Applications of Highly Accurate Localization and Navigation to Mobile Robot

Guo-Shing Huang  
National Chin-Yi University of Technology  
NCUT  
Taichung, Taiwan  
hgs@ncut.edu.tw

Jie-Cong Ciou  
National Chin-Yi University of Technology  
NCUT  
Taichung, Taiwan  
ngyma@yahoo.com.tw

Hsiung-Cheng Lin  
National Chin-Yi University of Technology  
NCUT  
Taichung, Taiwan  
hclin@ncut.edu.tw

Abstract - The optimal route planning is to seek the most appropriate path for robot to arrive the desired destination smoothly in the shortest time. In this paper, the Tangent method is firstly used to find the shortest moving route. Secondly, the Kalman Filtering algorithm is employed to amend the route errors at the \( k^{th} \) time during the moving status. Simultaneously, the route for next \( (k+1)^{th} \) time can be also estimated. Finally, the robot route is continuously adjusted using the fuzzy logic controller for the robot moving more smoothly and efficiently. Both simulation and experimental results confirm that the robot can reach the destination fast within no exceeding ±2cm localization error.

Keywords: Route Planning, Kalman Filtering, Fuzzy Logic Control, Ultrasonic Sensor, Laser Navigation System

I. INTRODUCTION

The optimal route planning for mobile robots is still a crucial research field [1-8]. The key research point is that the robot is able to correct its current position and estimate the next route to arrive at the destination accurately and quickly without collision. In recent years, a number of publications related to the navigation and route planning can be found in the literature, for instance, the Monte Carlo [1], histogram statistic [2], fuzzy control [3][4][5], the genetic algorithm[8], etc. In the past, the Kalman Filtering algorithm was mostly used in finding the posture degree of GPS or applied in such as inertia navigation, and information integration [9][10]. This paper uses the Tangent method to determine the robot’s shortest path, and the Kalman filtering algorithm is applied to estimate the location of the robot. The fuzzy logic controller is designed to amend the navigating error. Accordingly, the optimal navigation planning for the mobile robot can be achieved.

This paper is divided five sections. Section II describes the motion model of the mobile robot as well as the model of laser-based location measurement. The Tangent method combining with the Kalman Filtering algorithm and fuzzy control is presented in section III. Section IV provides both simulation and experimental results to verify the effectiveness of the proposed approach. The conclusions are given in section V.

II. DESCRIPTION OF ROBOT MOTION AND MEASUREMENT MODEL

A. Model of Robot Motion

The model of robot motion is depicted in Fig.1. The safe range between the robot and the obstacle is defined as the radius \( R_T \). The coordinates of the robot’s current location and its target destination are defined as \((x_R, y_R)\) and \((x_O, y_O)\), respectively. In this model, a number of ultrasonic sensors were installed to measure the object (robot, target and obstacle) locations within the 180-degree range area in the front of moving robot.

The robot’s heading angle is defined as \( \theta_R \). The angle \( \theta_{RG} \) and the distance \( D_{RG} \) between the robot and the target can be calculated from the equations (1) and (2), as follows.

\[
\theta_{RG} = \tan^{-1} \frac{y_O - y_R}{x_O - x_R} \quad (1)
\]

\[
D_{RG} = \sqrt{(x_O - x_R)^2 + (y_O - y_R)^2} \quad (2)
\]

The distance and included angle between the robot and obstacle are defined as \( D_{RO} \) and \( \theta_{RO} \), respectively. They can be also obtained similarly using the equations (1) and (2). The speed of robot motion (S) and its moving angle \( \theta \), are restricted as the range of S ∈ (0, 1.0) m/s and \( \theta \in (-30^\circ, 30^\circ) \). Note that \( \theta \) is positive when robot turns right, and \( \theta \) is negative when the robot turns left.

The robot motion equations are shown as follows.

\[
\Delta x_R = \Delta t \cos \theta_R \quad (3)
\]

\[
\Delta y_R = \Delta t \sin \theta_R \quad (4)
\]

\[
\Delta \theta_R = \theta_R(k + 1) - \theta_R(k) \quad (5)
\]

Note that \( \Delta \) is the robot moving distance. \( \Delta x_R \), \( \Delta y_R \), \( \Delta \theta_R \) and \( \Delta t \) is the changing amount in the abscissa, ordinate, moving angle and distance, respectively. Consider every change in \( \Delta \theta_R \), i.e., from the \( k^{th} \) time to \( (k+1)^{th} \) time so that the whole path of robot moving can be formed as a curve route.
B. Model of Location Measurement

The model of location measurement is shown in Fig. 2. The laser navigation system was used to settle the reflect reference location, including the distance and angle. Therefore, the robot coordinate \((x, y)\), and heading angle \((\theta)\) can be obtained using Triangular method. Note that \((x_n, y_n)\) is the \(n^{th}\) reflect coordinate, and \(\alpha_n\) is the angle of the vector \((v_n)\) from the robot \((x, y)\) to \((x_n, y_n)\). \(\rho_n\) is the length of \(v_n\) and \(\alpha_n\) is the angle between \(v_n\) and robot heading direction.

III. The route planning

To ensure the optimal path to be achieved for the moving robot, the Tangent method is firstly used to find the shortest route. The Kalman Filtering algorithm is then applied for the robot path estimation at next \((k+1)^{th}\) moment when the robot is moving ahead at the \(k^{th}\) moment. The fuzzy logic controller is to correct the moving route error and enables the \((k+1)^{th}\) moment achieving the minimum error. The flowchart of the proposed scheme for the route plan is as shown in Fig. 3.

A. The shortest route using the Tangent method

Consider the case with obstacles, the possible shortest route is shown in Fig. 4. The symbols \(R(= (x, y))\), \(B(= (x, y))\) and \(G(= (x, y))\) are to represent the location coordinates for the mobile robot, the obstacle and the destination, respectively. The maximum radius range of the obstacle is defined as \(r_B\). The straight line \((RG)\) between the robot and the destination can be described as follows.

\[
\frac{y - y_G}{y_G - y_R} = \frac{x - x_G}{x_G - x_R}
\]

The equation (6) can be rewritten as

\[
(y - y_G)x + (x_G - x_R)y + (x_Ry_G - x_Gy_R) = 0
\]

The distance \((d)\) from the obstacle \((x, y)\) to \(RG\) can be obtained from the following equation.

\[
d = \frac{|(y_G - y)x_G + (x_G - x_R)y_G + (x_Ry_G - x_Gy_R)|}{\sqrt{(y_G - y)^2 + (x_G - x)^2}}
\]

If \(d > r_B + r_G\), the robot can arrive at the destination along the straight line \(RG\) without collision. If \(d \leq r_B + r_G\), \(B\) is regarded as an unavoidable obstacle. Consequently, the radius \((r_B + r_G)\) can be formed as a collision circle, and its equation is obtained as follows.

\[
(x - x_G)^2 + (y - y_G)^2 = (r_B + r_G)^2
\]

Therefore, the tangent line from the \(R(x, y)\) to the collision circle can be expressed as

\[
y = m_t(x - x_G)
\]

On the other hand, the tangent line from the collision circle
to the \( G(x_G, y_G) \) can be expressed as

\[
(y - y_G) = m_x(x - x_G)
\]

(11)

The coordinates of \( c_1 \) and \( c_2 \) can be calculated from the equations (9) to (11) using the principles of geometry and trigonometry. Two possible shortest paths \((R \rightarrow c_1 \rightarrow G)\) and \((R \rightarrow c_2 \rightarrow G)\) can be thus obtained. Consequently, the shortest way can be determined, based on these paths.

B. The Route Estimation using Kalman Filtering

Kalman Filtering is employed to estimate the next route at the \((k + 1)^{th}\) time. The state variable is defined as

\[
x = \begin{bmatrix}
\Delta v_x \\
\Delta v_y \\
\theta_R \\
x_R \\
y_R
\end{bmatrix}
\]

(12)

where \( v_x \) and \( v_y \) are robot movement speeds at \( x \) axle and \( y \) axle direction, respectively. Define the measurement equation as

\[
z = \begin{bmatrix}
\Delta x_R \\
\Delta y_R \\
\theta_R \\
x_1 \\
y_1 \\
\vdots \\
x_n \\
y_n
\end{bmatrix}
\]

(13)

Therefore, the observation equation can be written as follows.

\[
z(k) = H(k)x(k) + \omega(k)
\]

(14)

The system state equation is written as

\[
x(k + 1) = \phi(k)x(k) + \omega(k)
\]

(15)

where \( k \) is the discrete time, \( H(k) \) is measurement function matrix, \( \omega(k) \) is the measurement error, and \( \omega(k) \) is the white noise process. The measurement function matrix \( H(k) \) and state transform matrix is described as follows.

\[
H(k) = \begin{bmatrix}
0 & 0 & 0 & 0 & 0 \\
0 & \Delta t & 0 & 0 & 0 \\
0 & 0 & 1 & 0 & 0 \\
0 & 0 & \rho_x \cos \theta_R & x_R & + 1 \\
0 & 0 & \rho_x \sin \theta_R & y_R & + 1 \\
\vdots & & & & \vdots \\
0 & 0 & \rho_x \cos \theta_R & x_R & + 1 \\
0 & 0 & \rho_x \sin \theta_R & y_R & + 1
\end{bmatrix}
\]

(16)

\[
\phi(k) = \frac{\Delta x_R}{\Delta t} \begin{bmatrix}
0 & 0 & 0 & 0 & 0 \\
0 & 0 & \Delta \theta_R & \theta_R & + 1 \\
0 & 0 & \Delta x_R & x_R & + 1 \\
0 & 0 & \Delta y_R & y_R & + 1
\end{bmatrix}
\]

(17)

Therefore, the robot moving location can be estimated according to the following steps.

1) Set the initial state: \( \hat{x}(0 | 0) = x_0 \) and initial variance-covariance matrix \( P(0 | 0) = P_0 \).

2) Estimate the state and variance-covariance matrix.

\[
\hat{x}(k + 1 | k) = \phi(k + 1, k)\hat{x}(k | k)
\]

(18)

\[
P(k + 1 | k) = P(k | k) + Q(k)
\]

(19)

where \( Q(k) \) is the process noise variance of \( \omega(k) \).

3) Update and measure new state and state estimation error variance.

\[
\hat{x}(k + 1 | k + 1) = \hat{x}(k + 1 | k) + K(k + 1)[z(k + 1) - \hat{z}(k + 1)]
\]

(20)

\[
P(k + 1 | k + 1) = [I - K(k + 1)H(k + 1)]P(k + 1 | k)
\]

(21)

where \( K(k + 1) \) is the Kalman Filtering gain.

\[
K(k + 1) = P(k + 1 | k)[H(k + 1)P(k + 1 | k)H(k + 1)^T + R(k + 1)]^{-1}
\]

(22)

where the \( R(k + 1) \) is the measurement noise covariance of \( v(k + 1) \). The state \( \hat{x}(k + 1 | k + 1) \) after filtering can be obtained so that the next robot route and moving angle can be estimated in advance.

C. Amendment of the moving path using Fuzzy Logic Controller

From Fig.4, the shortest path is assumed as \( R \rightarrow c_1 \rightarrow G \). However, this path is found with an abrupt change at the \( c_1 \) location. At this point in practice, the robot must stop a moment for the turning action before moving forward. This performance will normally take a long time and require a lot
of power consumption. To resolve this problem, this study proposes the fuzzy logic control (FLC) to modify the robot moving path, enabling the robot moving smoothly and efficiently. The block diagram of the robot moving modification with FLC is shown in Fig.4. The \( \theta_c \) is defined as the angle error between the robot current angle and desired angle. Two output variables, i.e., \( S_L, S_R \), from FLC are defined as the left wheel rotating speed and the right wheel rotating speed (\( S_e \)), respectively. The output of FLC is used to drive two DC motors using two PWM controllers.

**IV. PERFORMANCE RESULTS**

In order to verify the validity of the proposed algorithm, firstly we used the computer to run the simulation. Then, the practical implementation was carried out following up the same environment as the simulation.

**A. Software Simulation Results**

Initially, the Kalman Filtering combining the fuzzy logic controller to amend the route error was verified by MATLAB program. Note that the noise level was set up randomly within the measured signal of ±3%. Additionally, a sharp noise of localization error was added into the system at the 100-second and 200-second time domain. The simulation result is shown in Fig.7. Clearly, it is found that the system can correct the error in 2 seconds. In the normal situation, the average error is ±1cm if adding no noises.

Figure 8 shows the navigation simulation result with no obstacle. The robot route was pre-defined as 1→2→3, where the label (1) is the start point (3.6, 7.2), and the label (3) is the destination (5.4, 8.5). The coordinate of label (2), i.e., midway, is (4.7, 2.3). As can be seen, the robot route was very smooth, and the destination point was reached accurately via the midway.

**TABLE 1. Rule Table of FLC Fuzzy Reasoning**

<table>
<thead>
<tr>
<th>( S_L/S_R )</th>
<th>( D_G )</th>
<th>N</th>
<th>M</th>
<th>F</th>
</tr>
</thead>
<tbody>
<tr>
<td>PB</td>
<td>H</td>
<td>L</td>
<td>H</td>
<td>M</td>
</tr>
<tr>
<td>PM</td>
<td>M</td>
<td>L</td>
<td>M</td>
<td>H</td>
</tr>
<tr>
<td>PS</td>
<td>M</td>
<td>L</td>
<td>M</td>
<td>M</td>
</tr>
<tr>
<td>ZE</td>
<td>L</td>
<td>L</td>
<td>M</td>
<td>M</td>
</tr>
<tr>
<td>NS</td>
<td>L</td>
<td>M</td>
<td>L</td>
<td>M</td>
</tr>
<tr>
<td>NM</td>
<td>L</td>
<td>M</td>
<td>L</td>
<td>M</td>
</tr>
<tr>
<td>NB</td>
<td>L</td>
<td>H</td>
<td>L</td>
<td>H</td>
</tr>
</tbody>
</table>

**Fig.5 Block diagram of FLC scheme**

**Fig.6 Membership function of Input and output**

**Fig.7 Localization error response**
Figure 9 shows the navigation simulation results with adding two obstacles. The coordinates of start point, the destination, the first obstacle, and the second obstacle are set as (5.1, 1.9), (4.6, 8.8), (4.2, 6.9), and (5.6, 4.5). The outcome indicates that the robot can avoid the obstacles and arrive at the destination fast and accurately.

The practical navigation result with two obstacles is shown in Fig. 12. The near obstacle is firstly found on the way to the destination so that the first midway is determined. The route must be adjusted again when the second obstacle is found on the way, and the second midway is then determined. Obviously, the robot can avoid these obstacles and reach the destination. Actually, the same procedure has been extended to such the case as more obstacles existing. The localization error always does not exceed ±2cm.

B. Experimental Results with Real Hardware

In the hardware performance, the coordinate unit is 0.5 meter/dose, and the moving speed of robot was set as 0.5 m/s. The robot platform to carry out the experimental implementation was designed especially for the meal service robot, shown in Fig. 10. This robot’s head is equipped with SICK NAV 200 to offer the robot self-locating function. The waist is equipped with SICK LMS291 and the ring ultrasonic sensors for scanning the obstacles.

The practical navigation route with no obstacle is shown in Figure 11. The robot moving route is predefined from 1→2→3. This outcome reveals that the robot route is close to the simulation result (Fig. 8). The localization error is found within ±2cm.
V. CONCLUSIONS

The highly accurate localization and navigation to mobile robot has been developed successfully in this paper. The Tangent method is used to determine the shortest moving route. The Kalman Filtering algorithm is then applied to amend the route at the $k^{th}$ time, and the route for next $(k+1)^{th}$ time can be estimated. Additionally, the robot route is adjusted to be smooth by the fuzzy logic controller. Accordingly, the robot can reach the destination fast and smoothly within no exceeding ±2cm localization error even there may be several obstacles on the way. Both simulation and practical results confirm that the proposed scheme can be applied to the mobile robots in term of fast and accurate response to achieve the optimal route.

ACKNOWLEDGEMENTS

This research was supported by National Science Council under grant NSC 97-2221-E-167-018- MY3.

REFERENCES


