Detection of kidnapped robot problem in Monte Carlo localization based on the natural displacement of the robot

Iksan Bukhori and Zool Hilmi Ismail

Abstract

This article proposes a new method to detect the kidnapped robot problem event in Monte Carlo localization. The method is designed in such a manner that it can provide accurate detection across all time instances, whether the robot can still recognize part of the environment or is totally lost after kidnapping. The proposed method uses the sensor reading of the robot to determine if robot's displacement at particular time instance is considered a natural displacement or not. A series of simulations are designed to measure the accuracy of detection and how it compares to other methods. The simulations show that the proposed method outperforms the methods of detection based on the weight of particles.

Introduction

In mobile robotics localization, the kidnapped robot problem is defined as a condition when the robot is instantly moved to other position without being told during the operation of the robot.[1–4] The detection of kidnapped robot is one of the most difficult problems in Monte Carlo localization (MCL).[5] This is due to the nature of particle filter used in MCL itself, where the convergence process of hypotheses (called particles) causes the absence of particles in some areas, leading to a localization failure if the robot is kidnapped to that area.

Kidnapped robot problem does not often happen in practice; however, it is often used to test the ability of algorithm to recover from global localization failures. Furthermore, the mechanical and sensor faults can lead to condition similar to kidnapping, thus the detection of this event can be used as fault detection.[6]

For decades, several approaches are used to detect and solve kidnapped robot problem. Some solutions are based on visual recognition, such as the ones found in Majdik et al.[7] and Andreasson et al.[4] These approaches, however, are limited to the robot with visual-based sensor, such as camera. Some other approaches are more flexible by using the intrinsic parameters of the MCL itself instead of depending on the type of sensor used. Augmented MCL proposed by Thrun et al.[1] and MCL with mixture distributions[1,3,8] are some examples of this category.

In augmented MCL, random particles are injected in each iteration so that the possibility of particles' absence in kidnapping destination area is reduced. These random

Universiti Teknologi Malaysia, Kuala Lumpur, Malaysia

Corresponding author:
Iksan Bukhori, Universiti Teknologi Malaysia, Jalan Bukit Keramat, Gurney Heights Condominium C-5-3, Kuala Lumpur 54000, Malaysia.
Email: bukhoriiksan@gmail.com

particles are drawn from either uniform distribution over pose space or the posterior of the measurement. MCL with mixture proposal distribution combines regular MCL sampling with its dual distribution.

Despite its flexibility, the former two methods do not clearly draw a line between detection and recovery of kidnapping. This creates a problem when the concern is not only in the re-localization but also the needs to know when the kidnapping really happens, such as in fault detection.

Other solutions that also depend on intrinsic parameters of MCL can be found in Zhang et al.[5] and Yi and Choi.[9] Zhang et al.[5] use maximum weight of current particle set as the parameter to detect the kidnapping event. Yi. and Choi[9] use similar parameter; but instead of purely using current weight they use the *entropy* of the information which can be extracted from the weight. These two approaches address the detection and recovery separately, as what we prefer.

The rest of the article is organized as follows. In the section "Bayes filter and Monte Carlo localization" Bayes filter and MCL are briefly explained. The section "Definition of the terms in kidnapping" explains the definition of some terms used throughout the article. The section on "Map and measurement model" explains the characteristics of the map used in the article and the measurement model employed in all simulations. The section on "Review on existing methods" reviews the last two methods for kidnapping detection. The method we propose is then delivered in detail in the section "Proposed method." The section on "Simulation results" delivers the simulations result of the comparison between the proposed method and the Maximum Current Weight (MCW). The final section discusses the conclusion and the possible improvement to the method.

Bayes filter and MCL

In mobile robot localization practice, it is almost impossible for a robot to know exactly its coordinates and heading (collectively known as *pose*) in the given map. Rather, the robot should infer the data from environment. The obtained state is then called *belief*. The belief of the robot is defined as

$$\text{bel}(S_t) = p(s_t \mid z_{1:t}, u_{1:t}) \qquad (1)$$

This posterior is the probability distribution over the state $s_t = \langle x_t, y_t, \theta_t \rangle$, at time t, given all past measurements $z_{1:t} = \{z_1, z_2, \ldots, z_t\}$ and all past controls $u_{1:t} = \{u_1, u_2, \ldots, u_t\}$. Sometimes it is also useful to consider the belief *before* taking the current measurement, that is

$$\overline{\text{bel}}(s_t) = p(s_t \mid z_{1:t-1}, u_{1:t}) \qquad (2)$$

MCL is a Bayes-based localization algorithm. It provides a powerful tool to calculate posterior $\text{bel}(*)$, given measurement and control data,[1,10] Bayes filter is based on

Table 1. The Monte Carlo localization algorithm.

1.   **MCL Algorithm** (S_{t-1}, u_t, z_t, m)
2. $S_t = \bar{S}_t = \emptyset$
3.   **for** $n = 1$ to N **do**
4.   generate $s_t^{[n]} \sim p\left(s_t \mid s_{t-1}^{[n]}, \ u_{1:t}, \ m\right)$
5.   calculate weight $\omega_t^{[n]} = p\left(z_t \mid s_t^{[n]}, \ m\right)$
6. $\bar{S}_t = \quad \bar{S}_t + \ \left\langle s_t^{[n]}, \omega_t^{[n]} \right\rangle$
7.   **end for**
8.   normalize ω_t
9.   **for** $n = 1$ to N **do**
10.   draw $s_t^{[n]}$ with probability $\propto \omega_t^{[n]}$
11.   Add $s_t^{[n]}$ to S_t
12.   **end for**
13.   **return** S_t

Markov world assumption, that is, past and future data are independent if one knows the current state s_t.[1] By implementing Bayes rule and this Markov world assumption, the belief posterior can be defined as

$$\text{bel}(s_t) = \eta p(z_t \mid s_t) \ \overline{\text{bel}}(s_t) \qquad (3)$$

The term $p(s_t \mid s_{t-1}, u_{1:t})$ is defined as the *prediction* or *motion model*, since it reflects the state transition due to robot motion. The probability $p(z_t \mid s_t)$ itself is called *correction* or *sensor model*, since it incorporates sensor reading to update robot state. η is the normalization constant ensuring the final result to be normalized to one.

Bayes filter gives freedom to the choices of representation for the posterior. MCL represents the posterior $\text{bel}(s_t)$ by a set S_t of N weighted samples distributed according to the posterior.[1,9] The density of the samples proportionally represents the likelihood of the robot's pose being there

$$S_t = \langle s_t^{[n]}, \omega_t^{[n]} \rangle; \quad n = 1, 2, \ldots, N \qquad (4)$$

Each particle $s_t^{[n]}$ represents the hypothesis of the robot's pose at time t. The ω_t^n is the nonnegative number called weight of particle. It indicates how good particle $s_t^{[n]}$ in representing the robot's pose.

The basic MCL algorithm, as summarized by Thrun et al.[1] and Zhang et al.,[5] is depicted in Table 1. It accepts previous state S_{t-1}, past controls u_t, past measurements z_t, and map information m.

Definition of the terms in kidnapping

Before going further, we have to determine the definition of kidnapping detection and criteria of successful detection. We define kidnapping detection as the process detecting the time t, where $1 \leq t \leq T$, when kidnapping happens. It does not detect the place from where the robot has been kidnapped or where it is kidnapped to.

Kidnapping point is defined as the time instance t when the robot is kidnapped. We also define the criteria of successful detection as follows.

1. The detection should occur only once, since we consider single kidnapping event.
2. The time of detected kidnapping should be the same as the real kidnapping.

These two criteria are used to test the accuracy of the proposed method. Lastly, the recovery/relocalization process is defined as a method to localize the robot after the kidnapping event. We use recovery and relocalization interchangeably in this article.

Map and measurement model

Landmark-based map is used in this article. In this work, all landmarks have no feature distinguishing one from another, similar to corridor-based map with walls surrounding the robot, acting as the landmarks. That is, the robot can measure the distance and heading to the landmarks, without having any knowledge of the landmark involved. If we denote the range by r and bearing by ϕ, the sensor reading of the landmark at any time instance t can be defined as

$$f(z_t) = \{f_t^1, f_t^2, \ldots\} = \left\{ \begin{pmatrix} r_t^1 \\ \phi_t^1 \end{pmatrix}, \begin{pmatrix} r_t^2 \\ \phi_t^2 \end{pmatrix}, \ldots \right\} \quad (5)$$

With $r_{min} < r_t \leq r_{max}$ and $\phi_{min} < \phi_t \leq \phi_{max}$

Assuming all sensor readings are independent of each other, the probability of detecting particular features at any time instance t can be expressed as

$$p(f(z_t) \mid x_t, m) = \prod_i p(r_t^i, \phi_t^i \mid x_t, m) \quad (6)$$

The measurements of the landmarks can then be modeled by simple geometry as in equation (7)

$$\begin{pmatrix} r_t^i \\ \phi_t^i \end{pmatrix} = \begin{pmatrix} \sqrt{(m_{j,x} - x)^2 + (m_{j,y} - y)^2} \\ \text{atan2}(m_{j,x} - x, m_{j,y} - y) - \theta \end{pmatrix} + \begin{pmatrix} \varepsilon_{\sigma_r^2} \\ \varepsilon_{\sigma_\phi^2} \end{pmatrix} \quad (7)$$

Here $\varepsilon_{\sigma_r^2}$ and $\varepsilon_{\sigma_\phi^2}$ denote the zero-mean Gaussian error with variances σ_r^2 and σ_ϕ^2, respectively.

Review on existing methods

In this section, we will discuss two existing methods of kidnapping detection. The first one is the entropy method.[9] This approach utilizes the measure of information contained in the particle set. This information measure is called entropy and can be expressed as

$$H(t) = -\sum_{n=1}^{N} \omega_t^n \log \omega_t^n \quad (8)$$
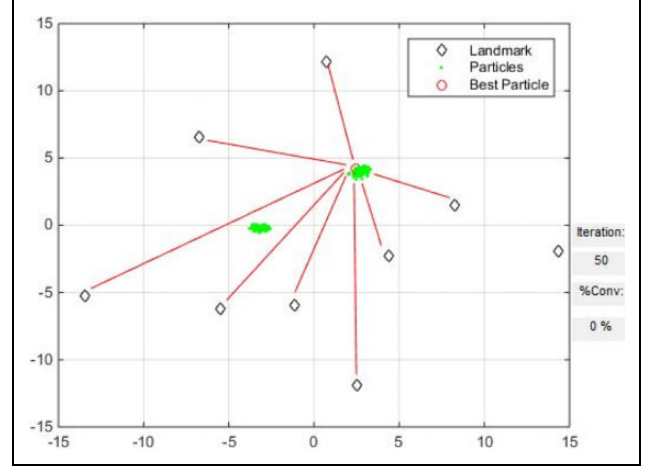

Figure 1. An example of kidnapping instance. The red straight line describes the sensor reading of the best particle.

The kidnapping event detection is defined as

$$\text{kidnapped}_t = \left\{ \begin{array}{ll} 1 & H(t) > \pi \\ 0 & \text{Otherwise} \end{array} \right\} \quad (9)$$

The second approach is included in Zhang et al.,[5] which uses maximum current weight as the trigger for kidnapping event detection. This method can be written as

$$\text{kidnapped}_t = \left\{ \begin{array}{ll} 1 & \omega_t^{max} > \gamma \\ 0 & \text{Otherwise} \end{array} \right\} \quad (10)$$

These two methods are derived from similar parameter, which is the current set of particles. In the environment such as corridor map or our landmark map, particles do not have to be near the true pose of the robot to obtain similar reading with robot's sensor. Any particle can give similar information about its surrounding, provided that the landmarks' positions are similar to the ones seen by the robot.

This problem will in turn reduce the ability to accurately detect the kidnapping event. One example of the problem is depicted in Figure 1.

In this example, the robot is being equipped with a range sensor of $r_{max} = 7$. At $t = 50$, the robot is kidnapped to $(-30, 30)$, well beyond the maximum range of the sensor. At this moment, the reading of robot's sensor will be maximum, that is, $r = r_{max}$. The same reading could be obtained by the best particle (red circle) since there are no landmark within its range also. This causes the parameter ω_t^{max} to jump higher, thus nulling the kidnapping detection as given in equation (10).

On the other hand, many particles are already converged around the best particle, but some particles are converging on different clusters. The difference in the weight between these particle sets will then reduce the entropy, thus nulling according to equation (9). In addition, kidnapping to such an area where the robot could never read any landmark will
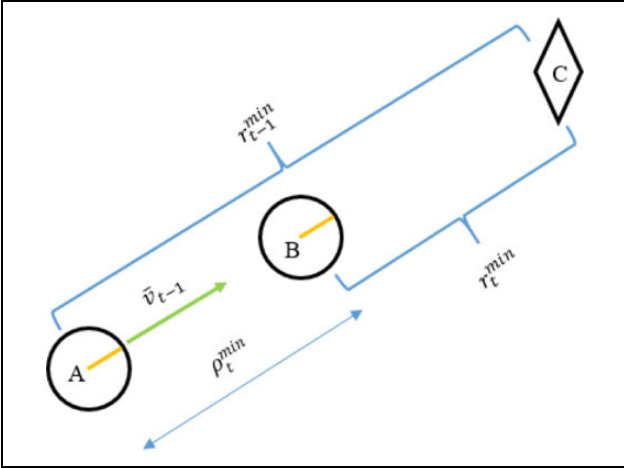
Figure 2. An example of robot's movement to achieve furthest possible distance in between two consecutive time intervals. Here the robot is aligned to the landmark C such that there is no need to do rotational motion, thus maximizing translational displacement.

Figure 3. One instance of the map and the desired trajectory of the robot.

keep the entropy high because the reading of the robot stays the same while the reading of the particles does not change much. This may give multiple kidnapping detection, while in reality it happens only once.

Using particle set as the parameter for the detection is too unpredictable, especially in featureless map. Therefore, a novel detection strategy is needed.

Proposed method

In this article, we propose a new approach to detect kidnapping event based on robot's sensor reading. In this method, we define a parameter

$$\rho_t^{\min} = |\operatorname{argmin}(r_t) - \operatorname{argmin}(r_{t-1})| \qquad (11)$$

That is, the change in distance to the nearest landmark detected between two consecutive time instances. The detector for kidnapping event is then expressed as

$$\text{kidnapped}_t = \begin{cases} 1 & \rho_t^{\min} > \alpha \\ 0 & \text{Otherwise} \end{cases} \qquad (12)$$

Threshold α defines the furthest natural displacement of the robot, that is, displacement to which the change can still be considered as due to the natural movement of the robot. The parameter we used in equation (11) can be depicted as in Figure 2. In this example, the robot is moving from A to B. In this case, the furthest possible distance the robot can travel can be achieved when the robot is already aligned with the landmark. From the figure, it can be seen that the maximum possible distance the robot can travel naturally can be defined as

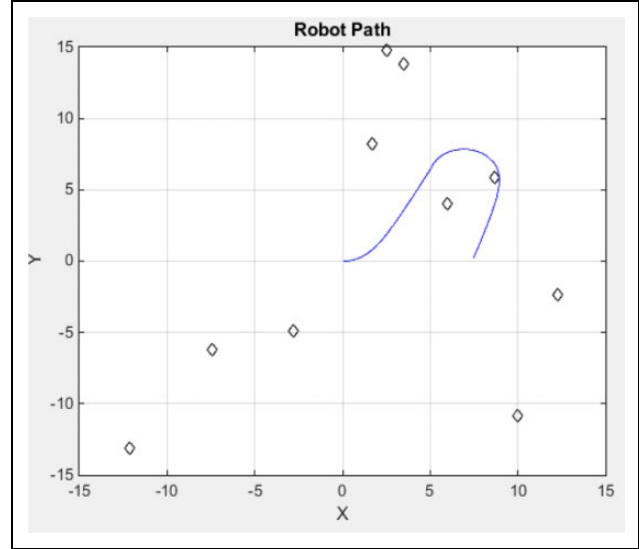$$\alpha = |\bar{v}_{t-1}| \times \delta t + \epsilon \qquad (13)$$

δt is the discretized time step of the robot and ϵ is a free parameter called compensation factor to cover the possibility that the robot is moving further by the noise. The higher the ϵ, the more selective it is to detect kidnapping; however, it will inevitably reduce the sensitivity. Assuming constant velocity, equation (13) can be reduced to

$$\alpha = |\bar{v}| \times \delta t + \epsilon \qquad (14)$$

An example of robot's movement to achieve furthest possible distance in between two consecutive time intervals. Here the robot is aligned to the landmark C such that there is no need to do rotational motion, thus maximizing translational displacement.

Simulation results

A series of simulations are run to test the performance of the proposed method compared to the other two. Each test with a single time instance of kidnapping event is run using 15×15 landmark-based map with 10 randomly placed landmarks. There are 100 time instances of kidnapping, $t_k = \{1, 2, 3, \ldots, 200\}$. For each t_k, the simulation with 100 time steps is run for 100 times.

The number of successful kidnapping detection is then calculated for each t_k and divided by 100 to obtain the percentage of success rate. An example of the map used and the desired trajectory of the robot is depicted in Figure 3.

Because the MCW fails in detection without recovery, other types of tests are devised. In these tests, the recovery strategy is implemented at the time step right after the kidnapping point ($t_{\text{recovery}} = t_k + 1$) regardless of the successfulness of the detection of either method. The recovery strategy employed is the most basic one, which is the

Table 2. The Monte Carlo localization (MCL) simulation setup.

Number of particles	α	γ	π
1000	1.01	0.0013	2.99
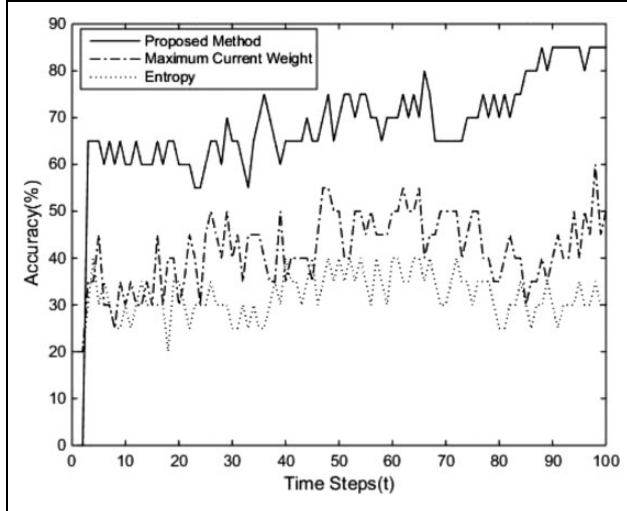
Figure 4. Detection accuracy for in-map kidnapping scenario.

particles' reinitialization method. This recovery process is executed by replacing current set of particles by the randomly distributed particles drawn from uniform distribution over the pose space inside the map, the same as particles' initialization at the very beginning of localization process. The normal trajectory of the robot without kidnapping is as depicted in Figure 3. There are two tests to run, the in-map kidnapping detection test and the out-of-map kidnapping detection test.

Table 2 shows the experiment setup applied for all simulations. All simulations are run on Windows® [version Windows 10 Education] machine 64-bit with 4 GB RAM.
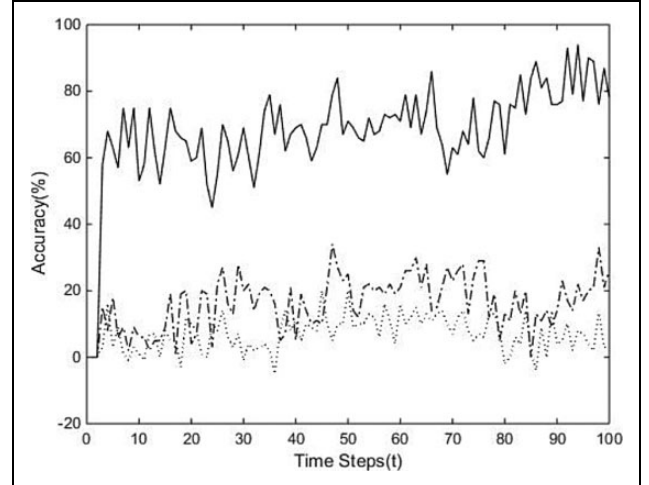
In-map kidnapping detection test

For the first test, the robot is kidnapped to a place in the map that is within the cover of relocalization process. It should be noted here that in this scenario no exclusion problem occurs, that is, the robot can detect all the landmarks in the map. The kidnapping event is expressed as follows

$$\begin{bmatrix} x_{t_k} \\ y_{t_k} \\ \theta_{t_k} \end{bmatrix} = \begin{bmatrix} -8 \\ 10 \\ 0.1\pi \end{bmatrix} \qquad (15)$$

The result of the simulation is depicted in Figure 4.

Out-of-map kidnapping detection test

This test is to see how the three methods handle the situations when the robot is kidnapped such that no landmark is

Figure 5. Detection accuracy for out-of-map kidnapping scenario.

read (totally lost) and thus the relocalization fails. The kidnapping event is defined as

$$\begin{bmatrix} x_{t_k} \\ y_{t_k} \\ \theta_{t_k} \end{bmatrix} = \begin{bmatrix} -30 \\ 30 \\ 0.1\pi \end{bmatrix} \qquad (16)$$

The results of the test is depicted in Figure 5.

Conclusion and future works

A new method in detecting the kidnapping event in MCL is proposed. Unlike other method, the proposed method uses the sensor reading of the robot directly between two consecutive time frames to determine whether the displacement of the robot is normal or not. A series of simulations are run to test the accuracy of the method across all kidnapping points. It shows that the proposed method outperforms maximum current weight method and entropy method during both in-map kidnapping scenario where the robot may still read some landmark and a total lost scenario where robot could not find any landmark within the range.

The method only relies on the range-based sensor without relying on any aspects of localization. Therefore, in future we plan to investigate the ability of the proposed method under different localization frameworks. Other plan including the application on real robot with TurtleBot as the platform, and investigation on the case of multiple kidnapping problem.

References

1. Thrun S, Burgard W, and Fox D. *Probabilistic robotics*. Cambridge, MA: MIT Press, 2005.
2. Hester T and Stone P. Negative information and line observations for Monte-Carlo localization. In: *Proceedings of IEEE international conference on robotics and automation ICRA 2008*, Pasadena, CA, 19–23 May 2008, pp. 2764–2769. IEEE. DOI: 10.1109/ROBOT.2008.4543629.
3. Thrun S, Fox D, Burgard W, et al. Robust Monte Carlo localization for mobile robots. *Artif Intell* 2001; 128(1–2): 99–141.
4. Andreasson H, Treptow A, and Duckett T. Localization for mobile robots using panoramic vision, local features and particle filter. In: *Proceeding of the 2005 IEEE international conference on robotics and automation (ICRA 05)*, Barcelona, Spain, 18–22 April 2005, pp. 3348–3353. IEEE. DOI: 10.1109/ROBOT.2005.1570627.
5. Zhang L, Zapata R, and Lépinay P, Self-adaptive Monte Carlo localization for mobile robots using range finders. *Robotica* 2012; 30: 229–244.
6. Duan Z, Cai Z, and Min H. Robust dead reckoning system for mobile robots based on particle filter and raw range scan. *Sensors* 2014; 14: 16532–16562.
7. Majdik A, Popa M, Tamas L, et al. New approach in solving the kidnapped robot problem. In: *2010 41st international symposium on robotics (ISR) and 2010 6th German conference on robotics (ROBOTIK)*, Munich, Germany, 7–9 June 2010. Verband der Elektrotechnik, Elektronik Und Informationstechnik (VDE).
8. Thrun S, Fox D, and Burgard W. Monte Carlo localization with mixture proposal distribution. In: *Proceedings of the AAAI national conference on artificial intelligence*, Atlanta, Georgia, USA, 11–15 July 2000, pp. 859–865. AAAI digital library, AAAI.org.
9. Yi C and Choi BU. Detection and recovery for kidnapped-robot problem using measurement entropy. In: Kim TH, et al. (eds), *Grid and distributed computing*, 2011, Berlin, Heidelberg: Springer, pp. 293–299.
10. Roumeliotis SI and Bekey GA. Bayesian estimation and Kalman filtering: a unified framework for mobile robot localization. In: *Proceedings of IEEE international conference on robotics and automation (ICRA '00)*, vol. 3, San Francisco, CA, 24–28 April 2000, pp. 2985–2992. IEEE. DOI: 10.1109/ROBOT.2000.846481.