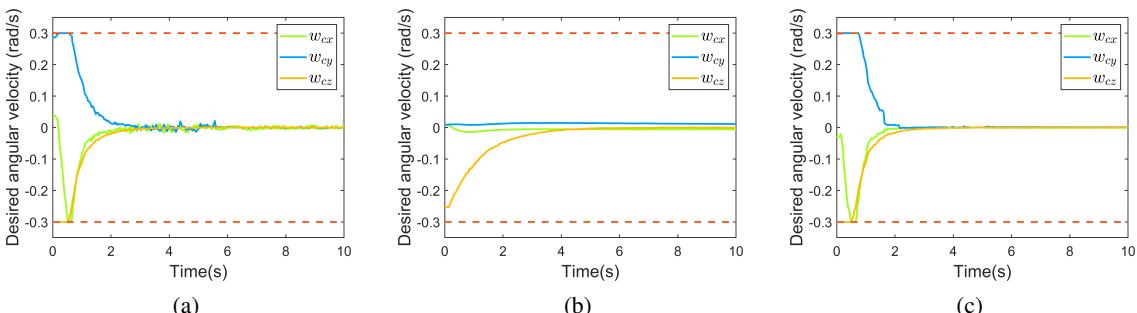


Fig. 8. Desired angular velocity of the camera in Cartesian space with large initial deviations in Test 1. (a) MPC-L, (b) MPC-H, and (c) FAMPC.

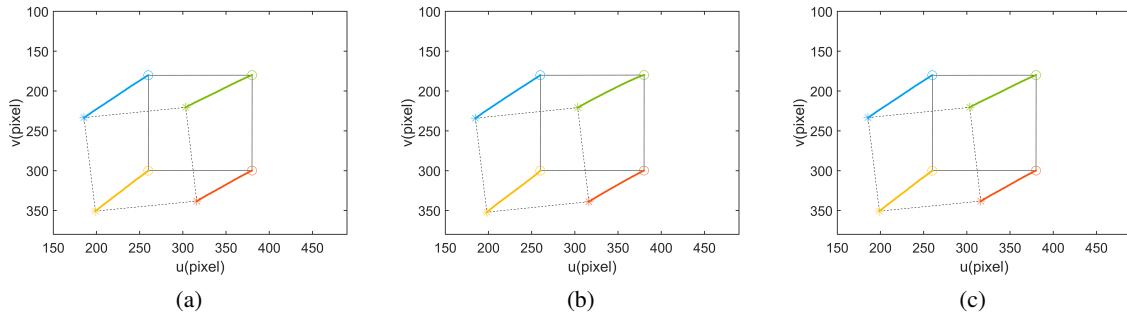


Fig. 9. Trajectory of the reference point tracking in the pixel plane with small initial deviations in Test 2. (a) MPC-L, (b) MPC-H, and (c) FAMPC.

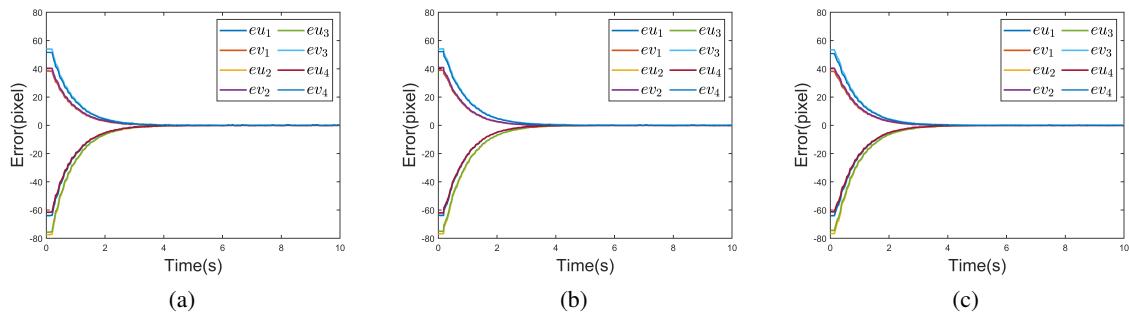


Fig. 10. Tracking errors with small initial deviations in Test 2. (a) MPC-L, (b) MPC-H, and (c) FAMPC.

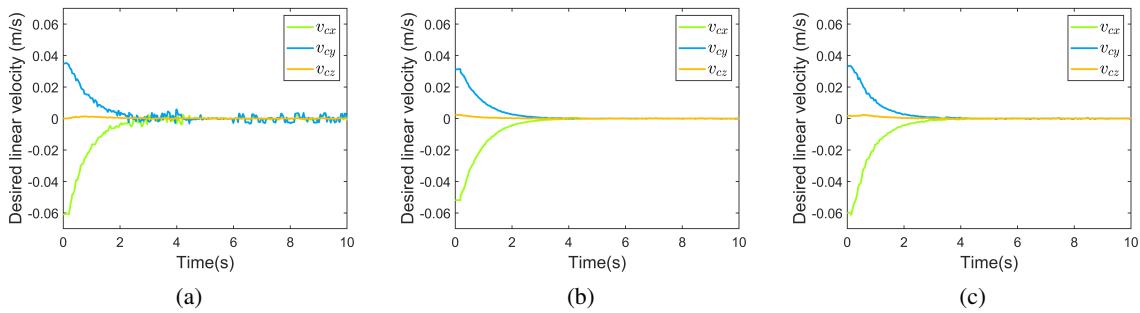


Fig. 11. Desired linear velocity of the camera in Cartesian space with small initial deviations in Test 2. (a) MPC-L, (b) MPC-H, and (c) FAMPC.

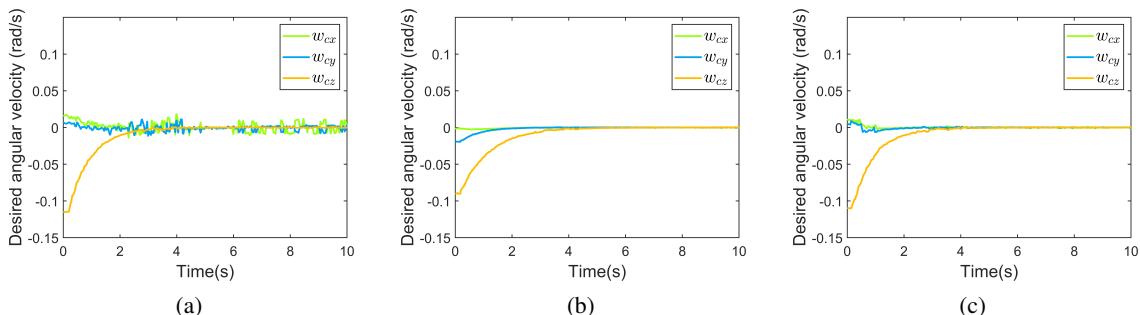


Fig. 12. Desired angular velocity of the camera in Cartesian space with small initial deviations in Test 2. (a) MPC-L, (b) MPC-H, and (c) FAMPC.

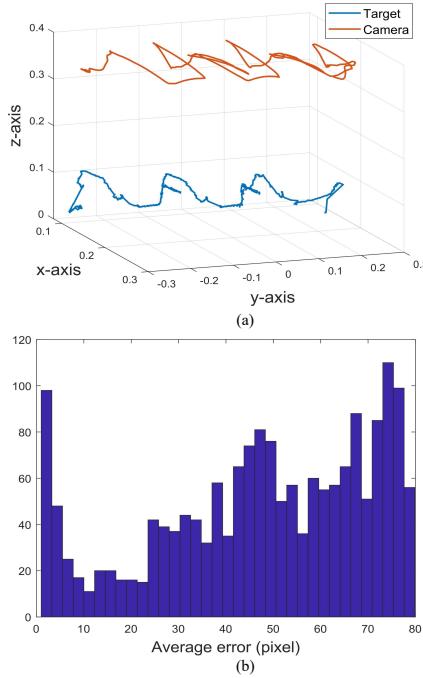


Fig. 13. Results for visual servoing for a dynamic target. (a) Target and camera trajectory, (b) Histogram of average error.

Showed that the proposed method has a smoother trajectory and faster response in comparison with the general MPC methods. In future work, we will focus on visual servoing in combination with neural networks and dynamics control side.

6. DECLARATION OF COMPETING INTEREST

The authors declare that there is no competing financial interest or personal relationship that could have appeared to influence the work reported in this paper.

REFERENCES

- [1] W. Wang, Q. Lai, H. Fu, J. Shen, H. Ling, and R. Yang, “Salient object detection in the deep learning era: An in-depth survey,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 44, no. 6, pp. 3239–3259, 2021.
- [2] H. Liu, W. Zhu, H. Dong, and Y. Ke, “Hybrid visual servoing for rivet-in-hole insertion based on super-twisting sliding mode control,” *International Journal of Control, Automation and Systems*, vol. 18, no. 8, pp. 2145–2156, 2020.
- [3] Y. Quan, M. Lei, W. Chen, and R. Wang, “Agv motion balance and motion regulation under complex conditions,” *International Journal of Control, Automation and Systems*, vol. 20, no. 2, pp. 551–563, 2022.
- [4] R. Reyes and R. Murrieta-Cid, “An approach integrating planning and image-based visual servo control for road
- [5] Z. Li, Y. Yuan, F. Ke, W. He, and C.-Y. Su, “Robust vision-based tube model predictive control of multiple mobile robots for leader–follower formation,” *IEEE Transactions on Industrial Electronics*, vol. 67, no. 4, pp. 3096–3106, 2020.
- [6] F. Janabi-Sharifi, L. Deng, and W. Wilson, “Comparison of basic visual servoing methods,” *IEEE/ASME Transactions on Mechatronics*, vol. 16, no. 5, pp. 967–983, 2010.
- [7] I. Siradjuddin, L. Behera, T. McGinnity, and S. Coleman, “Image-based visual servoing of a 7-dof robot manipulator using an adaptive distributed fuzzy pd controller,” *IEEE/ASME Transactions On Mechatronics*, vol. 19, no. 2, pp. 512–523, 2013.
- [8] B. Ahmadi, W.-F. Xie, and E. Zakeri, “Robust cascade vision/force control of industrial robots utilizing continuous integral sliding-mode control method,” *IEEE/ASME Transactions on Mechatronics*, vol. 27, no. 1, pp. 524–536, 2022.
- [9] J. Yang, X. Liu, J. Sun, and S. Li, “Sampled-data robust visual servoing control for moving target tracking of an inertially stabilized platform with a measurement delay,” *Automatica*, vol. 137, p. 110105, 2022.
- [10] X. Peng, J. Li, B. Li, and J. Wu, “Constrained image-based visual servoing of robot manipulator with third-order sliding-mode observer,” *Machines*, vol. 10, no. 6, p. 465, 2022.

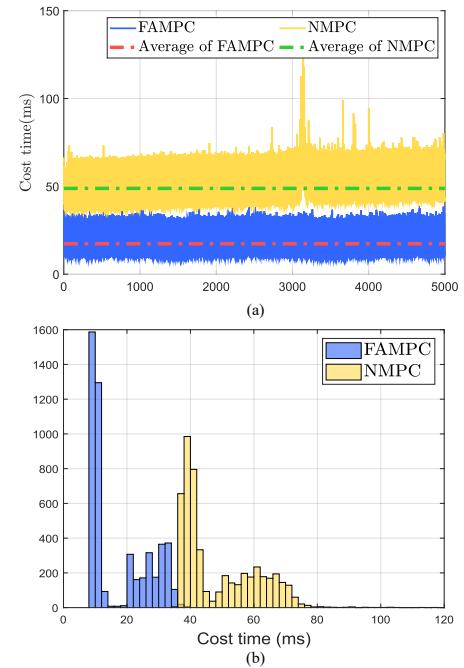


Fig. 14. Time comparison chart between FAMPC and NMPC. (a) Curve chart of cost time,(b) Histogram of cost time.

following and moving obstacles avoidance,” *International Journal of Control*, vol. 93, no. 10, pp. 2442–2456, 2020.

- [1] W. Wang, Q. Lai, H. Fu, J. Shen, H. Ling, and R. Yang, “Salient object detection in the deep learning era: An in-depth survey,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 44, no. 6, pp. 3239–3259, 2021.
- [2] H. Liu, W. Zhu, H. Dong, and Y. Ke, “Hybrid visual servoing for rivet-in-hole insertion based on super-twisting sliding mode control,” *International Journal of Control, Automation and Systems*, vol. 18, no. 8, pp. 2145–2156, 2020.
- [3] Y. Quan, M. Lei, W. Chen, and R. Wang, “Agv motion balance and motion regulation under complex conditions,” *International Journal of Control, Automation and Systems*, vol. 20, no. 2, pp. 551–563, 2022.
- [4] R. Reyes and R. Murrieta-Cid, “An approach integrating planning and image-based visual servo control for road
- [5] Z. Li, Y. Yuan, F. Ke, W. He, and C.-Y. Su, “Robust vision-based tube model predictive control of multiple mobile robots for leader–follower formation,” *IEEE Transactions on Industrial Electronics*, vol. 67, no. 4, pp. 3096–3106, 2020.
- [6] F. Janabi-Sharifi, L. Deng, and W. Wilson, “Comparison of basic visual servoing methods,” *IEEE/ASME Transactions on Mechatronics*, vol. 16, no. 5, pp. 967–983, 2010.
- [7] I. Siradjuddin, L. Behera, T. McGinnity, and S. Coleman, “Image-based visual servoing of a 7-dof robot manipulator using an adaptive distributed fuzzy pd controller,” *IEEE/ASME Transactions On Mechatronics*, vol. 19, no. 2, pp. 512–523, 2013.
- [8] B. Ahmadi, W.-F. Xie, and E. Zakeri, “Robust cascade vision/force control of industrial robots utilizing continuous integral sliding-mode control method,” *IEEE/ASME Transactions on Mechatronics*, vol. 27, no. 1, pp. 524–536, 2022.
- [9] J. Yang, X. Liu, J. Sun, and S. Li, “Sampled-data robust visual servoing control for moving target tracking of an inertially stabilized platform with a measurement delay,” *Automatica*, vol. 137, p. 110105, 2022.
- [10] X. Peng, J. Li, B. Li, and J. Wu, “Constrained image-based visual servoing of robot manipulator with third-order sliding-mode observer,” *Machines*, vol. 10, no. 6, p. 465, 2022.

- [11] Z. Li, C. Li, S. Li, and X. Cao, "A fault-tolerant method for motion planning of industrial redundant manipulator," *IEEE Transactions on Industrial Informatics*, vol. 16, no. 12, pp. 7469–7478, 2020.
- [12] P. D. Triantafyllou, G. A. Rovithakis, and Z. Doulgeri, "Constrained visual servoing under uncertain dynamics," *International Journal of Control*, vol. 92, no. 9, pp. 2099–2111, 2019.
- [13] M. Darby and M. Nikolaou, "Mpc: Current practice and challenges," *Control Engineering Practice*, vol. 20, no. 4, pp. 328–342, 2012.
- [14] G. Allibert, E. Courtial, and F. Chaumette, "Predictive control for constrained image-based visual servoing," *IEEE Transactions on Robotics*, vol. 26, no. 5, pp. 933–939, 2010.
- [15] K. Zhang, Y. Shi, and H. Sheng, "Robust nonlinear model predictive control based visual servoing of quadrotor uavs," *IEEE/ASME Transactions on Mechatronics*, vol. 26, no. 2, pp. 700–708, 2021.
- [16] X. Liu, J. Mao, J. Yang, S. Li, and K. Yang, "Robust predictive visual servoing control for an inertially stabilized platform with uncertain kinematics," *ISA Transactions*, vol. 114, pp. 347–358, 2021.
- [17] T. Wang, W. Xie, G. Liu, and Y. Zhao, "Quasi-min-max model predictive control for image-based visual servoing with tensor product model transformation," *Asian Journal of Control*, vol. 17, no. 2, pp. 402–416, 2015.
- [18] L. Dutta and D. K. Das, "A new adaptive explicit nonlinear model predictive control design for a nonlinear mimo system: An application to twin rotor mimo system," *International Journal of Control, Automation and Systems*, vol. 19, no. 7, pp. 2406–2419, 2021.
- [19] J. de Moura, J. Neto, and P. Rêgo, "A neuro-fuzzy model for online optimal tuning of pid controllers in industrial system applications to the mining sector," *IEEE Transactions on Fuzzy Systems*, vol. 28, no. 8, pp. 1864–1877, 2020.
- [20] T. Yang, N. Sun, and Y. Fang, "Adaptive fuzzy control for a class of mimo underactuated systems with plant uncertainties and actuator deadzones: Design and experiments," *IEEE Transactions on Cybernetics*, vol. 52, no. 8, pp. 8213–8226, 2022.
- [21] H. Shi, M. Xu, and K.-S. Hwang, "A fuzzy adaptive approach to decoupled visual servoing for a wheeled mobile robot," *IEEE Transactions on Fuzzy Systems*, vol. 28, no. 12, pp. 3229–3243, 2019.
- [22] S. Fu, J. Qiu, L. Chen, and M. Chadli, "Adaptive fuzzy observer-based fault estimation for a class of nonlinear stochastic hybrid systems," *IEEE Transactions on Fuzzy Systems*, vol. 30, no. 1, pp. 39–51, 2022.
- [23] T. Yang, N. Sun, and Y. Fang, "Neuroadaptive control for complicated underactuated systems with simultaneous output and velocity constraints exerted on both actuated and unactuated states," *IEEE Transactions on Neural Networks and Learning Systems*, pp. 1–11, 2021.
- [24] A. Hajiloo, M. Keshmiri, W.-F. Xie, and T.-T. Wang, "Robust online model predictive control for a constrained image-based visual servoing," *IEEE Transactions on Industrial Electronics*, vol. 63, no. 4, pp. 2242–2250, 2015.
- [25] P. Corke, *Robotics, vision and control: Fundamental algorithms In MATLAB*. Berlin: Springer, 2017.
- [26] F. Chaumette and S. Hutchinson, "Visual servo control. i. basic approaches," *IEEE Robotics and Automation Magazine*, vol. 13, no. 4, pp. 82–90, 2006.
- [27] M. Morato, J. Normey-Rico, and O. Senam, "Model predictive control design for linear parameter varying systems: A survey," *Annual Reviews in Control*, vol. 49, pp. 64–80, 2020.
- [28] J. Haviland and P. Corke, "Neo: A novel expeditious optimisation algorithm for reactive motion control of manipulators," *IEEE Robotics and Automation Letters*, vol. 6, no. 2, pp. 1043–1050, 2021.
- [29] P. Prieto-Entenza, N. Cazarez-Castro, L. Aguilar, S. Cardenas-Maciel, and J. Lopez-Renteria, "A lyapunov analysis for mamdani type fuzzy-based sliding mode control," *IEEE Transactions on Fuzzy Systems*, vol. 28, no. 8, pp. 1887–1895, 2019.
- [30] M. Diehl, H. J. Ferreau, and N. Haverbeke, "Efficient numerical methods for nonlinear mpc and moving horizon estimation," in *Nonlinear model predictive control*. Springer, 2009, pp. 391–417.
- [31] M. Vukov, A. Domahidi, H. J. Ferreau, M. Morari, and M. Diehl, "Auto-generated algorithms for nonlinear model predictive control on long and on short horizons," in *52nd IEEE Conference on Decision and Control*, 2013, pp. 5113–5118.
- [32] B. Houska, H. J. Ferreau, and M. Diehl, "Acado toolkit—an open-source framework for automatic control and dynamic optimization," *Optimal Control Applications and Methods*, vol. 32, no. 3, pp. 298–312, 2011.
- [33] E. Olson, "Apriltag: A robust and flexible visual fiducial system," in *2011 IEEE International Conference on Robotics and Automation*, 2011, pp. 3400–3407.



Tianqi Zhu received the B.Sc. degree in automation from the Qingdao University of Science and Technology, QingDao, China, in 2020. He is currently working toward the master degree with the School of Automation, Shanghai University of Electric Power, ShangHai, China. His research interests include motion planning, model predictive control and visual servoing.



Jianliang Mao received the B.Sc., M.Sc., and Ph.D. degrees from the School of Automation, Southeast University, Nanjing, China, in 2011, 2014, and 2018, respectively. From 2018 to 2021, he worked at the Research and Development Institute, Estun Automation Co., Ltd. He was selected into “Jiangsu Province entrepreneurship and innovation plan” in 2019. He is currently with the College of Automation Engineering, Shanghai University of Electric Power. His research interests include model predictive control, sliding mode control, vision based interactive control and their applications to electric drives and robot manipulators.



Linyan Han received the M.Sc. degree in Monitoring Technology and Automation Installation from Nanjing University of Aeronautics and Astronautics, Nanjing, China, in 2017, and Ph.D. degree in Control Science and Engineering from the School of Automation, Southeast University, Nanjing, China, in 2022. She is currently a postdoctoral researcher at the School of Mechanical Engineering, University of Leeds. Her interests include force control, nonlinear control theory and their applications to robotic systems.



Chuanlin Zhang received the B.Sc. degree in mathematics and the Ph.D. degree in control theory and control engineering from the School of Automation, Southeast University, Nanjing, China, in 2008 and 2014, respectively. He was a Visiting Scholar with the Energy Research Institute, Nanyang Technological University, Singapore, from 2016 to 2017; a visiting scholar with Advanced Robotics Center, National University of Singapore, from 2017 to 2018. Since 2014, he has been with the College of Automation Engineering, Shanghai University of Electric Power, Shanghai, where he is currently a Professor. His research interests include nonlinear system control theory and applications for robotic systems.