Proceedings of the 3rd International Conference on Signal Processing and Machine Learning
DOI: 10.54254/2755-2721/4/2023325

Identifying and updating local optimization methods in extended Kalman filter SLAM

Hanmo Liu

Hong Kong University of Science and Technology Computer Engineering, HongKong, China, 999077

hliucg@connect.ust.hk

Abstract. The essence of simultaneous location and mapping (SLAM) is an estimation problem. It requires mobile robots to use the sensor information to estimate the structure of the external environment in real-time and their position in the background. Smith and Cheeseman first applied the EKF estimation method to slam in the last century. The SLAM algorithm based on an extended Kalman filter (EKF-SLAM for short) is one of the research hotspots of slam technology. The EKF is improved based on the Kalman filter to make the Kalman filter suitable for nonlinear dynamic systems to solve the estimation problem of the nonlinear system model. Therefore, the Kalman filter can be applied to slam technology. The slam model based on filtering mainly includes three equations: Firstly, is to use the equation of motion, which will increase the uncertainty of robot positioning. Then the equation for initializing the guidepost according to the observation initializes the new state quantity according to the observation information. Lastly is the projection equation of the landmark state to comment, update and correct the condition according to the observation information to reduce the uncertainty. In the implementation, there exists a problem regarding storing the obstacle location. The current standard practice is to abandon all the existing data when the location number reaches the threshold. This will cause calculation time waste and instant high error [1]. In this paper, several improvement methods try to fully utilize the received message by different algorithms built on the ROS system. These include abandoning local data logic, updating data instead of deleting data, and retaining user data when updating new data. These advanced methods designed for EKF-SLAM will provide a more precise outcome with a low error range.

Keywords: Extended Kalman Filter; SLAM; SLAM improvement Algorithm.

1. Introduction

The EKF-SLAM is a critical primary method in the slam method. The specific implementation is to predict and estimate the vehicle position obtained from the odometer and the relative position observed in the laser radar during the robot’s operation. This process is divided into five steps: state pricing, measurement prediction; measurement; data association, and update. In this project, we use 16 4*4 square cylinder matrices as the experimental environment, and the distance between cylinders is 0.5m. During the car’s operation, the laser radar carries out 360-degree scanning input, identifies the landmarks of the cylinder, and estimates its state using the input cylinder coordinates. By mentioned

© 2023 The Authors. This is an open access article distributed under the terms of the Creative Commons Attribution License 4.0 (https://creativecommons.org/licenses/by/4.0/).

methods above, we want to solve the problem that the number of obstacles will exacerbate the instance error when the obstacle array is updated. Further, the data abandoned in the current method could be used to optimize and correct the obstacle location [2].

2. Analysis and Algorithm Implement

2.1. Experiment and current problem analysis

First, on the sensor side, the STM32 board connects to the raspberry pie through the communication interface and receives processing commands for the raspberry pie. The raspberry pie is mainly associated with a lidar sensor via a USB interface and an SPI interface display to control the raspberry pie. The raspberry pie is equipped with a 16G memory card to load the system.

On the chassis side, STM32 board electronics connect two large torque motors with an encoder to control the movement through the motor interface. The encoder motor can prevent the robot's moving angle or linear displacement. Stm32 board receives raspberry pie commands to operate the chassis through an electrical drive chip.

The robot starts from point (0,0) to the first point on the rectangular moving route and moves along the planned path, moving target points and constantly predicting its state. In the process of moving, the characteristic landmarks of the surrounding environment are identified, the observation values of these landmarks are used to correct the state prediction of the mobile robot, and a more accurate estimate is obtained to achieve the autonomous positioning of the mobile robot.

At the beginning of the observation period, the vehicle uses the laser scan to estimate the location of cylinders. The ROS-implemented simulation lidar scans the 360-degree environment by 1-degree increments and returns the relative angle to the direction the vehicle is pointed at. The return data outputs the cylinder center location using the binomial fitting algorithm. Every time the lidar returned the data and calculated the position of the center of the cylinder, it stored the vehicle's pose in x-y dimensions and all cylinder's center locations in the matrix μ. Further calculating the covariance matrix of matrix u.

$$\begin{pmatrix} x \\ y \\ \theta \\ m_{1,x} \\ m_{1,y} \\ \cdots \\ m_{n,x} \\ m_{n,y} \end{pmatrix} \begin{pmatrix} \sigma_{xx} & \sigma_{xy} & \sigma_{x\theta} & \sigma_{xm_{1,x}} & \sigma_{xm_{1,y}} & \cdots & \sigma_{xm_{n,x}} & \sigma_{xm_{n,y}} \\ \sigma_{yx} & \sigma_{yy} & \sigma_{y\theta} & \sigma_{ym_{1,x}} & \sigma_{ym_{1,y}} & \cdots & \sigma_{ym_{n,x}} & \sigma_{ym_{n,y}} \\ \sigma_{\theta x} & \sigma_{\theta y} & \sigma_{\theta\theta} & \sigma_{\theta m_{1,x}} & \sigma_{\theta m_{1,y}} & \cdots & \sigma_{\theta m_{n,x}} & \sigma_{\theta m_{n,y}} \\ \sigma_{m_{1,x}x} & \sigma_{m_{1,x}y} & \sigma_{m_{1,x}\theta} & \sigma_{m_{1,x}m_{1,x}} & \sigma_{m_{1,x}m_{1,y}} & \cdots & \sigma_{m_{1,x}m_{n,x}} & \sigma_{m_{1,x}m_{n,y}} \\ \sigma_{m_{1,y}x} & \sigma_{m_{1,y}y} & \sigma_{m_{1,y}\theta} & \sigma_{m_{1,y}m_{1,x}} & \sigma_{m_{1,y}m_{1,y}} & \cdots & \sigma_{m_{1,y}m_{n,x}} & \sigma_{m_{1,y}m_{n,y}} \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ \sigma_{m_{n,x}x} & \sigma_{m_{n,x}y} & \sigma_{m_{n,x}\theta} & \sigma_{m_{n,x}m_{1,x}} & \sigma_{m_{n,x}m_{1,y}} & \cdots & \sigma_{m_{n,x}m_{n,x}} & \sigma_{m_{n,x}m_{n,y}} \\ \sigma_{m_{n,y}x} & \sigma_{m_{n,y}y} & \sigma_{m_{n,y}\theta} & \sigma_{m_{n,y}m_{1,x}} & \sigma_{m_{n,y}m_{1,y}} & \cdots & \sigma_{m_{n,y}m_{n,x}} & \sigma_{m_{n,y}m_{n,y}} \end{pmatrix} \tag{1}$$

As shown above, assume the dimension of the location matrix is n, and the covariance matrix is an n*n matrix. As we do not store the former location data but only the current ones, and in matrix calculation, it needs to ergodic all the data in n rows and n columns, so the complexity of covariance is $O(n^2)$ [3]. Every time the location matrix is updated or a new location is added, the covariance matrix needs to be calculated again. This process is the preliminary calculation of power wasted and time used.

The current method of implementing the data is to discard all the data when the stored location achieves the threshold [4]. However, this method may cause several problems. First, it does not select the obstacles to be stored or discarded when the information is discarded. The former stored data may have higher accuracy and value than the newer ones. The measurement of a unique location may have lower accuracy than the previous ones, thus causing a higher error. Second, discarding all the data may cause instant inaccuracy caused by insufficient landmarks. Landmarks may be occlusion by other landmarks, thus causing the high error.

During a total of 154s of the simulation run, EKF-SLAM can recognize many obstacles well and update the attitude matrix in real-time with an average error of around 0.03m, which is highly

DOI: 10.54254/2755-2721/4/2023325

approaching the measurement error in the previous observation period. Higher short-term errors occur when the path changes abruptly, but the model can correct the errors after a period.

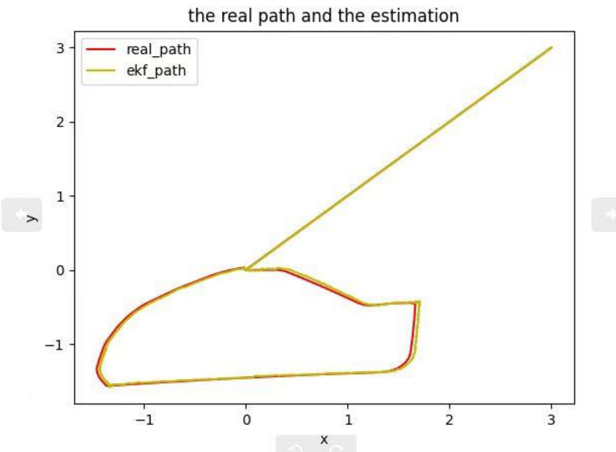

Figure1. The absolute path and the estimation.

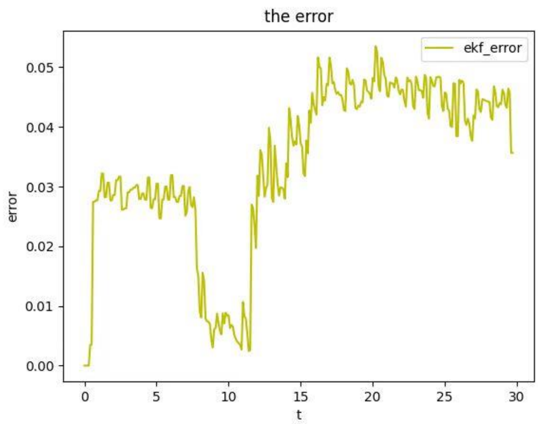

Figure 2. The error range.

2.2. Algorithm designed

Based on the current algorithm, our group proposes two improvement methods. The first possible improvement method is to use the Least Recently Used (LRU) algorithm. By creating a stack to track the use of recent times. By indexing all the existing obstacles from 1 to n and storing all the numbers in one stake. Every time detecting a specific landmark, we move the number from its current location to the highest place. Then the upper part of the stack remains all the indexes of recently used locations of landmarks. When the stored landmarks achieve the threshold, discard the latter half indexes pointed landmarks. This method can avoid dumping the more valuable-recently used landmarks but only delete the less valued ones- old ones. The selection of stack threshold depends on the performance of different single-chip computers. In general, it is expected that as many landmarks as possible will be recorded and compared. There is no substantial change in the calculation speed before the threshold value is reached, and the performance of a single-chip computer will decrease significantly when the threshold value is approached. It is recommended to experiment several times, depending on the performance of different single-chip computers.

The second possible improvement is updating the landmark position when detecting the existing landmark. The observation period has an error in the binomial fitting and lidar measurement period. One possible method to correct the mistake in each observation round is to average the measurement result every time we observe the effect. This will lead to another problem. The relative error will appear higher when the observation range is too short or too long. For example, if the landmark is too close, the ratio between relative measurement error compared to absolute distance is higher. Of the most faraway landmarks, the laser scan in one degree per sample may not have enough sample points; if there are only two points or one point per cylinder, it is not possible to estimate the center of the cylinder if no extra data such as radius of the cylinder is provided. The suggested algorithm includes two parts. Before the system starts its navigation process, manually test different error ranges at the extra distance between the vehicle and cylinder environment. The less error range's length has higher reliability, setting an optimal observation range. Whenever the landmark is observed, judge whether the distance is acceptable; if not, leave a mark on its observation record. When the vehicle follows a landmark that can be determined to be the same and has a better distance range, replace the old data with the more reliable one [5]. The other method is to do the weighted balance of the same landmark. The more reliable distance data has a higher weight than the other data. The only problem is that EKF-SLAM does not store previous data. Doing the balance needs long-time observation, which is hard for EKF-SLAM to keep track of.

Other algorithms are called Individual Compatibility Nearest Neighbor algorithm [6]. The map does not need to match all road markings with the observation data. A local area is determined according to the observation range of the sensor and the current position and posture of the robot. Only the feature points in the local area are associated with the observation data so that the calculation amount of the data association algorithm has been maintained in a stable range, ensuring the real-time performance of the SLAM algorithm. Firstly, the local association area is determined according to the parameters of the robot system. Locally associated regions are determined based on the robot's current estimated position and the sensor's observation range. Then dynamically adjust the threshold of the association gate after filtering the map features in the first step of the local association area and calculating the Mahala Nobis distance between the current moment's observations and all the features in the area. The system state error covariance matrix P(k) of the present time of the computer robot, the quantified value of the computer robot position error, and the emotional confidence level of the first observation value. Finally, the association result is obtained based on the nearest neighbor Association criterion. In the alternative set D(k), the only map feature with the smallest Mahala Nobis distance from the predicted location of the environmental feature is selected as the association result.

3. Conclusion

This article discusses several methods of improving the EKF-SLAM algorithm to solve the current problem of storing data too much, thus delaying the whole system and wasting RAM. The way of LRU and updated local variable is easy to achieve and may significantly impact the EKF-SLAM algorithm. To solve the problem, there are also some improvement algorithms, such as the nearest landmark reserve algorithm, but there must be one optimal solution. This article only discusses the specific scene where all the obstacles are cylinders. The method mentioned above could have higher or less used for further use due to different environments. It is meaningful in future studies to test the optimal way to satisfy different settings. In the future, we will test all the improvement algorithms and get the results.

Acknowledgment

Thanks to Professor Shi Ling for this project, which gives us many supports and advice. He pointed out the direction and taught us much essential research methods. Also, thank the HKUST ECE department, which gave us financial and lab resource support, which is vital to our UGV project.

Reference

[1] F. Qiang, "Analysis and Implementation of Improved SLAM Algorithm for Mobile Robot," Ordnance Industry Automation, pp. 86-89, 8 2018.
[2] F. Zhou, "Research on navigation method of Lunar Rover," Proceedings of the 37th Chinese Control Conference, pp. 4803-4808, 25-27 6 2018.

DOI: 10.54254/2755-2721/4/2023325

[3] L. M. Paz, "Divide and Conquer: EKF SLAM in O(n)," IEEE TRANSACTIONS ON ROBOTICS, VOL. 24, pp. 1107-1120, 10 2008.
[4] C. Yanghua, "Car Positioning Based on Lidar of Extended Kalman Filter（EKF） Algorithm," Industrial Control Computer, pp. 12-17, 2019.
[5] C. Jun and Z. Bi, "Improved EKF-SLAM algorithm based on landmark observation," Automation & Information Engineering, pp. 21-26, 2014.
[6] Z. Wandong, "SLAM algorithm research in a complicated environment," Harbin Engineering University, 2019.