Journal of Systems Engineering and Electronics
Vol. 35, No. 2, April 2024, pp.473 – 484

Manipulator tracking technology based on FSRUKF

SHI Guoqing[1], ZHANG Boyan[1], ZHANG Jiandong[1,*], YANG Qiming[1], HUANG Xiaofeng[1], QUE Jianyao[1], PU Junwei[1], and GENG Xiutang[2]

1. School of Electronics and Information, Northwestern Polytechnical University, Xi'an 710072, China;
2. Northwest Institute of Mechanical & Electrical Engineering, Xianyang 712000, China

Abstract: Aiming at the shortcoming that the traditional industrial manipulator using off-line programming cannot change along with the change of external environment, the key technologies such as machine vision and manipulator control are studied, and a complete manipulator vision tracking system is designed. Firstly, Denavit-Hartenberg (D-H) parameters method is used to construct the model of the manipulator and analyze the forward and inverse kinematics equations of the manipulator. At the same time, a binocular camera is used to obtain the three-dimensional position of the target. Secondly, in order to make the manipulator track the target more accurately, the fuzzy adaptive square root unscented Kalman filter (FSRUKF) is proposed to estimate the target state. Finally, the manipulator tracking system is built by using the position-based visual servo. The simulation experiments show that FSRUKF converges faster and with less error than the square root unscented Kalman filter (SRUKF), which meets the application requirements of the manipulator tracking system, and basically meets the application requirements of the manipulator tracking system in the practical experiments.

Keywords: square root unscented Kalman filter (SRUKF), fuzzy inference, manipulator, visual servo.

DOI: 10.23919/JSEE.2024.000009

1. Introduction

Since the world's first industrial robot was introduced in 1959, robotics research has made tremendous progress and has been widely used in manufacturing, service, defense security, and space exploration [1]. In traditional industrial production lines, the operational movements of the manipulator are usually planned by means of offline programming or teach-in programming, but those methods require that the motion state of the manipulator's operating object must be completely known, and once the motion of the operating object changes, the operation of the manipulator will deviate, leading to operational failure. With the demand for more diverse functions of the manipulator in the automated production process, manipulators are more often faced with operating objects with changing motion states, which poses a challenge to the traditional method of manipulator operation [2]. In recent years, with the improvement of hardware computing power, machine vision technology has developed rapidly and has been widely used in the industrial field, which greatly improves the efficiency of industrial production. It is a general trend to combine manipulator control with machine vision technology to expand the application scenarios of manipulators. The combination of machine vision and manipulators solves the problem of insufficient autonomy and intelligence in complex working environments and tasks, and improves the flexibility of manipulators [3].

In the area of manipulator tracking combined with machine vision, researchers have adopted many methods and achieved many impressive results, such as Kalman filtering (KF) [4], extended KF (EKF) [5], unscented KF (UKF) [6], particle filter [7], machine learning, and artificial intelligence [8]. Frese et al. [9,10] adopted EKF for tracking random motion targets based on machine vision, but the tracking effect of EKF is not satisfactory for strongly nonlinear motion such as random motion. Paing et al. [11] improved the estimation of object trajectories via KF for the capture of flying objects and proposed a least squares fit to accurately predict the capture time, position, and velocity of the manipulator. An improved UKF algorithm was introduced and used in [12]. Compared with the estimation algorithm of the standard UKF, the new algorithm effectively reduces the state estimation error and improves the tracking stability and the tracking accuracy. References [13,14] showed that the particle-based filter can track the strongly nonlinear targets better, but its accuracy is proportional to the number

Manuscript received November 09, 2022.
*Corresponding author.
This work was supported by Natural Science Basic Research Program of Shaanxi (2022JQ-593) and Key Research and Development Program of Shaanxi (2022GY-089).

of particles, and the conflict between accuracy and real-time performance is still a direction worthy of further research. In [15], a machine learning algorithm was used to learn the trajectory of a throwing object. By predicting the target trajectory and classifying it, the corresponding optimal capture action is selected and the capture of a high-speed throwing target is achieved, but the method has some limitations for the capture of random motion targets. Luo et al. [16] proposed a manipulator capture method by predicting the trajectory of a moving target in camera space. This method, although capable of fast-tracking targets, requires a large amount of data from specific motion models for learning and is not universal.

As the research progressed, researchers introduced the concept of "adaptive" into various filtering algorithms to achieve better target tracking results, which is implemented in different aspects. Qu et al. [17] introduced the adaptive fading KF (AFKF) algorithm and proposed the calculation of the forgetting factor for target tracking in the case of temporary occlusion of moving target image features, which applies to the extraction of non-robust target features for robot-operated vision tracking tasks. Li et al. [18] introduced and used an improved UKF to track moving targets for application on radar systems, where the scale factor is adaptively chosen to have a simplex sampling with the minimum bias. The algorithm reduces the computational effort and improves the accuracy of tracking the target. Zhang et al. [19] proposed an adaptive multi-level resampling-based particle filtering (ARPF) algorithm, which divides the particle resampling weight space algorithm into multiple levels and resamples the diversity of resampled particles through a rank entropy measure to improve the filtering accuracy.

With the rising of artificial intelligence technology in recent years, it has been widely used in various industries. The methods such as deep learning, reinforcement learning, and deep reinforcement learning have been applied to manipulator arm grasping and have achieved positive results.

The synthesis of the above references shows that there are limitations in the currently used tracking methods. EKF is not ideal for tracking random motion targets, while the pre-defined equations of motion approach will make the error accumulation impossible to predict the tracking for a long time, and the particle filtering approach cannot balance accuracy and real-time. While the introduction of "fuzzy" and "adaptive" has achieved better results in other target tracking.

In this paper, fuzzy adaptive square root UKF (FSRUKF) is proposed to solve the problems that KF is not applicable to non-linear motion and particle filtering is computationally intensive. The fuzzy inference system is applied to improve suqare root UKF (SRUKF), and the adaptive adjustment of SRUKF measurement noise variance matrix is achieved, which improves the estimation accuracy of the traditional algorithm. And the position-based visual servo is utilized to overcome the problem of conducting random motion targets that are difficult to track for a long time. The robot arm tracking system designed in this paper is able to complete the target tracking task through experimental verification.

This paper is organized as follows: Section 2 presents the modeling and kinematic analysis of the manipulator of the DOBOT four-degree-of-freedom manipulator platform. Section 3 introduces the imaging principle of the camera and builds the imaging model. Section 4 introduces SRUKF and FSRUKF respectively, and compares the two algorithms. Section 5 designs the experimental protocols using SRUKF and FSRUKF proposed in this paper, respectively, and conducts the practical experimental comparisons. Finally, conclusions are summarized in Section 6 .

2. Manipulator modeling and kinematic analysis

2.1 Manipulator modeling

A manipulator is essentially an open kinematic chain composed of jointed rods. According to the standard Denavit-Hartenberg (D-H) modeling method [20], four parameters are used to describe adjacent links and joints, where the position relationship between adjacent joint axes is described by the link length α_i and the link torsion angle α_i, and the position relationship between adjacent rods is described by the link offset d_i and the link rotation angle θ_i. The DOBOT four-degree-of-freedom manipulator platform is used in this paper. The structure of the manipulator and the joint coordinate system are shown in Fig. 1.

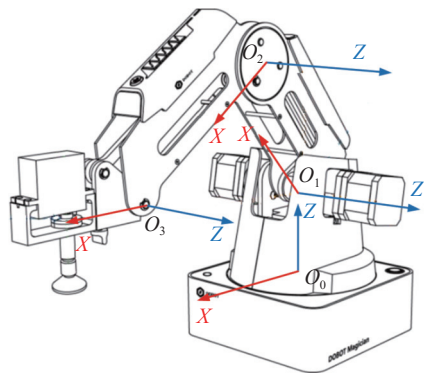


Fig. 1 DOBOT manipulator structure and joint coordinate system

According to the definition of each parameter in the D-H modeling method, the D-H parameter table of the DOBOT manipulator is obtained as shown in Table 1.

Table 1 DOBOT manipulator D-H parameter table

Link serial number	Link rotation angle θ_i/(°)	Link offset d_i/mm	Link length a_i/mm	Link torsion angle α_i/(°)
1	θ_1	138	0	−90
2	θ_2	0	135	0
3	θ_3	0	147	0
4	θ_4	0	0	0

The coordinate transformation between adjacent links of the manipulator can be represented by a homogeneous coordinate transformation matrix, as follows:

$$ {}^{i-1}_{i}\boldsymbol{T} = \begin{bmatrix} \cos\theta_i & -\sin\theta_i\cos\alpha_i & \sin\theta_i\sin\alpha_i & a_i\cos\theta_i \\ \sin\theta_i & \cos\theta_i\cos\alpha_i & -\cos\theta_i\sin\alpha_i & a_i\sin\theta_i \\ 0 & \sin\alpha_i & \cos\alpha_i & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix}. \tag{1} $$

The transformation relationship from the base coordinate system of the manipulator to the end-effector coordinate system is obtained by multiplying the adjacent homogeneous coordinate transformation matrices together, which is represented by the positive kinematic equation of the manipulator, as follows:

$$ {}^{0}_{4}\boldsymbol{T} = {}^{0}_{1}\boldsymbol{T}\cdot{}^{1}_{2}\boldsymbol{T}\cdot{}^{2}_{3}\boldsymbol{T}\cdot{}^{3}_{4}\boldsymbol{T} = \begin{bmatrix} c_1 & 0 & -s_1 & c_1(a_3c_{23}+a_2c_2) \\ s_1 & 0 & c_1 & s_1(a_3c_{23}+a_2c_2) \\ 0 & -1 & 0 & d_1-a_2s_2-a_3s_{23} \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{2} $$

where $c_1=\cos\theta_1$, $s_1=\sin\theta_1$, $c_{23}=\cos(\theta_2+\theta_3)$, $s_{23}=\sin(\theta_2+\theta_3)$.

2.2 Inverse kinematic equation

Inverse kinematics can calculate the value of each joint angle of the manipulator for known end-effector poses. The form of the end-effector's pose transformation matrix concerning the base is known, and it is assumed that the desired pose of the end-effector relative to the base is known and can be expressed as

$$ \boldsymbol{T}_{\text{End}} = \begin{bmatrix} a_x & b_x & c_x & l_x \\ a_y & b_y & c_y & l_y \\ a_z & b_z & c_z & l_z \\ 0 & 0 & 0 & 1 \end{bmatrix}. \tag{3} $$

Let

$$ {}^{0}_{4}\boldsymbol{T} = {}^{0}_{1}\boldsymbol{T}\cdot{}^{1}_{2}\boldsymbol{T}\cdot{}^{2}_{3}\boldsymbol{T}\cdot{}^{3}_{4}\boldsymbol{T} = \boldsymbol{T}_{\text{End}}. \tag{4} $$

There is

$$ \begin{bmatrix} c_1 & 0 & -s_1 & c_1(a_3c_{23}+a_2c_2) \\ s_1 & 0 & c_1 & s_1(a_3c_{23}+a_2c_2) \\ 0 & -1 & 0 & d_1-a_2s_2-a_3s_{23} \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} a_x & b_x & c_x & l_x \\ a_y & b_y & c_y & l_y \\ a_z & b_z & c_z & l_z \\ 0 & 0 & 0 & 1 \end{bmatrix}.. \tag{5} $$

Then the value of each joint angle of the manipulator is calculated by (6), (7), (8), and (9) as follows:

$$ \theta_1 = \tan^{-1}(l_y/l_x), \tag{6} $$

$$ \cos\theta_3 = \frac{(l_xc_1+l_ys_1)^2+(d_1-l_z)^2-a_2^2-a_3^2}{2\cdot a_2a_3}, \tag{7} $$

$$ \cos\theta_2 = \frac{(d_1-l_z)\cdot a_3s_3+(l_xc_1+l_ys_1)(a_3c_3+a_2)}{(a_3c_3+a_2)^2+(a_3s_3)^2}, \tag{8} $$

$$ \theta_4 = -(\theta_2+\theta_3). \tag{9} $$

3. Camera imaging principles and binocular camera distance measurement

When a camera captures a subject, the light from the subject first enters the camera through the camera aperture, where the camera's internal light sensor captures the light and converts the light signal from the subject into a digital signal for storage. The image acquired by the camera can then be read by a computer to obtain the stored data. This structural model can be approximated as a pinhole model [21]. The camera imaging model is shown in Fig. 2.

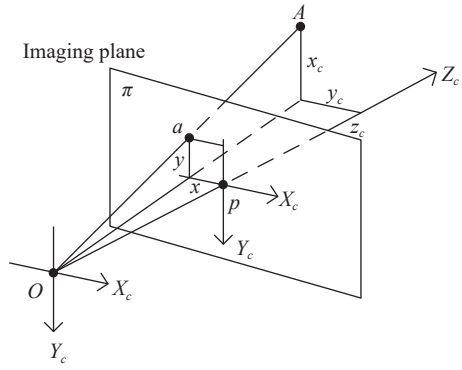


Fig. 2 Camera imaging model

Among them, O is the origin of the camera coordinate system, OA is the optical axis of the camera, A is the feature point on the target surface, and α is the mapping point of the feature point on the imaging plane.

In order to accurately describe the three-dimensional (3D) coordinates of the target in space, the following conventions are applied to the coordinate system used in the

camera imaging process. $O_wX_wY_wZ_w$ is the world coordinate system, which is used to describe the location of the target in real space; $O_cX_cY_cZ_c$ is the camera coordinate system, which is used to describe the spatial location of the target object relative to the camera; O_1XY is the image coordinate system, whose origin is located on the camera imaging plane center; O_0uv is the pixel coordinate system, and its origin is located in the upper left corner of the camera imaging plane. The relationship of each coordinate system is shown in Fig. 3.

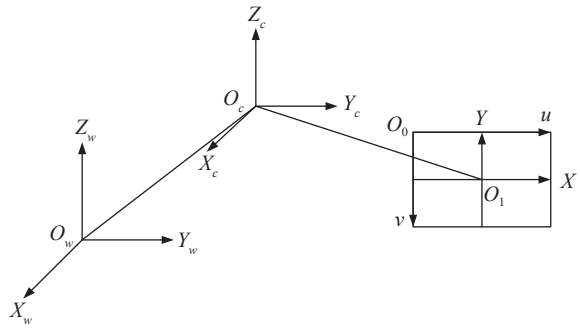


Fig. 3 Relationship between coordinate systems

The conversion relationship between the world coordinate system and the camera coordinate system can be expressed as

$$\begin{bmatrix} x_c \\ y_c \\ z_c \end{bmatrix} = \boldsymbol{R}\begin{bmatrix} x_w \\ y_w \\ z_w \end{bmatrix} + \boldsymbol{t} \tag{10}$$

where (x_c, y_c, z_c) is the coordinate of the target in the camera coordinate system, (x_w, y_w, z_w) is the coordinate of the target in the world coordinate system, $\boldsymbol{R}$ is the rotation matrix between the world coordinate system and the camera coordinate system, and $\boldsymbol{t}$ is the relative position vector between the camera coordinate system and the world coordinate system.

The conversion relationship between the coordinates in the image coordinate system and those in the camera coordinate system can be expressed as follows:

$$x = f\frac{x_c}{z_c}, \tag{11}$$

$$y = f\frac{y_c}{z_c}, \tag{12}$$

where x and y are the coordinates of the target point in the image coordinate system, x_c, y_c, z_c are the coordinates of the target point in the camera coordinate system, and f is the camera focal length.

The conversion relationship between the image coordinate system and the pixel coordinate system can be expressed as follows:

$$u = \frac{x}{\mathrm{dx}} + u_0, \tag{13}$$

$$v = \frac{y}{\mathrm{dy}} + v_0, \tag{14}$$

where u and v are the coordinates in the pixel coordinate system, u_0 and v_0 are the coordinates of the image coordinate origin in the pixel coordinate system, dx and dy represent the size of each pixel on each row and column respectively, and x, y are the coordinates in the image coordinate system.

The world coordinate system to pixel coordinate system conversion relationship, namely the camera linear imaging model, is obtained by associating the conversion relationships between the coordinate systems, as follows:

$$z_c\begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \begin{bmatrix} \frac{1}{\mathrm{dx}} & 0 & u_0 \\ 0 & \frac{1}{\mathrm{dy}} & v_0 \\ 0 & 0 & 1 \end{bmatrix}\begin{bmatrix} f & 0 & 0 & 0 \\ 0 & f & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}\cdot$$

$$\begin{bmatrix} R & t \\ 0 & 1 \end{bmatrix}\begin{bmatrix} x_w \\ y_w \\ z_w \\ 1 \end{bmatrix} = \begin{bmatrix} f & 0 & u_0 & 0 \\ 0 & f & v_0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}\begin{bmatrix} R & t \\ 0 & 1 \end{bmatrix}\begin{bmatrix} x_w \\ y_w \\ z_w \\ 1 \end{bmatrix} \tag{15}$$

where $\begin{bmatrix} f & 0 & u_0 & 0 \\ 0 & f & v_0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}$ is called the internal parameters of the camera, and $\begin{bmatrix} R & t \\ 0 & 1 \end{bmatrix}$ is called as the external parameters of the camera.

In this paper, the ZED binocular camera from Stereolabs is used as a vision sensor to obtain the coordinates of a point in 3D space by capturing the image of the object under test using two left and right cameras simultaneously [22] to obtain the distance z_c of the target in the depth direction to the camera. The basic principle of binocular vision ranging is shown in Fig. 4.

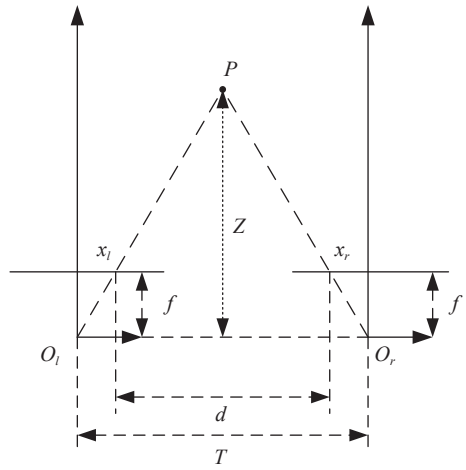


Fig. 4 Binocular camera distance measuring principle

Given that point P is a point in space, its position on the left and right imaging planes changes continuously as point P moves along the optical axis of the camera. The parallax d of point P is the distance between the x-coordinates of point P in the left and right image planes. The distance Z of the point P in the depth direction is inversely proportional to the parallax d, which can be deviated from the center of the projection point on the left and right imaging planes by subtracting the center distance T of the optical axis of the two cameras. The distance T between the centers of the two camera axes can be obtained by camera calibration.

4. FSRUKF

4.1 SRUKF

UKF solves the problem that KF cannot be applied to nonlinear systems, and offers higher estimation accuracy than EKF, as well as being easier to implement in engineering than EKF. The square root of the state covariance matrix needs to be calculated at each iteration of the Sigma sampling point update, which is the largest time overhead in the computation. In addition, as the filtering proceeds, the presence of computer rounding errors will cause the error covariance matrix and the state covariance matrix to lose their positivity, leading to filtering failure. In order to improve the efficiency and numerical stability of the filtering calculation, a square root form of UKF is derived [23].

The recursive formula of SRUKF in this paper is as follows [24].

Assume the following equations for the discrete nonlinear system:

$$\boldsymbol{x}_k = f(\boldsymbol{x}_{k-1}, \boldsymbol{u}_k), \tag{16}$$

$$\boldsymbol{y}_k = h(\boldsymbol{x}_k, \boldsymbol{v}_k), \tag{17}$$

where $f(\cdot)$ and $h(\cdot)$ are non-linear functions; $\boldsymbol{u}_k$ and $\boldsymbol{v}_k$ are procedure noise and measurement noise, respectively, and their corresponding covariance matrices are $\boldsymbol{Q}_k$ and $\boldsymbol{R}_k$.

Step 1 Initialization.

$$\hat{\boldsymbol{x}}_{\boldsymbol{0}} = E[\boldsymbol{x}_{\boldsymbol{0}}], \tag{18}$$

$$\boldsymbol{S}_{\boldsymbol{0}} = \mathrm{chol}\left((\boldsymbol{x}_{\boldsymbol{0}} - \hat{\boldsymbol{x}}_{\boldsymbol{0}})(\boldsymbol{x}_{\boldsymbol{0}} - \hat{\boldsymbol{x}}_{\boldsymbol{0}})^{\mathrm{T}}\right), \tag{19}$$

where $\hat{\boldsymbol{x}}_{\boldsymbol{0}}$, $\boldsymbol{S}_{\boldsymbol{0}}$ are the initial system state quantities and the initial square root of the error covariance matrix, respectively. chol(·) denotes the square root of the error covariance matrix obtained by Cholesky decomposition.

Step 2 Constructing Sigma sampling points.

$$\boldsymbol{\chi}_{k-1} = \begin{bmatrix} \hat{\boldsymbol{x}}_{k-1} & \hat{\boldsymbol{x}}_{k-1} + \gamma \boldsymbol{S}_{k-1} & \hat{\boldsymbol{x}}_{k-1} - \gamma \boldsymbol{S}_{k-1} \end{bmatrix} \tag{20}$$

where $\boldsymbol{\chi}_{k-1}$ denotes the $(k-1)$th Sigma vector, $\lambda = \alpha^2(L+\kappa) - L$ is a scaling parameter, α determines the dispersion of Sigma sampling points. It is usually taken as a small positive number. κ is usually set to 0, $\gamma = \sqrt{L+\lambda}$.

Step 3 Time update.

$$\boldsymbol{\chi}_{k|k-1} = f(\boldsymbol{\chi}_{k-1}, \boldsymbol{u}_{k-1}), \tag{21}$$

$$\hat{\boldsymbol{x}}_{k|k-1} = \sum_{i=0}^{2L} W_i^{(m)} \boldsymbol{\chi}_{i,k|k-1}, \tag{22}$$

$$\boldsymbol{S}_k^- = \mathrm{qr}\left\{\left[\sqrt{\boldsymbol{W}_1^{(c)}}(\boldsymbol{\chi}_{1:2N,k|k-1} - \hat{\boldsymbol{x}}_{k|k-1}) \quad \sqrt{\boldsymbol{Q}}\right]\right\}, \tag{23}$$

$$\boldsymbol{S}_k^- = \mathrm{cholupdate}\left\{\boldsymbol{S}_k^-, (\boldsymbol{\chi}_{0,k|k-1} - \hat{\boldsymbol{x}}_{k|k-1}), W_0^{(c)}\right\}, \tag{24}$$

$$\boldsymbol{\mathcal{Y}}_{k|k-1} = h(\boldsymbol{\chi}_{k|k-1}, \boldsymbol{v}_{k-1}), \tag{25}$$

$$\hat{\boldsymbol{y}}_{k|k-1} = \sum_{i=0}^{2L} W_i^{(m)} \boldsymbol{\mathcal{Y}}_{i,k|k-1}, \tag{26}$$

where qr(·) is the QR decomposition of the matrix, cholupdate(·) is the update function of decomposition, $W_i^{(m)}$ is the mean value weight, and $W_i^{(c)}$ is the covariance weight.

Step 4 Measurement update.

$$\boldsymbol{S}_{\tilde{y}_k} = \mathrm{qr}\left\{\left[\sqrt{W_1^{(c)}}(\boldsymbol{\mathcal{Y}}_{1:2N,k|k-1} - \hat{\boldsymbol{y}}_{k|k-1}) \quad \sqrt{\boldsymbol{R}}\right]\right\}, \tag{27}$$

$$\boldsymbol{S}_{\tilde{y}_k} = \mathrm{cholupdate}\left\{\boldsymbol{S}_{\tilde{y}_k}, (\boldsymbol{\mathcal{Y}}_{0,k|k-1} - \hat{\boldsymbol{y}}_{k|k-1}), W_0^{(c)}\right\}, \tag{28}$$

$$\boldsymbol{P}_{x_k y_k} = \sum_{i=0}^{2L} W_i^{(c)} [\boldsymbol{\chi}_{i,k|k-1} - \hat{\boldsymbol{x}}_{k|k-1}][\boldsymbol{\mathcal{Y}}_{i,k|k-1} - \hat{\boldsymbol{y}}_{k|k-1}]^{\mathrm{T}}, \tag{29}$$

$$\boldsymbol{K}_k = \left(\boldsymbol{P}_{x_k y_k} / \boldsymbol{S}_{\tilde{y}_k}^{\mathrm{T}}\right) / \boldsymbol{S}_{\tilde{y}_k}, \tag{30}$$

$$\hat{\boldsymbol{x}}_k = \hat{\boldsymbol{x}}_{k|k-1} + \boldsymbol{K}_k(\boldsymbol{y}_k - \hat{\boldsymbol{y}}_{k|k-1}), \tag{31}$$

$$\boldsymbol{U} = \boldsymbol{K}_k \boldsymbol{S}_{\tilde{y}_k}, \tag{32}$$

$$\boldsymbol{S}_k = \mathrm{cholupdate}\{\boldsymbol{S}_k^-, \boldsymbol{U}, -1\}, \tag{33}$$

where $\boldsymbol{P}_{x_k y_k}$ is cross-covariance between vectors $\boldsymbol{x}_k$ and $\boldsymbol{y}_k$, $\boldsymbol{K}_k$ is filter gain matrix, $\boldsymbol{S}_k$ is the square root of the error covariance matrix at the current moment.

4.2 FSRUKF

SRUKF improves the numerical stability of the filter. The statistical properties of system and measurement noise are generally assumed to be Gaussian white noise with known mean and covariance in the filtering process. However, due to uncertainties in the actual system, such

as the internal and external environment, the real system noise and measurement noise statistical properties are not known exactly. If the artificially set noise covariance is significantly different from the real covariance, it will lead to a decrease in filtering accuracy and may even lead to filtering divergence. To solve this problem, this paper introduces a fuzzy inference system to achieve real-time adjustment of the measurement noise covariance matrix so that filtering can be performed stably.

The fuzzy inference system is based on fuzzy set theory and fuzzy inference method, with fuzzy logic as the main computational tool and a system with the ability to handle fuzzy information. The fuzzy inference system is mainly composed of fuzzification, fuzzy inference, fuzzy inference rules, and defuzzification. The structure of the fuzzy inference system is shown in Fig. 5.

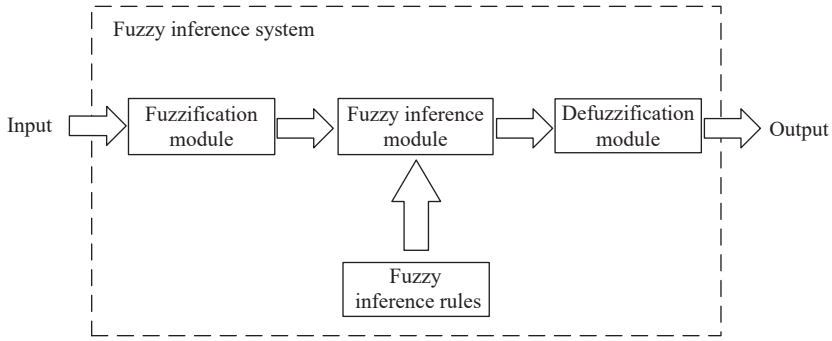


Fig. 5 Fuzzy inference system structure

The residual is defined as the difference between the real and the estimated measurements of the filter, which can be expressed as follows:

$$r = Z - \hat{Z}. \tag{34}$$

Assuming that N represents a window of time in the filtering process, the estimate of the residuals can be calculated by

$$C_{r,k} \approx \frac{1}{N} \sum_{i=k-N+1}^{k} \boldsymbol{r}_i \boldsymbol{r}_i^{\mathrm{T}} \tag{35}$$

where $C_{r,k}$ represents the residual estimation at moment k, and $\boldsymbol{r}_i$ represents the measured residual at time i.

The theoretical value of the residuals is calculated as follows:

$$\boldsymbol{P}_{r,k} = \sum_{i=0}^{2L} W_i^{(c)} \left(\boldsymbol{\mathcal{Y}}_{i,k|k-1} - \hat{\boldsymbol{Z}}_k\right)\left(\boldsymbol{\mathcal{Y}}_{i,k|k-1} - \hat{\boldsymbol{Z}}_k\right)^{\mathrm{T}} + \boldsymbol{R}_{k-1} \tag{36}$$

where $W_i^{(c)}$ is the covariance weight, $\boldsymbol{\mathcal{Y}}_{i,k|k-1}$ is the Sigma sample point after the nonlinear change of the measurement equation, and $\boldsymbol{R}_{k-1}$ is the measurement noise covariance matrix at $k-1$.

q_k is defined as the ratio of the residual estimated value to the theoretical residual value, then

$$q_k = \frac{\mathrm{tr}(C_{r,k})}{\mathrm{tr}(P_{r,k})} \tag{37}$$

where $\mathrm{tr}(\cdot)$ represents the trace of the mean value.

If the mathematical model is accurate, q_k, the ratio of the estimated residual variance to the theoretical value of the filtered residual variance should be around 1. If the ratio deviates from 1, it indicates that the measurement noise has changed, and the measurement noise covariance matrix $\boldsymbol{R}_k$ needs to be adjusted to bring q_k back to around 1. From (37), it can be obtained that increasing $\boldsymbol{R}_k$ can decrease q_k, and decreasing $\boldsymbol{R}_k$ can increase q_k. Therefore, q_k can be adjusted by $\boldsymbol{R}_k$.

$$\boldsymbol{R}_k = s_k^b \boldsymbol{R}_{k-1} \tag{38}$$

where b is a positive number that deflates the adjustment coefficient s_k.

$\boldsymbol{R}_k$ is adjusted by s_k. When s_k is greater than 1, $\boldsymbol{R}_k$ is increased; when s_k is less than 1, $\boldsymbol{R}_k$ is decreased; when s_k is 1, $\boldsymbol{R}_k$ is unchanged.

Taking q_k as the input of the fuzzy inference system, the adjustment coefficient s_k of the measurement noise covariance matrix is obtained after the fuzzy operation.

Define the input fuzzy subsets LESS to be less than 1, MORE to be greater than 1, EQUAL to be around 1. The corresponding output fuzzy subsets are DECREASE, INCREASE, and UNCHANGE respectively. Input and output fuzzy subset intervals and fuzzy partitions are shown in Table 2 and Table 3.

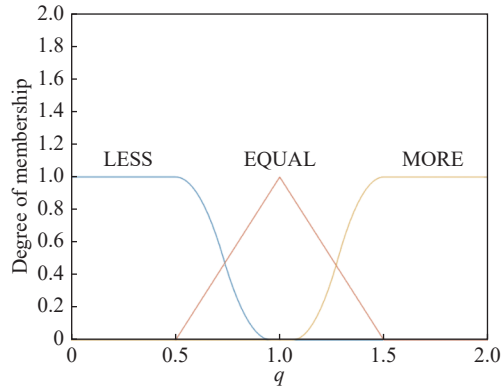
Table 2 Input fuzzy subset interval and fuzzy division

Input	Fuzzy division
[0,0.96]	LESS
[0.5,1.5]	EQUAL
[1.06,Inf]	MORE

Table 3 Output fuzzy subset interval and fuzzy division

Output	Fuzzy division
[0,0.98]	DECREASE
[0.6,1.4]	UNCHANGE
[1.02,Inf]	INCREASE

The fuzzy affiliation functions of the input and output variables are shown in Fig. 6 (a) and Fig. 6(b) respectively. q represents q_k and s represents s_k.



(a) Input affiliation function

(b) Output affiliation function

Fig. 6 Fuzzy system input and output affiliation function

According to the above-mentioned relationship between input and output variables, fuzzy inference rules are established as follows:

IF $q_k \in$ EQUAL, THEN $s_k \in$ UNCHANGE;

IF $q_k \in$ MORE, THEN $s_k \in$ INCREASE;

IF $q_k \in$ LESS, THEN $s_k \in$ DECREASE.

Because the center of gravity (COG) method of defuzzification can take into account all the relevant information about the fuzzy quantity and has better robustness，this paper adopts COG for defuzzification to obtain s_k.

FSRUKF is mainly modified in SRUKF measurement update phase. The ratio of the residual covariance estimated value to the theoretical value is obtained through the fuzzy inference system to obtain the adaptive adjustment coefficient s_k. The updated measurement noise variance $\boldsymbol{R}_k$ is obtained by updating the current measurement noise variance matrix $\boldsymbol{R}_{k-1}$ using the adaptive adjustment coefficient, and then the updated measurement noise variance $\boldsymbol{R}_k$ at $k=k+1$ is substituted into (39) for the QR decomposition.

$$\boldsymbol{S}_{\bar{y}_k} = \mathrm{qr}\left\{\left[\sqrt{W_1^{(c)}}\left(\boldsymbol{\mathcal{Y}}_{1:2N,k|k-1} - \hat{\boldsymbol{y}}_{k|k-1}\right) \ \sqrt{\boldsymbol{R}_{k-1}}\right]\right\} \tag{39}$$

Then the new square root of the covariance is obtained by factoring the update via (40).

$$\boldsymbol{S}_{\bar{y}_k} = \mathrm{cholupdate}\left\{\boldsymbol{S}_{\bar{y}_k}, \left(\boldsymbol{\mathcal{Y}}_{0,k|k-1} - \hat{\boldsymbol{y}}_{k|k-1}\right), W_0^{(c)}\right\} \tag{40}$$

The algorithm flow of FSRUKF is shown in Fig. 7.

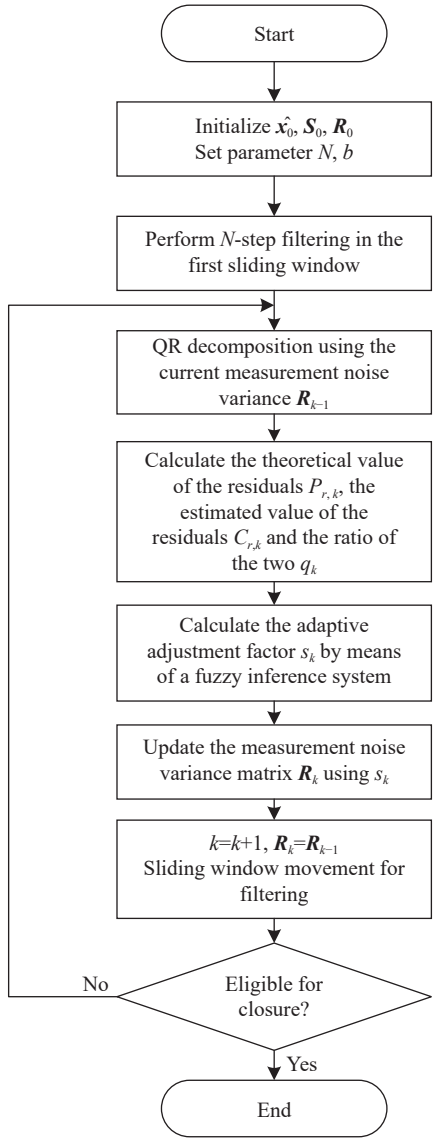


Fig. 7 FSRUKF algorithm flow

4.3 Simulation results and comparison

A typical highly nonlinear model, the univariate nonstationary growth model (UNGM), is used to validate the effectiveness of the improved algorithm, and the dynamic state-space model of UNGM is as follows:

$$x_k = \alpha \cdot x_{k-1} + \beta \frac{x_{k-1}}{1+x_{k-1}^2} + \gamma \cos(\eta \cdot (k-1)) + w_k, \quad (41)$$

$$z_k = \tau \cdot x_k^2 + v_k, \quad (42)$$

where α, β, γ, η, and τ are the nonlinear system parameters, k is the sampling moment, w_k and v_k are the process noise and measurement noise, respectively, and the system model parameters are set to $\alpha = 0.5$, $\beta = 25$, $\gamma = 8$, $\eta = 1.2$, and $\tau = 0.05$.

The exponential term b of the adaptive regulation coefficient indirectly affects the regulation of the measurement noise variance matrix $\boldsymbol{R}_k$ by influencing the adaptive regulation coefficient k_s. In order to select a suitable value for parameter b, it is necessary to investigate the influence of parameter b on the regulation of the system. Through a variety of simulation experiments, in general, the larger the value of parameter b is, the stronger the regulation effect on the system, and the shorter the transition time for the ratio q of the covariance estimate to the theoretical value to converge to 1. The smaller the value of parameter b is, the weaker the regulation effect on the system, and the longer the transition time for the ratio q of the covariance estimate to the theoretical value to converge to 1. When $b > 1$, it means that the adaptive adjustment coefficient is enlarged and the system can be adjusted faster, but the error is larger and the ratio q will oscillate around 1. When $b < 1$, it means that the adaptive adjustment coefficient is reduced, and the system can adjust more slowly, but the adjustment effect is more accurate. After the comprehensive analysis of the above, through multiple experimental simulations, the exponential term $b = 1.5$ of the adaptive adjustment coefficient is selected.

The nonlinear model is estimated by using SRUKF and FSRUKF. SRUKF and FSRUKF are run for 100 iterations with simulation times of 0.002 9 s and 0.05 s. The state estimation results and the absolute error of estimation are shown in Fig. 8 and Fig. 9, respectively.

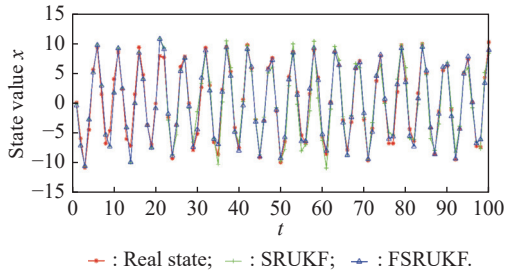


Fig. 8 Result of filtering estimation

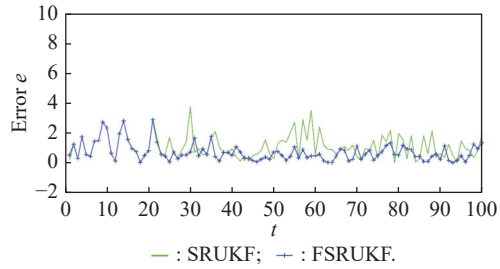


Fig. 9 Absolute error of estimation

From Fig. 8 and Fig. 9, it can be seen that both FSRUKF and SRUKF can complete the state estimation for the nonlinear model, but the estimation effect of FSRUKF is better than that of SRUKF. It can be seen from the figures that at the beginning, both FSRUKF and SRUKF perform the state estimation for the nonlinear model, and the difference between the two estimations is not large, while the error with the true value is larger. After the 22nd step, the error between SRUKF and the true value is still large, while FSRUKF can track the true value more accurately, so the filtering effect of FSRUKF is better than that of SRUKF. In addition, the convergence speed of FSRUKF is faster than SRUKF, and the error of FSRUKF is maintained at a relatively small level after the 22nd step, while the error of SRUKF is basically above FSRUKF.

In summary, FSRUKF has the following characteristics compared with SRUKF.

(i) FSRUKF can obtain higher estimation accuracy than SRUKF for nonlinear models with unknown measurement noise variance matrices.

(ii) FSRUKF is capable of adaptive adjustment of the noise variance matrix and thus has a faster convergence speed compared to SRUKF.

(iii) FSRUKF has a higher algorithm complexity than SRUKF because it uses a fuzzy inference system, which adds extra computational effort.

Therefore, in scenarios where the nonlinear system is used with an unknown measurement noise variance matrix and no strict requirements on algorithm complexity, FSRUKF may give better results than SRUKF in practical applications.

5. Experimental protocol design and experimental validation

5.1 Overall system solution

Based on the previous simulation experiments, it can be seen that FSRUKF has better convergence speed and estimation accuracy. Based on the proposed FSRUKF, the following experimental scheme is designed.

Firstly, the binocular camera continuously detects the change in target position, then corrects the measured target position information to compensate for the effects of

various time delays in the system, and compares the corrected target position information with the current end position of the manipulator to calculate the error between the two, and judges whether the error is less than the error threshold set in advance. If it is less than the grasping condition, the manipulator performs the grasping action to grasp the target; if not, the error is sent to the visual servo control system to plan the movement of the manipulator, and the control variables of each joint of the manipulator are calculated and sent to the manipulator controller to drive the manipulator to track the target. After the arm movement is completed, the position of the manipulator's end effector is updated and compared again with the target position information obtained by the binocular camera. The above steps are repeated until the grasping conditions are met. The position-based visual servo grasping system is shown in Fig. 10.

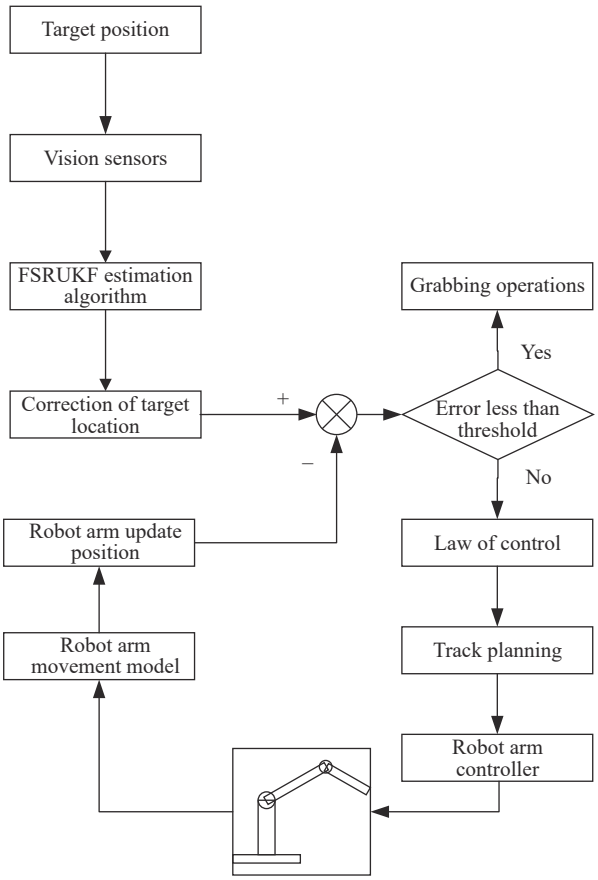


Fig. 10 Block diagram of position-based visual servo grasping system

5.2 Gripping system construction and experimental validation

The hardware devices used in the grasping experiment are Yuejiang DOBOT Magician manipulator, a stereolabs ZED binocular camera, a target object, and a PC control unit. The experimental scenario involves a human holding the target object in motion. The experimental platform for the manipulator tracking and grabbing system is shown in Fig. 11.

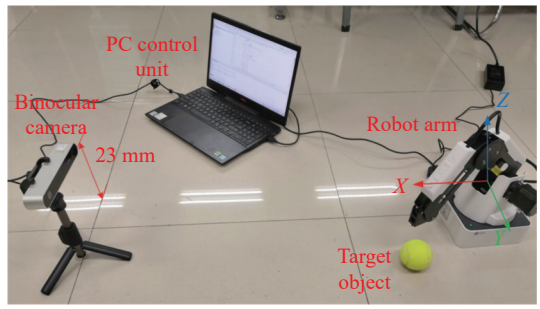


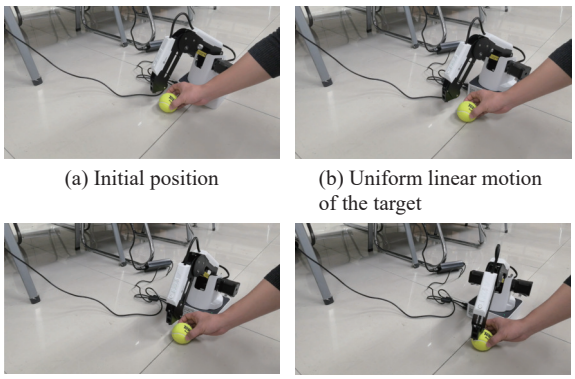
Fig. 11 Grabbing system experiment platform

Two types of target motion are tested separately. One is that the speed and direction of motion of the target do not change abruptly, in other words, the target moves in a straight line. The other is that the target's speed and direction of motion will undergo abrupt changes, in other words, the target performs random movements. The origin of the coordinate system of the experimental scene coincides with the origin of the manipulator coordinate system, where the direction of the Z-axis is perpendicular to the ground upwards, the direction of the X-axis is perpendicular to the Z-axis pointing straight ahead, and the direction of the Y-axis can be determined according to the right-handed criterion. The definition of each coordinate axis is shown in Fig. 11, and the coordinate axes are in centimeters.

Firstly, the target's speed and direction are tested without sudden changes. The target is held in a straight line on the horizontal ground, of which the maximum speed is controlled to be no more than 200 m/s so that the speed of the arm is always greater than the speed of the target to ensure that the arm could successfully track and grasp it. The position-based visual servo grasping system is used to track and grasp the target, and the movement direction of the target is shown in Fig. 12.

Fig. 12 Target direction

The initial position of the target is [173.31, −30, −10], and the initial position of the manipulator is [200, −50, −8]. The target starts to move along the *Y*-axis of the manipulator coordinate system. The manipulator starts to track the target after receiving the measurement data from the camera and coincides with the target at position [175, 26, −10] to achieve grasping. The experimental results are shown in Fig. 13.



(a) Initial position (b) Uniform linear motion of the target

(c) Tracking of the manipulator (d) Grip of the target

Fig. 13 Tracking crawling effect at a constant speed

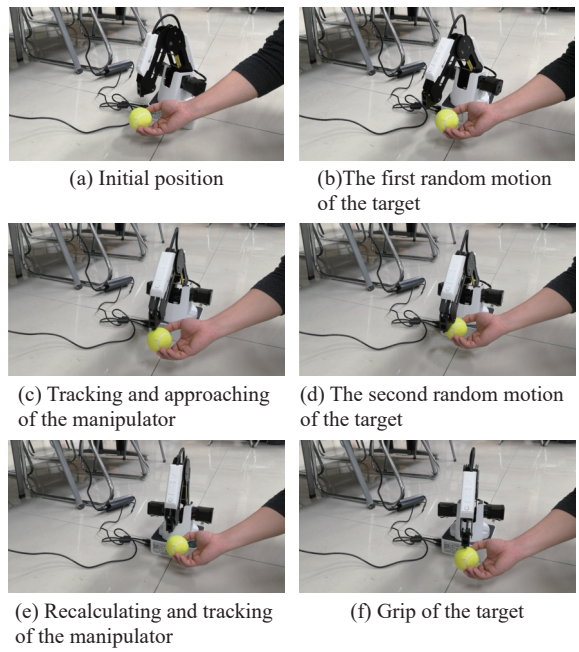
Under the condition that the dynamic characteristics of the target are the same, the linear tracking and grasping experiments are repeated. The experimental data are counted to obtain the tracking and grasping results without sudden changes in the speed and direction of the target, as shown in Table 4.

Table 4 Tracking and crawling results without sudden changes in speed

Algorithm	Number of experiments	Number of successful crawls/%	Success rate	Average tracking time/s
FSRUKF	20	17	85.00	0.74
SRUKF	20	15	75.00	0.91

From the experimental results, it can be seen that the grasping success rate and the average tracking time of FSRUKF algorithm are better than those of SRUKF when compared with SRUKF without sudden changes in target speed and direction, reflecting the superiority of FSRUKF proposed in this paper in target tracking and grasping.

In the following test, the velocity and direction of the target are suddenly changed, and the target is held in a 3D space with random motion, and the velocity of the target does not exceed 200 m/s. The experimental results are shown in Fig. 14. The manipulator begins to track the target after receiving the measurement data from the camera, as shown in Fig. 14 (a) and Fig. 14(b). In order to detect the manipulator's ability to track the target in the case of sudden acceleration movement of the target, the speed, and direction of the target are suddenly changed when the manipulator is about to coincide with the target, as shown in Fig. 14(c) and Fig. 14(d). After the manipulator moves to the planning position, it does not coincide with the target object and fails to grasp the target. Then the vision servo grasping system regains the target position and plans the trajectory of the manipulator, which tracks the target again along the planned trajectory, as shown in Fig. 14 (e), until the end of the manipulator is tracked to coincide with the target position to achieve the gripping of the target as shown in Fig. 14 (f).



(a) Initial position (b)The first random motion of the target

(c) Tracking and approaching of the manipulator (d) The second random motion of the target

(e) Recalculating and tracking of the manipulator (f) Grip of the target

Fig. 14 Tracking crawling effect with sudden changes in speed

The experiments are repeated several times under the same conditions of the target dynamic characteristics, and the results are counted to obtain the tracking and grasping results under the sudden change of target speed and direction, as shown in Table 5.

Table 5 Tracking and capturing results of sudden changes in speed

Algorithm	Number of experiments	Number of successful crawls	Success rate/%	Average tracking time/s
FSRUKF	20	12	60.00	3.27
SRUKF	20	8	40.00	3.86

As can be seen from the experimental results, although the manipulator fails to track the position of the target in time due to the sudden change in the speed and direction of the target during the target tracking process, the visual

servo is always detecting the tracking error of the manipulator. When the tracking error does not meet the conditions for successful grasping, the vision servo control will drive the manipulator to track the target again until the tracking and grasping conditions are met and the grasping system successfully grasps the target. The grasping strategy based on tracking error can ensure that when the manipulator fails to track the target in time, the grasping system immediately re-tracks and grasps the target, ensuring that the tracking and grasping system has certain robustness. In addition, compared with SRUKF, FSRUKF shows a higher success rate and a shorter average tracking time when grasping targets with abrupt changes in speed and direction, demonstrating the better robustness and advancement of the FSRUKF proposed in this paper.

6. Conclusions

This paper investigates the problem of motion target grasping and proposes a tracking and grasping scheme combining FSRUKF with a position-based visual servo. An experimental platform for a grasping system based on a visual servo is built, and experiments are conducted for both linear and random motion of the target. The experimental results show that the designed tracking and grasping system can achieve autonomous tracking and grasping of moving targets, with good feasibility and robustness.

References

[1] LIU X J, YU J J, WANG G B, et al. Research trend and scientific challenge of robotics. Bulletin of National Natural Science Foundation of China, 2016, 30(5): 425–431. (in Chinese)

[2] CARRON A, ARCARI E, WERMELINGER M, et al. Data-driven model predictive control for trajectory tracking with a manipulator. IEEE Robotics and Automation Letters, 2019, 4(4): 3758–3765.

[3] CHEN L, YANG H W, LIU P. Intelligent robot arm: vision-based dynamic measurement system for industrial applications. Proc. of the 12th International Conference on Intelligent Robotics and Applications, 2019: 120−130.

[4] KALMAN R E. A new approach to linear filtering and prediction problems. Journal of Basic Engineering, 1960, 82(1): 35–45.

[5] MCGEE L A, SCHMIDT S F. Discovery of the Kalman filter as a practical tool for aerospace and industry. California: National Aeronautics and Space Administration, 1985.

[6] DIVYA K S, RAMESH K S, RAO S K, et al. Underwater object tracking using unscented Kalman filter. Proc. of the 3rd International Conference on Advances in Computing, Communication Control and Networking, 2021: 1729−1733.

[7] SETOODEH P, HABIBI S, HAYKIN S. Particle filter in nonlinear filters: theory and applications. Hoboken: Wiley, 2022.

[8] MOHAMMED M Q, CHUNG K L, CHYI C S. Review of deep reinforcement learning-based object grasping: techniques, open challenges, and recommendations. IEEE Access, 2020, 8: 178450–178481.

[9] FRESE U, BAUML B, HAIDACHER S, et al. Off-the-shelf vision for a robotic ball catcher. Proc. of the IEEE/RSJ International Conference on Intelligent Robots and Systems, 2001: 1623−1629.

[10] KANG J, ZHENG R. Parabolic method of robot capture trajectory unknown based on EKF. Transducer and Microsystem Technologies, 2020, 39(1): 13–17. (in Chinese)

[11] PAING M S, WILLIAM NSHAMA E, UCHIYAMA N. Motion trajectory estimation of a flying object and optimal reduced impact catching by a planar manipulator. Proc. of the IEEE 29th International Conference on Robot and Human Interactive Communication, 2020: 920−925.

[12] CHEN Y, LIU Y. Improved unscented Kalman filtering algorithm applied to on-vehicle tracking system. Proc. of the International Conference on Computer, Internet of Things and Control Engineering, 2021: 64−68.

[13] XU Y B, XU K, WAN J W, et al. Research on particle filter tracking method based on Kalman filter. Proc. of the IEEE 2nd Advanced Information Management, Communicates, Electronic and Automation Control Conference, 2018: 1564−1568.

[14] GONG Z Y, QIU C R, TAO B, et al. Tracking and grasping of moving target based on accelerated geometric particle filter on colored image. Science China Technological Sciences, 2021, 64(4): 755–766.

[15] KIM S, SHUKLA A, BILLARD A. Catching objects in flight. IEEE Trans. on Robotics, 2014, 30(5): 1049–1065.

[16] LUO R C, LIAO C H. Robotic conveyor tracking with dynamic object fetching for industrial automation. Proc. of the IEEE 15th International Conference on Industrial Informatics, 2017: 369−374.

[17] QU J D, ZHANG F H, TANG Y X, et al. Dynamic visual tracking for robot manipulator using adaptive fading Kalman filter. IEEE Access, 2020, 8: 35113–35126.

[18] LI X T, ZHENG Y, SUN T T. Application of an improved and unscented Kalman filtering algorithm in target tracking. Proc. of the Chinese Control and Decision Conference, 2019: 18−22.

[19] ZHANG Y, LIU Z B, BIN Z, et al. A novel adaptive resampling particle filtering algorithm. Proc. of the IEEE International Conference on Computational Science and Engineering and IEEE International Conference on Embedded and Ubiquitous Computing, 2017: 297−300.

[20] DENAVIT J. A kinematic notation for lower-pair mechanisms based on matrices. Journal of Applied Mechanics, 1955, 22(2): 215–221.

[21] GAO B. Research on binocular vision system for industrial robots. Wuhan: Huazhong University of Science & Technology, 2019. (in Chinese)

[22] SUN X Y, JIANG Y Z, JI Y F, et al. Distance measurement system based on binocular stereo vision. Proc. of the IOP Conference Series: Earth and Environmental Science, 2019: 252−260.

[23] Rudolph V D M, ERIC W. Sigma-point Kalman filters for probabilistic inference in dynamic state-space models. Portland: Oregon Health & Science University, 2004.

[24] MERWE R V D, WAN E A. The square-root unscented Kalman filter for state and parameter-estimation. Proc. of the IEEE International Conference on Acoustics, 2002: 3461−3464.

Biographies

SHI Guoqing was born in 1974. He received his M.S. and Ph.D. degrees in system engineering from Northwestern Polytechnical University. He is an associate professor at the Department of System and Control Engineering in the School of Electronics and Information, Northwestern Polytechnical University, China. His research interests include integrated avionics system measurement and test technologies, development and design of embedded real-time systems, and modeling simulation and effectiveness evaluation of complex systems.

E-mail: shiguoqing@nwpu.edu.cn

ZHANG Boyan was born in 1997. She is currently working toward her M.S. degree in electronics and information technology with the School of Electronics and Information, Northwestern Polytechnical University. Her research interests include manipulator tracking technology, artificial intelligence and its application in unmanned aerial vehicle control and decision.

E-mail: forgetleft@mail.nwpu.edu.cn

ZHANG Jiandong was born in 1974. He received his M.S. and Ph.D. degrees in system engineering from Northwestern Polytechnical University. He is an associate professor at the Department of System and Control Engineering in the School of Electronics and Information at Northwestern Polytechnical University, China. His research interests include modeling simulation and effectiveness evaluation of complex systems, development and design of integrated avionics system, and system measurement and test technologies.

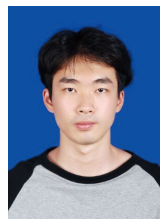
E-mail: jdzhang@nwpu.edu.cn

YANG Qiming was born in 1988. He received his M.S. degree from Northwestern Polytechnical University (NPU), Xi'an, China, in 2013. He received his Ph.D. degree in electronic science and technology from NPU in 2020. He is an assistant researcher in the School of Electronics and Information at Northwestern Polytechnical University, China. His main research interests are artificial intelligence and its application on control and decision of unmanned aerial vehicle.

E-mail: yangqm@nwpu.edu.cn

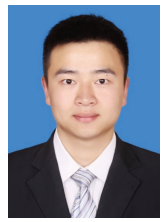
HUANG Xiaofeng was born in 1998. He received his B.S. degree in detection, guidance and control technology from Northwestern Polytechnical University, Xi'an, China. He is currently working toward his M.S. degree in control science and engineering with the School of Electronics and Information, Northwestern Polytechnical University. His research interests include avionics system, object detection and embedded system.

E-mail: huangxf@mail.nwpu.edu.cn



QUE Jianyao was born in 1997. He received his B.S. degree in detection, guidance and control technology from Northwestern Polytechnical University, Xi'an, China. He is currently working toward his M.S. degree in electronics and information technology with the School of Electronics and Information, Northwestern Polytechnical University. His research interests include avionics integrated systems, aviation fire control, and simulation of complex subsystems.

E-mail: 1910717821@qq.com



PU Junwei was born in 1995. He received his M.S. degree in the School of Electronics and Information from Northwestern Polytechnical University, Xi'an, China, in 2021. His research interests include embedded technology, manipulators, automation technology, container cluster management, and advanced RISC machine architecture chips.

E-mail: mychineseheart@163.com

GENG Xiutang was born in 1975. He received his B.S degree in the School of Mechanical Engineering from Dalian Jiaotong University, Dalian, China, in 1999, and M.S. degree in the School of Electronics and Information from Northwestern Polytechnical University, Xi'an, China, in 2004. He received his Ph.D. degree in the Department of Control Science and Engineering from Huazhong University of Science and Technology, Wuhan, China, in 2008. His research interests include information, control and intelligence computation.

E-mail: gengxiutang1@126.com