Multiobjective Optimization Based on Expensive Robotic Experiments under Heteroscedastic Noise

Ryo Ariizumi *Member, IEEE,* Matthew Tesch *Member, IEEE,* Kenta Kato, Howie Choset *Member, IEEE,* and Fumitoshi Matsuno *Member, IEEE,*

***Abstract*—In many engineering problems, including those related to robotics, optimization of the control policy for multiple conflicting criteria is required. However, this can be very challenging because of the existence of noise, which may be input dependent or heteroscedastic, and restrictions regarding the number of evaluations owing to the costliness of the experiments in terms of time and/or money. This paper presents a multiobjective optimization algorithm for expensive-to-evaluate noisy functions for robotics. We present a method for model selection between heteroscedastic and standard homoscedastic Gaussian process regression techniques to create suitable surrogate functions from noisy samples, and to find the point to be observed at the next step. This algorithm is compared against an existing multiobjective optimization algorithm, and then used to optimize the speed and head stability of the sidewinding gait of a snake robot.**

***Index Terms*—Learning and adaptive systems, multiobjective optimization, response surface method**

I. Introduction

THE optimization of multiple conflicting objectives, or multiobjective optimization (MOO), is commonly used for problems in many fields, including engineering [1], economics and finance [2]. In robotics, trade-offs between locomotion speed and energy efficiency appear very often. Other examples include maximizing the stability of a camera mounted on the head of a snake robot and its locomotion speed [3], maximization of the speed and stability of a quadruped robot [4], and the minimization of energy consumption and torque change in humanoid robots [5]. For such problems, one solution is to introduce a set of scalars that expresses relative weights, or preferences, among objectives. The MOO problem can then be converted into a single-objective optimization problem by aggregating the multiple objectives into a single one using an aggregation method, such as linear combination or Tchebycheff aggregation.

If the user does not have a clear preference among the objectives, the better option would be to find the entire set of the 'best' trade-off solutions, those which cannot be improved in any objective without becoming worse in at least one other objective. These solutions are called noninferior or Pareto optimal (Fig. 1). In addition, if a preference is given after generating the Pareto set, we can seek a compromise solution from the approximate Pareto set at hand.

In cases where the analytic form of the objective functions and the mathematical model of the system are available, the problem may be solved by searching for points that satisfy the Karush–Kuhn–Tucker condition [6], or by solving the Hamilton–Jacobi–Bellman equation [7]. Another popular approach is to use evolutionary algorithms (EAs) [8], because they enable solutions to be searched for in multiple directions simultaneously. If samples are cheap and the parameter and objective spaces are very high dimensional, EAs are an effective MOO algorithm.

However, in some robotic and other engineering problems, objective functions are accessible only through experiments and EA-based algorithms are therefore hard to apply. The challenges in these cases are twofold: first, since the observations are costly in terms of time and/or money, the number of observations must be severely limited; and second, noise in the observations makes it difficult to extract useful information from the samples. There are only a few, if any, methods that can be used in such cases; This likely prevents robotic researchers from using MOO methods.

One promising strategy for expensive MOO is the *response surface method* (RSM). In this type of algorithm, surrogate functions are constructed to fit the samples. These surrogates are then used, in place of the unknown true objective functions, to plan efficient experiments by balancing exploration and exploitation. In [9], the authors used RSM to design the path of a mobile robot to monitor environments intelligently by making use of noisy samples, and to this end, proposed an extension of the upper confidence bound. A detailed explanation of RSM for single-objective optimization can be found in [10]. The efficient global optimization algorithm [11], an RSM-based single-objective optimization method, is extended to multiobjective optimization based on the aggregation method in [12], [13]. Emmerich et al. [14] suggested using response surfaces to assist EAs, and proposed the *expected improvement in hypervolume* (EIHV) as the ranking criterion. In [15], they applied a similar approach to [14], but used the lower confidence bound of the improvement. In [16], the input to be evaluated on the next step is planned using different metrics, including approximate EIHV, and selecting four or five points in each step. In [17], the authors proposed using expected maximin fitness improvement, whose analytical form was also given for the 2-input case. In [18], another statistical measure based on the theory of random closed sets is proposed.

In terms of dealing with noise, most of the existing MOO methods have been evaluated for noiseless observations as in [12]. In the EA community, Teich [19] introduced the con-

R. Ariizumi is with the Department of Mechanical Science and Engineering, Graduate School of Engineering, Nagoya University, Nagoya, 464-8603, Japan, (email: ariizumi@nuem.nagoya-u.ac.jp).

K. Kato, and F. Matsuno are with the Department of Mechanical Engineering and Science, Graduate School of Engineering, Kyoto University, Kyoto, 606-8501, Japan, (email: kuromahat@gmail.com; matsuno@me.kyoto-u.ac.jp).

M. Tesch and H. Choset are with the Robotics Institute, Carnegie Mellon University, Pittsburgh, PA 15213, USA, (email: {mtesch, choset }@cs.cmu.edu).

cept of *probability of dominance* into a well-known EA-based MOO, the strength Pareto evolutionary algorithm (SPEA) [20], to make it robust to noise. Büche et al. [21] also extended SPEA to be robust to noise and outliers, and implemented it in optimizing the combustion process of a gas turbine. However, these methods cannot be used for expensive optimization as they require a considerable number of samples. Eskandari et al. [22] proposed the stochastic Pareto genetic algorithm, which is an extension of FastPGA [23], an EA for expensive MOO. However, their method also depends on empirical means and variances that require multiple evaluations for each input, which is not suitable for the optimization of expensive objectives. Fieldsend et al. [24] proposed the rolling tide evolutionary algorithm, which can handle noise that varies in time or space; however, this method requires too many evaluations to be used in robotic experiments.

In the single-objective optimization of robots, the problem of noise has been addressed using RSM with Gaussian process (GP) regression [25], modeling uncertainty induced by observation noise [26]-[28]. Zuluaga et al. [29] proposed using GP regression, and the response surface to determine whether a point is Pareto optimal. Independently from this, Tesch et al. [3] proposed using EIHV [14], [30] calculated based on the results of GP regression. Noiseless numerical examples exhibited the superiority of using EIHV over the aggregation function-based method proposed in [12]. However, the performance remained insufficient for noisy robotic experiments using a snake robot, and the authors had to take a mean of five runs per input to get a reliable result. One possible reason, aside from the overly large noise variance, is that the occurrence of homoscedastic noise (i.e., input-independent noise) was assumed in their work. As is verified later in this paper, the noise in robotic experiments is often not homoscedastic, and neglecting this sometimes results in poor function estimation. If the properties of the noise are known *a priori*, this knowledge may be able to be coded into the kernel function of the GP regression model. However, since our target is the optimization of expensive functions, we cannot expect accurate prior knowledge. Therefore, we need a more flexible framework. Other possible factors may include the existence of non-Gaussian noise, and the difference in the rate of change of the true objective functions (non-stationarity). There is some literature [31], [32] that deals with the non-stationarity problem; however, dealing with this problem is out of the scope of this paper.

To take input-dependent noise into account, heteroscedastic GP regression should be used in RSM. However, this kind of regression presents a difficult challenge because there is no analytical solution, and has been discussed extensively in the machine learning community [33], [34]. Among the available methods, we chose to use variational heteroscedastic Gaussian process (VHGP) regression [35] as it gives a reasonable result with a relatively small computation. VHGP regression has already been used for RSM in the context of single-objective optimization of a robot [36], but to our knowledge, we are the first to use this in MOO. In [32], treed GP regression is used to make the model more flexible. Although this method would be applicable to problems with heteroscedastic noise with relatively small calculation cost, some information would be lost by partitioning the search space into subregions, and training an independent GP regressor on each of them. Therefore, it would be better if we can do without partitioning the search area, in the case where the number of experiments are strictly restricted. Also, note that there are other ways to deal with the difficulty of noise that is not modeled as homoscedastic Gaussian noise. In [37], the authors introduce a hyperprior on a hyperparameter of the kernel function. This can attenuate the effect of unmodeled noise and outliers. In [32], [38], the authors use Student's t distribution, which is known to be more robust to outliers than the Gaussian distribution.

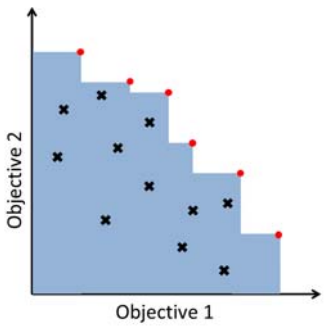


Fig. 1. Pareto-optimal points in objective function space in the case of bi-objective maximization. Points denoted by a black × mark are dominated by points with a red circle.

In this paper we propose a MOO method for expensive noisy objectives, particularly those with input-dependent noise. This method uses two GP regression methods to make surrogate functions and plan the best experiment based on EIHV. These GP regression methods enable us to make good surrogates from the data with input-dependent noise; however, the calculation of EIHV and model selection between these two are problematic, because in the heteroscedastic case, the predictive density is not Gaussian and is therefore analytically intractable. In this paper, the approximation of EIHV with reasonable calculation cost, and a novel method to determine which regression method to use at every step are also discussed. The effectiveness of the method is shown by numerical tests and robotic experiments. The contents of this paper partially appeared in [39]. Compared with our previous paper, this work includes further numerical verification of the EIHV approximation, the lack of which was the primary weak point in [39]. Moreover, additional numerical verifications are included, which make the efficacy and limitations of the proposed method clear. We also conduct new sets of robotic experiments with a different robot to show the efficacy of the method in actual robotic problems.

The remainder of this paper is organized as follows. In Section II, the algorithm is explained in detail. In Section III and IV, numerical and experimental validations are provided. Section V concludes the paper.

II. Proposed Algorithm

The problem we focus on in this paper is formulated as follows:

$$\begin{aligned} &\underset{\mathbf{x}}{\text{maximize}}\ \mathbf{f}(\mathbf{x}) \\ &\text{sub. to } x_i \in [x_{i\min}, x_{i\max}] \end{aligned} \tag{1}$$

The objective function $\mathbf{f}$ is a vector-valued function, each of whose elements corresponds to one objective. We assume that on the observation of $\mathbf{f}$, observation noise exists, whose level may vary according to the input $\mathbf{x}$. We further assume that the number of observations of $\mathbf{f}$ is severely limited, given that this method requires significant time and/or money to observe its value, as is often the case with robotic experiments. Although constraints are important in optimization, we do not consider any constraints other than that the input vector should be in a box region. Taking constraints into consideration is one of our directions for future work.

The proposed algorithm is shown in Algorithm 1. Lines 9 and 14 are the primary contributions of this paper.

In Line 2, experimental planning for initial evaluations is made through a Latin hypercube [40], which is suitable for GP regression as suggested in [11]. Note that because GP regression is conducted for each objective independent of others, the choice of a suitable experimental design method does not depend on the number of objectives. In Line 8, to make the algorithm robust to observation noise which may be heteroscedastic, we use both standard GP regression and heteroscedastic GP regression. For the heteroscedastic case, we use VHGP regression [35]. This method seeks the maximum of the lower bound of the marginal likelihood instead of the marginal likelihood itself, which is analytically intractable in the heteroscedastic setup.

However, the difficulty of calculating the marginal likelihood causes another problem, because it is usually used not only to tune hyperparameters, but also to select among different types of kernel functions (models). In this paper, we propose a novel model selection method based on leave-one-out (LOO) cross validation (Line 9) for deciding which GP – standard or heteroscedastic – to use. This method is especially efficient in cases where the sample size is small (less than about 40). In addition, because the resultant predictive distribution in the heteroscedastic case is not Gaussian, EIHV used at Line 14 also does not have any closed form if heteroscedastic regression is selected. Although the approximation can be calculated by Gauss–Hermite quadrature when the number of objectives is small, this is computationally very expensive and can make the procedure prohibitively slow. In this paper, an alternative approximation that is computationally much less expensive is proposed in Section II-B.

The lines 11–13 mean that the procedure is terminated if the maximum of EIHV becomes less than 100ϵ % of the current hypervolume, where $\epsilon \ll 1$ is a user-given positive value. It is important to note that because the goodness of the model is not taken into consideration in the calculation of EIHV, this can lead to premature termination in cases where the model is poor but confident in its prediction, especially at the beginning of the optimization. Nonetheless, we found this to be a very rare occurrence, and we in fact did not observe it in our numerical/experimental tests.

Algorithm 1 Proposed Algorithm

1: Given
   N_{i}: the number of initial sample points
   N_{m}: the maximum number of experiments
   $\mathbf{f}$: the vector-valued objective function evaluated through expensive experiments/simulations
   ϵ: a small positive constant
   X: set of inputs at which $\mathbf{f}$ is evaluated
   Y: set of observed objective function values
   ref: the user-defined reference for hypervolume (HV)
2: $X \leftarrow \{\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_{N_i}\}$: use Latin hypercube design
3: **for** $k = 1$ to N_{i} **do**
4: $\quad \mathbf{y} \leftarrow \mathbf{f}(\mathbf{x}_k)$
5: $\quad X \leftarrow \{Y, \mathbf{y}\}$
6: **end for**
7: **for** $j = 1$ to $N_{\mathrm{m}} - N_{\mathrm{i}}$ **do**
8: $\quad$ Perform regression (Section II-A)
9: $\quad$ Model selection between Standard GP and VHGP models (Section II-C)
10: $\quad$ Search for max EIHV (Section II-A)
11: $\quad$ **if** $\max \mathrm{EIHV} < \epsilon \mathrm{HV}$ **then**
12: $\quad\quad$ return X, Y
13: $\quad$ **end if**
14: $\quad \mathbf{x}_{\mathrm{new}} \leftarrow \mathrm{argmax}_{\mathbf{x}} \mathrm{EIHV}(\mathbf{x}|\mathrm{ref}, Y)$ (Section II-B)
15: $\quad \mathbf{y} \leftarrow \mathbf{f}(\mathbf{x}_{\mathrm{new}})$
16: $\quad X \leftarrow \{X, \mathbf{x}_{\mathrm{new}}\}$, $Y \leftarrow \{Y, \mathbf{y}\}$
17: **end for**
18: return X, Y

After making another observation, add the result to the data set (Line 16) and, if the budget has not been completely consumed, go back to Line 8. The return can be the approximated Pareto set and Pareto front, instead of the whole set of inputs and outputs.

In Section II-A, VHGP used in Line 8, and 10, which is used to handle heteroscedastic noise in samples, is explained. Line 14 is explained in Section II-B, and Line 9 in Section II-C.

A. Regression Method

In this research, VHGP regression [35] and standard GP regression is used. In this subsection, we review both of the regression methods. The tuning of hyperparameters corresponds to Line 8 of Algorithm 1, and the calculation of the predictive distribution (5), (10) and (11) to Line 10. As a surrogate function, the mean of the predictive distribution is used.

1) Standard Gaussian Process Regression [25]: Here we briefly review standard, i.e., homoscedastic, GP regression.

Consider the case where we are trying to fit the data $\mathcal{D} = \{(\mathbf{x}_1,\ y_1),\ \cdots,\ (\mathbf{x}_N,\ y_N)\}$ by a function f. The input vector space is assumed to be a subset of $\mathbb{R}^D$, the output space is $\mathbb{R}$, and $\mathbf{y}$ and $\mathbf{f}$ are defined as the vectors whose ith components are y_i and $f_i = f(\mathbf{x}_i)$, respectively. In standard GP regression,

we assume the following:

$$\begin{aligned} y &= f(\mathbf{x}) + \epsilon \\ f(\mathbf{x}) &\sim \mathcal{GP}(m(\mathbf{x}), k_f(\mathbf{x}, \mathbf{x}')) \\ \epsilon &\sim \mathcal{N}(0, \sigma_n^2), \end{aligned} \tag{2}$$

where $x \sim \mathcal{P}$ means that a random variable x is taken from a distribution (or a stochastic process) $\mathcal{P}$; ϵ is a noise term assumed to be taken independently from the same Gaussian distribution regardless of the input $\mathbf{x}$; and k_f is a user-defined kernel function that expresses our prior knowledge of the latent function. The mean function $m(x)$ is set to be $m(x) \equiv 0$ to make the following calculation concise; however, in real applications, this will also be used to code our prior knowledge.

One of the most frequently used kernel functions is the squared exponential kernel (SE kernel, Gaussian kernel):

$$k_f(\mathbf{x}_i, \mathbf{x}_j) = \sigma_f^2 \exp\left\{-\frac{1}{2}(\mathbf{x}_i - \mathbf{x}_j)^T M (\mathbf{x}_i - \mathbf{x}_j)\right\}, \tag{3}$$

where $M = \mathrm{diag}(l_1^{-2}, \ \cdots, \ l_D^{-2})$. Parameters σ_n, σ_f and l_i $(i = 1, \ \cdots, \ D)$ are called the hyperparameters and should be tuned based on the samples.

To tune the hyperparameters, we maximize the marginal likelihood or evidence:

$$p(\mathbf{y}) = \int p(\mathbf{y}|\mathbf{f}) p(\mathbf{f}) d\mathbf{f} = \int \mathcal{N}(\mathbf{y}|\mathbf{f}, \sigma_n^2 I) \mathcal{N}(\mathbf{f}|\mathbf{0}, K) d\mathbf{f}, \tag{4}$$

where I is the identity matrix of dimension N, and K is a kernel matrix whose (i, j) component is $k(\mathbf{x}_i, \ \mathbf{x}_j)$. In the standard setting, the marginal likelihood and its gradient can be calculated analytically.

Once the hyperparameters are determined through the method of maximum marginal likelihood, then the predictive distribution of y_* at an unknown point $\mathbf{x}_*$ is

$$y_* | \mathbf{x}_* \sim \mathcal{N}(\mathbf{k}_*^T (K + \sigma_n^2 I)^{-1} \mathbf{y}, k_{**} - \mathbf{k}_*^T (K + \sigma_n^2 I)^{-1} \mathbf{k}_*) \tag{5}$$

where $\mathbf{k}_* = [k(\mathbf{x}_1, \mathbf{x}_*) \ \cdots \ k(\mathbf{x}_N, \mathbf{x}_*)]^T$ and $k_{**} = k(\mathbf{x}_*, \mathbf{x}_*)$.

2) Variational Heteroscedastic Gaussian Process Regression [35]: Consider the case with input-dependent noise:

$$\begin{aligned} y &= f(\mathbf{x}) + \epsilon(\mathbf{x}) \\ f(\mathbf{x}) &\sim \mathcal{GP}(m(\mathbf{x}), k_f(\mathbf{x}, \mathbf{x}')) \\ \epsilon(\mathbf{x}) &\sim \mathcal{N}(0, e^{g(\mathbf{x})}) \\ g(\mathbf{x}) &\sim \mathcal{GP}(\mu_0, k_g(\mathbf{x}, \mathbf{x}')), \end{aligned} \tag{6}$$

where f is the latent objective function, g is the latent log noise variance, and k_f and k_g are user-defined kernel functions that express our prior knowledge of latent functions, defined in the same manner as in standard GP regression. The main point of this modeling is that it assumes that the noise level must also be determined by a GP. If the noise is not dependent on the input $\mathbf{x}$ and it can be written as $\epsilon(\mathbf{x}) = \epsilon \ (\mathrm{const.})$, then this model is the same as that of standard GP regression. In the heteroscedastic case, the marginal likelihood:

$$\begin{aligned} p(\mathbf{y}) &= \iint p(\mathbf{y}|\mathbf{f}, \mathbf{g}) p(\mathbf{f}|\mathbf{g}) p(\mathbf{g}) d\mathbf{f} d\mathbf{g} \\ &= \iint \mathcal{N}(\mathbf{y}|\mathbf{f}, \mathrm{diag}(e^{g_1}, \ \cdots, \ e^{g_n})) \\ &\qquad \times \mathcal{N}(\mathbf{f}|\mathbf{0}, K_f) \mathcal{N}(\mathbf{g}|\mu_0 \mathbf{1}, K_g) d\mathbf{f} d\mathbf{g}, \end{aligned} \tag{7}$$

which indicates that our confidence in our regression is not analytically tractable, making it difficult to tune the hyperparameters. To optimize the hyperparameters in this case, VHGP regression maximizes the variational lower bound on the marginal likelihood instead of marginal likelihood, with respect to the variational parameters and the hyperparameters.

Define a function F as follows:

$$F(q(\mathbf{f}), q(\mathbf{g})) = \log p(\mathbf{y}) - \mathrm{KL}(q(\mathbf{f}) q(\mathbf{g}) \| p(\mathbf{f}, \mathbf{g}|\mathbf{y})), \tag{8}$$

where $q(\mathbf{f})$ and $q(\mathbf{g})$ are the variational probability densities, and $\mathrm{KL}(\cdot\|\cdot)$ is the Kullback–Leibler (KL) divergence. Because KL divergence is non-negative, F gives the lower bound of the logarithm of the marginal likelihood $p(\mathbf{y})$. Therefore, we maximize F instead of the marginal likelihood. To obtain the maximization of F, the dependency on $q(\mathbf{f})$ can be eliminated by assuming that $q(\mathbf{g})$ is fixed, and using the variational principle as the first step. This results in the optimal $q(\mathbf{f})$ as a function of $q(\mathbf{g})$, and by substituting it back into F, F is transformed into what is called the marginalized variational bound:

$$F(q(\mathbf{g})) = \log Z(q(\mathbf{g})) - \mathrm{KL}(q(\mathbf{g}) \| p(\mathbf{g})), \tag{9}$$

where $Z(q(\mathbf{g}))$ is the normalizing constant of the optimal $q(\mathbf{f})$. This bound can be computed in closed form if $q(\mathbf{g})$ is restricted to be $q(\mathbf{g}) = \mathcal{N}(\mathbf{g}|\boldsymbol{\mu}, \Sigma)$. Furthermore, it can be shown that from the stationary equations $\partial F / \partial \boldsymbol{\mu} = 0$ and $\partial F / \partial \Sigma = 0$, $\boldsymbol{\mu}$ and Σ reduce to be a function of a common n-by-n diagonal matrix Λ. Therefore, F needs to be maximized with respect to these n parameters, i.e., the diagonal elements of Λ. Simultaneously, F can be maximized with respect to the model hyperparameters. These optimizations can be solved by, for example, the conjugate gradient method.

From the maximization of the lower bound of the marginal likelihood, a variational predictive density will be obtained:

$$\begin{aligned} q(y_*|\mathbf{x}_*) &= \iint p(y_*|g_*, f_*) q(f_*|\mathbf{x}_*) q(g_*|\mathbf{x}_*) df_* dg_* \\ &= \int \mathcal{N}(y_*|a_*, c_*^2 + e^{g_*}) \mathcal{N}(g_*|\mu_*, \sigma_*^2) dg_*, \end{aligned} \tag{10}$$

where a_*, c_*, μ_* and σ_* are determined by the kernel function, the new input $\mathbf{x}_*$, and the training data $\mathcal{D}$. Note that, though the predictive distributions of f_* and g_* are Gaussian, the resultant predictive distribution of y_* is not Gaussian and is analytically intractable. However, the mean and variance can be computed analytically:

$$\mathbb{E}_q[y_*|\mathbf{x}_*, \mathcal{D}] = a_*, \quad \mathbb{V}_q[y_*|\mathbf{x}_*, \mathcal{D}] = c_*^2 + e^{\mu_* + \sigma_*^2/2}. \tag{11}$$

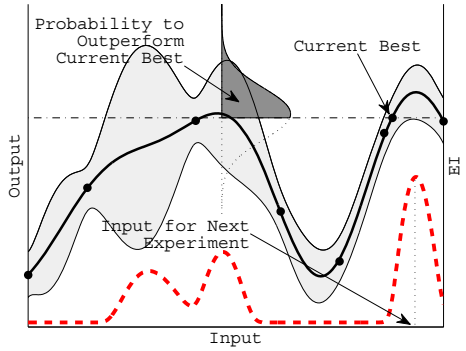


Fig. 2. A surrogate function (thick line) interpolating the sampled points of an unknown underlying function with estimated uncertainty (three sigma interval) shown in the gray area, and the expected improvement for single-objective maximization (thick dashed line).

B. Expected Improvement in Hypervolume

Expected Improvement (EI) is a popular statistical measure to make an efficient experimental plan for the next step, which automatically balances the trade-off between exploration and exploitation without requiring a tuning parameter. To define EI, we first have to define the improvement.

In the single-objective case, the improvement at $\mathbf{x}$ with the value y is the increase in the maximum sampled target value. The expectation of improvement is

$$EI(\mathbf{x}) = \int_{\max(\tilde{Y})}^{\infty} (y - \max(\tilde{Y}))p(y|\mathbf{x})dy, \tag{12}$$

where $\tilde{Y}$ is the set of sampled target values. Figure 2 illustrates the concept of expected improvement. The dark colored area represents the probability for the sample at the point to give a better result than the current best one. Since EI not only considers the probability that this point is better but also by how much, the point with the highest probability to outperform the current best point does not always have the highest EI value. In general, between two points with equal predictive mean, higher predictive variance implies higher EI, and two points with equal predictive variance, higher predictive mean implies higher EI. In this figure, the rightmost peak of the EI corresponds to its maximum, and therefore the input that attains this will be used for the next experiment. In the case where the predictive distribution $p(y)$ is Gaussian, the analytic form of EI can be obtained [11].

In MOO, because the solution is not a single point but a whole Pareto set of points, the improvement must capture the change in the quality of this set. One metric that expresses the quality of the set of solutions is the set's *hypervolume* [41]. This is the volume in objective space that is Pareto-dominated by at least one point in the Pareto subset of the set in question, at the same time dominating a user-defined reference point, which basically defines the lower bounds of objective values.

Let $HV(A)$ be the hypervolume of a set A; the improvement in the case where the output of m objective functions is $y \in \mathbb{R}^m$ can then be defined as

$$I(y) = HV(\tilde{Y} \cup y) - HV(\tilde{Y}). \tag{13}$$

If the predictive density is Gaussian, the closed form of the EIHV is given by Emmerich et al. [30]. In the case where the predictive density is expressed as (10), the EIHV will be

$$\begin{aligned} EI(\mathbf{x}) &= \iint I(y)p(y|g,\mathbf{x})p(g|\mathbf{x})dydg \\ &= \int EI(\mathbf{x}|g)p(g|\mathbf{x})dg, \end{aligned} \tag{14}$$

where

$$\begin{aligned} EI(\mathbf{x}|g) &= \int I(y)p(y|g,\mathbf{x})dy \\ &= \int I(y)\mathcal{N}(y|a,\Sigma)dy. \\ \Sigma &= \mathrm{diag}(c_1^2 + e^{g_1},\ \cdots,\ c_m^2 + e^{g_m}), \end{aligned} \tag{15}$$

where c_k and g_k correspond to the kth objective. For $EI(\mathbf{x}|g)$, the closed form derived in [30] can be used, and because $p(g|\mathbf{x})$ is a Gaussian density function, $EI(\mathbf{x})$ can be calculated numerically by Gauss–Hermite quadrature if the number of objectives m is small. However, even with the closed form [30], the calculation of EIHV is still time consuming, and the following approximation of EIHV gives equivalent or better results with much less computation, as will be shown by numerical examples in Section III-E2:

$$\bar{EI}(\mathbf{x}) = \int I(y)\mathcal{N}(y|a,\bar{\Sigma})dy, \tag{16}$$

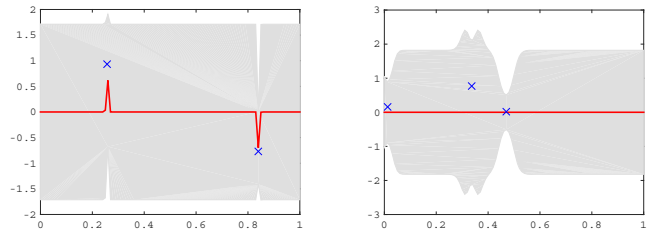
where

$$\bar{\Sigma} = \mathrm{diag}(c_1^2 + e^{\mu_1+\sigma_1^2/2},\ \cdots,\ c_m^2 + e^{\mu_m+\sigma_m^2/2}). \tag{17}$$

This is given by approximating the predictive density (10) by a Gaussian distribution with the same mean and variance as the true density calculated in (11), and can be calculated by the formula in [30]. In the limit of $\sigma \to 0$ (i.e., in the limit of no uncertainty in noise variance), (16) tends to be identical to (14). Therefore, (16) is expected to give a good approximation in the case where σ is small compared with $|\mu|$. Note that the value of EIHV itself is not so important for our purpose because we need only the maximizer of EIHV and not EIHV itself. Although this approximation may not be as accurate as Gauss–Hermite quadrature, numerical examples in Section III-E2 show that the discrepancy becomes small at the neighbor of the maximum of EIHV, which implies that this approximation is sufficient for experiment planning.

C. Model Selection

Because in most cases we have little prior knowledge of the objective functions, we have to choose the best prior distribution or model from multiple candidates. In particular, selection between the standard GP model and the HGP model is important. As the HGP model is more complex, it is more likely to overfit than the standard GP model. The problem is not only that the HGP model is more prone to overfitting, but also that this model sometimes results in a more problematic kind of overfitting. There are two kinds of overfitting, the second of which is specific to HGP regression.



(a) Common to both regression (b) Special to HGP regression

Fig. 3. Two types of overfitting. The crosses are samples, the solid lines are mean functions, and the colored areas show 2-σ intervals.

(i) The predictive mean is flat, except the vicinity of the points on which the samples are drawn, and the predictive variance becomes small only around the sampled points.
(ii) The predictive mean is flat even around the sampled points, and the predictive variance is large around the sampled points.

In Fig. 3, we show typical examples of both kinds of overfitting. The crosses are samples, the solid lines are predictive mean functions, and the colored areas illustrate predictive 2-σ intervals. Both types of overfitting cause the algorithm to be inefficient, but the second type is more difficult to resolve. In the second type of overfitting, any variation from the mean is attributed to the noise by adjusting the noise variance. This leads to large EIHV values around sampled points, rather than unevaluated regions, and leads to dense, non-informative experiments at a few fixed points. The standard GP model is free from the second type of overfitting, although it is sometimes too simple to explain the data when used alone. Therefore, it is essential for our method to select between the standard GP model and the HGP model, to make it useful for black-box objective functions.

However, selection between homoscedastic and heteroscedastic models is quite challenging because of the lack of common analytically tractable metrics between these two models. In the standard GP regression setting, marginal likelihood can serve as a model selection metric, but in the heteroscedastic case it is not analytically tractable. The variational lower bound is also not a valid alternative for model selection between homoscedastic and heteroscedastic models because it is necessarily smaller than the marginal likelihood for the heteroscedastic model (though it is equal to the marginal likelihood for the homoscedastic model). To overcome this difficulty, we propose using LOO validation. Instead, we could use Monte Carlo (MC) to calculate the marginal likelihood and choose the model with the largest marginal likelihood; however, as shown in Section III, it is generally difficult to obtain good results by MC calculation. The details on MC calculation are provided in the appendix.

Note that both of the ordinary squared form of cross validation error and the standardized residual proposed by Jones et al. [11] are not suitable for our application if used alone. As the accuracies of both predictions of the mean and the uncertainty are equally important, we need a metric that has the properties of both indicators, and that is normalized in some sense not to put too much stress on only one of them.

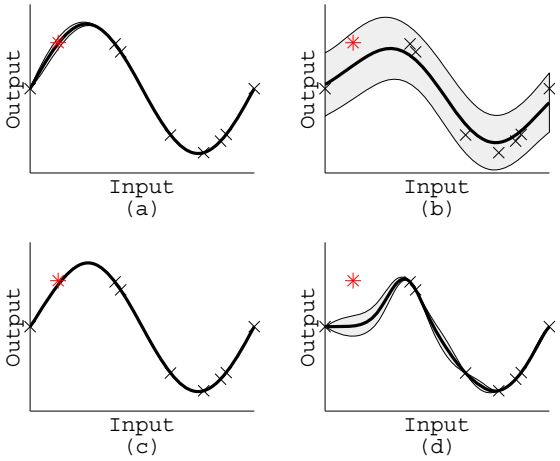


Fig. 4. Typical four patterns for regression in leave-one-out cross-validation. The red point marked by $*$ is the point that is not used in the regression, the blue line is the surrogate function, and the 3σ region is shown by the colored area. (a) Both a_i and p_i are small. (b) a_i is small but p_i is large. (c) a_i is large but p_i is small. (d) Both a_i and p_i are large.

The proposed LOO method will choose the better model between a VHGP model and a standard GP model in the following manner. Choose a point $\mathbf{x}_i$ that will serve as a test point and perform the regression without it. Let $\hat{y}_{i\backslash i}$ be the predicted function value at the training point $\mathbf{x}_i$, evaluated without using the data of point $\mathbf{x}_i$. Similarly, $\hat{\sigma}_{i\backslash i}$ is the standard deviation (standard GP) or the mean of it (VHGP) at point $\mathbf{x}_i$, evaluated without using $\mathbf{x}_i$. Let a_i and p_i be defined as follows.

$$a_i = \frac{|\hat{y}_{i\backslash i} - y_i|}{\hat{\sigma}_{i\backslash i}}, \quad p_i = |\hat{y}_{i\backslash i} - y_i|, \tag{18}$$

where $|\cdot|$ is absolute value, y_i is the sampled value at $\mathbf{x}_i$. a_i is the absolute value of the standardized cross-validated residual proposed in [11] and corresponds to the accuracy of the regression: If the regression is successful, a_i may typically be less than about 3 (the sampled point is in the 3σ interval of the prediction). p_i indicates how close the prediction is to the real value. These relations are illustrated in Fig. 4. These indicators are calculated both for standard GP regression (expressed by superscript 'std') and VHGP regression (expressed by superscript 'vh'), and the sum of their ratio is taken:

$$r_i^{(\mathrm{std})} = \frac{a_i^{(\mathrm{std})}}{a_i^{(\mathrm{vh})}} + \frac{p_i^{(\mathrm{std})}}{p_i^{(\mathrm{vh})}}, \quad r_i^{(\mathrm{vh})} = \frac{a_i^{(\mathrm{vh})}}{a_i^{(\mathrm{std})}} + \frac{p_i^{(\mathrm{vh})}}{p_i^{(\mathrm{std})}}. \tag{19}$$

If $\sum_{i=1}^n r_i^{(\mathrm{std})} \leq \sum_{i=1}^n r_i^{(\mathrm{vh})}$, standard GP regression will fit better than VHGP regression.

The computation requires that both the GP and HGP regressions are proportional to $O(N^3)$. Therefore, LOO will require $O(N^4)$ calculations. However, note that in theory, LOO can

easily be parallelized, and this will reduce the calculation time considerably. Regarding the calculation needed to find the maximizer of EIHV, it depends heavily on the properties of the constructed surrogate functions.

Another metric that would be suitable is the log pseudo-likelihood [25], which is defined as follows:

$$L_{\mathrm{LOO}} = \sum_{i=1}^{n} \log p(y_i|\mathbf{y}_{-i}, \boldsymbol{\theta}), \tag{20}$$

where $\mathbf{y}_{-i}$ is the targets except number i. In homoscedastic regression, $\boldsymbol{\theta}$ corresponds to $\mathbf{f}$, and in heteroscedastic regression to the pair of $\mathbf{f}$ and $\mathbf{g}$. The model that has larger L_{LOO}, or equivalently, smaller $-\log L_{\mathrm{LOO}}$ should be selected. Although it requires numerical integration of (10) in our case, the computational burden is comparable to our proposed metric. Detailed examination on this metric compared to ours is left to our future work, but empirically, these two metrics results in similar performances.

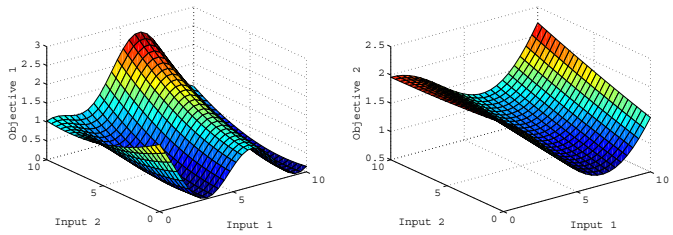
Note that some authors [38], [42] suggest that the problem of overfitting can be attenuated by restricting the intervals of hyperparameters. Although an implementation of this kind of restriction would make our method more robust, its verification is beyond the scope of this paper.

III. Numerical Tests

To test the efficacy of our method, we ran some numerical calculations using known 2-input-2-output functions: MAT, $\mathcal{T}_3$, $\mathcal{T}_4$ and $\mathcal{T}_6$, whose true hypervolumes are shown in Table I. The test function MAT was proposed in our previous research [3]. We modified the $\mathcal{T}_3$-$\mathcal{T}_6$ functions that appeared in [43] from 30-input-2-output functions into 2-input-2-output functions, and also into maximization problems instead of minimizations. $\mathcal{T}_5$ was omitted from those proposed in [43], given that it is a binary function and was out of the scope of this manuscript. In addition, as a higher order example, we use the $\mathcal{T}_3$ function again, but with 10-dimensional inputs. We also added noise on observations with known variance. In the test, we did not implement Lines 11-13 of Algorithm 1, but this process consumed the entire budget in every trial. However, in practice, both would be used to prevent extra experiments of less significance. We observed that for the optimization of the function MAT through the proposed method, $\max(EIHV)$ became smaller than $0.01HV$ within 35 iterations in more than 75 % of trials, which means that in most of the trials, the algorithm converged before the number of iterations reached the maximum, and could be stopped earlier if we also used Lines 11-13.

In this test, we only used 2-input-2-output functions for simplicity. Our method also works in higher dimensional input problems, although this will require more function evaluations than 2-input cases. In future work, we will strive to elucidate the relationship between the required number of experiments and the dimensionality of the input space.

In this section, we first describe the test functions, and then the additive noise. The performance metric for comparing MOO methods is explained in Section III-C. Settings that are commonly used throughout the section are explained in Section III-D, and the results are shown in Section III-E.



(a) Objective 1 (b) Objective 2

Fig. 5. Graph of the test function MAT

A. Test Functions

1) Test function MAT: The first test function used in this research is the one proposed in [3], which is defined as:

$$\begin{aligned} f_1(\mathbf{x}) &= f_1(x_1, x_2) = B(x_1, 2 + 0.5x_2)/20, \\ f_2(\mathbf{x}) &= f_2(x_1, x_2) = B(0.4x_1, 5 + 0.1x_2)/10, \end{aligned} \tag{21}$$

where the domain is $[0, 10] \times [0, 10]$, and $B(x_1, x_2)$ is the Branin function [44]. The reference point for hypervolume calculation is taken to be $[0, 0]$. The graphs of the functions are shown in Fig. 5.

2) Test function $\mathcal{T}_3$: The test function $\mathcal{T}_3$ used in this research is defined as follows:

$$\begin{aligned} &f_1(x_1, x_2, ..., x_D) = -x_1 \\ &f_2(x_1, x_2, ..., x_D) = -g\left\{1 - \sqrt{-f_1/g} + f_1 \sin(-10\pi f_1)/g\right\} \\ &g(x_2, x_3, ..., x_D) = 1 + 9\sum_{i=2}^{D} x_i/(D-1) \end{aligned} \tag{22}$$

where D is the dimensionality of the search space. The domain is $[0, 1]^D$. The graphs of the functions in the 2-D case are shown in Fig. 6. The reference point for the hypervolume calculation is taken to be $[-1, -10]$. Note that the Pareto front and the true hypervolume are invariant to D.

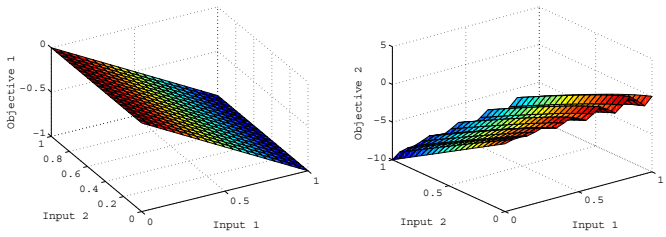
3) Test function $\mathcal{T}_4$: The test function $\mathcal{T}_4$ used in this research is defined as follows:

$$\begin{aligned} f_1(x_1, x_2) &= -x_1 \\ f_2(x_1, x_2) &= -g\left(1 - \sqrt{-f_1/g}\right) \\ g(x_2) &= 11 + x_2^2 - 10\cos(4\pi x_2) \end{aligned} \tag{23}$$

The domain is $[0, 1] \times [-5, 5]$. The graphs of the functions are shown in Fig. 7. The reference point for hypervolume calculation is taken to be $[-1, -45]$.

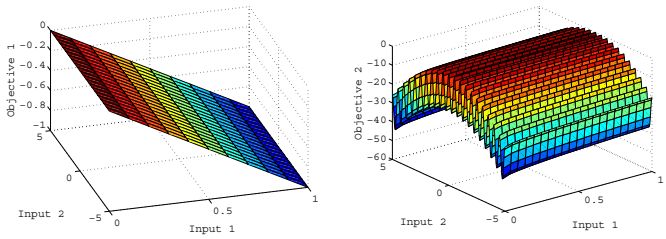
TABLE I
True Hypervolume

	MAT	$\mathcal{T}_3$	$\mathcal{T}_4$	$\mathcal{T}_6$
True HV	5.1013	10.0444	44.6667	6.7989



(a) Objective 1 (b) Objective 2

Fig. 6. Graph of the test function $\mathcal{T}_3$



(a) Objective 1 (b) Objective 2

Fig. 7. Graph of the test function $\mathcal{T}_4$

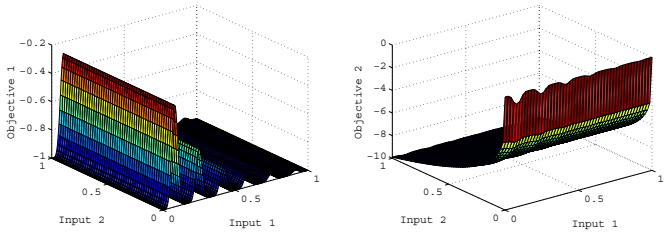
4) Test function $\mathcal{T}_6$: The test function $\mathcal{T}_6$ used in this research is defined as follows:

$$\begin{aligned} f_1(x_1, x_2) &= -1 + \exp(-4x_1)\sin^6(6\pi x_1) \\ f_2(x_1, x_2) &= -g\left\{1 - (f_1/g)^2\right\} \\ g(x_2) &= 1 + 9x_2^{\frac{1}{4}} \end{aligned} \tag{24}$$

The domain is $[0, 1] \times [0, 1]$. The graphs of the functions are shown in Fig. 8. The reference point for hypervolume calculation is taken to be $[-1, -10]$.

B. Additive Noise

We tested with two kinds of Gaussian noise; the first was homoscedastic with variance $r(x) = \bar{\sigma}_n^2 (= \text{const.})$, and the second was heteroscedastic with variance $r(x) = \{\sigma_n(\sin(\|\mathbf{x}\|) + 1)/2\}^2$.



(a) Objective 1 (b) Objective 2

Fig. 8. Graph of the test function $\mathcal{T}_6$

C. Performance Metric

For the selection metric of the optimization method, the hypervolume indicator [41] was used. This is a common unary indicator that has been examined closely [45]. We calculated the true noiseless function values at the points of the resultant approximated Pareto set. The hypervolume was calculated based on these true function values, instead of the sampled values, because hypervolume calculated based on noisy samples can be under- or overestimated. Note that the number of points in the approximated Pareto set varies from trial to trial, and is not constant. We ran 60 trials for each setting and calculated the empirical median, 25th percentile, and 75th percentile of the hypervolume. In the tests, the approximated Pareto front is generated from the evaluated points. However, note that it is suggested that constructing the approximated Pareto front from surrogate functions would give a better approximation than constructing it only from evaluations [46]. Therefore, the performance shown in what follows can be understood as a lower limit.

If there is no noise on the observations, the hypervolume will increase monotonically as the number of observations gets larger. However, in our case, because the algorithm plans the experiments and returns the approximated Pareto set based on the noisy samples, the corresponding estimated hypervolume can decrease.

D. Common Settings

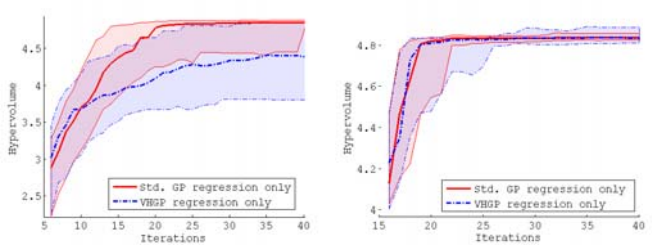
To solve the maximization problem of EIHV, we chose to first calculate EIHV at densely sampled points, and then used the maximizer among these points as the starting point of the gradient method. Of course, other kinds of maximization methods, like a gradient method with random restart, or some EAs, are applicable.

Regarding the initial settings of the hyperparameters, we set all of the initial values of the hyperparameters at 1 for standard homoscedastic GP regression. For heteroscedastic regression, we used the result of homoscedastic regression to set the initial hyperparameter values.

All numerical tests were repeated 60 times to make the results statistically reliable. To illustrate the results, hypervolumes were plotted against the number of evaluations (Figs. 9, 12-22). Because the distribution of hypervolumes after a fixed number of evaluations is skewed, we used the median and the 25th/75th percentiles instead of the mean and standard deviation, respectively. Thick lines refer to the median, and the colored areas refer to the region between the 25th and 75th percentiles.

E. Results

1) Need for Model Selection between Two Kinds of GPs: First, to show the need for the model selection discussed in Section II-C, we compare the performances of two cases: (i) standard GP regression only (existing method [3]) and (ii)



(a) 5 initial evaluations (b) 15 initial evaluations

Fig. 9. Comparison between (i) the existing method (std. GP regression only) and (ii) the method using VHGP regression only. The test function is MAT (hypervolume: 5.1013).

VHGP regression only. The tests were performed with two different settings for the number of initial points: 5 and 15. The test function MAT was used, and homoscedastic noise ($\bar{\sigma}_n = 0.15$) was added to observations. A maximum of 40 evaluations were performed for each trial. The results are shown in Fig. 9.

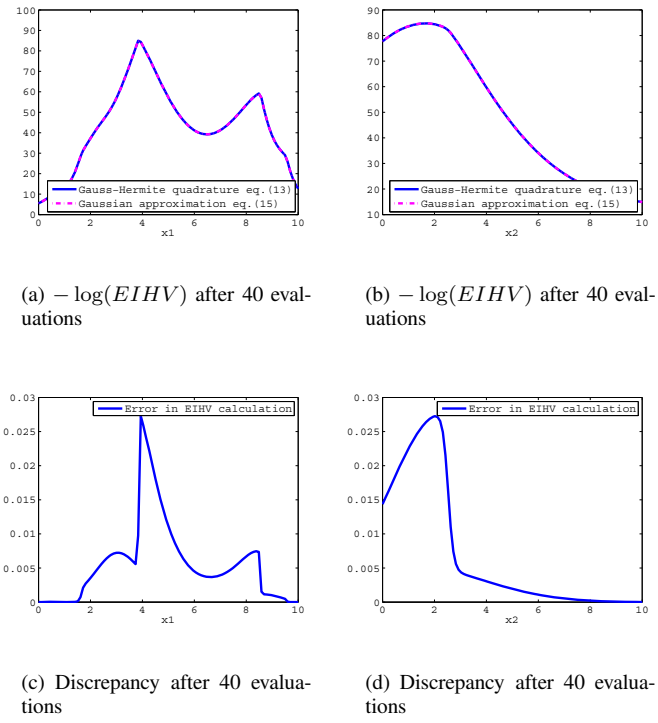
In the case with 5 initial evaluations, the existing method (standard GP regression only) clearly outperforms the other (VHGP regression only). This is because the VHGP model is more complex than the standard GP model, and tends to overfit to the small size data. In the case with 15 initial points, both methods work equally well.

The problem is that, in real problems, the necessary number of initial evaluations for VHGP regression is likely unknown, and this in turn shows the need for model selection between standard GP and VHGP. We show in the following that through the model selection methods proposed in Section II-C, the results will be at least as good as, and in many cases better than, the best of methods (i) and (ii).

2) Comparison between two EI calculations, (14) and (16): In this test, we compared the performances of the two calculations of EIHV in the case of VHGP regression: Gauss–Hermite quadrature (14) and Gaussian approximation of the predictive density (16). Here, only VHGP regression was used and not standard GP regression.

The tests were done for MAT with two kinds of additive noise: homoscedastic noise with $\bar{\sigma}_n = 0.15$, and sinus noise with $\sigma_n = 0.2$, as explained in Section III-B. The number of initial evaluations was set as 15. For Gauss–Hermite quadrature, 9 (3×3) nodes were used, which gave enough precision for EIHV calculation.

Figs. 10 and 11 show axis-aligned slices of the negative log EIHV surface after 40 evaluations as calculated by each method, and the corresponding slices of the discrepancy. The planes shown are selected to go through the point with maximum error. It can be seen that the discrepancy becomes large at the maxima of the negative log EIHV, and at the points with small negative log EIHV, the discrepancy becomes small. Because we need the minimizer of the negative log EIHV (maximizer of the EIHV), the approximation (16) is considered appropriate for our purpose, though it can be inaccurate in the area that we are not interested in. Additionally, we observed



(a) $-\log(EIHV)$ after 40 evaluations (b) $-\log(EIHV)$ after 40 evaluations

(c) Discrepancy after 40 evaluations (d) Discrepancy after 40 evaluations

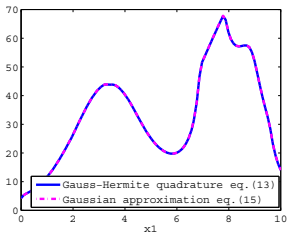
Fig. 10. Slices of the negative log EIHV surface calculated by Gauss–Hermite quadrature (14) and Gaussian approximation (16) ((a), (b)), and their difference ((c), (d)). The slices are axis-aligned ((a) and (c) with $x_2 = 2.0202$ and (b) and (d) with $x_1 = 3.9394$), and go through the point with maximum discrepancy between the methods. Homoscedastic noise ($\bar{\sigma}_n$=0.15) is used.

that σ_* in (10) are very small compared with $|\mu_*|$; in fact, $\sigma_*/|\mu_*|$ is around $10^{-5} - 10^{-10}$ at the minimizer of the negative log EIHV, which verifies the use of the approximation (16).

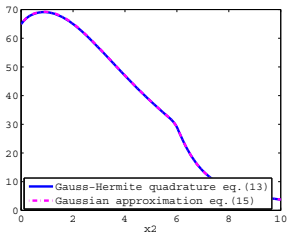
Hypervolumes were plotted against the number of evaluations in Fig. 12, and Fig. 13(a) shows the time consumption for each step in the case with sinus noise. The total time required for one trial was the integral of the curve, which was about 6 hours where Gauss–Hermite quadrature was used, and about 44 minutes in the other case. From these results, it can be seen that the calculation of EIHV through Gauss–Hermite quadrature is very time consuming given the insignificant improvements in accuracy.

Because our goal is to find the solution set with as few evaluations as possible and not to reduce the calculation cost, a method that requires a great amount of calculation can be used as long as it contributes to reducing the number of necessary observations. However, because the time-consuming calculation (14) did not reduce the required number of samples, we concluded that the approximation (16) should be applied instead.

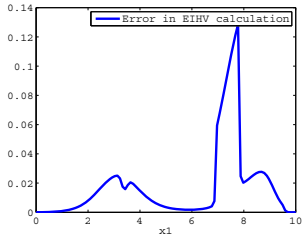
3) Comparison between two model selection methods: For selecting between standard GP and VHGP, two methods are compared: LOO as explained in Section II-C, and numerical calculation of marginal likelihood by MC. See the Appendix for details on MC calculation. MAT was used as the test function. The results are shown in Fig. 14, and time required



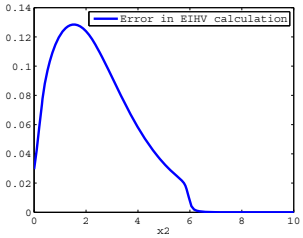
(a) $-\log(EIHV)$ after 40 evaluations



(b) $-\log(EIHV)$ after 40 evaluations

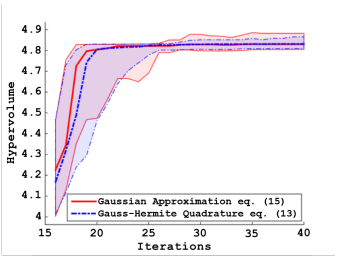


(c) Discrepancy after 40 evaluations

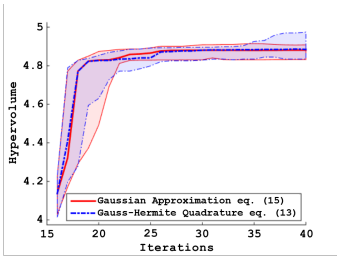


(d) Discrepancy after 40 evaluations

Fig. 11. Slices of the negative log EIHV surface calculated by Gauss–Hermite quadrature (14) and Gaussian approximation (16) ((a), (b)), and their difference ((c), (d)). The slices are axis-aligned with $x_1 = 7.7778$ ((a), (c)) and $x_2 = 1.5152$ ((b), (d)). Sinus noise (σ_n=0.2) is used.



(a) Homoscedastic noise ($\bar{\sigma}_n = 0.15$)



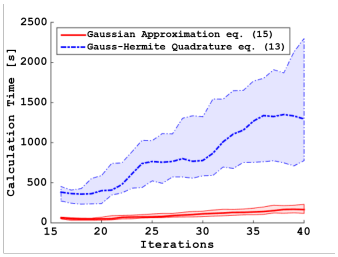
(b) Sinusoidal noise ($\sigma_n = 0.2$)

Fig. 12. Comparison between two metric calculations: (14) and its approximation (16). Because we use 15 initial evaluations, the graphs begin with 16 evaluations. The test function is MAT (hypervolume: 5.1013).

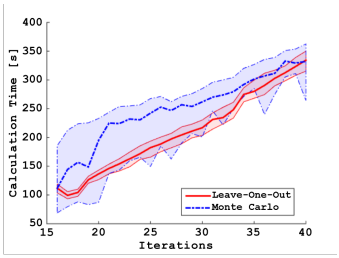
for each step of the procedure is shown in Fig. 13(b).

From Fig. 14 and 13(b), it can be seen that the performance is slightly better if LOO is used, but required time is also less with LOO if there are less than about 40 points. The poorer performance of MC is attributed to the limited precision of the calculation of marginal likelihood. Because precision through MC is proportional to the square root of the sample size, it is difficult to precisely calculate the marginal likelihood, which is typically much smaller than 1. Moreover, we usually use the logarithm of marginal likelihood, which amplifies the error as follows:

$$\log(p + \Delta p) - \log(p) \approx \Delta p / p > \Delta p, \tag{25}$$

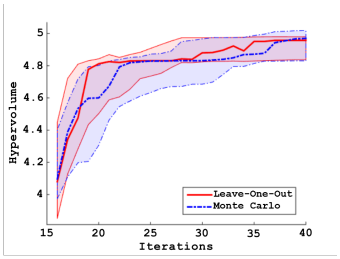


(a) Simplified EI vs Complete EI

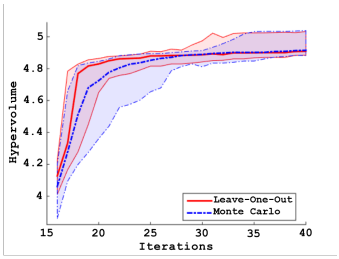


(b) LOO vs MC

Fig. 13. Time needed to complete one step of the procedure.



(a) Homoscedastic noise ($\bar{\sigma}_n = 0.15$)



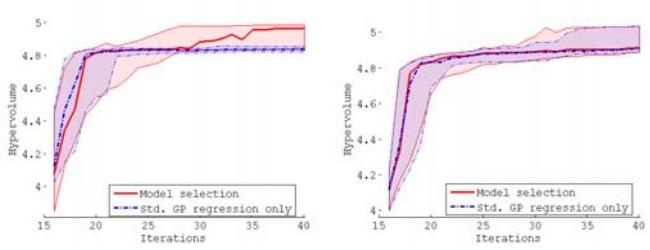
(b) Sinusoidal noise ($\sigma_n = 0.2$)

Fig. 14. Comparison between LOO and MC as the model selection method. The number of initial evaluations is 15. The test function is MAT (hypervolume: 5.1013).

where $p < 1$ is the marginal likelihood. Therefore, we need several hundred times more samples to get 10 times more precise estimation. However, as can be seen from Fig. 13(b), calculation time for LOO grows faster than that for MC and will be much slower if there are many more evaluations.

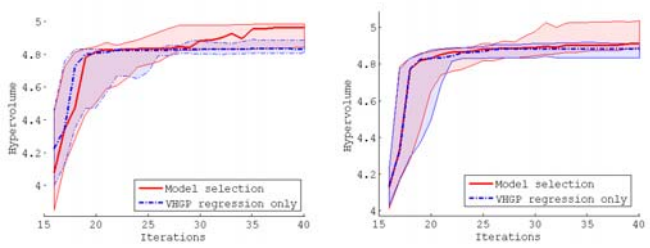
4) Comparison among three methods (2-D search space): Finally, the performance of the proposed method was tested against that of the existing method and the method that only uses VHGP regression. In this test, all 4 test functions were used to check the class of problems for which the proposed method works efficiently. The number of initial experiments was 15 for MAT, 20 for $\mathcal{T}_3$ (2-D) and $\mathcal{T}_4$, and 40 for $\mathcal{T}_6$. The results for MAT are shown in Fig. 15 and 16, $\mathcal{T}_3$ in Fig. 17 and 18, $\mathcal{T}_4$ in Fig. 19 and 20, and $\mathcal{T}_6$ in Fig. 21 and 22. It can be seen that except for $\mathcal{T}_6$, the proposed method outperforms the other two, or at least performs equally well.

For $\mathcal{T}_6$, the proposed method is outperformed by the existing method in the case where the noise is homoscedastic, but in the other cases, it works at least as well as the others. In this case, the method using VHGP regression exclusively performs worse than the other two, even in the existence of heteroscedastic noise. This is a somewhat counterintuitive result because VHGP regression seemed appropriate for fitting samples wherein the noise level is actually a function of input and should therefore provide a good surrogate. One possible explanation is that VHGP regression has too much complexity in its noise model to fit the samples of size 40 or so from $\mathcal{T}_6$,



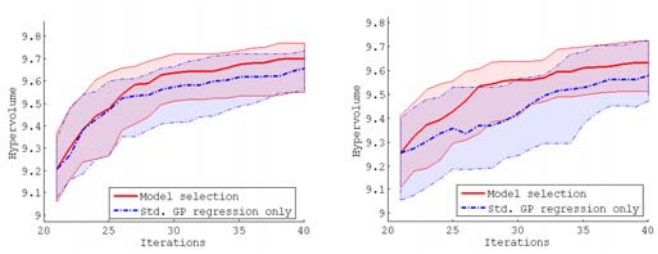
(a) Homoscedastic noise ($\bar{\sigma}_n = 0.15$) (b) Sinusoidal noise ($\sigma_n = 0.2$)

Fig. 15. Comparison between the proposed method (model selection) and the existing method (std. GP regression only) [3]. The number of initial evaluations is 15. The test function is MAT (hypervolume: 5.1013).



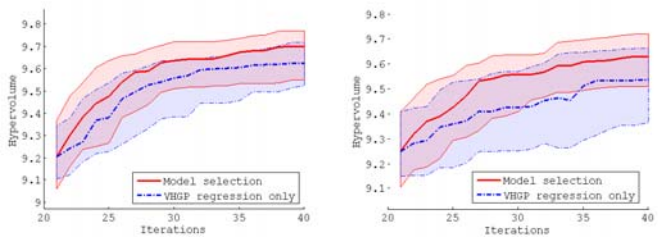
(a) Homoscedastic noise ($\bar{\sigma}_n = 0.15$) (b) Sinusoidal noise ($\sigma_n = 0.2$)

Fig. 16. Comparison between the proposed method (model selection) and the method that uses VHGP regression only. The test function is MAT (hypervolume: 5.1013).



(a) Homoscedastic noise ($\bar{\sigma}_n = 0.1$) (b) Sinusoidal noise ($\sigma_n = 0.2$)

Fig. 17. Comparison between the proposed method and the existing method [3]. The number of initial evaluations is 20. Test function is $\mathcal{T}_3$ (hypervolume: 10.0444).



(a) Homoscedastic noise ($\bar{\sigma}_n = 0.1$) (b) Sinusoidal noise ($\sigma_n = 0.2$)

Fig. 18. Comparison between the proposed method and the method that uses VHGP regression only. Test function is $\mathcal{T}_3$ (hypervolume: 10.0444).

overfitting the change for the latent function because of the noise. This can lead to a much flatter surrogate surface than the true latent function. For the case of homoscedastic noise, the proposed method is inferior to the existing method. This can be seen as a failure of our model selection method, but we note that a similar trend is observed, even when we used the maximization of the marginal likelihood calculated by MC.

5) 10-D tests: For this test, the 10-D version of the $\mathcal{T}_3$ function, i.e. $\mathcal{T}_3$ with $D = 10$ in (22), is optimized. In each trial, 50 initial points and 100 points in total are sampled. The additive noise is the same for the 2-D case of $\mathcal{T}_3$. The results are shown in Fig. 23. From the figures, it can be seen that both the existing method and the proposed method perform well, in that within only a small number of experiments, the hypervolumes get larger compared with their initial values. However, our proposed method outperforms the existing one in terms of final values.

IV. Experiments

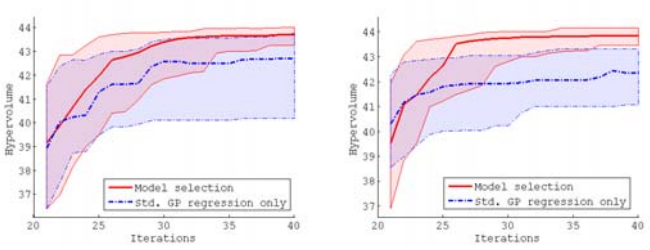
We conducted robotic experiments with a snake robot, which moves via sidewinding locomotion. The objective functions were set to be the speed and the stability of the robot head. Here, head stability is roughly inversely proportional to the amount of head motion, which is very important when operating the robot. Again, in the experiments we did not implement Lines 11-13 of the Algorithm 1.

The snake robot is often controlled by motions with a finite dimensional constrained control trajectory subspace (the same as the *gait model* described in [47]), which is defined as follows:

$$\alpha(n,t) = \begin{cases} \beta_{\text{even}} + A_{\text{even}} \sin(\theta), & n = \text{even}, \\ \beta_{\text{odd}} + A_{\text{odd}} \sin(\theta + \delta), & n = \text{odd}, \end{cases} \qquad (26)$$
$$\theta = (\omega_s n + \omega_t t)\,,$$

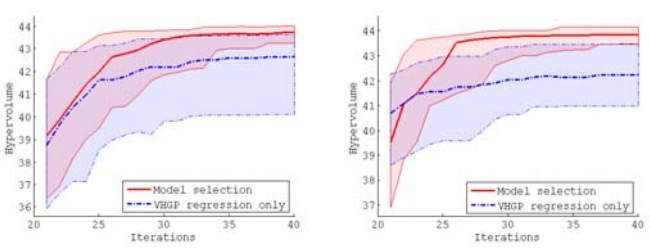
where $\alpha(n,t)$ is the nth joint angle at time t. This can be seen as an extension of the serpenoid curve [48] which models the shape of a real snake well. There are 7 free parameters (βs, As, δ, ω_s and ω_t), and by tuning these parameters, we can command the snake to move via many kinds of motions, including slithering, sidewinding, and rolling in an arc.

Snake robots are expected to be useful for many applications, including pipe inspections and urban search and rescue operations. In practice, the operator would have to rely on the camera image from the camera mounted on the head, which makes it challenging to operate the robot from a distance. Although a method that would provide information about the state of the robot based on the virtual chassis [49] has been proposed, it remains difficult to operate the robot if the camera is mounted on a shaky base. At the same time, it would be advantageous to move the robot faster, which will amplify



(a) Homoscedastic noise ($\bar{\sigma}_n = 0.2$) (b) Sinusoidal noise ($\sigma_n = 0.25$)

Fig. 19. Comparison between the proposed method and the existing method [3]. The number of initial evaluations is 20. Test function is $\mathcal{T}_4$ (hypervolume: 44.6667).



(a) Homoscedastic noise ($\bar{\sigma}_n = 0.2$) (b) Sinusoidal noise ($\sigma_n = 0.25$)

Fig. 20. Comparison between the proposed method and the method that uses VHGP regression only. Test function is $\mathcal{T}_4$ (hypervolume: 44.6667).

the movement of the head camera. Therefore, we chose two objective functions as follows.

1) The speed: the net displacement of the head after running the snake for 15 seconds.
2) The head stability: the stability of the image from the camera mounted on the head of the snake.

For head stability, we put an acceleration sensor on the head, and calculated the level of vibration as follows. Let $\boldsymbol{a}(t)$ be the read of acceleration sensor, then the head stability is defined as

$$(\text{Head Stability}) = -\int_0^T \|\boldsymbol{a}(t) - E[\boldsymbol{a}(t)]\|^2 dt, \qquad (27)$$

where $E[\boldsymbol{a}(t)]$ is the mean of $\boldsymbol{a}(t)$ during one run.

Although it is not clear only from the above definitions whether there is a trade-off between the 2 objectives, the proposed method, along with other MOO methods, can be used for cases where the objectives do not actually conflict. Therefore, if there is no reason to deny the existence of conflict between objectives, it is better to turn to MOO.

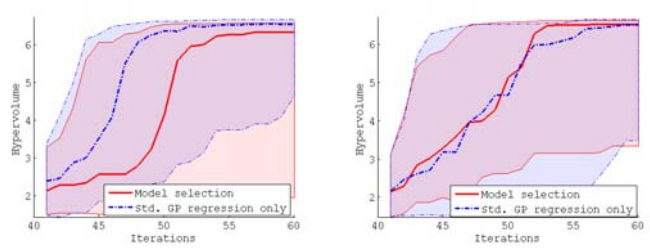
The consideration of heteroscedasticity in the robotic experiments is verified later in this subsection through another set of experiments, but essentially, it is appropriate considering that the larger the movement, the greater the noise level.

In these experiments, the model was restricted to the sidewinding parameter space as defined in [47] (the phase shift δ is fixed as $\pi/4$, the offsets βs as 0, $\omega_s = 3\pi/16$, and $\omega_t = 3\pi/5$), and we introduced an additional phase shift ϕ for the head module:

$$\alpha(1,t) = A_{\text{odd}} \sin(\theta + \delta + \phi). \qquad (28)$$

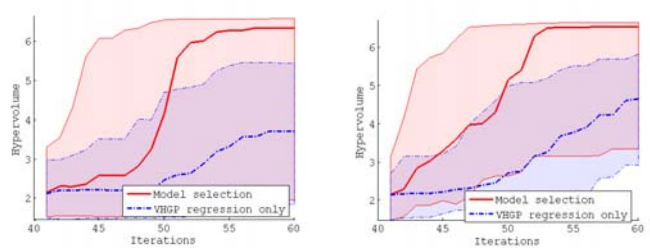
The ratio between A_{odd} and A_{even} was also fixed as $A_{\text{odd}}/A_{\text{even}} = 3/8$. The parameters that should be optimized are the amplitude A_{odd} and the phase shift ϕ. The domains of the parameters were set as $A_{\text{odd}} \in [0.23, 1]$ and $\phi \in [0, 2\pi]$.

The first 20 evaluations were planned for initialization



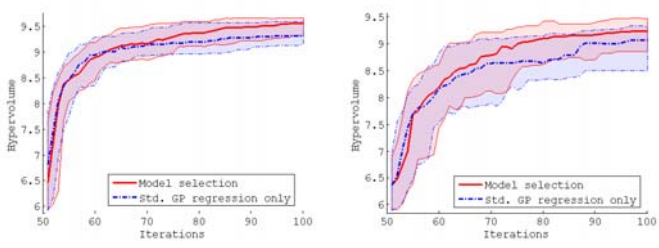
(a) Homoscedastic noise ($\bar{\sigma}_n = 0.1$) (b) Sinusoidal noise ($\sigma_n = 0.1$)

Fig. 21. Comparison between the proposed method and the existing method [3]. The number of initial evaluations is 40. Test function is $\mathcal{T}_6$ (hypervolume: 6.7989).



(a) Homoscedastic noise ($\bar{\sigma}_n = 0.1$) (b) Sinusoidal noise ($\sigma_n = 0.1$)

Fig. 22. Comparison between the proposed method and the method that uses VHGP regression only. Test function is $\mathcal{T}_6$ (hypervolume: 6.7989).



(a) Homoscedastic noise ($\bar{\sigma}_n = 0.1$) (b) Heteroscedastic noise ($\sigma_n = 0.2$)

Fig. 23. Comparison between the proposed method and the existing method. Test function is $\mathcal{T}_3$ with 10-D input (hypervolume: 10.0444).

Fig. 24. Snake robot

through Latin hypercube design and the 20 subsequent evaluations were selected using the proposed method; the same procedure was also performed for the existing method. The initial samples were shared by both methods, and in total, 10 sets of experiments were conducted for each method. Figs. 25(a) and 25(b) show the resulting Pareto fronts for the proposed method and the existing method [3], respectively, for both of which we used the same initial evaluations. By comparing the points that are the best in terms of speed, it is clear that the proposed method found solutions better (i.e., faster) than the existing method. The proposed method outperforms the other in terms of stability.

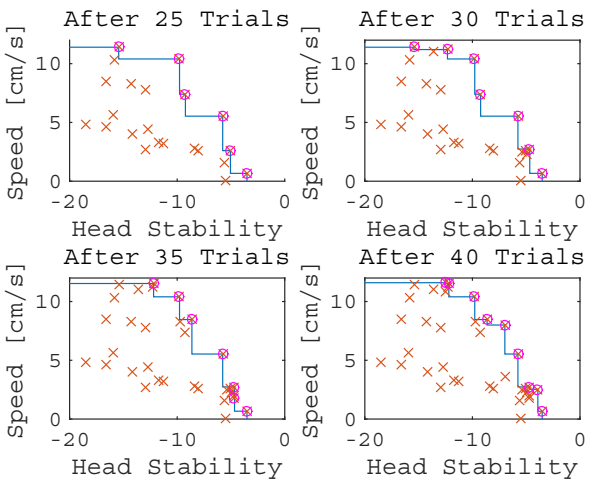
Figure 26 shows the response surfaces used in planning for the 40th experiment with the proposed method. The crosses are the samples, and the color shows the variance of the predictive density. In Fig. 26 (c) and (d), the variance is shown as color maps, from which it can be seen that the uncertainty is not uniform across the domain.

Figure 27 shows the plot of the hypervolumes. The thick lines represent the median, and the colored area shows the region between the 25th and 75th percentiles. Though comparison between hypervolumes is generally difficult because of the lack of knowledge of the true objective functions, it is clear from this figure that the proposed method outperforms the existing method. In Fig 27, we also show the median of 5 random searches as a dotted line. Note that we omit the error band for random search to keep the figure from becoming cluttered. From this, it is clear that both the existing and proposed methods are much more efficient than the random search.

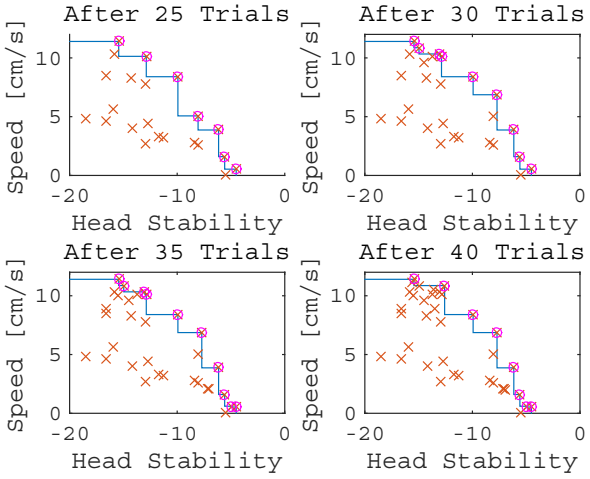
To verify the consideration of heteroscedasticity, we conducted another set of experiments. We chose 3 inputs: $\mathbf{x}_1 = [0.7177,\ 0.3569]$, $\mathbf{x}_2 = [0.5634, 0.0538]$, and $\mathbf{x}_3 = [0.3767, 0.5500]$, and evaluated the head stability for each. We took 5 samples for each input, which are shown in Table II. Each row corresponds to an input, and the last column shows the empirical standard deviation, which is the square root of the unbiased estimator of noise variances. It can be seen that there are large differences in the standard deviations. For example, the standard deviation for $\mathbf{x}_1$ is more than 10 times larger than that for $\mathbf{x}_3$. In fact, the null hypothesis, that the variances between these two sets of samples are equal, is rejected by a two-sample F-test with a 1 % significance level. These results clearly show that the noise is actually heteroscedastic in this robotic example.

V. Conclusions

In this paper we proposed a Pareto optimization method that can be used for the optimization of expensive noisy



(a) Proposed method



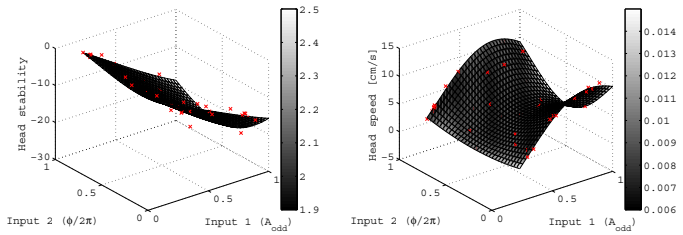
(b) Existing method

Fig. 25. Resulting Pareto front in the case of (a) the proposed method and (b) the existing method [3] after 25, 30, 35, and 40 evaluations. Observed data are shown in $\times$ marks and the circles are the elements in the Pareto front.

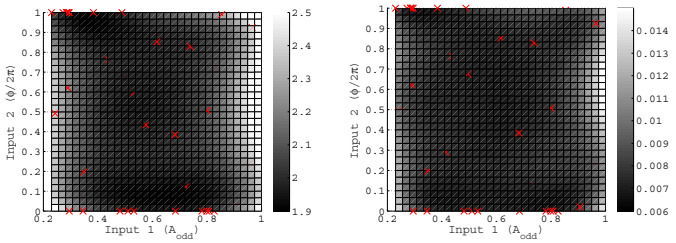
functions. By selecting between standard and heteroscedastic GPs using LOO, the performance of the proposed method becomes better than our previous method not only in the presence of heteroscedastic noise, but also in the case of homoscedastic noise. A model selection method that can be used to select between standard GP and heteroscedastic GP by LOO was also proposed. This method often results in better

TABLE II
Observed value of the head stability

	Samp.1	Samp.2	Samp.3	Samp.4	Samp.5	st. dev.
$\mathbf{x}_1$	19.0062	15.2082	22.6125	16.6942	17.4333	2.8252
$\mathbf{x}_2$	10.4526	9.4061	9.8783	9.6554	11.5510	0.8542
$\mathbf{x}_3$	7.0852	6.9981	6.4631	6.8942	7.1128	0.2644



(a) Head stability (b) Head speed



(c) Uncertainty in head stability estimation (d) Uncertainty in head stability estimation

Fig. 26. Example of response surface of head stability and speed after 39 experiments, which was used to plan the 40th experiment. The color shows the variance of the predictive distribution. (c) and (d) show the variance as color maps, from which the non-uniformity of the uncertainty can be ascertained.

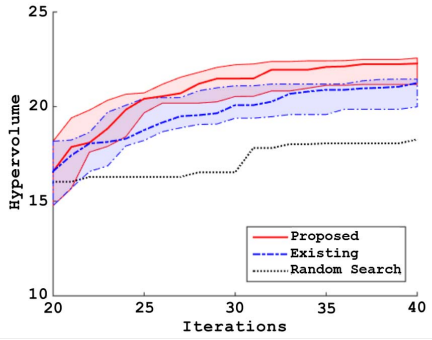


Fig. 27. Hypervolume indicator calculated from the experimental results

results with less computation than the numerical calculation of marginal likelihood by MC sampling.

In this research, we only used EIHV and did not compare with other potentially powerful alternative metrics, such as probability of improvement or upper confidence bound. Comparison between these methods in single-objective optimization in robotics can be found in [28], but for the multiobjective case it remains to be done. Another direction of future research will include extending our method to be able to deal with changing environments. Since every single observation in one environment will give some information about the performance in all the other environments, in particular for similar environments, it will be possible to accelerate the optimization procedure by using the results from other environments. Our method will be useful in the case where the function evaluation is expensive and the noise cannot be neglected; however, it is not so efficient if the dimensionality of the search space is very large, as is the case for hyper-redundant robots. Constructing an efficient MOO method that can be used for high-dimensional noisy objectives is another possible topic for future work.

APPENDIX

The marginal likelihood is calculated as follows:

$$p(\mathbf{y}) = \int p(\mathbf{y}|\mathbf{f},\mathbf{g})p(\mathbf{f})p(\mathbf{g})d\mathbf{f}d\mathbf{g}, \tag{29}$$

where $p(\mathbf{y}|\mathbf{f},\mathbf{g})$ is the likelihood function, and $p(\mathbf{f})$ and $p(\mathbf{g})$ are the prior density functions. Given that the priors are Gaussian, it is easy to draw samples from $p(\mathbf{f})$ and $p(\mathbf{g})$. Let the samples from the prior densities be $\mathbf{f}^{(i)}$ and $\mathbf{g}^{(i)}$ where $i = 1, ..., M$. Then, the approximation of the marginal likelihood $\hat{p}(\mathbf{y})$ can be obtained

$$\hat{p}(\mathbf{y}) = \frac{1}{M}\sum_{i=1}^{M} p(\mathbf{y}|\mathbf{f}^{(i)},\mathbf{g}^{(i)}). \tag{30}$$

We use $M \in [2 \times 10^5,\ 1 \times 10^6]$ points, where the exact number differs according to the convergence of the calculation. Although it is well known that in some cases this simple approximation can be inaccurate, it is easy to implement and gives reasonable results in our cases. Other much more sophisticated methods using Markov chain Monte Carlo, such as those proposed in [50],[51], can also be used. However, we found that, although they reduce the calculation time, they make no significant difference in hypervolume from the basic calculation (30). We also note that it is often the case that the sampling used in MCMC calculation is ill-conditioned, and the calculation becomes unreliable. In some cases, it even fails in drawing samples, in which cases naive MC also fails to return meaningful results.

ACKNOWLEDGMENT

This work was partially supported by the JSPS Institutional Program for Young Researcher Overseas Visits. This study was supported by a JSPS Grant-in-Aid for JSPS fellows.

REFERENCES

[1] R.T. Marler, and J.S. Arora, "Survey of multi-objective optimization methods for engineering," *Struct. Multidis. Optim.*, vol. 26, no. 6, pp. 369–395, Apr. 2004.

[2] M.G.C. Tapia, C.A. Coello Coello, "Applications of Multi-Objective Evolutionary Algorithms in Economics and Finance: A Survey," in *Proc. IEEE Congr. Evol. Comput.*, 2007, pp. 532–539.

[3] M. Tesch, J. Schneider, and H. Choset, "Expensive Multiobjective Optimization for Robotics," in *Proc. IEEE Int. Conf. Robot. Autom.*, Karlsruhe, Germany, 2013, pp. 973–980.

[4] M. Oliveira, L. Costa, A. Rocha, C. Santos, and M. Ferreira, "Multiobjective Optimization of a Quadruped Robot Locomotion Using a Genetic Algorithm," *Soft Computing in Industrial Applications*, Springer Berlin Heidelberg, vol. 96, pp. 427–436, 2011.

[5] G. Capi, M. Yokota, and K. Mitobe, "A New Humanoid Robot Gait Generation based on Multiobjective Optimization," in *Proc. IEEE/ASME Int. Conf. Adv. Intell. Mechatron.*, Monterey, CA, 2005, pp. 450–454.

[6] M. Ringkamp, S. Ober-Blöbaum, M. Dellnitz, and O. Schütze, "Handling High Dimensional Problems with Multi-Objective Continuation Methods via Successive Approximation of the Tangent Space," *Eng. Optim.*, vol. 44 no. 9, pp. 1117–1146, 2012.

[7] A. Kumar, and A. Vladimirsky, "An Efficient Method for Multiobjective Optimal Control and Optimal Control Subject to Integral Constraints," *J. Comput. Math.*, vol.28, no.4, pp. 517–551, Apr. 2010.

[8] A. Zhou, B.-Y. Qu, H. Li, S.-Z. Zhao, P.N. Suganthan, and Q. Zhang, "Multiobjective evolutionary algorithms: A survey of the state of the art," *Swarm and Evolutionary Computation*, vol. 1, no. 1, pp. 32–49, Mar. 2011.

[9] R. Marchant and F. Ramos, "Bayesian Optimisation for Intelligent Environmental Monitoring." in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, Algarve, Portugal, 2012, pp. 2242–2249.

[10] E. Brochu, V. Cora and N. de Freitas, "A Tutorial on Bayesian Optimization of Expensive Cost Functions, with Application to Active User Modelling and Hierarchical Reinforcement Learning." University of British Columbia, Tech. Rep., 2010.

[11] D.R. Jones, M. Schonlau, and W.J. Welch, "Efficient Global Optimization of Expensive Black-Box Functions," *J. Global Optim.*, vol. 13, no. 4, pp. 455–492, Dec. 1998.

[12] J. Knowles, "ParEGO: A Hybrid Algorithm With On-Line Landscape Approximation for Expensive Multiobjective Optimization Problems," *IEEE Trans. Evol. Comput.*, vol. 10, no. 1, pp. 50–66, Feb. 2005.

[13] Q. Zhang, and H. Li, "Expensive Multiobjective Optimization by MOEA/D with Gaussian Process Model," *IEEE Trans. Evol. Comput.*, vol. 14, no. 3, pp. 456–474, Jun. 2010.

[14] M. Emmerich, K. Giannakoglou, and B. Naujoks, "Single- and Multiobjective Evolutionary Optimization Assisted by Gaussian Random Field Metamodels," *IEEE Trans. Evol. Compt.*, vol.10, no. 4, pp. 421–439, Aug. 2006.

[15] W. Ponweiser, T. Wagner, D. Biermann, and M. Vincze, "Multiobjective Optimization on a Limited Budget of Evaluations Using Model-Assisted $\mathcal{S}$-Metric Selection," in *Proc. 10th Parallel Problem Solving Nat.*, Dortmund, Germany, 2008, pp. 784–794.

[16] T. Akhtar, and C. A. Shoemaker, "Multi objective optimization of computationally expensive multi-modal functions with RBF surrogates and multi-rule selection," J. Glob. Optim., 2015, http://dx.doi.org/10.1007/s10898-015-0270-y.

[17] J. Svenson, and T. Santner, "Multiobjective optimization of expensive-to-evaluate deterministic computer simulator models," *Computational Statistics and Data Analysis*, vol. 94, pp. 250–264, Feb. 2016 (in Progress)

[18] B. Binois, D. Ginsbourger, and O. Roustant, "Quantifying Uncertainty on Pareto Fronts with Gaussian Process Conditional Simulations," in *Proc. of Learning and Intelligent Optimization Conference*, vol. 243, no. 2, pp. 386–394, 2014

[19] J. Teich, "Pareto-Front Exploration with Uncertain Objectives", in *Proc. 1st Conf. Evol. Multi-Criterion Optim.*, Zurich, Switzerland, 2001, pp. 314–328.

[20] E. Zitzler, and L. Thiele, "Multiobjective Evolutionary Algorithms: A Comparative Case Study and the Strength Pareto Approach," *IEEE Trans. Evol. Comput.*, vol.3, no.4, pp. 257–271, Nov. 1999.

[21] D. Büche, P. Stoll, R. Dornberger, and P. Koumoutsakos, "Multiobjective Evolutionary Algorithm for the Optimization of Noisy Combustion Processes," *IEEE Trans. Syst. Man Cybern. C*, vol. 32, no.4, pp. 460–473, Nov. 2002.

[22] H. Eskandari, and C.D. Geiger, "Evolutionary multiobjective optimization in noisy problem environments," *J. Heuristics*, vol. 15, no. 6, pp. 559–595, Dec. 2009.

[23] H. Eskandari, and C.D. Geiger, "A fast Pareto Genetic Algorithm Approach for Solving Expensive Multiobjective Optimization Problems," *J. Heuristics*, vol. 14, no. 3, pp. 203–241, Jun. 2008.

[24] J. E. Fieldsend, and R. M. Everson, "The Rolling Tide Evolutionary Algorithm: A Multiobjective Optimizer for Noisy Optimization Problems," *IEEE Trans. Evol. Comput.*, vol. 19, no. 1, pp. 103–117, Feb. 2015.

[25] C.E. Rasmussen and C.K.I. Williams, *Gaussian Processes for Machine Learning*, Cambridge, MA, USA: The MIT Press, 2006,

[26] D. Lizotte, T. Wang, M. Bowling, and D. Schuurmans, "Automatic Gait Optimization with Gaussian Process Regression," in *Proc. Int. Joint Conf. Artificial Intell.*, Hyderabad, India, 2007, pp. 944–949.

[27] M. Tesch, J. Schneider, and H. Choset, "Using Response Surfaces and Expected Improvement to Optimize Snake Robot Gait Parameters," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, San Francisco, CA, 2011, pp. 1069–1074.

[28] R. Calandra, A. Seyfarth, J. Peters, and M.P. Deisenroth, "Bayesian Optimization for Learning Gaits under Uncertainty," *Ann. Math. Artif. Intell.*, vol. 76, pp. 5–23, 2016.

[29] M. Zuluaga, A. Krause, G. Sergent, and M. Püschel, "Active Learning for Multi-Objective Optimization," in *Proc. Int. Conf. Machine Learning*, Atlanta, GA, 2013, pp. 462–470.

[30] M. Emmerich and J.-W. Klinkenberg, "The computation of the expected improvement in dominated hypervolume of Pareto front approximations," Leiden Institute for Advanced Computer Science, Tech. Rep. 1, 2008.

[31] J.-A.M. Assael, Z. Wang, B. Shahriari, and N. Freitas, "Heteroscedastic Treed Bayesian Optimization," arXiv preprint arXiv:1410.7172, 2014.

[32] M.A. Taddy, H.K.H. Lee, G.A. Gray, and J.D. Griffin, "Bayesian Guided Pattern Search for Robust Local Optimization," *Technometrics*, vol. 51, no. 4, pp. 389–401, 2009.

[33] P. Goldberg, C. Williams, and C. Bishop, "Regression with Input-dependent Noise: A Gaussian Process Treatment," in *Proc. Adv. Neural Inf. Process. Syst.*, Denver, CO, 1997, pp. 493–499.

[34] K. Kersting, C. Plagemann, P. Pfaff, and W. Burgard, "Most Likely Heteroscedastic Gaussian Processes Regression," in *Proc. Int. Conf. Machine Learning*, Corvallis, OR, 2007, pp. 393–400.

[35] M. Lázaro-Gredilla, and M.K. Titsias, "Variational Heteroscedastic Gaussian Process Regression," in *Proc. Int. Conf. Machine Learning*, Bellevue, WA, 2011, pp. 841–848.

[36] S. Kuindersma, R. Grupen, and A. Barto, "Variable Risk Control via Stochastic Optimization," *Int. J. Robot. Res.*, vol. 32, no. 7, pp. 806–825, Jun. 2013.

[37] R. Martinez-Cantin, N. Freitas, E. Brochu, J. Castellanos, and A. Doucet, "A Bayesian exploration-exploitation approach for optimal online sensing and planning with a visually guided mobile robot." *Autonomous Robots*, vol. 27, no. 2, pp. 93–103, 2009

[38] R. Martinez-Cantin, "BayesOpt: A Bayesian Optimization Library for Nonlinear Optimization, Experimental Design and Bandits." *Journal of Machine Learning Research*, vol. 15, no. 1, pp. 3735–3739, 2014.

[39] R. Ariizumi, M. Tesch, H. Choset, F. Matsuno, "Expensive Multiobjective Optimization for Robotics with Consideration of Heteroscedastic Noise," in *Proc. IEEE/RSJ Int. Conf. Intell. Robotics Syst.*, Chicago, IL, 2014, pp. 2230–2235.

[40] M. D. Mackay, W. J. Conover, and R. J. Beckman, "A comparison of three methods for selecting values of input variables in the analysis of output from a computer code," *Technometrics*, vol. 21 no. 2, pp. 239–245, May. 1979.

[41] E. Zitzler and L. Thiele, "Multiobjective Optimization Using Evolutionary Algorithms - A Comparative Case Study," *Parallel problem solving from nature – PPSN V*, Springer, pp. 292–304, Sep. 1998.

[42] Z. Wang, F. Hutter, M. Zoghi, D. Matheson, and N. Freitas, "Bayesian optimization in a billion dimensions via random embeddings." *Journal of Artificial Intelligence Research*, vol. 55, pp. 361–387, 2016

[43] E. Zitzler, K. Deb, and L. Thiele, "Comparison of Multiobjective Evolutionary Algorithms: Empirical Results," *Evol. Comput.*, vol. 8, no. 2, pp. 173–195, 2000.

[44] L. Dixon and G. Szego, "The global optimization problem: an introduction," *Towards Global Optimization*, vol. 2, pp. 1–15, 1978.

[45] E. Zitzler, L. Thiele, M. Laumanns, C. M. Fonseca, and V. Grunert da Fonseca, "Performance Assessment of Multiobjective Optimizers: An Analysis and Review," *IEEE Trans. Evol. Comput.*, vol. 7, no. 2, Apr. 2003.

[46] R. Calandra, J. Peters, and M.P. Deisenroth, "Pareto Front Modeling for Sensitivity Analysis in Multi-Objective Bayesian Optimization," *NIPS Workshop on Bayesian Optimization*, vol. 5, 2014

[47] M. Tesch, K. Lipkin, I. Brown, R. Hatton, A. Peck, J. Rembisz and H. Choset, "Parameterized and Scripted Gaits for Modular Snake Robots," *Adv. Robot.*, vol. 23, no.9, pp. 1131–1158, 2009.

[48] S. Hirose, *Biologically Inspired Robots: Snake-Like Locomotiors and Manipulators.* Oxford, U.K.: Oxford Univ. Press, 1993.

[49] D. Rollinson and H. Choset, "Virtual Chassis for Snake Robots," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, San Francisco, CA, 2011, pp. 221–226.

[50] S. Chib, and I Jeliazkov, "Marginal Likelihood From the Metropolis-Hastings Output," *J. Americal Statistical Association*, vol. 96, no. 453, pp. 270–281, Mar. 2001.

[51] S. Chib, and I Jeliazkov, "Accept-Reject Metropolis-Hastings Sampling and Marginal Likelihood Estimation," *Statistica Neerlandica*, vol. 59, no. 1, pp. 30–44, Feb. 2005.

PLACE PHOTO HERE

Ryo Ariizumi received BS, ME, and PhD degrees in engineering from Kyoto University in 2010, 2012, and 2015, respectively. He was a postdoctoral researcher at Kyoto University, and is currently an assistant professor at Nagoya University.

His current research interests include control of redundant robots, and optimization of robotic systems.

PLACE PHOTO HERE

Matthew Tesch is a postdoctoral researcher at Carnegie Mellon University. His research focuses on machine learning methods for efficiently optimizing robotic systems from which data collection is time consuming or expensive. Application of this work has resulted in significant improvements of the locomotive capabilities of snake robots. Dr. Tesch received a BS in engineering from Franklin W. Olin College in 2007, and MS and PhD degrees in robotics from Carnegie Mellon University in 2011 and 2013, respectively.

PLACE PHOTO HERE

Kenta Kato Kenta Kato received a BS in mechanical engineering from the Kyoto University, Kyoto, Japan, in 2014. He is currently working toward his ME in mechanical engineering with the Department of Mechanical Engineering and Science, Kyoto University.

His current research interests include semi-autonomous control of rescue robots and machine learning, especially expensive multiobjective optimization.

PLACE PHOTO HERE

Howie Choset is a professor of robotics at Carnegie Mellon University. Motivated by applications in confined spaces, he has created a comprehensive program in snake robots, which has led to basic research in mechanism design, path planning, motion planning, and estimation. By pursuing the fundamentals, this research program has made contributions to coverage tasks, dynamic climbing, and large-space mapping.

Prof. Choset has already directly applied this body of work in challenging and strategically significant problems in diverse areas, such as surgery, manufacturing, infrastructure inspection, and search and rescue. He directs the Undergraduate Robotics Minor Program at Carnegie Mellon University and teaches an overview course on robotics, which uses series of custom developed Lego Labs to complement the course work.

Prof. Choset's students have won best paper awards at the RIA in 1999 and ICRA in 2003; his group's work has been nominated for best papers at ICRA in 1997, IROS in 2003 and 2007, and CLAWAR in 2012 (best biorobotics paper, best student paper); They also won best paper at IEEE Bio Rob in 2006, best video at ICRA 2011, and was nominated for best video in ICRA 2012. In 2002, the MIT Technology Review elected Choset as one of its top 100 innovators under 35 in the world. In 2005, MIT Press published a textbook entitled "Principles of Robot Motion," which was lead-authored by Choset.

PLACE PHOTO HERE

Fumitoshi Matsuno (M'94) received a PhD (Dr Eng) from Osaka University, Osaka, Japan, in 1986.

In 1986, he joined the Department of Control Engineering, Osaka University. He became a lecturer and associate professor in 1991 and 1992, respectively, in the Department of Systems Engineering, Kobe University. In 1996, he joined the Department of Computational Intelligence and Systems Science, Interdisciplinary Graduate School of Science and Engineering, Tokyo Institute of Technology, as an associate professor. In 2003, he became a professor with the Department of Mechanical Engineering and Intelligent Systems, University of Electro-Communications. Since 2009, he has been a professor with the Department of Mechanical Engineering and Science, Kyoto University, Kyoto, Japan. He is also a Vice-President of NPO International Rescue Systems Institute (IRS) and the Institute of Systems, Control and Information Engineers (ISCIE). His current research interests include robotics, control of distributed parameter systems and nonlinear systems, rescue support systems in fires and disasters, and swarm intelligence.

Dr. Matsuno has received many awards, including the Outstanding Paper Award in 2001 and 2006, the Takeda Memorial Prize in 2001 from the Society of Instrument and Control Engineers (SICE), and the Outstanding Paper Award in 2013 from the Information Processing Society of Japan. He is a fellow member of SICE, the Japan Society of Mechanical Engineers, the Robotics Society of Japan, and a member of IEEE, ISCIE, among other organizations. He served as the co-chair of the IEEE Robotics and Automation Society Technical Committee on Safety, Security, and Rescue Robotics; the chair of the Steering Committee of the SICE Annual Conference; and is an editor for the *Journal of Intelligent and Robotic Systems*, an associate editor for *Advanced Robotics* and *International Journal of Control, Automation, and Systems* among others, and an editorial board member with the Conference Editorial Board of the IEEE Control Systems Society.