Proceedings of The 2017 IEEE International
Conference on Real-time Computing and Robotics
July 14-18, 2017, Okinawa, Japan

Detecting and Solving the Kidnapped Robot Problem Using Laser Range Finder and Wifi Signal

Yiploon Seow, Renato Miyagusuku, Atsushi Yamashita and Hajime Asama[1]

***Abstract*— This paper presents an approach to detect and solve the kidnapped robot problem using range data from a laser range finder and wifi signals. Localization based on range finders has high accuracy, but fails to detect the kidnapped robot problem, a situation where a well-localized robot is moved to a random location without itself noticing about it. On the other hand, localization based on wifi signals has very high reliability and can be used to detect the occurrence of the kidnapped robot problem; but lacks accuracy. In our approach, a probability density function is constructed using particles sampled from the wifi signal models using kernel density estimation, then the likelihood of every laser range finder particle with respect to the constructed probability density function is calculated. The mobile robot reset its localization process if these probabilities are too low.**

I. Introduction

Localization is one of the fundamental abilities that any mobile robot should possess. Localization enables mobile robots to identify their own location in an area based on a prebuilt map. This is a requisite before they can perform most tasks, such as office documents delivery, museum tour guidance, etc. In order to perform robust mobile robot localization, many approaches have been developed in the past. The sensors employed play an important role in determining the accuracy of those approaches. Sensors such as Laser Range Finder (LRF) and RGB-D cameras are some of the most popular sensors for indoor localization. Another alternative is to use wireless signals such as Wifi for the localization. However, due to its lower accuracy, this sensor is not so popular.

As the approaches for mobile robot localization are getting mature, more complex problems have started to be addressed such as the Kidnapped Robot Problem (KRP). In this paper, KRP is the main issue that will be addressed. KRP is a situation where a well-localized mobile robot is teleported to an arbitrary location without being told. KRP differs from global mobile robot localization as the mobile robot strongly believes itself to be somewhere else at the time of the kidnapping, as shown in Fig. 1, making it much more difficult to solve. KRP is considered as the worst-case scenario for sensor failure, thus, is often used to test mobile robots recovery ability when localization failure occurs.

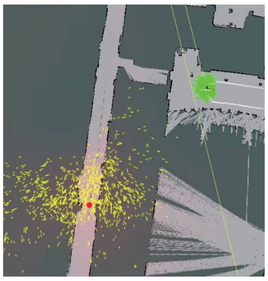


Fig. 1. When KRP occurs, LRF particles (green arrows) fail to relocate, making the robot belief it is still located at its previous location; while Wifi particles (yellow arrows) successfully relocate near to its current location (red dot).

II. Related research

One popular algorithm for localization is the Monte Carlo Localization (MCL) algorithm [1]–[5]. MCL is preferred over other approaches as it has been proved to be accurate and robust in tackling the robot localization problem [1]. For example, in [1], using a LRF and a sonar, MCL was preferred over Kalman-filters because of its ability to represent multi-modal distributions, enabling global localization. And over grid-based Markov localization algorithms for its higher localization accuracies.

MCL is not limited to range sensors, and can be applied to most sensors. For example, MCL has been used with a color camera in [2]; where features extracted from the camera images, such as flags and goals, were used as data input for the MCL. It is important to note that MCL's localization depends on the sensitivity of the sensor it uses. Therefore, sensor selection is essential. Among popular sensors for localization, LRF has the minimal error in environments with visible features [6]. Thus, in this work, LRF is chosen as one of the sensors to be used.

Unfortunately, even with such accurate sensor, MCL fails to localize the mobile robot when the KRP occurs. This is due to one of the characteristics of MCL itself. MCL employs a particle filter to localize the mobile robot, and via this filter, the particles are first distributed evenly on the map and then converge into the area where the robot is likely located. If the mobile robot is then kidnapped to an arbitrary position and that position is not covered by any particles, MCL will not be able to localize the robot [7].

For instance, Lenser et al. [8] proposed an extension to MCL, named Sensor Resetting Localization (SRL). In classic MCL, the localization error is corrected little by little based on each sensor reading and motion model. Hence, huge errors

*This work was funded by Tough Robotics Challenge, ImPACT Program of the Council for Science, Technology and Innovation (Cabinet Office, Government of Japan).

[1] Y. Seow, R. Miyagusuku, A. Yamashita and H. Asama are with the Department of Precision Engineering, Graduate School of Engineering, The University of Tokyo, Japan. {seow, miyagusuku, yamashita, asama} at robot.t.u-tokyo.ac.jp

978-1-5386-2035-9/17/$31.00 © 2017 IEEE

occurring in short times, such as collision or teleportation (KRP) due to unexpected situations can not be corrected. Under such situations, the robot is lost and MCL fails to localize the robot. However, SRL manages to localize the robot even under such situation by replacing some samples of the estimated locations with samples drawn directly from the probability distribution based on the sensor readings. By slowly replacing these samples, the estimated probability distribution slowly shifts to the correct robot location. Unfortunately, SRL faces needless resetting due to wrong sensor readings. To solve this, an expansion of SRL called ER method was proposed by R. Ueda et al. [9]. The difference between ER method and SRL is that ER method constructs a probability distribution based on its previous probability distribution, not based on it newest sensor readings. As a result, the shape of the probability distribution is not changed during the resetting but expanded. The expanded probability distribution would covers the actual robot location and then converge to that location.

Another type of robot localization uses Received Signal Strength (RSS) measurements emitted by wireless devices such as a Wifi modems to perform mobile robot localization [10]–[12]. One of the benefits of using wireless-based indoor localization is that robot location can be easily sampled directly from the sensor's models and current sensor measurements, allowing for fast global localization. Therefore, the robot can always be localized, even when the mobile robot is kidnapped, as long as the mobile robot is in the range of the Wifi coverage [12]. However, the accuracy of wireless-based localization is lower than that of range sensor based localization, the average error of wireless-based localization could reach few meters [6].

In summary, robot localization using MCL algorithm with LRF has high accuracy but fails when KRP occurs, while robot localization that applies MCL algorithm with Wifi signals has low accuracy but is robust to KRP. Thus, it would be best if both approaches can be combined while keeping both benefits: high accuracy and robustness to KRP. Multi-sensor systems including Wifi have been previously proposed; for instance, Ito et al. [13] proposed the usage of both RGB-D camera images and Wifi signals when performing mobile robot localization via MCL, and the proposed approach was called W-RGB-D approach. In this work, RGB-D camera was used to extract features from the images captured, which were used as data input for the MCL algorithm. For initialization, Wifi signals were sampled and a probability density function (PDF) was constructed based on the signal strength. Particles were initialized at area with high probabilities. Then, localization which used the features mentioned above was conducted. In this work, they managed to perform global initialization using Wifi signal as the initiative particles for the MCL algorithm. In their approach, detection of the KRP while the robot stands still was mentioned, but no further details were provided.

In this work, we propose an approach for detecting and solving the KRP using both LRF and Wifi signal measurements while the robot is moving.

III. Approach

In this section, we describe our proposed approach, which is composed of a LRF-based MCL, a Wireless based localization algorithm, and our proposed method to detect and solve the KRP. Our approach's flow is shown in Fig. 2.

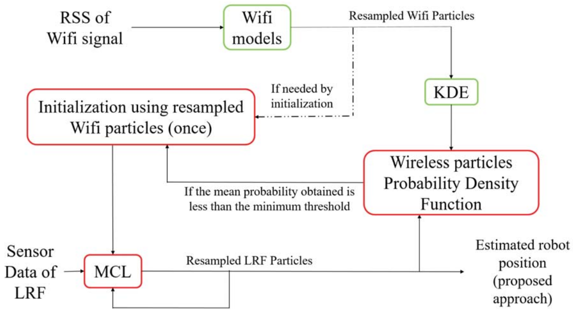

RSS of Wifi signal
Wifi models
Resampled Wifi Particles
If needed by initialization
KDE
Initialization using resampled Wifi particles (once)
Wireless particles Probability Density Function
If the mean probability obtained is less than the minimum threshold
Sensor Data of LRF
MCL
Resampled LRF Particles
Estimated robot position (proposed approach)

Fig. 2. Program flow of the proposed approach.

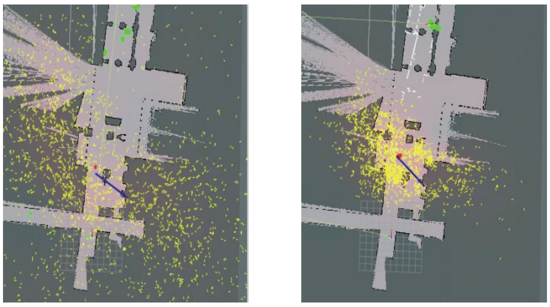
A. Monte Carlo Localization

In our approach, we employed MCL [14] which represents the belief of the position of the mobile robot as a cluster of weighted samples called particles. These particles have four important parameters: x, y, θ, and p, where x, y and θ represent the mobile robot's position in the map, and p is a weight, analogous to a probability/likelihood.

In MCL, the particles are updated via three steps: *prediction*, *correction* and *resampling*. This process is also called particle filtering. In the *prediction* stage, MCL uses a motion model to predict the next coordinates of each particles based on the robot movement, with a certain degree of errors. Next, in the *correction* stage, MCL uses an observation model to compute the weight of those particles based on the newest sensor readings. Lastly, in the *resampling* stage, new particles are drawn from the existing particles and replace them. Particles with higher weights have a higher chance to be reselected, and thus at the end of the *resampling* stage, the new cluster of particles is more focused on those high weight particles, converging to the real mobile robot position.

B. LRF based MCL

Without any prior information, LRF-based MCL uniformly distributes its particles across the map and those particles will be converged into a single high likelihood robot position. This process is necessary for global localization. Although LRF-based MCL is fast and can achieve high accuracy, it is not reliable when it is used to perform global localization in symmetric environments. Usually, in symmetric environments, where multiple possible locations need to be traced for some time, LRF-based localization tends to converge into one of those possible locations in short time, leading to incorrect robot localization [15]. When the global localization fails, the whole process need to be restarted, thus wasting a lot of time. For initialziation, instead of sampling uniformly, it is possible to sample directly from the Wifi posteriors, as done in [13]. Such initialization schemes reduce convergence time and enhance robustness. In our approach, we noted that when a single Wifi measurement was used for bootstrapping, LRF-based MCL sometimes failed to converge. LRF being very precise sensors, require high concentration of samples at initialization to guarantee convergence; which did not always occured. Figure 3a shows an example of samples obtained from a single measurement. To increase samples concentration, we made particles first converge over several Wifi measurements in a MCL-like approach. This yielded higher concentration of particles, as shown in Fig. 3b, lowering the risk of convergence failures. By doing so, after bootstrapping the concentrated particles, LRF-based MCL managed to not only always successfully converge, but also achieve higher accuracies. The disadvantage is that some time was wasted while waiting for the measurements in the bootstrapping stage.



(a) Initialization from single measurement

(b) After 5 measurements

Fig. 3. The yellow arrows are the Wifi particles. In (a), the Wifi particles are widely distributed, but after 5 measurements, the Wifi particles converged and become more concentrated (b).

C. Wireless based localization

In this work, Gaussian Processes (GPs) and Path loss models are used to learn location-signal strength mappings. The exact approach taken in this work is the same presented in [12]. For completeness of ideas, the core components of this approach are presented in this subsection.

Given a training dataset, GPs can learn a continuous function where each point is considered to have normal distribution. For our problem we define our training dataset as $(\mathbf{X}, \mathbf{Z})$ where $\mathbf{X} \in \mathbb{R}^{n\times 2}$ is the location matrix of n input samples $\mathbf{x}_i \in \mathbb{R}^2$; and $\mathbf{Z} \in \mathbb{R}^{n\times m}$ the matrix of received signal strength RSS measurements vector $\mathbf{z}_i \in \mathbb{R}^m$, with $\mathbf{z}_i = [z_{(i,0)}, \cdots, z_{(i,m-1)}]$ being the signal strength information from m different access points in the environment.

The GP approach makes two assumptions. First, each data pair $(\mathbf{x}_i, \mathbf{z}_i)$ is assumed to be drawn from a noisy process $\mathbf{z}_i = f(\mathbf{x}_i) + \epsilon$, where ϵ is the noise generated from a Gaussian distribution with known variance σ_n^2. Second, any two output values are assumed to be correlated by a covariance function based on their input values as $cov(\mathbf{z}_p, \mathbf{z}_q) = k(\mathbf{x}_p, \mathbf{x}_q) + \sigma_n^2 \delta_{pq}$, where $k(\mathbf{x}_p, \mathbf{x}_q)$ is a kernel, σ_n^2 the variance of ϵ and δ_{pq} is one only if $p = q$ and zero otherwise. Given these assumptions, for any finite number of data points, the GP can be considered to have a multivariate Gaussian distribution:

$$\mathbf{z} \sim \mathcal{N}(m(\mathbf{x}),\, k(\mathbf{x}_p, \mathbf{x}_q) + \sigma_n^2), \tag{1}$$

and therefore be fully defined by a mean function $m(\mathbf{x})$ and a kernel function $k(\mathbf{x}_p, \mathbf{x}_q)$.

For wireless based localization, the chosen mean function is a Path loss model (which is a parametric function which is a simplification of the physical phenomena of electromagnetic wave propagation through space) of the form:

$$PL(\mathbf{x}) = k0 - k1 \log(d) + \epsilon_{pl}, \tag{2}$$

with d being the euclidean distance between the position where the RSS measurements were taken and the position of the access point $|\mathbf{x} - (apx,\ \ , apy)|$, k a positive constant and ϵ_{pl} a Gaussian noise with variance σ_{pl}^2.

While the kernel is chosen to be a squared exponential kernel:

$$k_{se}(\mathbf{x}_p, \mathbf{x}_q) = \sigma_{se}^2 \exp\left(-\frac{|\mathbf{x}_p - \mathbf{x}_q|^2}{l_{se}^2}\right), \tag{3}$$

with free parameters σ_{se}^2 (known as the signal variance), and l_{se} (known as the length-scale). These free parameters are often referred to as hyper-parameters ($\theta_{se} = [\sigma_{se}^2,\ l_{se}]$), and are learned from the training data. All parameters in eq. (2) and (3) are learned from the training dataset.

For making new predictions $\mathbf{z}_*$ at any arbitrary location $\mathbf{x}_*$, we can condition the predicted signal strength vector on its location and the training dataset, obtaining

$$p(\mathbf{z}_* | \mathbf{x}_*, \mathbf{X}, \mathbf{Z}) \sim \mathcal{N}(\mathbb{E}[\mathbf{z}_*], \mathrm{var}(\mathbf{z}_*)) \tag{4}$$

where,

$$\begin{aligned} \mathbb{E}[\mathbf{z}_*] &= PL(\mathbf{x}_*) + \mathbf{k}_*^T(\mathbf{K} + \sigma_n^2 \mathbf{I}_n)^{-1}(\mathbf{Z} - PL(\mathbf{X})), && (5) \\ \mathrm{var}(\mathbf{z}_*) &= k_{**} - \mathbf{k}_*^T(\mathbf{K} + \sigma_n^2 \mathbf{I}_n)^{-1}\mathbf{k}_*, && (6) \end{aligned}$$

with $\mathbf{K} = cov(\mathbf{X}, \mathbf{X})$ being the covariance matrix between all training points $\mathbf{X}$, usually called Gram Matrix; $\mathbf{k}_* = cov(\mathbf{X}, \mathbf{x}_*)$ the covariance vector that relates the training points $\mathbf{X}$ and the arbitrary location $\mathbf{x}_*$; $k_{**} = cov(\mathbf{x}_*, \mathbf{x}_*)$ the variance of the location.

Importantly for our approach, this can be efficiently computed as both the inversion of the Gram matrix, and its product with the difference between the training outputs ($\mathbf{Z}$) and the predictions of the inputs by the Path loss model ($PL(\mathbf{X})$), is fixed for a set of learned kernel hyper-parameters. Therefore can be cache, making mean predictions scale in $O(n)$ and variance predictions in $O(n^2)$ with respect to the number of training data points. In practice, for datasets in the order of several hundreds of points and few hundreds of access points, predictions can be computed extremely fast; and even learning which scales in $O(n^3)$ is performed in just a few seconds. However, if larger training datasets are required to be used scalability can be greatly improved to the order of 10^7 points using distributed Gaussian Processes [16].

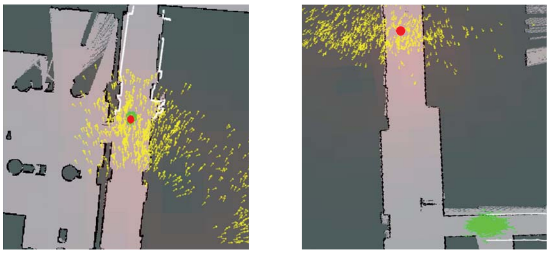
Mean and variance prediction computations are extremely important, as for new signal strength measurements, inferences are computed solely from these two metrics as:

$$p(\mathbf{x}_* | z_j, \mathbf{X}, \mathbf{Z}) \ = \Phi(z_j - \mathbb{E}[\mathbf{z}_*], \mathrm{var}(\mathbf{z}_*)). \tag{7}$$

D. *Kidnapped robot problem*

We can detect the occurrence of the KRP by identifying the distance between the Wifi particles cluster and LRF particles cluster. However, calculating the average distance between both clusters, which are two different probability distributions, is an oversimplification. In this section, we will discuss the method implemented in our approach, which used to detect the KRP.

1) Detecting KRP: In our approach, we have two set of particles operating while the localization is running: those from using a LRF and those sampled directly from the Wifi posterior particles. MCL with LRF plays the role of localize the mobile robot while the Wifi particles play the role of detecting the KRP. In order to detect the occurrence of KRP, both LRF and Wifi particles need to be compared and evaluated. We assume that the further away both LRF and Wifi particles are to each other, the likelihood that KRP occurred is higher. In order to evaluate this, we built a PDF based on the Wifi sensor posterior and evaluate the likelihood of the LRF particles to belong to this distribution. As mentioned before, the wireless-based localization is robust to KRP, thus, in our work, the Wifi particles are treated as the reference to indicate where the robot is, as shown in Fig. 4. In this work, we argue that the likelihood of LRF particles to belong to the Wifi PDF is inversely proportional to the occurrence of KRP. Kernel Density Estimation (KDE) is chosen as the method to construct the PDF.


(a) No KRP

(b) KRP

Fig. 4. When no KRP occurs (a) both LRF particles (green arrows) and Wifi particles (yellow arrows) are close to the ground truth (red dot). When KRP occurs (b), LRF particles remain at their previous position, far from the ground truth; while Wifi particles update their positions to remain close to the ground truth.

2) Kernel density estimation: KDE is a method to construct PDF via a non-parametric way, which means the value of parameters such as mean and variance of the data are not necessary needed in constructing the PDF. Via KDE, each sample from the reference distribution is assigned a normal distribution, then all of the normal distributions from the reference samples are be added up to form the PDF. In this work we define the kernel density estimator $\widehat{p}(\mathbf{x}_*)$ of the probability value $p(\mathbf{x}_*)$ at reference location $\mathbf{x}_*$ as:

$$\widehat{p}(\mathbf{x}_*) = \frac{1}{sh\sqrt{2\pi}} \sum_{\mathbf{x}_i \in \mathbf{X}_s} exp\left(-\frac{(\mathbf{x}_* - \mathbf{x}_i)(\mathbf{x}_* - \mathbf{x}_i)^T}{2h^2}\right), \tag{8}$$

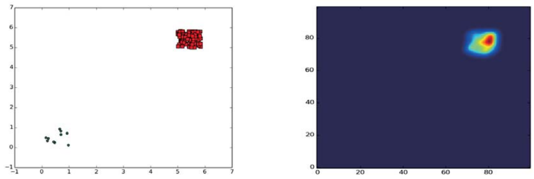
where h is a smoothing parameter, usually referred to as bandwidth; and $\mathbf{X}_s \in \mathbb{R}^{s\times 2}$ is the matrix composed of all s reference location samples.

The value of h influences the estimation strongly. In our approach, the value of h is decided using Scott's Rule:

$$h = s^{\frac{-1}{(d+4)}}, \tag{9}$$

with $d = 2$ being the number of dimensions of our location samples.

Then, the probabilities of the evaluating samples can be calculated based on the reference PDF. A threshold value is selected, which acts as the indicator for the occurrence of the KRP. If the mean probability of the evaluating samples is lower than the threshold, KRP occurs. In this case, the reference sample was referring to the Wifi particles, while the evaluating sample was referring to the LRF particles. Figure 5 shows an example of visualizing the PDF using color, where red represents high probability while blue represent low probability.

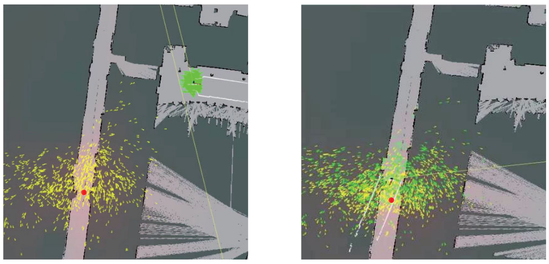


(a) LRF and Wifi particles

(b) Probability of a particle to belong to the Wifi particle set

Fig. 5. Using Wifi particles (red dots) shown in (a) as inputs, KDE is capable of generating the PDF of the particles shown in (b). If LRF particles are then evaluated using this PDF, when KRP has occurred as in the example, the probabilities are expected to be low.

3) Resetting the localization: When probability obtained is lower than the threshold, the robot localization is reset by using Wifi particles as prior information, as a one-time initialization of LRF particles, as shown in Fig. 6. This is to narrow down the particles to a certain wide area where the robot is most probably located at. As mentioned in the previous section, the Wifi particles were allowed to converged before being used as to reinitialize the localization. After the initialization, the sensor data input is switched back to LRF data which enable the particles to further converge into a highly precise location.


(a) KRP occurs

(b) Reset localization

Fig. 6. When KRP occurs (a), the localization will reset (b) and the particles will be reinitialized and shift near to the ground truth (red dot).

IV. Experiment Setup

Experiments were conducted on the ground floor of the Engineering Building No. 2 of The University of Tokyo, Hongo campus by tele-operating a Pioneer 3DX mobile robot equipped with a LRF and a Wifi sensor. First, the mobile robot was navigated around the first floor of the building in order to build the map of that particular floor and collect the RSS of the Wifi signal emitted by various Wifi routers. The layout of that particular floor was scanned using a LRF, which was mounted on the robot, while the RSS measurements of the Wifi signals were collected via a laptop, which was put on top of the robot. Fast-Slam [17] was used to build the map.

Three cases of KRP were emulated in the same floor as shown in Fig. 7. The red circles indicate the location before the robot is kidnapped while the purple circles indicate the location where the robot is being kidnapped to. A trolley was used to move the mobile robot during the kidnapping process. The movement of the robot's wheels during the kidnapping process were kept at minimal so no odometry could be recorded which could tell the mobile robot that kidnapping is happening. At the same time, the LRF was turned off to prevent the robot identify any environment changes.

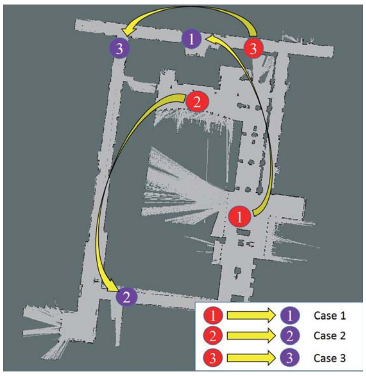


Fig. 7. Paths of robot being kidnapped for each emulated cases.

A. Implementation details

The main software framework was employed on Robot Operating System (ROS). There are three KRP emulations named Case 1, Case 2 and Case 3. All emulations were performed on a Panasonic CF-SX1 with Intel 2.6GHz Quad-Core i5-2540M and 8GB memory laptop, by using time-stamped logs of the acquired data.

In our approach, in addition to the normal LRF-based MCL, for KRP, we added Wifi models and KDE. Wifi models are sampled in order to obtain the likely robot locations from given RSS measurements, which is the Wifi particles initialization process, while the KDE is used to evaluate the occurrence of the KRP. Furthermore, there is also a update process where Wifi particles are sampled and used as the reference samples for the KDE. Both processes running at 0.25Hz. In our approach, the number of particles used in the LRF-based MCL uses 2000 particles at initialization, which are reduced to 700 once they converge.

To evaluate the performance of Wifi model and KDE, the computational time needed for the initialization process and update process of Wifi model, and the evaluation process of KDE were recorded and tabulated. the computation time of three processes are categorized into mean±standard deviation(std) and max as shown in Table I. The Wifi model and the KDE are executed in the rate of 0.25Hz.

TABLE I
Computation Time for Three Processes

	Initialization process [s]		Update process [s]		Evaluation process [s]	
	mean±std	max	mean±std	max	mean±std	max
Case 1	2.46±0.38	3.13	0.16±0.12	0.74	0.25±0.11	0.86
Case 2	2.33±0.29	2.85	0.10±0.07	0.26	0.25±0.12	0.91
Case 3	2.57±0.52	3.45	0.08±0.07	0.27	0.25±0.11	0.59

Although the sensors data was collected at an earlier time, during the playback of the data, the data time-stamps were recreated and adjusted so the robot would belief no time passed between the beginning and end of the kidnapping. Thus, the results would be no different from using real time data obtained on the mobile robot.

In this experiment, for the LRF-based MCL, the maximum number of particles was set to be 2000 and the minimum number of particles was 700.

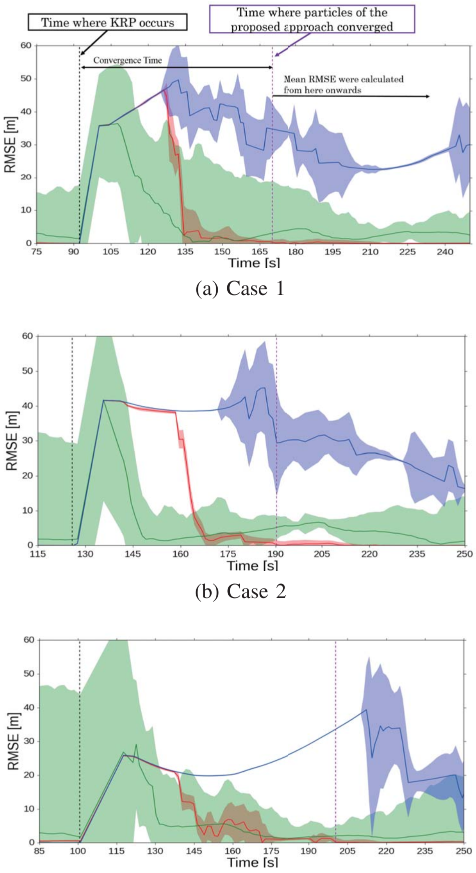
V. Results

For comparison purposes, we run MCL using our approach, as well as LRF-only and Wifi-only. Localization errors for all three sets of data were calculated and tabulated. Figure 8 shows the result of the emulated KRP Case 1, 2 and 3, where the changes of mean Root Mean Squared Error (RMSE) between the robot position and ground truth, and the standard deviation (SD) of the particles, before and after the KRP occurred, are displayed. In addition, the moment where the particles of our approach converged is set when the mean RMSE and SD are less than 0.5m and 2.0m respectively. Table II shows the results for all three emulated KRP cases. Those results are the convergence time of the particles and the mean RMSE of the robot position after the particles converged for three emulated KRP cases.

TABLE II
Experimental results

	Convergence time [s]	Mean RMSE [m]		
		Wifi+LRF	Wifi	LRF
Case 1	77.8	0.18	2.61	33.10
Case 2	64.5	0.11	3.13	23.43
Case 3	99.8	0.12	2.35	30.02

For explanation, we focus on Fig. 8a. From the figure, we can clearly see that before KRP occurs, our approach and the LRF approach have a small RMSE but not for the Wifi approach. When the KRP occurs, all three appraoches



(a) Case 1

(b) Case 2

(c) Case 3

Fig. 8. Graphs constructed using data from (a) Case 1, (b) Case 2 and (c) Case 3. Red lines and shades represent the RMSE and SD of our proposed approach (Wifi+LRF), blue lines and shades represent the RMSE and SD of LRF-only approach, while green lines and shades represent the RMSE and SD of Wifi-only approach.

show increasing RMSE due to unable to localize the mobile robot correctly. However, the Wifi approach managed to recover from the KRP and successfully localize the mobile robot location for the rest of the navigation, although the RMSE is still not good enough. On the other hand, the MCL approach fails to recover from the KRP. The RMSE of the MCL approach keeps on increasing and fluctuating for the rest of the navigation. The combination of the above approaches which is our approach shares the properties of high accuracy and robust to KRP. When KRP occurred, our approach performs similar to the MCL approach, which has an increasing RMSE, but after a certain time, our approach managed to detect the KRP and recover from the KRP by reset the localization using the Wifi particles as prior information. As a result, our approach managed to achieved small RMSE after KRP is detected. The graphs were constructed using the data just before KRP is going to happen so that we can observe and analyze the changes/behaviors of all three approaches due to the KRP.

VI. Conclusions

The objective of this paper was the development of a system that integrates both Wifi and LRF measurements. The system should have high accuracy and be robust to KRP. From the experiment, we can see that our proposed approach manages to detect and recover from the KRP, and achieves high accuracy in localizing the mobile robot. However, the particles convergence time of the proposed approach was longer than expected. Furthermore, if the particles converged and the estimated mobile robots position is always near to the ground truth for the rest of the navigation, this approach failed to adjust and refine the estimated mobile location. Future work will address these issues.

References

[1] F. Dellaert, D. Fox, W. Burgard, and S. Thrun, "Monte carlo localization for mobile robots," in *Robotics and Automation (ICRA), 1999 IEEE International Conference on*, vol. 2, 1999, pp. 1322–1328.

[2] T. Rofer and M. Jungel, "Vision-based fast and reactive monte-carlo localization," in *Robotics and Automation (ICRA), 2003 IEEE International Conference on*, vol. 1, 2003, pp. 856–861.

[3] A. Hornung, S. Oßwald, D. Maier, and M. Bennewitz, "Monte carlo localization for humanoid robot navigation in complex indoor environments," *International Journal of Humanoid Robotics*, vol. 11, no. 02, p. 1441002, 2014.

[4] A. Muzio, L. Aguiar, M. R. Máximo, and S. C. Pinto, "Monte carlo localization with field lines observations for simulated humanoid robotic soccer," in *Robotics Symposium and IV Brazilian Robotics Symposium (LARS/SBR), 2016 XIII Latin American*, 2016, pp. 334–339.

[5] D. Fox, W. Burgard, F. Dellaert, and S. Thrun, "Monte carlo localization: Efficient position estimation for mobile robots," *AAAI/IAAI*, vol. 1999, pp. 343–349, 1999.

[6] J. Biswas and M. Veloso, "Wifi localization and navigation for autonomous indoor mobile robots," in *Robotics and Automation (ICRA), 2010 IEEE International Conference on*, May 2010, pp. 4379–4384.

[7] I. Bukhori, Z. Ismail, and T. Namerikawa, "Detection strategy for kidnapped robot problem in landmark-based map monte carlo localization," in *Robotics and Intelligent Sensors (IRIS), 2015 IEEE International Symposium on*, 2015, pp. 75–80.

[8] S. Lenser and M. Veloso, "Sensor resetting localization for poorly modelled mobile robots," in *Robotics and Automation (ICRA), 2000 IEEE International Conference on*, vol. 2, 2000, pp. 1225–1232.

[9] R. Ueda, T. Arai, K. Sakamoto, T. Kikuchi, and S. Kamiya, "Expansion resetting for recovery from fatal error in monte carlo localization-comparison with sensor resetting methods," in *Intelligent Robots and Systems (IROS), 2004 IEEE/RSJ International Conference on*, vol. 3, 2004, pp. 2481–2486.

[10] A. Schwaighofer, M. Grigoras, V. Tresp, and C. Hoffmann, "GPPS: A Gaussian Process Positioning System for Cellular Networks." in *Advances in Neural Information Processing Systems 16 (NIPS 2003)*, 2003, pp. 579–586.

[11] B. Ferris, D. Haehnel, and D. Fox, "Gaussian processes for signal strength-based location estimation," in *In Proc. of Robotics Science and Systems*, 2006, pp. 1–8.

[12] R. Miyagusuku, A. Yamashita, and H. Asama, "Improving gaussian processes based mapping of wireless signals using path loss models," in *Intelligent Robots and Systems (IROS), 2016 IEEE/RSJ International Conference on*, 2016, pp. 4610–4615.

[13] S. Ito, F. Endres, M. Kuderer, G. Diego Tipaldi, C. Stachniss, and W. Burgard, "W-RGB-D: floor-plan-based indoor global localization using a depth camera and wifi," in *Robotics and Automation (ICRA), 2014 IEEE International Conference on*, 2014, pp. 417–422.

[14] S. Thrun, W. Burgard, and D. Fox, *Probabilistic robotics*. MIT press, 2005.

[15] A. Milstein, J. N. Sánchez, and E. T. Williamson, "Robust global localization using clustered particle filtering," in *AAAI/IAAI*, 2002, pp. 581–586.

[16] M. P. Deisenroth and J. W. Ng, "Distributed gaussian processes." in *ICML*, 2015, pp. 1481–1490.

[17] S. Zaman, W. Slany, and G. Steinbauer, "Ros-based mapping, localization and autonomous navigation using a pioneer 3-dx robot and their relevant issues," in *Electronics, Communications and Photonics Conference (SIECPC), 2011 Saudi International*, 2011, pp. 1–5.