BAYESIAN OPTIMIZATION FOR MONITORING THE DYNAMIC ENVIRONMENT

by

TIANYU GAO

A thesis submitted to the

School of Graduate Studies

Rutgers, The State University of New Jersey

In partial fulfillment of the requirements

For the degree of

Master of Science

Graduate Program in Mechanical and Aerospace Engineering

Written under the direction of

Xiaoli Bai

and approved by

_____

_____

_____

New Brunswick, New Jersey

October, 2020

ABSTRACT OF THE THESIS

Bayesian Optimization for Monitoring the Dynamic Environment

By Tianyu Gao

Thesis Director:

Xiaoli Bai

How to sample the data in an optimization algorithm is important in an environmental monitoring problem. Ensuring the sampling method practical while obtaining useful information as much as possible to reduce time and energy cost during optimization is the key. This thesis focuses on the implementation of Bayesian Optimization (BO) to monitor a time-varying three-dimensional environment. The BO algorithm is based on the Gaussian Processes (GPs) surrogate models which are non-parametric regression methods, and uses the reward function for decision making. An uniquely designed kernal function is used in GPs to learn the underlying pattern of spatial and temporal variations. A series of theoretical but less practical experiments are developed to prove the capability of BO, together with presenting the importance of temporal information. A continuous path planning is designed to replace the waypoint planning for a real path design in the environmetal monitoring. Furthermore, this planning is effective to balance the trade-off between the exploration and the exploitation in the optimization problem.

Acknowledgements

First of all, I would like to thank my M.S. advisor, Dr. Xiaoli Bai, for her valuable supervision through my M.S. study and research. Dr. Bai is a perfect scholar and an advisor. She introduced me into the professional field of research. She has offered me a great number of opportunities and inspiring guidances to shape my skills. Without her support I would hardly be able to complete this research for thesis.

Then I would like to appreciate the attendance of Prof. Qingze Zou and Dr. Laurent Burlion to the committee, offering their constructive comments for the improvement of my research.

Special acknowledgement goes to my group members Dr. Gaurav Misra and Dr. Hao Peng for their professional guidance and encouragement in my study and research. I also would like to thank my friends and school mates. Their support and countless suggestestions help me finally complete my study.

Dedication

I dedicate my dissertation to my parents, Enqing Gao and Ying Li.

Table of Contents

Chapter 1

Introduction

Environmental monitoring is widely used for the surveillance of the distribution of a certain substance, such as the ozone concentration in the climate change and the light intensity in a particular room.[1] Furthermore, in a more severe situation such as the leakage of pollutant from some industrial facilities, the comprehensive and accurate monitoring of the space of incident is required more urgently to provide the information for the related governmental regulations.

Setting up a static network of sensors is one of the solutions to maintain a stable and continuous monitoring in a few scenarios mentioned above. However, besides the expense of the preset network, it is usually not possible to deploy an extensive sensor network when the timely information is required such as the leakage of noxious gas. A mobile robot with sensor attached is more appropriate for such mission to obtain a timely information. It can be deployed fast and easy to the area of interest and acquires more precise data to prominent locations based on its environment-based path design. The method based on the autonomous planning of data acquisition for the usage of environmental surveillance is called *intelligent environmental monitoring* (IEM).[1]

As the mission requires building a model of the concentration over the domain based on previously sampled area, a regression model and an optimization algorithm for sampling are necessary. Jan Gosmann[2] implemented the Gaussian Processes (GPs) and the Bayesian Optimization (BO) in monitoring a time-invariant Gaussian plume distribution. However, different from the static and stable plume, the time-varying environment studied in our thesis is more complicated. A new covariance function, which is critical to the use of GPs regression, is introduced to independently capture the spatial and temporal variations of the data. The necessary functional modifications and improvements are made based on the code of the GPs supplied by Eric[3].

On the other hand, Roman Marchant[1, 4] has made significant contributions to the application of IEM, including monitoring the time-varying ozone concentration and static luminosity distribution with a mobile robot. His work on designing the informative path for robot tracking has achieved remarkable improvement to the process of data sampling. Based on the idea of path planning of Marchant[4], we extend the problem from a 2-D circumstance to a more complex 3-D time-varying environment. Besides, we pay more attention on making use of the temporal optimization, since the temporal variation is as important as the spatial variation in the time-varying environment. In the thesis, we improve the design of monitoring step by step, and eventually develop a practical and effective method to achieve an effective monitoring to the realistic environment.

In the rest of the thesis, the background information about the environment, the regression model, and the optimization algorithm are explained in Chapter. 2. The details of ideas about how we design the experiments and evaluate them are presented in Chapter. 3. The simulation results of those experiments are listed and analyzed in Chapter. 4. Finally, the contribution and future work of this thesis are summarized in Chapter. 5.

Chapter 2

Theoretical Background

2.1 Gaussian Processes

This section summarizes the basics about GPs. It is a stochastic process, meaning each finite and linear collection of those random variables has a multivariate normal distribution. It has been used as a well developed regression method to learn the correlation between the output value and input variables, along with measuring the uncertainty of the output distribution because of the stochasticity of GPs. More details of GPs can be referred from Rasmussen et. al.[5].

2.1.1 Gaussian Processes Fundamentals

The Gaussian Processes regression is one of the supervised learning methods. It learns the pattern between inputs and corresponding target values over a continuous domain. Many applications [1, 3, 4] have proven GPs a powerful tool for non-linear regression. They are very suitable to the optimization algorithm used in this thesis since they are non-parametric and produce a posterior probability density function over the feasible domain. On the other hand, they are very capable of learning both spatially and temporally correlated data. Considering the time-varying environment this thesis focuses on, GPs will be more advantageous under the couple of spatial and temporal variances. Consequently, they are extensively used for regression in the subsequent experiments of this thesis.

In the function that maps input to the output, GPs define a multivariate Gaussian distribution over the space to model the objective function $f(\mathbf{x})$. The model is specified by a mean function $m(\mathbf{x})$ and a covariance function $k(\mathbf{x},\mathbf{x'})$. The observation y we acquired from the environment is the combination of true value $f(\mathbf{x})$ and the Gaussian distributed noise with covariance σ_n^2,

where f is distributed as a GP with mean and covariance functions, i.e.,

$$f(x) \sim GP(m(\mathbf{x}), k(\mathbf{x},\mathbf{x'})) \tag{2.1}$$

where $m(\mathbf{x})$ and $k(\mathbf{x},\mathbf{x'})$ can be expressed as the expectation over the function space.

$$m(\mathbf{x}) = \mathbb{E}_f[f(\mathbf{x})] \tag{2.2}$$

$$k(\mathbf{x},\mathbf{x'}) = \mathbb{E}_f[(y(\mathbf{x}) - m(\mathbf{x}))(y(\mathbf{x'}) - m(\mathbf{x'}))] \tag{2.3}$$

In the thesis, the mean function is set to zero as $m(\mathbf{x}) = 0$, which is a common choice unless the mean needs a more accurate function of representation.

During the training of the GP model in the supervised learning, a set of observations $S = \{X, \mathbf{y}\} = \{\mathbf{x}_i, y_i\}_{i=1}^N$ is gathered from $y(\mathbf{x})$, where $\mathbf{x}_i \in \mathbb{R}^D$ are the N sampled locations in the D dimensional space and $y_i \in \mathbb{R}$ are the N corresponding noisy outputs of objective function. The trained GP is referred as $f^*(\mathbf{x}^*)$ which can estimate the Gaussian distribution over the domain at any specified location $\mathbf{x}^*$. Considering the sampled outputs $\{y_i\}_{i=1}^N$ from environmental objective function $f(\cdot)$ is noisy in the real process, the joint distribution of the observations y at locations X and estimated value f^* at $\mathbf{x}^*$ are given by

$$\begin{pmatrix} y \\ f^* \end{pmatrix} \sim \mathcal{N}\left(0, \begin{bmatrix} K(X,X) + \sigma_n^2 I & K(X,\mathbf{x}^*) \\ K(\mathbf{x}^*,X) & K(\mathbf{x}^*,\mathbf{x}^*) \end{bmatrix}\right) \tag{2.4}$$

And the estimated mean and covariance are

$$m(\mathbf{x}^*)|X,\mathbf{y} = K(\mathbf{x}^*,X)[K(X,X) + \sigma_n^2 I]^{-1}\mathbf{y} \tag{2.5}$$

$$k(\mathbf{x}^*)|X,\mathbf{y} = k(\mathbf{x}^*,\mathbf{x}^*) - K(\mathbf{x}^*,X)[K(X,X) + \sigma_n^2 I]^{-1}K(\mathbf{x}^*,X)^\top \tag{2.6}$$

where $K(X,X)$ is the covariance matrix defined in a component-wise manner as

$$K(X,X')_{(i,j)} = k(\mathbf{x}_i, \mathbf{x}'_j), \text{ with } \mathbf{x}_i \in X \text{ and } \mathbf{x}_j \in X' \tag{2.7}$$

In addition $K(\mathbf{x},X)$ is a covariance vector regarding the training data, defined by

$$K(\mathbf{x},X)_i = k(\mathbf{x},\mathbf{x}_i) \in \mathbb{R}^{1\times N}, \text{ with } \mathbf{x}_i \in X \tag{2.8}$$

As Eq. 2.6 has presented, the covariance function with its hyper-parameters have a strong influence on the estimation at un-sampled location. There are many kinds of covariance functions, each has its own corresponding hyper-parameters. The relevant covariance functions to

the thesis are listed in Sec. 2.1.2, and the method to determine the hyper-parameters is introduced in 2.1.3.

2.1.2 Covariance Functions

In the regression problem using GPs, the covariance function directly determines the performance of the trained model. The covariance function is also known as kernel function which reflects the correlation between two of the locations in the input space. The covariance matrix formed by the element-wise values of covariance function has shown how it determines the estimated value of GP model in Eq. 2.5. Moreover, the covariance function of posterior in Eq. 2.6 reveals a lower variance than the prior in Eq. 2.3, since more information is gained from the samples X.

In the thesis, we will focus on stationary covariance functions, which are expressed only in terms of the distance $r \in \mathbb{R}^1$ between two input locations. To achieve a better representation of the distance in every dimension of input, r is defined as follows,

$$r = \sqrt{(\mathbf{x}-\mathbf{x}')^\top L(\mathbf{x}-\mathbf{x}')} \tag{2.9}$$

where $L = diag(\lambda_1^{-2}, ..., \lambda_D^{-2})$ is a diagonal matrix of size D consisted of the length-scale parameters of the input variables. There are two typical classes of covariance functions used in the thesis, the Matérn covariance and Periodic covariance. The mathematical expression and the related hyper-parameters of the covariance functions are presented in Table.2.1.

Table 2.1: The involved covariance functions.

Name	**Equation**	**Hyperparameters**
Matérn 3	$\sigma_f^2(1+\sqrt{3}r)\exp(-\sqrt{3}r)$	σ_f, L
Matérn 5	$\sigma_f^2(1+\sqrt{5}r+\frac{5}{3}r^2)\exp(-\sqrt{5}r)$	σ_f, L
Periodic	$\sigma_f^2\exp(\frac{-2\sin^2(2\pi\varphi(t-t'))}{\gamma})$	$\sigma_f, \varphi, \gamma$

Since the environment is time-varying, it not only has spatial variations but also shows complicated temporal pattern. In such circumstance, it is not appropriate to couple the spatial and temporal variation together through one single covariance. So a more complex form of the

covariance is designed in order to learn the space and time components independently, which we call it the spatial-temporal *separable covariance* function[6]. Let function $f(s;t)$ represents the environmental objective function, where $s \in \mathbb{R}^D$ is the coordinate in a spatial D-dimension space, $t \in \mathbb{R}^+$ is the one-dimension time variable. The modeled GPs will place a prior as follows, similar as Eq. 2.1,

$$f(s;t) \sim GP(m(s;t), k((s;t),(s;t)')) \tag{2.10}$$

where $k((s;t),(s;t)')$ is the separable covariance function. In the thesis, by a number of tests, we finally design the separable covariance function to be the product of a spatial Matérn 5 covariance function and a temporal sum of Matérn 3 and periodic functions. The advantage to factorize the covariance into spatial and temporal components is that the independent temporal covariance can capture time-specific periodic behavior, which is difficult to achieve only by the independent length-scale in one covariance function. The separable covariance is presented as follows,

$$k((s;t),(s;t)'|\theta) = k_{space}(s,s'|\theta_s)k_{time}(t,t'|\theta_t) \tag{2.11}$$

$$k_{space}(s,s'|\theta_s) = k_{Matérn5}(s,s') \tag{2.12}$$

$$k_{time}(t,t'|\theta_t) = k_{Matérn3}(t,t') + k_{Periodic}(t,t') \tag{2.13}$$

where θ represents the corresponding hyper-parameters in the covariance function. Hence, the covariance matrix in Eq. 2.7 and vector in Eq. 2.8 will become

$$K((S;T),(S;T))_{(i,j)} = k((s;t)_i,(s;t)_j), \text{ with } (s;t)_i,(s;t)_j \in (S;T) \tag{2.14}$$

$$K((s;t)^*,(S;T))_i = k((s;t)^*,(s;t)_i) \in \mathbb{R}^{1\times N} \tag{2.15}$$

The only difference from Eq. 2.5 and 2.6 of the regular GPs is that the input changes from $\mathbf{x}$ to $(s;t)$.

2.1.3 Hyper-parameter Training

Training a GP model means determining the hyper-parameters which define the mean and the covariance functions of the model. The full set of hyper-parameters with the separable covariance is given by $\theta = [\theta_s, \theta_t, \sigma_n]$. The trained hyper-parameters is the optimal set θ^* that best

describes the objective function. One of the ways to determine the optimal hyper-parameters is by maximizing a goal function which involves the training data and the hyper-parameters. In our work, we use one of the popular goal functions, the maximum a posteriori estimate (MAP)[3]. We assume the independent Gaussian distributions as prior distributions on the log-transformed hyper-parameters $\overline{\theta}$.

$$\overline{\theta}_i \sim \mathcal{N}(\mu_i, \sigma_i), \text{ where } \overline{\theta}_i = \log \theta_i, \text{ and } \theta_i \in [\theta_s, \theta_t, \sigma_n] \tag{2.16}$$

where μ_i and σ_i are the mean and covariance that are manually set for the prior of parameters.

The MAP log likelihood is presented as follows,

$$\begin{aligned} L_{MAP}(\overline{\theta}) = &-\frac{1}{2}\mathbf{y}^\top \Sigma^{-1}\mathbf{y} - \frac{1}{2}\log(|\Sigma|) - \frac{N}{2}\log(2\pi) \\ &+ \sum_{i=1}^{P}\left(-\frac{1}{2}\log(2\pi) - \frac{1}{2}\log(\sigma_i^2) - \frac{1}{2\sigma_i^2}(\overline{\theta}_i - \mu_i)^2\right) \end{aligned} \tag{2.17}$$

where $\Sigma = K(X,X) + \sigma_n^2 I$ (see Eq. 2.5) is the sum of covariance matrix and noise's variance. P is the number of hyper-parameters. N is the number of training data. The optimized hyper-parameter set is estimated by the following problem

$$\theta^* = \underset{\theta}{\operatorname{argmax}}\, L_{MAP}(\theta) \tag{2.18}$$

2.2 Bayesian Optimization

The Bayesian Optimization (BO) is one of the most effective algorithms used for searching for the extreme location of an unknown function which is costly to be evaluated. More detailed descriptions about the theory of BO can be found in [7, 8, 9]. The target of Bayesian optimization is to locate the maximum x^* of an unknown function $f(\cdot)$ by

$$x^* = \underset{\mathbf{x}\in\mathbb{R}^D}{\operatorname{argmax}}\, f(\mathbf{x}) \tag{2.19}$$

Different from regular optimization algorithms, BO is particularly suitable for the black-box function optimization when there is no gradient information, and direct evaluation of the function is expensive.

BO is an iterative algorithm based on Bayes' theorem. It sets up a belief for the objective function $f(\cdot)$ with acquired data set $\{X, f(X)\}$ to produce a new estimation of f in next

iteration. In the thesis, GPs are used to represent f over the feasible space. They provide an expected value of f at a known location with an associate variance which evaluates the uncertainty of estimation, as is shown in Eq. 2.3. In each iteration, the next sampled location is tasked to maximize a certain expected improvement. The function that defines and evaluates this improvement for sampled location over the domain is called the *acquisition function*, $h(\cdot)$, which will be explained later. So the goal of BO is to appropriately select the location for next step of evaluation. Through a well designed the acquisition function, BO is able to choose the most promising location that may bring the highest improvement when having the location samples from the unknown function.

2.2.1 Acquisition Functions

The acquisition function $h(\mathbf{x})$ is a fundamental part of BO since it defines the preference of direction for optimization. In this way it reflects the expectation from the user and guides the decision in each iteration until the optimization completes. Generally, the acquisition function should present the probability of finding a higher value of f. As GPs are the representation of the unknown objective function, which offer both the estimated values and the uncertainties over the domain, it becomes important to make a balance between exploitation and exploration while searching for the maximum. The exploitation means the algorithm focusing on a higher estimated mean to increase the maximum. On the other hand, the exploration means the algorithm tries to find a better result through the estimated uncertainties. The most popular acquisition functions are Information Gain (IG), Probability of Improvement (PI), Expected Improvement (EI), and Upper Confidence Bound (UCB).

Information Gain

IG is the simplest acquisition function since it only focuses on where the variance is the highest. Its goal is to sample where not enough understanding of the function has been known, without considering the estimated value. Thus, the IG acquisition function is equivalent to the variance of the GP estimation over the domain, which is not useful for many optimization problems as it is a pure exploration function.

Probability of Improvement

PI computes the probability of improvement than the best location $f(\mathbf{x}^+)$ in the owned data

set. The formula is shown as follows

$$PI(\mathbf{x},\xi) = \Phi\left(\frac{\mu(\mathbf{x}) - f(\mathbf{x}^+) - \xi}{\sigma(\mathbf{x})}\right) \tag{2.20}$$

where $\Phi(\cdot)$ is the Cumulative Density Function (CDF) of the standard normal distribution, and $\xi \geq 0$ is the trade-off parameter balancing the exploration and exploitation. A larger ξ denotes to a higher exploration, and the tendency turns to exploitation while ξ goes to zero.[10]

The first derivative function of PI regarding $\mathbf{x}$ is also presented as follows, which is necessary for a gradient-based optimization algorithm.

$$\nabla PI(\mathbf{x},\xi) = \nabla\left(\frac{\mu(\mathbf{x}) - f(\mathbf{x}^+) - \xi}{\sigma(\mathbf{x})}\right)\phi\left(\frac{\mu(\mathbf{x}) - f(\mathbf{x}^+) - \xi}{\sigma(\mathbf{x})}\right) \tag{2.21}$$

$$= \frac{\nabla\mu(\mathbf{x})\sigma(\mathbf{x}) - \nabla\sigma(\mathbf{x})(\mu(\mathbf{x}) - f(\mathbf{x}^+) - \xi)}{\sigma^2(\mathbf{x})}\phi\left(\frac{\mu(\mathbf{x}) - f(\mathbf{x}^+) - \xi}{\sigma(\mathbf{x})}\right) \tag{2.22}$$

where ϕ denotes to the Probability Density Function (PDF) of the standard normal distribution. The equations of $\nabla\mu(\mathbf{x})$ and $\nabla\sigma(\mathbf{x})$ are

$$\nabla\mu(\mathbf{x}) = \sum_{i=1}^{N}[\Sigma^{-1}\mathbf{y}]_i \nabla k(\mathbf{x},\mathbf{x}_i) \tag{2.23}$$

$$\nabla\sigma(\mathbf{x}) = \frac{\sigma^2(\mathbf{x})}{2\sigma(\mathbf{x})} \tag{2.24}$$

$$\nabla\sigma^2(\mathbf{x}) = -\sum_{j=1}^{N}\sum_{i=1}^{N}\Sigma_{ij}^{-1}\left[\nabla k(\mathbf{x},\mathbf{x}_i)k(\mathbf{x},\mathbf{x}_j) + k(\mathbf{x},\mathbf{x}_i)\nabla k(\mathbf{x},\mathbf{x}_j)\right] \tag{2.25}$$

Σ is the same matrix as in Eq. 2.17 where $\Sigma = K(X,X) + \sigma_n^2 I$.

Expected Improvement

EI is the acquisition function that measures the improvement of the chosen location. As is shown as follows, EI will always be positive, only be zero when no improvement at the point such as at the known samples.

$$EI(\mathbf{x},\xi) = \begin{cases} (\mu(\mathbf{x}) - f(\mathbf{x}^+) - \xi)\Phi(Z) + \sigma(\mathbf{x})\phi(Z) & \text{if } \sigma(\mathbf{x}) > 0 \\ 0 & \text{if } \sigma(\mathbf{x}) = 0 \end{cases} \tag{2.26}$$

where

$$Z = \begin{cases} \frac{\mu(\mathbf{x}) - f(\mathbf{x}^+) - \xi}{\sigma(\mathbf{x})} & \text{if } \sigma(\mathbf{x}) > 0 \\ 0 & \text{if } \sigma(\mathbf{x}) = 0 \end{cases} \tag{2.27}$$

The parameter ξ here has the same effect as it has in Eq. 2.20. Lizotte[10] suggests $\xi = 0.01$ works well in most cases through the experiments.

The first derivative function of EI as $\sigma(\mathbf{x}) > 0$ is as follows

$$\begin{aligned}\nabla EI(\mathbf{x},\xi) =& \nabla\mu(\mathbf{x})\Phi(Z) + [\mu(\mathbf{x}) - f(\mathbf{x}^+) - \xi]\phi(Z)\nabla Z \\ &+ \nabla\sigma(\mathbf{x})\phi(Z) + \sigma(\mathbf{x})[-Z\phi(Z)\nabla Z]\end{aligned} \tag{2.28}$$

$$\nabla Z = \frac{\nabla\mu(\mathbf{x})\sigma(\mathbf{x}) - \nabla\sigma(\mathbf{x})(\mu(\mathbf{x}) - f(\mathbf{x}^+) - \xi)}{\sigma^2(\mathbf{x})} \tag{2.29}$$

Although EI and PI are powerful functions on optimizing the global optimum, their complex formulas bring a heavy cost of computation when the evaluation of acquisition function increases. EI and PI with their derivative functions are powerful to locate the global maximum. However, their potentials will be studied in future works.

Upper Confidence Bound

UCB acquisition function[11] involves the estimated mean and variance in a simple form as

$$UCB(\mathbf{x},\kappa) = \mu(\mathbf{x}) + \kappa\sigma(\mathbf{x}) \tag{2.30}$$

The parameter κ is related to the trade-off between exploration and exploitation.

Marchant[4] brings an idea that the distance should be considered in the reality that monitoring a large area with a mobile UAV, which could save energy of engine from long distance travels. However, considering the area is small and the speed of robot is high enough to cover the area in our problem, we will not involve the distance in the acquisition function at this time. The regular UCB will be used in the thesis as the acquisition function.

2.2.2 Algorithm

The details of the BO algorithm is shown in Table. 2.2. The notation $\mathbf{x}$ is equivalent to (s,t) from the separable covariance of GPs. Line 3 is the optimization of the acquisition function, where we introduce the DIRECT algorithm by Finkel[12] to locate an initial guess for the optimum, then use a MATLAB solver 'fmincon'[13] to find the global optimum starting at the

Table 2.2: Bayesian Optimization algorithm.

Algorithm 1 Bayesian Optimization

Inputs: f, h, data samples $\{X, \mathbf{y}\}$
Outputs: GP model, $\mathbf{x}_{max}$, y_{max}
1. **while** $i \leq i_{max}$
2. Train GP model with data $\{X, \mathbf{y}\}$
3. $\mathbf{x}^* \leftarrow \text{argmax}_{\mathbf{x} \in A^D} h(\mathbf{x})$
4. $y^* = f(\mathbf{x}^*) + \varepsilon$
5. Augment data set $\{X, \mathbf{y}\}$ with $[\mathbf{x}^*, y^*]$
6. **if** $y^* > y_{max}$
7. $y_{max} \leftarrow y^*$
8. $\mathbf{x}_{max} \leftarrow \mathbf{x}^*$
9. **end if**
10. $i \leftarrow i + 1$
11. **end while**

initial guess. $A^D \in \mathbb{R}^D$ in Line. 3 is a finite vector space which is the feasible space for vector $\mathbf{x}$.

It is obvious that the original problem of maximizing the objective function $f(\cdot)$ has been switched to searching the maximum of the acquisition function $h(\cdot)$. The acquisition function based on GPs is like a surrogate model to the objective function. In fact, with an appropriate selection from the acquisition function, the optimization will become easier and faster in the domain, because the acquisition function can provide additional derivative information such as Eq. 2.21 and 2.28, and is much cheaper to be evaluated than the objective function.

2.3 Gaussian Puff Environment

This section gives the details about the environmental model studied in the thesis. The environment we study is a dispersion model of gas, which simulates a mass of gas released into the space at one or multiple sources. This phenomenon is quite common in environmental monitoring applications, such as locating the sudden leakage of pollutant gas along its pipeline and monitoring the gas emission in the drain system of industry. Therefore the environmental model should be dynamic, considering the wind effect, the diffusion characteristics, and the variation in both space and time.

The Gaussian puff dispersion model[14] is chosen to be the time-varying environment in the thesis. Different from the static Gaussian plume model with constant emission rate of gas,

the puff model is dynamic and emits gas intermittently from its sources. The blob-like puff is released at certain time T and blown by the constant horizontal wind with a speed of $u = 1$m/s, which successfully creates a time-varying gas concentration. At a location x, y, z and time t in the wind frame, we assume the wind is horizontal for simplicity, the concentration is the superposition of all the puffs generated so far as follows,

$$c(x,y,z,t) = \sum_{i=1}^{I} \left\{ \frac{Q_s^i}{8(\pi a(x-X_s)^b)^{3/2}} \exp\left(-\frac{(x-X_s-u(\tau t - T_s^i))^2 + (y-Y_s)^2}{2a(x-X_s)^b} \right) \right. \\ \left. \left[\exp\left(-\frac{(z-H_s)^2}{2a(x-X_s)^b} \right) + \exp\left(-\frac{(z+H_s)^2}{2a(x-X_s)^b} \right) \right] \right\} \tag{2.31}$$

where $[X_s, Y_s, H_s]$ are the coordinates of source s in the wind frame. a and b are the dispersion coefficients[15] whose values are manually set. $\tau \in (0,1)$ is a time coefficient which equals to 1 in the original model. However, since in the original puff model the dispersion is too fast, it can easily be monitored as the level of concentration goes down to zero too soon. So we further make τ smaller than 1 to slow down the temporal variation of the model, to make it harder to be monitored.

In Eq. 2.31, Q_s^i is the i-th amount of emission released by source s at time T_s^i; a total number of I emissions have been released by source s before time t. However in the thesis, only one puff is released by each source at time zero, which is because the release of puff will bring a sudden step change to the environment. As in reality, the sudden emission is unpredictable, which means no information about it can be provided to the regression model, which will be impossible to learn. So in the thesis, we only study the dynamics of environment with a single puff at the initial time.

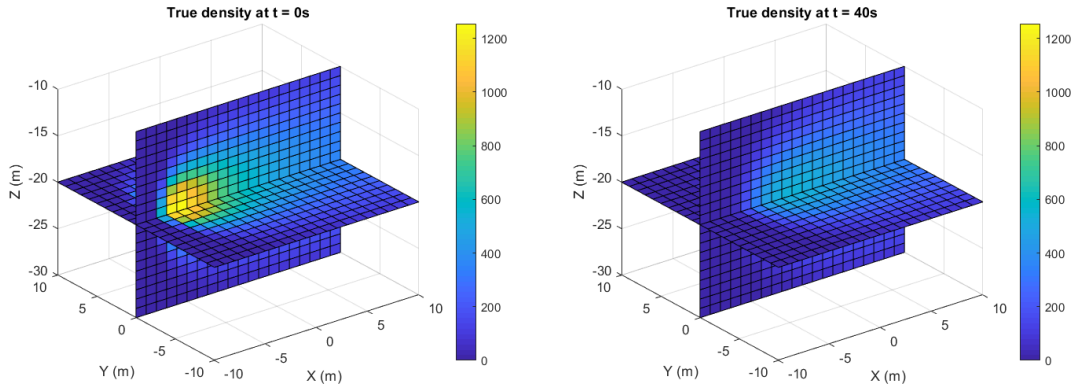
It should be noted that the puff model is cited from a quadrotor simulator[14], the coordinate vector is in the NED (north, east, down) reference frame. Thus the height coordinate is on the negative z-axis.

Table 2.3: The setting of the Gaussian puff model.

Parameter	Q_s^1 (g/s)	T_s^1 (s)	u (m/s)	τ	a	b	Room size (m)
Value	1.0	0	1	0.1	3.3	0.86	$20 \times 20 \times 20$

Table. 2.3 lists the set of the parameters of the time-varying environment model studied in the thesis. The single source is located at $[X_s, Y_s, H_s] = [-9, 0, -20]$ in the room. Fig. 2.1

illustrates the environmental dispersion at time $t = 0$ s and $t = 40$ s, where the concentration of gas is obviously decaying over time.



(a) $t = 0$ s (b) $t = 40$ s

Figure 2.1: The true gas dispersion of the environment at two instances.

Chapter 3

Experiment Design

The Gaussian Process regression, the Bayesian Optimization algorithm, and the environment model have been well explained in Chapter. 2. The detailed design of experiments will be introduced in this chapter based on the theoretical frame that has been presented. However, some settings are varied among different experiments. The design logic is to transfer gradually from idealization to practical realization. It means in the beginning, the experiment is conducted theoretically which includes some unrealistic assumptions. The goal is to verify the effectiveness of the proposed method. Then the assumption will be replaced to a real condition to prove the practicability of the method.

3.1 Sampling by Discrete Data

Before using a mobile device to collect sample, we firstly assume there is no spatial or temporal restriction to the sampling process. The sample can be measured in the environment at any location and time as request. This is not practicable in the real world since neither the sensor network is such intensive nor any mobile sensors is able to move such timely and precisely. However, these experiments are conducted to show the capability of BO in monitoring a time-varying environment. A more realistic problem will be studied in Sec. 3.2.

The experimental procedure basically follows the algorithm presented in Table. 2.2. A constant number of i_{max} and a sampling rate (the number of sample collected in a certain period of time) are fixed in order to ensure the total number, N, of samples is the same. And those sample are collected within the same time interval $[0, T_{end}]$. The main distinction among the experiments in this section, is the way to deal with the optimized temporal variable t^*, which is related to Line. 3 and 4 in Table. 2.2.

3.1.1 Free Sampling over Time

In this experiment, the procedure is exactly the same as BO but change the input format from $\mathbf{x}$ to (s,t), as in Table. 3.1. Each iteration in Line. 3 of Table. 3.1, the optimized output includes both spatial and temporal terms. By the DIRECT algorithm[12] and MATLAB solver 'fmincon'[13], the spatial and temporal variables are searched within the bounds of spatial feasible space A^D and time limit T_{end}, respectively.

Table 3.1: Free sampling algorithm.

Algorithm 2 Free sampling of BO
Inputs: f, h, data samples $\{(S,T),\mathbf{y}\}$
Outputs: GP model, $(s,t)_{max}$, y_{max}
1. **while** $i \leq i_{max}$
2. Train GP model with data $\{(S,T),\mathbf{y}\}$
3. $(s^*,t^*) \leftarrow \text{argmax}_{s\in A^D,\, t\in[0,T_{end}]}\, h(s,t)$
4. $y^* = f(s^*,t^*)+\varepsilon$
5. Augment data set $\{(S,T),\mathbf{y}\}$ with $[(s^*,t^*),y^*]$
6. **if** $y^* > y_{max}$
7. $y_{max} \leftarrow y^*$
8. $(s,t)_{max} \leftarrow (s^*,t^*)$
9. **end if**
10. $i \leftarrow i+1$
11.**end while**

3.1.2 Time-abandon Sampling

Sec. 3.1.1 allows the acquisition function to find the sample that brings the highest improvement within the predefined feasible domain in Line 3, and measures the sample directly in Line. 4 of Table. 3.1. However, this design is un-practical. Because it is possible that a sample $(s^*,t^*)_{i+1}$ optimized in $(i+1)$-th iteration has a smaller temporal variable than the sample $(s^*,t^*)_i$ collected in previous i-th iteration, which means $t^*_{i+1} < t^*_i$. This time travel is obviously impossible in real sample measurement.

In the experiments performed by Marchant[1], he abandons the temporal variable t^* and uses the optimized spatial location s^* only as the designated position for the UAV to sample. Moreover, the data is measured as soon as the UAV arrives position s^*. Considering to maintain the right order of time, a similar design is implemented in this section. We no longer use t^* as the time for sampling, instead, an iteration-order-based time is assumed to be the time for

sample in each iteration. The detail is presented in Table. 3.2, where $\delta t = T_{end}/i_{max}$ is the time gap between iterations.

Table 3.2: Time-abandon sampling algorithm.

Algorithm 3 Time-abandon sampling of BO

Inputs: f, h, data samples $\{(S,T),\mathbf{y}\}$, δt
Outputs: GP model, $(s,t)_{max}$, y_{max}
1. **while** $i \leq i_{max}$
2. Train GP model with data $\{(S,T),\mathbf{y}\}$
3. $(s^*,t^*) \leftarrow \text{argmax}_{s\in A^D,\, t\in[0,T_{end}]}\, h(s,t)$
4. Get current time $t_i \leftarrow i \times \delta t$ of i-th iteration
5. $y^* = f(s^*,t_i) + \varepsilon$
6. Augment data set $\{(S,T),\mathbf{y}\}$ with $[(s^*,t_i),y^*]$
7. **if** $y^* > y_{max}$
8. $y_{max} \leftarrow y^*$
9. $(s,t)_{max} \leftarrow (s^*,t_i)$
10. **end if**
11. $i \leftarrow i+1$
12. **end while**

After all, this design is based on an assumption that a mobile sensor is assigned to sample at s^* in i-th iteration, then it arrives and measures at time t_i. It should be noted that the BO with this design is no longer be able to find a true global optimum, since the optimized temporal part t^* is not considered in the sampling.

3.1.3 Ascending Order Sampling

The design in this section is built on Sec. 3.1.2. In order to both make use of the optimized temporal part t^* and comply with the right order of time, the temporal bounds for searching during optimization are designed to a practical interval as Table. 3.3, Line 4. This design is based on a similar assumption as Sec. 3.1.2, that the mobile sensor is fast enough to arrive s^* on time t^*.

3.2 Sampling over Linear Path

In Sec. 3.1, the sensor is assumed capable of arriving at any position s^* on time t^* as BO has requested, which is impossible in reality when we are monitoring an environment without an intensive sensor network. Now we assume that we are solving a real monitoring problem of

Table 3.3: Ascending order sampling algorithm.

Algorithm 4 Ascending order sampling of BO

Inputs: f, h, data samples $\{(S,T),\mathbf{y}\}$, δt
Outputs: GP model, $(s,t)_{max}$, y_{max}
1. Current time $t = 0$
2. **while** $i \leq i_{max}$
3. Train GP model with data $\{(S,T),\mathbf{y}\}$
4. Set bounds $t_{lb} \leftarrow t$ and $t_{ub} \leftarrow i \times \delta t$
5. $(s^*,t^*) \leftarrow \text{argmax}_{s\in A^D,\, t\in[t_{lb},t_{ub}]}\, h(s,t)$
6. $y^* = f(s^*,t^*) + \varepsilon$
7. Augment data set $\{(S,T),\mathbf{y}\}$ with $[(s^*,t^*),y^*]$
8. **if** $y^* > y_{max}$
9. $y_{max} \leftarrow y^*$
10. $(s,t)_{max} \leftarrow (s^*,t^*)$
11. **end if**
12. $i \leftarrow i+1$
13. $t \leftarrow t^*$
14. **end while**

the time-varying environment. An UAV is involved in this section as the mobile sensor, which will fly following the planned path while collecting samples on the way with a constant time interval. As the total running time $Tend$ and time interval δt are fixed, the data collected are still the same in different experiments. So instead of stopping the simulation after a number of iterations in Sec. 3.1, the simulation in this section will end when the time exceeds its limit.

3.2.1 Waypoint Planning

The waypoint planning means the planned path is connected by waypoints, and the paths between each pair of waypoints are straight lines. This design is based on Sec. 3.1.3. The improvement is the waypoint planning considers the actual differences between the spatial distance and time span between two optimized waypoints, that makes sure the UAV with sensor could practically fly along the path and collect additional samples on the way. The optimized spatial location s^* is assigned to be the next waypoint, and the UAV will arrive there at time t^*. In order to make sure the flying speed along each straight line is reasonable, the temporal bounds for optimization should be modified by a constant Δt which is determined manually considering the size of the feasible space A^D. It should be noted that since the data is collected with a fixed time interval, which is not considered in the optimization of t^*, the waypoints

may not be included in samples. However, if the time interval is small enough, the spatial and temporal deviations are acceptable.

Table 3.4: Linear path algorithm.

Algorithm 5 Linear path planning of BO

Inputs: f, h, data samples $\{(S,T),\mathbf{y}\}$, δt, Δt
Outputs: GP model
1. **while** $t_{now} < T_{end}$
2. Train GP model with data $\{(S,T),\mathbf{y}\}$
3. Set bounds $t_{lb} \leftarrow t_{now}$ and $t_{ub} \leftarrow t_{now} + \Delta t$
4. $(s^*,t^*) \leftarrow \text{argmax}_{s\in A^D,\, t\in[t_{lb},t_{ub}]}\, h(s,t)$
5. **while** $t_{now} < t^*$
6. Move UAV towards (s^*,t^*)
7. $t_{now} \leftarrow t_{now} + \delta t$, update UAV position
8. Collect samples from $y = f + \varepsilon$
9. **end while**
10. Augment data set $\{(S,T),\mathbf{y}\}$ with new samples
11. $t_{now} \leftarrow t^*$
12. **end while**

In the algorithm with linear path planning, there are many samples are collected along the path instead of only sampling at the optimized location as Sec. 3.1. The BO therefore will run much less times in a same period of simulation time. Those samples collected along the path are not happen to be the global maximum since they are not involved within the optimization process. In this circumstance, locating the global maximum with samples as Sec. 3.1 is not possible anymore. BO has to locate the maximum by obtaining a GP model that can best represent the environment. This realistic change makes us focus more on how to collect informative samples that ba able to train a GP model representing the objective function $f(\cdot)$ accurately.

3.2.2 Continuous Path Planning

The idea of continuous path is proposed by Marchant[4] for a 2-D path planning. It generalizes the waypoint line into a continuous path. The optimization algorithm in the waypoint planning has not considered the majority of samples which are collected along the paths. The samples at optimized waypoints are just a few. So in fact, most samples are not elaborated to gain useful information for the monitoring. The continuous path planning uses a parameterized curve to actively accumulate the most information, or reward, during the sampling along the

curve. Then the optimization problem in Line. 3, Algorithm 2, Table. 3.1 is transformed into the optimization of the best parameters of the continuous path, by maximizing the reward function.

Path Parameterization

In the thesis, a third order polynomial is defined to be the path function $C(u|\beta)$, where β is the parameter vector and $u \in [0,1]$ is the normalized variable of the function. Considering a 3-dimensional space, $D = 3$,

$$C(u|\beta) : [0,1] \rightarrow \mathbb{R}^4$$

$$u \rightarrow X(u) = a_1u^3 + b_1u^2 + c_1u + d_1 \tag{3.1}$$

$$Y(u) = a_2u^3 + b_2u^2 + c_2u + d_2 \tag{3.2}$$

$$Z(u) = a_3u^3 + b_3u^2 + c_3u + d_3 \tag{3.3}$$

$$t(u) = \frac{1}{v_l}\int_0^u \|C(\gamma|\beta)\|d\gamma \tag{3.4}$$

where $(X(u), Y(u), Z(u), t(u))$ denote the corresponding coordinates and time with a variable u. The parameter vector is $\beta = \{v_l, a_1, b_1, c_1, d_1, a_2, b_2, c_2, d_2, a_3, b_3, c_3, d_3\}$, where v_l is the speed in the current piece of path. It is important to note that according to Eq. 3.4, the temporal variable needs no optimization any more, which means we won't bother to determine the temporal bounds for optimization process. With continuous path, the designated time t^* is now obtained by the length of the curve (determined by the curve parameters) and the optimized speed v_l.

There are some initial conditions that could help to reduce the number of parameters, such as a known initial position as Eq. 3.5 and first partial derivative as Eq. 3.6.

$$X_{i-1}(1) = X_i(0) = d_1, \;\; Y_{i-1}(1) = Y_i(0) = d_2, \;\; Z_{i-1}(1) = Z_i(0) = d_3 \tag{3.5}$$

$$\frac{X'_{i-1}(1)}{\|C'_{i-1}(1)\|} = X'_i(0) = c_1, \;\; \frac{Y'_{i-1}(1)}{\|C'_{i-1}(1)\|} = Y'_i(0) = c_2, \;\; \frac{Z'_{i-1}(1)}{\|C'_{i-1}(1)\|} = Z'_i(0) = c_3 \tag{3.6}$$

where the subscript $i-1$ denotes to the $(i-1)$-th piece of path, and the subscript i is to the i-th piece who starts at where the $(i-1)$-th piece ends. In Eq. 3.6, $\|C'(1)\|$ indicates the magnitude of the vector of the first derivative. This management of the first derivative maintains the information of the heading direction of the UAV, while avoiding the problem that the next designed curve rushes out of the domain with an over-large first derivative. These conditions left seven parameters $\beta = \{v_l, a_1, b_1, a_2, b_2, a_3, b_3\}$.

Reward Function

The reward function $r(\cdot)$ is associated with the decision. In Table. 3.4, the reward function, which is also the acquisition function $h(\cdot)$, is evaluated at a single location. But in the continuous path, the reward function is designed to accumulate all reward of sampling along the continuous path. Thus, its expression corresponds to an integral of the acquisition function $h(\cdot)$. Marchant[4] has proposed a form of reward function as Eq. 3.7.

$$r(C(u|\beta)|h) = \int_{C(u|\beta)} h(v)dv \tag{3.7}$$

However, optimizing Eq. 3.7 respect to β may lead to a problem which will produce an endless stretching of the curve. This is because when $h(\cdot)$ is a positive acquisition function, any extension of the curve will always increase the value of integral, which weakens the benefit from the optimization.

So we decide to subtract a baseline b from the acquisition function, which makes the reward function to Eq. 3.8, where $h(v)$ is designed to be the UCB (Eq. 2.30) when we implement the continuous path in the thesis.

$$r(C(u|\beta)|h,b) = \int_{C(u|\beta)} [h(v) - b]dv \tag{3.8}$$

$$= \int_0^1 [\mu(C(u|\beta)) + \kappa\sigma(C(u|\beta)) - b]\|C'(u|\beta)\|du \tag{3.9}$$

b is a scalar. It helps to change the positive domain of acquisition function to a domain has both positive and negative values. Although there are many methods to determine its value, we eventually set it as the mean of acquisition values of current data samples. The function of integral may not have an analytical solution, particularly depending on the choice of acquisition function and covariance function. So the integral is numerically computed using a global adaptive quadrature approximation.

Thus the optimization problem is transformed to maximizing the reward respect to all possible paths. The optimal path parameters β^* will determine the curve function $C(u,\beta^*)$ including the path and speed that the UAV should track.

$$\beta^* = \underset{\beta}{\operatorname{argmax}}\, r(C(u,\beta)|h,b) \tag{3.10}$$

In the thesis, this optimization of reward is solved the same way as the Algorithm. 1, with DIRECT algorithm[12] and a MATLAB solver 'fmincon'[13].

Algorithm

The algorithm of continuous path planning is presented in Table. 3.5.

Table 3.5: Continuous path algorithm.

Algorithm 6 Continuous path planning of BO
Inputs: f, h, data samples $\{(S,T),\mathbf{y}\}$, δt
Outputs: GP model
1. **while** $t_{now} < T_{end}$
2. Train GP model with data $\{(S,T),\mathbf{y}\}$
3. Update $b = mean(h(S,T))$
4. $\beta^* \leftarrow \text{argmax}_\beta\, r(C(u,\beta)|h,b)$
5. $(s^*,t^*) \leftarrow C(u,beta^*)$
6. **while** $t_{now} < t^*$
7. Move UAV along C
8. $t_{now} \leftarrow t_{now} + \delta t$, update UAV position
9. Collect samples from $y = f + \varepsilon$
10. **end while**
11. Augment data set $\{(S,T),\mathbf{y}\}$ with new samples
12. $t_{now} \leftarrow t^*$
13.**end while**

3.3 Metrics

A better BO algorithm leads to a more informative sample collection that could train a GP model that precisely represent the environment monitored. Various experiments are quantitatively evaluated using four different performance indicators, to present the error between the true concentration of the environment $\mu_t(\mathbf{x})$ and the estimated value $\mu(\mathbf{x})$ from the GP model. The GP model is trained with a fixed number of data after enough samples of data have been collected during the simulation. Then the performance of GP is evaluated over the whole domain using a fine grid resolution with M samples. The evaluation is conducted at several moments of time to see its variation with time.

Root Mean Squared Error

The first metric is the Root Mean Squared Error (RMSE), that reveals the mean error of the estimated values.

$$RMSE = \sqrt{\frac{1}{M}\sum_{i=1}^{M}(\mu(\mathbf{x}_i) - \mu_t(\mathbf{x}_i))^2} \tag{3.11}$$

with $\mathbf{x} = (s,t)$ for the spatial-temporal case. However, the averaged error means that the indicator has treated the low values of concentration equally as the high peaks in the domain. Therefore we propose two methods to give more attention to the error at the high concentration, which is more important to our problem. The first method is using the DIRECT algorithm to locate the place of high concentration based on the environment model in Eq. 2.31, then evaluating the metrics within the 1-meter cube surrounding that place. Although this method is not practical in real application, because we are not able to possibly know the place of high concentration without the accurate environment model, still, it is only for the performance evaluation of GP at the high concentration.

Weighted Root Mean Squared Error

The second method is using a developed metric called Weighted Root Mean Squared Error (WRMSE).

$$WRMSE = \sqrt{\sum_{i=1}^{M} \alpha(\mathbf{x}_i)(\mu(\mathbf{x}_i) - \mu_t(\mathbf{x}_i))^2} \tag{3.12}$$

where $\alpha(\mathbf{x}_i)$ is a weight coefficient as

$$\alpha(\mathbf{x}_i) = \frac{\mu_t(\mathbf{x}_i)}{\sum_{j=1}^{M} \mu_t(\mathbf{x}_j)} \tag{3.13}$$

WRMSE is in fact similar to RMSE but the weight for the squared errors is changed from an identical $\frac{1}{M}$ to Eq. 3.13 that depends on the local true concentration. This modification gives more importance to the error whose true concentration of the studied phenomenon is higher. Then the evaluation by WRMSE will become more convincing over the domain. This concern is from a logical assumption that in a leakage of some dangerous gas, it is more important for the monitoring to locate and have precise estimation in the high concentrated area than in the entire domain when the time and source are short. On the other hand, by comparing the RMSE and WRMSE, we can determine whether the BO has found the high concentration, so these two metrics will be analyzed together.

Mean Log Loss

As GP model is one of the probability-based stochastic models. The Mean Log Loss (MLL)

is used to evaluate the negative log probability of the true concentration under the model estimation. The estimated error and associated uncertainty are both taken into account.

$$\begin{aligned} MLL &= \frac{1}{M}\sum_{i=1}^{M}(-\log p(\mu_t(\mathbf{x}_i)|\mu,\sigma))^2 \\ &= \frac{1}{M}\sum_{i=1}^{M}(\frac{1}{2}\log(2\pi\sigma^2(\mathbf{x}_i)) + \frac{(\mu(\mathbf{x}_i)-\mu_t(\mathbf{x}_i))^2}{2\sigma^2(\mathbf{x}_i)} \end{aligned} \tag{3.14}$$

Weighted Mean Log Loss

Similar to WRMSE, the Weighted Mean Log Loss (WMLL) weights over the MLL. This metric gives more importance to the high concentrated area as well, taking the variance of GP model into account. The weight coefficient is the same as Eq. 3.13.

$$WMLL = \sum_{i=1}^{M}\left(\alpha(\mathbf{x}_i)(\frac{1}{2}\log(2\pi\sigma^2(\mathbf{x}_i)) + \frac{(\mu(\mathbf{x}_i)-\mu_t(\mathbf{x}_i))^2}{2\sigma^2(\mathbf{x}_i)}\right) \tag{3.15}$$

Relative Difference

Relative difference (RD) is one of the relative metrics that imply the scale of the relative error. The definition of RD is shown in Eq. 3.16.

$$RD = \frac{\mu(\mathbf{x}_i)-\mu_t(\mathbf{x}_i)}{(|\mu(\mathbf{x}_i)|+|\mu_t(\mathbf{x}_i)|)/2} = 2\times\frac{\mu(\mathbf{x}_i)-\mu_t(\mathbf{x}_i)}{|\mu(\mathbf{x}_i)|+|\mu_t(\mathbf{x}_i)|} \tag{3.16}$$

It reveals that the value of RD is between -2 and 2. The RD tells the ratio of error to the total of the estimated and the true values, while maintaining the sign of the error. Moreover, its advantage than the Relative Error (the ratio of error to the true concentration) is the value of relative difference will not become a singularity when the true concentration is zero. The zero value of concentration in a particular area is not avoidable in the gas dispersion model. However, it should be noted that because the denominator of Eq. 3.16, the absolute value of RD can easily become large (close to 2) where the true value μ_t is small and the error is relatively big. So it is not appropriate to use RD as one of the performance indicators in the entire domain, since the mean of RDs from M samples can be easily effected by the small change of error at where the true concentration is small.

Chapter 4

Simulation Results

In this chapter, the proposed tests are simulated and the results are analyzed in two parts: the discrete sampling experiments which include three scenarios, and the continuous sampling experiments including two scenarios. The order of scenarios follows Sec. 3.1 and 3.2. Some parameters of the objective function of the environment model are presented in Table. 2.3. The acquisition function used is UCB in Eq. 2.30 with $\kappa = 0.1$ determined by a trail and error process. The environment are using the same settings as Table. 2.3, and the UCB is implemented as the acquisition function h in all scenarios in this chapter.

4.1 Discrete Samplings

In each iteration of BO, the GP model is trained and only one sample is obtained by optimizing the acquisition function, as is shown in Table. 2.2, Algorithm 1. The sample optimized is believed to bring the most improvement on exploitation and exploration for the next GP model, which means the simulation is aiming to explore the environment more efficiently and locate the global optimum as fast as possible. It is clear in Fig. 2.1 that the high concentration of the environment is changing with time, both quantitatively and spatially. So a desired result in this section has to estimate precisely at the position of high concentration, or at least in the area surrounding that place.

The environment settings are shown in Table. 2.3. And the information of simulations in this section are presented in Table. 4.1. The settings are all the same in the following experiments in Sec. 4.1. The test interval means the temporal interval where the performance of GP is

Table 4.1: The setting of the discrete sampling experiments.

Sampling T_{end} (s)	δt (s)	Iteration i_{max}	Test interval (s)
10	0.1	100	[0, 10]

evaluated. The GP is trained with 101 data points after the simulation stops, which includes 1 initial data and 100 samples collected by 100 times of iteration.

4.1.1 Free Sampling

The highest concentration found by the method is shown in Table. 4.2. t_{max} is 0.001 s, meaning the method has found a maximum very early in the timeline. This is not surprised because the time-free sampling can sample any location at any time. $f_{max}(t_{max})$ denotes to the value of true global maximum at time t_{max}, which is used to assess the maximum found by the method.

Table 4.2: The maximum found by the free sampling method.

$f(s_{max}, t_{max})$ (mg/m^3)	s_{max} (m)	t_{max} (s)	Iteration serial	$f_{max}(t_{max})$
1277.218	[-6.46, -0.01, -19.99]	0.001	99	1277.224

Fig. 4.1 presents the collected samples by colored points. The color of each point indicates the time variable of the sample, which is the optimized temporal variable by BO in the free sampling method. It should be noted that as BO with free sampling can sample freely along the timeline, the distribution of samples are not necessarily uniform along the timeline.

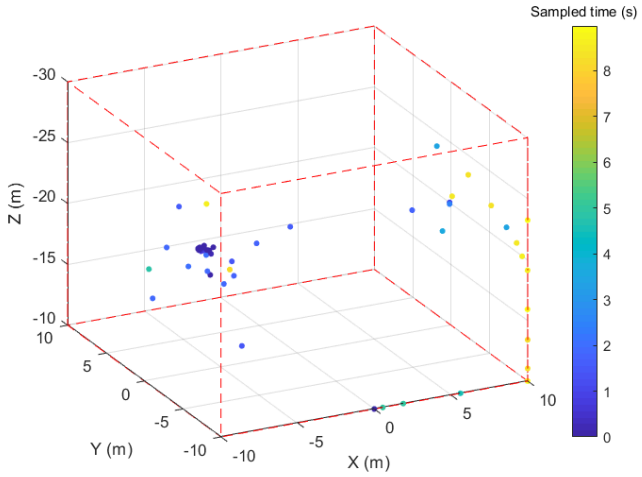
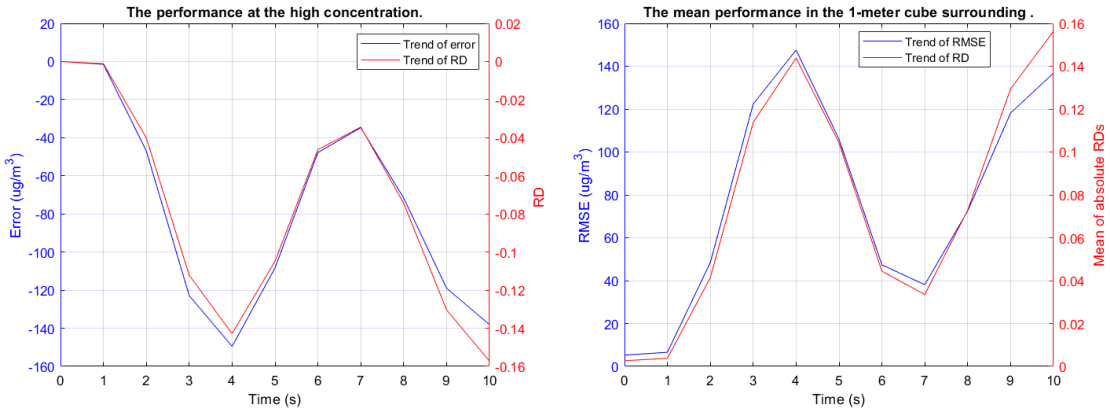


Figure 4.1: The samples collected by free sampling, colored by the time.

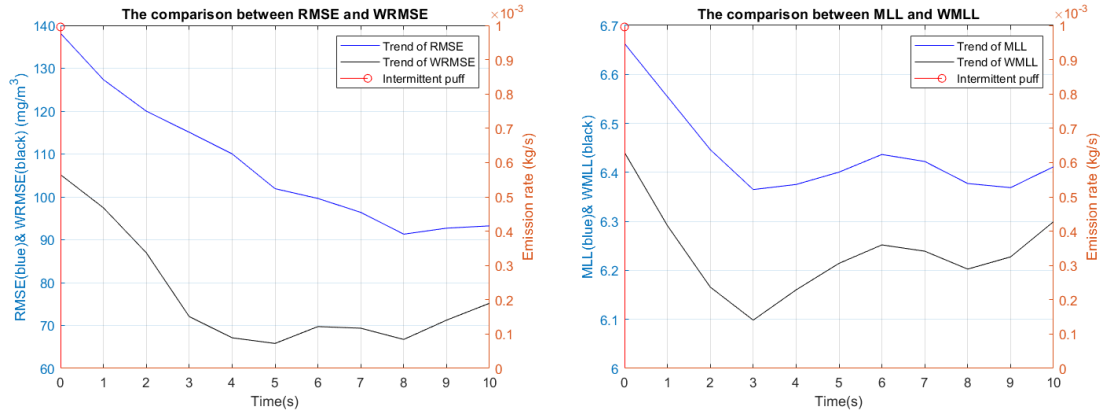
Fig. 4.2 and 4.3 present the performance of GP near the high concentration (global maximum) and in the entire domain, respectively. It should be noted that Fig. 4.2 (a) shows the trend of error instead of RMSE or MLL, because it is the performance at the point of high concentration. Moreover, as the location and value of the high concentration is time varying, it

is reasonable that a continuous sampling is necessary to maintain an accurate monitoring to the high concentration.



(a) Error and RD (b) RMSE and RD

Figure 4.2: The performance (a) at and (b) around the time-varying global maximum by free sampling.



(a) (b)

Figure 4.3: The overall performance of (a) RMSE & WRMSE and (b) MLL & WMLL by free sampling.

Fig. 4.2 illustrates that the GP has a nearly zero error at the high concentration when the time begins ($t \in [0, 1]$ s), then the error increases with time. Considering the concentration of blue points in Fig. 4.1 near the global maximum and Table. 4.2, this method has obviously collected many samples whose temporal variable is small at the right location that is the global maximum. This ability of finding the global maximum both spatially and temporally has proven the capability of BO in searching the global maximum in a 4-dimensional problem (spatial 3-D plus temporal 1-D).

However, this method also reveals some disadvantages when we examine the performance related to the temporal variation. Besides ignoring the real time order of sampling process, the BO has lost interest to the high concentration once it has found the global maximum at $t = 0.001$ s, which induces the increase of error at the global maximum in Fig. 4.2. Although there is another good time at $t = 7$ s, this is resulted by several samples near the maximum whose temporal variable is around 7. The overall error at the global maximum is obviously getting worse after $t = 1$ s. And since the only element that encourages BO to explore other places in the room is the exploration from the acquisition function, the performances in the room in Fig. 4.3 are just decent. However, the weighted performances are much better because of the contribution from finding the global maximum.

4.1.2 Time-abandon Sampling

The highest concentration found by the method is shown in Table. 4.3. Compared with Table. 4.2, this method has not temporally found the global maximum, which is comprehensible since the concentration is decaying with time and it is impossible to find the maximum at the beginning of iterations. Anyway, at time t_{max}, the value of concentration found is close enough to the true global maximum at the same time.

Table 4.3: The maximum found by the time-abandoned sampling method.

$f(s_{max}, t_{max})$ (mg/m^3)	s_{max} (m)	t_{max} (s)	Iteration serial	$f_{max}(t_{max})$
1140.340	[-5.87, 0.08, -19.92]	3.3	34	1145.956

Fig. 4.4 presents the collected samples by colored points. Although the optimized temporal variable is abandoned during sampling in this method, BO still has successfully find the location of global maximum as the concentration of points has revealed.

Fig. 4.5 presents a wonderful decrease of error around the global maximum. It is achieved by a continuous sampling in that area. Why BO has a persistent interest in the high concentration area than Sec. 4.1.1? We presume it is because the temporal variation brings significant change to the value of acquisition function. In the experiment, since the actual sample has a different temporal variable from the optimized sample that BO wants, the actual sample cannot gain enough knowledge to the area of high concentration. BO has to keep asking for the same

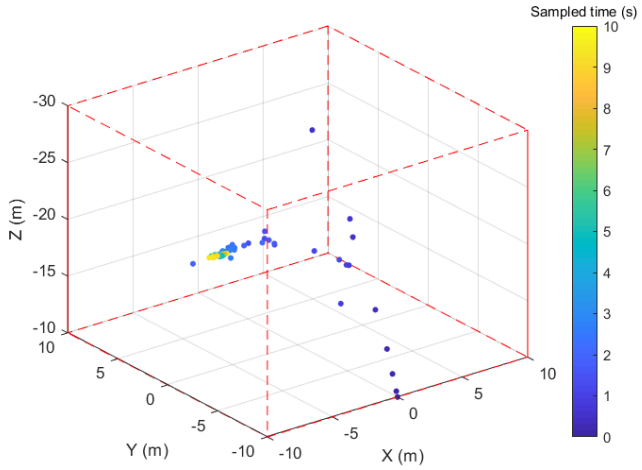
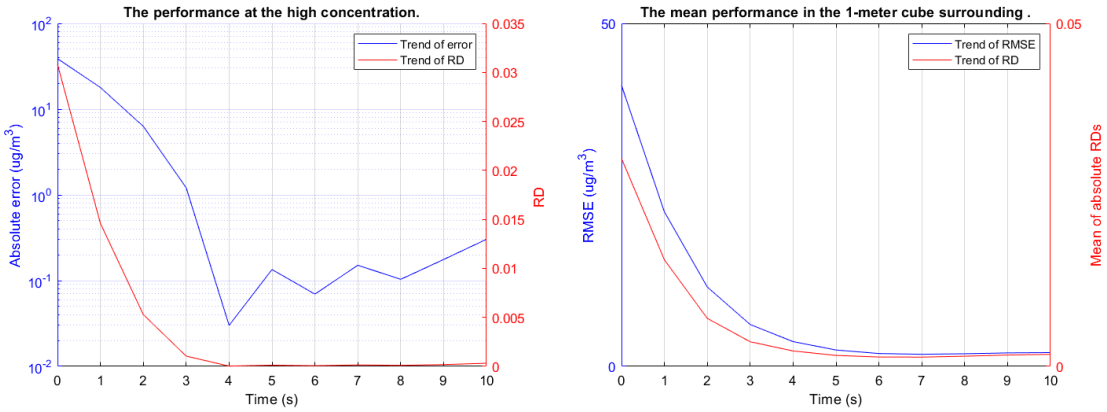


Figure 4.4: The samples collected by time-abandon sampling, colored by the time.

location at a different time whose corresponding value of acquisition function is still high.

In the meantime, as the optimized sample always locates near the global maximum, the BO losses its exploration to the rest of the room, which induces a high error in the room in Fig. 4.6. Both the regular and weighted performances of this method are worse than Sec. 4.1.1 in Fig. 4.3



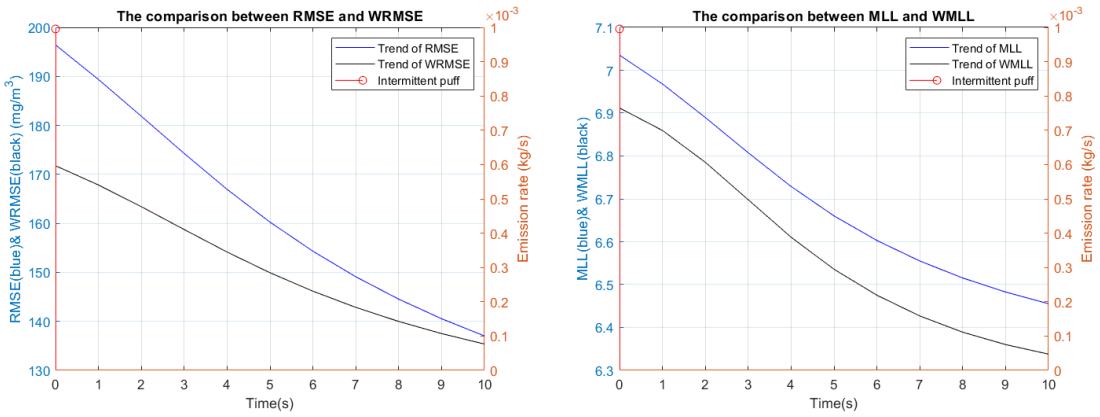
(a) Error and RD

(b) RMSE and RD

Figure 4.5: The performance (a) at and (b) around the time-varying global maximum by time-abandon sampling.

4.1.3 Ascending Order Sampling

The highest concentration found by the method is shown in Table. 4.4. This maximum is also not the temporally global maximum. But if we comparing it with Table. 4.3, this maximum



(a) (b)

Figure 4.6: The overall performance of (a) RMSE & WRMSE and (b) MLL & WMLL by time-abandon sampling.

found is closer to the global maximum at the same time. This proves the advantage of involvement of the optimized temporal variable. Since BO in this method has the ability to make use of both the optimized spatial and temporal variables in the next sample, it will be easier to obtain a true global maximum than Sec. 4.1.2.

Table 4.4: The maximum found by the ascending order sampling method.

$f(s_{max}, t_{max})$ (mg/m^3)	s_{max} (m)	t_{max} (s)	Iteration serial	$f_{max}(t_{max})$
1145.899	[-6.11, 0.01, -19.99]	3.3	42	1145.920

Fig. 4.7 presents a more dispersed distribution of points. As we bring the optimized temporal variable back in use, the BO seems less interested in the area of high concentration than in Sec.4.1.2.

However, this change has little influence on the performance around the global maximum, as Fig. 4.8 has illustrated. The error at the point of global maximum only increases from 10^{-1} to 10^{0}, and RD still stays close to zero. In addition, the overall performances in the room in Fig. 4.9 are much better than in Fig. 4.6. The weighted performances of both WRMSE and WMLL become as good as in Fig. 4.3.

The design of this method has presented its advantages on both finding the global maximum and monitoring the entire domain. However, because the bounds of spatial and temporal variables in the optimization are not well designed base on the concern that whether a mobile sensor has the ability to arrive the spatial location before the temporal variable, this method is

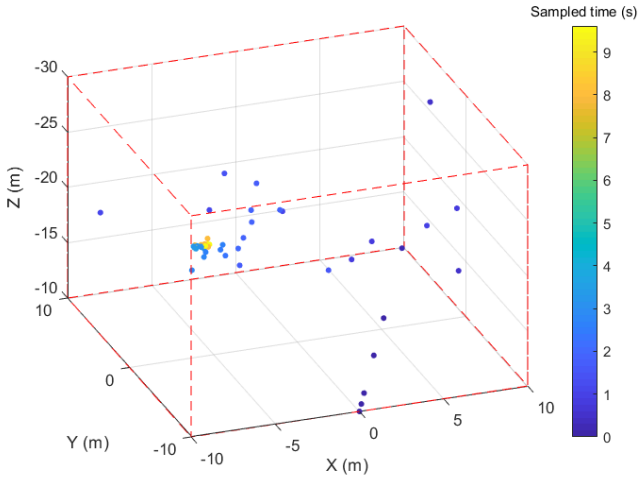
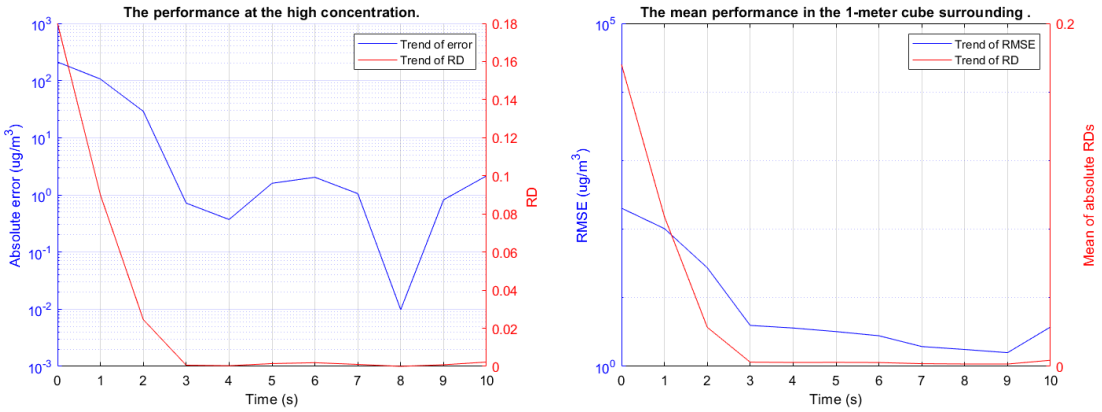


Figure 4.7: The samples collected by ascending order sampling, colored by the time.

still not practical enough for a monitoring application with UAV. A really practical result of sampling method will be presented in the following chapter.



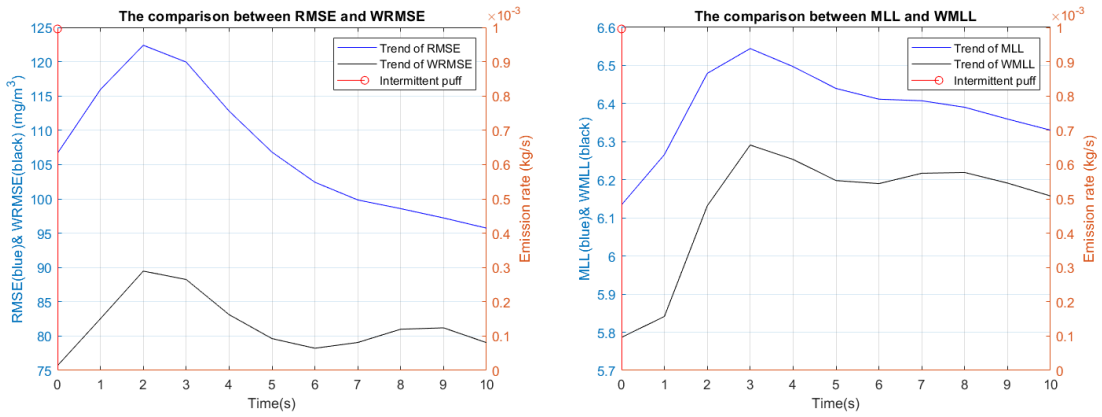
(a) Error and RD

(b) RMSE and RD

Figure 4.8: The performance (a) at and (b) around the time-varying global maximum by ascending order sampling.

4.2 Continuous Samplings

In this section, each time of iteration of BO will provide a path. And most of the samples are collected along the path instead of only at the optimized points as Sec. 4.1. In this circumstance, the expectation of potential improvement induced by the optimized sample is not dominant anymore. Learning the variation based on the samples collected on the path, in order to model the environment well both spatially and temporally, has become the main goal of this section.



(a) (b)

Figure 4.9: The overall performance of (a) RMSE & WRMSE and (b) MLL & WMLL by ascending order sampling.

The maximum point as Table. 4.2, 4.3, or 4.4 among the samples collected during simulation, will not be shown in this section. In the meantime, more data samples than Sec. 4.1 will be collected in the experiments of this section.

The settings of the experiments in Sec. 4.2 is presented in Table. 4.5.

Table 4.5: The setting of the continuous sampling experiments.

Sampling T_{end} (s)	δt (s)	Iteration i_{max}	Test interval (s)
40	0.1	400	[0, 40]

4.2.1 Waypoint Planning

As the waypoint is optimized with a similar method in Sec. 4.1.3, both spatial and temporal variables are included. Some samples are collected on the way to the waypoint with a constant rate of sampling. (0.1 s per point) The number of them are determined by the time difference between starting time and the optimized time of the waypoint.

Fig. 4.10 presents the paths obtained in this simulation. It is clear that the dark blue path goes aimlessly in the beginning. Then the high concentration is found and the paths become stationary in that area (yellow paths). Although it takes more samples before finding the area of high concentration than the experiments in Sec. 4.1, it is better because the method runs much less times of the optimization of waypoint.

The performance of this method is similar to Sec. 4.1.2, which achieves high accuracy at

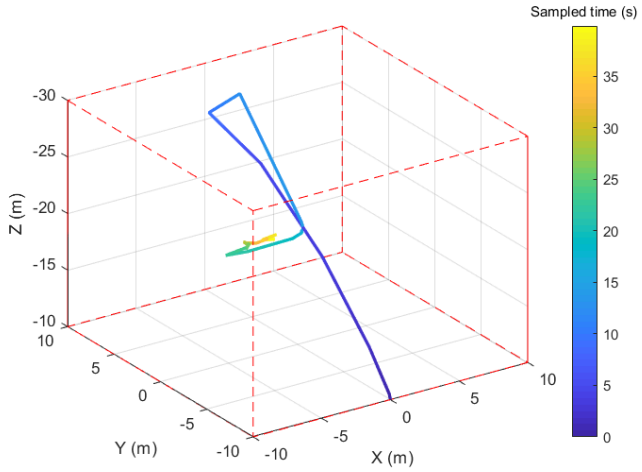
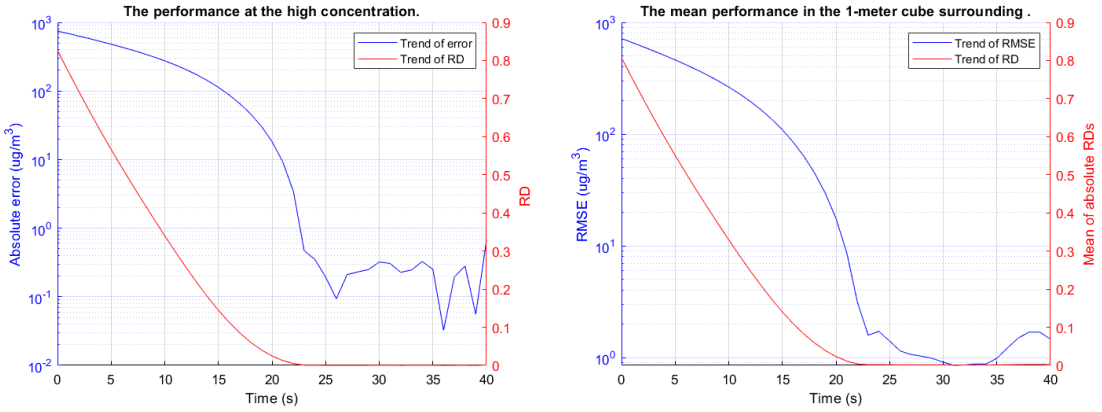


Figure 4.10: The samples collected by waypoint planning, colored by the time.

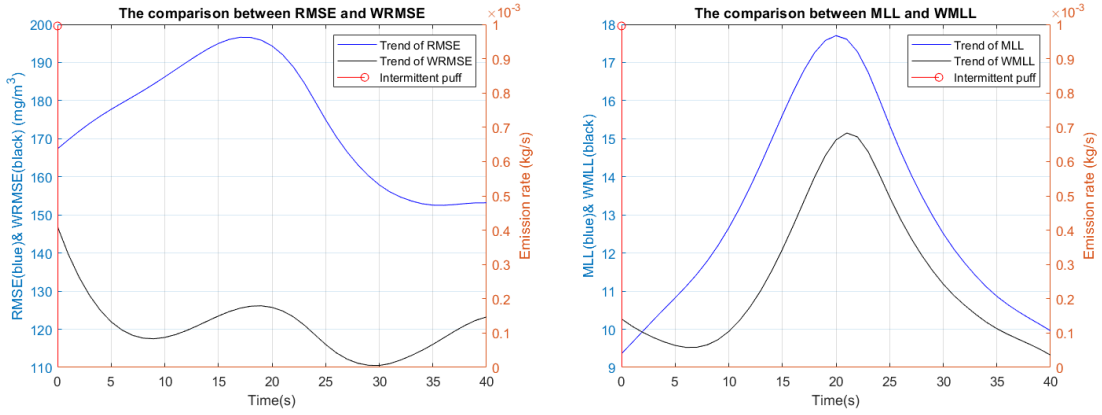
the global maximum once the global maximum is found as Fig. 4.11, while maintaining large errors in the room as Fig. 4.12. This problem is induced by the inappropriate balance between exploitation and exploration of the acquisition function. When we set $\kappa = 0.1$ in Eq. 2.30, in the environment studied, the high concentration at the maximum is always able to overwhelm the value of uncertainty in other places. As the paths planned later decide to concentrate around the global maximum, BO has lost its ability to know the rest of the room.

In a real leakage of a toxic gas, BO with the waypoint planning already has enough exploitation to fast locate the high concentration. But after that, the exploration should be preferred for a precise monitoring to the rest disaster area. To improve the ability of exploration of BO, increasing κ will be a good idea. However, the balance should be taken care of depending on the environment, because an over-large exploration will weaken the ability of finding the global optimum if the value of optimum is relatively small than the expected improvement brought by exploring the uncertainty. Balancing the exploration and exploitation is always a hard work. Another idea is to design a dynamic system to switch among several acquisition functions under certain conditions[16, 17]. The acquisition functions with high exploitation or exploration will be chosen independently to meet different requirements of BO. However, this balancing method is not further studied in this section, because the path planning proposed in next section has solved the problem.



(a) Error and RD (b) RMSE and RD

Figure 4.11: The performance (a) at and (b) around the time-varying global maximum by waypoint planning.



(a) (b)

Figure 4.12: The overall performance of (a) RMSE & WRMSE and (b) MLL & WMLL by waypoint planning.

4.2.2 Continuous Path Planning

Based on the waypoint planning, the continuous planning will not only focus on the potential improvement from the target waypoint, but also the improvement that may brought by the samples along the path. Fig. 4.13 illustrates the path designed by this method. As the figure has revealed, the path starts at one edge of the room and end at another, while pass through the center of the room where the global maximum is. Although this method implements the same acquisition function as Sec. 4.2.1, it overcomes the disadvantage by taking the reward of the path into consideration.

In Sec. 4.2.1, it has been explained that a balancing method in order to switch between

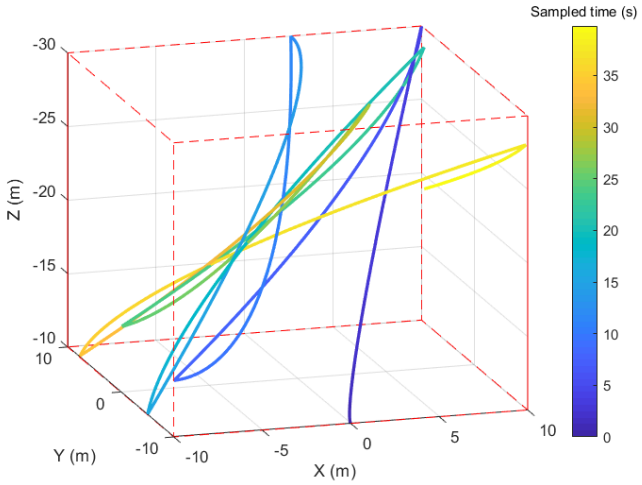
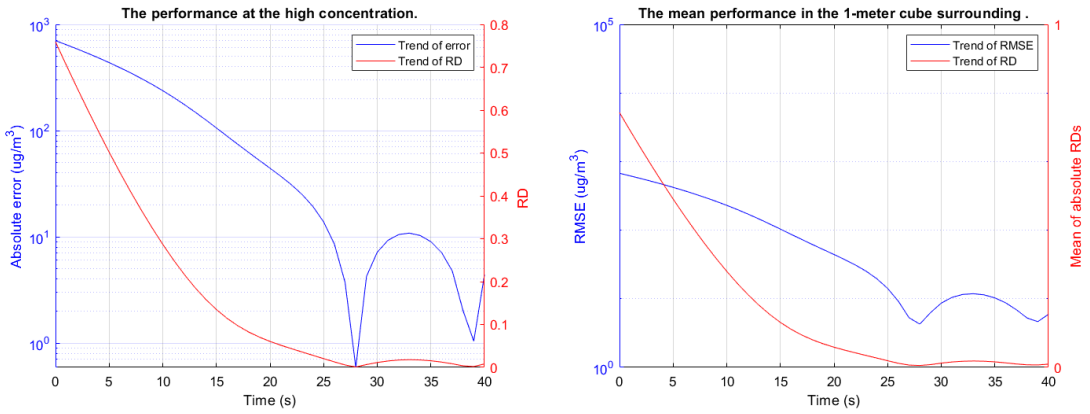


Figure 4.13: The samples collected by continuous planning, colored by the time.

exploitation and exploration is necessary to maintain an accurate monitoring of the environment at all time. But the continuous path can pay less attention on the balance, since it will try to collect reward as much as possible from both exploitation and exploration along the path as long as the value of the acquisition function is positive. Because increasing the value of integral of the reward function is the goal of optimization, the optimized curve parameters will extend the path to any location whose value of acquisition function is positive, no matter this value is based on a higher exploitation or a preference of exploration. In such a circumstance, switching from exploitation to exploration while the global maximum has been found is not essential in the continuous path planning, since the curve will eventually be optimized to explore the unknown area for a higher reward. This is a significant advantage of the continuous path planning than the previous waypoint planning.

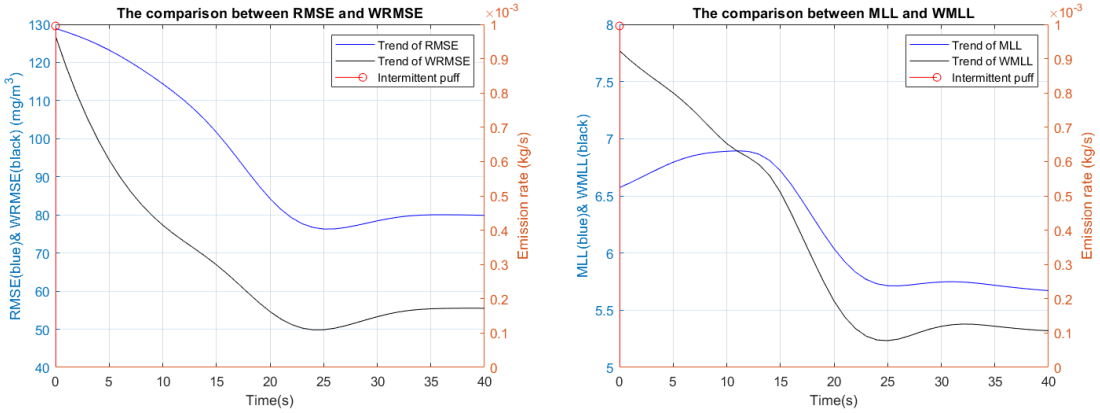
In Fig. 4.14, it takes more time for BO to reduce error and RMSE to the level of 10^1, than in Fig. 4.11. But the errors are still in an acceptable interval since the corresponding RD is close to zero. Moreover, this loss on exploitation brings great improvement on the performances in the room as Fig. 4.15 has shown. The RMSE and MLL are gradually reduced while maintaining a continuous decrease of WRMSE in Fig. 4.15 (a), which indicates the method keeps approaching the area of high concentration before finding it.

The continuous path planning has proven itself a practical and effective method in the problem of monitoring a time-varying environment with a mobile sensor. The path designed is practical for a UAV to track in a 3-D room. And the simulation results of the method have



(a) Error and RD (b) RMSE and RD

Figure 4.14: The performance (a) at and (b) around the time-varying global maximum by continuous planning.



(a) (b)

Figure 4.15: The overall performance of (a) RMSE & WRMSE and (b) MLL & WMLL by continuous planning.

presented the capability of a fast optimum location and the subsequent precise modeling to the entire domain. Comparing with the waypoint planning, the continuous path planning is more efficient on sample collection because the reward function considers the samples gathered along the path to be informative. Moreover, the continuous plan is a more reliable method since it requires less attention on balancing the trade-off between exploration and exploitation to achieve a continuously precise monitoring to the environment.

Chapter 5

Conclusion

This thesis presents a few designs of sampling methods to use Bayesian Optimization (BO) monitoring a time-varying environment. The experimental results show that with more realistic assumptions added to the design, the method is more practical but less capable of finding the true global optimum. However, a balance can be achieved between locating the particular global optimum and modeling the entire environment in the process. The core algorithm is Bayesian Optimization, a sequential design strategy for global optimization of black-box functions, particularly suitable for optimizing expensive-to-evaluate functions. The underlying regression model is Gaussian Processes (GPs). We study a case for which there is a pollutant gas leakage in an open area, which is modeled by a Gaussian puff model, a single source time-varying dispersion model. A mobile sensor is deployed in that 3-D area to locate the high concentration of leakage and to collect samples for modeling the environment.

Two sampling methods including 5 scenarios are studied. In the first discrete sampling method, three experiments are implemented corresponding to three different treatments to the optimized temporal component of the samples. In the second continuous sampling method, two kinds of path planning are implemented to collect data for environmental modeling. The continuous path planning of the second scenario is developed from the waypoint planning of the first scenario.

The results from the first method have shown that with a more realistic design of sampling, a more balanced performance can be achieved between fast locating the global optimum and precise modeling the entire environment, in terms of Root Mean Squared Error (RMSE), Weighted Root Mean Squared Error (WRMSE), Mean Log Loss (MLL), Weighted Mean Log Loss (WMLL), and Relative Difference (RD). The results of continuous path planning have

proven the method a practical design of monitoring the time-varying environment, as its algorithm can efficiently collect informative samples for the GP model and requires less work on the balance of exploration and exploitation.

Future work will be based on the extension of application of the continuous path planning. More complex environment with multi sources and multi emissions of puff will be studied in the future.

Bibliography

[1] Roman Marchant and Fabio Ramos. Bayesian optimisation for intelligent environmental monitoring. In *2012 IEEE/RSJ international conference on intelligent robots and systems*, pages 2242–2249. IEEE, 2012.

[2] Jan Gosmann and Manfred Opper. Gaussian processes for plume distribution estimation with uavs. 2013.

[3] Eric Bradford, Artur M Schweidtmann, and Alexei Lapkin. Efficient multiobjective optimization employing gaussian processes, spectral sampling and a genetic algorithm. *Journal of global optimization*, 71(2):407–438, 2018.

[4] Roman Marchant and Fabio Ramos. Bayesian optimisation for informative continuous path planning. In *2014 IEEE International Conference on Robotics and Automation (ICRA)*, pages 6136–6143. IEEE, 2014.

[5] Carl Edward Rasmussen. Gaussian processes in machine learning. In *Summer School on Machine Learning*, pages 63–71. Springer, 2003.

[6] Amarjeet Singh, Fabio Ramos, Hugh Durrant Whyte, and William J Kaiser. Modeling and decision making in spatio-temporal processes for environmental surveillance. In *2010 IEEE International Conference on Robotics and Automation*, pages 5490–5497. IEEE, 2010.

[7] Eric Brochu, Vlad M Cora, and Nando De Freitas. A tutorial on bayesian optimization of expensive cost functions, with application to active user modeling and hierarchical reinforcement learning. *arXiv preprint arXiv:1012.2599*, 2010.

[8] Favour M Nyikosa, Michael A Osborne, and Stephen J Roberts. Bayesian optimization for dynamic problems. *arXiv preprint arXiv:1803.03432*, 2018.

[9] Bobak Shahriari, Kevin Swersky, Ziyu Wang, Ryan P Adams, and Nando De Freitas. Taking the human out of the loop: A review of bayesian optimization. *Proceedings of the IEEE*, 104(1):148–175, 2015.

[10] Daniel James Lizotte. *Practical bayesian optimization*. University of Alberta, 2008.

[11] Dennis D Cox and Susan John. A statistical method for global optimization. In *[Proceedings] 1992 IEEE International Conference on Systems, Man, and Cybernetics*, pages 1241–1246. IEEE, 1992.

[12] Daniel Edwin Finkel et al. Global optimization with the direct algorithm. 2005.

[13] Find minimum of constrained nonlinear multivariable function. MathWorks Support. `https://www.mathworks.com/help/optim/ug/fmincon.html`.

[14] Renzo De Nardi. The qrsim quadrotors simulator. *RN*, 13(08):08, 2013.

[15] John M Stockie. The mathematics of atmospheric dispersion modeling. *Siam Review*, 53(2):349–372, 2011.

[16] Dillon Sterling, Tyler Sterling, YuMing Zhang, and Heping Chen. Welding parameter optimization based on gaussian process regression bayesian optimization algorithm. In *2015 IEEE international conference on automation science and engineering (CASE)*, pages 1490–1496. IEEE, 2015.

[17] Matthew D Hoffman, Eric Brochu, and Nando de Freitas. Portfolio allocation for bayesian optimization. Citeseer.