

Analysis of Indoor Robot Localization Using Ultrasonic Sensors

Sairah Naveed¹, Nak Yong Ko²

¹Department of Control and Instrumentation Engineering, Chosun University, Gwangju, Korea

²Department of Control, Instrumentation and Robotic Engineering, Chosun University, Gwangju, Korea

Ijfis

Abstract

This paper analyzes the Monte Carlo localization (MCL) method, which estimates the pose of an indoor mobile robot. A mobile robot must know where it is to navigate in an indoor environment. The MCL technique is one of the most influential and popular techniques for estimation of robot position and orientation using a particle filter. For the analysis, we perform experiments in an indoor environment with a differential drive robot and ultrasonic range sensor system. The analysis uses MATLAB for implementation of the MCL and investigates the effects of the control parameters on the MCL performance. The control parameters are the uncertainty of the motion model of the mobile robot and the noise level of the measurement model of the range sensor.

Keywords: Localization, Mobile robot, Motion model, Measurement model, Monte Carlo localization, Range sensor

1. Introduction

The estimation of position and orientation is vital for navigation of a mobile robot. The estimation of location, called localization, has been studied extensively. There are many methods that have been proposed and implemented. These methods include simple dead reckoning, least squares, and other more complicated filtering approaches.

The most intuitive and trivial method, which is also impractical, is dead reckoning. This method integrates the velocity over time to determine the change in robot position from its starting position. Other localization systems use beacons [1–5] placed at known positions in the environment. The beacons use range data to the robot. The range sensors use ultrasonic or radio frequency signals to determine the distances between the robot and the beacons. The least squares method or filtering method uses the range data to estimate the position of the mobile robot. Uncertainty in robot motion and the noise in the range measurement affect the performance of the estimation; additionally, some control parameters of the methods should be adjusted according to the levels of uncertainty and noise.

There have been several filtering approaches for localization. Several of the major approaches are the Kalman filter (KF), extended Kalman filter (EKF) [6], unscented Kalman filter (UKF), and particle filter (PF) [7–9]. All of these filters follow the Bayesian filter approach. The variations in the Kalman filter assume that the uncertainties in robot motion and measurement are Gaussian.

Received: Feb. 21, 2014

Revised : Mar. 18, 2014

Accepted: Mar. 19, 2014

Correspondence to: Nak Yong Ko

(nyko@chosun.ac.kr)

©The Korean Institute of Intelligent Systems

©This is an Open Access article distributed under the terms of the Creative Commons Attribution Non-Commercial License (<http://creativecommons.org/licenses/by-nc/3.0/>) which permits unrestricted non-commercial use, distribution, and reproduction in any medium, provided the original work is properly cited.

The pure KF only uses a linear model of robot motion and sensor measurement. To handle the nonlinearity in robot motion and sensor measurement, the EKF approximates the nonlinearity with a first-order linear model. UKF, a recent development of KF, does not approximate the robot motion and sensor measurement. Instead, it uses a nonlinear model of motion and measurement as it is. It uses samples called the sigma points to describe the probabilistic properties of the robot motion and sensor measurement. UKF also assumes that all the uncertainty involved in the system is Gaussian.

Though EKF has been used widely and successfully for mobile robot localization, it sometimes provides unacceptable results because real data can be very complex, involving elements of non-Gaussian uncertainty, high dimensionality, and nonlinearity. Moreover, EKF requires derivation of a highly complicated Jacobian for linearization. Therefore, Monte Carlo methods have been introduced for localization.

Monte Carlo localization (MCL) is based on a particle filter [10–12]. MCL can solve the global localization and kidnapped robot problem in a highly robust and efficient way. The Monte Carlo localization method uses a set of samples, called the particles, to depict the probabilistic features of the robot positions. In other words, rather than approximating probabilistic distributions in parametric form, as is the case for KFs, it describes the probability as it is using the particles. MCL has the great advantage of not being subject to linearity or Gaussian constraints on the robot motion and sensor measurement.

This paper is concerned with the estimation of robot position and orientation in an indoor environment. We have used a sensor system comprised of static ultrasonic beacons and one mobile receiver installed on the robot. The robot navigates through predefined path points. The exteroceptive measurement information is the range data between the robot and the beacons. In this experiment, we have used a differential drive mobile robot. The paper contributes to understanding the effect of control parameters on the localization performance. How the uncertainty in robot motion and sensor measurement affects the location estimation has been exploited through experiments.

The remainder of this paper is organized as follows. In Section 2, MCL and its fundamentals are discussed. Section 3 illustrates details of the experiment and analysis of MCL. Section 4 covers the discussion of the experiments, and Section 5 concludes the paper.

2. Monte Carlo Localization and its Models

The MCL method iterates sampling and importance resampling in the frame of the Bayesian filter [13] approach for localization of a mobile robot. It is alternatively known as the bootstrap filter [14], the Monte Carlo filter [15], the condensation algorithm [16], or the survival of the fittest algorithm [17]. All of these methods are generally known as particle filters.

MCL method can approximate almost any probabilistic distribution of practical importance. It is not bound to a limited parametric subset of probabilistic distributions as in the case of an EKF localization method for a Gaussian distribution. Increasing the total number of particles increases the accuracy of the approximation. However, a large number of particles degrades the computational efficiency that is needed for real-time application of the MCL. The idea of MCL is to represent a belief about a robot position with a particle set χ_t , each representing a hypothesis on the robot pose (x, y, θ) .

Monte Carlo localization repeats three steps: 1) application of a motion model, 2) application of a measurement model, and 3) resampling of particles. These three steps are explained in Table 1 using pseudocode.

Table 1. Monte Carlo localization (MCL) algorithm

MCL algorithm (χ_{t-1}, u_t, z_t, m)
$\bar{\chi}_t = \chi_t = \emptyset$
for $p = 1$ to P
$\mu_t^{[p]} = \text{Motion model}(u_t, \mu_{t-1}^{[p]})$
$\mu_{t, \text{belief}}^{[p]} = \text{Measurement model}(z_t, \mu_t^{[p]}, m)$
$\mu_t^{[p]} = \text{Resampling}(\{\mu_t^{[p]}\}, \mu_{t, \text{belief}}^{[p]} p = 1, \dots, P)$
end for
return χ_t

In Table 1, the prediction phase starts from the set of particles χ_{t-1} and applies the motion model to each particle $\mu_{t-1}^{[p]}$ in a particle set χ_{t-1} . In a measurement model, the importance factor, sometimes called the belief $\mu_{t, \text{belief}}^{[p]}$ of each particle is determined. The information in the measurement z_t is incorporated into the particle set χ_t via the importance factor $\mu_{t, \text{belief}}^{[p]}$. After the belief $\mu_{t, \text{belief}}^{[p]}$ calculation, resampling is performed on the basis of belief $\mu_{t, \text{belief}}^{[p]}$. Resampling transforms the particle set into another particle set of the same size, which finally yields the estimated particle set χ_t for time t .

The resulting sample set χ_{t-1} usually consists of many dupli-

cates. It refocuses the particle set into a region of high posterior probabilities. The particles that were not contained in χ_t have lower belief. It should be noted that the resampling process neither includes the particles in order from the highest belief nor excludes the particles in order from the lowest belief. Thus, the set $\chi_t = \{x_t^{[1]}, x_t^{[2]}, x_t^{[3]}, \dots, x_t^{[p]}\}$ consists of P particles, which represent the probable locations of the robot at time t .

2.1 Motion Model

A motion model is used to predict the pose $\mu_t^{[p]}$ of the robot from the previous pose $\mu_{t-1}^{[p]}$ using the control command or proprioceptive motion sensing (v, ω). Table 2 shows the proposed motion model that was used in MCL [6].

Table 2. Motion model

Motion model($\chi_{t-1}, v, \omega, \Delta t, motionpara$)
for $p = 1$ to P $\hat{v} = v + \text{sample}(\alpha_1 v^2 + \alpha_2 \omega^2)$ $\hat{\omega} = \omega + \text{sample}(\alpha_3 v^2 + \alpha_4 \omega^2)$ $\hat{\gamma} = \text{sample}(\alpha_5 v^2 + \alpha_6 \omega^2)$ $\mu_{t,x}^{[p]} = \mu_{t-1,x}^{[p]} - \frac{\hat{v}}{\hat{\omega}} \sin \mu_{t-1,\theta}^{[p]} + \frac{\hat{v}}{\hat{\omega}} \sin(\mu_{t-1,\theta}^{[p]} + \hat{\omega} \Delta t)$ $\mu_{t,y}^{[p]} = \mu_{t-1,y}^{[p]} + \frac{\hat{v}}{\hat{\omega}} \cos \mu_{t-1,\theta}^{[p]} - \frac{\hat{v}}{\hat{\omega}} \cos(\mu_{t-1,\theta}^{[p]} + \hat{\omega} \Delta t)$ $\mu_{t,\theta}^{[p]} = \mu_{t-1,\theta}^{[p]} + \hat{\omega} \Delta t + \hat{\gamma} \Delta t$ end for return $\mu_t^{[p]}$

In Table 2, x_t^p, y_t^p , and θ_t^p constitute a particle $\mu_t^{[p]}$ that represents the pose of the robot. Δt is the algorithm time step, and v and ω are the translational and rotational velocities measured by the wheel encoders on the robot. The variable *motionpara* consists of $\alpha_1, \alpha_2, \alpha_3, \alpha_4, \alpha_5$, and α_6 , which represent the motion uncertainty. The function *sample* generates a random number from the Gaussian random variable with zero mean and variance of $\alpha_i v^2 + \alpha_j \omega^2$.

2.2 Measurement Model

In the measurement model, the belief $\mu_{t,belief}^{[p]}$ of the predicted particles is computed utilizing the received range information from the beacons. Table 3 shows the proposed measurement model that we have to implement in MCL. \bar{z}_t^i is the distance between the predicted particle and the beacons.

$\text{prob}(r_t^i - \hat{r}_t^i, \delta_r)$ in Table 3 is the Gaussian probability distribution of the measurement noise in the range information.

Table 3. Measurement model

Measurement model($\mu_t^{[p]}, z_t, m$)
for all particles $= (\mu_{t,x}^{[p]}, \mu_{t,y}^{[p]}, \mu_{t,\theta}^{[p]}, \mu_{t,belief}^{[p]})$ for all beacons $z_t^i = r_t^i$ do $\bar{z}_t^i = \hat{r}_t^i = \sqrt{(m_{i,x} - \mu_{t,x}^{[p]})^2 + (m_{i,y} - \mu_{t,y}^{[p]})^2}$ $\mu_{t,belief}^{[p]} = \mu_{t,belief}^{[p]} \times \text{prob}(r_t^i - \hat{r}_t^i, \delta_r)$ end for end for return $\mu_{t,belief}$

The measurement noise can be caused by unexpected objects, crosstalk between different signals, and specular reflection of the signals.

2.3 Resampling

Finally, all particles are resampled, i.e., a new set of particles are drawn from the current set χ_t on the basis of the belief $\mu_{t,belief}^{[p]}$. We use systematic resampling, which is also known as stochastic universal resampling because it is fast and simple to implement.

3. Experiment and Analysis

The experiments are conducted in a classroom with chairs and tables on which computers and monitors are located. The four ultrasonic beacons are installed on the ceiling.

Figures 1, 2 shows the experimental setup which indicates the trajectory of the robot motion and the location of ultrasonic beacons. The positions of the beacons and way points are provided in Tables 4 and 5, respectively.

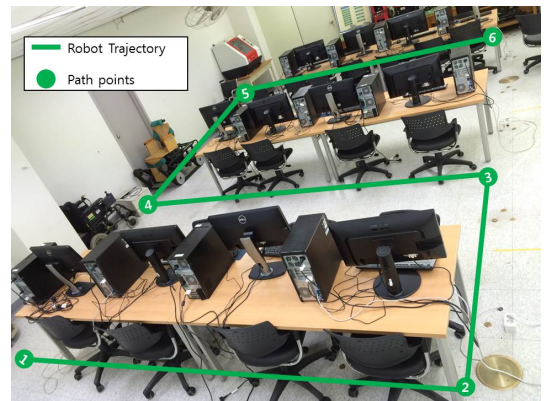


Figure 1. Capstone design laboratory (environment for robot navigation experiment).

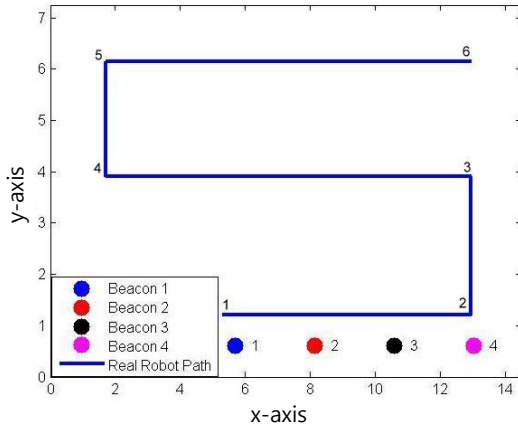


Figure 2. Experimental setup.

We use the MRP-NRLAB 02 differential drive robot (see Figure 3) manufactured by RED ONE technologies and the USAT A105 ultrasonic sensor system (see Figure 4) from the company Korea LPS. The work area for the experiment is $14.5m \times 7.25m$. The receiver of the sensor system is mounted on the robot, and beacons are attached on the ceiling of the room.

Table 4. Locations of the beacons

Beacon	x position (m)	y position (m)
1	5.7	0.6
2	8.15	0.6
3	10.6	0.6
4	13.05	0.6

Table 5. Locations of the way points

Way point	x position (m)	y position (m)
1	5.3	1.2
2	12.95	1.2
3	12.95	3.9
4	1.7	3.9
5	1.7	6.15
6	1.7	6.15

The robot is controlled by a joystick and uses the wheel encoder data to calculate the translational velocity v and rotational velocity ω . The ranges between the robot and beacons are measured by the ultrasonic sensor system to correct the predicted robot location. The initial pose (x_o, y_o, θ_o) of the robot is $(5.3m, 1.2m, 0.00rad)$. The robot navigates with a translational velocity of $v = 0.3m/sec$ and rotational velocity $\omega = 0.1rad/sec$. The MCL is implemented using MATLAB with 300 particles. Table 6 lists the values of the control parameter that are used to investigate the performance of MCL.



Figure 3. Differential drive robot MRP-NRLAB02.

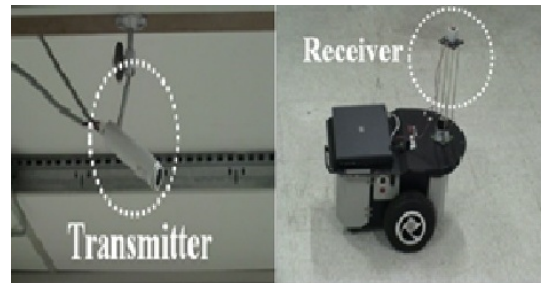


Figure 4. Ultrasonic sensor system USAT A105.

During the experiment, the ultrasonic range measurement system often fails in detecting some of the ranges. Because of the failure, ambiguity and large estimation error are observed in the path segments from 3 to 4, 4 to 5, and 5 to 6.

Figure 5 shows the plots of the estimated trajectories by colored lines according to the values of the control parameters, as shown in Table 6. The asterisks represent the positions of the beacons. Figures 6 and 7 illustrate the distance error and orientation error between the estimated and real robot trajectories. Tables 7 and 8 list the mean and standard deviation of the distance error and orientation error of the estimation, respectively. From the experimental results, the estimation error of the robot is observed to decrease when the appropriate control parameter values are used.

4. Discussion

A variety of techniques and suggestions have been proposed for mobile robot localization [18–24]. It is evident that a comparison of different techniques was difficult because of a lack of commonly accepted test standards and procedures. We have developed our own experimental environment using four ultrasonic beacons and six way points for robot navigation. An

Table 6. Control parameter values

Motion uncertainty	Case (i)	Case (ii)	Case (iii)
α_1	0.6	0.7	0.8
α_2	0.1	0.2	0.3
α_3	0.6	0.7	0.8
α_4	0.2	0.3	0.4
α_5	0.13	0.14	0.15
α_6	0.2	0.3	0.4
Measurement noise	Case (i)	Case (ii)	Case (iii)
δ_r	0.15	0.25	0.35

Table 7. Mean of distance error and orientation error

Case	Mean of distance error	Mean of orientation error
i	0.4173	0.1416
ii	0.4046	0.1008
iii	0.2210	0.0278

Table 8. Standard deviation (SD) of distance error and orientation error

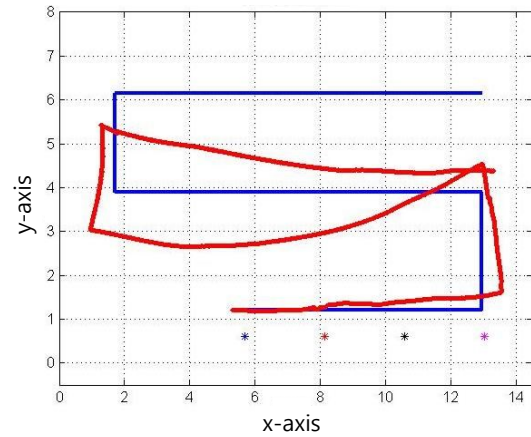
Case	SD of distance error	SD orientation error
i	0.4761	0.1102
ii	0.3331	0.0868
iii	0.1760	0.0269

analysis of MCL was performed under three different cases, as provided in Table 6. Each case in Table 6 is categorized according to the low, medium, and high values of motion uncertainty and measurement noise. The investigation of MCL analysis was based on the following:

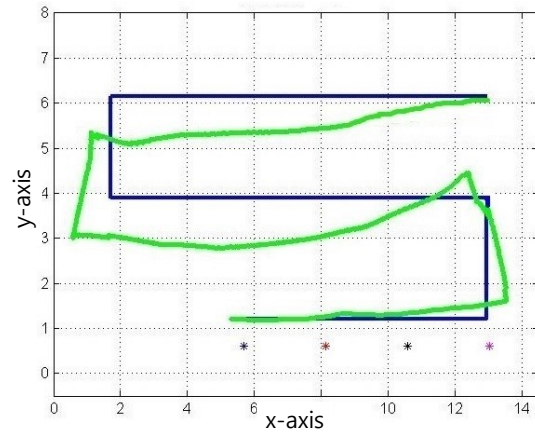
- 1) plotting the trajectory of the estimated position against the real robot trajectory,
- 2) calculating and plotting the distance error and orientation error between the real robot position and the estimated position, and
- 3) evaluation of mean and standard deviation of the distance error and orientation error.

Large estimation error is observed in (a) and (b) in Figure 5, which corresponds to cases (i) and (ii), respectively. Figure 5c, which corresponds to case (iii), shows the least estimation error. The results also suggest that the motion error and measurement noise of the experiment are relatively high.

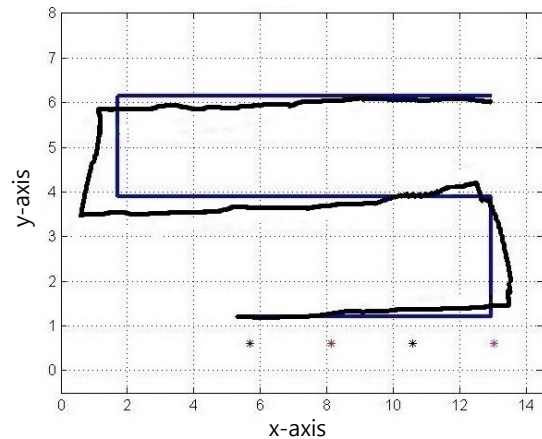
Figures 7 and 8 show the distance error and orientation error for the experiments. The mean and standard deviation of the errors are listed in Tables 7 and 8, respectively. The results are shown in Figures 5 through 7, and Tables 7 and 8 indicate that the proper selection of control parameters is crucial for



(a) The red line shows the estimated trajectory according to the control parameter values of case (i) in Table 6.

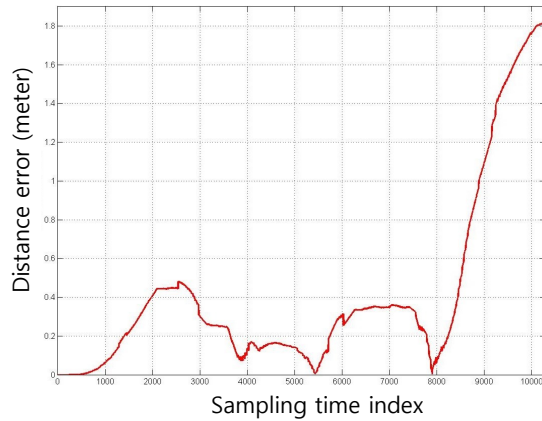


(b) The green line shows the estimated trajectory according to the control parameter values of case (ii) in Table 6.

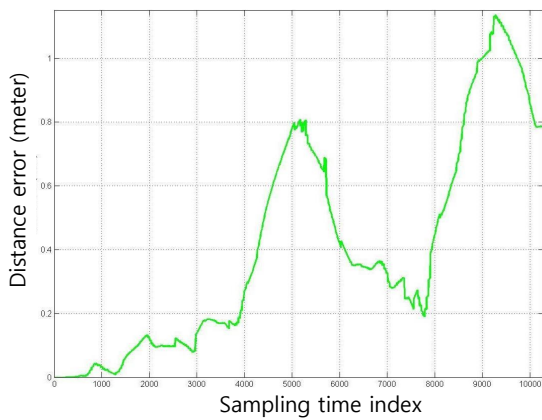


(c) The black line shows the estimated trajectory according to the control parameter values of case (iii) in Table 6.

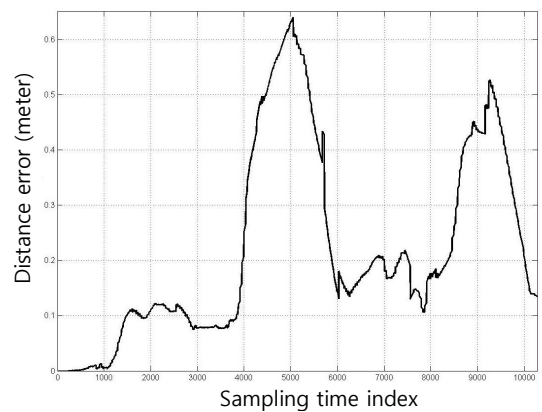
Figure 5. Comparison of estimated trajectories using different parameter values.



(a) Distance error plot between particles and real robot trajectory related to case (i) in Table 6.



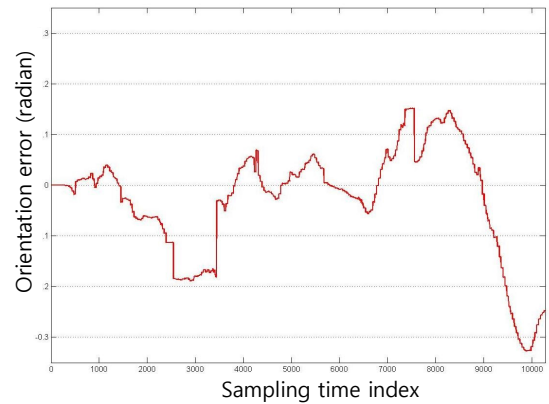
(b) Distance error plot between particles and real robot trajectory related to case (ii) in Table 6.



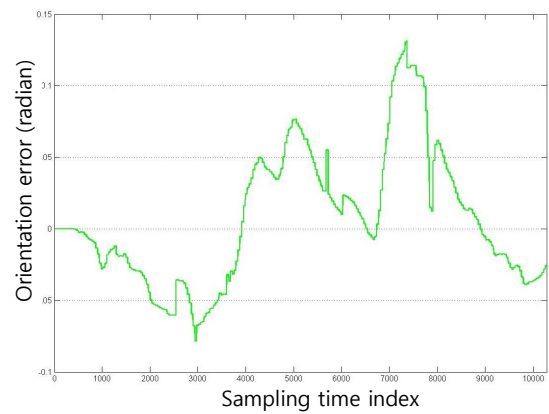
(c) Distance error plot between particles and real robot trajectory related to case (iii) in Table 6.

Figure 6. Distance error between estimated and actual robot trajectories.

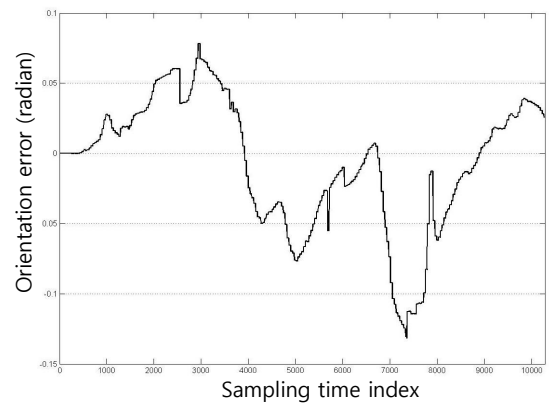
the use of MCL. MCL fails if the values of the control parameters assume that the uncertainty of robot motion is lower than



(a) Orientation error plot between estimated and real robot trajectory related to case (i) in Table 6.



(b) Orientation error plot between estimated and real robot trajectory related to case (ii) in Table 6.



(c) Orientation error plot between estimated and real robot trajectory related to case (iii) in Table 6.

Figure 7. Orientation error between estimated and actual robot orientation.

the actual uncertainty. Likewise, assuming that the measurement uncertainty is lower than that of the actual measurement deteriorates the estimation performance.

5. Conclusion

This paper shows results of MCL used for localization of a mobile robot in an indoor environment. The experiment uses a differential drive robot, which uses wheel encoder data and range data from four fixed beacons.

The experiments compare three different cases, which represent localization under different control parameter values. The control parameters adjust the uncertainty of robot motion and sensor measurement noise. From the comparison, it is concluded that assuming that the motion uncertainty and measurement noise are lower than the actual values causes poor estimation performance. This is the case for KF approaches, where the mismatch in the process error level and measurement error level causes poor estimation performance.

Conflict of Interest

No potential conflict of interest relevant to this article was reported.

Acknowledgments

This work was supported by an NRF grant funded by the Korean government (NRF-2013-R1A1A4A01-012469).

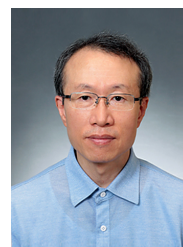
References

- [1] T. S. Jin, H. S. Kim, and J. W. Kim, "Landmark detection based on sensor fusion for mobile robot navigation in a varying environment," *International Journal of Fuzzy Logic and Intelligent Systems*, vol. 10, no. 4, pp. 281-286, Dec. 2010. <http://dx.doi.org/10.5391/IJFIS.2010.10.4.281>
- [2] M. Peca, "Ultrasonic localization of mobile robot using active beacons and code correlation," in *Research and Education in Robotics-EUROBOT 2009, Communications in Computer and Information Science Vol. 82*, A. Gottscheber, D. Obdrlek, and C. Schmidt, Eds. Heidelberg, Berlin: Springer, 2010, pp. 116-130. http://dx.doi.org/10.1007/978-3-642-16370-8_11
- [3] S. C. Kang and T. S. Jin, "Global map building and navigation of mobile robot based on ultrasonic sensor data fusion," *International Journal of Fuzzy Logic and Intelligent Systems*, vol. 7, no. 3, pp. 198-204, Sep. 2007.
- [4] L. Moreno, J. Armingol, S. Garrido, A. de la Escalera, and M. Salichs, "A genetic algorithm for mobile robot localization using ultrasonic sensors," *Journal of Intelligent and Robotic Systems*, vol. 34, no. 2, pp. 135-154, Jun. 2002. <http://dx.doi.org/10.1023/A:1015664517164>
- [5] J. J. Leonard and H. F. Durrant-Whyte, "Mobile robot localization by tracking geometric beacons," *IEEE Transactions on Robotics and Automation*, vol. 7, no. 3, pp. 376-382, Jun. 1991. <http://dx.doi.org/10.1109/70.88147>
- [6] G. Welch and G. Bishop, "An introduction to the Kalman filter," in *Proceedings of SIGGRAPH '01, The 28th International Conference on Computer Graphics and Interactive Techniques*, Los Angeles, CA, August 12-17.
- [7] W. Jeong, Y. J. Kim, J. O. Lee, and M. T. Lim, "Localization of mobile robot using particle filter," in *SICE-ICASE International Joint Conference*, Busan, Korea, October 18-21, 2006, pp. 3031-3034. <http://dx.doi.org/10.1109/SICE.2006.315151>
- [8] S. Thrun, W. Burgard, and D. Fox, *Probabilistic Robotics*, Cambridge, MA: MIT Press, 2005.
- [9] S. Thrun, "Particle filters in robotics," in *Proceedings of the Eighteenth Conference on Uncertainty in Artificial Intelligence*, Alberta, Canada, July, 2002, pp. 511-518.
- [10] S. Thrun, D. Fox, W. Burgard, and F. Dellaert, "Robust Monte Carlo localization for mobile robots," *Artificial Intelligence*, vol. 128, no. 1-2, pp. 99-141, May 2001. [http://dx.doi.org/10.1016/S0004-3702\(01\)00069-8](http://dx.doi.org/10.1016/S0004-3702(01)00069-8)
- [11] D. Fox, W. Burgard, F. Dellaert, and S. Thrun, "Monte Carlo localization: efficient position estimation for mobile robots," in *Proceedings of the Sixteenth National Conference on Artificial Intelligence and Eleventh Conference on Innovative Applications of Artificial Intelligence*, Orlando, FL, July 18-22, 1999, pp. 343-349.
- [12] F. Dellaert, D. Fox, W. Burgard, and S. Thrun, "Monte Carlo localization for mobile robots," in *Proceedings of the IEEE International Conference on Robotics and Automation*, Detroit, MI, May 10-15, 1999, pp. 1322-1328. <http://dx.doi.org/10.1109/ROBOT.1999.772544>
- [13] A. Doucet, S. Godsill, and C. Andrieu, "On sequential Monte Carlo sampling methods for Bayesian filtering," *Statistics and Computing*, vol. 10, no. 3, pp. 197-208, Jul. 2000. <http://dx.doi.org/10.1023/A:1008935410038>

- [14] N. J. Gordon, D. J. Salmond, and A. F. M. Smith, "Novel approach to nonlinear/non-Gaussian Bayesian state estimation," *IEE Proceedings, Part F: Radar and Signal Processing*, vol. 140, no. 2, pp. 107-113, Apr. 1993.
- [15] G. Kitagawa, "Monte Carlo filter and smoother for Non-Gaussian nonlinear state space models," *Journal of Computational and Graphical Statistics*, vol. 5, no. 1, pp. 1-25, Mar. 1996. <http://dx.doi.org/10.2307/1390750>
- [16] M. Isard and A. Blake, "CONDENSATION: conditional density propagation for visual tracking," *International Journal of Computer Vision*, vol. 29, no. 1, pp. 5-28, Aug. 1998. <http://dx.doi.org/10.1023/A:1008078328650>
- [17] K. Kanazawa, D. Koller, and S. Russell, "Stochastic simulation algorithms for dynamic probabilistic networks," in *Proceedings of the Eleventh conference on Uncertainty in artificial intelligence*, Montreal, Canada, August 18-20, 1995, pp. 346-351.
- [18] D. W. Kim, Y. Igor, E. S. Kang, and S. Jung, "Design and control of an omni directional cleaning robot based landmarks," *Journal of Korean Institute of Intelligent Systems*, vol. 23, no. 2, pp. 100-106, Apr. 2013. <http://dx.doi.org/10.5391/JKIIS.2013.23.2.100>
- [19] S. K. Jeong, T. G. Kim, and N. Y. Ko, "Programming toolkit for localization and simulation of a mobile robot," *Journal of Korean Institute of Intelligent Systems*, vol. 23, no. 4, pp. 332-340, Aug. 2013. <http://dx.doi.org/10.5391/JKIIS.2013.23.4.332>
- [20] Q. N. Van, H. M. Eum, J. Lee, and C. H. Hyun, "Vision sensor-based driving algorithm for indoor automatic guided vehicle," *International Journal of Fuzzy Logic and Intelligent Systems*, vol. 13, no. 2, pp. 140-146, Jun. 2013. <http://dx.doi.org/10.5391/IJFIS.2013.13.2.140>
- [21] T. Edlinger and E. Von Puttkamer, "Exploration of an indoor-environment by an autonomous mobile robot," in *Proceedings of the IEEE/RSJ/GI International Conference on Intelligent Robots and Systems and Advanced Robotic Systems and the Real World*, Munich, Germany, September 12-16, 1994, pp. 1278-1284. <http://dx.doi.org/10.1109/IROS.1994.407463>
- [22] R. Talluri and J. K. Aggarwal, "Position estimation techniques for an autonomous mobile robot: a review," in *Handbook of pattern recognition and computer vision*, C. H. Chen, L. F. Pau, and P. S. P. Wang, Eds. River Edge, NJ: World Scientific Publishing, 1993, pp. 769-801
- [23] C. Durieu, H. Clergeot, and F. Monteil, "Localization of a mobile robot with beacons taking erroneous data into account," in *Proceedings of the IEEE International Conference on Robotics and Automation*, Scottsdale, AZ, May 14-19, 1989, pp. 1062-1068. <http://dx.doi.org/10.1109/ROBOT.1989.100122>
- [24] M. Betke and L. Gurvits, "Mobile robot localization using landmarks," *IEEE Transactions on Robotics and Automation*, vol. 13, no. 2, pp. 251-263, Apr. 1997. <http://dx.doi.org/10.1109/70.563647>

Sairah Naveed received her B.E. in electronic engineering from NED University of Engineering and Technology, Karachi, Pakistan in 2009. Currently she is pursuing her M.S. degree in the Department of control and instrumentation engineering,

Chosun University, Korea. During 2010-2011 she worked in the Dept. of Electronic Engineering, Sir Syed University of Engineering and Technology, Pakistan as a laboratory engineer. Since 2012 she is working as a research assistant in Autonomous Robot Control System Laboratory, Chosun University, Gwangju, Korea. Her research interests include mobile robot localization, mapping and modeling of robot simulation system with uncertainty.



Nak Yong Ko received his B.S., M.S., and Ph.D. degrees from the Department of control and instrumentation engineering, Seoul National University, Korea, in the field of robotics. He has been at the Dept. Control, Instrumentation, and Robot Engineering, Chosun University, Korea, since 1992.

During 1996-1997 and 2004-2005, he worked as a visiting research scientist at the Robotics Institute of Carnegie Mellon University. His research interests include autonomous motion of mobile robots and underwater robots (localization, map building, navigation, planning and collision avoidance), and manipulator force/torque control.