International Journal of Mechatronics and Applied Mechanics
2025, Issue 19, Vol. 1

PREDICTIVE CONTROL OF ROBOTIC ARM BASED ON IMAGE PROCESSING AND FORECASTING ALGORITHM

Jieyu Tian[1], Yingya Huang[1]*
[1]College of Railway Locomotive and Rolling Stock, Hebei Vocational College of Rail Transportation, Shijiazhuang, 050051, China

Abstract - The traditional visual servo system of the PUMA560 robotic arm suffers from model complexity and difficulties. This study proposes a multivariate feedback-based visual servo control strategy to address these challenges and improve the system's performance. The proposed strategy uses the difference between the center of the target object and the desired coordinates as the feedback signal, combined with a forecasting algorithm to create a predictive control system. Simulation experiments were conducted to evaluate the system's performance. The experimental results show that the proposed model-free adaptive method, integrating forecasting algorithms and image processing, significantly reduces tracking error. The optimized method achieves a tracking error of around ±5 pixels, compared to ±10 pixels without optimization and ±7 pixels without the forecasting algorithm. This research introduces an innovative predictive control strategy that effectively overcomes the challenges of unknown parameters and models in visual servo systems, improving tracking accuracy and performance for robotic arms.

Keywords: Image processing, Robotic arm, Forecasting algorithm, Visual servo system, Predictive control

1. Introduction

Nowadays, the robotics industry is growing as an automated and intelligent device that can replace humans in performing many complex and repetitive tasks without human intervention [1]. However, the rapid development of robotics has also brought many challenges. Mobile robotic arm is a common type of mobile robot, which consists of a motion platform and a robotic arm mounted on that platform [2], and this structure not only endows the mobile robotic arm with a wide range of motion, but also enhances the flexibility of its independent operation [3]. The visual servo system of the robotic arm is a nonlinear, highly coupled and complex system containing many variables, which acquires the external image information through the camera and extracts the pixel coordinates of the target feature points to perform the task [4]. Because of the uncertainty of the dynamic environment and the complexity of its own dynamic characteristics, the robotic arm will be limited in performing tasks dependent on data training and other problems. To further strengthen the working ability of the robotic arm, it is necessary to establish a more accurate model of its camera and enable it to better track the target needs to design a better control system. The study combined the image processing and forecasting algorithms in the multi-layer step forecasting method to optimize the traditional servo control system of the robotic arm, which is designed to improve the performance of the robotic arm and to enhance the robotic arm's tracking effect on the target.

The total research is divided into four parts, the first part is to summarize and discuss the domestic and international research on robotic arms. The second part is to improve the servo system using image processing and forecasting algorithm. The third part is to verify the performance of the model-free adaptive control system. The fourth part is to summarize the whole article.

2. Related Work

With the continuous development of industrial technology, the robotic arm gradually replace manual labor to complete many more complex and difficult work, to improve the accuracy of the robotic arm gripping items and complete the efficiency of the tracking task, and the traditional visual servo system in the completion of these higher requirements of the work is gradually strenuous, and therefore need to improve it [5]. Image processing, is the technique of

https://doi.org/10.17683/ijomam/issue19

analyzing an image using a computer to achieve a desired result [6]. Ngugi L. C. and his research team used image processing techniques to perform plant disease recognition in order to detect plant diseases fast and accurately. The results of field tests demonstrated that the method has high generalizability, short training times, and high accuracy in plant disease recognition [7]. Cong et al. proposed a visual servo control method based on decoupled images and integrated image processing into the system to increase the flexibility of robotic arm sorting. The experimental results showed that the algorithm can effectively increase the system's stability and efficiency in real-time [8]. The CNN model and image processing techniques were combined by Taira T and his team to automatically detect appearance defects during the production process. The experimental results showed that this method can effectively reduce the system cost and improve the production line inspection efficiency [9]. In an attempt to address the issue of traditional robotic arms' inability to distinguish between overlapping parts in poorer environmental conditions, a weighted integrated neural network model was presented by Cheng B. and associates. This model enables the processing of overlapping areas of photos when paired with integrated learning and stochastic cropping technology. The experiments' outcomes indicated how effective this approach is for the image categorization problem [10]. To address the robotic arm's inefficiency in automatically sorting and recycling garbage, Huang D. et al. proposed a method that combines sensors and image recognition. Through experiments, they demonstrated that the system can effectively improve the accuracy and efficiency of garbage recycling in practice [11].

Additionally, Zahaf and the members of his group redesigned the prediction system using LMI and proposed an HFTPC strategy for industrial robotic arms that might experience faults or delay variations when using hybrid actuators. The results of the simulation experiment demonstrated the method's effectiveness and robustness [12]. To address the issue that the WBC of incomplete mobile robotic arms needs to be optimized and combined with primitive dyadic neural networks, Yuan and his team proposed a MIMO and MPC based WBC strategy for mobile robotic arms. Trajectory tracking experiments on a mobile dual-armed robot were used to confirm the method's efficacy [13]. Tika et al. proposed a hierarchical control method for cooperative work tasks in narrow spaces in response to the problem of two robotic arms working simultaneously in the same narrow space that is susceptible to inflation. They did this by combining kinematics and a distance minimization algorithm to turn the trajectory planning problem into a temporal problem, which allowed two robotic arms to work simultaneously in the same narrow space [14]. To solve the issue that the robot's robotic arm must assess its surroundings before it can begin interacting, Pankert et al. proposed a control scheme that incorporates a tracking trajectory reference. Experiments demonstrated that the approach increases the robotic arm's responsiveness in comparison to the previous one [15]. To control the robotic arm using a system for dynamic compensation, Huang et al. addressed the issue of slower control of spherical soft robotic arms with errors. They proposed a prediction method without offset model, which was shown in an experiment to reduce the tracking error by 35% and enhance the robotic arm's performance [16].

To summarize, researchers at home and abroad have made a lot of experiments to improve the robotic arm system or improve the system algorithm, which also proved the practicality and effectiveness of image processing. However, the robotic arm still exists in the actual execution of the task cannot make timely feedback on the unknown target, and there still exists some unavoidable image errors. Therefore, the study suggested combining forecasting and image processing methods to boost the robotic arm's tracking effect, lower picture mistakes, and improve performance.

3. Image Processing and Forecasting Algorithms for Predictive Control of Robotic Arms

3.1 Adaptive Control Algorithm Based on Image Processing

The research suggested model-free adaptive algorithms based on image processing to accomplish the precise execution of the robotic arm's task. The PUMA560 is a multifunctional robotic arm with six degrees of freedom. This will help to further strengthen its static positioning and dynamic tracking ability, as the camera positioning work can be challenging and result in the robotic arm performing the task with poor accuracy [17]. Image processing refers to the analysis and manipulation of image data by computers, and the core purpose is to improve image quality and reduce the impact of image noise and distortion on the stability of the control system. Image processing is divided into three steps: image preprocessing, edge detection of the target image, and feature point extraction of the target image [18]. The target image captured by the camera on the robotic arm often contains a large amount of noise that causes fluctuations in the pixel values, increasing the complexity of image processing, so image pre-processing is crucial. After evaluating the effect of three common filtering algorithms after processing the image respectively, the Gaussian filtering method has a better processing effect as well as a more stable processing effect, so the study chooses it as the algorithm for denoising processing. Its expression is shown in Eq. (1).

$$\begin{cases} H(x,y) = \dfrac{1}{2\pi\sigma^2} e^{-\frac{a^2+b^2}{2\sigma^2}} \\ G(x,y) = f(x,y) * H(x,y) \end{cases} \tag{1}$$

In Eq. (1), $H(x,y)$ is a two-dimensional Gaussian distribution. The image need to be exposed to edge detection once it has been denoised. The edge is defined as the collection of pixels whose surrounding pixel values fluctuate considerably. It serves as the demarcation line between various areas of the image and is one of its salient aspects. Edge detection i.e. detecting the gray scale of the image, as Eq. (2).

$$G\left(f(x,y)\right)=\left[\left(\partial f/\partial x\right)^2+\left(\partial f/\partial y\right)^2\right]=\left|\Delta_x f\right|+\left|\Delta_y f\right| \quad (2)$$

In Eq. (2), $G\left(f(x,y)\right)$ denotes the image gray scale. Canny edge detection operator is a second order multilateral detection algorithm that first uses Gaussian filtering to process the image noise and then calculates the magnitude and direction of its gradient. The first order differential convolution template expression is as Eq. (3).

$$S_1=\begin{bmatrix}-1 & -1\\ 1 & 1\end{bmatrix}, S_2=\begin{bmatrix}1 & -1\\ 1 & -1\end{bmatrix} \quad (3)$$

The expression for the gradient magnitude is displayed as Eq. (4).

$$\begin{cases}\varphi(x,y)=\sqrt{\left(\varphi_1^2+\varphi_2^2\right)}\\ \varphi_1(x,y)=f(x,y)*S_1\\ \varphi_2(x,y)=f(x,y)*S_2\end{cases} \quad (4)$$

The expression for the direction of the gradient is as Eq. (5).

$$\theta_\varphi=\tan^{-1}\frac{\varphi_2(x,y)}{\varphi_1(x,y)} \quad (5)$$

The study suggests an enhanced smoothing method to include in the calculation of those pixels whose difference in grayscale with the center pixel is less than a set threshold value, in order to retain those pixels that have a greater difference with the center pixel and carry valid information. The canny edge detection algorithm in the use of Gaussian filtering in noise reduction will weaken the effect on the edge of the image because the processing will be unable to distinguish between the edge and the noise. The optimized Canny operator is able to extract the edge details of the image more effectively, with better suppression of noise and less interference by noise. The third step is the extraction of feature points of the target image, i.e., the points in the image where the gray scale changes drastically or the curvature of the edges is large. The primary goal of image processing is to determine the precise pixel coordinates of the target object's center point. Since this cannot be done directly due to the grayscale difference, the study tried to measure the target object's corner points indirectly and use the average value of the corner points' pixels to determine the center coordinates. Curvature Scale Space (CSS) is a modern tool in the field of computer vision that can be used to recognize the corner features of a target object. A larger scale image is first selected to calculate its edge curvature, and the curve expression considering the scale space is as Eq. (6).

$$\begin{cases}\Gamma(u,\sigma)=\left(x(u,\sigma),y(u,\sigma)\right)\\ x(u,\sigma)=x(u)\otimes g(u,\sigma)\\ y(u,\sigma)=y(u)\otimes g(u,\sigma)\end{cases} \quad (6)$$

In Eq. (6), Γ is the curve, u is the arc length, and $g(u,\sigma)$ is the Gaussian kernel with scale σ. The CSS-based feature point extraction process is as Fig. 1.

In Fig. 1, A threshold is set to remove points that might be rounded corners from the initial candidate point set, which is made up of a collection of points with the largest curvature. These points are then thoroughly examined to further weed out pseudo-corner points caused by environmental factors. The control system of the robotic arm visual servo is driven by the difference between the center point coordinates and the desired coordinates obtained by image processing, and the kinematic relationship can be described by the image Jacobi matrix. The role of the image Jacobi matrix is to allow the robotic arm to adjust its motion according to the error feedback signal in the image space. When the coordinates in the image plane match the desired coordinates, it means that the robotic arm has reached the predetermined position and completed the task. To make the movement of the robotic arm more flexible, the study introduces a model-free adaptive algorithm to achieve the establishment of a dynamicized data model to improve the movement performance of the robotic arm. Fig. 2 shows the architecture of the model-free adaptive algorithm-based robotic arm tracking control system.

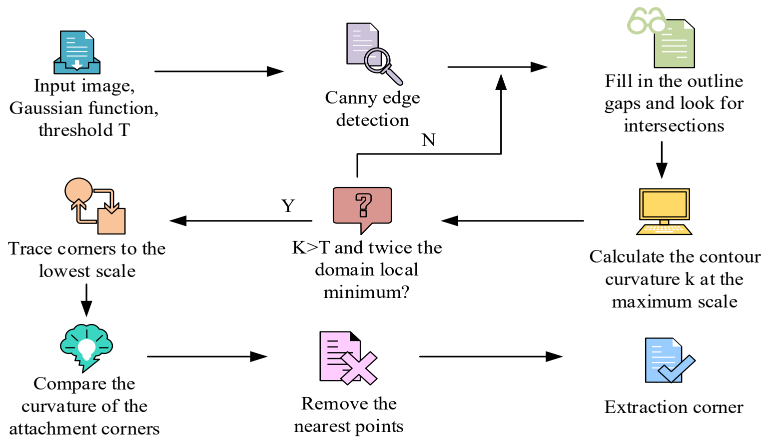


Figure 1: Feature point extraction flow chart

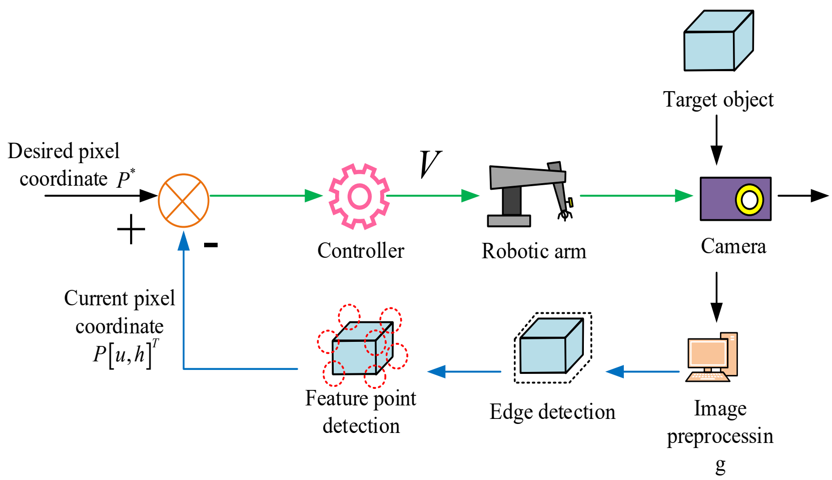


Figure 2: Adaptive tracking control system framework

$$\hat{\varphi}_c(k)=\hat{\varphi}_c(k-1)+\frac{\eta(\Delta p(k))}{\mu+\|\Delta\upsilon(k-1)\|^2}-\frac{\hat{\varphi}_c(k-1)\Delta\upsilon(k-1)\Delta\upsilon^T(k-1)}{\mu+\|\Delta\upsilon(k-1)\|^2} \tag{7}$$

As Fig. 2, the system receives the desired coordinate points as input. The captured image is then preprocessed to output the current pixel coordinates. Based on the output data and the adaptive model, the robotic arm's position is then determined and adjusted, ultimately realizing the task of localizing the target object. The expression of its control algorithm is as Eq. (7).

In Eq. (7), $\hat{\varphi}_c(k)$ is the pseudo-Jacobian matrix of the system and η is the long-step factor. When it happens $|\hat{\varphi}_c(k)|<\varepsilon$ or under the condition of $|\hat{\varphi}_c(k)|>a\varepsilon$, or in the third case, $sign(\hat{\varphi}_{ii}(k))\neq sign(\hat{\varphi}_{ii}(1)),\hat{\varphi}_{ii}(k)=\hat{\varphi}_{ii}(1)$, the expression is as Eq. (8).

$$\Delta\upsilon(k)=\upsilon(k-1)+\frac{\rho\upsilon_c^T(k)(P^*(k+1)-p(k))}{\lambda+\|\varphi_c(k)\|^2} \tag{8}$$

1.2 Predictive Control Method for Robotic Arm by Image Processing and Forecasting

To enhance the robotic arm's control effect, the study used an adaptive control system to modify the motion trajectory of the arm after collecting the image processing findings. There are still some issues because the prediction model in the predictive robotic arm's current control system is built using the image Jacobi matrix. For instance, there are issues with the camera's internal and external settings as well as the precise distance between the camera and the object of interest. Because of this, the study suggests using a forecasting algorithm to give predictive control for robotic arm control. Predictive control aims to decompose the control task into a series of chronological steps, predict the task action trajectory within a predefined timeframe, and relay the command to the system's optimization module to optimize the robotic arm's trajectory route based on performance metrics until the stopping condition is reached.

The predictive control motion process of the PUMA560 robotic arm is a typical nonlinear discrete-time dynamic with multiple information inputs along with multiple information output behaviors. Its expression is as Eq. (9).

$$p(k+1)=f\left(p(k),\ldots,p\left(k-n_{\xi}\right),\upsilon(k),\ldots,\upsilon\left(k-n_{q}\right)\right) \quad (9)$$

In Eq. (10), $\upsilon(k)\in R^{6}$ and $p(k)\in R^{2}$ are the inputs and outputs of the robotic arm system at moment k, respectively, n_p and n_υ are random integers, and $f(\ldots): R^{n_u+n_y+2}\mapsto R$ is a nonlinear vector function. An equivalent dynamic linearized data model can be used to replace the predictive control system of the robotic arm. Its expression is as Eq. (10).

$$\Delta p(k+1)=\varphi(k)\Delta\upsilon(k) \quad (10)$$

In Eq. (11), $\varphi(k)\in R^{2\times 6}$ is the pseudo partial derivative of the system.

Then the expression of the output prediction equation for one step forward is as Eq. (11).

$$p(k+1)=p(k)+\varphi_c(k)\Delta\upsilon(k) \quad (11)$$

If $\Delta V(k+j-1)=0, j>N_\upsilon$ exists, then the prediction equation of the robotic arm servo system whose expression is as Eq. (12).

$$p_N(k+1)=E(k)p(k)+A_1(k)\Delta V_N(k) \quad (12)$$

In Eq. (12), N_υ is the control time domain constant. The control system of a robotic arm servo system aims to achieve precise, stable and responsive motion control, while guaranteeing the safety and reliability of the system. Important components including the robotic arm's dynamic behavior, the tasks that need to be completed, and the operating environment must all be carefully taken into account while building and designing such a system. When a certain criterion function is input to the system, the control law of the system can be obtained, the expression of which is as Eq. (13).

$$\begin{cases} I=\sum_{i=1}^{N}\left(p(k+1)-p^{*}(k+i)\right)^{2}+\lambda\sum_{j=0}^{N_\upsilon-1}\Delta\upsilon^{2}(k+j) \\ \Delta U_N(k)=\left[A_1^T(k)A_1(k)+\lambda I\right]^{-1}A_1^T(k)\left[P_N^{*}(k+1)-E(k)p(k)\right] \end{cases} \quad (13)$$

In Eq. (13), $p^{*}(k+i)$ is the desired position of the servo system at the moment k+1, and $\lambda>0$ is the weighting factor. Then the control input of the servo system at this moment has the expression as Eq. (14).

$$\upsilon(k)=\upsilon(k+1)+g^{T}\Delta U_{N_\upsilon}(k) \quad (14)$$

In Eq. (14), $g=[1,0,\cdots,0]^{T}$. Therefore, when $N_\upsilon=1$, the input of the robotic arm servo system reaches as Eq. (15).

$$\upsilon(k)=\upsilon(k+1)+\frac{1}{\varphi_c^{2}+\lambda/N}\frac{1}{N}\left[\varphi_c(k)\sum_{i=1}^{N}\left(p_N^{*}(k+i)-E(k)p(k)\right)\right] \quad (15)$$

After the input function of the system is known, estimation algorithms as well as forecasting algorithms need to be designed since it contains the unknown system PDD $\varphi_c(k),\varphi_c(k+1),\cdots,\varphi_c(k+N_\upsilon-1)$. For the prediction of PDD, there are self-correction methods and multilayer recursive order prediction methods, etc. The self-correction method is a strategy to dynamically adjust and optimize the control system, which allows the system to automatically adjust its control parameters during operation to adapt to changes in the environment or changes in the performance of the system, and its flowchart is as Fig. 3.

As indicated in Fig. 3, the adaptive controller can automatically adjust the control parameters according to the real-time feedback of the system performance to adapt to the changes in the dynamic characteristics of the system. While the multilayer recursive forecasting method is to decompose the complex forecasting problem into multiple levels or stages, each level or stage solves a part of the forecasting problem, which is usually used to deal with systems with multiple related variables and complex dynamics, and its flowchart is as Fig. 4.

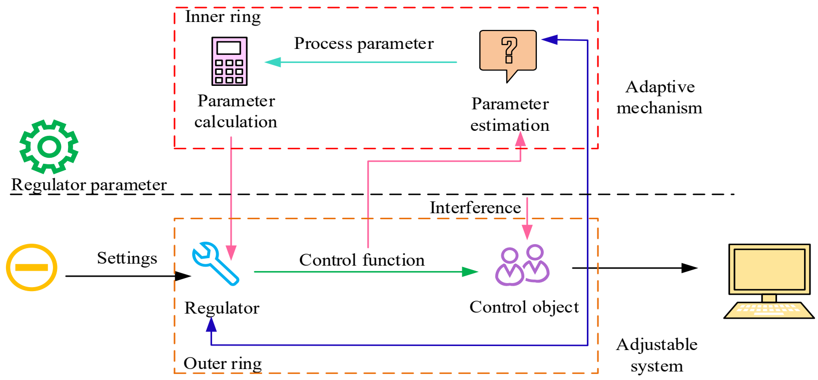


Figure 3: Self-correcting control system structure diagram

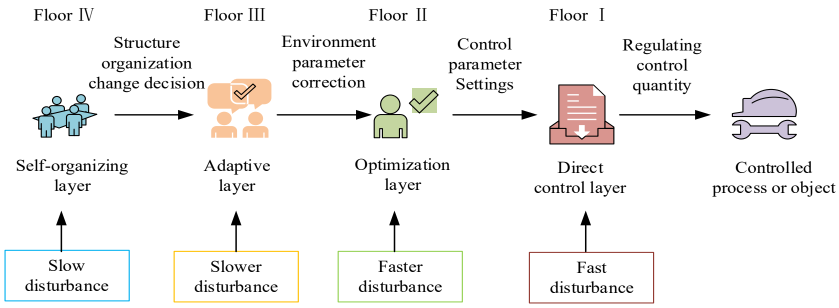


Figure 4: Structure diagram of multilevel hierarchical prediction method

According to Fig. 4, the multilayer recursive forecasting method's advantage is its ability to handle multivariate and dynamic relationships in complex systems while enabling the application of various forecasting models and techniques at various levels, particularly for those systems that are challenging for a single model or method to accurately predict. Comprehensively comparing the degree of fitness of the two to the system, the multilayer recursive order prediction method is chosen to predict the position parameters of the robotic arm servo system. An estimation sequence satisfying the autoregressive model is performed for it. Combining the parameter estimation algorithm of the servo system, the control algorithm and the parameter forecasting algorithm, the trajectory prediction control scheme of the robotic arm performing the task in the control time domain $N_v = 1$ can be derived. The expression is as Eq. (16).

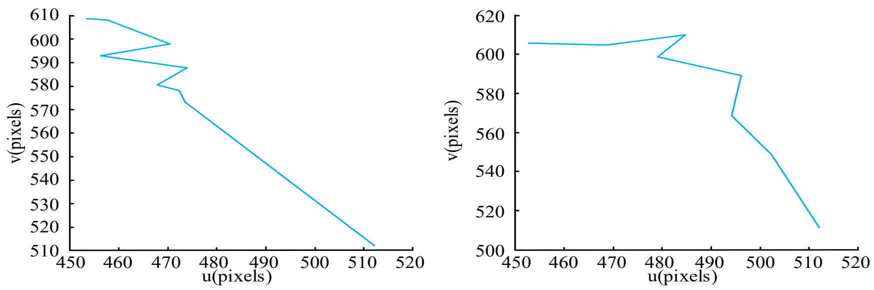
$$\begin{cases} \upsilon(k) = \upsilon(k+1) + \dfrac{1}{\varphi_c^2 + \lambda / N} \dfrac{1}{N}\left[\varphi_c(k)\sum_{i=1}^{N}\left(p_N^*(k+i) - p_m(k)\right)\right] \\ \hat{\varphi}_c(k) = \hat{\varphi}_c(k+1) + \dfrac{\eta\Delta\upsilon(k-1)}{\mu + \Delta\upsilon(k-1)}\left[\Delta p(k) - \hat{\varphi}_c(k-1)\Delta\upsilon(k-1)\right] \end{cases} \tag{16}$$

4. Performance Validation of Model-free Adaptive Visual Servo Control and Forecasting

The study explores the construction method of the model-free adaptive control system and utilizes the model to generate the input and output data of the visual servo system, and the proposed algorithm is tested by simulation simulation through MATLAB's Robotics Toolbox, and the efficacy of the study's proposed predictive control system for the robotic arm is examined through these training data. The visual servo positioning performance of the PUMA560 robotic arm was first verified by setting the position of the target object feature point relative to the base scale system of the robotic arm as $[0.3 \quad 0.1 \quad 2.5]^T$ m, the desired image plane position as $[512 \quad 512]^T$, and the movement time of the robotic arm as 20 s. The starting joint position of the robotic arm is $[0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0]$, the sampling period is 0.05 s, the initial velocity of the end-effector of the robotic arm is $\upsilon(1) = \upsilon(2) = [0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0]^T$, and the initial position of the feature point of the target object is $p(1) = p(2) = [453 \quad 608]^T$. The image trajectories of the original servo system and the servo system with no model adaption are compared, and the experimental results are as Fig. 5.

From Fig. 5(a), it is obvious that both systems are able to complete the visual servo positioning work, but the positioning of the unoptimized system needs to be adjusted in detail many times. From Fig. 5(b), it can be get that the motion trajectory of the servo system with no model adaptive is more silky smooth than the unoptimized system, and it can adjust the position quickly in order to arrive at the target desired position quickly. The results of the positioning error comparison between the two are displayed in Fig. 6.

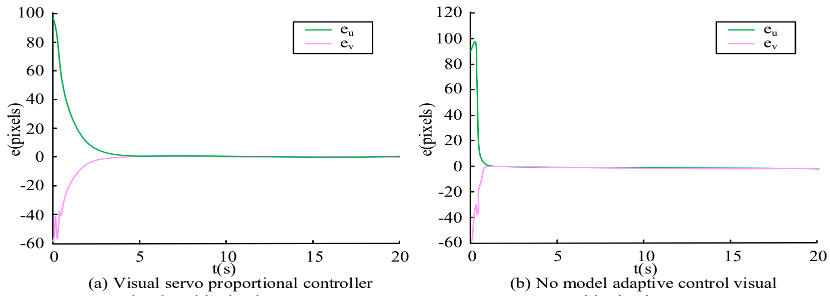
From Fig. 6(a), it can be seen that before 5s, the unoptimized control system has a certain image error, and after 5s the image errors e_u and e_v basically disappear, and the state of the control system of the robotic arm is gradually stabilized. From Fig. 6(b), it can be get that before 1.5s, the model-less adaptive servo system still has image error, but the system corrects the error faster, so that the image error is completely eliminated after 1.5s, and the visual servo system reaches a stable state. Taken together, the proposed model-free adaptive servo system possesses better performance and stability in visual control compared to the unoptimized system. In addition, the trajectory tracking performance of the predictive control system of the robotic arm is then simulated and experimented. The center of circle $\begin{bmatrix} 0 & 0.1 & 2.5 \end{bmatrix}$ is set as the working radius of the robotic arm is 0.4m, and the trajectory of the tracked task target is a uniform circular motion. The comparison graph of the tracking results of the robotic arm is as Fig. 7.



(a) Visual servo proportional controller visual positioning image trajectory

(b) Model-free adaptive control of visual positioning image trajectory

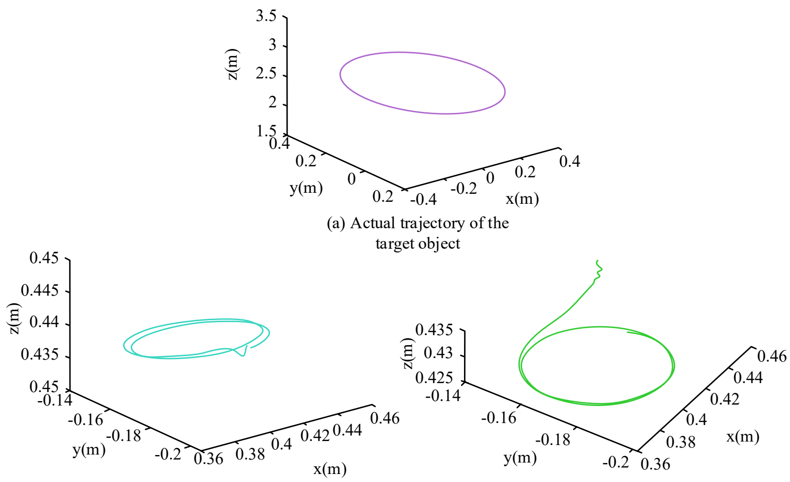
Figure 5: Visual positioning image track comparison diagram



(a) Visual servo proportional controller visual positioning image error

(b) No model adaptive control visual positioning image error

Figure 6: Visual servo positioning error result comparison diagram

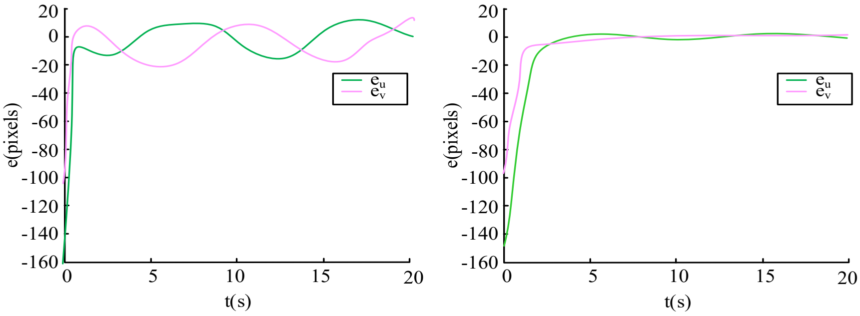


(a) Actual trajectory of the target object

(b) Based on classical servo proportional control visual servo control manipulator end trajectory

(c) End trajectory of manipulator based on model-free adaptive visual servo control

Figure 7: Comparison of the tracking results of the robot arm



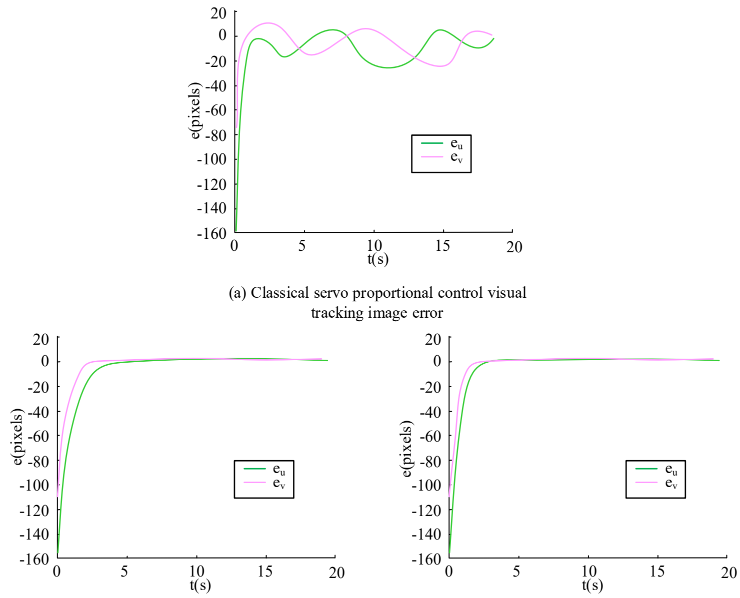
(a) Classical servo proportional control visual tracking image error

(b) No model adaptive control visual tracking image error

Figure 8: Error comparison diagram of two visual tracking images

From Fig. 7(a), the tracking effect of the classical visual servo-proportional control system is demonstrated when guiding the robotic arm to track a target object that is moving in a circular motion at a constant speed. From Fig. 7(b), the robotic arm predictive control system is able to complete the intended target trajectory when tracking a circular trajectory under the control of the unoptimized system. From Fig. 7(c), it can be learned that the proposed robotic arm predictive control system of the study is able to accomplish the task efficiently and accurately. The image errors tracked by the two methods are compared and the results are as Fig. 8.

As can be seen from Fig. 8(a), the image errors e_u and e_v of the tracking trajectory of the unoptimized visual servo system both fluctuate considerably and can never be stabilized around 0, and its image error basically maintains fluctuating within the range of ±20 pixels. From Fig. 8(b), it can be seen that the image of the tracking trajectory of the model-less adaptive servo system on the target has a certain image error before about 1.7s, and the image error of both e_u and e_v is close to about 0 after 1.7s, which means that the servo system gradually converges to a stable state. Taken together, the model-free adaptive robotic arm control method based on model-free adaptive has a better trajectory tracking convergence speed, can enter the task trajectory route more quickly, and the image error is smaller, which is better for the task completion. After verifying that the model-free adaptive image processing based method is effective, the performance of the servo system incorporating the forecasting algorithm is then verified. The desired pixel coordinates of the target object's feature points are set to be $p^* = \begin{bmatrix} 621 & 721 \end{bmatrix}^T$. The target tracking task is carried out with the unoptimized servo system, the servo system with model-free adaptive control, and the servo system with model-free adaptive control incorporating the forecasting algorithm respectively, and in the absence of perturbation, their comparison of the image tracking error results is as Fig. 9.



(a) Classical servo proportional control visual tracking image error

(b) No model adaptive predictive control visual servo tracking error

(c) Model-free adaptive predictive control visual servo tracking error combined with prediction algorithm

Figure 9: Comparison of image tracking error results of three methods

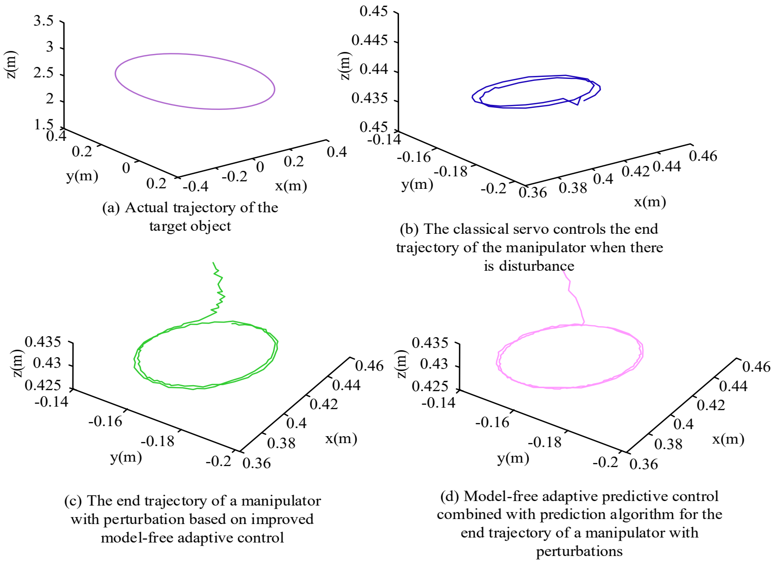


Figure 10: Comparison diagram of the end trajectory results of the manipulator with perturbations

From Fig. 9(a), it can be seen that in the case of no perturbation, the robotic arm controlled by the unoptimized servo system fluctuates a lot in the image error in the task of tracking the target object, and the system is always in an unstable state, and the task is accomplished with a general effect. From Fig. 9(b), it can be seen that the model-free adaptive control method enables the servo system of the robotic arm to complete the tracking task better, and the tracking error of the image is basically 0 after 3s, i.e., the control system enters a stable state. When controlling the robotic arm for target tracking, Fig. 9(c) shows that the optimized servo system in conjunction with the forecasting algorithm can converge the trajectory well and deal with the image error very quickly. After approximately 1.8 seconds, the robotic arm enters the state of 0 image error, demonstrating that its performance is superior to that of both the model-free adaptive control system and the unoptimized control system. Based on this experimental condition, the external interference with amplitude of $\left[-5 \quad 5\right]$ pexels is added to the output band of the predictive control system, and the externally related parameters are set to $\eta = 0.2, \mu = 0.005, \rho = 0.005, \lambda = 100$, $N_\upsilon = 1$, $N_\upsilon = 3$. The experimental results of trajectory tracking are as Fig. 10.

From Fig. 10(a), it can be get that the path is the actual trajectory of the target. From Fig. 10(b), it is obvious that the unoptimized servo system controlled the robotic arm travel path has a large gap with the target path and the tracking effect is relatively unsatisfactory. In Fig. 10(c), the servo system optimized by the model-free adaptive model controls the robotic arm for target tracking, and the tracking result graph shows a more drastic path jitter. In Fig. 10(d), the model-free adaptive control method combined with the forecasting algorithm is able to pre-process the image and adjust the tracking path of the robotic arm in time in order to reach the expected target path in a very short period of time, and this control method has the fastest convergence speed, and it is also the one that can basically maintain the stable state in the plane coordinate system of the image. On this basis, to further evaluate the performance of the robotic arm controlled by the three different methods in terms of end error convergence, the end error convergence curves of the robotic arm controlled by the three methods are further compared, and the results are as in Fig. 11.

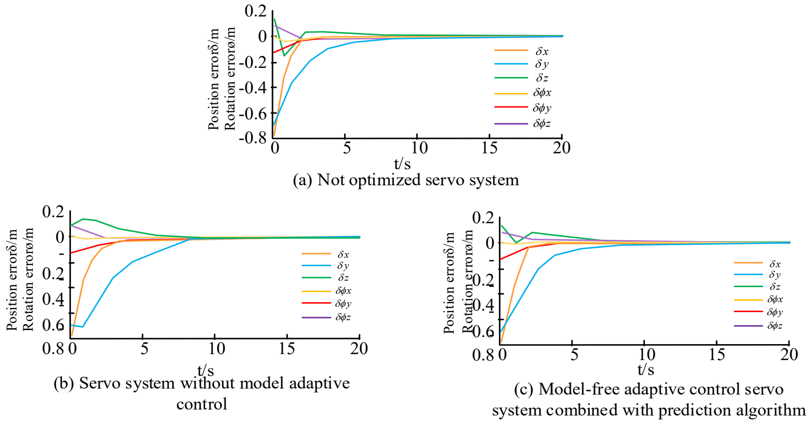
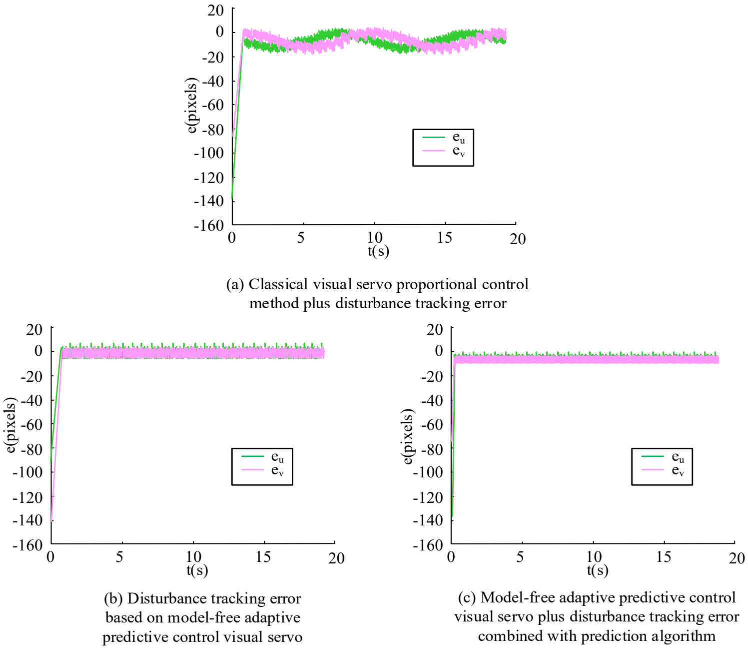


Figure 11: Comparison of the end error convergence curves of the three methods



(a) Classical visual servo proportional control method plus disturbance tracking error

(b) Disturbance tracking error based on model-free adaptive predictive control visual servo

(c) Model-free adaptive predictive control visual servo plus disturbance tracking error combined with prediction algorithm

Figure 12: Image tracking error comparison diagram of three methods

From Fig. 11(a), the position and orientation of the robotic arm end controlled by the unoptimized initial system change rapidly, but the end error fluctuates more, especially more significantly in the position change. From Fig. 11(b), it can be demonstrated that, in contrast, the end error curve of the robotic arm controlled by the model-free adaptive approach is smoother. According to Fig. 11(c), combining the forecasting algorithm based on the model-free adaptive control method predicts and adjusts the image's position beforehand, resulting in a much smaller change in the image's end position error and angular error. This further optimizes the shortcomings of the first two methods in the end control and produces better performance. In addition, a comparison graph of the image tracking errors of the three control methods after adding perturbation is as Fig. 12.

From Fig. 12(a), the robotic arm controlled by the unoptimized system has a tracking error difference amplitude floating around ±10 pixel points when performing the target tracking task, i.e., the tracking results are not very stable. From Fig. 12(b), it can be known that the proposed method of the study is able to track the desired trajectory and follow the expected path for the task better, and the tracking error is ±7 pixel points floating up and down. From Fig. 12(c), it is shown that adding the forecasting algorithm to the control system further improves the performance of the system tracking and the tracking error is only around ±5 pixel points. Taken together, the model-free adaptive control method that combines the forecasting algorithm and image processing possesses higher performance and smaller error when controlling the robotic arm for the task.

5. Conclusions

The research is based on the image processing method, which uses the difference between the image coordinates of the target object's center and the desired coordinates as the feedback signal. The multilayer recursive order prediction method is then combined in the prediction algorithm to create a model-free adaptive prediction controller, which is compared through simulation experiments. This approach addressed the issues with the traditional prediction system of the PUMA560 robotic arm, including the complexity of the control system model and the challenge of obtaining the camera calibration parameters. The experimental results demonstrated that the model-free adaptive method based on image processing can reach the desired route faster in the tracking task, and the system is in a stable state after 1.5 s, which is 3.5 s higher than the stabilization time of the traditional method. After combining with the forecasting algorithm, the system was able to control the robotic arm to track the target in a better way, and the tracking error is about ±10 pixel points compared to the tracking error of about ±10 pixel points for the unoptimized system and about ±7 pixel points for the system, which does not incorporate the forecasting algorithm. Compared with the tracking error of ±10 pixels for the unoptimized system and ±7 pixels for the system without the prediction algorithm, the tracking error of the final optimized method was about ±5 pixels. In summary, the model-free adaptive control approach that included forecasting and image processing algorithms was more effective, performed better, and had a lower error rate when it came to robotic arm predictive control.

However, the use of a single vision sensor has some limitations, and future research is expected to improve the flexibility and autonomy of the robotic arm system by combining the vision sensor with additional sensors, allowing it to better adapt to the complex external environment.

Reference

[1] Cheng B, Wu W, Tao D. Random cropping ensemble neural network for image classification in a robotic arm grasping system. IEEE Transactions on Instrumentation and Measurement, 2020, 69(9): 6795-6806.

[2] Kuttler C, Nussbaum A K, Dick T P. An algorithm for the prediction of proteasomal cleavages. Journal of molecular biology, 2000, 298(3): 417-429.

[3] Ranganathan G. An economical robotic armplaying chess using visual servoing. Journal of Innovative Image Processing (JIIP), 2020, 2(03): 141-146.

[4] Fusco F, Kermorgant O, Martinet P. Integrating features acceleration in visual predictive control. IEEE Robotics and Automation Letters, 2020, 5(4): 5197-5204.

[5] Yuan W, Liu Y H, Su C Y. Whole-body control of an autonomous mobile manipulator using model predictive control and adaptive fuzzy technique. IEEE Transactions on Fuzzy Systems, 2022, 31(3): 799-809.

[6] Heshmati-Alamdari S, Karras G C, Kyriakopoulos K J. A predictive control approach for cooperative transportation by multiple underwater vehicle manipulator systems. IEEE Transactions on Control Systems Technology, 2021, 30(3): 917-930.

[7] Ngugi L C, Abelwahab M, Abo-Zahhad M. Recent advances in image processing techniques for automated leaf pest and disease recognition–A review. Information processing in agriculture, 2021, 8(1):27-51.

[8] Cong V D. Visual servoing control of 4-DOF palletizing robotic arm for vision based sorting robot system. IJIDeM: International Journal on Interactive Design and Manufacturing, 2023, 17(2):717-728.

[9] Taira T, Zheng W, Miyagi T, et al. Visual Inspection System for Piston-Ring Parts via Machine Learning.IEEJ Transactions on Industry Applications, 2023, 143(2):101-105.

[10] Cheng B, Wu W, Tao D. Random cropping ensemble neural network for image classification in a robotic arm grasping system. IEEE Transactions on Instrumentation and Measurement, 2020, 69(9): 6795-6806.

[11] Huang D, Gan B. Intelligent Garbage Recycling: Design and Implementation Exploration of Automatic Classification System.Meteorological and Environmental Research, 2024, 15(2):37-39.

[12] Zahaf A, Bououden S, Chadli M. Robust fault tolerant optimal predictive control of hybrid actuators with time-varying delay for industrial robot arm. Asian Journal of Control, 2022, 24(1): 1-15.

[13] Yuan W, Liu Y H, Su C Y. Whole-body control of an autonomous mobile manipulator using model predictive control and adaptive fuzzy technique. IEEE Transactions on Fuzzy Systems, 2022, 31(3): 799-809.

[14] Tika A, Gafur N, Yfantis V. Optimal scheduling and model predictive control for trajectory planning of cooperative robot manipulators. IFAC-PapersOnLine, 2020, 53(2): 9080-9086.

[15] Pankert J, Hutter M. Perceptive model predictive control for continuous mobile manipulation. IEEE Robotics and Automation Letters, 2020, 5(4): 6177-6184.

[16] Huang Y, Hofer M, D'Andrea R. Offset-free model predictive control: A ball catching application with a spherical soft robotic arm//2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). IEEE, 2021, 7(2) 563-570.

[17] Ciria A, Schillaci G, Pezzulo G. Predictive processing in cognitive robotics: a review. Neural Computation, 2021, 33(5): 1402-1432.

[18] Fusco F, Kermorgant O, Martinet P. Integrating features acceleration in visual predictive control. IEEE Robotics and Automation Letters, 2020, 5(4): 5197-5204.