

Research Article

End-Effector Pose Estimation in Complex Environments Using Complementary Enhancement and Adaptive Fusion of Multisensor

Mingrui Luo ^{1,2}, En Li ², Rui Guo,³ Jiaxin Liu,⁴ and Zize Liang²

¹The School of Artificial Intelligence, University of Chinese Academy of Sciences, Beijing 100049, China

²The State Key Laboratory of Management and Control for Complex Systems, Institute of Automation, Chinese Academy of Sciences, Beijing 100190, China

³State Grid Shandong Electric Power Company, Jinan, China

⁴State Grid Liaoning Electric Power Company Limited, Shenyang, China

Correspondence should be addressed to En Li; en.li@ia.ac.cn

Received 25 February 2021; Revised 20 March 2021; Accepted 25 March 2021; Published 16 April 2021

Academic Editor: Bin Gao

Copyright © 2021 Mingrui Luo et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Redundant manipulators are suitable for working in narrow and complex environments due to their flexibility. However, a large number of joints and long slender links make it hard to obtain the accurate end-effector pose of the redundant manipulator directly through the encoders. In this paper, a pose estimation method is proposed with the fusion of vision sensors, inertial sensors, and encoders. Firstly, according to the complementary characteristics of each measurement unit in the sensors, the original data is corrected and enhanced. Furthermore, an improved Kalman filter (KF) algorithm is adopted for data fusion by establishing the nonlinear motion prediction of the end-effector and the synchronization update model of the multirate sensors. Finally, the radial basis function (RBF) neural network is used to adaptively adjust the fusion parameters. It is verified in experiments that the proposed method achieves better performances on estimation error and update frequency than the original extended Kalman filter (EKF) and unscented Kalman filter (UKF) algorithm, especially in complex environments.

1. Introduction

In special manufacturing, troubleshooting, and other fields, there will inevitably be a closed and narrow operating environment, which is high risk and inefficient for manual operation. Therefore, the research on robots that can be used in this complex environment is significant and challenging, especially since the traditional manipulators are unable to do the job due to their bulky structure.

Kinematic redundancy is one of the most important properties that determine whether the robots can accomplish the task in the highly constrained environments mentioned above [1]. Manipulators with kinematic redundancy are called redundant manipulators. Redundant manipulators can be implemented in different ways, such as the wheeled mobile manipulator used for part handling [2, 3], the hyper-

redundant manipulator used for engine manufacturing [4] or spacecraft maintenance [5], and the flexible bionic manipulator inspired by the octopus claw [6] or the elephant trunk [7]. Although redundant manipulators have excellent kinematic performance, they also face the challenge of control accuracy due to the accuracy of end-effector pose estimation.

Traditional manipulators generally use joint encoders as sensor inputs and use kinematics models to calculate the pose of the end-effector as information feedback for motion control. Since the accuracy of the joint encoder can be very high, the pose calculated by the kinematics model has low noise. However, redundant manipulators usually have a large number of joints and long slender links, which means the pose will be affected by the transmission gap, mechanical vibration, elastic deformation, and other factors that deviate from the calculated value. As a result, the manipulators are in an uncertain state of

kinematics and dynamics, so complex control models need to be established to ensure control accuracy [8–10].

Complex control models may introduce system uncertainty, so many researchers try to solve the problem of unreliable encoders in redundant manipulators by studying sensor processing methods. Analyzing the source of the measurement error and establishing a compensation model can effectively improve the sensor accuracy. For example, in a cable-driven redundant manipulator, the accuracy of the encoder-based pose estimation can be improved by correcting the cable hole location error, cable length error [11], and nonnegligible cable mass [12].

In addition to error correction for a single type of sensor, multisensor fusion is also the main research direction. The most commonly used sensors fused with encoders including angle sensors, vision sensors, and inertial sensors. For one thing, adding angle sensors at joints and fusing with motor encoders can correct transmission errors and improve pose estimation accuracy [13]. For another, adding one or more eye-to-hand cameras to the environment and performing the visual measurement on redundant manipulators can correct deformation errors [14] and is also an effective method for flexible robot pose estimation [15]. Moreover, the Kalman filter and its improved methods are also used to fuse motor encoders and inertial sensors, which have good dynamic response performance and can correct vibration errors [16]. Because the sensor and the fusion algorithm are lightweight, it is especially suitable for small bionic robots [17].

The sensor processing methods mentioned above can indeed solve the problem of end-effector pose estimation, but there are also many unsatisfactory shortcomings, especially when applied in complex environments. Therefore, the estimator constructed in this paper uses the compensation model and multisensor fusion at the same time, but in different processing stages. In a dynamic environment, although the preestablished compensation model is prone to failure and cannot accurately reflect the relationship between the sensor and the estimated state, it can effectively correct the wrong measurement [18]. Therefore, this paper does not directly use the compensation model for state estimation but as the preprocessing part of the state estimation to enhance the original data of the sensor. In terms of multisensor fusion, although visual-inertial fusion has become an effective method of mobile robot navigation [19], it is not common in end-effector pose estimation of redundant robots, and it is mostly the fusion of eye-to-hand camera and kinematic sensors [20]. In a narrow environment, it is difficult to apply an eye-to-hand camera, so the vision sensor needs to be arranged at the end of the manipulator, which means that the vision sensor is more susceptible to the influence of obstacles. Therefore, based on the Kalman filter algorithm, this paper establishes a nonlinear motion prediction of the end-effector and a synchronization update model of the multirate sensors that combines vision sensors, inertial sensors, and encoders. Moreover, since the neural network can improve the robustness of pose estimation [21], this paper introduces the radial basis function neural network to adjust parameters adaptively. Through the above-mentioned improved fusion strategy, the pose estimation of the end-

effector can be more adaptable to complex environments and can better overcome the adverse effects of sensor noise and different sampling rates of multisensor fusion.

The rest of this paper is organized as follows: Section 2 introduces the overall method. Section 3 introduces the complementary enhancement of the original data. Section 4 introduces the fusion strategy including the nonlinear prediction, the synchronization update, and the adaptive adjustment. Section 5 introduces the experimental results and analysis, and Section 6 summarizes the paper.

2. The Overall Method of the Pose Estimation

Figure 1 shows the small redundant robot studied in this paper, which can enter the narrow space to accomplish the specified tasks.

Due to the miniaturized mechanical structure, when the end-effector is equipped with a load, the connecting rod between the joints will inevitably be bent and deformed. This deformation error will accumulate at the end-effector; as a result, the pose estimation based only on the joint encoders produces deviation. Therefore, structured-light RGB-D camera, magnetic angular rate, and gravity (MARG) sensor are added, together with the joint encoders.

In this paper, the fusion pose estimation method proposed mainly includes two parts: preenhancement and adaptive fusion. The system block diagram is shown in Figure 2.

The RGB-D camera and the MARG sensor can compensate for the error caused by a single encoder measurement to a certain extent, but at the same time, it is easy to reduce the robustness of the system due to the interference of the external environment. Considering that the RGB-D camera and the MARG sensor themselves are integrated sensors composed of multiple sensing units, each sensing unit has complementary characteristics, which can be used to enhance the original data of the sensors so that the fusion of multiple sensors can be completed based on robust data. According to the above theory, preenhancement is added before multisensor fusion.

The preenhancement part includes the image enhancement of the RGB-D camera and the quaternion extended Kalman filter (QEKF) fusion attitude calculation of the MARG sensor. In the image enhancement part, the color image is used to repair the depth image, which can make the measurement of the RGB-D camera more accurate and reliable in a narrow environment (Section 3.1). In the QEKF part, the magnetometer, accelerometer, and gyroscope, which are all parts of the MARG sensor, are fused to correct the drift error of each independent sensing unit (Section 3.2).

Through the preenhancement part, the robustness of each independent sensor is improved, but the following problems still need to be overcome in the multisensor fusion process: (a) nonlinear state model, (b) the sampling rate of different sensors differs greatly, and (c) sensor abnormal handling in extreme environments.

In the adaptive fusion part, the RBF and adaptive UKF (RBF-AUKF) method is modified to solve the above problems, which mainly includes pose prediction, synchronization update, and adaptive adjustment. In the pose prediction part, by establishing the pose prediction model of the end-effector

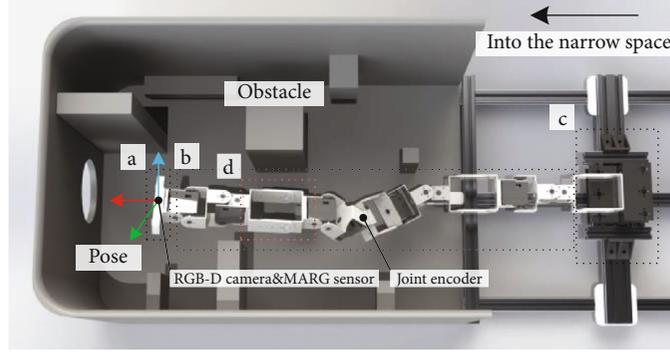


FIGURE 1: The small redundant robot (a) end-effector, (b) redundant manipulator, (c) base, and (d) connecting rod.

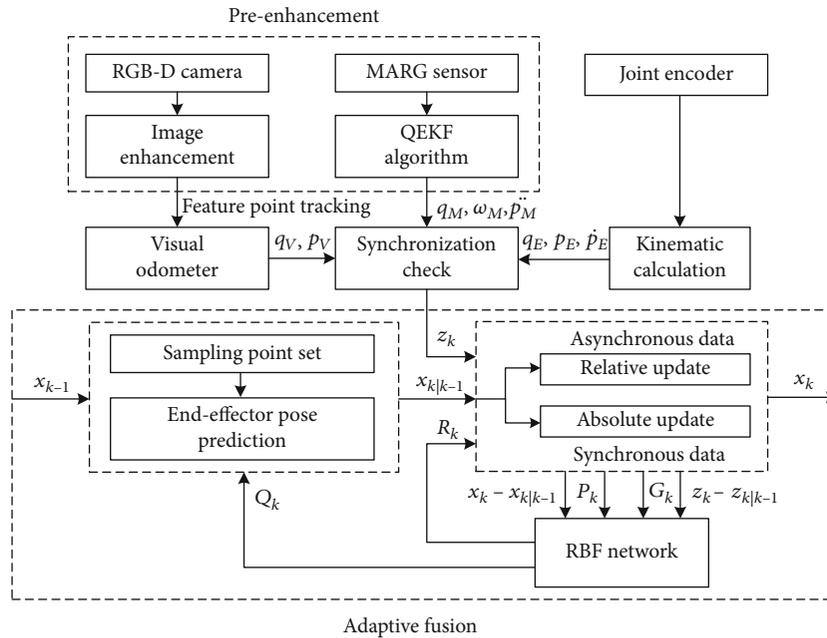


FIGURE 2: Block diagram of the pose estimation method.

and the nonlinear processing method, the pose state $x_{k|k-1}$ at the next moment can be predicted based on the pose state x_{k-1} at the previous moment (Section 4.1). In the synchronization update part, a synchronization check is performed on the preenhanced quaternion attitude q_V and position p_V obtained by the visual odometer; quaternion attitude q_M , angular velocity ω_M , and acceleration \ddot{p}_M obtained by the QEKF algorithm; and quaternion attitude q_E , position p_E , and velocity \dot{p}_E obtained by the kinematic calculation. According to the check result, the control variable is relatively updated when the sensor data z_k is asynchronous, while the state variable is absolutely updated when the sensor data z_k is synchronous. After the prediction and update, the fusion pose estimation x_k at this moment is obtained (Section 4.2). In the adaptive adjustment part, this paper selects the prediction error $x_k - x_{k|k-1}$, estimation error covariance matrix P_k , Kalman gain G_k , and observation error $z_k - z_{k|k-1}$ after updating as features. Based on the RBF neural network, the process, and observation noise

covariance matrix Q_k, R_k is adaptively adjusted to improve the robustness of fusion (Section 4.3).

3. Sensor Enhancement Based on Complementary Characteristics

3.1. Enhancement of the Depth Image. The structured-light RGB-D camera is composed of a color camera, an infrared camera, and an infrared transmitter and obtains the depth value through the speckle feature of the infrared structured light projected on the surface of the object [22]. In a closed and narrow environment, infrared light is easily reflected and absorbed by objects multiple times, so it is easier to produce invalid depth values locally, which in turn introduces additional noise for pose estimation.

Since the invalid depth value occurs locally, the invalid point can be repaired by a valid point that is in a similar region to the invalid point. Since the color camera is not affected by

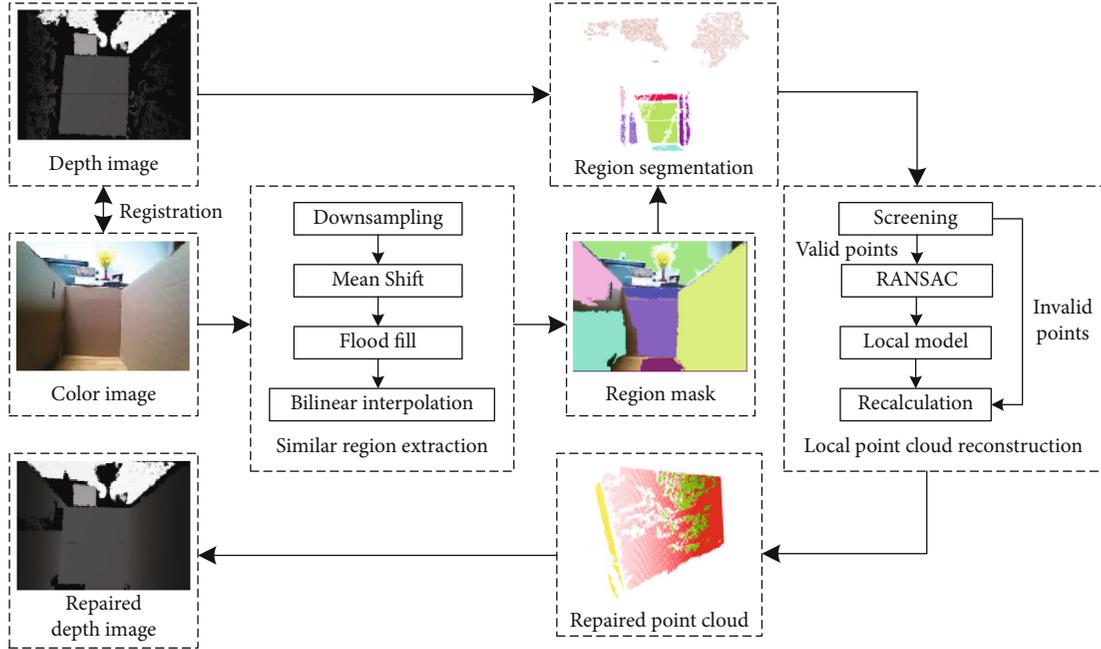


FIGURE 3: Depth image enhancement in a narrow environment.

the narrow environment, it can be used to determine whether the valid point and the invalid point belong to the same type of patch gathered by the point cloud, that is, the similar region. Therefore, the enhanced method proposed in this paper includes two steps, namely, similar region extraction and local point cloud reconstruction, as shown in Figure 3. The former extracts similar regions in the depth image through the texture features of the registered color image. The latter uses valid points in the similar region to construct a local model of the region and then recalculates invalid points in the region with the model to complete the enhancement repair process.

In the similar region extraction step, the Mean Shift algorithm is used to extract the color texture information of the color image, and the Flood Fill algorithm is used to extract the corresponding color-connected domains to form a similar region mask. Finally, the mask is applied to the depth image to obtain the region segmentation of the similar point cloud. Since the Mean Shift algorithm needs to perform multiple iterative calculations for each pixel, it will affect the real-time performance of the system in practical applications. In order to speed up the calculation process, this paper downsamples the input color image to a lower resolution, then forms the similar region mask on the low-resolution image, and finally performs bilinear interpolation on the mask to restore the original scale.

In the local point cloud reconstruction step, by fitting the patch model in the similar region and recalculating the depth value of the invalid point on the patch, the invalid point is repaired. The least-squares method is the basic method to fit the patch model, but in practical applications, the region segmented by the color image has deviation, and not all valid points can be used to fit the patch model. In order to form a more reliable local model, the RANSAC algorithm is used to filter the points used to fit the model.

Although the invalid point does not have the correct depth value, it contains the coordinate information (u, v) of the pixel plane. Combining with the local model $\hat{z} = ax + by + c$ that has been fitted, the spatial coordinate (x, y, \hat{z}) of the invalid point can be recalculated, as shown in equation (1). Finally, the invalid point in the depth image is repaired.

$$\begin{cases} x = (u - c_x) \frac{\hat{z}}{f_x}, \\ y = (v - c_y) \frac{\hat{z}}{f_y}, \\ \hat{z} = ax + by + c, \end{cases} \quad (1)$$

where c_x, c_y is the optical center offset of the camera and f_x, f_y is the focal length of the camera.

3.2. Enhancement of the Attitude Calculation. The MARG sensor is a microelectromechanical system (MEMS) composed of a three-axis accelerometer, a three-axis gyroscope, and a three-axis magnetometer. However, due to the impact of sensor manufacturing accuracy and operating environment interference, the raw measurement data of the MARG sensor is generally noisy. From the analysis of its working principle, the gyroscope has good dynamic response characteristics, but it is easy to produce cumulative errors in the attitude calculation, and the accelerometer and magnetometer are just the opposite. Therefore, the QEKF algorithm is used to fuse the three measurement units of the MARG sensor so that the output attitude angle data can be enhanced, as shown in Figure 4.

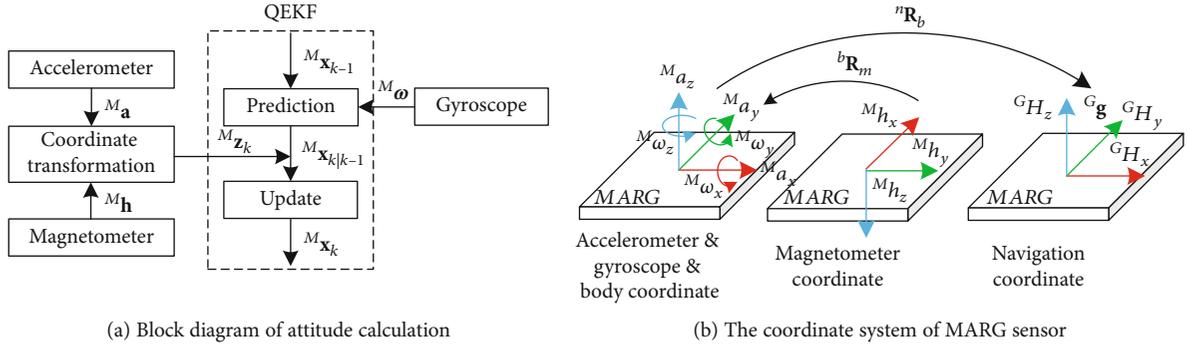


FIGURE 4: Attitude calculation enhancement.

There are three coordinate systems in this method. Firstly, the acceleration vector ${}^M\mathbf{a} = [{}^M a_x \ {}^M a_y \ {}^M a_z]^T$ measured by the accelerometer and the angular velocity vector ${}^M\boldsymbol{\omega} = [{}^M\omega_x \ {}^M\omega_y \ {}^M\omega_z]^T$ measured by the gyroscope are both in the body coordinate system. Secondly, the magnetic field intensity vector ${}^M\mathbf{h} = [{}^M h_x \ {}^M h_y \ {}^M h_z]^T$ measured by the magnetometer is in the coordinate system that can be transformed to the body coordinate system by the rotation matrix ${}^b\mathbf{R}_m$. Lastly, the geomagnetic field intensity vector ${}^G\mathbf{H} = [{}^G H_x \ {}^G H_y \ {}^G H_z]^T$ and the gravitational acceleration vector ${}^G\mathbf{g} = [0 \ 0 \ -g]^T$ are in the navigation coordinate system that can be transformed from the body coordinate system by the rotation matrix ${}^n\mathbf{R}_b$.

Suppose the system state variables ${}^M\mathbf{x}_k$ at the k -th time are

$${}^M\mathbf{x}_k = \begin{bmatrix} {}^M q_{0,k} & {}^M q_{1,k} & {}^M q_{2,k} & {}^M q_{3,k} & {}^M b_{x,k} & {}^M b_{y,k} & {}^M b_{z,k} \end{bmatrix}^T, \quad (2)$$

where ${}^M q_{0,k}, {}^M q_{1,k}, {}^M q_{2,k}, {}^M q_{3,k}$ is the quaternion value of the rotation transformation matrix ${}^n\mathbf{R}_b$ at the k -th time and ${}^M b_{x,k}, {}^M b_{y,k}, {}^M b_{z,k}$ is the accumulated drift value of the gyroscope on the X , Y , and Z axes at the k -th time.

Suppose the system control variables ${}^M\mathbf{u}_k$ at the k -th time is

$${}^M\mathbf{u}_k = \begin{bmatrix} {}^M\omega_{x,k} - {}^M b_{x,k-1} & {}^M\omega_{y,k} - {}^M b_{y,k-1} & {}^M\omega_{z,k} - {}^M b_{z,k-1} \end{bmatrix}^T. \quad (3)$$

Suppose the system observation variables ${}^M\mathbf{z}_k$ at the k -th time is

$${}^M\mathbf{z}_k = \begin{bmatrix} {}^M\mathbf{a}_k & {}^M\mathbf{h}_k \end{bmatrix}^T. \quad (4)$$

According to the discrete-time model of the quaternion attitude differential equation [23], the following equation can be obtained:

$${}^M\mathbf{q}_k = \left[\mathbf{I}_{4 \times 4} + {}^M\boldsymbol{\Omega}_{k-1} \frac{T}{2} \right] {}^M\mathbf{q}_{k-1}, \quad (5)$$

$${}^M\boldsymbol{\Omega}_{k-1} = \begin{bmatrix} 0 & {}^M b_{x,k-1} - {}^M\omega_{x,k} & {}^M b_{y,k-1} - {}^M\omega_{y,k} & {}^M b_{z,k-1} - {}^M\omega_{z,k} \\ {}^M\omega_{x,k} - {}^M b_{x,k-1} & 0 & {}^M\omega_{z,k} - {}^M b_{z,k-1} & {}^M b_{y,k-1} - {}^M\omega_{y,k} \\ {}^M\omega_{y,k} - {}^M b_{y,k-1} & {}^M b_{z,k-1} - {}^M\omega_{z,k} & 0 & {}^M\omega_{x,k} - {}^M b_{x,k-1} \\ {}^M\omega_{z,k} - {}^M b_{z,k-1} & {}^M\omega_{y,k} - {}^M b_{y,k-1} & {}^M b_{x,k-1} - {}^M\omega_{x,k} & 0 \end{bmatrix}, \quad (6)$$

where T is the discrete-time interval, $\mathbf{I}_{i \times j}$ is the identity matrix with i rows and j columns, ${}^M\mathbf{q}_k = [{}^M q_{0,k} \ {}^M q_{1,k} \ {}^M q_{2,k} \ {}^M q_{3,k}]^T$ is the quaternion vector of ${}^n\mathbf{R}_b$ at the k -th time, and ${}^M\boldsymbol{\Omega}_k$ is the component of angular velocity in the body coordinate system.

Substituting equations (2) and (3) into equation (5), (6) can obtain the state prediction equation:

$${}^M f({}^M\mathbf{x}_{k-1}, {}^M\mathbf{u}_{k-1}) = \begin{bmatrix} \mathbf{I}_{4 \times 4} + {}^M\boldsymbol{\Omega}_{k-1} \frac{T}{2} & \mathbf{0}_{4 \times 3} \\ \mathbf{0}_{3 \times 4} & \mathbf{I}_{3 \times 3} \end{bmatrix} \begin{bmatrix} {}^M\mathbf{q}_k \\ {}^M b_{x,k-1} \\ {}^M b_{y,k-1} \\ {}^M b_{z,k-1} \end{bmatrix}, \quad (7)$$

where $\mathbf{0}_{i \times j}$ is the zero matrix with i rows and j columns.

To transform the measured value of the sensor, the following equations can be obtained:

$${}^M\mathbf{a} = {}^n\mathbf{R}_b^{-1} {}^G\mathbf{g}, \quad (8)$$

$${}^M\mathbf{h} = {}^n\mathbf{R}_b^{-1} {}^b\mathbf{R}_m^{-1} {}^G\mathbf{H}. \quad (9)$$

Then, the system observation equation ${}^M h({}^M \hat{\mathbf{x}}_k)$ was obtained:

$$\begin{cases} {}^M \hat{a}_{x,k} = 2g \left({}^M \hat{q}_{1,k} {}^M \hat{q}_{3,k} - {}^M \hat{q}_{0,k} {}^M \hat{q}_{2,k} \right), \\ {}^M \hat{a}_{y,k} = 2g \left({}^M \hat{q}_{2,k} {}^M \hat{q}_{3,k} + {}^M \hat{q}_{0,k} {}^M \hat{q}_{1,k} \right), \\ {}^M \hat{a}_{z,k} = g \left({}^M \hat{q}_{0,k}^2 - {}^M \hat{q}_{1,k}^2 - {}^M \hat{q}_{2,k}^2 + {}^M \hat{q}_{3,k}^2 \right), \\ {}^M \hat{h}_{x,k} = {}^G H_x \left({}^M \hat{q}_{0,k}^2 - {}^M \hat{q}_{1,k}^2 + {}^M \hat{q}_{2,k}^2 - {}^M \hat{q}_{3,k}^2 \right) + 2{}^G H_y \left({}^M \hat{q}_{1,k} {}^M \hat{q}_{2,k} - {}^M \hat{q}_{0,k} {}^M \hat{q}_{3,k} \right) - 2{}^G H_z \left({}^M \hat{q}_{2,k} {}^M \hat{q}_{3,k} + {}^M \hat{q}_{0,k} {}^M \hat{q}_{1,k} \right), \\ {}^M \hat{h}_{y,k} = 2{}^G H_x \left({}^M \hat{q}_{1,k} {}^M \hat{q}_{2,k} + {}^M \hat{q}_{0,k} {}^M \hat{q}_{3,k} \right) + {}^G H_y \left({}^M \hat{q}_{0,k}^2 + {}^M \hat{q}_{1,k}^2 - {}^M \hat{q}_{2,k}^2 - {}^M \hat{q}_{3,k}^2 \right) - 2{}^G H_z \left({}^M \hat{q}_{1,k} {}^M \hat{q}_{3,k} - {}^M \hat{q}_{0,k} {}^M \hat{q}_{2,k} \right), \\ {}^M \hat{h}_{z,k} = -2{}^G H_x \left({}^M \hat{q}_{2,k} {}^M \hat{q}_{3,k} - {}^M \hat{q}_{0,k} {}^M \hat{q}_{1,k} \right) - 2{}^G H_y \left({}^M \hat{q}_{1,k} {}^M \hat{q}_{3,k} + {}^M \hat{q}_{0,k} {}^M \hat{q}_{2,k} \right) + {}^G H_z \left({}^M \hat{q}_{0,k}^2 - {}^M \hat{q}_{1,k}^2 - {}^M \hat{q}_{2,k}^2 + {}^M \hat{q}_{3,k}^2 \right), \end{cases} \quad (10)$$

where ${}^M \hat{a}_{x,k}$, ${}^M \hat{a}_{y,k}$, ${}^M \hat{a}_{z,k}$, ${}^M \hat{h}_{x,k}$, ${}^M \hat{h}_{y,k}$, ${}^M \hat{h}_{z,k}$ is the theoretical measurement value of the sensors and ${}^M \hat{q}_{0,k}$, ${}^M \hat{q}_{1,k}$, ${}^M \hat{q}_{2,k}$, ${}^M \hat{q}_{3,k}$ is the quaternion attitude predicted by equation (7).

According to the EKF algorithm, calculate the enhanced quaternion attitude at the k -th time:

$$\begin{cases} {}^M \hat{\mathbf{x}}_k = {}^M f({}^M \mathbf{x}_{k-1}, {}^M \mathbf{u}_{k-1}), \\ {}^M \hat{\mathbf{P}}_k = {}^M \mathbf{F}_{k-1} {}^M \mathbf{P}_{k-1} {}^M \mathbf{F}_{k-1}^T + {}^M \mathbf{Q}_{k-1}, \\ {}^M \mathbf{G}_k = {}^M \hat{\mathbf{P}}_k {}^M \mathbf{H}_k^T \left({}^M \mathbf{H}_k {}^M \hat{\mathbf{P}}_k {}^M \mathbf{H}_k^T + {}^M \mathbf{R}_k \right)^{-1}, \\ {}^M \mathbf{x}_k = {}^M \hat{\mathbf{x}}_k + {}^M \mathbf{G}_k \left[{}^M \mathbf{z}_k - {}^M h({}^M \hat{\mathbf{x}}_k) \right], \\ {}^M \mathbf{P}_k = \left(\mathbf{I} - {}^M \mathbf{G}_k {}^M \mathbf{H}_k \right) {}^M \hat{\mathbf{P}}_k, \end{cases} \quad (11)$$

where ${}^M \hat{\mathbf{x}}_k$ is the prediction of the system state, ${}^M \hat{\mathbf{P}}_k$ is the prediction of the estimation error covariance matrix ${}^M \mathbf{P}_k$, ${}^M \mathbf{F}_{k-1} = (\partial {}^M f(\mathbf{x}_{k-1}, \mathbf{u}_{k-1})) / (\partial \mathbf{x}_{k-1})|_{\mathbf{x}_{k-1} = {}^M \hat{\mathbf{x}}_{k-1}}$, ${}^M \mathbf{H}_k = (\partial {}^M h(\mathbf{x}_k)) / (\partial \mathbf{x}_k)|_{\mathbf{x}_k = {}^M \hat{\mathbf{x}}_k}$, ${}^M \mathbf{G}_k$ is the Kalman gain, \mathbf{I} is the identity matrix, and ${}^M \mathbf{Q}_{k-1}$ and ${}^M \mathbf{R}_k$ are the noise matrix.

To further improve the robustness of the system, the process noise matrix ${}^M \mathbf{Q}_{k-1}$ and the observation noise matrix ${}^M \mathbf{R}_k$ in equation (11) adopt an adaptive form.

Suppose that the angular velocity measurement value ${}^M \boldsymbol{\omega}$ obtained by the gyroscope obeys the Gaussian distribution with variance σ_g^2 , and the drift value ${}^M \mathbf{b} = [{}^M b_x \quad {}^M b_y \quad {}^M b_z]^T$ obeys the Gaussian distribution with variance σ_b^2 , substituting equation (7) to obtain the following:

$${}^M \mathbf{Q}_{k-1} = \mathbf{I}_{7 \times 7} \begin{bmatrix} (\sigma_g^2 + \sigma_b^2) \left({}^M q_{1,k-1}^2 + {}^M q_{2,k-1}^2 + {}^M q_{3,k-1}^2 \right) \frac{T^2}{4} \\ (\sigma_g^2 + \sigma_b^2) \left({}^M q_{0,k-1}^2 + {}^M q_{2,k-1}^2 + {}^M q_{3,k-1}^2 \right) \frac{T^2}{4} \\ (\sigma_g^2 + \sigma_b^2) \left({}^M q_{0,k-1}^2 + {}^M q_{1,k-1}^2 + {}^M q_{3,k-1}^2 \right) \frac{T^2}{4} \\ (\sigma_g^2 + \sigma_b^2) \left({}^M q_{0,k-1}^2 + {}^M q_{1,k-1}^2 + {}^M q_{2,k-1}^2 \right) \frac{T^2}{4} \\ \sigma_b^2 \\ \sigma_b^2 \\ \sigma_b^2 \end{bmatrix}. \quad (12)$$

Suppose that the acceleration measurement value ${}^M \mathbf{a}$ obtained by the accelerometer obeys the Gaussian distribution with variance σ_a^2 , and the magnetic field intensity measurement value ${}^M \mathbf{h}$ obtained by the magnetometer obeys the Gaussian distribution with variance σ_m^2 . Considering the influence of external acceleration and external magnetic field strength,

$${}^M \mathbf{R}_k = \begin{bmatrix} [k_a \| {}^M \mathbf{a}_k - {}^G \mathbf{g}_k \| + \sigma_a^2] \mathbf{I}_{3 \times 3} & 0_{3 \times 3} \\ 0_{3 \times 3} & [k_m \| {}^M \mathbf{h}_k - {}^G \mathbf{H}_k \| + \sigma_m^2] \mathbf{I}_{3 \times 3} \end{bmatrix}, \quad (13)$$

where k_a and k_m are correction coefficients.

4. Sensor Fusion Based on Modified RBF-AUKF Method

4.1. End-Effector Pose Prediction Model. The RGB-D camera and MARG sensor are both arranged at the end of the

manipulator, so the sensor data after preenhancement processing is directly in the end-effector coordinate system. The joint encoders are arranged at each joint of the manipulator, so it is necessary to transform the angle of each joint to the pose in the end-effector coordinate system through forward kinematics calculation.

Suppose the system state variables \mathbf{x}_k at the k -th time is

$$\mathbf{x}_k = [\mathbf{p}_k \quad \mathbf{q}_k]^T, \quad (14)$$

where $\mathbf{p}_k = [p_{x,k} \quad p_{y,k} \quad p_{z,k}]^T$ is the position of the end-effector at the k -th time and $\mathbf{q}_k = [q_{0,k} \quad q_{1,k} \quad q_{2,k} \quad q_{3,k}]^T$ is the quaternion attitude of the end-effector at the k -th time.

Suppose the system control variables \mathbf{u}_k at the k -th time is

$$\mathbf{u}_k = [\boldsymbol{\omega}_k \quad \dot{\mathbf{p}}_k \quad \ddot{\mathbf{p}}_k]^T, \quad (15)$$

where $\boldsymbol{\omega}_k = [\omega_{x,k} \quad \omega_{y,k} \quad \omega_{z,k}]^T$ is the X , Y , and Z axis angular velocities of the end-effector, $\dot{\mathbf{p}}_k = [\dot{p}_{x,k} \quad \dot{p}_{y,k} \quad \dot{p}_{z,k}]^T$ is the position velocity of the end-effector at the k -th time, and $\ddot{\mathbf{p}}_k = [\ddot{p}_{x,k} \quad \ddot{p}_{y,k} \quad \ddot{p}_{z,k}]^T$ is the position acceleration of the end-effector at the k -th time.

According to the differential model of motion [24], the system state prediction equation $f(\mathbf{x}_{k-1}, \mathbf{u}_{k-1})$ can be obtained:

$$\left\{ \begin{array}{l} \mathbf{p}_k = \mathbf{p}_{k-1} + {}^n\mathbf{R}_{b,k-1} \begin{bmatrix} \dot{p}_{x,k-1} T + \ddot{p}_{x,k-1} \frac{T^2}{2} \\ \dot{p}_{y,k-1} T + \ddot{p}_{y,k-1} \frac{T^2}{2} \\ \dot{p}_{z,k-1} T + \ddot{p}_{z,k-1} \frac{T^2}{2} \end{bmatrix}, \\ \mathbf{q}_k = \mathbf{q}_{k-1} + \frac{T}{2} \begin{bmatrix} 0 & -\omega_{x,k-1} & -\omega_{y,k-1} & -\omega_{z,k-1} \\ \omega_{x,k-1} & 0 & \omega_{z,k-1} & -\omega_{y,k-1} \\ \omega_{y,k-1} & -\omega_{z,k-1} & 0 & \omega_{x,k-1} \\ \omega_{z,k-1} & \omega_{y,k-1} & -\omega_{x,k-1} & 0 \end{bmatrix} \mathbf{q}_{k-1}, \end{array} \right. \quad (16)$$

where T is the discrete-time interval and ${}^n\mathbf{R}_{b,k-1}$ is the rotation matrix from the body coordinate system to the navigation coordinate system as shown in equation (17).

$${}^n\mathbf{R}_{b,k-1} = \begin{bmatrix} q_{0,k-1}^2 + q_{1,k-1}^2 - q_{2,k-1}^2 - q_{3,k-1}^2 & 2(q_{1,k-1}q_{2,k-1} - q_{0,k-1}q_{3,k-1}) & 2(q_{1,k-1}q_{3,k-1} + q_{0,k-1}q_{2,k-1}) \\ 2(q_{1,k-1}q_{2,k-1} + q_{0,k-1}q_{3,k-1}) & q_{0,k-1}^2 - q_{1,k-1}^2 + q_{2,k-1}^2 - q_{3,k-1}^2 & 2(q_{2,k-1}q_{3,k-1} - q_{0,k-1}q_{1,k-1}) \\ 2(q_{1,k-1}q_{3,k-1} - q_{0,k-1}q_{2,k-1}) & 2(q_{2,k-1}q_{3,k-1} + q_{0,k-1}q_{1,k-1}) & q_{0,k-1}^2 - q_{1,k-1}^2 - q_{2,k-1}^2 + q_{3,k-1}^2 \end{bmatrix}. \quad (17)$$

The state prediction equation mentioned in equation (16) has obvious nonlinear characteristics. Both EKF and UKF algorithms can solve the problem of nonlinear state estimation. However, when the calculation performance can meet the requirements, the estimation accuracy of the UKF algorithm is higher [25]. By sampling, the UKF algorithm approximates the probability density distribution of the nonlinear prediction equation, and the constructed sampling point set is as follows:

$$[\chi_k]_i = \begin{cases} \mathbf{x}_k, & i = 0, \\ \mathbf{x}_k + \left[\sqrt{(N + \lambda)\mathbf{P}_{k-1}} \right]_i, & i = 1, \dots, N, \\ \mathbf{x}_k - \left[\sqrt{(N + \lambda)\mathbf{P}_{k-1}} \right]_i, & i = N + 1, \dots, 2N, \end{cases} \quad (18)$$

where $[\chi_k]_i$ is the i -th column of the sample point set matrix at the k -th time, which is a 7-dimensional column vector, $\left[\sqrt{(N + \lambda)\mathbf{P}_{k-1}} \right]_i$ is the i -th column of the matrix $\sqrt{(N + \lambda)\mathbf{P}_{k-1}}$, which is an N -dimensional column vector,

λ is the scale parameter to adjust the approximation accuracy, and \mathbf{P}_{k-1} is the estimation error covariance matrix.

4.2. Multirate Update Model. Due to the limitation of sensor processing capacity, the measurement update frequency of different sensors is not the same, or even quite different, which means that even if the sensors are triggered to collect at the same time, the measurement of each sensor cannot be received at the same time.

Although it is difficult for sensors to publish data synchronously all the time, it is possible to approximate the software synchronization of multiple sensors through timestamp alignment. The basic strategy is to set a sliding time window; when all sensor data falls within this window, it can be considered that the sensor data is synchronous. As is shown in Figure 5, d is the size of the sliding time window, which determines the accuracy of data synchronization. Its maximum size is the smallest update time interval among all sensors. The smaller the size, the higher the accuracy of the time, but the lower the frequency of synchronous data.

The UKF algorithm corrects the prediction model through sensor observations. Therefore, when the algorithm is used for multisensor fusion, the fusion accuracy is greatly affected by the time synchronization [26]. Because the

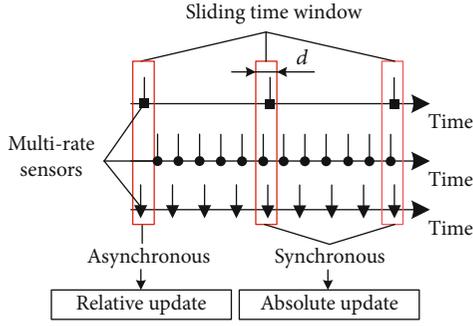


FIGURE 5: An update model that maintains an average update frequency.

update frequency of synchronous data is generally much lower than the average update frequency of data, the real-time performance of synchronization fusion estimation is poor. To solve this shortcoming, a modified synchronization fusion algorithm is proposed including two update models, namely, the absolute state update model and the relative state update model.

Define a matrix $\mathbf{H}_{m \times n}$ with m rows and n columns as a state update matrix, and its i -th row and j -th column satisfy the following:

$$[\mathbf{H}_{m \times n}]_{i,j} = \begin{cases} 1, & \text{measurement } i \text{ is related to state } j, \\ 0, & \text{otherwise.} \end{cases} \quad (19)$$

The absolute state update model updates the system state variables including the position and attitude. The number of rows and columns in the state update matrix $\mathbf{H}_{m \times n}$ satisfies the following:

$$m = \sum_{k=1}^{N_A} s_k, \quad n = 7, \quad (20)$$

where N_A is the total number of sensors synchronized with the system state variables and N_A must be equal to N_{\max} which is the maximum number of sensors; otherwise $\mathbf{H}_{m \times n} = 0$. s_k is the total number of the system state variables that the k -th sensor can observe.

The relative state update model updates the system control variables including velocity and acceleration. The number of rows and columns in the state update matrix $\mathbf{H}_{m \times n}$ satisfies the following:

$$m = \sum_{k=1}^{N_R} s_k, \quad n = 9, \quad (21)$$

where N_R is the total number of sensors synchronized with the system control variables and N_R must satisfy $2 \leq N_R \leq N_{\max}$; otherwise $\mathbf{H}_{m \times n} = 0$. s_k is the total number of system control variables that the k -th sensor can observe.

When the multisensor data is not all synchronized, the relative state update is used to increase the update frequency

locally, and when the data is all synchronized, the system state is globally corrected to improve the estimation accuracy. The final observation equation is as follows:

$$h(\mathbf{x}_{k|k-1}, \mathbf{u}_k) = \begin{bmatrix} \mathbf{H}_A & 0 \\ 0 & \mathbf{H}_R \end{bmatrix} \begin{bmatrix} \mathbf{x}_{k|k-1} \\ \mathbf{u}_k \end{bmatrix}, \quad (22)$$

where \mathbf{H}_A is the absolute state update matrix determined by equation (20) and \mathbf{H}_R is the relative state update matrix determined by equation (21).

According to the UKF algorithm, calculate the fusion pose estimation \mathbf{x}_k of the end-effector:

$$\left\{ \begin{array}{l} \chi_{k|k-1} = f(\chi_{k-1}, \mathbf{u}_{k-1}), \\ \mathbf{x}_{k|k-1} = \sum_{i=0}^{2n} W_i [\chi_{k|k-1}]_i, \\ \mathbf{P}_{x,k|k-1} = \sum_{i=0}^{2n} W_i \left[[\chi_{k|k-1}]_i - \mathbf{x}_{k|k-1} \right] \left[[\chi_{k|k-1}]_i - \mathbf{x}_{k|k-1} \right]^T + \mathbf{Q}_k, \\ \mathbf{y}_{k|k-1} = h(\chi_{k|k-1}, \mathbf{u}_k), \\ \mathbf{z}_{k|k-1} = \sum_{i=0}^{2n} W_i [\mathbf{y}_{k|k-1}]_i, \\ \mathbf{P}_{z,k|k-1} = \sum_{i=0}^{2n} W_i \left[[\mathbf{y}_{k|k-1}]_i - \mathbf{z}_{k|k-1} \right] \left[[\mathbf{y}_{k|k-1}]_i - \mathbf{z}_{k|k-1} \right]^T + \mathbf{R}_k, \\ \mathbf{P}_{xz,k|k-1} = \sum_{i=0}^{2n} W_i \left[[\chi_{k|k-1}]_i - \mathbf{x}_{k|k-1} \right] \left[[\mathbf{y}_{k|k-1}]_i - \mathbf{z}_{k|k-1} \right]^T, \\ \mathbf{G}_k = \mathbf{P}_{xz,k|k-1} \mathbf{P}_{z,k|k-1}^{-1}, \\ \mathbf{x}_k = \mathbf{x}_{k|k-1} + \mathbf{G}_k (\mathbf{z}_k - \mathbf{z}_{k|k-1}), \\ \mathbf{P}_k = \mathbf{P}_{x,k|k-1} - \mathbf{G}_k \mathbf{P}_{z,k|k-1} \mathbf{G}_k^T, \end{array} \right. \quad (23)$$

where $\chi_{k|k-1}$ is the state prediction of sampling points determined by equation (16), $\mathbf{x}_{k|k-1}$ is the weighted mean of the state prediction, W_i is the weight value determined by equation (24), $\mathbf{P}_{x,k|k-1}$ is the weighted variance of the state prediction, $\mathbf{y}_{k|k-1}$ is the state observation of sampling points determined by equation (22), $\mathbf{z}_{k|k-1}$ is the weighted mean of the state observation, $\mathbf{P}_{z,k|k-1}$ is the weighted variance of the state observation, $\mathbf{P}_{xz,k|k-1}$ is the weighted variance of the state prediction and observation, \mathbf{z}_k is the sensor measurement value, \mathbf{G}_k is the Kalman gain, \mathbf{Q}_k and \mathbf{R}_k are the noise matrix.

$$W_i = \begin{cases} \frac{N}{(N + \lambda)}, & i = 0N, \\ \frac{N}{2(N + \lambda)}, & i = 1, \dots, 2N. \end{cases} \quad (24)$$

4.3. Adaptive Adjustment of the Noise Matrix. It can be seen from equation (23) that the process noise covariance matrix \mathbf{Q}_k and the observation noise covariance matrix \mathbf{R}_k can affect

the confidence level of the data during state prediction and update. In general applications, the variance of system state noise and sensor measurement noise is a fixed value that can be counted in advance. However, in a complex environment, the variance of noise will fluctuate to a certain extent. If the fixed value is still set in advance, it will affect the accuracy of the final fusion state estimation. Therefore, the system identification ability of the RBF network needs to be used to adaptively adjust the noise variance.

The RBF network includes a three-layer structure of input layer, hidden layer, and output layer, as shown in Figure 6. The prediction error $\mathbf{x}_k - \mathbf{x}_{k|k-1}$, estimation error covariance matrix \mathbf{P}_k , Kalman gain \mathbf{G}_k , and observation error $\mathbf{z}_k - \mathbf{z}_{k|k-1}$ shown in equation (23) can reflect the deviation and fluctuation during the prediction and update process in the UKF and can be used as features of the input layer.

Considering the similarity between the features, select the features according to the position state, attitude state, and the observations of each sensor:

$$\begin{cases} \mathbf{x}_k - \mathbf{x}_{k|k-1} = [\mathbf{e}_{p,k} & \mathbf{e}_{q,k}]^T, \\ \mathbf{P}_k = \begin{bmatrix} \mathbf{P}_{pp,k} & \mathbf{P}_{pq,k} \\ \mathbf{P}_{qp,k} & \mathbf{P}_{qq,k} \end{bmatrix}, \\ \mathbf{G}_k = [\mathbf{G}_{s1,k} & \mathbf{G}_{s2,k} & \mathbf{G}_{s3,k}], \\ \mathbf{z}_k - \mathbf{z}_{k|k-1} = [\mathbf{e}_{s1,k} & \mathbf{e}_{s2,k} & \mathbf{e}_{s3,k}]^T, \end{cases} \quad (25)$$

where $\mathbf{e}_{p,k}$, $\mathbf{e}_{q,k}$ are the position and attitude prediction error; $\mathbf{P}_{pp,k}$, $\mathbf{P}_{pq,k}$, $\mathbf{P}_{qp,k}$, $\mathbf{P}_{qq,k}$ are the estimation error covariance matrix of position, position-attitude, attitude-position, and attitude; $\mathbf{G}_{s1,k}$, $\mathbf{G}_{s2,k}$, $\mathbf{G}_{s3,k}$ are the Kalman gain of the three sensors; and $\mathbf{e}_{s1,k}$, $\mathbf{e}_{s2,k}$, $\mathbf{e}_{s3,k}$ are the observation error of the three sensors.

Extract the main information of the feature matrix in equation (25) to obtain the input layer feature vector \mathbf{x}_{input} , which contains 10 feature information:

$$\mathbf{x}_{input} = [\|\mathbf{e}_{p,k}\| \quad \|\mathbf{e}_{q,k}\| \quad tr(\mathbf{P}_{pp,k}) \quad tr(\mathbf{P}_{qq,k}) \quad \|\mathbf{G}_{s1,k}\| \\ \cdot \|\mathbf{G}_{s2,k}\| \|\mathbf{G}_{s3,k}\| \|\mathbf{e}_{s1,k}\| \|\mathbf{e}_{s2,k}\| \|\mathbf{e}_{s3,k}\|]^T. \quad (26)$$

From input layer to hidden layer, radial basis function is used to activate neurons:

$$\phi_i(\mathbf{x}_{input}) = e^{-\|\mathbf{x}_{input} - \mu_i\| / \sigma_i^2}, \quad (27)$$

where μ_i , σ_i are the center and width of the i -th hidden layer neuron.

The linear function is used from hidden layer to output layer:

$$\mathbf{y}_{output,j} = \sum_i \omega_{ij} \phi_i(\mathbf{x}_{input}), \quad (28)$$

where ω_{ij} is the weight of the i -th hidden layer neuron to the j -th output layer neuron.

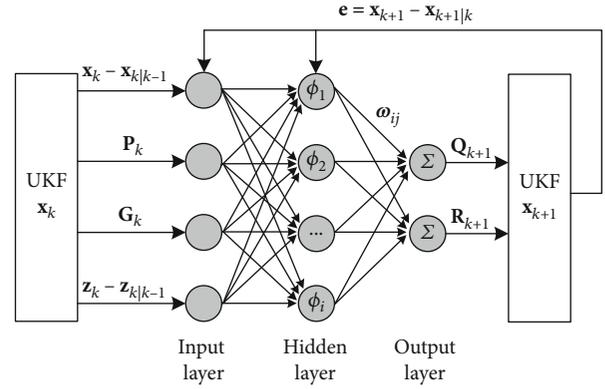


FIGURE 6: RBF network for adjusting noise matrix.

The output of the output layer is as follows:

$$\mathbf{y}_{output} = [\sigma_p \quad \sigma_q \quad \sigma_{s1} \quad \sigma_{s2} \quad \sigma_{s3}]^T, \quad (29)$$

$$\begin{cases} \mathbf{Q}_{k+1} = \mathbf{I}[\sigma_p \quad \sigma_q]^T, \\ \mathbf{R}_{k+1} = \mathbf{I}[\sigma_{s1} \quad \sigma_{s2} \quad \sigma_{s3}]^T, \end{cases} \quad (30)$$

where σ_p and σ_q are the standard deviations of the process noise of the position and attitude and σ_{s1} , σ_{s2} , and σ_{s3} are the standard deviations of the observation noise of the three sensors.

The training process of the network is divided into two stages: offline and online. The offline training stage uses the data with known noise variance to obtain the initial neuron center and connection weights. At this stage, the output error of the network is the difference between the true noise variance and the network predicted noise variance. Since the noise variance in the offline stage is not the noise variance in the actual experiment, it is easy to produce overfitting in the offline training stage, which needs to be corrected through the online stage. The online training stage uses the updated prediction error $\mathbf{x}_{k+1} - \mathbf{x}_{k+1|k}$ at the next time as the output error of the network. The training goal is to minimize the prediction error:

$$\text{Loss} = \arg \min_{\mu_i, \omega_{ij}} \|\mathbf{x}_{k+1} - \mathbf{x}_{k+1|k}\|. \quad (31)$$

5. Experiments and Results

5.1. Experimental Device. The mechanical structure of the experimental device consisted of a 1-DOF motion platform and an 8-DOF redundant manipulator, as shown in Figure 7. The joint encoders were arranged at each rotary joint, and the RGB-D camera and the MARG sensor were arranged at the end of the manipulator. The RGB-D camera was connected to the industrial personal computer (IPC) through USB, and the MARG sensor and the joint encoders were connected to the sensor collector through the 485 bus and then connected to the IPC through USB.

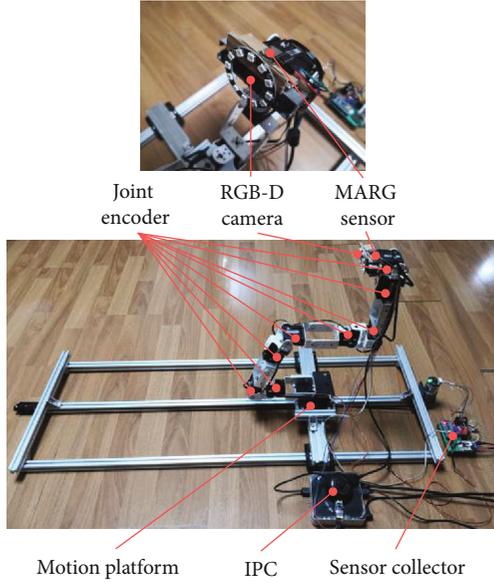


FIGURE 7: Experimental device.

5.2. Experiment of the Enhancement Part. In order to test the effect of the sensor complementary enhancement method proposed in this article, the experimental device was placed in a narrow artificially built environment, considering that the RGB-D camera focuses more on static repeatability and the MARG sensor focuses more on dynamic stability. Therefore, when testing the enhancement effect of the RGB-D camera, a periodically repeated expected motion trajectory was used, while the MARG sensor used a random expected trajectory. The experimental results are shown in Figures 8 and 9.

Comparing Figures 8(a) and 8(b), it can be seen that the point cloud after enhancement is closer to the real environment and has a less invalid blind area caused by the narrow environment. Figure 8(c) further demonstrates the impact of depth image enhancement on the accuracy of position estimation. It can be seen from the figure that the position trajectory obtained by using the original image information fluctuates more drastically and deviates from the reference trajectory to a greater degree. But the position trajectory obtained by using the enhanced image information is closer to the reference trajectory, and the overall fluctuation is smaller, which is more conducive to the fusion pose estimation with other sensors.

Comparing the desired and enhanced quaternion attitude in Figure 9, the overall calculation error of the QEKF algorithm proposed in this paper is within the expected range, which means acceleration information can compensate for the integral drift of angular velocity and geomagnetic field information can calibrate angle measurement errors. Through the enhancement of attitude calculation, it can provide more reliable attitude data for multisensor fusion.

5.3. Simulation of the Fusion Part. Because the statistical tests are very sensitive when the sample size is small, both simulation tests and actual experiments were carried out to complete the verification of the multisensor fusion pose esti-

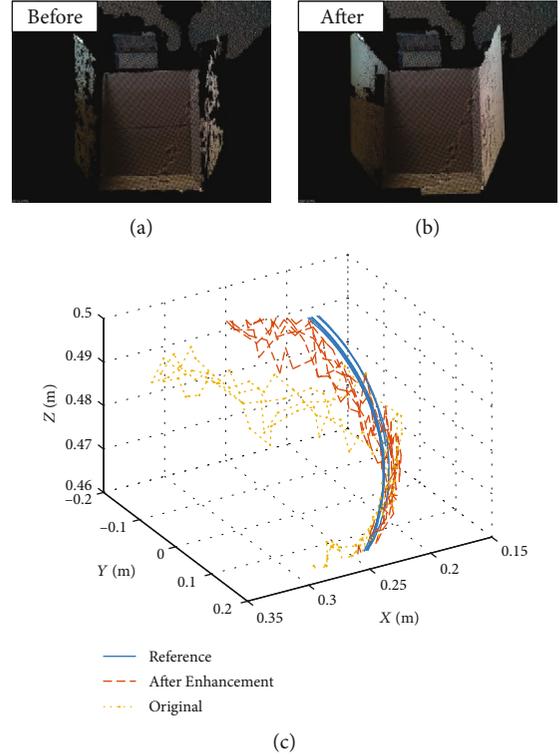


FIGURE 8: Results of depth image enhancement.

mation method. Simulation tests were mainly used to test the stability of the method under extreme conditions, and actual experiments were mainly used to test the fusion effect of the method in real environments. It was easy to adjust the sensor parameters in simulation environments, so a large number of samples can be generated to test the performance of the method. Although the experimental samples that were collected in the real environments were limited, these samples were more targeted and reliable after the screening of the experimental methods by the simulation tests. At the same time, to present the experimental results as objectively as possible, the root mean square (RMS) errors of multiple experimental results were used to evaluate the performance differences of different methods.

Since real sensors can only be simulated numerically under simulation conditions, the actual state and sensors were simplified during the simulation. Different from the real system, an observation state and two sensors were set up in the simulation environment. The observation noise of sensor 1 was smaller than that of sensor 2, and sensor 1 only produced a stable zero offset, while the zero offset of sensor 2 increased with time. Figure 10 and Table 1 show the effect of the synchronization update method on the fusion result.

Since the noise of sensor 2 is greater than that of sensor 1, the RMS error of sensor 2 is greater than that of sensor 1 when the state of a single sensor is estimated. In the simulation, the data update frequency f_1 of sensor 1 was different from the data update frequency f_2 of sensor 2. If the data synchronization of the sensor was not considered, the state update was performed when the data of any sensor inputted.

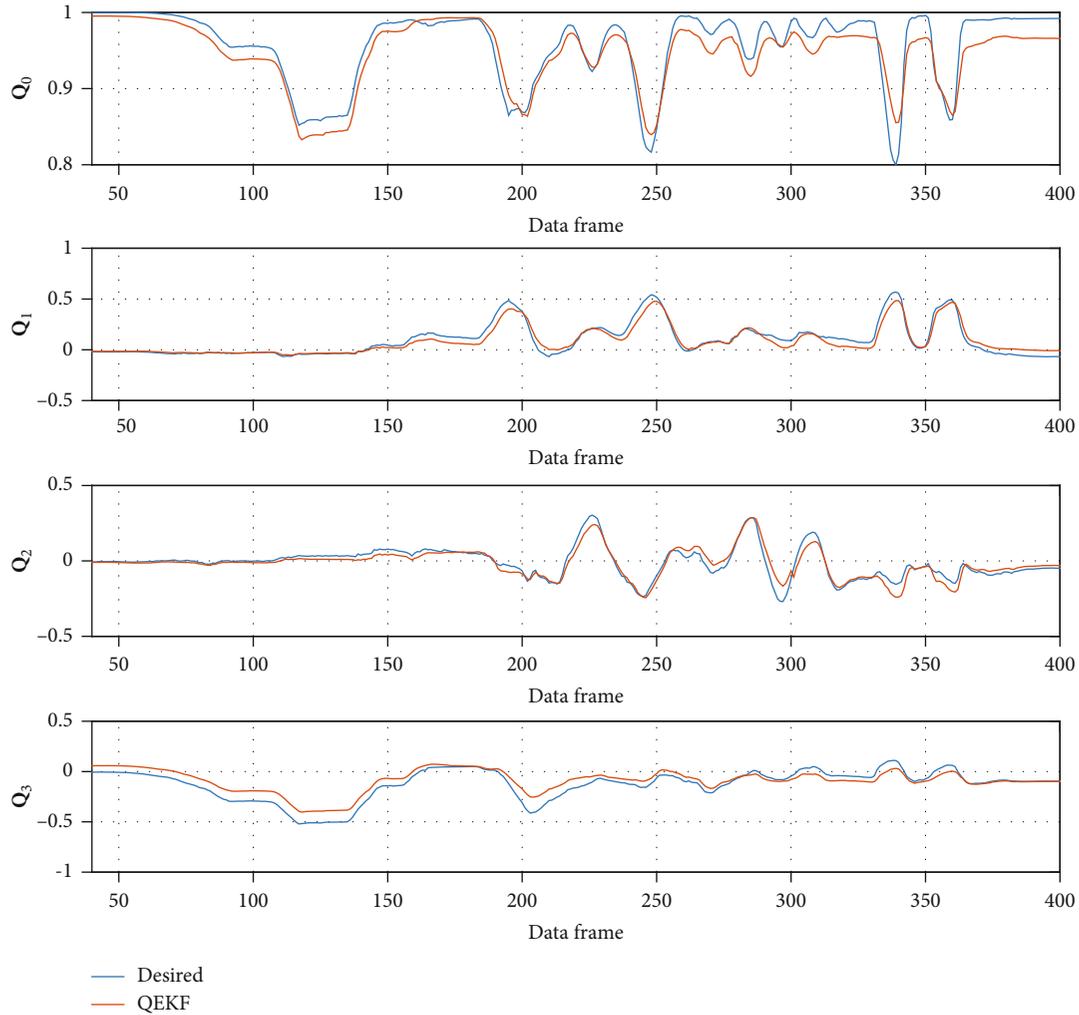


FIGURE 9: Results of attitude calculation enhancement.

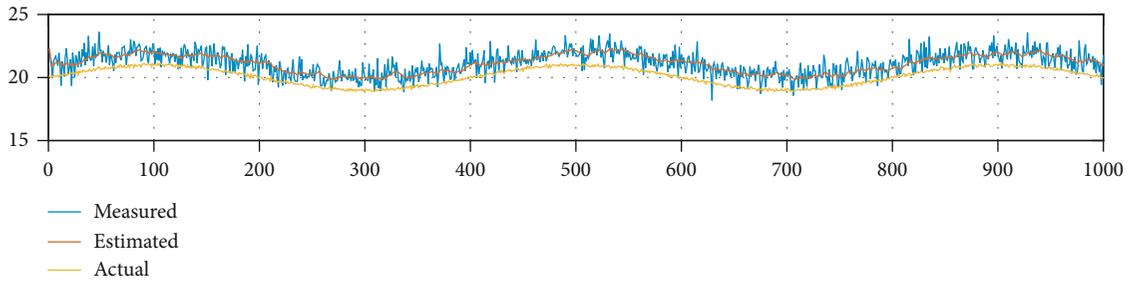
From the experimental results shown in Figures 10(c) and 10(d), the RMS error when $f_1 > f_2$ is less than that when $f_1 < f_2$, indicating that the fusion result is affected by the sensor update frequency, and it is biased towards high-sampling-rate data sources. If the high-sampling-rate data is noisy, the fusion accuracy cannot be significantly improved. The above defects are improved in the modified synchronization update method. It can be seen from the experimental results shown in Figure 10(e) that the RMS error of the fusion data is significantly reduced, and there will be no sawtooth fluctuations in the asynchronous fusion. Therefore, the data synchronization of sensors is an important factor in multirate sensor fusion.

In the simulation environment, the process and observation noise conformed to the Gaussian distribution, but the variance of the distribution changed at every moment and satisfied a nonlinear model. The RBF network was designed to predict the variances of the process and observation noise and was trained when the filter was updated. In the simulation, the variances predicted by the network in the test set were compared with the preset variances, and the coefficient

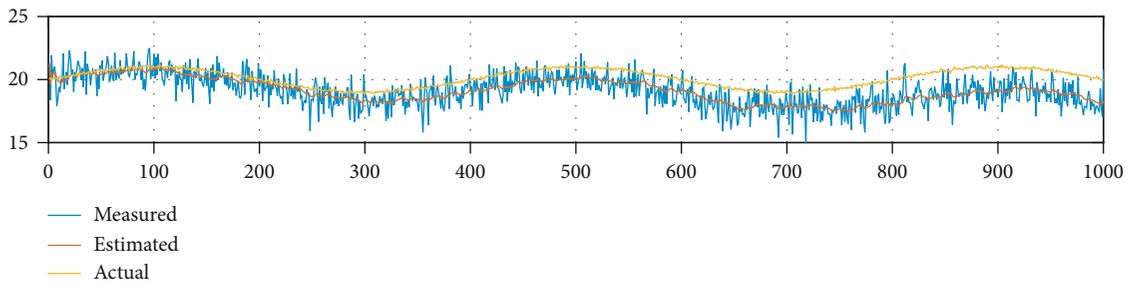
of determination (marked as R -square) was calculated. The closer the coefficient of determination is to 1, the higher the accuracy of the network. It can be seen from the results shown in Figure 11 that the network can predict the variances of the noise accurately and has no low coefficients of determination (0.71 for the process noise and 0.84 for the observation noise).

Although the coefficient of determination cannot be obtained when the true variance is unknown, the accuracy of the network can be indirectly reflected through the state estimation deviation. In other words, if the state estimation deviation can remain stable even in extreme cases, it means that the network's prediction of the parameters is accurate. In the simulation, the process noise and the observation noise of the sensor were designed to increase with time. Figure 12 and Table 2 show the fusion effects of different fusion algorithms in this case.

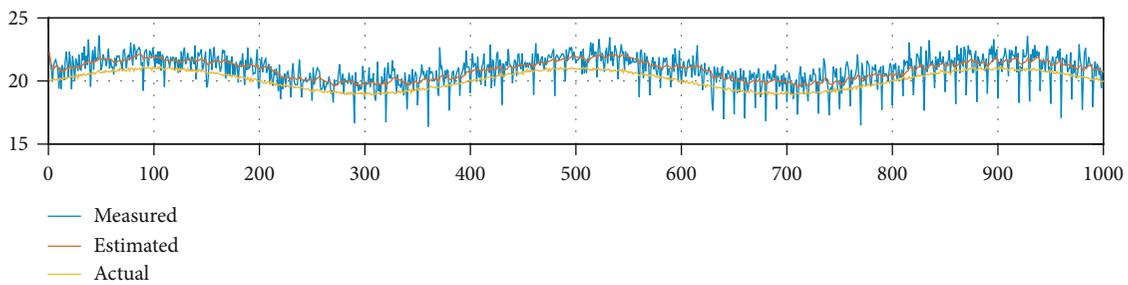
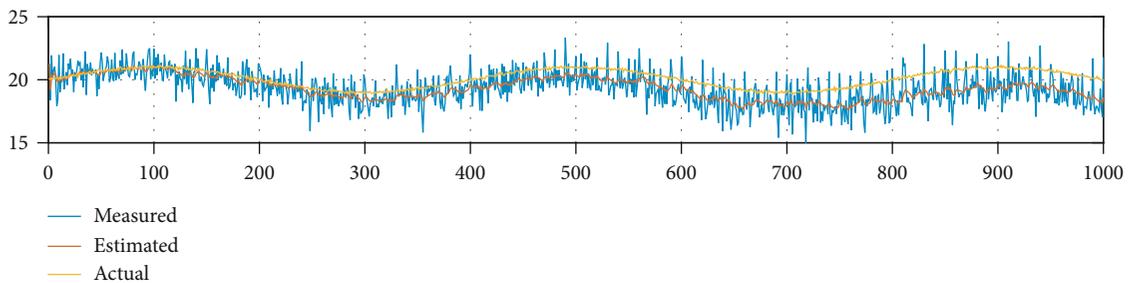
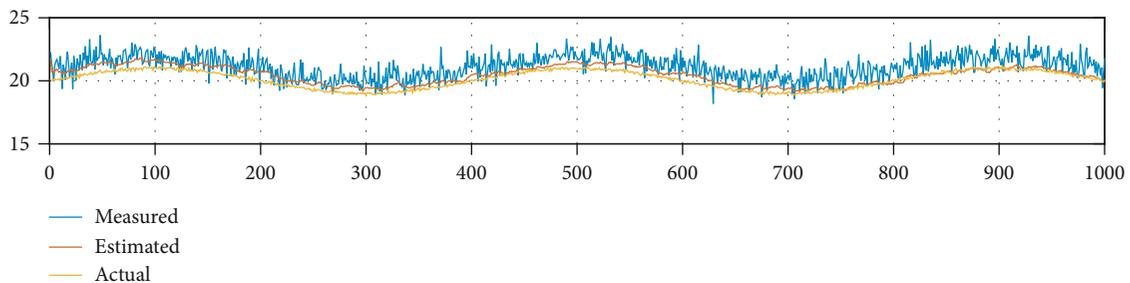
It can be seen from the experimental results that at the beginning, the system noise is small and the estimation error of each fusion method is not much different, but as time goes by, the system noise increases. The original EKF



(a) Single sensor 1



(b) Single sensor 2

(c) Sensor 1 and 2 update frequency: $f_1 > f_2$ (d) Sensor 1 and 2 update frequency: $f_1 < f_2$ 

(e) Sensor 1 and 2 synchronized

FIGURE 10: The impact of synchronization.

TABLE 1: RMS errors of different update methods.

	Update method				
	Sensor1 Single	Sensor2 Single	Sensors 1 and 2 $f_1 > f_2$	Sensors 1 and 2 $f_1 > f_2$	Sensors 1 and 2 Synchronized
RMS error	1.02	1.16	0.824	0.962	0.413

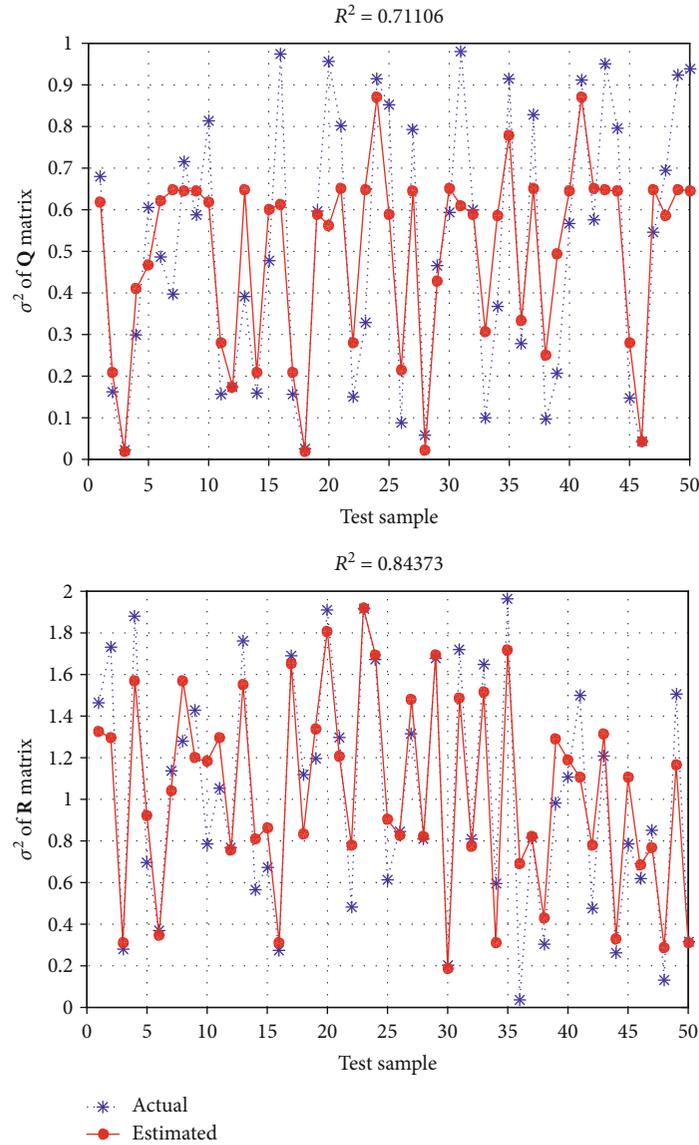


FIGURE 11: Prediction of the noise matrix.

and UKF algorithms cannot adaptively adjust the filter parameters, which are easily affected by noise and produce state estimation deviations. Through the RBF network proposed in this paper, the noise covariance matrix can be predicted, so that the EKF and UKF algorithms have adaptive adjustment capabilities. The modified RBF-AEKF and RBF-AUKF methods have smaller RMS errors than the original EKF and UKF methods, so they can be more robust in complex environments.

5.4. Experiment of the Fusion Part. Since the environment established in the simulation can only approximately reflect the actual situation, it is still necessary to conduct experiments in the real environment. In actual experiments, the end-effector of the manipulator periodically moved around the Z-axis of the base coordinate system, and the pose of the end-effector was set as the desired value. The position estimations p_x, p_y, p_z along the X, Y, and Z axes were obtained by fusion of the visual odometer based on the

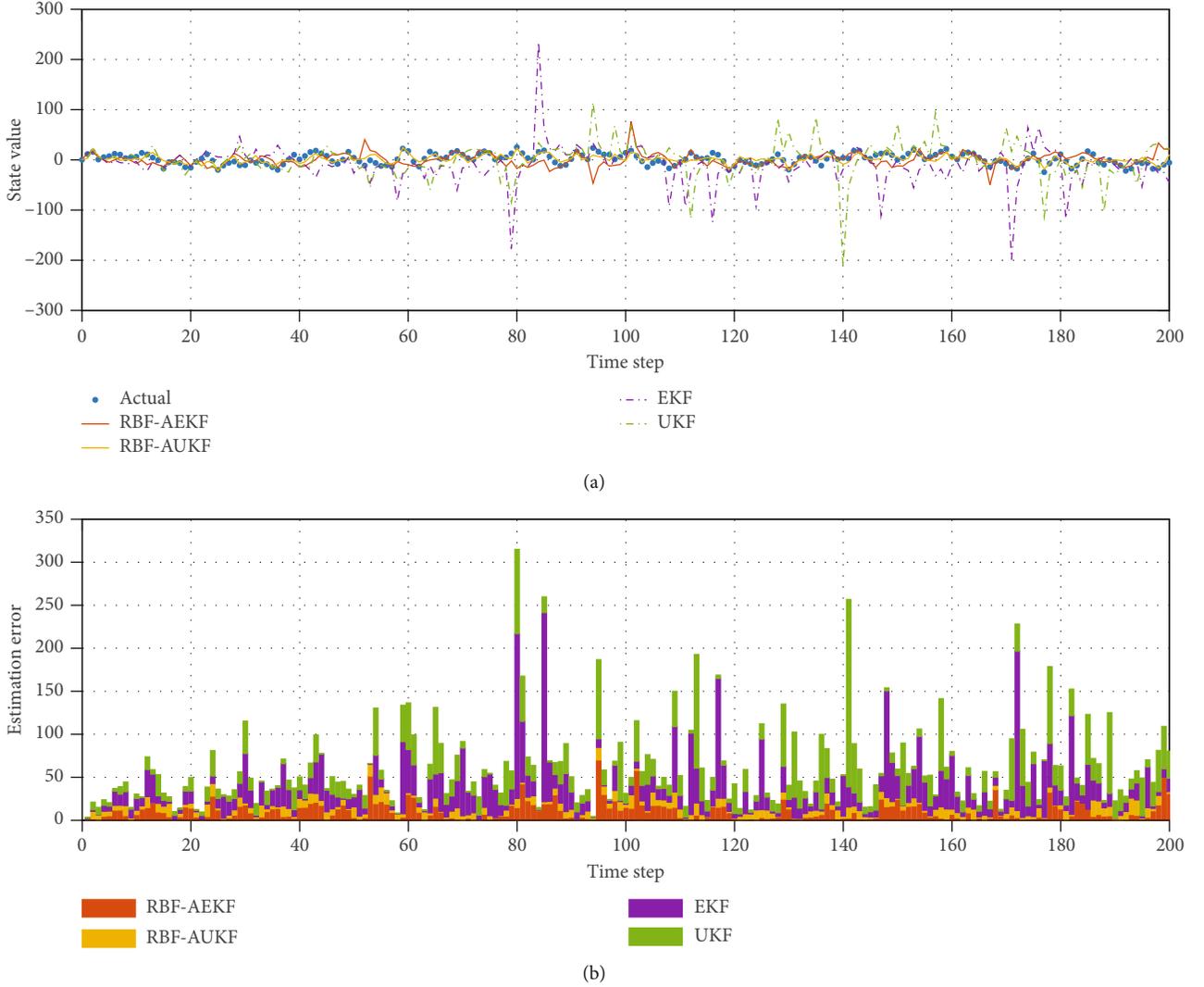


FIGURE 12: The fusion effects of different fusion algorithms.

TABLE 2: RMS errors of different fusion methods.

	Fusion method			
	EKF	UKF	RBF-AEKF	RBF-AUKF
RMS error	39.46	33.42	14.98	7.863

RGB-D camera and the kinematics calculation based on joint encoders. And the quaternion attitude estimations q_0, q_1, q_2, q_3 were obtained by fusion of the MARG sensor, visual odometer, and encoders. The experiment compares the original UKF method and the modified adaptive UKF method proposed in this paper, and the results are shown in Figures 13 and 14 and Table 3.

It can be seen from Table 3 that the RMS error of the pose estimation after multisensor fusion is smaller than that of a single sensor. Compared with that of the original method, the RMS error of the modified method proposed in this paper is significantly reduced in position estimation, but there is little difference in attitude estimation. The main reason is the modified method proposed in this paper is better than the

traditional method due to its better adaptive effect when the noise fluctuation is large. In this experiment, the overall fluctuation of the sensor in attitude measurement is smaller than that of position measurement, so the modified method is close to the original method in attitude estimation.

It can be seen from Figure 13 that the visual odometer is prone to local sudden changes due to environmental interference and the encoder is prone to measurement offset. Compared with the original method, the modified method handles the measurement data of local sudden change better, so the overall RMS error is smaller. And because of the combination of absolute and relative updates, the overall update rate is maintained. In this experiment, the original method was updated only 258 times, while the modified method was updated 479 times, so the real-time response rate of state estimation was improved. It can be seen from Figure 14 that the rotation angle of the desired trajectory on the X and Y axes is small, and the rotation angle on the Z -axis is large. When the rotation angle is small, it can be seen that the sensor has obvious measurement drift, when the rotation

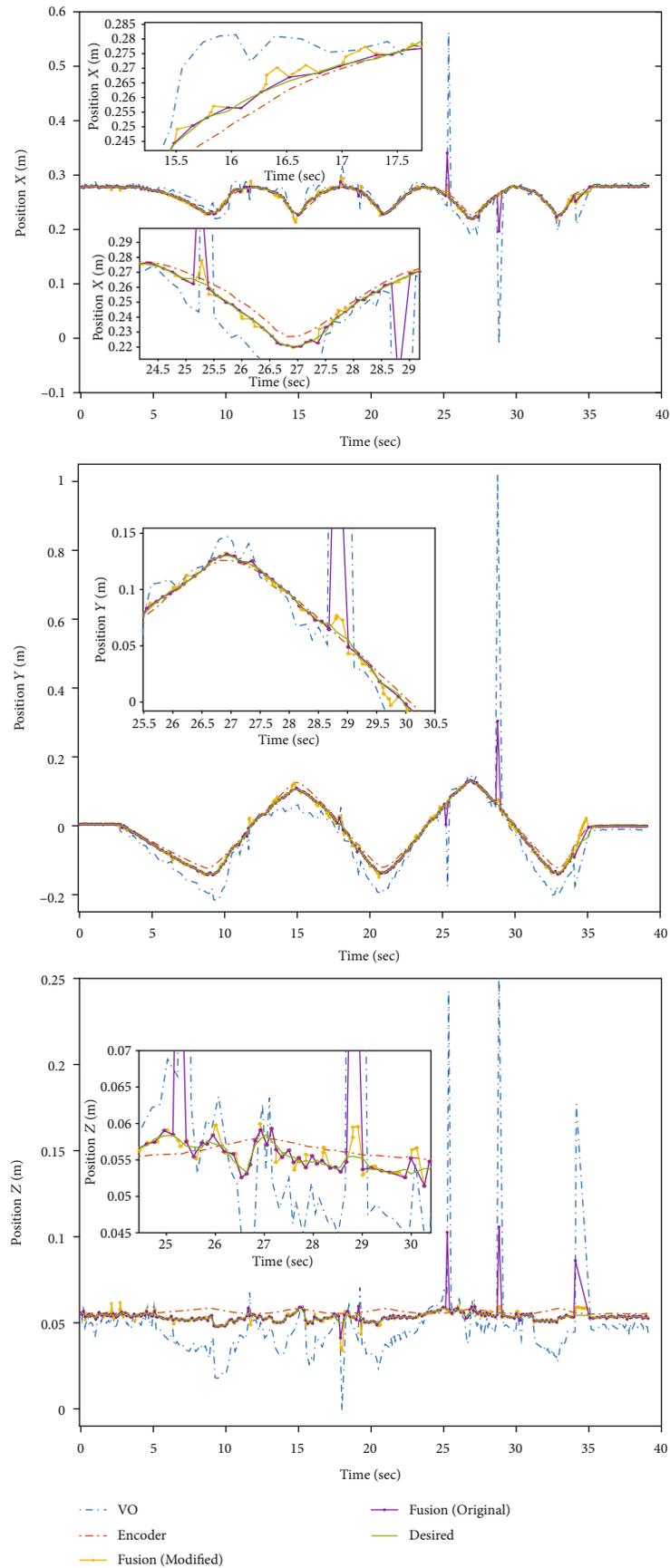


FIGURE 13: Results of position estimation.

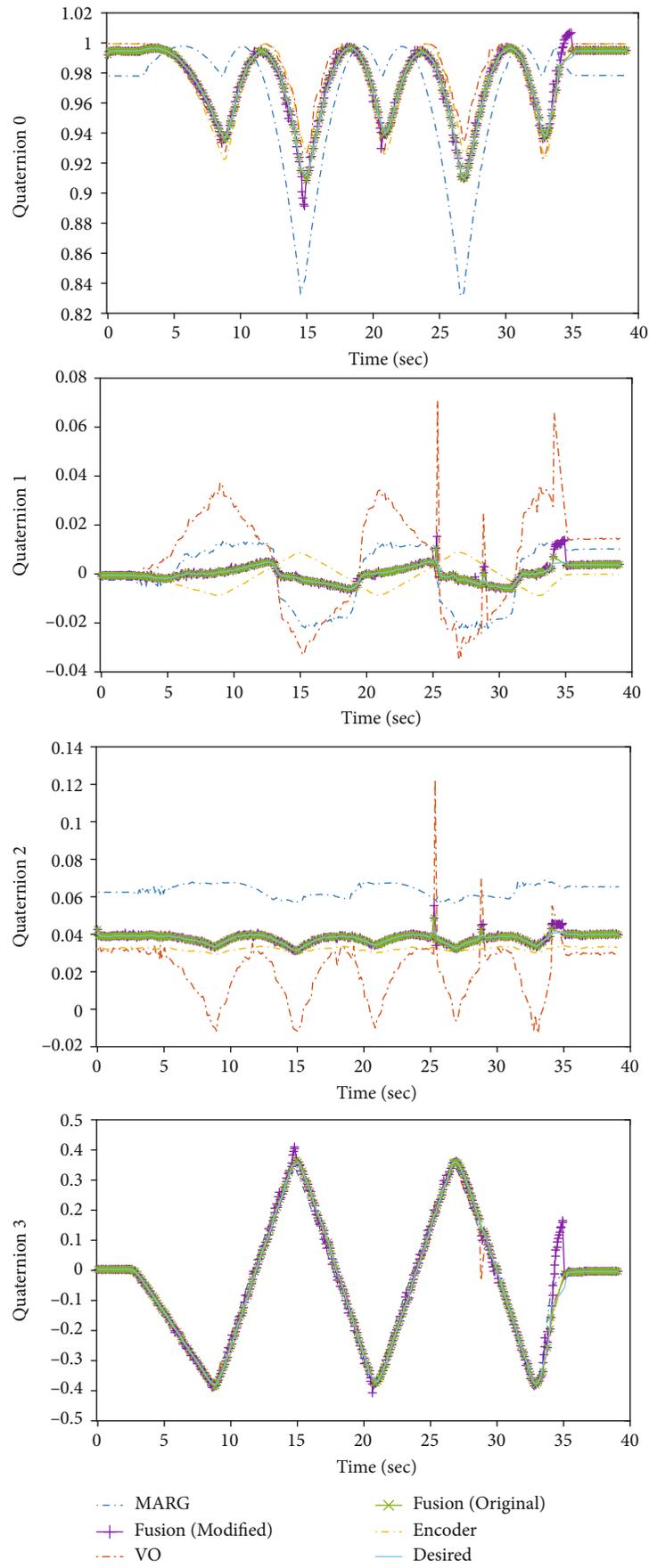


FIGURE 14: Results of attitude estimation.

TABLE 3: RMS errors of the original method and modified method.

RMS error	Estimated variable						
	P_x (cm)	P_y (cm)	P_z (cm)	q_0	q_1	q_2	q_3
MARG	—	—	—	0.319	0.101	0.260	0.273
VO	2.76	7.02	2.19	0.109	0.194	0.220	0.232
Encoder	0.303	1.08	0.360	0.0933	0.0624	0.0584	0.133
Fusion (modified)	0.175	0.364	0.108	0.00805	0.00564	0.00645	0.0482
Fusion (original)	0.663	1.59	0.475	0.00805	0.00564	0.00645	0.0482

angle is large, it is easy to produce local sudden changes. The modified method in this paper can compensate the measurement drift of each sensor, reduce the influence of local sudden changes on the state estimation, and thus can better follow the desired trajectory.

6. Conclusions

In order to improve the accuracy of the end-effector pose estimation, this paper proposes a modified method based on multisensor fusion, including preenhancement and adaptive fusion. The following conclusions can be drawn from the experimental results. Firstly, the depth image obtained by the RGB-D camera and the attitude obtained by the MARG sensor is enhanced, which can make the multisensor fusion process in a complex environment more stable. Secondly, the synchronization of sensors has a great influence on the results of multisensor fusion. The control variable is relatively updated when asynchronous, and the state variable is absolutely updated when synchronous, which can balance the update frequency and fusion accuracy. Thirdly, predicting the noise matrix through a neural network can make the algorithm adjust the fusion ratio of each sensor adaptively according to changes in the environment, which helps improve robustness. Finally, in the simulation environment, the RMS errors of the original UKF and EKF method are 4.25 times and 5.02 times that of the modified RBF-UKF method, respectively, and in the actual environment, the average pose estimation errors of the single sensor and the original UKF method are 8.57 times and 3.91 times that of the modified RBF-UKF method, and the update frequency of the original method is only 54% of the modified method. In summary, the method proposed in this paper can improve the accuracy of end-effector pose estimation and has high application value and practical engineering value.

Data Availability

The data used to support the findings of this study are available from the corresponding author upon request.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This work was supported in part by the National Key R&D Program of China under Grant 2018YFB1307400 and in part by the National Natural Science Foundation of China under Grant 61873267.

References

- [1] G. S. Chirikjian and J. W. Burdick, "The kinematics of hyper-redundant robot locomotion," *IEEE Transactions on Robotics and Automation*, vol. 11, no. 6, pp. 781–793, 1995.
- [2] M. H. Korayem and H. Ghariblu, "Maximum allowable load on wheeled mobile manipulators imposing redundancy constraints," *Robotics and Autonomous Systems*, vol. 44, no. 2, pp. 151–159, 2003.
- [3] M. H. Korayem, V. Azimirad, A. Nikoobin, and Z. Boroujeni, "Maximum load-carrying capacity of autonomous mobile manipulator in an environment with obstacle considering tip over stability," *International Journal of Advanced Manufacturing Technology*, vol. 46, no. 5–8, pp. 811–829, 2010.
- [4] X. Dong, D. Axinte, D. Palmer et al., "Development of a slender continuum robotic system for on-wing inspection/repair of gas turbine engines," *Robotics and Computer-Integrated Manufacturing*, vol. 44, pp. 218–229, 2017.
- [5] J. Q. Peng, W. F. Xu, T. L. Liu, H. Yuan, and B. Liang, "End-effector pose and arm-shape synchronous planning methods of a hyper-redundant manipulator for spacecraft repairing," *Mechanism and Machine Theory*, vol. 155, p. 25, 2021.
- [6] F. Renda, M. Giorelli, M. Calisti, M. Cianchetti, and C. Laschi, "Dynamic model of a multibending soft robot arm driven by cables," *IEEE Transactions on Robotics*, vol. 30, no. 5, pp. 1109–1122, 2014.
- [7] M. Rolf and J. J. Steil, "Efficient exploratory learning of inverse kinematics on a bionic elephant trunk," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 25, no. 6, pp. 1147–1160, 2014.
- [8] C. C. Cheah, M. Hirano, S. Kawamura, and S. Arimoto, "Approximate Jacobian control for robots with uncertain kinematics and dynamics," *IEEE Transactions on Robotics and Automation*, vol. 19, no. 4, pp. 692–702, 2003.
- [9] B. Xiao, S. Yin, and O. Kaynak, "Tracking control of robotic manipulators with uncertain kinematics and dynamics," *IEEE Transactions on Industrial Electronics*, vol. 63, no. 10, pp. 6439–6449, 2016.
- [10] M. H. Korayem, A. M. Shafei, F. Absalan, B. Kadkhodaei, and A. Azimi, "Kinematic and dynamic modeling of viscoelastic robotic manipulators using Timoshenko beam theory: theory and experiment," *International Journal of Advanced*

- Manufacturing Technology*, vol. 71, no. 5-8, pp. 1005–1018, 2014.
- [11] C. T. Lin, W. J. Zhang, and H. Yuan, “Investigation on the tip positioning accuracy of cable-driven serpentine manipulators,” *Applied Sciences-Basel*, vol. 10, no. 20, p. 17, 2020.
- [12] N. Riehl, M. Gouttefarde, S. Krut, C. Baradat, and F. Pierrot, “Effects of non-negligible cable mass on the static behavior of large workspace cable-driven parallel mechanisms,” in *IEEE International Conference on Robotics and Automation ICRA*, p. 2503, Kobe, Japan, 2009.
- [13] A. Fortin-Cote, P. Cardou, and A. Campeau-Lecours, “Improving cable driven parallel robot accuracy through angular position sensors,” in *2016 Ieee/Rsj International Conference on Intelligent Robots and Systems (Iros 2016)*, pp. 4350–4355, Daejeon, South Korea, 2016.
- [14] T. Dallej, M. Gouttefarde, N. Andreff, P. E. Herve, and P. Martinet, “Modeling and vision-based control of large-dimension cable-driven parallel robots using a multiple-camera setup,” *Mechatronics*, vol. 61, pp. 20–36, 2019.
- [15] X. Ma, P. W.-Y. Chiu, and Z. Li, “Shape sensing of flexible manipulators with visual occlusion based on Bezier curve,” *IEEE Sensors Journal*, vol. 18, no. 19, pp. 8133–8142, 2018.
- [16] M. H. Korayem, M. Yousefzadeh, and S. Kian, “Precise end-effector pose estimation in spatial cable-driven parallel robots with elastic cables using a data fusion method,” *Measurement*, vol. 130, pp. 177–190, 2018.
- [17] Z. Bing, L. Cheng, A. Knoll, A. Thong, K. Huang, and F. Zhang, “Slope angle estimation based on multi-sensor fusion for a snake-like robot,” in *2017 20th International Conference on Information Fusion (Fusion)*, Xian, People’s Republic of China, 2017.
- [18] Z. Gao and S. X. Ding, “Sensor fault reconstruction and sensor compensation for a class of nonlinear state-space systems via a descriptor system approach,” *IET Control Theory and Applications*, vol. 1, no. 3, pp. 578–585, 2007.
- [19] J. Kelly and G. S. Sukhatme, “Visual-Inertial sensor fusion: localization, mapping and sensor-to-sensor self-calibration,” *International Journal of Robotics Research*, vol. 30, no. 1, pp. 56–79, 2011.
- [20] S. A. Khalilpour, R. Khorrambakht, H. Damirchi, H. D. Taghirad, and P. Cardou, “Tip-trajectory tracking control of a deployable cable-driven robot via output redefinition,” *Multibody System Dynamics*, vol. 4, pp. 1–28, 2020.
- [21] A. Kuzdeuov, M. Rubagotti, and H. A. Varol, “Neural network augmented sensor fusion for pose estimation of tensegrity manipulators,” *IEEE Sensors Journal*, vol. 20, no. 7, pp. 3655–3666, 2020.
- [22] H. Sarbolandi, D. Lefloch, and A. Kolb, “Kinect range sensing: structured-light versus time-of-flight kinect,” *Computer Vision and Image Understanding*, vol. 139, pp. 1–20, 2015.
- [23] A. M. Sabatini, “Quaternion-Based extended Kalman filter for determining orientation by inertial and magnetic sensing,” *IEEE Transactions on Biomedical Engineering*, vol. 53, no. 7, pp. 1346–1356, 2006.
- [24] T. Moore and D. Stouch, “A generalized extended Kalman filter implementation for the robot operating system,” in *Intelligent Autonomous Systems 13, Advances in Intelligent Systems and Computing*, E. Menegatti, N. Michael, K. Berns, and H. Yamaguchi, Eds., pp. 335–348, Springer, 2016.
- [25] E. A. Wan and R. van der Merwe, “The unscented Kalman filter for nonlinear estimation,” in *Proceedings of the IEEE 2000 Adaptive Systems for Signal Processing, Communications, and Control Symposium (Cat. No. 00EX373)*, pp. 153–158, McMaster Univ, Lake Louise, Canada, 2000.
- [26] A. Smyth and M. L. Wu, “Multi-rate Kalman filtering for the data fusion of displacement and acceleration response measurements in dynamic system monitoring,” *Mechanical Systems and Signal Processing*, vol. 21, no. 2, pp. 706–723, 2007.