SIMULTANEOUS OBSTACLE PREDICTION AND REAL-TIME ADAPTIVE TRAJECTORY PLANNING FOR ROBOTIC MANIPULATORS IN DYNAMIC ENVIRONMENTS

Wei LIU[1], Tingyu ZHOU[2*], Cheng JING[3], Yongheng MA[4]

In order to improve the obstacle avoidance ability, the continuity and adaptability of re-planning trajectory of robotic manipulators in narrow dynamic environments, a real-time adaptive motion planning strategy of 6R manipulator is proposed. The detection model of unknown obstacle position and its motion trend is established, and the segment sphere envelope box model is proposed to detect the collision. According to the interference relationship between the trajectory of manipulator and potential unknown obstacles, the trajectory of manipulator is planned and adjusted in real time. MOPSO(Multi-Objective Particle Swarm Optimization) based on R2 index selection (R2SMMOPSO) strategy is used to optimize the objective functions.

Keywords: Collision detection; Multi-objective optimization; KF(Kalman Filter); R2SMMOPSO

1 Introduction

Robots occupy an increasing proportion in production and manufacturing. Robot arm motion planning [1-2] is an indispensable part in robotic technology. The motion planning technology of a manipulator mainly consists of obstacle avoidance technology and trajectory planning technology.

[1] PhD, School of Automotive Engineering, Yancheng Institute of Technology, China

[2*]PhD, School of Automotive Engineering, Yancheng Institute of Technology, China (Corresponding author), e-mail: tyrong23557@163.com

[3] PhD, School of Automotive Engineering, Yancheng Institute of Technology, China

[4] PhD, School of Automotive Engineering, Yancheng Institute of Technology, China

Effective prediction of the trajectory of potential unknown obstacles can help the manipulator avoid obstacles and produce smoother and safer trajectories [3-4]. Wang [5] proposed an improved potential field algorithm for trajectory planning of multi degree of freedom robots. Das [6] integrates the new FK(Forward Kinematics) kernel into the proxy collision inspector Fastron, which improved the speed of obstacle avoidance. Some studies optimize the trajectory [7-8] of the end-effector by improving the multi-objective trajectory planning algorithm and optimal planning technology. There are also studies to improve the accuracy of trajectory tracking by putting forward a new trajectory planning technology [9-10], which can make the control system more stable and avoid obstacles effectively. Research has been done to improve the efficiency of trajectory calculation and control stability of a system by proposing a new system control framework and layering the system [11-12]. Jin [13] presents a method of combining damped least squares (DLS) with compensation control to eliminate singularity, and solves the puzzle by using chaotic particle swarm optimization (CPSO) algorithm.

Previous studies have shown that simultaneous obstacle avoidance and trajectory planning of a manipulator can improve the timely accuracy of obstacle avoidance and the smoothness of trajectory operation [14]. At present, in narrow dynamic indoor environments, the requirement for the ability of self-regulation of motion trajectory and the ability to adapt to the environment will be higher. Our contribution to the development of knowledge in the approached field includes the method of recognition and location of typical mechanical parts based on the RGB-D camera, and an improved autonomous navigation strategy in narrow indoor environments. A real-time adaptive motion planning strategy is proposed in this manuscript. While effectively avoiding obstacles, optimize the operation trajectory through the multi-objective particle swarm optimization algorithm refered to R2 index strategy, so as to strengthen the smoothness of the real-time task trajectory of the manipulator.

2 Robot Modeling

2.1 Kinematic Analysis of Robots

It is feasible to use four kinematics parameters to described a mechanical arm: angle between two connecting rods θ、Two-link distance d, link twist angle α、Connector length a. The D-H parameters of the IRB 120 robot are listed in Table 1.

Table 1

The D-H parameters of IRB 120 robot

Connecting rod	θ_i/(°)	d_i/(mm)	a_i/(mm)	α_i/(°)	Joint angle range/(°)
1	θ_1	290	0	-90	[-165,165]
2	θ_2-90	0	270	0	[-110,110]
3	θ_3	0	70	-90	[-110,70]
4	θ_4	302	0	90	[-160,160]
5	θ_5	0	0	-90	[-120,120]
6	θ_6-180	72	0	0	[-400,400]

Based on the D-H parameter method, the transformation matrix of the connecting rod coordinate system {i} relative to {i-1} is:

$$
{}_{i}^{i-1}T = Rot(x,\alpha_{i-1})Trans(a_{i-1},0,0)Rot(z,\theta_i)Trans(0,0,d_i) =
\begin{bmatrix}
\cos\theta_i & -\sin\theta_i & 0 & \alpha_{i-1} \\
\sin\theta_i\cos\alpha_{i-1} & \cos\theta_i\cos\alpha_{i-1} & -\sin\alpha_{i-1} & -d_i\sin\alpha_{i-1} \\
\sin\theta_i\sin\alpha_{i-1} & \cos\theta_i\sin\alpha_{i-1} & \cos\alpha_{i-1} & d_i\cos\alpha_{i-1} \\
0 & 0 & 0 & 1
\end{bmatrix} \tag{1}
$$

2.2 Collision detection model

The segment sphere envelope method is used to simplify the collision between manipulator and potential unknown obstacles, and then transforms the collision detection problem into the position relationship between spheres and cylinders.

2.2.1 Collision between spheres

The manipulator at the end of IRB120 robot and irregular obstacles can envelope through the sphere and carry out corresponding collision detection, as shown in Fig. 1

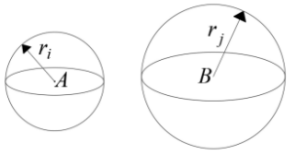


Fig. 1 Sphere and sphere

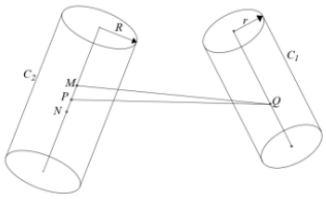


Fig. 2 Positional relationship between cylinders

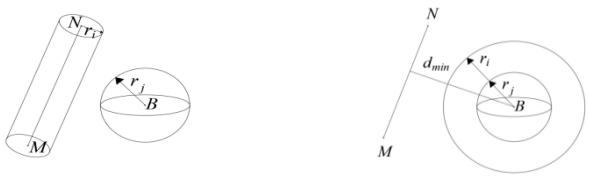
2.2.2 Collision between a cylinder and a cylinder

When the connecting rod of IRB120 collides with the obstacle enveloped by the cylinder, the corresponding collision can be transformed into the collision

between the cylinders. The positional relationship between the cylinders is shown in Fig. 2.

2.2.3 Collision between a sphere and a cylinder

The connecting rod and joint of IRB120 robot can be represented by cylinder and sphere respectively. The positional relationship is shown in Fig. 3 (a):



(a)Cylinder and sphere (b)Segment and sphere

Fig. 3 Diagram of the relationship between sphere and column positions

3 Real-time Adaptive Motion Planning Strategy

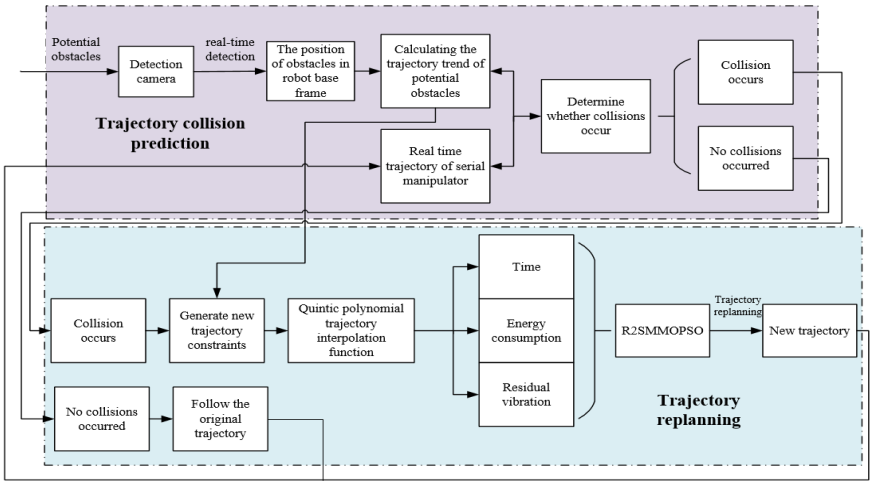


Fig. 4 Real-time adaptive motion planning control process in one time step

It can be seen that the proposed real-time adaptive motion planning strategy is mainly divided into two parts: trajectory collision prediction and trajectory replanning. The trajectory collision prediction part detects the potential obstacles around the manipulator in real time through the camera, and predicts the motion trajectory of the potential obstacles through the prediction step of Kalman filter. When a collision is predicted, the trajectory replanning part will generate new trajectory constraints of the manipulator in real time according to the trajectory trend of potential obstacles. Aiming at time, energy consumption and

residual vibration, the trajectory interpolation function is optimized through R2SMMOPSO to obtain a real-time trajectory that can avoid obstacles.

3.1 Real time trajectory prediction model of potential obstacles based on Kalman filter

Taking the IRB120 robot base as the origin, the spatial position Cartesian system is established. By installing a Kinect visual sensor, the position, velocity and acceleration of the unknown obstacle, which can be written as:

$$T_{ri} = \{(p_{i1}, v_{i1}, a_{i1}), (p_{i2}, v_{i2}, a_{i2}), ..., (p_{im}, v_{im}, a_{im})\} \tag{2}$$

Where, p_{ij} is the position vector of the ith unknown barrier at the jth moment, v_{ij} is the velocity vector of the ith unknown barrier at the jth moment, and a_{ij} is the acceleration vector of the ith unknown barrier at the jth moment.

According to the kinematics formula, a matrix form of obstacle state equation can be obtained as follows:

$$\begin{bmatrix} p_{ik} \\ v_{ik} \end{bmatrix} = \begin{bmatrix} 1 & \Delta t & \frac{1}{2}\Delta t^2 \\ 0 & 1 & \Delta t \end{bmatrix} \begin{bmatrix} p_{i(k-1)} \\ v_{i(k-1)} \\ a_{i(k-1)} \end{bmatrix} + w_{k-1} \tag{3}$$

Obstacle observation equation is:

$$Z_k = H_{k-1} \begin{bmatrix} p_{i(k-1)} \\ v_{i(k-1)} \\ a_{i(k-1)} \end{bmatrix} + s_{k-1} \tag{4}$$

Known obstacle's state and observation equation, the Kalman filter state equation can be expressed as:

$$X_k = A_{k-1} X_{k-1} + B_{k-1} U_{k-1} + W_{k-1} \tag{5}$$

And the observation equation can be expressed as:

$$Z_k = H_{k-1} X_{k-1} + S_{k-1} \tag{6}$$

In the state equation, X_K represents a vector about the state at the current moment, A_{k-1} represents the state transition matrix of the system, X_{k-1} represents a vector about the state vector at the previous time, B_{k-1} is the control matrix of the system, U_{K-1} represents the vector of control at the previous moment, W_{K-1} represents the process noise of the system, Z_K represents the observed measurement at the current moment, H_{K-1} is the observation matrix of the system,

and S_{K-1} is the measurement noise of the system.

Kalman filter calculation is divided into two steps: time update and state update. Recursive algorithm is used to achieve the optimal state estimation of the system. The equation of time update is:

$$\begin{cases} \hat{X}_{\bar{k}} = A_{k-1}\hat{X}_{k-1} + w_{k-1} \\ P_{\bar{k}} = A_{k-1}P_{k-1}A_{k-1}^{T} + Q \end{cases} \tag{7}$$

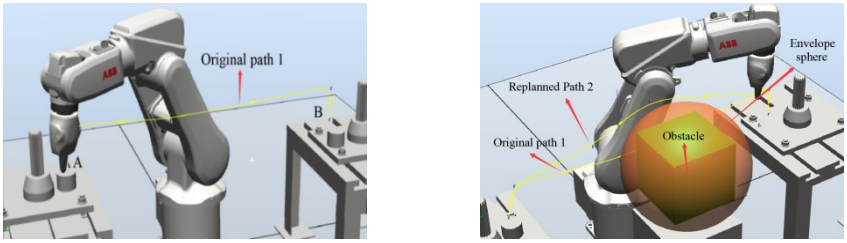
The state update equation is:

$$\begin{cases} K_k = P_{\bar{k}}H_{k-1}^{T}(H_{k-1}P\bar{k}H_{k-1}^{T} + R)^{-1} \\ \hat{X}_k = \hat{X}_{\bar{k}} + K_k(Z_k - H_{k-1}\hat{X}_{\bar{k}}) \\ P_k = (I - K_kH_{k-1})P_{\bar{k}} \end{cases} \tag{8}$$

Where, $\hat{X}_{\bar{k}}$ is the prior estimation of the current state, $P_{\bar{k}}$ represents the matrix of the prior error at the current time, and P_K represents the matrix of the posterior error at the current time.

Substituting equations (3) and (4) into the time update equation (7) and state update equation (8) of Kalman filter will complete the optimal estimation of the current state of potential obstacles and the trajectory prediction of the next time.

Combined with the built line segment sphere envelope box collision detection model, the method of collision prediction is simulated in the animation scene, which is based on Kalman filter. As shown in Fig. 5.



(a)Original task trajectory with barrier-free (b)Re-planning trajectory when encountering obstacles

Fig. 5 Diagram of the effect of a robot avoiding obstacles and re-planning its trajectory in RobotStudio

When there are no obstacles, the robot runs along the original path 1 in Fig. 5 (a); After the obstacle is detected, the obstacle trajectory is effectively predicted,

and the collision between obstacles is detected in real time through the segment sphere envelope method. When the collision is predicted, the robot re plans the path 2 in real time, as shown in Fig. 5 (b).

3.2 Real time trajectory optimization algorithm of manipulator

3.2.1 Interpolation function of manipulator joint trajectory

Select several path points in the task path of the robot as the target, the real-time path adopts the interpolation trajectory established by quintic polynomial. The general formula can be written as:

$$\theta_i(t) = a_0 + a_1 t + a_2 t^2 + a_3 t^3 + a_4 t^4 + a_5 t^5 \tag{9}$$

In the formula, i represents the joint number, and a_0, a_1,..., a_5 is the trajectory interpolation coefficient of the joint.

In the process of solving the interpolation coefficient of the trajectory, the state values of each joint at the initial and final trajectors are solved according to the inverse kinematics of the mechanical. For considering to understand, the initial position of each joint is set as θ_o, the end position is set as θ_f, the initial velocity is set as v_o, the end velocity is set as v_f, the initial acceleration is set as α_o, and the end acceleration is set as α_f. The initial and end parameters are substituted into the established trajectory interpolation general formula (9), The parameter matrix converted to robot kinematics equation is:

$$\begin{bmatrix} a_0 \\ a_1 \\ a_2 \\ a_3 \\ a_4 \\ a_5 \end{bmatrix} = \begin{bmatrix} 1 & t_o & t_o^2 & t_o^3 & t_o^4 & t_o^5 \\ 0 & 1 & 2t_o & 3t_o^2 & 4t_o^3 & 5t_o^4 \\ 0 & 0 & 2 & 6t_o & 12t_o^2 & 20t_o^3 \\ 1 & t_f & t_f^2 & t_f^3 & t_f^4 & t_f^5 \\ 0 & 1 & 2t_f & 3t_f^2 & 4t_f^3 & 5t_f^4 \\ 0 & 0 & 2 & 6t_f & 12t_f^2 & 20t_f^3 \end{bmatrix}^{-1} \begin{bmatrix} \theta_o \\ v_o \\ \alpha_o \\ \theta_f \\ v_f \\ \alpha_f \end{bmatrix} \tag{10}$$

Thus, the track interpolation coefficients of each joint are calculated, and the track interpolation coefficients are substituted into equation (9) to get the five-degree polynomial joint space interpolation track equation of each joint.

3.2.2 Establishment of objective function for trajectory optimization

The time-consuming sum of each time step in the running process of the robot can be expressed by this equation:

$$S_1 = T_{total} = \sum_{t_o}^{t_f} t = \sum_{i=0}^{1} (t_{i+1} - t_i) \tag{11}$$

The energy consumption optimization function is expressed by this equation:

$$S_2 = E = \int_0^T |\tau \bullet V(t)| dt \tag{12}$$

In the formula, τ Contains the force and torque of the driver, and V(t) represents the angular speed of the joint.

Since the residual vibration of the joint is directly related to the jerk of the rotation of the arm joint, the residual vibration of the arm joint can be written as:

$$S_3 = J_{jerk} = \sum_{n=1}^{N} \sqrt{\frac{1}{T} \int_0^T J_i^2 dt} \tag{13}$$

According to the operating conditions, we can get the motion constraint from each joint, these parameters are shown in Table 2:

Table 2

Constraints from each joint

Constraints	Constraint Value					
	1	2	3	4	5	6
Q_{jmax}(°)	165	110	70	160	120	400
V_{jmax}(rad/s)	1.92	1.48	1.57	2.442	2.09	2.26
A_{jmax}(rad/s^2)	5.57	6.39	5.39	5.04	6.39	6.57
J_{jmax}(rad/s^3)	20	25	20	25	20	20

3.2.3 Multi-objective particle swarm optimization algorithm according to the R2 index selection strategy(R2SMMOPSO)

The location and speed update formulas of Particle Swarm Optimization (PSO) can be written as follows:

$$v_i^{t+1} = wv_i^t + c_1 r_1 (x_{pb,i}^t - x_i^t) + c_2 r_2 (x_{gb,i}^t - x_i^t) \tag{14}$$

$$x_i^{t+1} = x_i^t + v_i^{t+1} \tag{15}$$

In this equation, c_1 and c_2 represent two learning factors for the individual and global optimum values, r_1 and r_2 represent two random values with uniform distribution (0,1), and v_i^t and x_i^t represent the velocities and locations of the t-th generation particles i, respectively, w is a coefficient to express the inertia. Individual optimal location and global optimal location of particle i are $x_{pb,i}^t$ and $x_{gb,i}^t$, respectively.

R2SMMOPSO uses the R2 index as a distribution strategy, the improved

R2 indicator formula can be expressed by this equation:

$$R2(A,W,z^*)=\frac{1}{|W|}\sum_{\omega\in W}\min_{x\in A}\left\{\max_{i=1,2,...,m}\frac{1}{\omega_i}\left|f_i(x)-z^*\right|\right\} \tag{16}$$

In the equation, W is a set of uniform weight vectors, and each weight vector in M target spaces $\omega=(\omega_1,\omega_2,\ldots,\omega_m)$ all belong to W. $1/|W|$ represents the probability that each weight vector is selected.

In the process of manipulator trajectory optimization, the expression of adjusting inertia weight based on improved cosine can be written as:

$$w(t)=a\times\cos(\frac{t}{GEN}\times pi\times c)+b \tag{17}$$

The HV (HyperVolume) index value is obtained by using R2SMMOPSO algorithm with four groups of coefficients to run independently 20 times on UF series test function. After testing, it is found that when a=0.25, b=0.65 and c=2 are selected as the coefficients of inertia weight formula, HV index has the most test function of the optimal value, which indicates that this group of coefficients is better than other coefficients.

In order to prevent the emergence of local optimal solution, we choose the Gaussian mutation strategy to initialize the particles. The Gaussian mutation strategy can be written as:

$$\begin{cases} x_i^{t+1}(j)=N(\dfrac{x_{gb,i}(j)-x_{pb,i}(j)}{2},\left|x_{gb,i}(j)-x_{pb,i}(j)\right|) \\ v_i^{t+1}(j)=-v_i^t(j) \end{cases} \tag{18}$$

In the equation, n (m, σ) Indicates the use of the mean m and σ Normal distribution random number.

The implementation process of trajectory planning of R2SMMOPSO is discribed in Fig. 6:

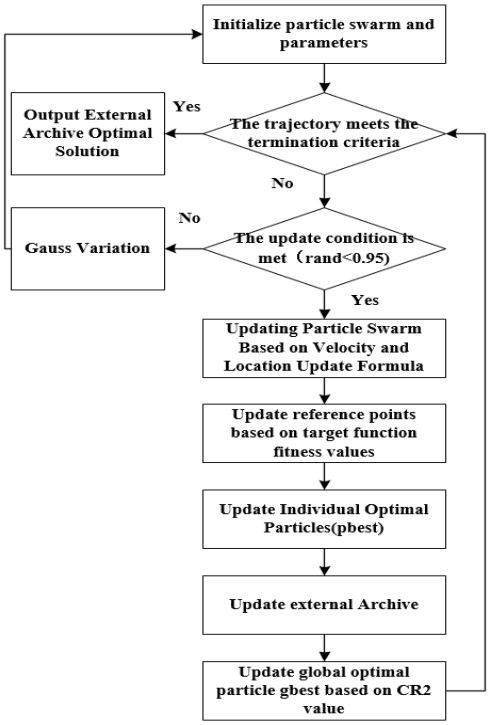


Fig. 6 Process of trajectory optimization of R2SMMOPSO

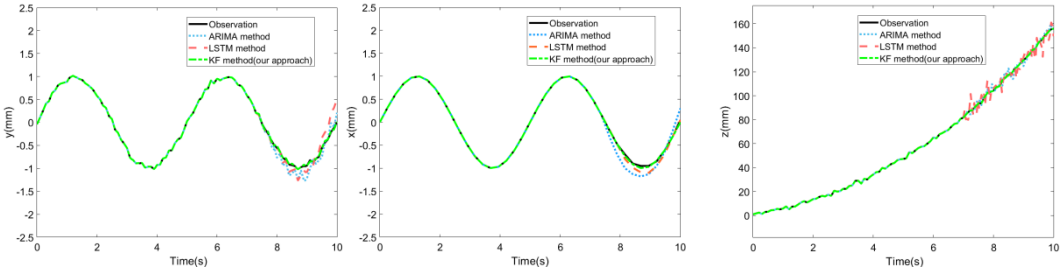
So far, the real-time obstacle avoidance and trajectory planning model of the manipulator has been built.

4 Simulation and result analysis

In order to judge whether the proposed strategy is effective, simulation experiments are carried out, which are potential obstacle trajectory prediction simulations and overall real-time adaptive motion planning strategy simulations.

4.1 Simulation analysis of potential obstacle trajectory prediction

The manipulator joint model is established in MATLAB based on the D-H parameters of IRB120 robot, compared with the two classical prediction algorithms of autorepressant integrated moving average (ARIMA) and the long short term memory (LSTM), the change trend of the predicted value of potential obstacles under different prediction algorithms and the actual observed value of detection camera in x, y and z directions in 10s is shown in Fig. 7.



(a)Predicted trajectory (X axis) (b)Predicted trajectory (Y axis) (c)Predicted trajectory (Z axis)

Fig. 7 Prediction results of spatial motion trajectories in different directions in 10s

It can be seen that the predicted trajectory trend is similar to the actual observation value, but the predicted trajectory error is large. The trajectory predicted by KF prediction method in 7-10s basically coincides with the actual observation value of the camera. The trajectory prediction accuracy of obstacles is obviously better than ARIMA and LSTM methods.

4.2 Simulation analysis of overall real-time adaptive motion planning strategy

After the completion of the real-time motion planning model, it is integral to evaluate the operation effect of the proposed real-time adaptive motion planning control strategy of the overall manipulator, the control system is simulated. Fig. 8. has shown the change trend of position, velocity, acceleration and acceleration of each joint within 10s:

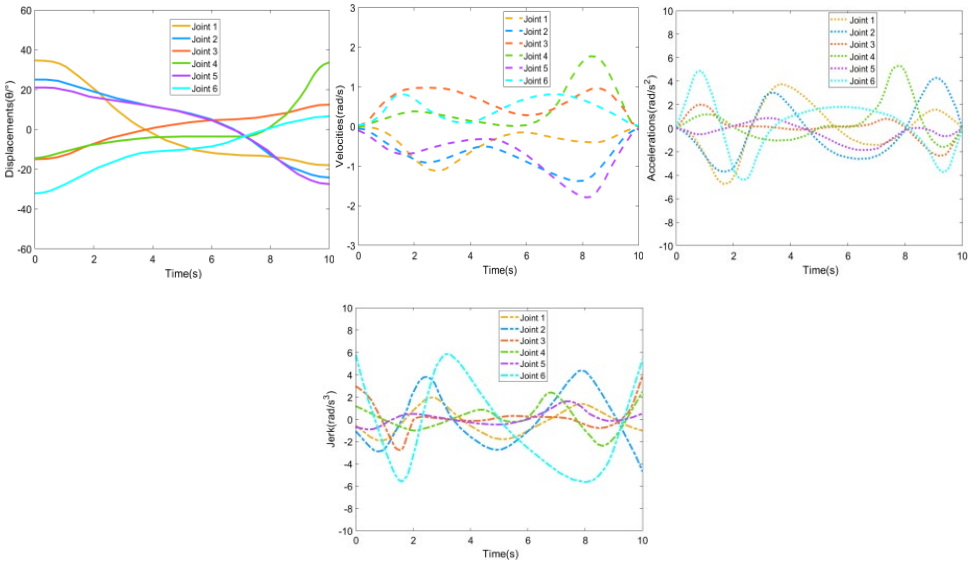
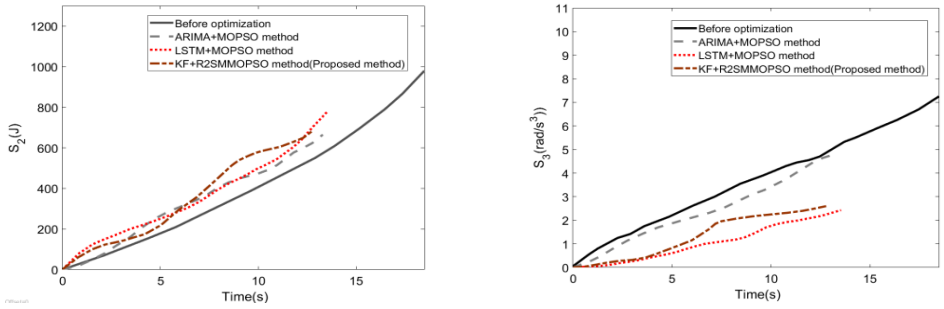


Fig. 8 Variation trend of some kinematic parameters of each joint within 10s

It can be seen that the initial angular velocity and acceleration of the joint are zero, which meets the requirements at the beginning and end of the task. The maximum jerk of any joint is not more than 5.784rad/s^3. It is not easy to produce joint vibration caused by excessive joint impact.

The time, energy consumption and remaining vibration of the joints for trajectory planning of several classical algorithms are compared as discribed in Fig. 9.



(a) (b)

Fig. 9 Time, energy consumption and joint residual vibration results for trajectory planning under different motion planning strategies

It is apparent that when the Kalman filter proposed is combined with R2SMMOPSO, it has better optimization results than other optimization methods in terms of time-consuming, energy consumption and joint residual vibration. In summary, the proposed Kalman filter combined with R2SMMOPSO can find a comprehensive and optimal joint motion track in dynamic environments.

5 Experiment and results analysis

The physical experiment platform based on IRB120 robot and Kinect camera in the narrow dynamic laboratory environment is shown in Fig. 10. to evaluate the real-time motion planning strategy of the proposed manipulator. The Kinect depth camera detects potential obstacles near the robot in real time. The upper computer reprogrammes the joint trajectory according to the real-time trajectory of the current joint and the state information of potential obstacles of IRB120 robot, the real-time planned joint trajectory is sent to the industrial computer to control the axis driver to drive each joint of the robot. Similarly to the simulation, the robot end grasping task is set to move from the point (24, 112, 14) to the target point (- 32, 553, - 31). The experiment of real-time motion planning of the manipulator is shown in Fig. 11.

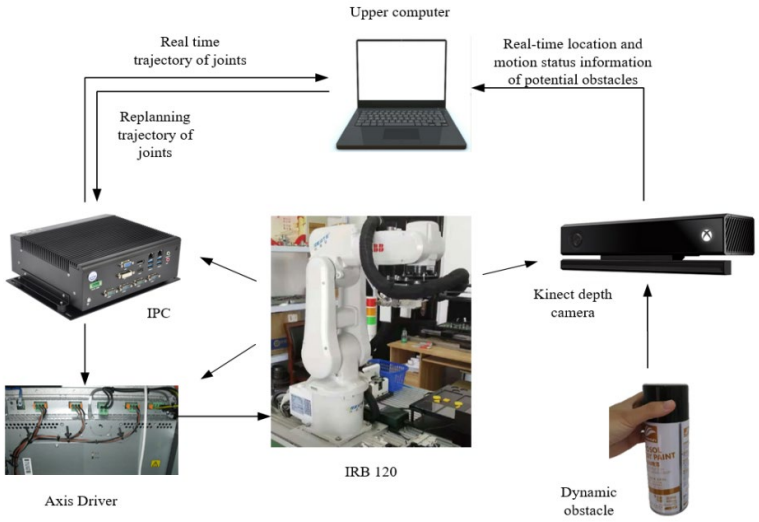


Fig. 10 Real-time motion planning physical experiment platform of IRB 120 robot

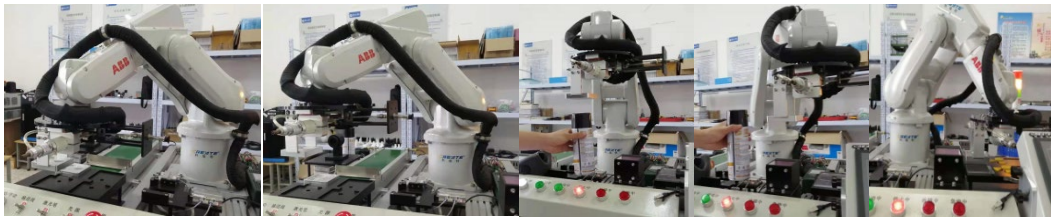


Fig.11 Real-time motion planning and movement of the manipulator during the experiment

It can be seen that the manipulator avoids obstacles in time when there are unknown obstacles, and the task trajectory is re-planned. The completion of the task of the manipulator under the control of this strategy are recorded in Table 3.

Table 3

The results of real-time motion planning experiments of the manipulator

	S_1(s)	S_2(J)	S_3(rad/s^3)
Before optimization	18.26	985.48	7.27
After optimization	12.73	687.29	2.61

From Table 3, it is apparent that the real-time optimized trajectory of the motion trajectory has made greater improvements in the time consumption S_1, energy consumption S_2 and residual joint vibration S_3 than before.After trajectory optimization, the time efficiency is improved by 30.3%, the energy consumption

is reduced by 30.1%, and the remaining vibration of the joint is reduced by 64.1%. It can be seen that compared with the three parameters before optimization, there is a greater improvement.

6 Conclusions

This paper proposed a real-time motion planning strategy for the 6R manipulator to improve the obstacle avoidance efficiency, the continuity and the adaptability of the re-planned trajectory of the manipulator in the dynamic environments. Compared with the simulation results of collision prediction, the potential obstacle trajectory prediction based on Kalman filter can ensure high prediction accuracy and stability. It is proved that the proposed real-time motion planning method can effectively avoid potential obstacles and re-plan an efficient, safe and smooth trajectory in narrow dynamic environments.

The disadvantage of this method is that only Kinect vision sensor is used to observe the position of obstacles. The shape and position recognition of obstacles are not accurate enough. We can consider using depth camera to obtain the accurate shape and spatial position of obstacles, so as to make the envelope of line segment sphere more accurate and obtain more space for robot obstacle avoidance trajectory planning. And this is our goal for further research in the future.

This method can be widely used in the environment of human-computer interaction in the narrow environment of industrial production workshop, which improves the safety and automation of work.

Acknowledgements

Thank Dr. Ni for his help in the experiment and Dr. Zhao for the data collection and analysis. This work was supported by National Natural Science Foundation of China (Grant No. 51405419), Natural Science Foundation of the Jiangsu Higher Education Institutions of China (Grant No. 18KJB460029), Research and practice innovation plan for Postgraduates in Jiangsu Province (Grant No. SJCX21_XY018) and Postgraduate Research&Practice Innovation Program of Yancheng Institute of Technology (Grant No. SJCX22_XY052).

REFERENCES

[1]. *Qian Y, Yuan J, Wan W.* Improved Trajectory Planning Method for Space Robot-System with Collision Prediction. Journal of Intelligent & Robotic Systems, 2020, 99(2):289-302.

[2]. *Liu Wei, Cheng Jin, Wan Ping, Jing Cheng, Ma Yongheng and Chen Keshiting* (2021) Dynamic Characteristics and Anti-slip Grasping of Two-Finger Translational Manipulator. Frontiers in Neurorobotics. 15:684317.

[3]. *Wang Y , Sheng Y , Wang J , et al.* Optimal Collision-Free Robot Trajectory Generation Based on Time Series Prediction of Human Motion. IEEE Robotics and Automation Letters, 2017:226-233.

[4]. *Xia C, Weng C Y, Zhang Y, et al.* Vision-Based Measurement and Prediction of Object Trajectory for Robotic Manipulation in Dynamic and Uncertain Scenarios. IEEE Transactions on Instrumentation and Measurement, 2020, 69(11):8939-8952.

[5]. *Wang, W., Zhu, M., Wang, X., He, S., He, J., & Xu, Z.* (2018). An improved artificial potential field method of trajectory planning and obstacle avoidance for redundant manipulators. International Journal of Advanced Robotic Systems, 15(5), 1

[6]. *Das, N., & Yip, M. C.* (2020). Forward Kinematics Kernel for Improved Proxy Collision Checking.
IEEE Robotics and Automation Letters, 5(2), 2349–2356.

[7]. *Chen, D., Li, S., Wang, J., Feng, Y., & Liu, Y.* (2019). A multi-objective trajectory planning method
based on the improved immune clonal selection algorithm. Robotics and Computer-Integrated Manufacturing, 59, 431–442.

[8]. *Barnett, E., & Gosselin, C.* (2020). A Bisection Algorithm for Time-Optimal Trajectory Planning
Along Fully Specified Paths. IEEE Transactions on Robotics, 37(1), 131-145.

[9]. *Wen, S., Hu, X., Lv, X., Wang, Z., & Peng, Y.* (2019). Q-learning trajectory planning based on Takagi–Sugeno fuzzy parallel distributed compensation structure of humanoid manipulator. International Journal of Advanced Robotic Systems, 16(1).

[10]. *Oliveira, P. W., Barreto, G. A., & Thé, G. A. P.* (2020). A General Framework for Optimal Tuning of PID-like Controllers for Minimum Jerk Robotic Trajectories. Journal of Intelligent & Robotic Systems, 99(3-4), 467-486.

[11]. *Cailhol, S., Fillatreau, P., Zhao, Y., & Fourquet, J.-Y.* (2019). Multi-layer path planning control for the simulation of manipulation tasks: Involving semantics and topology. Robotics and Computer-Integrated Manufacturing, 57, 17–28.

[12]. *Angelov D , Hristov Y , Burke M , et al.* Composing Diverse Policies for Temporally Extended Tasks. IEEE Robotics and Automation Letters, 2020, 5(2), 2658-2665.

[13]. *Jin, R., Rocco, P., & Geng, Y.* (2020). Cartesian trajectory planning of space robots using a multi-objective optimization. Aerospace Science and Technology, Volume108, 106360.

[14]. *Mukadam, M., Dong, J., Dellaert, F., & Boots, B.* (2018). STEAP: simultaneous trajectory estimation and planning. Autonomous Robots. 43(2), 415-434.