Synchronous Monitoring Method of Multi-manipulator Trajectory Signals Based on Machine Learning

Xiao-zheng Wan[1(✉)], Song Zhang[2], Ji-ming Zhang[1], Hui Chai[1], and Huan-yu Zhao[1]

[1] Institute of Oceanographic Instrumentation, Qilu University of Technology (Shandong Academy of Sciences), Qingdao 266061, China
wxz18661817997@163.com

[2] College of New Energy, China University of Petroleum, Qingdao 266580, China

Abstract. The traditional trajectory signal monitoring method uses a combination of sensors and mathematical models to detect the trajectory signal of the manipulator, which is susceptible to interference from uncertain factors, resulting in low monitoring accuracy and poor real-time performance. In response to the above problems, this research designed a method for synchronous monitoring of multi-manipulator trajectory signals based on machine learning. After the multi-manipulator dynamic model is established, the coordinate system of the trajectory acquisition equipment is calibrated according to the principle of binocular vision. After preprocessing the trajectory image of the manipulator, the CLDNN model is used to realize the synchronous monitoring of the trajectory signal. The simulation experiment results show that the monitoring method shortens the monitoring time by about 62.5%, and has higher monitoring accuracy, which proves that its performance is better.

Keywords: Machine learning · Multiple manipulators · Manipulator trajectory · Signal synchronization · Signal monitoring

1 Introduction

A mechanical arm is an automatic operating device that can imitate certain actions of a human arm to grasp, carry objects or operate tools in a fixed program, and it is an important branch of industrial robots. The robotic arm is an important equipment in automated production, with the advantages of high safety and long working hours. After more than half a century of development, the control accuracy of industrial manipulators has been continuously improved, its types have become more and more abundant, and the application fields have also continued to expand.

There is a strong coupling between multiple manipulators, so although there are independent controllers for each mobile manipulator, each mobile manipulator should not be controlled separately, but should be considered as a whole [1]. It is an effective way to improve the control precision of multi-manipulator to monitor the error between its

S. Wang et al. (Eds.): IoTCare 2021, LNICST 415, pp. 129–139, 2022.
https://doi.org/10.1007/978-3-030-94182-6_10

moving trajectory and the theoretical trajectory of control instruction when controlling multi-manipulator.

Traditional monitoring method based on mathematical model is used to establish a mathematical model of the moving trajectory of the multi-manipulator, and the signals are monitored by comparing the difference between the feedback of the manipulator and the theoretical value of the model [2]. However, because the robotic arm system is a system with multiple inputs and multiple outputs, and has strong nonlinearity and strong coupling, these characteristics and the interference existing outside the system make the mathematical model of the robotic arm must have some uncertain factors, which is difficult Ensure the accuracy of the monitoring method. Some scholars also proposed a sensor-based tracking signal monitoring method for multi-manipulator. In this method, a variety of sensors are installed on the manipulator arm, and the tracking signals of the manipulator arm are monitored by analyzing the signals sent back by the sensors [3]. The accuracy of this monitoring method is not only limited by the accuracy of the sensor, but also the sensor will be interfered in the complex working environment of the robotic arm, and the real-time monitoring is difficult to guarantee.

With the development of computer vision, machine learning and sensor technology, various intelligent robot systems relying on computer vision, machine learning and advanced sensors are constantly coming out. The use of machine learning can improve the monitoring efficiency of the monitoring method, and can monitor multiple objects at the same time [4]. According to the above analysis content, in order to improve the real-time performance of trajectory signal monitoring, this article will take the multi-mobile manipulator as the research object, and study the method of synchronous monitoring of multi-manipulator trajectory signal based on machine learning.

2 Method Design

2.1 Multi-manipulator Motion Analysis and Dynamic Modeling

In order to be able to monitor the trajectory signals of multiple manipulators simultaneously, a mathematical model of the multi-mobile manipulator system must be established first. Usually, a fixed coordinate system is defined on each link of the robot arm, and two adjacent coordinate systems can represent each other, and their relative positions are generally described by the transformation matrix between the two adjacent links. The connecting rods are numbered starting from the fixed base of the robotic arm. Generally, the coordinate system at the fixed base is O_o, and so on, so the coordinate system on the connecting rod i is O_i.

For the robotic arm, the coordinate system of all links can be established in sequence according to the following steps [5]:

Find out the corresponding joint axis of each link and extend it in the direction of its axis.

Find the common perpendicular between the joint axis $i-1$ and the joint axis i or the intersection point of the joint axis $i-1$ and the joint axis i, and use the intersection point of the common perpendicular line and the joint axis or the intersection point of the joint axis $i-1$ and the joint axis i as the connection The origin of the rod coordinate system O_i is used to establish the coordinate system.

Specify the direction of the Z_i axis as the direction of the i axis.

Specify that the X_i axis is along the direction of the common perpendicular between the joint axis $i-1$ and the joint axis i, and the joint axis $i-1$ points to the joint axis i. If the joint axis $i-1$ and the joint axis i intersect, the X_i axis is specified to be perpendicular to the joint axis $i-1$ And the plane where the joint axis i lies.

The Y_i axis direction is determined by the right-hand rule based on the X_i axis and Z_i axis directions.

In special cases, when the first joint variable is 0, the coordinate system O_o and the coordinate system O_1 coincide. For the coordinate system O_n, the origin and the direction of the X_n axis can be selected arbitrarily. When selecting, the connecting rod parameter is usually selected as 0.

It can be seen from the establishment rules of the coordinate system that when the axis Z_i and the axis of the joint i coincide, there are two choices for the direction of Z_i: the axial direction and the reverse extension direction. You can choose one of them without affecting the final result. When two joint axes intersect, it is impossible to determine which joint axis is in front and which is behind. The direction of the X_i axis cannot be defined according to the rules. You can choose between two possibilities, and then determine the direction of the y axis based on the right-hand rule based on the previous determination., When the joint axis $i-1$ is parallel to the joint axis i, the origin position of the coordinate system O_i can be arbitrarily selected.

The relationship between the linear velocity of the end of the robotic arm in Cartesian space and its joint angular velocity in the joint space is expressed by the Jacobian matrix.

The hypothesis of the expected trajectory at the end of the robotic arm is shown in Eq. (1):

$$x = x(q) \tag{1}$$

In the formula, x represents the trajectory of the end of the robotic arm in Cartesian space, and q represents the rotation angle of each joint of the robotic arm in the joint space. Take the derivation operation on both sides of formula (1) to time respectively to obtain the relationship between q and x as shown in formula (2):

$$\dot{x} = J(q)\dot{q} \tag{2}$$

In the formula, $\dot{x}$ is the speed of the end of the manipulator in Cartesian space; $\dot{q}$ is the angular velocity of the joint rotation of the manipulator in the joint space; $J(q)$ is the Jacobian matrix of the manipulator. The elements in row i and column j of the matrix are:

$$J_{ij}(q) = \frac{\partial x_i(q)}{\partial q_j} \tag{3}$$

The multi-manipulator dynamic model studied in this paper is obtained on the basis of the Lagrangian function method. The kinetic energy K of the system is defined as the sum of the translational kinetic energy and the rotational kinetic energy of each link, and its expression is as follows:

$$K = \sum_{i=1}^{n} \left(\frac{1}{2} I_i \omega_i^2 + \frac{1}{2} m_i \dot{r}_i^2 \right) \tag{4}$$

The potential energy P is defined as the sum of the gravitational potential energy of each connecting rod and its expression is as formula (5):

$$P = \sum_{i=1}^{n} m_i g r_{iy} \tag{5}$$

In the above formula, I_i is the central inertia tensor of the connecting rod; ω_i is the angular velocity of the connecting rod in the inertial coordinate system; $\dot{r}$ is the linear velocity of the connecting rod in the inertial coordinate system; r_{iy} is the direction along the y-axis in the inertial coordinate system Weight.

Bring the total kinetic energy and total potential energy equation of the robot arm link into the Lagrangian function, and then obtain the partial derivative of it, and finally the dynamic equation of the robot arm can be obtained as shown in the following formula [6]:

$$D(q)\ddot{q} + B(q, \dot{q})\dot{q} + G(q) = u \tag{6}$$

In the formula, $D(q)$ is the inertia matrix of the multi-manipulator, $B(q, \dot{q})$ is the centrifugal force and Ge force matrix of the multi-manipulator, $G(q)$ is the gravity term matrix, q is the current joint angle vector of the manipulator, and u is the controller output in the controlled manipulator. The driving torque on each joint. After the above-mentioned multi-manipulator motion trajectory model is established, the visual acquisition equipment that collects the motion trajectory of the mechanical arm is calibrated.

2.2 Trajectory Acquisition Equipment Dual Target Positioning

The pinhole imaging model of the camera actually uses the above geometric relationship constraints to simulate the image formation process, that is, to establish the coordinate conversion relationship between the two-dimensional image collected by the binocular camera and the three-dimensional objective object. The ideal imaging model of the camera is the pinhole imaging model, which does not consider the distortion of the camera in reality. There are four coordinates in this model: the world coordinate system, the camera coordinate system, the image coordinate system and the pixel coordinate system [7].

In an ideal camera imaging model, the default pixel plane is parallel to the image plane, and the pixel coordinate system is known, that is, the coordinates of point R in the two-dimensional digital image, and then the points of pixel $R(u, v)$ and world coordinates are derived in turn The derived relationship between $R_w(x_w, y_w, z_w)$. Due to the manufacturing process of the camera, it cannot guarantee that the imaging plane of the camera is absolutely perpendicular to the optical axis of the camera. From the imaging situation of the camera under ideal conditions, the mathematical derivation of the generalized pinhole imaging model can be derived as follows:

$$\begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \frac{1}{z_w} \begin{bmatrix} f_x & -f_x/\tan\theta & u_0 & 0 \\ 0 & f_y/\sin\theta & v_0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} R_{3\times3} & T_{3\times1} \\ 0 & 1 \end{bmatrix} \begin{bmatrix} x_w \\ y_w \\ z_w \\ 1 \end{bmatrix} \tag{7}$$

In the formula, θ is the angle between the imaging plane and the optical axis of the camera; f_x, f_y, u_0, and v_0 are collectively referred to as the camera internal parameters; $R_{3\times3}$ is the camera internal parameter matrix; $T_{3\times1}$ is the camera's projection transformation matrix. After calibrating the multi-mechanical ratio trajectory image acquisition device, the acquired trajectory image is processed.

2.3 Robotic Arm Trajectory Image Processing

According to the brightness component of the pixel in the robot arm trajectory image, the conversion relationship between RGB and YUV color space is constructed as shown in the following formula [8].

$$Y(i,j) = \alpha_1 R(i,j) + \alpha_2 G(i,j) + \alpha_3 B(i,j) \tag{8}$$

In the formula, (i, j) is the position of any pixel in the image; α_1, α_2, and α_3 correspond to the weights of the three color components of R, G, and B in the RGB color space; $R(i,j)$, $G(i,j)$, and $B(i,j)$ are the brightness values of the three components of the image; $Y(i,j)$ is the grayscale value of the image after grayscale conversion. After converting the image gray level, a Gaussian low-pass filter is used to process image noise.

For image denoising processing of size $M \times N$, the transfer function of the two-dimensional Gaussian low-pass filter is as follows:

$$\begin{cases} H(u,v) = e^{-\frac{D(u,v)^2}{2\sigma^2}} \\ D(u,v) = \left[\left(u - \frac{M}{2}\right)^2 + \left(v - \frac{N}{2}\right)^2\right]^{\frac{1}{2}} \end{cases} \tag{9}$$

In the formula, (u, v) is the image pixel of the input Gaussian low-pass filter; $H(u, v)$ is the filter transfer function; $D(u, v)$ is the distance from the center of the frequency rectangle; σ is the variance of the image gray value. Image denoising processing will result in blurring of the edges of the image. Sobel operator is used to detect the edges of the image. The calculation formula of Sobel operator is as follows [9]:

$$\nabla f = \begin{bmatrix} \frac{\partial f}{\partial x} & \frac{\partial f}{\partial y} \end{bmatrix}^T \tag{10}$$

$$mag(\nabla f) = \left[\left(\frac{\partial f}{\partial x}\right)^2 + \left(\frac{\partial f}{\partial y}\right)^2\right]^{\frac{1}{2}} \tag{11}$$

$$\partial(x,y) = \arctan\left(\frac{\frac{\partial f}{\partial x}}{\frac{\partial f}{\partial y}}\right) \tag{12}$$

In the formula, ∇f is the image gradient; $mag(\nabla f)$ is the image gradient amplitude; $\partial(x, y)$ is the image direction angle.

After image preprocessing, the image is transformed from the spatial domain to the fuzzy domain according to the following formula:

$$\mu_{ij} = \tan\left(\frac{x_{ij} - x_{\min}}{x_{\max} - x_{\min}}\right) \tag{13}$$

$$\begin{cases} C = \gamma \frac{|\mu_{ij} - \overline{\mu}_{ij}|}{|\mu_{ij} + \overline{\mu}_{ij}|} \\ C' = \varphi(C) \end{cases} \tag{14}$$

In the formula, x_{ij} is the gray value of the image; μ_{ij} is the membership value of the fuzzy domain of the image; C is the local image contrast, which is also the fuzzy enhancement operator; $\varphi(\,)$ is the inverse transform function of the fuzzy domain; γ is the fuzzy parameter of the fuzzy enhancement. Generally, in order to obtain high-definition images of the robotic arm trajectory, the camera needs to continuously shoot multiple high-definition images. In order to improve the efficiency of track signal monitoring, the image is compressed.

2.4 Image Compression Processing

After processing the robot arm trajectory image, the PLOT compression algorithm is used to compress the image signal to reduce the image processing dimension. The essence of the PLOT compression algorithm is to find all the main trends in the collected signal, and use the segmented linear trend to replace the original collected data to achieve the best possible data compression effect. For the prediction time series object y_t, suppose its model is a discrete time signal sampled at equal intervals as shown in Eq. (15) [10].

$$y_t = \mu_t + \varepsilon_t \tag{15}$$

In the formula, μ_t is the expected value of the process at sampling time t, ε_t is the error between the expected and actual values, and $[t_{j-1}, t_j]$ represents the sampling time period. Assuming that ε_t is a random variable and obeys an independent normal distribution $N(0, \sigma^2)$, the trend of the process mean μ_t can be approximated by a piecewise linear expression. If the trend of the time series changes at time t_{j-1}, then in the j th sampling time period $[t_{j-1}, t_j]$, the specific form of the linear expression of the analog μ_t is shown in Eq. (16).

$$\mu_t = \mu_{t_{j-1}} + \delta_{t_{j-1}} + \beta_j(t - t_{j-1}) \tag{16}$$

In the formula, $\delta_{t_{j-1}}$ is the change of the process expectation value at time t_{j-1}, and β_j is the change slope of the process expectation value during the sampling period of $[t_{j-1}, t_j]$.

The core idea of the PLOT data compression algorithm is to simulate the linear function corresponding to all sampling time period $[t_{j-1}, t_j]$ as accurately as possible, not only to detect the linear change trend, but also to identify the linear trend change point. The time t_j, t_j is the starting point of the next sampling time period. time. In the process of piecewise linear estimation, it is necessary to detect and process the abnormal point data in the sampled signal at the same time to reduce the influence of abnormal points on the linear trend estimation. The linear function in the sampling time period is obtained by linear fitting all data points in the current time period using the least square algorithm, and the new sampling point must be re-fitted in the same way after the new sampling point appears. At the same time, the least squares estimate the variance of the fitting process through the mean square error. The judgment of the segment interval point is to use the linear function in the current time period to predict the next time sampling point, and then judge whether the new data point falls within the prediction interval. If it falls within it, it will not be processed. It is only used to refit the data trend. If it falls outside the prediction interval, it means that the data trend has changed or is an abnormal point at the next moment, and then processed accordingly. Use the neural network structure in machine learning to realize the synchronous monitoring of multi-manipulator trajectory signals.

2.5 Realize the Simultaneous Monitoring of Multiple Robotic Arms Trajectory Signals

This paper constructs a CLDNN model accordingly. The structure of the CLDNN network is to connect several layers of CNN after the input layer to extract local features. The output of the CNN is poured into several layers of LSTM units to reduce time domain changes. The output of the last layer of LSTM is input to the fully connected layer, with the purpose of adding features The space is mapped to an output layer that is easier to classify. After the convolutional layer, the BN layer is deployed for normalization to avoid over-fitting. The fully connected layer connects all features and transmits the output value to the softmax classifier.

The following cross entropy loss function is selected as the loss function of the model:

$$J(\theta) = -\frac{1}{n}\sum_{i=1}^{n}\left[y_i \ln \hat{y}_i - (1-y)\ln\left(1-\hat{y}_i\right)\right] \tag{17}$$

The original cross entropy is used to indicate the degree of similarity between two distributions. Here it is used to indicate the degree of similarity between the distribution of the true value y and the distribution of the network predicted value $\hat{y}$. The smaller the cross entropy, the more similar the two.

Considering that the depth of the network has an extremely important impact on the classification efficiency of the network itself, after the number of network layers continues to increase, the method of backpropagation will cause the gradient of the parameters to gradually become diffuse during the convergence process, and even disappear directly, resulting in The parameters do not converge for a long time. Let the network fit the residual function $F(x) = H(x) - x$ instead of fitting $H(x)$, so that the original function becomes $F(x) + x$. In this article, the residual structure block is defined as:

$$y = F(x, \{W_i\}) + x \tag{18}$$

In the formula, x and y respectively represent the input and output vectors of the structure block, and the function $F(x, \{W_i\})$ represents the residual mapping that needs to be learned. With the minimum output of the residual block as the goal, the convolutional network is trained with sample data to determine the parameters of the neural network.

The processed multi-manipulator trajectory image is output to the convolutional neural network, and the trajectory in the image is matched with the preset mechanical arm motion trajectory under the action of the activation function in the hidden layer. Determine the error between the actual robot trajectory and the theory, so as to realize the monitoring of the trajectory signal. So far, the research on the synchronous monitoring method of multi-manipulator trajectory signals based on machine learning has been completed.

3 Test Experiment

3.1 Experiment Content

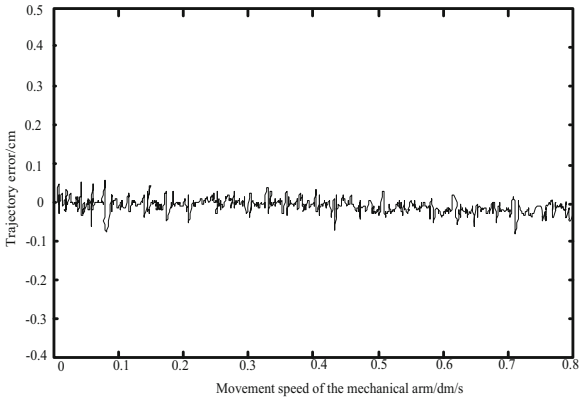
This experiment is a comparative experiment. The synchronous monitoring method of multi-manipulator trajectory signals based on machine learning studied above is used as the experimental group, and the traditional trajectory signal monitoring method is used as the comparison group. By comparing the error and response time of the two signal monitoring methods, the reliability of the monitoring method studied above is verified.

3.2 Experimental Environment

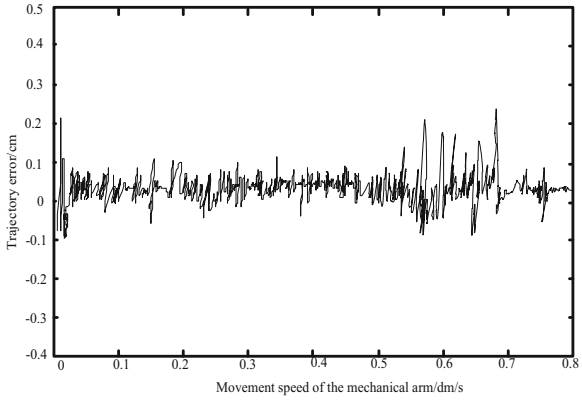
The experimental platform includes Denso VP6242G and Quanser's open structure control module. The control module has the functions of all industrial systems and serves as an interface for interaction with the QUARC software. The closed loop frequency of the controller is 4kHz. Users can adjust the gain of the controller through the QUARC software, or design their own controller in the simulation environment, and directly use the completely open commands to bypass the internal PID controller. The experimental platform realizes the rotation of each joint through the servo motors at the joints and is equipped with corresponding measurement sensors. The measurement data controller and the QUARC software interact to realize the reproduction of the end trajectory of the manipulator.

3.3 Experimental Result

The comparison results of the two trajectory signal monitoring methods to monitor the trajectory error of the multi-manipulator are shown in Fig. 1.



(a) Experimental group method monitoring error



(b) Comparison group method monitoring error

Fig. 1. Comparison of monitoring errors

From the analysis of Fig. 1, it can be seen that compared with the error change curve of the contrast group, the monitoring error of the method in the experimental group for the trajectory of the multi-manipulator is significantly smaller. Although the monitoring errors of the two groups were within the allowable range, the fluctuation of the monitoring error curve of the contrast group method was significantly more severe than that of the experimental group. Therefore, the monitoring performance of the proposed method is better.

When the two methods are monitoring the trajectory of the same parameter manipulator, the response time of the monitoring method is shown in Table 1.

Table 1. Comparison of response time of monitoring methods/ms

Serial number	Experimental group method	Comparison group method
1	7.32	17.93
2	6.97	18.66
3	6.94	18.74
4	6.99	17.88
5	7.02	19.63
6	6.96	19.71
7	7.23	19.85
8	7.04	17.82
9	7.02	19.34
10	6.86	17.89

By analyzing the data in Table 1, it can be seen that the response time of the method in the experimental group in monitoring the trajectory signal of the manipulator is less than that of the method in the contrast group. After averaging the data in the table, it was found that the average response time of the experimental group method was 7.035 ms, and the average response time of the contrast group method was 18.745 ms. The above results indicate that the trajectory signal monitoring method studied in this paper has higher monitoring accuracy, real-time performance and better performance.

4 Concluding Remarks

Industrial manipulator is an industrial automation equipment which includes machinery, communication, automatic control, computer and other disciplines. It plays an important role in high-end manufacturing industry. In order to improve the control accuracy of the manipulator, this paper studies a multi-manipulator trajectory signal synchronous monitoring method based on machine learning in view of the problems existing in the traditional trajectory signal monitoring methods. Based on the establishment of dynamic model of multi-manipulator, this method calibrates the coordinate system of trajectory acquisition equipment by using binocular vision principle. Then, after preprocessing the trajectory image of the manipulator, the CLDNN model is used to realize the synchronous monitoring of the trajectory signal. This study also proves that the method has better performance through experiments.

References

1. Zhu, Z., et al.: Obstacle avoidance path planning of space redundant manipulator based on a collision detection algorithm. J. Northwest. Polytech. Univ. **38**(01), 183–190 (2020)

2. Ren, W., Jiang, M.: Research on trajectory planning algorithm of 6-DOF manipulator. J. Anhui Polytech. Univ. **34**(06), 52–59 (2019)
3. Qian, Q., Zhang, A., Sun, Y.: Event triggered adaptive robust trajectory tracking control for multi-joint manipulators. Acta Armamentarii **40**(08), 1732–1739 (2019)
4. Xu, Y., et al.: Application of improved LMD algorithm in signal detection of power quality disturbance in microgrid. Power Syst. Technol. **43**(01), 332–341 (2019)
5. Li, Y., et al.: Research on model of active tiles in data centers based on machine learning. Comput. Simul. **36**(12), 180–185+248 (2019)
6. Chen, H., Li, L.: Trajectory planning of time optimal manipulator based on complex method. J. Mech. Transm. **43**(03), 72–75+94 (2019)
7. Ma, X., Jiang, X.: Analysis for the generalization ability of machine learning based full reference image quality metrics. J. Commun. Univ. China Sci. Technol. **26**(04), 42–49 (2019)
8. Zhou, X., He, X., Zheng, C.L Radio signal recognition based on image deep learning. J. Commun. **40**(07), 114–125 (2019)
9. Liu, S., et al.: Parallel fractal compression method for big video data. Complexity **2018**, 2016976 (2018)
10. Liu, S., Fu, W., He, L., Zhou, J., Ma, M.: Distribution of primary additional errors in fractal encoding method. Multimed. Tools Appl. **76**(4), 5787–5802 (2014). https://doi.org/10.1007/s11042-014-2408-1