Mathematical Biosciences and Engineering

http://www.aimspress.com/journal/mbe

MBE, 20(6): 10495–10513.
DOI: 10.3934/mbe.2023463
Received: 15 February 2023
Revised: 22 March 2023
Accepted: 27 March 2023
Published: 10 April 2023

Research article

Model predictive control for constrained robot manipulator visual servoing tuned by reinforcement learning

Jiashuai Li[1], Xiuyan Peng[1], Bing Li[1,*], Victor Sreeram[2], Jiawei Wu[1], Ziang Chen[1] and Mingze Li[1]

[1] College of Intelligent Systems Science and Engineering, Harbin Engineering University, Nantong street, Harbin 150001, China

[2] School of Electrical, Electronic, and Computer Engineering, The University of Western Australia, Crawley, WA 6009, Australia

* **Correspondence:** Email: libing265@hrbeu.edu.cn.

Abstract: For constrained image-based visual servoing (IBVS) of robot manipulators, a model predictive control (MPC) strategy tuned by reinforcement learning (RL) is proposed in this study. First, model predictive control is used to transform the image-based visual servo task into a nonlinear optimization problem while taking system constraints into consideration. In the design of the model predictive controller, a depth-independent visual servo model is presented as the predictive model. Next, a suitable model predictive control objective function weight matrix is trained and obtained by a deep-deterministic-policy-gradient-based (DDPG) RL algorithm. Then, the proposed controller gives the sequential joint signals, so that the robot manipulator can respond to the desired state quickly. Finally, appropriate comparative simulation experiments are developed to illustrate the efficacy and stability of the suggested strategy.

Keywords: robot manipulator; image-based visual servoing; model predictive control; reinforcement learning

1. Introduction

Robot visual servo control is an advanced robotics technology widely used in agricultural [1], industrial [2] and medical [3] scenarios. Vision sensors enable agile access to rich environmental information, enabling robots to achieve precise and efficient operations in unstructured environments. The categories of eye-in-hand configuration and eye-to-hand configuration for visual servoing are based on the spatial relationship between the vision sensor and the robot. Distinguished by the information fed back from the vision sensor, visual servoing is classified into position-based visual servoing [4–6], image-based visual servoing [7–11] and hybrid visual servoing [12–14]. Among them, the research and application of IBVS are the most extensive. This study discusses the design of an image-based visual servo controller.

In visual servo tasks, the system constraints are an influential factor that must be considered. If any image feature is out of the field of the camera, it can be said to violate the visibility constraint. An actuator constraint violation occurs when the robot's maximum permissible torque is exceeded by the controller's input torque. All these violations of constraints can result in the failure of the visual servo task. Therefore, meeting the system constraints is a problem that must be solved in visual servo control. Model predictive control can naturally build system constraints into the optimization problem to ensure constraint satisfaction by constructing optimization problems to solve controller actions [15]. Thus, model predictive control methods are often used to execute constrained visual servo control [16–21]. A conjugate visual predictive control strategy was presented in [16] with internal and external constraints in uncertain environments for visual servoing. A predictive control method that simultaneously considers system constraints and uncertainties was proposed in [17], where the prediction model uses a depth-independent Jacobian matrix. However, it was only validated on a planar robot manipulator. A quasi-minimum-maximum MPC method applied to visual servoing was proposed in [18], where the depth values are fixed constants in the Jacobian matrix. A predictive control strategy based on a nonlinear state observer was proposed in [19] to achieve attitude control of the unmanned aerial vehicle (UAV) by using the IBVS method. A nonlinear model predictive control technique [20] based on Gaussian processes was suggested for limiting mobile robots while considering camera visibility limits and robot hardware constraints. An nonlinear model predictive model (NMPC) strategy [21] was proposed to solve the six-degree-of-freedom robotic constrained visual servo but with a heavy computational burden. It is necessary to select the weight matrix of the objective function in the model predictive controller, which reflects the relative importances of different system state variables to the objective function. Generally, the choice of the weight matrix is full of subjectivity and requires many attempts. However, even after a large number of tests, the best control quality cannot be obtained. The weight matrix tuning method of model predictive control has always been a concern. Shridhar and Cooper proposed a multivariable model predictive control adjustment strategy, but it is only applicable to unconstrained systems [22]. Particle swarm and genetic algorithms have also been used to rectify the model predictive controller parameters [23–25].

The discipline of artificial intelligence greatly benefits from the interdisciplinary study of machine learning [26, 27]. Among different approaches, RL, which has gained prominence recently, uses learning procedures to accomplish a certain goal to explain and address the issue that arises in an agent's interaction with the environment [28–31]. Many existing works use RL to assist intelligent control methods, to bring better control in various working scenarios. An adaptive PID control method [28] was proposed for wind energy conversion system control using actor-critic learning to adaptively adjust the controller parameters, and the robustness of the presented control strategy was confirmed by comparative simulation. A deep RL method [29] was used to learn an effective adaptive PID gain adjustment strategy for the control of a marine vessel. A double-Q-learning algorithm [30] was presented to adjust a multilevel PID controller, which was used for simulation research and experiments of a mobile robot. A maximum entropy depth RL method [31] was presented to adjust an AUV controller, to maintain the balance between performance and robustness. The Q-learning algorithm was adopted to adaptively adjust the visual servo gain in [32], which significantly improved the convergence speed and stability compared to the traditional IBVS method with a fixed servo gain. The effectiveness of the proposed method was verified by simulations and experiments. Several works combine model predictive control with RL to carry out research [33–35]. A model predictive control parameter rectification method based on RL was proposed in [33], and the ideal weight matrix was explored in a short time. The NPMC methods tuned by RL were applied to control a UAV in [34, 35].

Although reinforcement learning has been applied to visual servoing, the most commonly used algorithm is still Q-learning. However, the performance of Q-learning in continuous action spaces needs to be improved [36]. DDPG is a reinforcement learning algorithm based on deep neural networks and policy gradient methods which can effectively solve the problem of continuous action spaces by outputting a probability distribution of continuous action values instead of hard-coding discrete actions [37]. Compared with Q-learning, DDPG uses neural networks to approximate the value function and policy, which can converge faster and apply a loss function that combines action value and policy gradient, thus better optimizing the policy and value function. In addition, DDPG also uses techniques such as experience replay buffer and batch training to improve the stability of the algorithm [38].

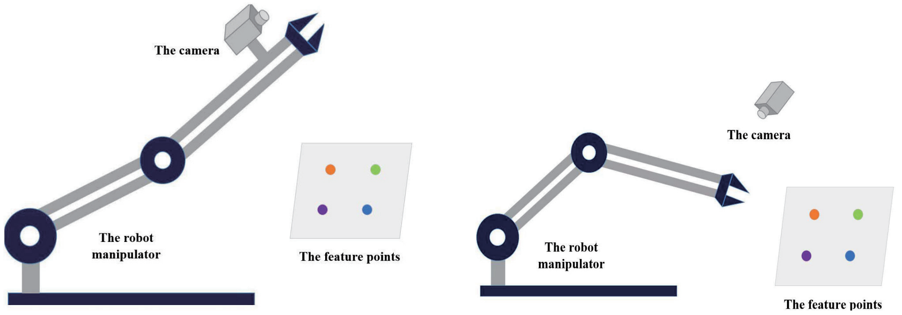
Inspired by the above works, this work presents a model predictive control strategy for robot manipulator visual servoing tuned by DDPG. The main contributions of this paper are as follows:

- To improve the servo efficiency and control accuracy of constrained IBVS systems, this paper proposes an MPC-based IBVS method tuned by DDPG. A depth-independent image interaction matrix is established as the predictive model, and more suitable predictive control weight matrix parameters are trained offline by the DDPG algorithm. Then, the control signal is given through the MPC controller.
- Compared with the traditional MPC-based IBVS method, the proposed method compensates for the disadvantages of the traditional trial-and-error method in tuning the weight matrix parameters and provides better weight matrix parameters, thereby improving visual servo efficiency and steady-state accuracy.
- Compared with the MPC-based IBVS method tuned by Q-learning, the proposed method obtains higher cumulative rewards in the continuous visual servo space and reduces the settling time.

The subsequent structure of this study is as follows. In Section 2, the model of the visual servo system is established to provide the prediction model needed by the model predictive controller. In Section 3, the model predictive controller of IBVS tuned by RL is designed. First, the visual servo model predictive control method is introduced, while later the RL and policy gradient algorithm is introduced, and finally the DDPG-based model prediction control weight matrix strategy is described. In Section 4, the DDPG learning process is introduced, and the effectiveness of the proposed method is verified by simulations. Finally, the research conclusion of this study is summarized in Section 5.

2. System modeling

In robot manipulator visual servo tasks, visibility constraints and robot joint constraints are the system constraint terms that must be considered. The MPC method can solve the system input-output constraint problem, so this study is based on the MPC method to solve the constrained visual servo



(a) eye-in-hand configuration visual servoing (b) eye-to-hand configuration visual servoing

Figure 1. Two configurations of visual servoing.

problem. The predictive model is a very important part of MPC, which can predict the future values of the process output according to the input of the designed controller and the past status of the system. In this study, the association between the rates of variation of image coordinates of feature points and the joint velocity is described by building a depth-independent Jacobian matrix. Figure 1 shows schematic diagrams of the robot manipulator visual servo eye-in-hand configuration and eye-to-hand configuration, respectively. In the two configurations, the image coordinates of feature points can be uniformly represented as

$$s = \begin{pmatrix} u \\ v \end{pmatrix} = \frac{1}{Z_i} \begin{pmatrix} c_1^T \\ c_2^T \end{pmatrix} P \begin{pmatrix} \alpha \\ 1 \end{pmatrix} \tag{2.1}$$

where $s = (u, v)^T$ denotes the 2-D coordinates of the feature point in the image framework, and u and v represent the pixel coordinates projected on the u- and v-axes, respectively. Z_i denotes the depth of the feature point concerning the camera framework. $C \in \Re^{3\times 4}$ denotes the unknown perspective projection matrix, and c_1^T, c_2^T, and c_3^T are the first, second and third rows of the matrix C, respectively. α denotes the 3-D coordinates of the feature point in the Cartesian coordinate system. P denotes the homogeneous transformation matrix, which is up to the forward kinematics of the robot manipulator. By differentiating both sides of the above equation simultaneously with respect to time, it can be obtained that

$$\begin{aligned} \dot{s} &= \frac{1}{Z_i^2} \left(\begin{pmatrix} c_1^T \\ c_2^T \end{pmatrix} P \begin{pmatrix} \dot{\alpha} \\ 1 \end{pmatrix} \dot{q} Z_i - \dot{Z}_i \begin{pmatrix} c_1^T \\ c_2^T \end{pmatrix} P \begin{pmatrix} \alpha \\ 1 \end{pmatrix} \right) \\ &= \frac{1}{Z_i} \begin{pmatrix} c_1^T \\ c_2^T \end{pmatrix} P \begin{pmatrix} \dot{\alpha} \\ 1 \end{pmatrix} \dot{q} - \frac{1}{Z_i^2} \dot{Z}_i \begin{pmatrix} c_1^T \\ c_2^T \end{pmatrix} P \begin{pmatrix} \alpha \\ 1 \end{pmatrix} \end{aligned} \tag{2.2}$$

The depth Z_i can be linearly expressed as

$$Z_i = {c_3}^T P \begin{pmatrix} \alpha \\ 1 \end{pmatrix}. \tag{2.3}$$

Taking the derivative of both sides of Eq (2.3), it can be obtained that

$$\dot{Z}_i = c_3^T P \begin{pmatrix} \dot{\alpha} \\ 1 \end{pmatrix} \dot{q}. \tag{2.4}$$

Substituting Eq (2.4) into Eq (2.2), the time-related association between the joint velocity and the variation of the feature points can be given as

$$\begin{aligned}\dot{s} &= \frac{1}{Z_i}\begin{pmatrix} c_1^T \\ c_2^T \end{pmatrix} P \begin{pmatrix} \dot{\alpha} \\ 1 \end{pmatrix} \dot{q} - \frac{1}{Z_i} s c_3^T P \begin{pmatrix} \dot{\alpha} \\ 1 \end{pmatrix} \dot{q} \\ &= \frac{1}{Z_i}\begin{pmatrix} c_1^T - u c_3^T \\ c_2^T - v c_3^T \end{pmatrix} P \begin{pmatrix} \dot{\alpha} \\ 1 \end{pmatrix} \dot{q} \\ &= \frac{1}{Z_i} \Omega \dot{q}\end{aligned} \tag{2.5}$$

where q denotes the joint angle of the robot manipulator, $\dot{q}$ denotes the joint velocity of the robot manipulator, and Ω denotes the image Jacobian matrix, which is independent of depth. It can be given as

$$\Omega = \begin{pmatrix} c_1^T - u c_3^T \\ c_2^T - v c_3^T \end{pmatrix} P \begin{pmatrix} \dot{\alpha} \\ 1 \end{pmatrix} \tag{2.6}$$

In the formulas above, the depth-independent image Jacobian matrix Ω and the depth of feature point Z_i are nonlinear. However, they can be represented linearly with regressor matrices and unknown parameter vectors by the following property.

Property 1. *For any vector χ, the products $\Omega\chi$ and $Z_i\chi$ can be parameterized in a linear form as*

$$\Omega\chi = A(\chi, q, s)\theta, \tag{2.7}$$

$$Z_i\chi = B(\chi, s)\theta, \tag{2.8}$$

where $A(\chi, q, s)$ and $B(\chi, s)$ are the regressor matrices. θ is the corresponding parameter vector determined by the products of the camera parameters and the robot kinematic parameters.

3. MPC controller design tuned by RL

A constrained control system is usually a kind of system that has constraints on control inputs, outputs and system states. In practical application scenarios, robot visual servo systems will be constrained by image visibility, physical constraints of joints, etc. Therefore, a robot visual servo system is a typical constraint control system. MPC is widely used in the control of constrained systems. In this study, the model predictive controller samples the image coordinates of feature points within a given sampling period and transforms the visual servo problem into a constrained optimization problem to generate the most suitable joint torque signal and minimize the incremental change in joint torque while minimizing the deviation of the predicted image. The purpose of visual servo control is to make the control process as stable as possible while completing visual servo control. The objective function in the optimization problem includes weight matrices, and an appropriate weight matrix will effectively improve the control performance of visual servoing. Previous weight matrices were obtained by researchers through repeated experiments, and it was challenging to achieve the best control effect. In this study, the DDPG-based RL strategy is adopted to modify the appropriate weight matrix of the objective function of MPC to optimize the visual servo performance of MPC-based IBVS. The design of the visual servo model predictive control is

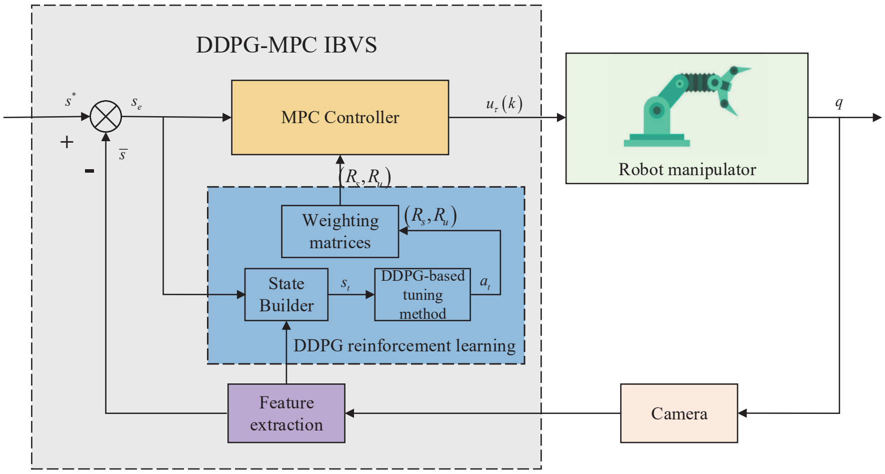


Figure 2. The control scheme of the proposed DDPG-MPC IBVS algorithm.

introduced in the last section, the RL and policy gradient algorithm is introduced in Section 2, and the reinforcement-learning-based objective function weight matrix rectification method is introduced in Section 3. The control scheme of the presented DDPG-based MPC (DDPG-MPC) IBVS algorithm is given in Figure 2.

3.1. Model predictive control for visual servoing

To develop the MPC-based IBVS controller, the discrete-time system model is obtained according to Eq (2.5).

$$s(k+1) = s(k) + T_d \frac{1}{Z_i} \Omega u_\tau(k) \tag{3.1}$$

When the control sequence obtained from the prediction model is applied to the IBVS system, the predicted system output for the next N_p time steps is

$$\begin{aligned}
s(k+1 \mid k) =& s(k) + T_d \frac{1}{Z_i(k)} \Omega(k) u_\tau(k) \\
s(k+2 \mid k) =& s(k) + T_d \frac{1}{Z_i(k)} \Omega(k) u_\tau(k) + T_d \frac{1}{Z_i(k+1)} \Omega(k+1) u_\tau(k+1) \\
&\vdots \\
s\left(k+N_p \mid k\right) =& s(k) + T_d \frac{1}{Z_i(k)} \Omega(k) u_\tau(k) + \cdots + T_d \frac{1}{Z_i(k+i-1)} \Omega(k+i-1) u_\tau(k+N_c-1)
\end{aligned} \tag{3.2}$$

where N_p is predictive time domain, N_c is control time domain, and T_d is the sampling period. The following constrained optimization problem can be solved to calculate the optimal joint torque input.

$$\Pi : \min_{\Delta U_\tau(k)} (s_e(k), \Delta U_\tau(k)) \tag{3.3}$$

subject to

$$s_e(k+i) = \bar{s}(k+i) - s^*(k+i) \tag{3.4}$$

$$\begin{aligned}\Delta U_\tau(k) = \Big[&\Delta u_\tau(k)^T, \Delta u_\tau(k+1)^T \cdots \\ &\Delta u_\tau(k+N_p-2)^T, \Delta u_\tau(k+N_p-1)^T\Big]\end{aligned} \tag{3.5}$$

$$\Delta u_\tau(k) = u_\tau(k) - u_\tau(k-1) \tag{3.6}$$

where s^* denotes the desired image coordinates of feature points, $\bar{s}$ denotes the predicted image coordinates of feature points, s_e denotes the image deviation within the prediction period, $\Delta u_\tau(k)$ denotes the changing values of the control input, and $\Delta U_\tau(k)$ denotes the optimal sequence of the changing value of the control input within the prediction period. To minimize the predicted image coordinate deviation and the minimum control input variation, the quadratic cost function can be described as

$$G\left(s_e(k), \Delta U_\tau(k)\right) = \sum_{i=1}^{N_p} \|\bar{s}(k+i) - s^*(k+i)\|^2_{R_s} + \sum_{i=1}^{N_c} \|\Delta u_\tau(k+i-1)\|^2_{R_u} \tag{3.7}$$

where N_p denotes the prediction horizon, N_c denotes the control horizon, and $N_c \leq N_p$ normally. $R_s \geq 0$ and $R_u \geq 0$ denote the weight matrices of the image coordinate deviation and the sequence of changing values of the control input, respectively. The constraints in the limited visual servoing system can be presented by the following formulas:

$$s^{\min} \leq s(k) \leq s^{\max} \tag{3.8}$$

$$U_\tau{}^{\min} \leq U_\tau(k) \leq U_\tau{}^{\max} \tag{3.9}$$

$$q^{\min} \leq q(k) \leq q^{\max} \tag{3.10}$$

$$\dot{q}^{\min} \leq \dot{q}(k) \leq \dot{q}^{\max} \tag{3.11}$$

Equation (3.8) represents the visibility constraints, Eq (3.9) represents the torque constraints, Eq (3.10) represents the joint angle constraints, and Eq (3.11) represents the joint velocity constraints.

The weight matrices R_s and R_u in the minimization function affect the control effect of the model predictive control. The matrix R_s is used to describe the weights of the control deviations. When the value of R_s is too large, the image coordinates of feature points will converge to the target image coordinates during the control process. However, it ignores the control input variation, resulting in the appearance of an input jitter, which reduces the response quality of the control process. The matrix R_u is used to describe the weights of the change value of control inputs. When the value of the weight matrix R_u used to describe the control input variable is too large, it will cause the control process to pay too much attention to the slow change of the control input variables, and the control system's response time will lengthen such that the visual servo task cannot be completed quickly. It is time-consuming to adjust the weight matrix of the objective function manually, and even then, the optimal control state of the model predictive controller cannot be reached. In this work, an RL-based method is proposed to adjust the weight matrix of the objective function, and it is introduced in the next section.

3.2. RL and policy gradient algorithm

RL has excellent ability to solve sequential problems. In addition, RL is a computational method, and robots can achieve their goals by interacting with the environment. RL is usually considered a Markov decision process (MDP). MDP is based on the tuple $\langle S, P, r, \delta \rangle$ of the Markov reward process, denoted as $\langle S, A, P, r, \delta \rangle$:

- S is the collection of states for the robot.
- A is the collection of actions that the robot performs according to its current state.
- $P(s' \mid s, a)$ is the state transition function of the current state s for action A to arrive at state s'
- $r(s, a)$ is the reward function used to evaluate the reward generated by action a under the current state s.
- $\delta \in [0, 1]$ is a discount factor, representing the importance of the future return series.

The robot makes a decision of action in a state of the environment and applies the action to the environment. The environment changes, and the reward is passed on to the robot in the next state. This kind of interaction is iterative, and the robot's goal is to maximize the accumulated reward expectation over the process of multiple rounds. When the robot is in the environment s_t to perform action a_t, there will be a reward r_{t+1}, and the environment will be updated to s_{t+1}. When the time approaches infinity, the accumulated reward r_t can be expressed as

$$r_t = r_{t+1} + \delta r_{t+2} + \delta^2 r_{t+3} + \cdots = \sum_{k=0}^{\infty} \delta^k r_{t+k+1}. \tag{3.12}$$

In RL, the DDPG is a combination of the deterministic policy gradient and the deep neural network, which can effectively solve the problem of continuous action space. In this study, DDPG is adopted to adjust the appropriate IBVS model predictive control objective function weight matrix. The strategy for tuning weight matrix parameters can be expressed by ω.

The learning goal is to find the strategy with the highest expected cumulative return, which can be expressed as

$$\omega^*(s) = \arg\max_{\omega} \mathbb{E}\left[\sum_{t=0}^{\infty} \delta^t r\left(s_t, \omega\left(s_t\right)\right) \mid s_0 = s\right] \tag{3.13}$$

The gradient under the highest accumulated reward is expressed as:

$$J(\omega) = \mathbb{E}\left[\sum_{t=0}^{\infty} \delta^t r\left(s_t, \omega\left(s_t\right)\right)\right] \tag{3.14}$$

According to the deterministic strategy gradient theory, the parameterized vector η of the optimal strategy is expressed as

$$\nabla_\eta J(\omega) = \mathbb{E}_{s_t \sim \rho}\left[\nabla_a Q(s, a) \mid_{a=\omega(s|\eta)} \nabla_\eta \omega(s \mid \eta)\right] \tag{3.15}$$

where ρ is a random probabilistic strategy, and $Q(s, a)$ denotes the action value function.

3.3. MPC weight matrices tuned by DDPG

To optimize the visual servo control performance by tuning the MPC weight matrices based on DDPG, the element of the MDP solved by DDPG can be expressed as follows:

- As shown by Eq (3.1), the variation in feature points is jointly influenced by $T_d \frac{1}{Z_i} \Omega$ and $\dot{q}$. The parameters of the weight matrices determine the output joint velocity $\dot{q}$, which depends on $T_d \frac{1}{Z_i} \Omega$. There, S is defined as $T_d \frac{1}{Z_i} \Omega$.
- A is defined as the MPC weight matrix (R_s, R_u), where $0 I_{8\times8} \le R_s \le 30 I_{8\times8}$, $0 I_{2\times2} \le R_u \le 5 I_{6\times6}$.

- $r(s, a)$ represents the effect on the MPC-based IBVS task with action a under state s. If the deviation between the state of the feature points and the desired state of the u- and v- axes is within the threshold value, a positive reward will be given. If the visual servo task fails, nonpositive reward will be given. Every control action that affects the image coordinate deviation and deviation rate will be punished accordingly, so that the visual servo task is successful, and the control efficiency is improved.

$$r_t = \begin{cases} 1 & for |s_{e(u)}| < \zeta, |s_{e(v)}| < \zeta \\ -1 & \text{visual servo task failed} \\ -s_e^T W_e s_e - s_{\dot{e}}^T W_{\dot{e}} s_{\dot{e}} & else \end{cases}$$

The process of the DDPG-based MPC IBVS algorithm is shown in Algorithm 1. Four networks are required in the DDPG algorithm: The actor network is denoted as ω^Q, the critic network is denoted as ω^τ, the actor target network is denoted as $\omega^{Q'}$, and the critic target network is denoted as $\omega^{\tau'}$. The robot performs action a_t at state s_t, updates to the next state $s_t + 1$ and obtains the reward r_t. The Q-value of the critic target network Q_{c-t} can be obtained by the following formula:

$$Q_{c-t} = r_i + \delta Q'\left(s_{t+1}, \tau'\left(s_{t+1}|\omega^{\tau'}\right)|\omega^{Q'}\right) \tag{3.16}$$

The minimal loss function of the critical network is calculated by the gradient descent algorithm

$$L_{\omega^\tau} = \frac{1}{N}\sum_{i=1}^{N} Q_{c-t} - Q\left(s_t, a_t|\omega^\tau\right), \tag{3.17}$$

$$\nabla L_{\omega^\tau} = \frac{1}{N}\left[Q_{c-t} - Q\left(s_t, a_t|\omega^\tau\right)\right]\nabla_{\omega^\tau} Q\left(s_t, a_t|\omega^Q\right) \tag{3.18}$$

The actor network parameters are updated with the following formula:

$$\nabla_{\omega^Q} J \approx \frac{1}{N}\sum_i \nabla_a Q\left(s, a|\omega^\tau\right)|_{a=Q(s|\omega^Q)} \nabla_{\omega^Q} Q(s|\omega^Q). \tag{3.19}$$

The actor target network and the critic target network are updated by the exponential smoothing method.

$$\omega^{Q'} \leftarrow \sigma\omega^Q + (1-\sigma)\omega^{Q'} \tag{3.20}$$

$$\omega^{\tau'} \leftarrow \sigma\omega^\tau + (1-\sigma)\omega^{\tau'} \tag{3.21}$$

4. Experiments and discussion

To confirm the effectiveness of the proposed method, this section gives the simulation comparison experiments of different IBVS methods acting on the same visual servo task. The differences in performance between different control methods will be described in detail.

The simulation object of this study is a visual servo system of a 6-DOF robot manipulator [39]. The camera is placed on the last joint of the 6-DOF robot manipulator. The focal length of the camera f_0 is 0.0005 m. The scaling factors along the u- and v- axes are 269,167 pixels/m and 267,778 pixels/m,

Algorithm 1: DDPG-based MPC IBVS Algorithm

```
Initialize soft updating rate $\sigma$ and discount factor $\delta$;
Initialize the parameterized actor network $\omega^{Q}$ and parameterized critic network $\omega^{\tau}$;
Initialize the parameterized actor target network $\omega^{Q'}$ and parameterized critic target network
  $\omega^{\tau\prime}$;
Initialize Gaussian noise $\kappa$;
Initialize memory pool $M_p$;
for episode = 1,2,$\cdots$, $N$ do
    Perceive initialization environment $s_1$;
    for $t$ = 1,2,$\cdots$, $T$ do
        Select action $a_t$= $(\mathrm{R_s}, \mathrm{R_u})$=$\tau\,(s_t|\omega^{\tau})$ + $\kappa$;
        The IBVS MPC produces the input torque under the weight matrix set in this step;
        Observe the reward $r_t$ and observe the new state $s_t + 1$;
        Store the state transition data pair $(s_t, a_t, r_t, s_{t+1})$ into the memory pool $M_p$;
        Copy $M$ members $(s_t, a_t, r_t, s_{t+1})$ from $M_p$ randomly;
        Calculate $Q_{c-t}$ according to Eq. (3.16);
        Calculate the critic network according to Eq. (3.16) and (3.18);
        Calculate the actor network according to Eq. (3.19);
        Calculate the target networks according to Eq. (3.20) and (3.21);
    end
end
```

respectively. The coordinates of the image feature points are fed back from the camera and passed to the 6-DOF robot manipulator, which generates a change in pose so that the image feature points respond from the initial position to the desired position. Simulation training and experiments were performed using MATLAB/Simulink on a laptop computer with a 2.3 GHz Intel Core i7. In RL, the process of an intelligence executing a certain strategy to reach the termination state from the start state is usually referred to as an episode. In this study, the following rules need to be followed each time during the learning process:

1) During the learning process, if the squared difference between the current coordinates of the image feature point and the desired coordinates is lower than the set threshold, it is considered that the visual servo task is successful, and this round of learning ends.
2) During the learning process, if the squared difference between the current coordinates of the image feature point and the desired coordinates is higher than the set threshold, it is considered that the visual servo task has failed, and the current round of learning is ended.
3) During the learning process, if the current coordinates of the image feature point break the image constraint, the visual servo task is considered to have failed, and the current round of learning ends.

In the actor and actor target network, the actor and actor target are the input quantities, and action a is the output quantity. In the critic and critic target networks, the action pair (s, a) is the input, and the action value function $Q(s, a)$ is the output. The activation function gives the neural network a nonlinear modeling capability. In this study, the activation function is chosen as the ReLU function described by Eq (4.1).

$$f_r = \max(0, t) \tag{4.1}$$

The reward weight matrices are set as follows:

$$W_e = \begin{bmatrix} 0.5 & 0 \\ 0 & 1 \end{bmatrix}, \tag{4.2}$$

$$W_{\dot{e}} = \begin{bmatrix} 0.1 & 0 \\ 0 & 0.05 \end{bmatrix} \tag{4.3}$$

A total of 1000 experiments were carried out, with 20 episodes in each round. The reward value for each round of experiments is the average value of 20 experiments. The hyperparameters of the DDPG algorithm are shown in Table 1.

Table 1. The hyperparameters of DDPG.

Parameter	Value
Discount Factor (δ)	0.95
Network soft update parameters (σ)	0.0005
Experience replay pool size	10^5
Numbers of hidden layers	2
First hidden layer size	40
Second hidden layer size	30

In the simulation training and experiments, the visual servo has four image feature points. The 3-D Cartesian coordinates of the image feature points are mapped to the two-dimensional image framework, arranged counterclockwise from $O_1 \rightarrow O_2 \rightarrow O_3 \rightarrow O_4$. We set the desired position of the visual servo feature points in the image plane as $(400, 525)^T$pixels, $(720, 525)^T$pixels, $(720, 320)^T$pixels, $(400, 320)^T$pixels.

Remark 1. *The visibility constraint of the camera u-axis and v-axis can be expressed as* $u_{\min} = 0, u_{\max} = 1292, v_{\min} = 0, v_{\max} = 964$. *If the initial coordinates of randomly generated image feature points are outside the camera vision constraint, a set of initial image feature point coordinates will be randomly generated again until the initial coordinates of all image feature points are within the camera vision constraint.*

The purpose of the comparative simulation experiments is to demonstrate the accuracy and stability of the DDPG-MPC IBVS method proposed in this paper. The simulation is divided into two parts. First, the control effect of this method is compared with the traditional model predictive control visual servo method (MPC-IBVS) [21]. Then, the control effect of the proposed method in this study is compared with the model predictive control IBVS method tuned by Q-learning (Q-learning-MPC IBVS).

4.1. Comparison with traditional IBVS methods

In this section, the control effects of the proposed DDPG-MPC IBVS method and traditional MPC-IBVS are analyzed. In addition, to prove that the proposed method has a more stable and accurate control effect, this section selects two groups of different weight matrices for the MPC-IBVS method, which are called MPC-IBVS-A and MPC-IBVS-B. The weight matrix of MPC-IBVS-A is set as $R_s = 2I_{8\times8}, R_u = I_{6\times6}$, the predictive time domain is set as $N_p = 5$, and the control time domain is set as $N_c = 2$. The weight matrix of MPC-IBVS-B is set as $R_s = 25I_{8\times8}, R_u = 5I_{6\times6}$, the predictive time domain is set as $N_p = 5$, and the control time domain is set as $N_c = 2$. The sampling period of the controller is 40 ms.

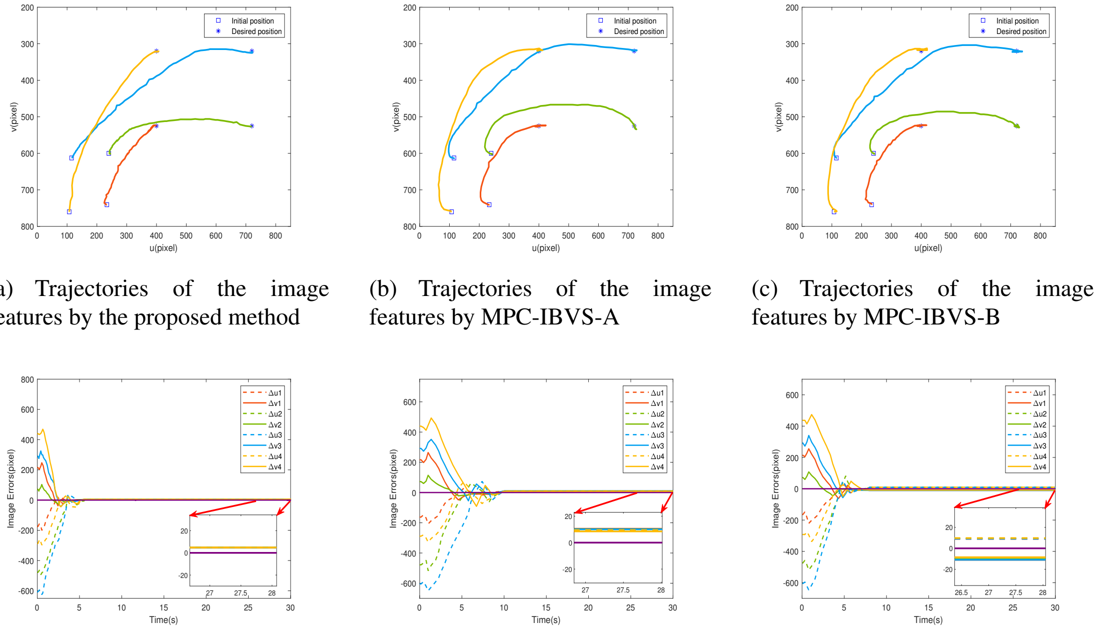
The weight matrix after training and rectification by the proposed method in the last part is

$$R_s = \begin{bmatrix} 10 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 8 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 7 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 5 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 5 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 6 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 8 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 9 \end{bmatrix}, R_u = \begin{bmatrix} 5 & 0 & 0 & 0 & 0 & 0 \\ 0 & 3 & 0 & 0 & 0 & 0 \\ 0 & 0 & 2 & 0 & 0 & 0 \\ 0 & 0 & 0 & 3 & 0 & 0 \\ 0 & 0 & 0 & 0 & 2 & 0 \\ 0 & 0 & 0 & 0 & 0 & 4 \end{bmatrix} \tag{4.4}$$

The predictive time domain is set as $N_p = 5$, and the control time domain is set as $N_c = 2$. To test the stability of various control methods, the desired feature point image coordinates are set to $(400, 525)^T$ pixels, $(720, 525)^T$ pixels, $(720, 320)^T$ pixels, $(400, 320)^T$ pixels. The first set of initial feature point image coordinates p_1 are set to $(233.5, 740.7)^T$ pixels, $(240, 600)^T$ pixels, $(115.2, 612.7)^T$ pixels, $(107.3, 760.1)^T$ pixels, and the second set of initial feature point image coordinates p_2 are set to $(917.2, 350.1)^T$ pixels, $(919, 200.3)^T$ pixels, $(801.9, 226.1)^T$ pixels, $(800.2, 374.8)^T$ pixels. The robot manipulator joint velocity constraint is limited to 0.5 rad/s.

First, Figure 3 gives the comparative simulation results of the DDPG-MPC IBVS, MPC-IBVS-A and MPC-IBVS-B methods under the initial coordinates p_1. From Figure 3(a)–(c), it can be observed that under all three IBVS methods, the image feature points can successfully respond from the initial coordinates to the desired coordinates and finally stabilize in the desired state. However, Figure 3(d)–(f) shows that the image deviation converges at the fastest rate under the proposed method compared to the other methods. The settling time of the DDPG-MPC IBVS method is approximately 5 s. The settling time of MPC-IBVS-A is approximately 10 s. The settling time of MPC-IBVS-B is approximately 8 s. The steady-state error of the proposed method is less than those of MPC-IBVS-A and MPC-IBVS-B. The above experiments show that the proposed method has better control performance when the initial position is p_1.

Then, the results of the comparative simulation at the initial coordinates p_2 are given in Figure 4. From Figure 4(a)–(c), it can be observed that the image feature points can also respond from the initial coordinates to the desired coordinates and stabilize in the desired state under all three MPC-based IBVS methods. Similar to the above simulation results, the proposed method can make the image feature points respond and stabilize at the desired coordinates as fast as possible (approximately 5 s). Different from the above simulation results, in the initial state p_2, the settling time of MPC-IBVS-A (approximately 8 s) is longer than that of MPC-IBVS-B (approximately 10 s). This indicates that the



(a) Trajectories of the image features by the proposed method

(b) Trajectories of the image features by MPC-IBVS-A

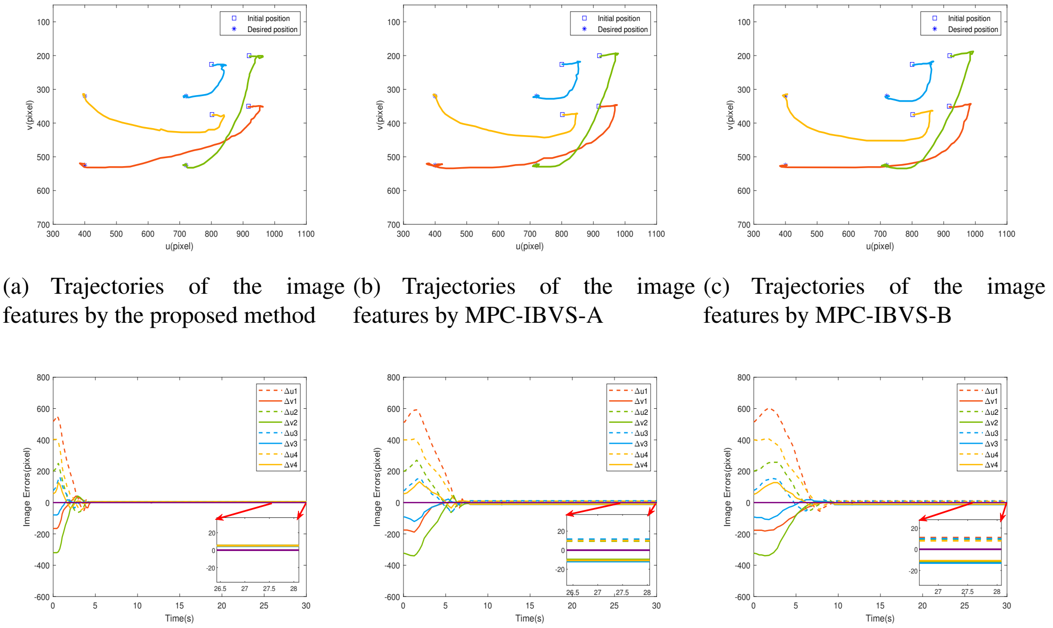
(c) Trajectories of the image features by MPC-IBVS-B

(d) Image errors by the proposed method

(e) Image errors by MPC-IBVS-A

(f) Image errors by MPC-IBVS-B

Figure 3. Comparative simulation results under initial coordinates p_1.



(a) Trajectories of the image features by the proposed method

(b) Trajectories of the image features by MPC-IBVS-A

(c) Trajectories of the image features by MPC-IBVS-B

(d) Image errors by the proposed method

(e) Image errors by MPC-IBVS-A

(f) Image errors by MPC-IBVS-B

Figure 4. Comparative simulation results under initial coordinates p_2.

different parameters of the weight matrix of the objective function set considered cannot remain stable in giving the optimal control effect in different visual servo tasks. This is because the weight matrix contains multiple parameters, and the selection of the optimal weight matrix parameters by humans is also not achievable in practical applications. With the increase in accumulated rewards, the intensive training method can gradually find the optimal weight matrix parameters, thus achieving better visual servo control performance.

In the IBVS tasks, the relationship between the target object and the visual servo system is usually represented by four or more feature points. As shown in Eq (4.4), the four feature points in this study correspond to the eight parameters in the image deviation weight matrix R_s. At the same time, the 6-DOF manipulator corresponds to six parameters in the weight matrix R_u. Then, there are fourteen parameters of the weight matrices that need to be set in the MPC-based IBVS method. Although these fourteen parameters can be given artificially through trial and error, it is possible to complete the visual servo task, but finding the optimal weight matrix parameters requires a lot of work, which is impossible in practical applications. The proposed method can get better weight matrix parameters with a certain number of training sets in less than one hour, improve the servo efficiency and accuracy and effectively improve the control quality.

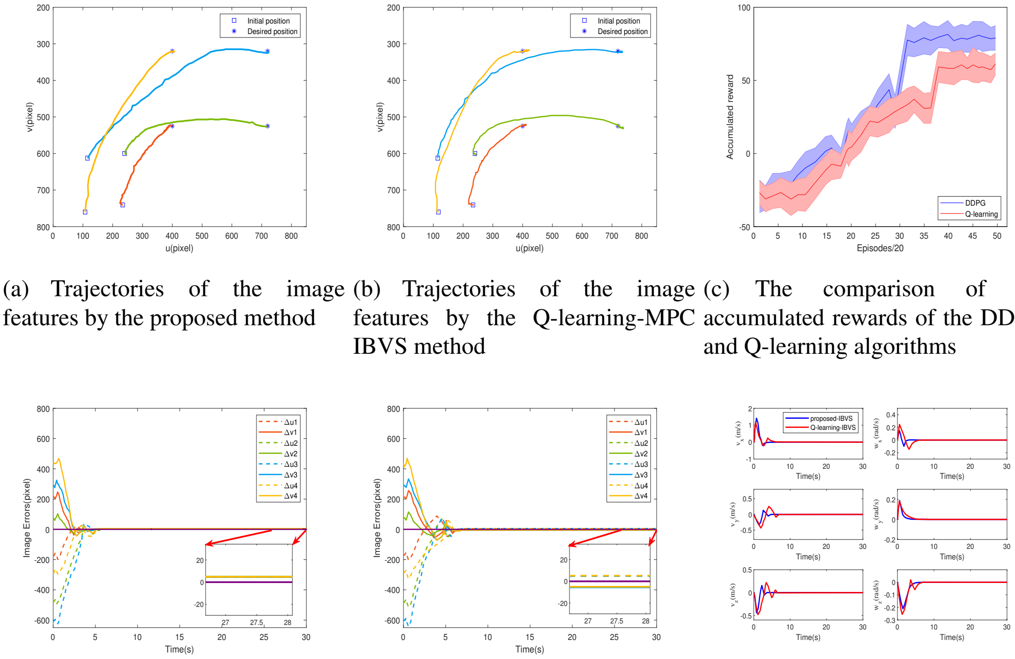
The evaluation results of the above simulation experiments are shown in Table 2, which demonstrates that compared with the traditional MPC-based IBVS method, the DDPG-MPC IBVS method can complete the constrained visual servo task with higher control quality.

Table 2. The evaluation results of different MPC-based IBVS methods.

Initial state	Control method	DDPG-MPC-IBVS	MPC-IBVS-A	MPC-IBVS-B
p_1	Settling time (s)	5.24	9.98	8.13
	Image deviation (pixels)	4.98	10.21	10.5
	Control overshoot	12%	20%	19%
p_2	Settling time (s)	4.89	8.07	10.1
	Image deviation (pixels)	5.1	11.26	11.18
	Control overshoot	12%	18%	20%

4.2. *Comparison with the Q-learning-MPC IVBS method*

In this part, the results of the comparative simulation between the DDPG-MPC IBVS method and the Q-Learning-MPC IBVS method are analyzed. Q-learning is an extremely important RL algorithm, and $Q(s, a)$ is the benefit obtained by taking action a in a certain state s. The environment will provide feedback on the corresponding reward according to the action reward, so the main idea of the Q-learning algorithm is to build a Q-table with states and actions to store Q-values and then select the action that can obtain the maximum gain according to the Q-values. In the simulation of this study, the Q-learning algorithm is composed of a low-latitude Q-table. The objective function weight matrix is discretely tuned by the Q-learning algorithm.



(a) Trajectories of the image features by the proposed method

(b) Trajectories of the image features by the Q-learning-MPC IBVS method

(c) The comparison of the accumulated rewards of the DDPG and Q-learning algorithms

(d) Image errors by the proposed method

(e) Image errors by the Q-learning-MPC IBVS method

(f) The camera velocities of the proposed method and the Q-learning-MPC IBVS method

Figure 5. Comparative simulation results of different RL algorithm IBVS methods.

In the comparison experiments, the desired coordinates of the feature points are the same as in the previous section, and the initial coordinates of the feature points are p_1. From Figure 5(a),(b) we find that both the DDPG-MPC IBVS method and the Q-learning-MPC IBVS method can cope well with the system constraints. From Figure 5(d),(e), it can be observed that DDPG-MPC IBVS has a faster servo deviation convergence speed than the Q-Learning-MPC IBVS method, at 5 s versus 7 s, respectively. Furthermore, the deviation response curve based on DDPG-MPC IBVS is smoother. This is because the action set of Q-learning is discrete. In contrast, DDPG trains and corrects continuous visual servo actions through continuous network functions. Therefore, the DDPG algorithm tunes the more appropriate weight matrix of the objective function, so as to achieve a better visual servo control. As shown in Figure 5(c), the cumulative rewards of the two RL methods increase with the increase of rounds. However, the final cumulative rewards of the DDPG algorithm are higher than that of the Q-learning algorithm, which indicates that the DDPG algorithm has achieved better training results. This means that the DDPG-IBVS method can adjust more suitable weight matrix parameters, which leads to better visual servo control performance. The changes of the camera velocity $V = \left[v_x, v_y, v_z, w_x, w_y, w_z\right]^T \in R^{6\times1}$ of the two methods are shown in Figure 5(f). The camera velocities do not fluctuate too much under the two methods, which means that the response processes are both smooth.

To describe the effectiveness of the proposed method more vividly, we quantitatively analyze the visual servo control effect for the model prediction control rectified by two different reinforcement learning algorithms. As seen from Table 3, after 30 random experiments, both methods can complete

Table 3. The evaluation results of different RL-MPC IBVS methods.

Control method	DDPG-MPC IBVS	Q-learning-MPC IBVS
Success rate	100%	100%
Average overshoot	12%	14%
Average settling time (s)	5.12	7.39
Average image deviation (pixels)	5.06	5.2

the constrained visual servo task with a high success rate and similar average image deviation. However, the DDPG-MPC IBVS method always has a faster convergence speed.

In conclusion, the proposed method has faster servo efficiency and better control quality compared to the existing IBVS methods. The proposed method can effectively perform visual servo tasks.

5. Conclusions

In visual servoing, system constraints contain visibility constraints, and actuator constraints must be considered. To solve the constrained visual servo problem, a model predictive control IBVS method tuned by RL is proposed in this study. First, a depth-independent Jacobian matrix is established as the predictive model, and the optimal control input is found by minimizing the cost function of the predictive error. Different from traditional model predictive control methods, the weight matrix of the objective function is adjusted offline by the DDPG algorithm. Appropriate states, rewards and actions are defined in the training progress. Then, the accumulated rewards converge to specific values, which means that the DDPG algorithm has successfully learned the appropriate weight matrix parameters. Finally, in simulation experiments of a 6-DOF manipulator, the control effect of the proposed method is compared and analyzed with other visual servo control methods, and we find that the proposed method has better performance. In future work, we plan to explore the delayed visual servo control strategies caused by low-quality visual signals. We will design a predictive control IBVS method based on a time-delay predictive model, to improve the control stability of visual servo systems with time delay.

Acknowledgments

This research was funded by the Natural Science Foundation of Heilongjiang Province, grant number KY10400210217, and the Fundamental Strengthening Program Technical Field Fund, grant number 2021-JCJQ-JJ-0026.

Conflict of interest

The authors declare there is no conflict of interest.

References

1. X. Liang, M. Peng, J. Lu, C. Qin, A visual servo control method for tomato cluster-picking manipulators based on a TS fuzzy neural network, *Trans. ASABE*, **64** (2021), 529–543. https://doi.org/10.13031/trans.13485
2. R. J. Chang, C. Y. Lin, P. S. Lin, Visual-based automation of Peg-in-Hole microassembly process, *Trans. ASABE*, **133** (2011), 041015. https://doi.org/10.1115/1.4004497
3. A. A. Palsdottir, M. Mohammadi, B. Bentsen, L. N. S. A. Struijk A dedicated tool frame based tongue interface layout improves 2D visual guided control of an assistive robotic manipulator: A design parameter for Tele-Applications, *IEEE Sensors J.*, **22** (2022), 9868–9880. https://doi.org/10.1109/JSEN.2022.3164551
4. Y. W. Zhang, Y. C. Liu, Z. W. Xie, Y. Liu, B. S. Cao, H. Liu, Visual servo control of the Macro/Micro manipulator with base vibration suppression and backlash compensation, *App. Sci. Basel*, **12** (2022). https://doi.org/10.3390/app12168386
5. R. Sharma, S. Shukla, L. Behera, Position-based visual servoing of a mobile robot with an automatic extrinsic calibration scheme, *Robotica*, **38** (2020), 831–844. https://doi.org/10.1017/S0263574719001115
6. S. Heshmati-alamdari, A. Eqtami, G. C. Karras, D. V. Dimarogonas, K. J. Kyriakopoulos, A Self-triggered position based visual servoing model predictive control scheme for underwater robotic vehicles, *Machines*, **8** (2020). https://doi.org/10.3390/machines8020033
7. Y. Zhao, W. F. Xie, S. Liu, Image-based visual servoing using improved image moments in 6-DOF robot systems, *Int. J. Control Autom. Syst.*, **11** (2013), 586–596. https://doi.org/10.1007/s12555-012-0232-9
8. O. Tahri, H. Araujo, F. Chaumette, Y. Mezouar, Robust image-based visual servoing using invariant visual information, *Robot. Auton. Syst.*, **61** (2013), 1588–1600. https://doi.org/10.1016/j.robot.2013.06.010
9. D. J. Guo, X. Jin, D. Shao, J. Y. Li, Y. Shen, H. Tan, Image-based regulation of mobile robots without pose measurements, *IEEE Control Syst. Lett.*, **6** (2022), 2156–2161. https://doi.org/10.1109/LCSYS.2021.3139288
10. N. Garcia-Aracil, C. Perez-Vidal, J. M. Sabater, R. Morales, F. J. Badesa, Robust and cooperative image-based visual servoing system using a redundant architecture, *Sensors*, **11** (2011), 11885–11900. https://doi.org/10.3390/s111211885
11. S. T. Liu, J. X. Dong, Robust online model predictive control for image-based visual servoing in polar coordinates, *Trans. Inst. Meas. Control*, **42** (2020), 890–903. https://doi.org/10.1177/0142331219895074
12. A. Rastegarpanah , A. Aflakian, R. Stolkin, Improving the manipulability of a redundant arm using decoupled hybrid visual servoing, *Appl.Sci. Basel*, **11** (2022). https://doi.org/10.3390/machines8020033
13. Z. He, C. Wu, S. Zhang, X. Zhao, Moment-Based 2.5-D visual servoing for textureless planar part grasping, *IEEE Trans. Ind. Electron.*, **66** (2019), 7821–7830. https://doi.org/10.1109/TIE.2018.2886783

14. F. Yan, B. Li, W. Shi, D. Wang, Hybrid visual servo trajectory tracking of wheeled mobile robots, *IEEE Access*, **6** (2018), 24291–24298. https://doi.org/10.1109/ACCESS.2018.2829839

15. X. J. Li, J. A. Gu, Z. D. Huang, C. Ji, S. X. Tang, Hierarchical multiloop MPC scheme for robot manipulators with nonlinear disturbance observer, *Math. Biosci. Eng.*, **19** (2022), 12601–12616. https://doi.org/10.3934/mbe.2022588

16. M. Mohammad Hossein Fallah, F. Janabi-Sharifi, Conjugated visual predictive control for constrained visual servoing, *J. Intell. Robotic Syst.*, **101** (2021), 1–21. https://doi.org/10.1007/s10846-020-01299-6

17. Z. Qiu, S. Hu, X. Liang, Model predictive control for constrained image-based visual servoing in uncalibrated environments, *Asian J. Control*, **21** (2019), 783–799. https://doi.org/10.1002/asjc.1756

18. T. Wang, W. Xie, G. Liu, Y. Zhao, Quasi-min-max model predictive control for image-based visual servoing with tensor product model transformation, *Asian J. Control*, **17** (2015), 402–416. https://doi.org/10.1002/asjc.871

19. J. Gao, G. Zhang, P. Wu, X. Zhao, T. Wang, W. Yan, Model predictive visual servoing of fully-actuated underwater vehicles with a sliding mode disturbance observer, *IEEE Access*, **7** (2019), 25516–25526. https://doi.org/10.1109/ACCESS.2019.2900998

20. Z. Jin, J. Wu, A. Liu, W. A. Zhang, L. Yu, Gaussian process-based nonlinear predictive control for visual servoing of constrained mobile robots with unknown dynamics, *Robotics Auton. Syst.*, **136** (2021), 103712. https://doi.org/10.1016/j.robot.2020.103712

21. G. Allibert, E. Courtial, F. Chaumette, Predictive control for constrained image-based visual servoing, *IEEE Trans. Robotics*, **26** (2010), 933–939. https://doi.org/10.1109/TRO.2010.2056590

22. R. Shridhar, D. J. Cooper, A tuning strategy for unconstrained multivariable model predictive control, *Indust. Eng. Chem. Res.*, **37** (1998), 4003–4016. https://doi.org/10.1021/ie980202s

23. R. Suzuki, F. Kawai, H. Ito, C. Nakazawa, Y. Fukuyama, E. Aiyoshi, Automatic tuning of model predictive control using particle swarm optimization, *IEEE Swarm Intell. Symp.*, (2007), 221–226. https://doi.org/10.1109/SIS.2007.367941

24. K. Han, J. Zhao, J. Qian, A novel robust tuning strategy for model predictive control, *World Congr. Intell. Control Autom.*, **2** (2006), 6406–6410.

25. J. Van der Lee, W. Svrcek, B. Young, A tuning algorithm for model predictive controllers based on genetic algorithms and fuzzy decision making, *ISA Trans.*, **47** (2008), 53–59. https://doi.org/10.1016/j.isatra.2007.06.003

26. S. Q. Chen, Y. Yang, R. Su, Deep reinforcement learning with emergent communication for coalitional negotiation games, *Math. Biosci. Eng.*, **19** (2022), 4592–4609. https://doi.org/10.3934/mbe.2022212

27. G. Ciaburro, Machine fault detection methods based on machine learning algorithms: A review, *Math. Biosci. Eng.*, **19** (2021), 11453–11490. https://doi.org/10.3934/mbe.2022534

28. M. Sedighizadeh, A. Rezazadeh, A modified adaptive wavelet PID control based on RL for wind energy conversion system control, *Adv. Electr. Comput. Eng.*, **10** (2010), 153–159. https://doi.org/10.4316/AECE.2010.02027

29. D. Lee, S. J. Lee, S. C. Yim, Reinforcement learning-based adaptive PID controller for DPS, *Ocean Eng.*, **216** (2020), 108053. https://doi.org/10.1016/j.oceaneng.2020.108053

30. I. Carlucho, M. De Paula, G. G. Acosta, Double Q-PID algorithm for mobile robot control, *Expert Syst. Appl.*, **137** (2019), 292–307. https://doi.org/10.1016/j.eswa.2019.06.066

31. T. Chaffre, G. Le Chenadec, K. Sammut, E. Chauveau, B. Clement, Direct adaptive pole-placement controller using deep reinforcement learning: Application to auv control, *IFAC-PapersOnLine*, **54** (2021), 333–340. https://doi.org/10.1016/j.ifacol.2021.10.113

32. M. Kang, H. Chen, J. X. Dong, Adaptive visual servoing with an uncalibrated camera using extreme learning machine and Q-leaning, *Neurocomputing*, **402** (2020), 384–394. https://doi.org/10.1016/j.neucom.2020.03.049

33. M. Mehndiratta, E. Camci, E. Kayacan, Automated tuning of nonlinear model predictive controller by reinforcement learning, *IEEE/RSJ Int. Confer. Intell. Robots Syst.*, (2018), 3016–3021.

34. P. T. Jardine, S. N. Givigi, S. Yousefi, Experimental results for autonomous model-predictive trajectory planning tuned with machine learning, *IEEE Int. Syst. Confer.*, (2017), 663–669.

35. K. M. Cabral, S. R. B. dos Santos, S. N. Givigi, C. L. Nascimento, Design of model predictive control via learning automata for a single UAV load transportation, *IEEE Int. Syst. Confer.*, (2017), 656–662.

36. F. Wang, B. M. Ren, Y. Liu, B. Cui, Tracking moving target for 6 degree-of-freedom robot manipulator with adaptive visual servoing based on deep reinforcement learning PID controller, *Rev. Sci. Instrum.*, **93** (2022), 045108. https://doi.org/10.1063/5.0087561

37. Z. H. Jin, J. H. Wu, A. D. Liu, W. A. Zhang, L. Yu, Policy-based deep reinforcement learning for visual servoing control of mobile robots With visibility constraints, *IEEE Trans. Ind. Electron.*, **69** (2021), 1898–1908. https://doi.org/10.1109/TIE.2021.3057005

38. Y. C. Liu, C. Y. Huang, DDPG-based adaptive robust tracking control for aerial manipulators with decoupling approach, *IEEE Trans. Cybern.*, **52** (2021), 8258–8271. https://doi.org/10.1109/TIE.2021.3057005

39. P. M. Kebria, S. Al-Wais, H. Abdi, S. Nahavandi, Kinematic and dynamic modelling of ur5 manipulator, *IEEE Int. Confer. Syst. Man Cybern.*, (2016), 004229–004234.

40. F. Chaumette, S. Hutchinson, Visual servo control. i. basic approaches, *IEEE Robotics Autom. Mag.*, **13** (2006), 82–90. https://doi.org/10.1109/MRA.2006.250573