Obstacle Avoidance Trajectory Planning for Gausian Motion of Robot Based on Probability Theory

Haozheng Li, Yiyang Bian

***Abstract*—When the robot's movement has process noise, or its closed-loop feedback sensors have specific observation errors, the robot will present significant uncertain movement. The non-deterministic movement state is described by the Gaussian distribution which is widespread in nature. The probability theory combing with the robot's linear control and Kalman filter estimation is used to plan the trajectory and evaluate the A priori probability distribution. Liner control method is used in combination with Kalman filter to establish error model of Gaussian motion system. Then, all feasible trajectories are assessed by the Gaussian motion model by calculating the probability of avoiding obstacles and arriving at the target. For the optimal trajectory planning, spline method is used to calculate a set of feasible paths. Theoretically, all those trajectories can get the aim point and avoid the obstacles. But for the uncertainty of the robot's behavior, the robot still has the probability of collision and miss the target. Through Gaussian movement prior probability estimates, the trajectory with the maximum probability value is the optimal on under the non-deterministic Gaussian motion state of the robot.**

Keywords—Gaussian motion, robot, probability, trajectory planning

I. Introduction

In this experiment, on the basis of the linearization of the robot model, the observation sensor Kalman filter is used to estimate the error probability of the robot moving along the predetermined trajectory in advance, and the expected value of the point seat probability and the possible error on the predefined probability are expressed by covariance. Generate a probability ellipse for the probability distribution after each motion. And calculate the collision probability with obstacle after each movement, so as to get the collision probability of the whole track. On this basis, the spline method is used to generate multiple trajectories, and the collision probability of each trajectory is calculated. Finally, the trajectory with the lowest collision probability is selected as the optimal trajectory.

In this experiment, the robot trajectory, motion control cycle, trajectory analysis based on Bayesian theory and iteration of sensor error distribution are combined to quantitatively calculate the actual motion success probability of the robot. The experimental results show that the trajectory planning method based on probability theory can effectively estimate the error distribution, variation trend and success probability of robots reaching the target point in each movement period.

II. Relate Work

At present, most research on trajectory planning methods of robotic systems are based on the deterministic assumptions of the system, and there are many excellent results[1-3].In the research of robot non-determinism, Bry[4] added system non-determinism to the node generation process of the random expansion tree. Each time a new node is generated, the Monte Carlo method is used to test the variance probability of the new node and carry out trajectory planning. Toit[5] et al. combined LQR control with Kalman filter in the robot control system, and used the rolling time domain method to reduce the movement deviation of the robot system. Sun[6] et al. applied multi-core and multi-threaded programming technology to apply Toit's LQG non-deterministic sampling optimization method to real-time calculations, and simulated the real-time trajectory planning of medical probes and mobile robots.

III. Priori Estimation of Trajectory Error Distribution

A. Kalman filter estimation

Due to the errors in the process of robot movement, the position deviation between the actual output of the system and the system instruction is subject to Gaussian distribution in each control cycle. And because the observation system also has observation noise subject to Gaussian distribution, the deviation state of robot trajectory cannot be accurately reflected. Therefore, moving Kalman filter can effectively estimate the position of the robot. The principle of the Kalman filter is not introduced here. In this experiment, Kalman filter is used to estimate the error in the process of gaussian trajectory planning.

Prediction

$$X_k = A_k X_k + B_k u_{k-1} \tag{1}$$

$$P_k = A_k P_{k-1} A_k^T + V_k M_k V_k^T \tag{2}$$

Adjusting

$$K_k = P_k H_k^T (H_k P_k H_k^T + W_k N_k W_k^T)^{-1} \tag{3}$$

$$X_k = X_k + K_k (Z_k - H_k X_k) \tag{4}$$

$$Pk = (I - K_k H_k) P_k \tag{5}$$

X_k is the optimal estimation of the system state within the k cycle, and P_k is the covariance estimation of the system state. K_k is Kalman gain matrix

IV. Collision probability calculation

A. Gaussian motion probability ellipse representation

Through the iteration of the trajectory modeling and estimation process in Section 4, the normal distribution variance of the robot trajectory points in each control cycle is obtained. The probability ellipse can be used to represent the distribution of variance. Assume that the covariance matrix at the trajectory point p_i is T_i. As shown in Figure 2, random points that obey the normal distribution with an expected variance of p_i are distributed, and each random point represents a possible position of a robot.

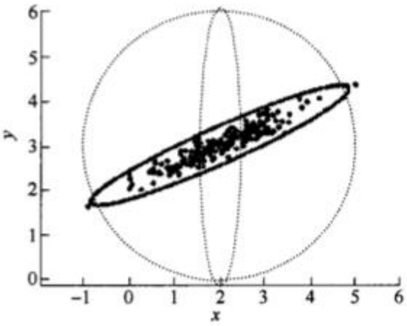


Fig. 1. Probability Ellipse Generation Process

The detailed motion probability ellipse generation method is as follows:

The equation of an axis-aligned ellipse with a major axis of length $2a$ and a minor axis of length $2b$, centered at the origin, is defined by the following equation:

$$(\frac{x}{a})^2 + (\frac{y}{b})^2 = 1 \tag{6}$$

In the motion probability ellipse, the length of the axes are defined by the standard deviations σ_x and σ_y of the data. Then choose s and let the scale of the resulting ellipse represents a chosen confidence level. Such that the equation of the error ellipse becomes:

$$(\frac{x}{\sigma_x})^2 + (\frac{y}{\sigma_y})^2 = s \tag{7}$$

σ_x and σ_y represent the standard deviations of x and y, respectively, and s is related to confidence level corresponds. s can be easily obtained using the cumulative Chi-Square distribution. And then we can get an Axis-aligned confidence ellipses.

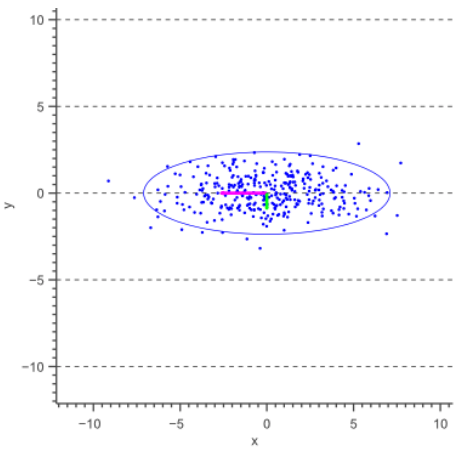


Fig. 2. Axis-aligned confidence ellipses

In cases where the data is not uncorrelated, such that a covariance exists, the resulting error ellipse will not be axis aligned. In this case, the reasoning of the above paragraph only holds if we temporarily define a new coordinate system such that the ellipse becomes axis-aligned, and then rotate the resulting ellipse afterwards.

First calculate the covariance matrix of random variables x and y, calculate the eigenvector of the matrix according to the covariance matrix (the green is the eigenvector in the direction of the short axis of the ellipse; the pink is the eigenvector in the direction of the long axis of the ellipse), we define the ellipse The eigenvector in the major axis direction is v_1, and the eigenvector in the minor axis direction of the ellipse is v_2. And define the angle between the major axis of the ellipse and the positive direction of the x-axis as α.

$$\alpha = arctan\frac{v_1(y)}{v_2(x)} \tag{8}$$

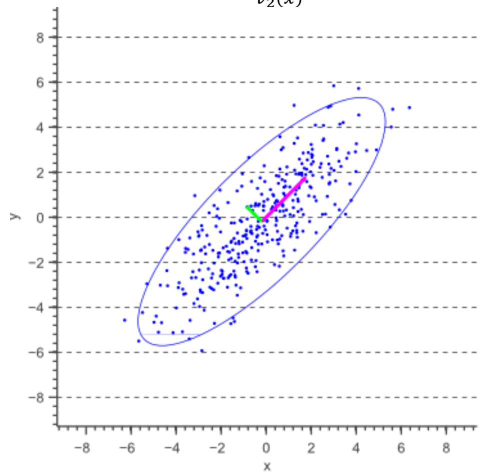


Fig. 3. Arbitrary confidence ellipses

B. Obstacle avoidance probability calculation

When the robot performs Gaussian motion near an obstacle, as long as the probability ellipse intersects the obstacle, the robot may collide with the obstacle. From the principles of mathematical statistics, this probability will be reflected in repeated trials of the same trajectory. For the same planned trajectory, the same obstacle, and the same control period, collisions sometimes occur after multiple tests, sometimes not. When the test sample tends to infinity, the ratio of the number of collisions to the total number of tests will be infinitely close to a certain probability value.

There are two ways to calculate the collision probability:

1. Calculate the area ratio of the motion probability ellipse and the obstacle collision interference. For this, the Monte Carlo method can be used to calculate the area of intersection.

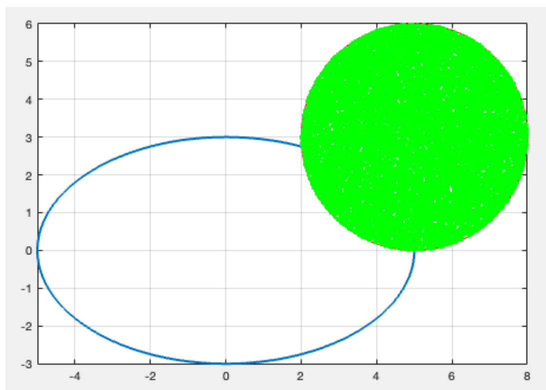


Fig. 4. Example of Monte Carlo method

As shown in Figure 2, put N sampling points into the obstacle area. Count the number of sampling points in the probability ellipse area n. The area of the intersecting part should be:

$$area = \frac{n}{N} * (Area\ of\ ellipse) \quad (9)$$

2. The obstacle is expressed by the local linearization method, and the vector from the obstacle point closest to the robot's optimal estimated position $\hat{x}_{k|k-1}$ to the robot's optimal estimated position $\hat{x}_{k|k-1}$ is represented by $\hat{a}_i$. The linearized expression at the obstacle point closest to the optimal estimated position of the robot can be expressed by equation 10.

$$\tilde{a}_l^T \tilde{x}_{k|k-1} = \tilde{b}_i \quad (10)$$

As shown in Figure 5, in order to project the two-dimensional probability ellipse into a one-dimensional space for normal distribution probability calculation, a plane perpendicular to the vector $\tilde{a}_i$ is used to divide the probability ellipse with a distribution of $x_{k|k-1} \sim N(\hat{x}_{k|k-1}, P_{k|k-1})$ and project to the one-dimensional space Obtain the normal distribution of $\zeta \sim N(\tilde{a}_l^T \hat{x}_{k|k-1}, \tilde{a}_l^T P_{k|k-1}, P_{k|k-1}\hat{a}_i)$. Thus, in one-dimensional space, the collision probability of normal distribution is calculated.

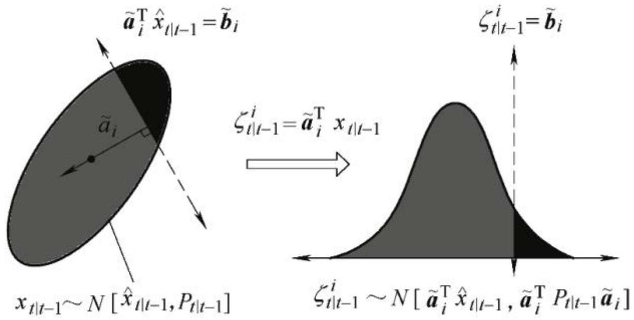


Fig. 5. Probability ellipse collision probability calculation

Since the motion of the robot is a continuous motion in multiple control cycles, each control cycle has the probability of collision with obstacles. The success probability of the final robot trajectory is the factorial of the collision-free probability of the robot in each cycle, as shown in equation (11). When the robot collides without obstacles in a control period, its posterior probability can be further calculated, that is, in the same sample space, the optimal estimated sample position and probability of no collision in this period.

$$p_{suc} = p(\Lambda_{k=0}^{l} x_k \in X_s) = \Pi_{k=0}^{l} p(x_{k|k-1} \in X_s) \quad (11)$$

In the formula, p_{suc} is the probability of the final robot successfully reaching the destination through Gaussian motion. X_s represents the safe state space where the robot avoids obstacles.

C. Trajectory optimization method

In order to plan the optimal trajectory of the robot's Gaussian motion according to the robot's motion error characteristics and sensor error characteristics before the robot moves, some autonomous optimization strategies can be used, such as genetic algorithm, random extended tree (RRT) and other methods to get a lot of A predefined trajectory to reach the target point. Then use the method based on probability theory described above to evaluate the probability of success of each trajectory.

Among all those method, one thing should be noted is: according to actual simulation research, it is not suitable to use optimization methods such as RRT* for pre-defined trajectory planning while applying the method in this paper. Because these methods have their own optimization strategies, the resulting set of predefined trajectories is the result of screening through this optimization strategy. On the one hand, these trajectories cannot uniformly cover the robot state space due to optimization screening, and on the other hand, the optimization strategy cannot guarantee consistency with the safety of the robot's Gaussian motion.

As shown in Figure 3, if an optimization method such as RRT* is used to plan a set of predefined trajectories to reach the destination point, due to the optimization strategy of the RRT method itself, in order to obtain the shortest trajectory path, the trajectory will bypass the obstacle in a closer way obstacle. However, in the Gaussian motion of the robot, these trajectories will be the probability ellipse of the method intersecting with the obstacle, which means that the robot moving along these paths will be in danger of colliding with the obstacle. The ideal trajectory may be a trajectory far away from the obstacle, but the RRT* method cannot be obtained due to the setting of its own optimization strategy.

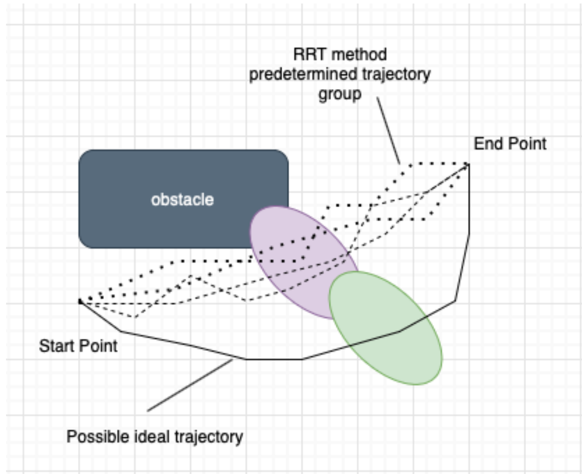
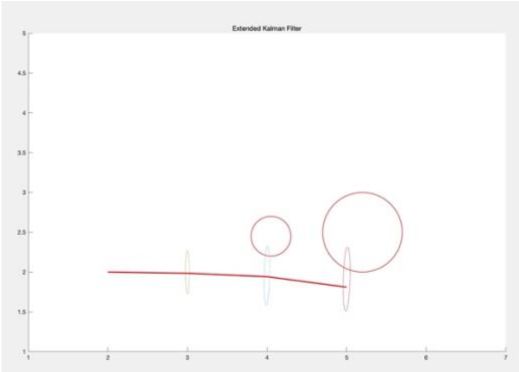


Fig. 6. Applicability of trajectory pre-planning method

V. SIMULATION TEST

To verify the effectiveness of the algorithm, the starting position of the robot was set, and the movement instructions were specified. In this experiment, Kalman filter is used to carry out continental iteration on the trajectory. While simulating the trajectory motion, the geometric information of the probability ellipse for each cycle and the collision with environmental obstacles are calculated. Calculate the probability of success for all trajectories.

As for the experimental setup, two obstacle areas are set up within the 6*6 area. The starting position of the robot is (2,2) and the ending position is (5,1.8).



No.	P(collision)
1	0
2	0.0432
3	0.0815
Total	0.00352

VI. CONCLUSION

In this paper, the probability theory, linear system control simulation and extended Kalman filter method are used to estimate the successful trajectory of the robot. The simulation test verifies the effectiveness of the algorithm. The following points need to be elaborated:

- Motion errors need to be calculated through experiments before the imputation planning, and its accuracy directly affects the accuracy of the probability estimation of the imputation results in this paper.
- It can be seen from the policy test that the greater the trajectory probability ellipse, the greater the robot motion error. When the probability ellipse intersects the obstacle, the robot has the probability of colliding with the obstacle.

REFERENCES

[1] Yershova A, LaValle S M. Improving motion-planning algorithms by efficient nearest-neighbor searching[J]. IEEE Transactions on Robotics, 2007, 23(1): 151-157.

[2] Qi R, Zhou W, Wang T. An obstacle avoidance trajectory planning scheme for space manipulators based on genetic algorithm[J]. Jiqiren/Robot, 2014, 36(3): 263-270.

[3] Otte M, Correll N. C-FOREST: Parallel shortest path planning with superlinear speedup[J]. IEEE Transactions on Robotics, 2013, 29(3): 798-806.

[4] Bry A, Roy N. Rapidly-exploring random belief trees for motion planning under uncertainty[C]//2011 IEEE international conference on robotics and automation. IEEE, 2011: 723-730.

[5] Du Toit N E, Burdick J W. Robot motion planning in dynamic, uncertain environments[J]. IEEE Transactions on Robotics, 2011, 28(1): 101-115.

[6] Sun W, van den Berg J, Alterovitz R. Stochastic extended LQR for optimization-based motion planning under uncertainty[J]. IEEE Transactions on Automation Science and Engineering, 2016, 13(2): 437-447.