# HDD Slider Air Bearing Design Optimization Using a Surrogate Model

Naozumi TSUDA
Visiting industrial fellow
Dept of Mechanical Engineering
University of California, Berkeley
Email: naozumi@me.berkeley.edu

David B Bogy
William S. Floyd, Jr. Distinguished Professor in Engineering
Department of Mechanical Engineering
University of California, Berkeley
Email: dbogy@cml.me.berkeley.edu

## *1. ABSTRACT*

This report addresses a new optimization method in which the DIRECT algorithm is used in conjunction with a surrogate model. The DIRECT algorithm itself can find the global optimum with a high convergence rate. However the convergence rate can be much improved by coupling DIRECT with a surrogate model. The surrogate model known as the Kriging model is used in this research. It is determined by using sampling points generated by the DIRECT algorithm. This model expresses the shape of a hyper surface approximation of the cost function over the entire search space. Finding the optimum point on this hyper surface is very fast because it is not necessary to solve the time consuming air bearing equations. But the optimum point on the hyper surface only determines an area where the true optimum point may exist. By using this optimum candidate as one of the DIRECT sampling points, we

can eliminate many cost function evaluations. To illustrate the power of this approach we first present some simple optimization examples using known difficult functions. Then we determine the optimum design of a slider with 5nm flying height (FH) starting with a design that has a 7nm FH. Finally in order to find a feasible design we present an optimization method that takes into account the sensitivity of the design to design variable variations.

## *2. INTRODUCTION*

In order to achieve a recording areal density of 1Tbit/in$^2$ in hard disk drives (HDD), the magnetic spacing between the read/write transducer and the magnetic medium must be reduced to approximately 5-6nm. As the FH of the air bearing slider is decreased, the ratio of the fly height modulation to the total fly height increases and degrades the read/write signal quality. Air bearing surface (ABS) designers have mainly focused on the static characteristics of the slider, such as FH, pitch and roll. However in order to design an ABS with small fly height modulation (FHM), the designer must consider the dynamic characteristics of the ABS. Dynamic simulation is much more expensive than static evaluation so the optimization of a slider design based on its dynamic characteristics becomes very expensive. Therefore we need to develop a searching algorithm that can find the global optimum design with a relatively small number of cost function evaluations.

In this report, we first show how the searching process works, and then we introduce the Kriging model, which is used as a surrogate model of the cost function. After a brief discussion of these concepts, we show some simple optimization examples. Using these examples we show a comparison of the convergence rate of the proposed surrogate method and the DIRECT algorithm, which has been previously studied and implemented in the CML

ABS optimization program. Finally we present an example of ABS optimization with sensitivity considerations.

## *3. PROPOSED SEARCHING METHOD*

### 3.1 Combination of DIRECT and DACE

In optimization research in the field of DACE (design and analysis of computer experiments) [1] [2], the Kriging model is used as a surrogate model for the time consuming cost function. Usually a data sampling method is employed to construct the surrogate model. Some examples of data sampling methods for DACE are Monte Carlo sampling, Latin hypercube sampling [3], and orthogonal array sampling [4]. The data sampling method itself is unrelated to the function that is to be searched for the optimum. It just generates sampling points that are supposed to be suitable for constructing the surrogate model.

All of these data sampling methods are different from the classical design of experiments (DOE) methods, such as the central composite design, in how the sampling points are spread in the search space. Data sampling methods for DACE tend to cover the entire search space whereas DOE methods tend to locate points at the boundary of the search space [5].

Since the DIRECT algorithm is a global optimization algorithm, it generates sampling points, as cost function evaluation points, that are spread over the entire search space. So we can use the DIRECT algorithm not only as the optimization algorithm but also as the data sampling algorithm.

### 3.2 Process configuration

Figure 1 shows a diagram of the process configuration. There are two processes. One is the DIRECT process and the other is the Kriging process.

DIRECT works as the optimization process using the ordinary DIRECT algorithm, except this process uses optimum candidates obtained from the Kriging process.

The Kriging process forms a Kriging model using sample points at which the cost function is evaluated in the DIRECT process, in other words, in our case, an air bearing simulation is performed using the program CMLAir. It is formed after each iteration of DIRECT. After forming the Kriging model, the optimum point on the Kriging model is searched for using the DIRECT algorithm version III (DIRECTIII), which is one of the locally biased DIRECT algorithms proposed by Zhu in 2002 [5]. It uses fewer sampling points and divides the search area twice in one iteration. Its convergence speed is fastest among the several versions of the DIRECT algorithm. After an optimum candidate is obtained in the Kriging process, this design is passed on the DIRECT process for further investigation.

### 3.3 Insertion of the predicted optimum into the DIRECT process

An optimum candidate obtained from the Kriging model is inserted into the line of samples, called link lists, generated by the DIRECT algorithm. As this optimum candidate is obtained from the Kriging process, it does not have the cell information needed by the DIRECT algorithm for grouping and dividing cells. To create these data, the cell that has the closest cost function value to the optimum candidate is searched for from the top of the line of the samples. The cell information of this closest cell is used as the cell information of the optimum candidate cell. After this cell is inserted, it is treated as a cell generated by DIRECT.

From the numerical experiments performed we found that this optimum candidate can be the best among the samples in the early stages of the search. But the optimum candidate obtained from the Kriging process cannot be a best design among the samples as the dividing proceeds farther. The DIRECT algorithm can find more precise candidates than those provided by the Kriging process if DIRECT uses many samples. There are several ways to resolve this problem. One typical way that is often used in optimization methods using a surrogate model is to limit the search area around the optimum candidate when the convergence rate becomes low. Then the surrogate model is formed in the limited area. By doing this, the surrogate model becomes more accurate since it is not disturbed by the sample points far from the candidate point. We don't consider these techniques any further in this report.

## *4. CONSTRUCTING A SURROGATE MODEL*

### 4.1 Summary of the Kriging model

The Kriging model consists of two parts, a regression model and a correlation model. A polynomial of any order can be used as a regression model. However, a polynomial cannot be fit well enough to many sample points since the shape of the cost function is usually very complex. On the other hand the shape of a polynomial is very simple, even one of high order. In the Kriging model, a correlation model also works well to fit the sample points. Figure 2(a) shows examples of a Kriging model. We first tried to form a Kriging model for the Branin function using just five sample points. The black dots are sample points on the Branin function. Off course five sample points are too few to accurately fit the Branin function. But one can see the structure of the Kriging model very well. The flat part of the surface is formed by a 1st order regression model, and the peaks and dimples are formed by the correlation model. The function values at the points in the area where samples do not exist are interpolated according to the distances from the existing sample points. As the number of sample points is increased, the shape of the Kriging model becomes closer to the Branin function, as shown in Fig. 2(b).

### 4.2 Formulation

The MATLAB Kriging Toolbox [6] was used to form the Kriging model. The purpose is to form a linear predictor $\hat{y}(x)$ at the location $x \in \Re^n$ using the sample points $\{s_i\} \subset \Re^n, i = 1,2,\cdots,m$ , which are obtained by the DIRECT algorithm. In other words, $\hat{y}(x)$ represents the predicted cost function at a certain value of the design parameter $x$ of the slider, where $x$ could be, for example, a dimension of the rail or some other parameter such as the recess depth. We assume a linear predictor $\hat{y}(x)$ defined as follows.

$$\hat{y}(x) = c(x)^T y_s \tag{1}$$

where $y_s$ is the column vector of the function values at the sample points, namely $y_s = [y(s_1) \quad y(s_2) \quad \cdots \quad y(s_m)]^T$ . Also, $c(x) \in \Re^m$ is the vector of weights applied to $y_s$. We will determine $c(x)$ in the remainder of this section.

Kriging, also called DACE (design and analysis of computer experiments), is a statistical method originating from the field of geostatistics, which is based on the use of spatial correlation functions [7]. In the probabilistic model, the total search area is assumed to be an infinite family of random functions [8]. The Kriging model is comprised of a regression function and a random function described as follows

$$Y(x) = \beta^T f(x) + Z(x) \tag{2}$$

The deterministic response $\hat{y}(x)$ is a realization of a random function $Y(x)$. $\beta^T f(x)$ is a regression term, where $\beta = [\beta_1 \quad \beta_2 \quad \cdots \quad \beta_k]^T$ and $f(x) = [f_1(x) \quad f_2(x) \quad \cdots \quad f_k(x)]^T$. The number of terms of the regression part is k and $\beta$ is a coefficient of the regression terms. For example, if a 2nd order polynomial is chosen as a regression model in the two dimensional case, $f(x)$ is written as

$$f(x) = [1 \quad x_1 \quad x_2 \quad x_1^2 \quad x_2^2 \quad x_1 x_2]^T . \tag{3}$$

$Z(x)$ is a random function, which is assumed to have a zero mean, variance $\sigma^2$, and correlation $R(w, x)$ between $x$ and any other point $w$.

To describe the cost function with fidelity, Kriging weights are determined by minimizing the mean squared error of the predicted value $\hat{y}(x)$. The error between the predicted value and the actual value $Y(x)$ at the location $x$ is described as follows,

$$\begin{aligned} c(x)^T y_s - Y(x) &= c(x)^T (F\beta + Z) - (f(x)^T \beta + z) \\ &= c(x)^T Z - z + (F^T c(x) - f(x))^T \beta \end{aligned} , \tag{4}$$

where $F = [f(s_1), \ldots, f(s_m)]^T \in \Re^{m \times k}$ and $Z = [z_1, \ldots, z_m]^T$. $z_1, \ldots, z_m$ are realizations of $Z(s_i), i = 1,2,\ldots,m$ and $z$ is a realization of $Z(x)$. To keep the predictor unbiased, we need to have

$$F^T c(x) - f(x) = 0 \tag{5}$$

Under this condition, the mean squared error (MSE) of the predicted value is,

$$
\begin{aligned}
MSE[\hat{y}(x)] &= E\left[(c(x)^T y_s - Y(x))^2\right] \\
&= E\left[(c(x)^T Z - z)^2\right] \\
&= E\left[c(x)^T Z(c(x)^T Z)^T - 2c(x)^T Zz + z^2\right] \\
&= E\left[c(x)^T ZZ^T c(x) - 2c(x)^T Zz + z^2\right] \quad .
\end{aligned} \tag{6}
$$

Because the random function $Z(x)$ is assumed to have zero mean, variance $\sigma^2$ and correlation $R(w,x)$ , it follows that $E[z^2] = \sigma^2, E[Zz] = \sigma^2 r$ and $E[ZZ^T] = \sigma^2 R$ , where $r \in \Re^m$ is a correlation vector between the known sample points and an unexplored point, and $R \in \Re^{m \times m}$ is the correlation matrix between samples. Therefore the MSE becomes

$$
\begin{aligned}
MSE[\hat{y}(x)] &= E\left[c(x)^T ZZ^T c(x) - 2c(x)^T Zz + z^2\right] \\
&= c(x)^T \sigma^2 Rc(x) - 2c(x)^T \sigma^2 r + \sigma^2 \\
&= \sigma^2 (c(x)^T Rc(x) - 2c(x)^T r + 1) \quad .
\end{aligned} \tag{7}
$$

The Lagrangian function is used to minimize $MSE[\hat{y}(x)]$ with respect to $c(x)^T$ and subject to the constraint (5).

$$
L(c(x),\lambda) = \sigma^2 (c(x)^T Rc(x) - 2c(x)^T r + 1) - \lambda^T (F^T c(x) - f(x)) \tag{8}
$$

where $\lambda \in \Re^k$ is the Lagrange multiplier. By taking the gradient of this Lagrangian with respect to $c(x)$ and setting the gradient to zero, we obtain the following results.

$$
\begin{aligned}
&Rc(x) + F\tilde{\lambda} = r \\
&F^T c(x) = f(x) \ ,
\end{aligned} \tag{9}
$$

where $\tilde{\lambda} = -\dfrac{\lambda}{2\sigma^2}$.

The solution of (8) is

$$
\begin{aligned}
&\tilde{\lambda} = (F^T R^{-1} F)^{-1} (F^T R^{-1} r - f(x)), \\
&c(x) = R^{-1}(r - F\tilde{\lambda}).
\end{aligned} \tag{10}
$$

Inserting (10) into (1) we obtain,

$$
\begin{aligned}
\hat{y}(x) &= (r - F\tilde{\lambda})^T R^{-1} y_s \\
&= r^T R^{-1} Y - (F^T R^{-1} r - f)^T (F^T R^{-1} F)^{-1} F^T R^{-1} Y \\
&= r^T R^{-1} Y - (F^T R^{-1} r - f)^T \beta^* \\
&= f^T \beta^* + r^T R^{-1}(Y - F\beta^*) \\
&= f^T \beta^* + r^T \gamma^* \qquad (11)
\end{aligned}
$$

As can be seen in equation (4), $f^T$ depends on the kind of polynomial chosen. On the other hand, there are several correlation functions available in the MATLAB Kriging Toolbox. Here we used a cubic type function as follows.

$$
\begin{aligned}
R(\theta, s_i, x) &= \prod_{j=1}^{n} R(\theta, s_{ij} - x_j) \\
&= \prod_{j=1}^{n} R(1 - 3\xi_j^2 + 2\xi_j^3), \quad \xi_j = \min\{1, \theta_j \mid s_{ij} - x_j \mid\} \qquad (12)
\end{aligned}
$$

where $s_{ij}$ and $x_j$ are the jth components of a known sample point $s_i$ and the unexplored point $x$, respectively.

Now $r^T$ is the correlation between the known sample points and the unexplored point, and $r(x)$ can be described as:

$$
r(x) = [R(\theta, s_1, x) \quad \cdots \quad R(\theta, s_m, x)]^T . \qquad (12)
$$

In the same way, the element of matrix $R \in \Re^{m \times m}$ can be described as,

$$
R_{ij} = R(\theta, s_i, s_j), \quad i, j = 1, ..., m. \qquad (13)
$$

The optimal value $\theta$ is determined by using a maximum likelihood estimation in each dimension so that

$$
\min_{\theta}\{\varphi(\theta) \equiv |R|^{\frac{1}{m}} \hat{\sigma}^2\}, \qquad (14)
$$

where $|R|$ is the determinant of R, and

$$\hat{\sigma}^2 = \frac{1}{m}(y_s - F\beta^*)^T R^{-1}(y_s - F\beta^*) . \tag{15}$$

## *5. SOME OPTIMIZATION EXAMPLES*

### 5.1 Test functions

To verify the proposed algorithm we used three simple functions as cost functions, which are defined as follows:

2-D: $F(x_1, x_2) = 100\,(x_1 - x_2^2)^2 + (1 - x_2)^2$, where $x_1, x_2 \in [-2.048, 2.048]$.

10-D: $F(x_1, x_2, x_3, x_4, x_5, x_6, x_7, x_8, x_9, x_{10}) = (x_1 - 0.1)^2 + (x_2 - 0.2)^2 +$

$(x_3 - 0.3)^2 + (x_4 - 0.4)^2 +$

$(x_5 - 0.5)^2 + (x_6 - 0.6)^2 +$

$(x_7 - 0.7)^2 + (x_8 - 0.8)^2 +$

$(x_9 - 0.9)^2 + (x_{10} - 1.0)^2.$

*where* $x_i \in [0,1]$, $i = 1,\ldots 10.$

20-D: $F(x_1, x_2, x_3, x_4, x_5, x_6, x_7, x_8, x_9, x_{10}, x_{11}, x_{12}, x_{13}, x_{14}, x_{15}, x_{16}, x_{17}, x_{18}, x_{19}, x_{20}) =$

$(x_1 - 0.05)^2 + (x_2 - 0.1)^2 + (x_3 - 0.15)^2 + (x_4 - 0.2)^2 +$

$(x_5 - 0.25)^2 + (x_6 - 0.3)^2 + (x_7 - 0.35)^2 + (x_8 - 0.4)^2 +$

$(x_9 - 0.45)^2 + (x_{10} - 0.5)^2 + (x_{11} - 0.55)^2 + (x_{12} - 0.6)^2 +$

$(x_{13} - 0.65)^2 + (x_{14} - 0.7)^2 + (x_{15} - 0.75)^2 + (x_{16} - 0.8)^2 +$

$(x_{17} - 0.85)^2 + (x_{18} - 0.9)^2 + (x_{19} - 0.95)^2 + (x_{20} - 1.0)^2.$

*where* $x_i \in [0,1]$, $i = 1,\ldots 20.$

The first one, called the Rosenbrock function, is known as a “tough” function for optimization algorithms even though its dimension is just 2. The global minimum of this function is located at the bottom of wide spread flat valley. This characteristic makes optimization algorithms likely to get trapped at a local minimum [5]. The other two functions are used to show if the algorithm can handle multi-dimensional cases. All of these functions have only one global minimum.

### 5.2 Results

Figures 3 ~ 5 show the optimization results. The solid marks represent the proposed method. We tried the proposed method with four different versions of the DIRECT algorithm, DIRECT, DIRECTI, DIRECTII and DIRECTIII. Detailed definitions of these are given in Zhu [5]. The hollow marks represent the results using the various DIRECT algorithms. All of these figures show that all four versions of the DIRECT algorithm combined with Kriging model are faster than the corresponding versions of the DIRECT algorithm alone. We see the remarkable advantage in the 20-D case. The proposed method is 5 times faster than DIRECTIII, which is the fastest version of the DIRECT algorithms.

## *6. SLIDER ABS DESIGN OPTIMIZATION*

### 6.1 Problem

We chose the Information Storage Industry Consortium (INSIC) 7nm fly height Pico slider as the prototype slider (Pico refers to a certain set of overall dimensions of the slider). Its rail shape and constraint conditions are shown in Fig. 6.

The optimization problem defined here is to optimize the rail shape so that the flying height is 5nm and the roll profile is as flat as possible across the disk.

### 6.2 Constraints and Weights

We used relative constraints to reduce the dimension of the problem. These relative constraints are reasonable from a manufacturing point of view. The resulting number of design variables is eight.

The disk rotation speed is 7600rpm. The air bearing simulations are performed at three radius positions designated OD, MD and ID. We want to find a design that predicts the target flying height, i.e. 5nm at all three radii.

The weights of the cost function are chosen as,

*1(FH Max Difference term) + 9(FH term) + 1(Roll term) +*

*1(Roll Cutoff term) + 1(Pitch Cutoff term) + 1(Vertical Sensitivity term) +*

*1(Pitch Sensitivity term) + 1(Roll Sensitivity term) + 1(Negative Force term) .*

### 6.3 Results

Figure 7 shows the convergence rate of the proposed method and the DIRECT method. The convergence rate of the combination of DIRECT and Kriging is faster than that of the combination of DIRECTIII and Kriging in the example optimization cases. However, in the slider optimization case, the convergence tendency is different. The combination of DIRECTIII and Kriging converged faster than the others and the combination of DIRECT and Kriging converged only slightly faster than DIRECTIII alone.

## *7. DISCUSSION*

In the Figs 3 ~ 5, the DIRECT algorithms combined with the Kriging model show faster convergence rate than the DIRECT algorithms alone. And there are no remarkable differences among these. The slight difference of the convergence speed was caused by the difference of the number of samples in an iteration.

The Kriging model is a hyper-surface interpolated among the sample points. Therefore unexplored regions can be searched using the Kriging model without system analysis. Namely, a local-searching around a candidate region is done by using the Kriging model without fly height calculations. This is the main reason why the DIRECT algorithms combined with the Kriging model can get optimum designs faster than the DIRECT algorithms alone. However, the Kriging model does not have enough spatial resolution as the sample points generated by DIRECT algorithms. There is no way other than to wait until the DIRECT algorithm generates more samples to find a better design. We can see that the convergence rate suddenly improves after not improving after several iterations. It is conceivable that the DIRECT algorithms generated samples in the region where a better design might exist. By using these sample points the Kriging model could be formed with enough fidelity and a better design could be found without fly height calculations.

Different from the test function, the result of the slider optimization shows that DIRECTIII combined with the Kriging model shows a faster convergence rate than the standard DIRECT combined with the Kriging model. The reason that the DIRECT algorithm combined with the Kriging model did not converge well is that the fidelity of the Kriging model after approximately 75 samples were generated was not good, and the Kriging model could not find a good optimum candidate.

## *8. SENSITIVITY OPTIMIZATION*

### 8.1 Cost function for sensitivity optimization

Every dimension of the slider has some tolerance range. It is important to design a slider that is robust against the modulation of these dimensions. Even if the slider is designed with remarkably good characteristics at the nominal dimensions, it is not a feasible design if this characteristic degrades when the dimensions change within a tolerance range. Therefore it is very important to design a slider that is insensitive to slight modulations of the dimensions as well as being close to the target slider design

Both DIRECT and the combination of DIRECT and Kriging search for the optimum design based on the cost function values. The smallest function value corresponds to the best design of the whole design set. However, as already mentioned, actual designers not only want the design whose function value is smallest but also the design that is robust against manufacturing error.

The variation of the cost function caused by the fluctuation of the design variables does not represent directly the variation of the characteristics of the slider caused by manufacturing error. But we tried to minimize the sensitivity of the characteristics of the slider against the fluctuation of the design variables using worst-case analysis [9][10] for the first step.  A Modified cost function enables the algorithm to find the best design for actual designers. This cost function can be described as,

$$f_{new}(X) = Max\{ f_{org}(X - H), f_{org}(X), f_{org}(X + H)\}, \tag{16}$$

where X is the vector whose elements are the nominal design variables and H is the vector whose elements are the tolerance ranges. In this work, the tolerance ranges are the same across all dimensions.

## 8.2 Test function

To validate that this cost function works as expected we used the following test cost function (Fig. 8).

$$f = -\frac{1-e^{-40\cdot 0.6^2}}{1-e^{-8\cdot 0.6^2}} \cdot e^{-4\cdot\left\{(x_1-0.2)^2-(x_2-0.2)^2\right\}} - e^{-20\cdot\left\{(x_1-0.8)^2-(x_2-0.8)^2\right\}},\ x_1, x_2 \in [0,1] \qquad (17)$$

The surface defined by this function has one local minimum, point A, and one global minimum, point B. The differential coefficient around point A (0.2, 0.2) is smaller than that around point B (0.8, 0.8). The cost function value around point B is slightly smaller than that of point A. It means that point B has the lowest function value, however, it is more sensitive to the modulation of the parameters than point A. Therefore, if the optimum point on this surface is sought without consideration of sensitivity, the optimization algorithm will find point B as the optimum.

In Fig. 9, the modified cost function value is shown. The cost function value at the point A directly represents the value at point A. However, if the parameter modulates within the range $h_a$, the cost function value should take the range of values between A+ $h_a$ or A- $h_a$. Therefore, the cost function value at point A should be represented by the maximum value within the tolerance range $\pm h_a$.

### 8.3 Relation between tolerance range and optimum design

Since the cost function value is represented by the maximum value within the tolerance range, the optimum design is strongly related to the tolerance range. Figure 10 shows how the optimum design could change as the tolerance range changes. Figure 10(a) shows the relation of the cost function at points A and B with a wide tolerance range. Figure 10(b) shows the case of a narrow tolerance range. In case of the wide tolerance range, the function value at point B becomes greater than that at point A because the differential coefficient around point B is greater than at point A. However, if the tolerance range is set smaller, point B could be the optimum design. Such inversion of the design could easily occur when the function values at the local minimum and global minimum are close.

### 8.4 Optimization result

Figure 11(a) shows the wide tolerance (5%) case and Fig. 11(b) shows the narrow tolerance (1%) case. In the wide tolerance case, the DIRECT algorithm can find the insensitive design. And in the narrow tolerance case, the point that has the lowest function value and wide differential coefficient becomes an optimum design, as intended.

### 8.5 Summary

By using a modified cost function value, the DIRECT algorithm can find an insensitive design. Because the surface used as a cost function has one local minimum and one global minimum, and these function values are close, the sensitive design could be an optimum design when the tolerance is small. High differential coefficients around the optimum design does not become a problem because the tolerance is narrow enough and the maximum function value around this design within the tolerance range is low enough compared with the insensitive design.

## *9. SUMMARY AND CONCLUSION*

(1) We combined the DIRECT algorithm and the Kriging model. This method enables us to find the global minimum with a reduced number of cost function evaluations. The DIRECTIII algorithm is the fastest version among the DIRECT algorithms, however, the combination of DIRECT and Kriging is faster than DIRECTIII.

(2) We tried a worst-case analysis to find a robust design against the fluctuation of the design variables. By using it, we can get not only a low value of the cost function but also an insensitive design. However, because of the relation between the tolerance range and cost function value, the optimum design can change with tolerance. Namely, if a designer sets a narrow tolerance range the optimum design could be a sensitive one, but because the tolerance range is narrow, degradation of the slider's performance characteristics does not occur. Even if a designer must accept the wide tolerance range, it may be possible to get a design whose cost function value is small enough and is also insensitive against the design parameter changes.

## ACKNOWLEDGEMENT

This research was supported by the Computer Mechanics Laboratory and Fujitsu Laboratories, LTD.

**REFERENCES**

1. Koehler, J. R. and Owen, A. B., "Computer Experiments," *Handbook of Statistics (Ghosh, S . and Rao, C. R., eds.),* Elsevier Science, New York, 1996, pp. 261-308.

2. Sacks, J., Welch, W. J., Mitchell, T. J. and Wynn, H. P., "Design and Analysis of Computer Experiments," *Statistical Science,* Vol. 4, No. 4, 1989, pp. 409-435.

3. McKay, M. D., Beckman, R. J., and Conover, W. J., “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, 1979, pp. 239-245.

4. Koehler, J. R., and Owen, A. B., “Computer Experiments,” Vol. 13 *of Handbook of Statistics*, Elsevier-Science, New York, NY, eds. S. Ghost and C. R. Rao, 1996, pp. 239-245.

5. H. Zhu and D. B. Bogy, “Hard Disk Drive Air Bearing Design: Modified DIRECT Algorithm and its Application to Slider Air bearing Surface Optimization” Tribology International (Austrib02 special issue), Vol. 37, Issue 2, Feb. 2004, pp 193-201.

6. Søren N. Lophaven, Hans Bruun Nielsen, Jacob Søndergaard ,“A MATLAB Kriging Toolbox”, Technical Report IMM-TR-2002-12, Technical University of Denmark

7. Alison L. Marsden, et al., “Optimal aeroacoustic shape design using the surrogate management framework”, CAAM Technical Reports TR04-05, Rice University

8. Hans WackernagelTT, “Multivariate Geostatistics: An Introduction With Applications”, Springer-Verlag

9. A. Parkinson, “Robust Mechanical Design Using Engineering Models”, Transaction of The ASME, Vol.117, June 1995.

10. N. Hirokawa, “Computational Cost Improvement of Robust Optimal Design by Cumulative Function Approximation”, AIAA-2004-4381.

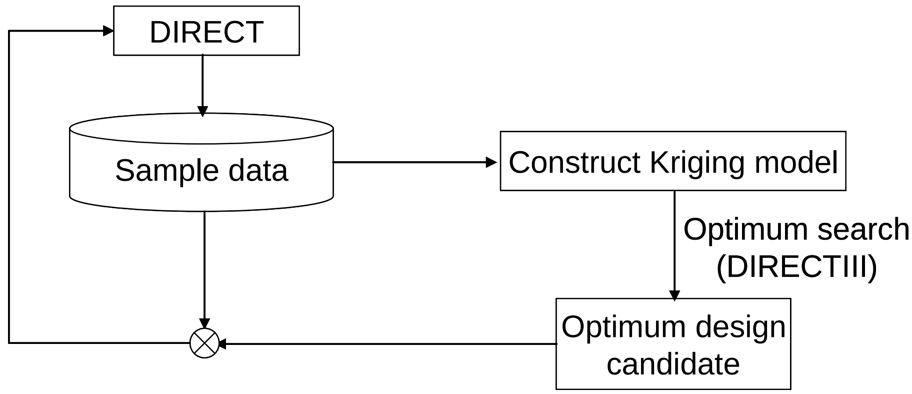
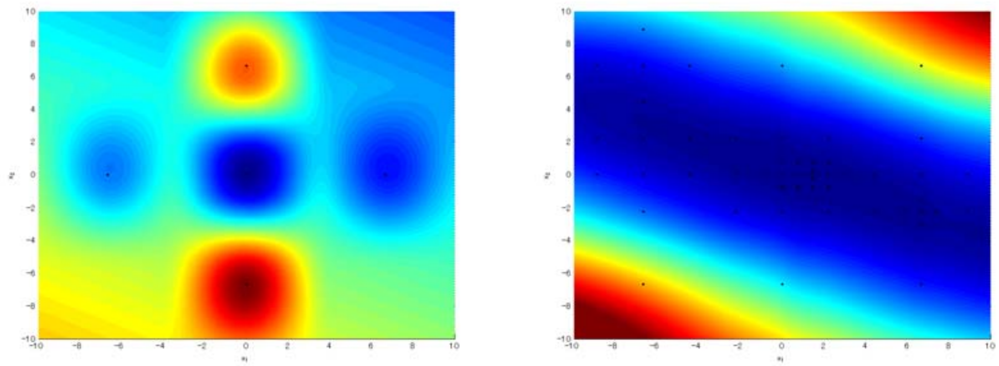



Fig. 1 Process configuration



(a) 5 points (b) 53 points

Fig. 2 Example of Kriging model for Branin function

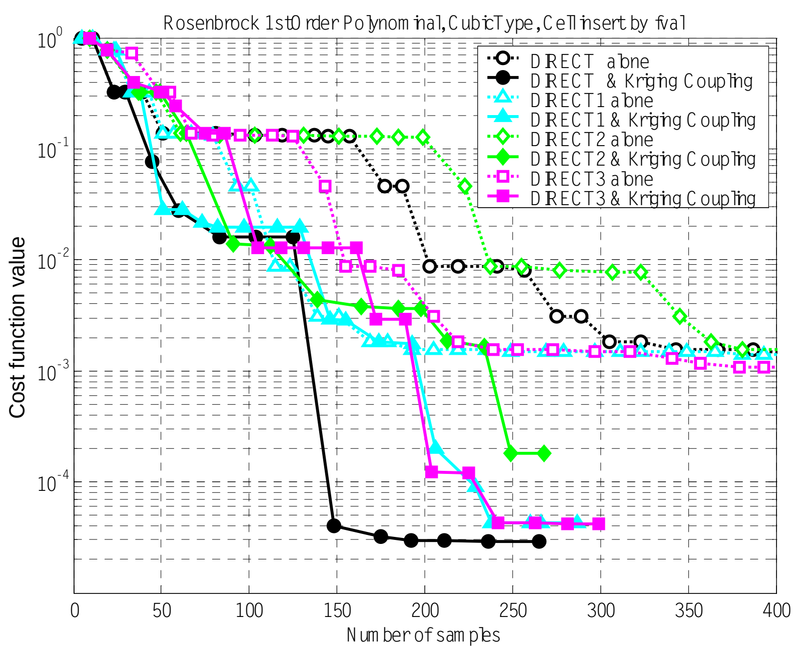



Fig. 3 2-D case (Rosenbrock function)

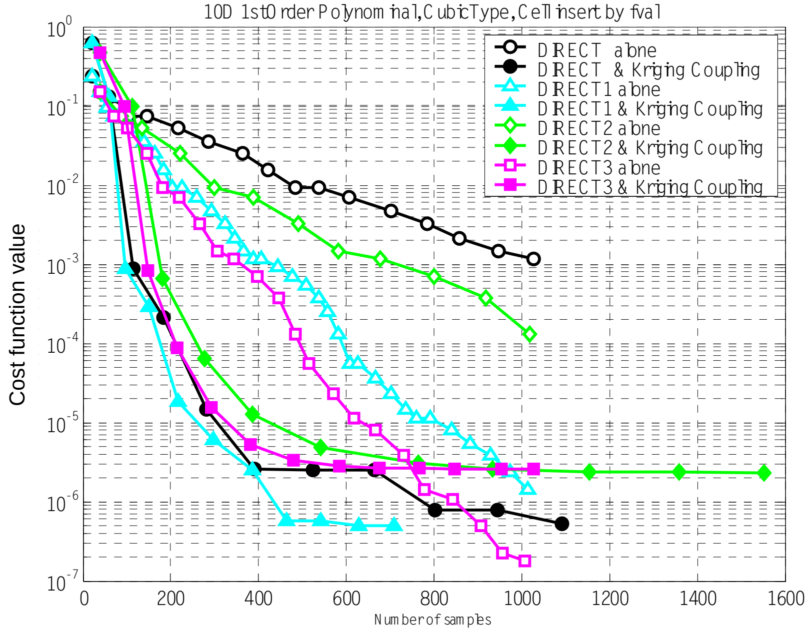



Fig. 4 10-D case

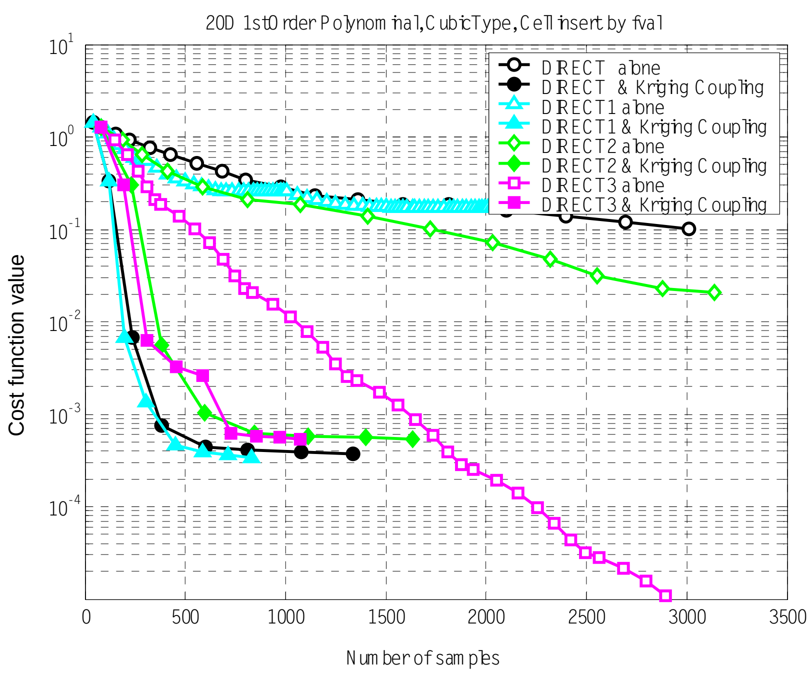



Fig. 5 20-D case

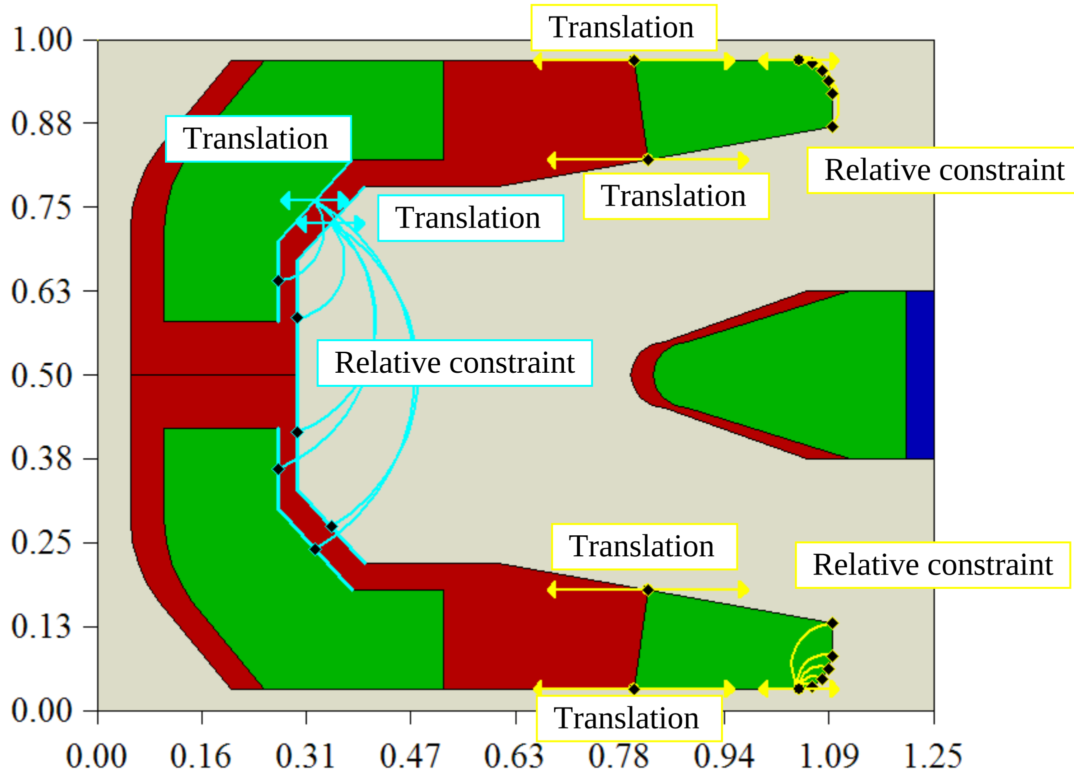



Fig. 6 INSIC 7nm flying height slider and constraints

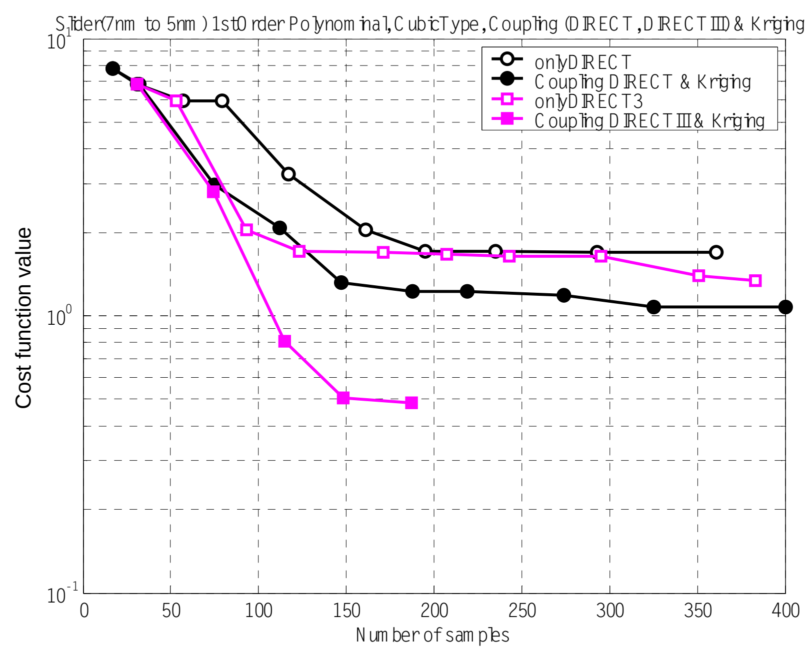



Fig. 7 Slider optimization (7nm FH to 5nm FH)

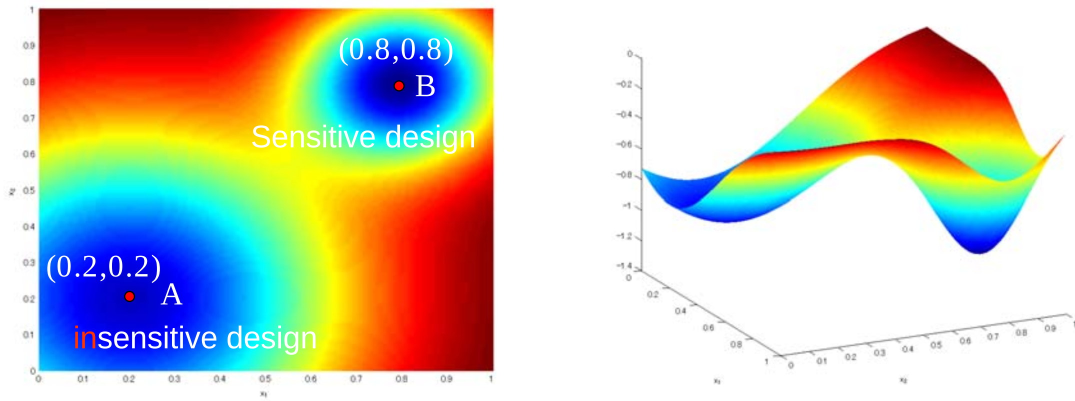



Fig. 8 Test function for sensitivity optimization

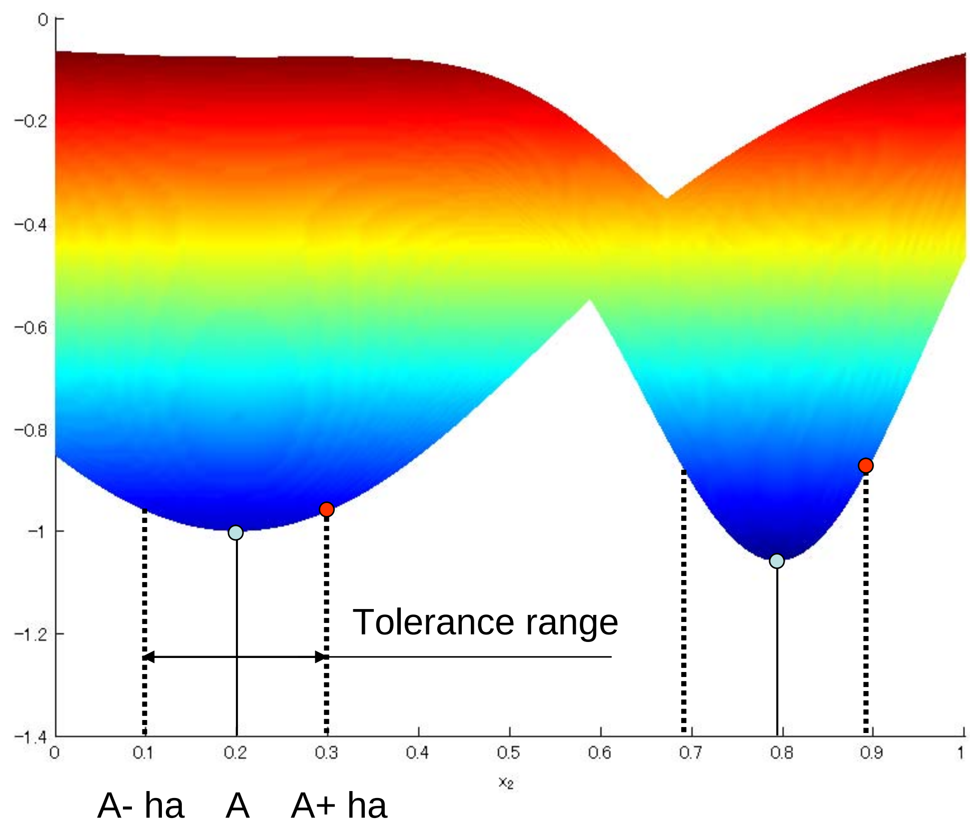
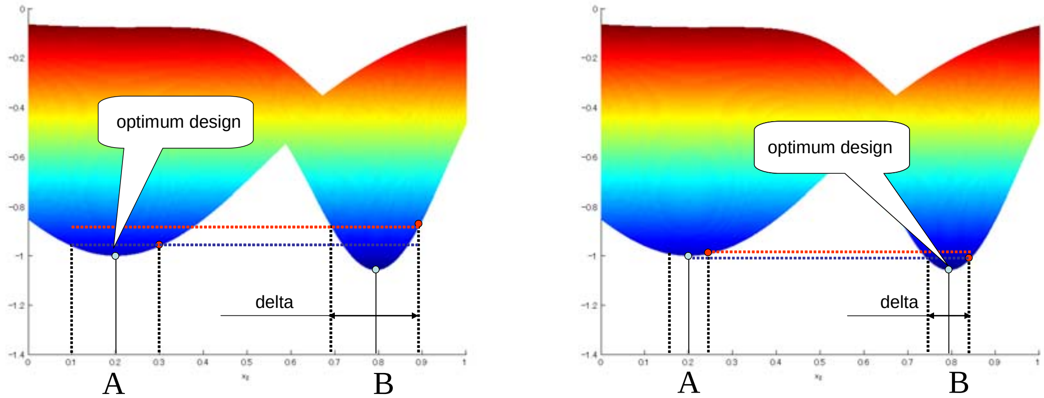
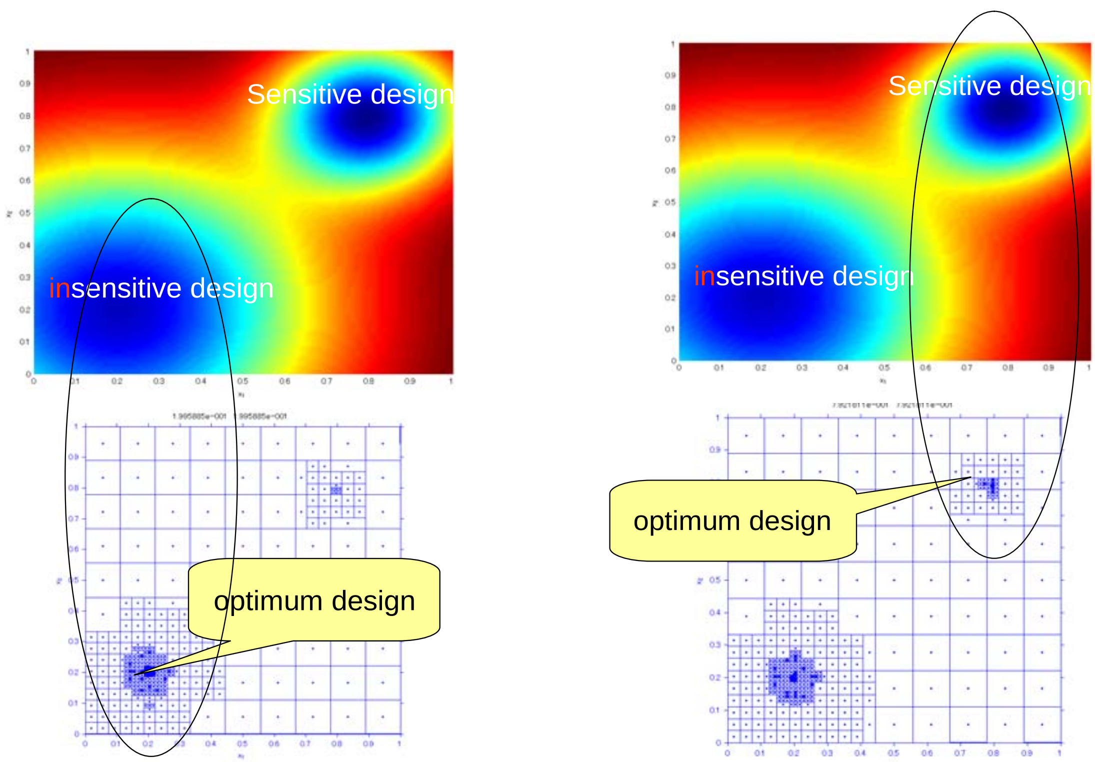



Fig. 9 Modified cost function value




(a) Wide tolerance (b) Narrow tolerance

Fig. 10 Relation between tolerance range and optimum design




(a) Wide tolerance (b) Narrow tolerance

Fig. 11 Optimum design