

Autonomous mobile robot localization using Kalman filter

*Nabil Zhafri Mohd Nasir*¹, *Muhammad Aizzat Zakaria*^{1,*}, *Saifudin Razali*¹, and *Mohd Yazid bin Abu*¹

¹Faculty of Manufacturing Engineering, Universiti Malaysia Pahang, 26600 Pekan, Pahang, Malaysia

Abstract. Autonomous mobile robot field has gain interest among researchers in recent years. The ability of a mobile robot to locate its current position and surrounding environment is the fundamental in order for it to operate autonomously, which commonly known as localization. Localization of mobile robot are commonly affected by the inaccuracy of the sensors. These inaccuracies are caused by various factors which includes internal interferences of the sensor and external environment noises. In order to overcome these noises, a filtering method is required in order to improve the mobile robot's localization. In this research, a 2-wheeled-drive (2WD) mobile robot will be used as platform. The odometers, inertial measurement unit (IMU), and ultrasonic sensors are used for data collection. Data collected is processed using Kalman filter to predict and correct the error from these sensors reading. The differential drive model and measurement model which estimates the environmental noises and predict a correction are used in this research. Based on the simulation and experimental results, the x, y and heading was corrected by converging the error to 10 mm, 10 mm and 0.06 rad respectively.

1 Introduction

A fully automated mobile robot will require the robot to be able to pinpoint its current poses and heading in a stated map of an environment. The process of determining its pose is named localization. Mobile robot localization often gets intact with accuracy and precision problem. Commonly known as *position tracking* or *position estimation*. Localization is a fundamental perceptual problem in robotics [1].

In a study [2] stated that the real challenge in localization of mobile robot is to perform correct error association between data collected from sensor and its environmental model. Localization is the process of establishing correspondence between robot's local coordinate system and the map coordinate system, which is described in a global coordinate system and entirely independent of a robot's pose. Knowing the facts that most sensors integrated to robot are not to be a noise-free sensor [3], pinpointing the pose of a robot requires to be inferred from a certain preset data. Depending on a single sensor is usually insufficient for defining the robot's pose. Several sensors are required in order to increase the precision of the integrated data.

* Corresponding author: maizzat@ump.edu.my

Acquired data from robot sensors is required to be combined in order to estimate the actual pose of the robot. The Kalman filter is considered to be the best possible, optimal, estimator for systems with uncertainty [4-5]. The Kalman filter is a data processing algorithm or filter, which is useful for the reason that only knowledge about system inputs and outputs is available for estimation purposes [6]. In short, Kalman filter is an iterative mathematical process which uses sets of equations and consecutive data inputs in estimating the true value in a system containing uncertainty. Kalman filter is implemented for combining the sensory data for estimating the state of the robot which is the location and current heading of the robot [7].

This paper presents the implementation of Kalman filter to improve the mobile robot localization by developing a sets of mathematical models correlates to the driving model, position prediction, and position estimation models. Furthermore, the fabrication of 2-wheeled mobile robot for the purpose of collecting measurements of its surrounding based on the integrated sensors. The study is bounded by only considering a static, known indoor environment.

2 Simulation and experimental setup

2.1 Kalman filter modelling

In general, mobile robots require a driving system which allows it to move about its specified surrounding environment. Mobile robot uses in this project are based on a two-wheeled mobile robot driving system. These two-wheeled mobile robot are controlled referencing differential drive system which makes it the least complex system and suitable for a common indoor environment navigation. Thus, less complexity in its position estimation.

2.1.1 Position prediction model

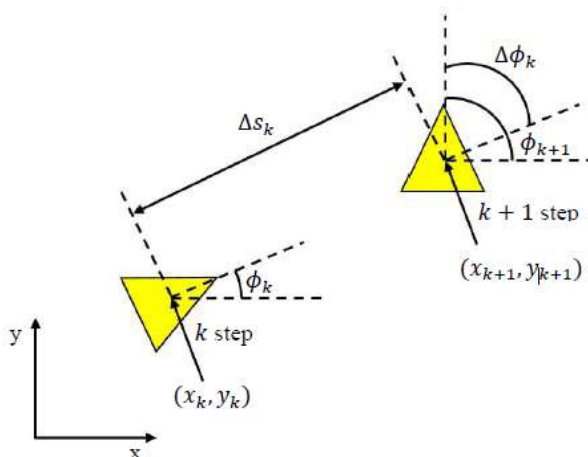


Fig. 1. Initial and predicted, position and orientation.

Figure 1 shows the position and orientation of mobile robot for certain different time steps. An initial coordinate denotes X_k are made up of x and y coordinate and heading angle θ referencing the centre of gravity of the mobile robot. The time step is denoted as k . The control input denotes as U_k are based on robot relative displacement, s , and heading

angle, ϕ . is first assumed that no noise is affecting the system for constructing the system model. This is to idealize the system to accurately estimate robots pose.

$$X_k = [x_k, y_k, \phi_k]^T \quad (1)$$

$$U_k = [\Delta s_k, \Delta \phi_k]^T \quad (2)$$

Updated position of a mobile robot X_k , are estimated based on its previous position X_{k-1} , previous heading angle $\Delta \theta_{k-1}$, and the previous travels displacement s_{k-1} .

$$x_k = f_x(X_{k-1}, U_{k-1}) = x_{k-1} + \Delta s_k \cos\left(\phi_{k-1} + \frac{\Delta \phi_k}{2}\right) \quad (3)$$

$$y_k = f_y(X_{k-1}, U_{k-1}) = y_{k-1} + \Delta s_k \sin\left(\phi_{k-1} + \frac{\Delta \phi_k}{2}\right) \quad (4)$$

$$\phi_k = f_\theta(X_{k-1}, U_{k-1}) = \phi_{k-1} + \Delta \phi_k \quad (5)$$

The priori coordinate and priori error covariance can be measured as

$$\hat{X}_k^- = f(\hat{X}_{k-1}^+, \hat{U}_{k-1}) \quad (6)$$

$$P_k^- = \nabla f_x P_{k-1}^+ \nabla f_x^T + \nabla f_u U_{k-1} \nabla f_u^T + Q_{k-1} \quad (7)$$

Denoting U_k as controlled noise which is due respect to steer angle and velocity. ∇f_x and ∇f_u are Jacobian matrices describing the function of state transition referencing the state X and control input U .

2.1.2 Position estimation model

Reference walls are placed as landmarks with a ready set distance from square path sets for the mobile robot. Actual range between the robot and the reference wall are estimated using ultrasonic sensors. The centre of gravity of the mobile robot, (x_k, y_k) are set to be the origin on the local coordinate frame. Coordinate positioning the ultrasonic sensor are defined as (x'_s, y'_s) as shown in Figure 2.

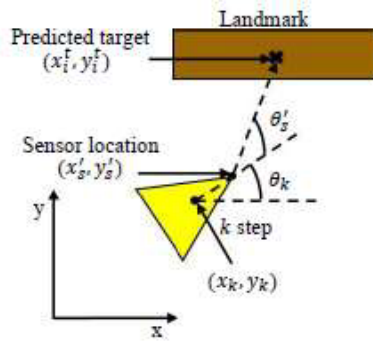


Fig. 2. Landmark prediction measurement.

Geometrical transformation of translation and rotation are used for converting the local coordinate frame to global coordinate frame.

$$\begin{bmatrix} X_{s,k} \\ Y_{s,k} \\ \theta_{s,k} \end{bmatrix} = \begin{bmatrix} x_k + x'_s \cos(\theta_k) - y'_s \cdot \sin(\theta_k) \\ y_k + x'_s \sin(\theta_k) - y'_s \cdot \cos(\theta_k) \\ \theta_k + \theta'_s \end{bmatrix} \quad (8)$$

$$l_{i,k}^t = \begin{bmatrix} x_{i,k}^t \\ y_{i,k}^t \end{bmatrix} = \begin{bmatrix} X_{s,k} + h_{i,k} \cos(\theta_{s,k}) \\ Y_{s,k} + h_{i,k} \sin(\theta_{s,k}) \end{bmatrix} \quad (9)$$

denoting $l_{i,k}^t$ as the coordinate of the predicted landmark, and $h_{i,k}$ as the actual displacement measured from the ultrasonic sensor. The distance $h_{i,k}^t$ between ultrasonic sensor and landmark can be estimated one the predicted landmark coordinate $l_{i,k}^t$ is acquired, based on

$$h_{i,k} = \sqrt{(X_{s,k} - x_{i,k}^t)^2 + (Y_{s,k} - y_{i,k}^t)^2} \quad (10)$$

Difference between predicted heading and the actual heading measured by the encoder and yaw orientation θ_k get from IMU are determined by

$$Z_k = \begin{bmatrix} \sqrt{(X_{s,k} - x_{i,k}^t)^2 + (Y_{s,k} - y_{i,k}^t)^2} - h_{i,k}^t \\ \theta_{i,k} - \theta_{k,IMU} \end{bmatrix} \quad (11)$$

Kalman gain, K_k , the posterior state, \hat{X}_k^+ , and the posterior error covariance, P_k^+ can be estimated and update by taking

$$K_k = P_k^- H^T (H P_k^- H^T + R_k)^{-1} \quad (12)$$

$$\hat{X}_k^+ = \hat{X}_k^- + K_k (Z_k) \quad (13)$$

$$P_k^+ = P_k^- - P_k^- K_k H \tag{14}$$

2.2 Experimental setup

2.2.1 Design of experiment

Figure 3 shows the robot and environmental setup being done to monitor the mobile robot poses in a known static environment. Four landmarks is set as B1, B2, B3, ad B4. The initial position of the robot is placed at the bottom right corner of the 1x1 meter square, and it is defined as the origin coordinate (x, y, θ) . The robot is programmed to move along the four line creating a square path referencing the 1x1 meter square making counter-clockwise movements.

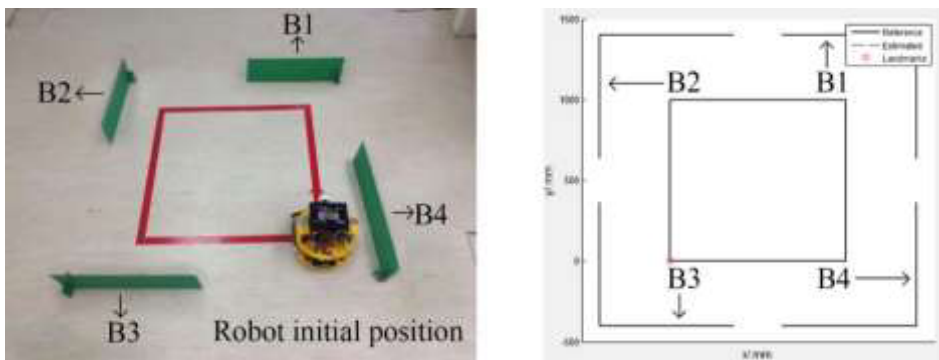


Fig. 3. Square path experimental setup and GUI representation.



Fig. 4. Ultrasonic actual distance measurement.

Error variance on the ultrasonic sensor are determined by comparing the raw data initially measured by the sensors, and the actual-measured distances between the sensors and the landmark placed. By means of performance, the sensors are capable of detecting range 0.04 to 30 meter, with resolution of 0.01 meter.

2.2.2 Robotics parameters

Based on the calibration done on sensors and several measurements on the mobile robot's base, robotics parameters are declared as presets for the system.

Table 1. Specified mobile robot parameters.

Parameter	Description	Value
w_d	Wheel diameter (<i>mm</i>)	145
w_b	Wheelbase diameter (<i>mm</i>)	285
n_p	Encoder pulse per revolution	2000
θ'_s	Ultrasonic angle (<i>deg</i>)	25°
$\sigma_{\Delta s}^2$	Odometry distance error (<i>mm</i>)	720
$\sigma_{\Delta \theta}^2$	Odometry orientation error (<i>deg</i>)	1°
σ_r^2	Dual ultrasonic error (<i>mm</i>)	72

3 Results and discussion

3.1 Error variance

To capture the motion data, mobile robot was programmed to navigate through 1x1 meter square path for 10 cycles to determine the encoder error. Odometry error variance, position displacement and robot's heading are closely noted.

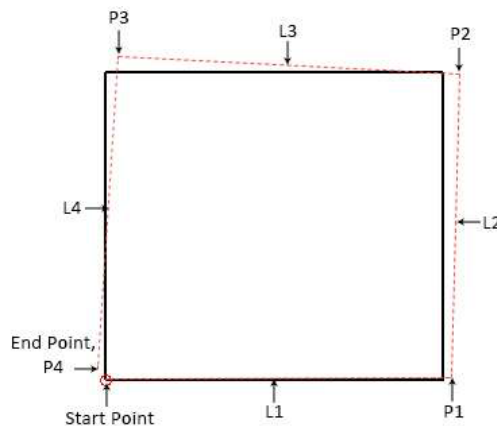


Fig. 5. Single run for programmed path.

Robotics trajectory for a single run is as shown in Figure 5 where points of measurement are denoted as labelled. From the figure, it can be seen that the reading is increasingly deviated from its original value due to error affecting its performance.

3.2 Kalman filter localization

Based on the previously stated mobile robot parameters, the error variance of the ultrasonic sensor, distances and orientation are used as the initial state noise covariance and the measurement noise covariance to be post process in the Kalman filter for determining the mobile robot location in current environment. Figure 6 (a) shows the plotted estimated path of the mobile robot observing only on the odometry for localization. The estimated path is drifted away compared to the true trajectory of the robot. Since Kalman filter is done offline, thus the robot is not programmed with any measurement model which able to correct the position and heading of the mobile robot during its operation. This causes the deviation increases over time step.

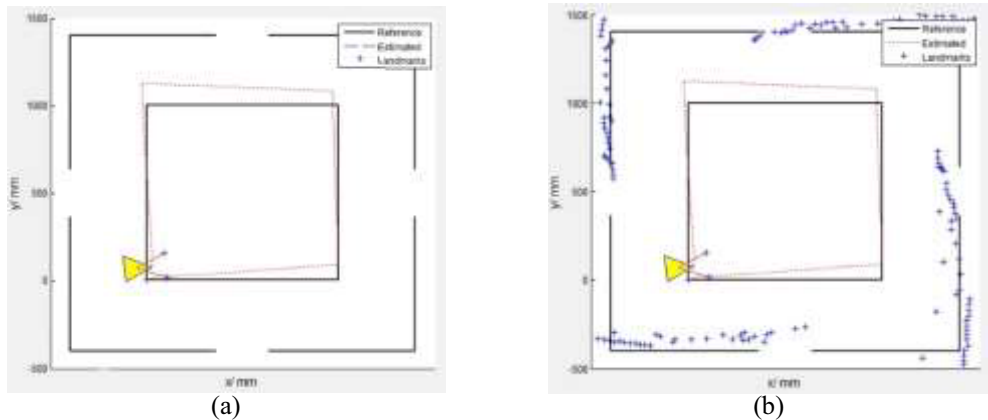


Fig. 6. Estimated trajectory plotted data (a) Using odometry. (b) Includes ultrasonic sensor.

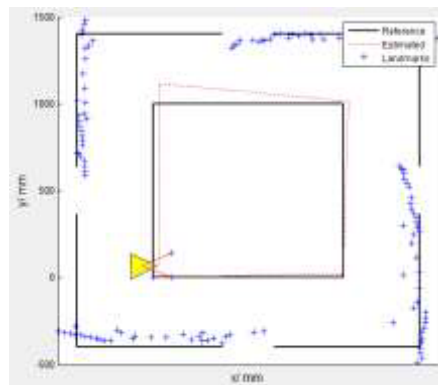


Fig. 7. Estimated trajectory plotted data using Kalman Filter.

Figure 7 shows the exact same data as in Figure 6 which plotted all the sensory data but using Kalman filter as inaccuracy filter. The path can be seen clearly to have improved in its accuracy and precision. It had improved the localization error through the runs. Based on the experiment done, it can be assumed that the average error affecting the robot's localization as shown in Table 2.

Table 2. Average position and orientation error.

Algorithm	Error		
	x, mm	y, mm	ϕ, mm
Odometry Localization	43	48	0.078
Kalman Filter Localization	10	10	0.060

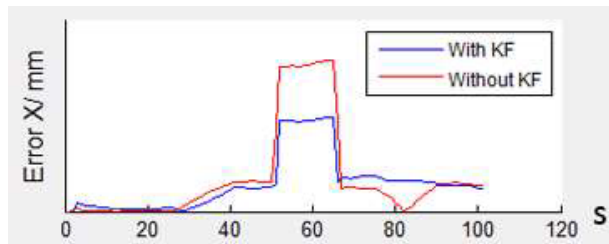


Fig. 8. Error in x-orientation.

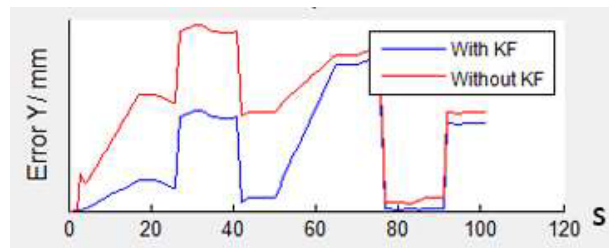


Fig. 9. Error in y-orientation.

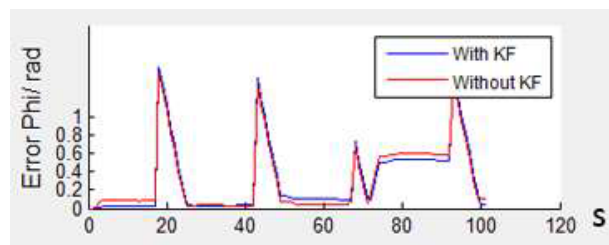


Fig. 10. Error in phi-orientation.

Figure 8, Figure 9, Figure 10 are showing the error comparisons between the estimated and the Kalman-filtered error for the x-coordinate, y-coordinate and the θ -orientation of the mobile robot for the above plotted mobile robot trajectory. Based on the figures, 0 on the x-axis represents the actual desired data which is comprised with zero error. Data plotted based on raw data are represented with the red lines, while the data implemented with Kalman filter are distinguished with the blue lines. From the results shown by the figure, it

can be clearly seen that Kalman filter implementation provides more accurate data which is much closer to the actual data. The Kalman filter implemented is proven to reduce an average of 23.25% of the error in mobile robot localization.

4 Conclusions

In this paper, the localization of the robot was improved by the implementation of Kalman filter. Based on the experimental and simulation results of the proposed system, the x, y, and heading or orientation errors is observed to be reduced to 0.1 meter, 0.1 meter and 0.06 rad angles respectively. The implementation of Kalman filter somehow introduces to delay in the systems processing. Kalman filter also depends greatly on the prediction model, an error in creating the prediction model will surely cause all the output data to be deviated from the required reference data.

The authors are obliged to thank the Universiti Malaysia Pahang for providing a platform for the project and providing financial assistance under research grant, RDU16043.

References

1. Siegwart and Nourbaksh, "Mobile Robot Localization," *Introd. to Auton. Mob. Robot.*, 2004.
2. F. Hardarson, "Stability analysis and synthesis of statically balanced walking for quadruped robots," *R. Inst. Technol.*, p. 154, 2002.
3. E. Elnahrawy and B. Nath, "Cleaning and querying noisy sensors," *Proc. 2nd ACM Int. Conf. Wirel. Sens. networks Appl. - WSNA '03*, p. 78, 2003.
4. R. E. Kalman, "A New Approach to Linear Filtering and Prediction Problems," *J. Basic Eng.*, vol. 82, no. 1, p. 35, 1960.
5. M. Grewal and A. Andrews, *Kalman filtering: theory and practice using MATLAB*, vol. 5. 2011.
6. R. Negenborn, "Robot localization and kalman filters," *Www. Negenborn Net/KalmanFilterImplementation (thesis). Pdf*, 2003.
7. J. Biswas and M. Veloso, "Multi-sensor mobile robot localization for diverse environments," *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, vol. 8371 LNAI, pp. 468–479, 2014.