Adaptive algorithm for mobile robot movement control

V N Zhidkov
Moscow Aviation Institute (National Research University), 4, Volokolamskoe shosse, Moscow, 125993, Russia
E-mail: vladimir_zhidkov@mail.ru

Abstract. The paper reviews the goal of controlling mobile robot movement along the planned path. It proposes an adaptive robot movement control algorithm, allowing to increase the precision of holding the robot within the prescribed path under the conditions of unstable navigation field. The adaptation of the algorithm to jamming environment is in redistributing weights of the weighted sum of estimates given by the ultrasound navigation system (USNS) and wheel pulse transducers (WPT). When precision of the USNS reduces affect of the WPT on the final estimate increases while affect of the USNS decreases. Such a redistribution is carried out when the system is operating due to changing elements of a measurement noise matrix in Kalman filter [1] with use of which a mobile robot heading angle is estimated. Corresponding elements of the measurement noise matrix increase along with increase of deviation of the mobile robot trajectory from a straight line. The extent of deviation calculated as the standard error of a regression is defined by means of the statistical analyses involving the last \( n \) measurements contained in buffer store of the USNS. The obtained results demonstrate significant benefit of the developed mobile robot heading estimation algorithm in comparison with the conventional filtering algorithm basing on using the classical Kalman filter with the measurement noise matrix whose elements are fixed.

1. Introduction

One of the crucial directions for enhancing social robotic technologies is the development of autonomous robots, serving as human assistants. Autonomous robots shall plan the route and independently follow the preplanned route in accordance with their tasks to be fulfilled [2, 3]. As a rule, a preplanned route is composed of direct line segments. Therefore, the primary task of a mobile robot (MR) control algorithm is to ensure MR movement from one route point to another with the prescribed speed and minimum deviation from the direct line, connecting such points.

2. Mobile robot control algorithm

By now time a great variety of algorithms based on self-training has been developed [4–8]. However most of such algorithms require to use substantial computational resources. The algorithm proposed below is simple enough and undemanding to controller characteristics but at the same time it permits to control a mobile platform with high quality in conditions of an unstable navigation field.

For guiding the mobile robot (MR) to the assigned point (target), the angular misalignment value \( d\Psi \) is used (refer to figure 1)

\[
d\Psi = \Psi_t - \Psi.
\] (1)
The direction to the target is defined as follows:

$$\psi_t = \arcsin \frac{Y_t - Y(t)}{D}, \quad (2)$$

where

$$D = \sqrt{(X_t - X(t))^2 + (Y_t - Y(t))^2} \quad (3)$$

is the distance to the target.

The MR coordinates ($X(t)$ and $Y(t)$) are measured by the ultrasound navigation system (USNS) with sufficient precision (the average error equals to 2–3 cm) provided there are no severe interferences. When the distances to the target are relatively large (several meters), the error in determining the direction towards the target does not exceed 0.01 rad; however, it increases with the MR approaching the target.

**Figure 1.** MR guidance diagram.

MR heading angle is defined based on the data, containing MR coordinates in two points (points 1 and 2 in figure 1)

$$\Psi = \arcsin \frac{Y(t) - Y(t - T_v)}{S}, \quad (4)$$

where $S$ is linear distance between the coordinates, used for heading calculation

$$S = \sqrt{X(t) - X(t - T_v)^2 + Y(t) - Y(t - T_v)^2}. \quad (5)$$

The precision of heading calculation significantly depends on the length of $S$ segment. The issue is further aggravated by the fact that within $T_v$ time interval, the robot does not follow a straight line (refer to figure 1). In this case, the actual MR heading can substantially differ from the one, determined using equation (4). Therefore, the error in defining heading $\Psi$ constitutes the primary factor, determining the $d\Psi$ (1) calculation error.

To increase the precision of MR heading determination, it is necessary to use additional navigational data, which can be obtained, for example, from wheel pulse transducers (WPT), measuring the angular velocity of MR wheels. The measured angular velocity is used for identifying variations in MR heading. The final heading estimate shall be the weighted total of
the estimate, obtained via the USNS, and the estimate, generated using the data provided by the WPTs.

The adjustment based on the interference level is performed by redistributing the weighting coefficients within the weighted total. When the USNS precision drops, the impact of WPTs on the final weighted estimate shall increase, and the impact of USNS shall decrease. The most common algorithm, allowing to fulfill this task, is the Kalman filter [1].

3. Adaptive algorithm for mobile robot heading estimation

The diagram of the developed mobile robot heading estimation algorithm is provided in figure 2.

\[
\dot{\Psi} = \omega; \\
\dot{\omega} = w,
\]

where \(\Psi\)—is heading; \(\omega\)—angular mobile platform (MP) rotation rate relative to the vertical axis; \(w\)—flat noise with prescribed intensity and zero expected value.

\[Z\] measurement vector contains two items:

\[z_1 = \Psi + v_\Psi; \]
\[z_2 = \omega + v_\omega,
\]

where \(v\) are flat noises with prescribed intensity and zero expected value.
Heading $\Psi$ is not measured directly. USNS is used to measure MP coordinates at a certain time point $t$: $X(t)$ and $Y(t)$. The measured coordinates are stored in FIFO type buffer memory. The memory size allows keeping the measurements, obtained during a certain preceding period (approximately 1 sec). With the measurement cycle of 0.2 sec, the latest five measured coordinate values are stored. Then, two coordinate pairs are taken: $X(t)$, $Y(t)$, which are the latest measured coordinates (the 1st pair), and $X(t-T_v)$, $Y(t-T_v)$, which are the coordinates, preceding the current ones by a certain period $T_v$ (the 2nd pair).

The “measured” heading value $z_1$ is determined as follows:

$$z_1 = \arcsin \frac{Y(t) - Y(t-T_v)}{S},$$

where $S$ is linear distance between the coordinates, used for heading calculation

$$S = \sqrt{X(t) - X(t-T_v)^2 + Y(t) - Y(t-T_v)^2};$$

$z_2$ is the “measured” MP angular velocity relative to the vertical axis, which is also not measured directly, but determined as follows:

$$z_2 = \frac{2(V_1 - V_2)}{L},$$

where $V_1 = W_1/R$ is linear velocity of the left wheel rim; $V_2 = W_2/R$ is linear velocity of the right wheel rim; $W_1, W_2$ are MP wheel rotation rates (measured using wheel pulse transducers); $R$ is wheel radius; $L$ is distance between MP wheels.

The algorithm adaptivity is achieved by modifying the $R(1,1)$ factor of the measurement noise covariation matrix, included into the Kalman filter algorithm [1]. An increase in this factor results in a decreased weight of the USNS measurements in the total heading estimate. A decrease in $R(1,1)$ value leads to a reduction in the weight of measurements, obtained from the pulse transducers

$$R(1,1) = K_r \cdot E_L;$$

$K_r$ coefficient is selected empirically. $E_L$ is an estimate of the mobile platform deviation from a straight line. $E_L$ value can be determined by statistically processing the last $n$ measurements, stored in the USNS measurement buffer memory, as the standard error of the following linear regression equation:

$$E_L = \sum_{i=1}^{n} \frac{(Y_i(t) - Y_i^*(t))^2}{(n-1)},$$

where $Y_i(t)$ is measured $Y$ coordinate; $Y_i^*(t)$ is the coordinate, calculated by the regression equation.

In another alternative, which is easier for implementation, the MP path deviation from a straight line is found as the modulus of MP measured angular velocity relative to the vertical axis (13):

$$E_L = abc(z_2).$$

4. Simulation results

The modeling of the proposed adaptive algorithm for estimating MP coordinates was carried out using “MatLab—Simulink” system. The generalized simulation results are presented in figures 3 and 4.

Figure 3 shows the dependency of the MP heading angle estimation accuracy on the value of its path deviation from a straight line. The angular MP velocity relative to its degrees) vertical axis is used as a parameter, indicating the deviation value.
Figure 3. MP heading angle estimate accuracy depending on the path linearity level.

Figure 4. Heading angle estimate accuracy depending on the distance between coordinate measurement points.

The provided results evidently demonstrate that the adaptive algorithm has a significant advantage over the traditional one. The reason is that to prevent the divergence of the Kalman filter in case of a significant curvature of the MP path, the traditional algorithm requires setting relatively large values of the measurement noise intensity $R(1, 1)$. On the other hand, the use of the adaptive algorithm allows keeping the $R(1, 1)$ value relatively small for small path curvature levels, which significantly increases the accuracy of the heading angle estimation.

5. Conclusions

The implementation of the defined algorithm resulted in the development of a concise software code, occupying approximately 5 KB of the controller memory and having the processing speed of 0.01 sec per 1 filtering cycle (for Arduino UNO controller).

The algorithm testing demonstrated that with the root-mean-square deviation of the calculated heading equal to 10 angular degrees, root-mean-square deviation of MP angular rotation rate equal to 1 deg/sec, and MP turning rate of 20 deg/sec, the heading estimate error is reduced to a value below one angular degree within a few seconds of filtering.

References

[1] Jazwinski A H 2007 Stochastic Processes and Filtering Theory (New York: Dover Publications, Inc.)
[2] Zhidkov V N, Kim N V, Mamonov A V et al 2019 On the problem of developing software and hardware instruments for the social robot 12th International Conference on Developments in eSystems Engineering (DeSE) (Kazan, Russia) pp 239–242
[3] Zhidkov V N, Kim N V and Pimenov V N 2019 Mobile social robot Fifth All-Russia Research and Practical Workshop “Unmanned vehicles with artificial intelligence elements”
[4] Becker-Ehmck P, Karl M, Peters J and van der Smagt P 2020 Learning to fly via deep model-based reinforcement learning arXiv:2003.08876
[5] Bhardwaj M, Boots B and Mukadam M 2019 Differentiable gaussian process motion planning International Conference on Robotics and Automation
[6] Bojarski M, Del Testa D, Dworakowski D, Firner B, Flepp B, Goyal P, Jackel L, Monfort M, Muller U, Zhang J et al 2016 End to end learning for self-driving cars
[7] Bruce J, Sunderhauf N, Mirowski P, Hadsell R and Milford M 2017 One-shot reinforcement learning for robot navigation with interactive replay arXiv:1711.10137
[8] Chen C, Liu Y, Kreiss S and Alahi A 2019 Crowd-robot interaction: Crowd-aware robot navigation with attention based deep reinforcement learning arXiv:1809.08835