Gaussian Processes with Input-dependent Noise Variance for Wireless Signal Strength-based Localization

Renato Miyagusuku, Atsushi Yamashita and Hajime Asama[1]

***Abstract*— Gaussian Processes have been previously used to model wireless signals strength for its use as sensory input for robot localization. The standard Gaussian Process formulation assumes that the outputs are corrupted by identically independently distributed Gaussian noise. Even though, in general, wireless signals strength do not have homogeneous noise variance. If enough data samples are collected, the noise variance in office-like environments is usually low. In such cases the noise assumption holds. Previous work has demonstrated the viability of wireless signal strength-based localization in such office-like environments. We intend to extend the applicability of these models to perform robot localization in search and rescue scenarios. In such environments, we expect wireless signals strength measurements to be corrupted with high heteroscedastic noise variance. To extend the applicability of previous approaches to these scenarios, we relax the assumption regarding output noise, by considering that the noise variance depends on the inputs. In this work, we describe how this can be done for the specific case of modeling wireless signal strength. Our results show how relaxing this assumption helps improve localization using a synthetic data set generated by artificially increasing noise variance of real data taken from tests performed on a standard office-like environment.**

I. Introduction and motivation

Robot localization or position estimation is the problem of determining a robot's pose relative to a given map of the environment - in this work the pose is understood as the position in a $x - y$ Cartesian coordinate system and the robot's heading direction. Robot localization has been labeled as "the most fundamental problem to providing a mobile robot with autonomous capabilities" [1], as robot's knowledge of its pose is essential for most non-trivial tasks.

The use of wireless signals for robot localization in indoor, GPS-denied, locations has gained popularity in recent years [7]. Our focus in this work is on Gaussian Processes (GPs)[15], which are a type of fingerprinting technique. Fingerprinting refers to the technique of first obtaining samples of the measurements in known locations in the environment (training points), to then predict the location of new values based on new measurements. This is done by matching new measurements to the closest training points or to models based on the training data.

The standard GP formulation makes two assumptions: first, outputs are assumed to be corrupted by i.i.d. Gaussian noise; second, the covariance of the outputs can be modeled by a kernel function dependent on the inputs. Given training data, the system is reformulated as a multivariate Gaussian distribution, from which the mean and variance for new measurements can be estimated. It is important to notice that this formulation does not take into account the variance of measurements at each training point. Mainly because when modeling sensing for most applications, if enough training samples can be obtained, the noise at each training point is often reduced to signal-independent sensor noise. In this case it is a fair assumption that measurements are corrupted by i.i.d. Gaussian noise.

Gaussian Processes with these assumptions have been successfully used for wireless signal strength-based localization [9], [3], [2] in office-like settings. We wish to extend their application for harsher environments, or when training data has to be collected on the fly. In both cases training samples will most likely have high non homogeneous variance - i.e. training samples are composed of heteroscedastic data. In this case the i.i.d. Gaussian noise assumption no longer holds, and a new one need to be made.

Wireless signal strength can be characterized considering the signal's propagation thorough space. The signal propagation phenomena is accurately obtained by using Maxwell's equations, however, these are rarely used because of their complexity. Simple models can be obtained by modeling wireless signals path loss, shadowing and multipath effects. Path loss is caused by the dissipation of the power radiated by the transmitter, and it relates to the distance between the transmitter and receiver. Shadowing effects are the result of power absorption by obstacles, such as walls or furniture. Multipath effects are caused by signals reaching the receiver by several paths, like signals bouncing into walls and reaching the receiver from different directions.

Multipath and shadowing effects not only affect the mean value of the signal strength measured, but also its variance. As both effects are position dependent, the assumption of input-dependent noise variance for modeling signal strength comes naturally. An approach for GPs with input-dependent noise has already been proposed in [5]. We propose its reformulation and addition into current wireless signal strength-based localization algorithms. With this new input-dependent noise assumption, we have obtained localization algorithms that better handle variance estimation with heteroscedastic training samples. The work herein presented exclusively deals with wireless signal strength measurements, however, the approach could be extended to any sensor that can be modeled by a kernel function and has input-dependent

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

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

variance in its training samples.

Input-dependent wireless signal strength variance becomes more important in settings where shadowing and multipath effects are stronger. It is easy to imagine that this is the case in disaster struck environments. When robots are used in such scenarios, it is often desirable to at least have the capability of monitoring them. For this, a wireless network must be deployed in the affected area. Network deployment using simple robots, some times called “robotic routers”, has been the target of several studies, such as [11], [8]. These wireless networks provide the essential infrastructure to guarantee reliable communications between ground stations and robots, which usually determines the success of search and rescue missions [12]. We intend to use such networks for robot wireless signals strength-based localization under the approach just mentioned.

Our results show how our new assumption helps improve localization in a synthetic data generated by artificially increasing noise variance of real data taken on the fly from tests performed on a standard office-like environment.

II. Robot localization problem

As previously mentioned, robot localization or position estimation is the problem of determining a robot's pose relative to a given map of the environment. Monte Carlo Localization (MCL) algorithms are popular localization algorithms used in robotics [13], having as most appealing characteristics their ease of implementation and good performance across a broad range of localization problems. An MCL algorithm is essentially a particle filter, which is an implementation of the Bayes filter, combined with probabilistic models of robot perception and motion [4].

For this work, the pose in the localization problem is defined as $\mathbf{s}$. Given that planar motion is considered, this pose consists of three values: the robot's position in a $x-y$ Cartesian coordinate $(\mathbf{x}_x\ \mathbf{x}_y)$ and the robots heading direction θ - i.e., $\mathbf{s} = [\mathbf{x}_x\ \mathbf{x}_y\ \theta]$. For robot localization, the possible actions the robot can take are considered to be two: (a) the robot can influence its pose through its actuators, and (b) it can gather information about the state through its sensors. Although these interactions usually co-occur, without loss of generality it is assumed that for any time step t the robot first actuates and then senses. Actuating data carries the information of this change of the robot's pose and will be denoted by the vector $\mathbf{a}_t$ - the variable $\mathbf{a}_t$ denotes the change of pose in the time interval $(t-1;t]$ and no action is assumed to occur at time $t=0$. The environment observations provide the information about a momentary state of the environment, i.e, sensors' measurements, and at time t will be denoted by the vector $\mathbf{y}_t$.

In general, the Bayes filter addresses the problem of estimating any state ($\mathbf{s}$ in the robot's localization case) considering the robot state evolution as a partially observable Markov chain (Hidden Markov Model - HMM). Furthermore, it makes the assumption that the environment is a Dynamic Bayesian Network (DBN) often called a Two-Timeslice BN (2TBN) where given $\mathbf{s}_{t-1}$, $\mathbf{s}_t$ becomes independent of all previous states $\mathbf{s}_{0:t-2}$, $\mathbf{a}_{0:t-1}$ and $\mathbf{y}_{0:t-1}$.

Now, the main idea of Bayes filtering is to estimate the pose using a probability density estimation of the state space $\mathbf{s}_t$ conditioned on the time series data $\mathbf{a}_{0:t}, \mathbf{y}_{0:t}$ and previous states $\mathbf{s}_{0:t-1}$. This posterior is called the *belief* of $\mathbf{s}$ - $Bel(\mathbf{s})$. Using Bayes' rule and the Markov assumption introduced by the DBN it can be obtained that:

$$Bel(\mathbf{s}_t) \propto p(\mathbf{y}_t|\mathbf{s}_t) \int p(\mathbf{s}_t|\mathbf{s}_{t-1}, \mathbf{a}_t) Bel(\mathbf{s}_{t-1}) d\mathbf{s}_{t-1} \quad (1)$$

which is the basic equation for all Bayesian filters, including the MCL and the dual MCL (see [13] for full description of the algorithms and proofs).

In order to implement eq. (1), three things are required: (a) a way to represent $Bel(\mathbf{s})$ and a priori distribution for $Bel(\mathbf{s}_0)$, which is usually assumed to be an uniform distribution; (b) the next state transition probability $p(\mathbf{s}_t|\mathbf{s}_{t-1}, \mathbf{a}_t)$; and (c), the perceptual likelihood $p(\mathbf{y}_t|\mathbf{s}_t)$. In MCL and the dual MCL, the belief $Bel(\mathbf{s})$ is represented by a particle filter. Particle filters represent any distribution by a set of s weighted samples also called *particles*, distributed according to that distribution. The next state transition probability $p(\mathbf{s}_t|\mathbf{s}_{t-1}, \mathbf{a}_t)$ is implemented by a robot motion model - which varies depending on the robot used. A complete description of these models can be found at [13], [10]. Finally, the perceptual likelihood $p(\mathbf{y}_t|\mathbf{s}_t)$ depends on the sensors used for the localization. This probability can be understood as the likelihood of observing a measurement $\mathbf{y}_t$ at location p_t. The calculation of this metric considering input-dependent noise is the main contribution of the work herein presented.

III. Modeling signal strength with input-dependent variance

A. Preliminaries

Gaussian Processes are a generalization of normal distributions to functions, describing functions of finite-dimensional random variables. In a nutshell, given some training points, a GP generalizes these points into a continuous function where each point is considered to have normal distribution, hence a mean and a variance. The essence of the method resides is assuming a correlation between values at different points, this correlation is characterized by a covariance function or a kernel.

The standard GP formulation is as follows. Given some training data $(\mathbf{X}, \mathbf{Y})$ where $\mathbf{X} \in \mathbb{R}^{n \times d}$ is the matrix of n input samples $\mathbf{x}_i, \in \mathbb{R}^d$; and $\mathbf{Y} \in \mathbb{R}^{n \times m}$ the matrix of corresponding output samples $\mathbf{y}_i \in \mathbb{R}^m$; two assumptions are made. First, each data pair $(\mathbf{x}_i, \mathbf{y}_i)$ is assumed to be drawn from a process with i.i.d. Gaussian noise:

$$\mathbf{y}_i = f(\mathbf{x}_i) + \epsilon, \quad (2)$$

where ϵ is the noise generated from a Gaussian distribution with known variance σ_n^2.

Second, any two output values, $\mathbf{y}_p$ and $\mathbf{y}_q$, are assumed to be correlated by a covariance function based on their input values $\mathbf{x}_p$ and $\mathbf{x}_q$. In conjunction with the first assumption, we get that:

$$cov(\mathbf{y}_p, \mathbf{y}_q) = k(\mathbf{x}_p, \mathbf{x}_q) + \sigma_n^2 \delta_{pq} \tag{3}$$

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.

Finally, given these assumptions, for any finite number of data points, the GP can be considered to have a multivariate Gaussian distribution, and therefore be fully defined by a mean function $m(\mathbf{x})$ and a covariance function $cov(\mathbf{x}_p, \mathbf{x}_q)$. Without loss of generality, it is common to define $m(\mathbf{x})$ as the zero-function, as the $m(\mathbf{x})$ can be subtracted from training data prior to passing it to the GP. In this case, a GP is fully defined only by the covariance function, and estimation can be for an unknown data point $\mathbf{x}_*$, conditioned on training data $(\mathbf{X},\ \mathbf{Y})$ becomes:

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

where,

$$\mathbb{E}[\mathbf{y}_*] = \mathbf{k}_*^T (\mathbf{K} + \sigma_n^2 \mathbf{I}_n)^{-1} \mathbf{y}, \tag{5}$$

$$\mathrm{var}(\mathbf{y}_*) = k_{**} - \mathbf{k}_*^T (\mathbf{K} + \sigma_n^2 \mathbf{I}_n)^{-1} \mathbf{k}_*, \tag{6}$$

and, $\mathbf{K} = cov(\mathbf{X}, \mathbf{X})$ the $n \times n$ covariance matrix between all training points $\mathbf{X}$, $\mathbf{k}_* = cov(\mathbf{X}, \mathbf{x}_*)$ the covariance vector that relates the training points $\mathbf{X}$ and the test point $\mathbf{x}_*$; $k_{**} = cov(\mathbf{x}_*, \mathbf{x}_*)$ the variance of the test point and I_n the identity matrix of rank n.

This formulation ignores the variance of the training samples $\mathbf{Yvar}$. In order to incorporate this information into the system, Goldberg [5] changed the first assumption by considering that each data pair $(\mathbf{x}_i, \mathbf{y}_i)$ is drawn from a process with known variance that depends on $\mathbf{x}_i$. That is:

$$\mathbf{y}_i = f(\mathbf{x}_i) + v(\mathbf{x}_i), \tag{7}$$

where $\mathbf{Yvar} = \{v(x_0), .., v(x_n)\}$.

With this new assumption, eq. (3) becomes:

$$cov(\mathbf{y}_p, \mathbf{y}_q) = k(\mathbf{x}_p, \mathbf{x}_q) + var(\mathbf{x}_p) \delta_{pq}, \tag{8}$$

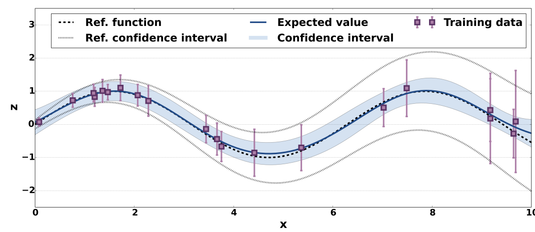
and $\mathbb{E}[\mathbf{y}_*],\ \mathrm{var}(\mathbf{y}_*)$ from 4:

$$\mathbb{E}[\mathbf{y}_*] = \mathbf{k}_*^T (\mathbf{K} + \mathbf{Kv})^{-1} \mathbf{y}, \tag{9}$$

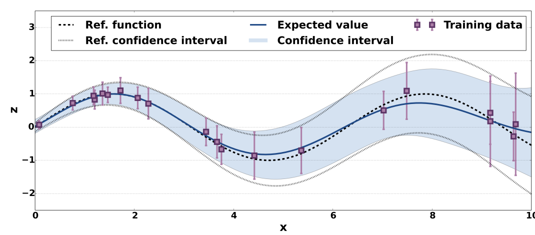
$$\mathrm{var}(\mathbf{y}_*) = k_{**} + v(\mathbf{x}_*) - \mathbf{k}_*^T (\mathbf{K} + \mathbf{Kv})^{-1} \mathbf{k}_*, \tag{10}$$

with $\mathbf{Kv} = \mathrm{diag}(\mathbf{Yvar})$ and $v(\mathbf{x}_*)$ the predicted measurement variance that would be obtained at $\mathbf{x}_*$. The estimation of $v(\mathbf{x}_*)$ now becomes a second regression problem. This new regression problem can be solved by another GP, interpolation of training points (as this time the variance of this function is not required) or any other regression algorithm.

To illustrate the differences between the standard GP formulation and this new approach, we show the predictions generated using a toy function $y = \sin(x)$ with variance $v = (x+1)/15$. For each training point (20 in the example) 5 samples are drawn from the described toy function. Fig. 1 shows a comparison of the predictions using a standard GP, which uses only the mean of the samples, and the proposed approach, which also considers the variance of the samples. From Fig. 1 it can be appreciated that the first GP predicts homogeneous low variances. Contrary, the second GP successfully models the toy-function variance. This is a simple example, however the same properties observed here will be obtained in the next section when generating signal strength maps.



(a) Standard GP



(b) Proposed method

Fig. 1. Prediction generated for a toy function $f = \sin(x)$ with variance $v = (x+1)/15$

B. Problem formulation

We consider the problem of localizing a robot using wireless signal strength measurements as primary sensory input. It is assumed that the robot has wireless capabilities, specifically an 802.11-compliant Wireless Network Interface Controller (WNIC). Standard 802.11 cards have a built-in Received Signal Strength (RSS) indicators, that will be used by the robot to acquire the signals strengths. Furthermore, it is assumed that a wireless network composed of m access points has been deployed, either by robotic routers or some WLAN infrastructure is available, and is static (i.e., the variations in signal readings are due to sensor noise or signal propagation effects, and not by robotic routers moving). Regarding the wireless signals, it is also assumed that the heading direction of the robot does not affect measurements (it is assumed that both the robot and the access points have antennas with fairly homogeneous radiation patterns - e.g., omnidirectional antennas). Therefore the perceptual likelihood $p(\mathbf{y}_t | \mathbf{s}_t)$ is simplified to $p(\mathbf{y}_t | \mathbf{x}_t)$.

Given these considerations, our approach creates models for the RSS measurements using GPs. For using any GP, it is first necessary to obtain training data $(\mathbf{X},\ \mathbf{Y},\ \mathbf{Yvar})$. With $\mathbf{X} \in \mathbb{R}^{n \times 2}$ being the matrix of n input samples $\mathbf{x}_i$ that correspond to the $x - y$ Cartesian coordinates where the samples were taken; $\mathbf{Y} \in \mathbb{R}^{n \times m}$, the matrix created from the sampled mean of RSS measurements taken from m access points at n positions; and $\mathbf{Yvar} \in \mathbb{R}^{n \times m}$ the variance corresponding to each RSS measurements. It is important to notice that the signals originated from each access point are easily distinguishable by their MAC address - which is a unique identifier assigned to every access point and transmitted as part of the IEE802.11 protocol.

The first problem to be considered is the acquisition of $\mathbf{X}$. In previous approaches [9], [3], data was collected manually, therefore it was possible to self-label the positions $\mathbf{x}_i$ where the data was taken. However, in search and rescue scenarios this would not be feasible; therefore, the systems must collect labeled training data on its own. This can be achieved using Gaussian Process Latent Variable Models [6], which has already been used to generate RSS maps from unlabeled data similar to those generated with labeled data[2].

The second problem to be considered is the correct estimation of RSS variance. Data is collected on the fly, and therefore not much time is taken for collecting data samples. Therefore, the number of these samples can vary from zero to the order of tens of measurements. It is specially problematic when no measurements or only one measurement is obtained, as the sampled variance would be zero, misleading the algorithm into believing there is high confidence in the measurement, when it is in fact the opposite. Zero measurements at any given point represent the absence of RSS measurements at the particular time that it was sensed. This absence of RSS measurements can be either by random occurrences, like glitches or corrupted packages that are dumped by the WNIC; or the product of shadowing effects, which provides valuable information about that point. One approach is to simply not consider zero measurements and only work with non-zero ones. However, we prefer applying a prior over the variance estimation, and using the information into our system.

C. Kernels and hyper-parameter optimization

For the GPs problem, a kernel must be selected. For our implementation, we use an squared exponential kernel, also commonly referred as the radial basis function or the Gaussian kernel. It is defined as:

$$k_{se}(\mathbf{x}_p, \mathbf{x}_q) = \sigma_{se}^2 \exp\left(-\frac{|\mathbf{x}_p - \mathbf{x}_q|^2}{l_{se}^2}\right), \quad (11)$$

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.

As the data $\mathbf{Y}$ and $\mathbf{Yvar}$ are generated from m different access points, there are two options for defining the kernel: (a) to use a single kernel that best models the behavior of all m access points or (b) to have m independent kernels, each a best fit for its corresponding access point. In our approach we opted for the first option, as we consider that the wireless signal propagation phenomena is highly environmental dependent and given a common environment for all access points, a single kernel should be able to model all the relationships. In order to find this optimum kernel, we need to find the maximum a posteriori estimation of the parameters θ, which occurs when $p(\theta|\mathbf{X}, \mathbf{Y}, \mathbf{Yvar})$ is maximized. Using Bayes' rule and assuming an uninformative prior distribution $p(\theta|\mathbf{X})$:

$$p(\theta|\mathbf{X}, \mathbf{Y}, \mathbf{Yvar}) = p(\mathbf{Y}, \mathbf{Yvar}|\mathbf{X}, \theta). \quad (12)$$

To solve this optimization, we consider as two separate problems the optimization of kernel parameters for estimating $\mathbf{Y}$ and those for estimating the variance $\mathbf{Yvar}$. This is a more a practical consideration, as we have found no major differences between the joint optimization of parameters by Maximization-Expectation, and that of independently maximizing the parameters.

Having this in consideration, we redefine the problem of finding the maximum a posteriori estimation of $p(\theta|\mathbf{X}, \mathbf{Y}, \mathbf{Yvar})$ as that of minimizing the negative log likelihood (nll_{GP}) of $p(\mathbf{Y}|\mathbf{X}, \theta)$, given by:

$$nll_{GP} = 0.5nm\log(2\pi) + \sum_{i=1}^{m} 0.5\log|\mathbf{K} + \mathbf{Kv}_i| + 0.5\mathbf{y}_i^T(\mathbf{K} + \mathbf{Kv}_i)^{-1}\mathbf{y}_i \quad (13)$$

with $\mathbf{y}_i \in \mathbb{R}^n$ being the vector composed by the means of the RSS samples obtained from access-point$_i$, and $\mathbf{Kv}_i = \mathrm{diag}(\mathbf{yvar}_\mathrm{i})$ for $\mathbf{yvar}_i$ representing the variance estimation of $\mathbf{y}_i$. Finally, optimization can be done by calculating the partial derivatives of nll_{GP} with respect to θ and performing conjugate gradient descend.

D. Variance estimation

To solve the second regression problem, the variance vector $\mathbf{yvar}_i$ is assumed to have been generated by a function dependent on the position $v(\mathbf{X})$ and some small noise ϵ. It is convenient to reformulate this regression problem using the variable change $\mathbf{z} = \log v(\mathbf{X})$. This way we ensure that predicted variances will be always positive and placing a prior with zero mean is equivalent to placing a prior with mean ones, which by adding an offset on $\mathbf{z}$ can become a prior with any desired value. With this change and the assumption initially taken, we can re-state the problem as:

$$\mathbf{z} = v(\mathbf{X}) + \epsilon \quad (14)$$

which, if assuming ϵ as i.i.d. Gaussian noise, is exactly the formulation for standard GP. Therefore, we opted to solve the second regression problem with a standard GP. However, as we stated before, any regression algorithm can be used.

Therefore using eq. (5) and making the variable exchange, the estimated value of $v(\cdot)$ for a new input can be calculated as:

$$\mathbf{yvar}_* = \exp\left(\mathbf{kz}_*^T(\mathbf{Kz} + \sigma_n^2\mathbf{I}_n)^{-1}\log(\mathbf{Yvar})\right), \quad (15)$$

with $\mathbf{Kz} = cov(\mathbf{X}, \mathbf{X})$ and $\mathbf{kz}_* = cov(\mathbf{X}, \mathbf{x}_*)$ being generated from a covariance function different than that the one used to solve the GP with input-dependent noise.

E. Calculating posterior probabilities

As stated at the beginning of the section, our goal is to compute the likelihood of the set of new RSS measurements $\mathbf{y}_{new}$ to have been generated from an specific location $\mathbf{x}_*$. We do this by individually calculating the conditional probability for each access point that had non-zero values. As the measurements are assumed to have a Gaussian probability distribution, we only require the first and second-order statistics (i.e., the predicted mean $\mathbb{E}[\mathbf{y}_*]$ and predicted variance $\mathrm{var}(\mathbf{y}_*)$), which are calculated from eq. (9) and (10).

We do this only for non-zero elements as it was not uncommon during testing that at some points, signals were not found even though during the training phase they were measured. Signals can be absent by a random occurrence or simply because the access points was not active at the time.

Finally, once posterior probabilities for each access points are found, we fuse the information using its geometric average instead of using its product, as this would lead to overconfident estimates [3].

Therefore, we define:

$$p(\mathbf{y}_{new}|\mathbf{x}_*) = \left(\prod_k p(y_{\{new,k\}}|\mathbf{x}_*)\right)^{1/|k|} \qquad (16)$$

for $k \in \{1, ..., m\}$, with $y_{*,k} \neq 0$, as the output of the GP.

IV. Experimental Results

We tested and compared our approach with previous incarnations of GP-based localization by collecting data from a building at the University of Tokyo, Fig. 2 shows a simplified blueprint of the building. The building is approximately 70m long by 50m wide.

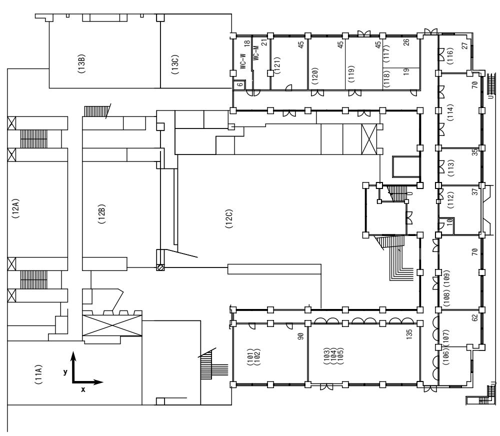


Fig. 2. Blueprint of Engineering Building 2 at the University of Tokyo

The dataset was constructed by collecting RSS samples at 166 different positions. RSS samples were taken using a Panasonic SX1 laptop, which has a sensing range of 0 to -90 dBm, any value under this will not be sensed at all. Whenever an access point was not sensed, a value of -90 dBm was assigned by default. RSS measurements were scaled by a factor of 15 and offset them by 20/3, as to obtain data in the range of 20/3 to 0 - the scaling and offset were done so the prior set on $\mathbf{z}$ - the log variance estimator, would be a zero mean function. Over 190 access points were detected in the area. However, many of these access points provided almost identical information as it is common for routers to have several antennas, each with its own WNIC and MACADDRESS. After eliminating repeating access points and those that had few non-zero measurements, we retained 21 access points which are used for all the testings. In order to test the approach, a 5-fold cross-validation scheme was used.

A. Comparison of Posterior probabilities

The perceptual likelihood is the core of MCL and dual MCL, as it provides information about the environment through sensing. Figure 3 shows a comparison between the posterior probability distributions generated by the standard GP formulation and by our approach. A black X marks the true pose. Red areas indicate high probability while blue indicate low. A good perceptual likelihood would generate high probability areas in the vicinity of the true position, and as low as possible in areas far from it.

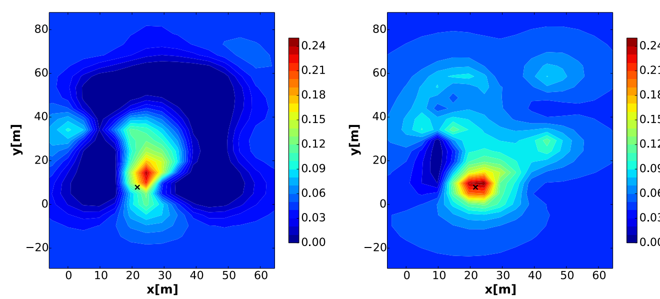


Fig. 3. Posterior probability distributions for (left) the standard GP formulation, (right) our approach

As it can be observed from Fig. 3 both GPs generate adequate posterior probabilities, our approach generating slightly better outputs as the high probability area is closer to the true position, than in the other approach.

B. Artificial addition of noise

We will now assess the performance of both approaches with synthetic data sets generated by artificially increasing noise variance on the real data set. To generate the synthetic data, first a vector $\mathbf{v}_u$ composed of values sampled uniformly at random between 0 and a maximum noise variance is created. For all non-zero values of $\mathbf{Y}$ the new data will be sampled from a normal distribution with mean $\mathbf{Y}$ and variance $\mathbf{Yvar}+\mathbf{v}_u$ - the new data is generated only for non-zero values, cause it is not feasible for RSS signals bellow the sensing threshold to be sensed, no matter how much noise is added. Lastly, the new vector $\mathbf{Yvar}$ is computed based on the new data $\mathbf{Y}$. It is our assumption that this increase in variance will happen in real disaster scenarios.

Figure 4 shows the posterior probability conditioned on the same test point that Fig. 3 but with synthetic data generated with maximum variance of 0.4.

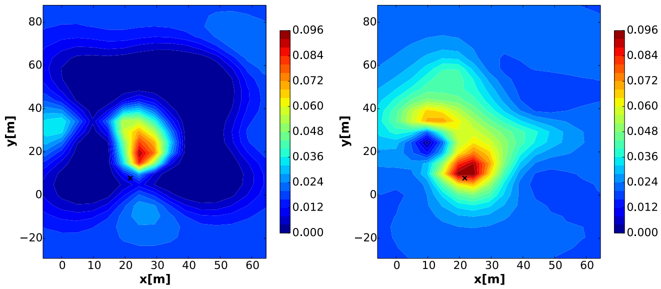
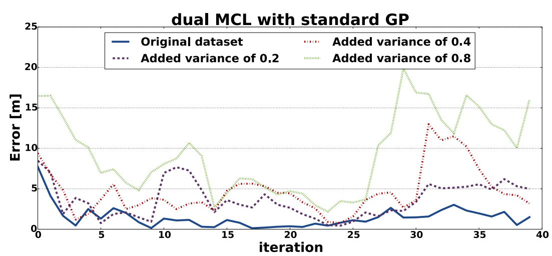


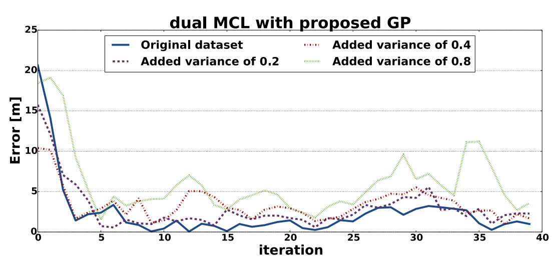
Fig. 4. Posterior probability distribution using synthetic data generated with maximum variance of 0.4 for (left) the standard GP formulation, (right) our approach

It can be seen that the values for the posterior probabilities become quite lower for both approaches when noise is added (drop from 0.24 to 0.096). On the one hand, for the standard GP formulation, it can be noted that high probability area moves further from the true position, which is a very undesirable effect. On the other hand, for our formulation, the system becomes much more uncertain of the true position (the high probability area increases); however, it still remains near the true position.

C. Results with dual MCL



(a) Standard GP formulation



(b) Our approach

Fig. 5. Errors in location estimation using a dual MCL for synthetic data generated with different maximum noise variances

Finally we assess and compare the performance of our approach using a dual MCL algorithm. We selected the dual MCL as it is highly dependent on the likelihood model. A complete description of the dual MCL algorithm can be found at [14]. Figure 5 shows the errors in location estimation for different synthetic data when using the standard GP formulation or our approach as perceptual likelihood. It can be observed from the simulations that the localization accuracy is adversely affected when the noise variance increases, for both approaches. However, the impact is lesser when using our approach. Furthermore, the system is able to handle maximum noise variances of up to 0.4 without inducing much error. Considering that RSS data was scaled by a factor of 15, therefore a noise variance of 0.4 represents a noise of $\pm$ 20 dBm.

V. Conclusions

We have presented an approach for wireless signal strength-based localization that relaxes the assumption of i.i.d Gaussian noise, by considering input-dependent noise. This approach generates more consistent posterior density distribution than the standard Gaussian Process formulation, in environments with high noise variance. Furthermore, when used as the perceptual likelihood of a dual MCL algorithm, it has an accuracy error lower than 5m for almost all the time, even when injected with noises of $\pm$ 20 dBm. Testing involving high noise variance was performed using synthetic data, therefore, it remains as future work its validation in real scenarios. Nonetheless, the results obtained are encouraging, and suggest the possible applicability of our approach in search and rescue scenarios.

References

[1] Ingemar J Cox. Blanche-an experiment in guidance and navigation of an autonomous robot vehicle. *Robotics and Automation, IEEE Transactions on*, 7(2):193–204, 1991.

[2] Brian Ferris, Dieter Fox, and Neil D Lawrence. WiFi-SLAM Using Gaussian Process Latent Variable Models. In *IJCAI*, volume 7, pages 2480–2485, 2007.

[3] Brian Ferris, Dirk Haehnel, and Dieter Fox. Gaussian processes for signal strength-based location estimation. In *In proc. of robotics science and systems*, 2006.

[4] Dieter Fox, Wolfram Burgard, Frank Dellaert, and Sebastian Thrun. Monte carlo localization: Efficient position estimation for mobile robots. *AAAI/IAAI*, 1999:343–349, 1999.

[5] Paul W Goldberg, Christopher KI Williams, and Christopher M Bishop. Regression with input-dependent noise: A Gaussian process treatment. *Advances in neural information processing systems*, 10:493–499, 1997.

[6] Neil D Lawrence. Gaussian process latent variable models for visualisation of high dimensional data. *Advances in neural information processing systems*, 16(3):329–336, 2004.

[7] Hui Liu, Houshang Darabi, Pat Banerjee, and Jing Liu. Survey of wireless indoor positioning techniques and systems. *Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on*, 37(6):1067–1080, 2007.

[8] Renato Miyagusuku, Atsushi Yamashita, and Hajime Asama. Distributed algorithm for robotic network self-deployment in indoor environments using wireless signal strength. In *Proc. 13th Inteligent Conf. on Intelligent Autonomous Systems (IAS-13)*, 2014.

[9] Anton Schwaighofer, Marian Grigoras, Volker Tresp, and Clemens Hoffmann. GPPS: A Gaussian Process Positioning System for Cellular Networks. In *NIPS*, 2003.

[10] Bruno Siciliano and Oussama Khatib. *Springer handbook of robotics*. Springer Science & Business Media, 2008.

[11] Tsuyoshi Suzuki, Ryuji Sugizaki, Kuniaki Kawabata, Yasushi Hada, and Yoshito Tobe. Deployment and management of wireless sensor network using mobile robots for gathering environmental information. In *Distributed Autonomous Robotic Systems 8*, pages 63–72. Springer, 2009.

[12] Satoshi Tadokoro. *Rescue Robotics: DDT Project on Robots and Systems for Urban Search and Rescue*. Springer, 2009.

[13] Sebastian Thrun, Wolfram Burgard, and Dieter Fox. *Probabilistic robotics*. MIT press, 2005.

[14] Sebastian Thrun, Dieter Fox, Wolfram Burgard, et al. Monte carlo localization with mixture proposal distribution. In *AAAI/IAAI*, pages 859–865, 2000.

[15] Christopher KI Williams and Carl Edward Rasmussen. *Gaussian processes for machine learning*. the MIT Press, 2006.