

## Design optimization using support vector regression

Yongbin Lee<sup>1</sup>, Sangyup Oh<sup>1</sup> and Dong-Hoon Choi<sup>2,\*</sup>

<sup>1</sup>Research Engineer, The Center of Innovative Design Optimization Technology (iDOT), Hanyang University,  
17 Haengdang-Dong, Sungdong-Gu, Seoul 133-791, Korea

<sup>2</sup>Director, The Center of Innovative Design Optimization Technology (iDOT), Hanyang University,  
17 Haengdang-Dong, Sungdong-Gu, Seoul 133-791, Korea

(Manuscript Received February 15, 2007; Revised June 23, 2007; Accepted August 3, 2007)

---

### Abstract

Polynomial regression (PR) and kriging are standard meta-model techniques used for approximate optimization (AO). Support vector regression (SVR) is a new meta-model technique with higher accuracy and a lower standard deviation than existing techniques. In this paper, we propose a sequential approximate optimization (SAO) method using SVR. Inherited latin hypercube design (ILHD) is used as the design of experiment (DOE), and the trust region algorithm is used as the model management technique, both adopted to increase efficiency in problem solving. We demonstrate the superior accuracy and efficiency of the proposed method by solving three mathematical problems and two engineering design problems. We also compare the proposed method with other meta-models such as kriging, radial basis function (RBF), and polynomial regression.

*Keywords:* Support vector regression; Trust region algorithm; Inherited latin-hypercube design; Sequential approximate optimization

---

### 1. Introduction

The use of meta-model techniques to improve the efficiency of design optimization is an area of active study [1]. Polynomial regression (PR) and kriging are representative meta-model techniques that are often applied to approximate optimization (AO). However, it is difficult to approximate highly non-linear systems using PR, and the technique is inefficient when the number of design variables is large. Kriging can approximate highly non-linear systems but demands the use of a global optimization procedure. Because of the difficulty of this procedure, kriging may create an inappropriate meta-model. SVR is an excellent technique in the approximation of a highly non-linear system [2, 3], and is easy to generate. But until now, the successful use of SVR has depended on the designer's definition of three dependent parameters. In

this study, appropriate and reasonable parameters are suggested that will allow designers to apply SVR to general problems without specific knowledge of the parameters.

Generally, an AO, which creates a single meta-model in the design space, cannot find the true optimum even if the meta-model describes the non-linearity of the system very well. Therefore, we use the sequential approximation optimization (SAO) method in which the accuracy of the detected optimum is progressively increased. We propose the SAO method using SVR with a set of suggested user-defined parameters as a meta-modeling technique. We also propose the trust region algorithm, which guarantees global convergence, as a model management technique [4, 5]. We adopt an inherited latin-hypercube design (ILHD) for the design of experiments (DOE) which reuses previous data, improving the efficiency of the SAO [6]. Finally, throughout several examples (three mathematical problems and two engineering design problems), we evaluate the

---

\*Corresponding author. Tel.: +82 2 2290 0478, Fax.: +82 2 2291 4070

E-mail address: dhchoi@hanyang.ac.kr

DOI 10.1007/s12206-007-1027-4

accuracy of the solution and the efficiency of the performance, comparing the proposed method with SAO by using other meta-models such as kriging, radial basis function (RBF), and PR.

## 2. Proposed method (SAO using SVR)

### 2.1 Design of experiment

The proposed SAO method uses optimal latin-hypercube design (OLHD) [7] in the first iteration and ILHD [6] in subsequent iterations as a DOE method.

OLHD is a method that combines optimal DOE and LHD [8] to overcome a weakness in LHD. LHD may create a design in which all of the points are located along the diagonal of the design space, resulting in an experimental design with poor statistical qualities. OLHD constructs an optimal design set as it controls the design points located randomly in the design space. Research on the proper optimal criteria and optimization method is in progress. Entropy [9], IMSE [10], maximin distance, minimax distance [11] and centered L2 discrepancy [12] are introduced as optimal criteria, and the SA algorithm, the CP algorithm, the ESE algorithm and GAs were used to optimize the design set.

ILHD reuses data by inheriting the sampling points of previous iterations for use in the current design space. The number of new sampling points and their positions are determined by the number of inherited sampling points and their positions. The combination of new and inherited sampling points keeps the structure of LHD.

### 2.2 Meta-model

The proposed SAO method uses support vector regression (SVR) as a meta-model.

#### 2.2.1 Support vector regression

The purpose of SVR is to find a flat function,  $f(x)$ , that has at most  $\varepsilon$  deviation from each of the targets of the training inputs [2]. For the linear case,

$$f(\mathbf{x}) = \langle \mathbf{w} \cdot \mathbf{x} \rangle + b \quad (1)$$

where  $\langle a \cdot b \rangle$  is the dot product of  $a$  and  $b$ . The flatness of the function in Eq. (1) is proportional to  $\mathbf{w}$ . One way to ensure a small value for  $\mathbf{w}$  is to minimize

the norm, i.e.,  $\|\mathbf{w}\|^2 = \langle \mathbf{w} \cdot \mathbf{w} \rangle$ . We can write this problem as a convex optimization problem:

$$\begin{aligned} & \text{Minimize} \quad \frac{1}{2} \|\mathbf{w}\|^2 \\ & \text{subject to} \quad \begin{cases} y_i - \langle \mathbf{w} \cdot \mathbf{x}_i \rangle - b \leq \varepsilon \\ \langle \mathbf{w} \cdot \mathbf{x}_i \rangle + b - y_i \leq \varepsilon \end{cases} \end{aligned} \quad (2)$$

where  $i$  is the number of sampling points. The assumption in Eq. (2) is that a function,  $f(x)$ , exists that approximates all pairs of sampling points,  $(\mathbf{x}_i, y_i)$ , with  $\varepsilon$  precision, which means that the convex optimization problem is feasible. Sometimes, the problem is not feasible, or we may want to allow for error. One can introduce slack variables,  $\xi_i, \xi_i^*$ , to cope with otherwise infeasible constraints of the optimization problem, Eq. (2). The introduction of slack variables gives Eq. (3):

$$\begin{aligned} & \text{Minimize} \quad \frac{1}{2} \|\mathbf{w}\|^2 + C \sum_{i=1}^l (\xi_i + \xi_i^*) \\ & \text{subject to} \quad \begin{cases} y_i - \langle \mathbf{w} \cdot \mathbf{x}_i \rangle - b \leq \varepsilon + \xi_i \\ \langle \mathbf{w} \cdot \mathbf{x}_i \rangle + b - y_i \leq \varepsilon + \xi_i^* \\ \xi_i, \xi_i^* \geq 0 \end{cases} \end{aligned} \quad (3)$$

It turns out that in most cases the optimization problem, Eq. (3), can be solved more easily in its dual formulation. In addition, the dual formulation provides a way to extend SVR to nonlinear functions. We will use a standard dualization method utilizing Lagrange multipliers:

$$\begin{aligned} L = & \frac{1}{2} \|\mathbf{w}\|^2 + C \sum_{i=1}^l (\xi_i + \xi_i^*) \\ & - \sum_{i=1}^l \alpha_i (\varepsilon + \xi_i - y_i + \langle \mathbf{w} \cdot \mathbf{x}_i \rangle + b) \\ & - \sum_{i=1}^l \alpha_i^* (\varepsilon + \xi_i^* - y_i + \langle \mathbf{w} \cdot \mathbf{x}_i \rangle - b) \\ & - \sum_{i=1}^l (\eta_i \xi_i + \eta_i^* \xi_i^*) \end{aligned} \quad (4)$$

$$\partial_b L = \sum_{i=1}^l (\alpha_i^* - \alpha_i) = 0$$

$$\partial_{\mathbf{w}} L = \mathbf{w} - \sum_{i=1}^l (\alpha_i^* - \alpha_i) \mathbf{x}_i = 0$$

$$\partial_{\xi_i} L = C - \alpha_i - \eta_i = 0$$

$$\partial_{\xi_i^*} L = C - \alpha_i^* - \eta_i^* = 0$$

Here  $L$  is the Lagrangian and  $\eta_i, \eta_i^*, \alpha_i, \alpha_i^*$  are the Lagrange multipliers. Hence, the dual variables in Eq. (4) have to satisfy positivity constraints. Eq. (3) becomes Eq. (5):

$$\begin{aligned} \text{Maximize } & -\frac{1}{2} \sum_{i,j=1}^l (\alpha_i - \alpha_i^*)(\alpha_j - \alpha_j^*) \langle \mathbf{x}_i, \mathbf{x}_j \rangle \\ & - \varepsilon \sum_{i=1}^l (\alpha_i + \alpha_i^*) + \sum_{i=1}^l y_i (\alpha_i - \alpha_i^*) \\ \text{subject to } & \sum_{i=1}^l (\alpha_i - \alpha_i^*) = 0 \\ & \alpha_i, \alpha_i^* \in [0, C] \end{aligned} \tag{5}$$

In Eq. (4),  $w$  can be formulated as follows:

$$w = \sum_{i=1}^l (\alpha_i^* - \alpha_i) \mathbf{x}_i. \tag{6}$$

So the linear regression model, Eq. (1), can be re-written as follows:

$$f(\mathbf{x}) = \sum_{i=1}^l (\alpha_i^* - \alpha_i) \langle \mathbf{x}_i, \mathbf{x} \rangle + b. \tag{7}$$

There are three advantages to transforming the optimization problem into dual form. First, the optimization problem becomes a quadratic programming problem with linear constraints and a positive definite Hessian matrix, thus ensuring a unique global optimum. Second, the input data always appear in the dot product so the calculation is easy, regardless of the dimension of the input vector. Third, non-linear function approximations can be easily achieved by exchanging the dot product of input vectors for the kernel function.

Table 1 lists common kernel functions. By applying one of the kernel functions listed in Table 1 to the dot product of the input vectors in Eq. (5), we obtain

Table 1. Kernel functions.

Linear	$k(\mathbf{x}, \mathbf{x}') = \mathbf{x}^T \mathbf{x}'$
Polynomial	$k(\mathbf{x}, \mathbf{x}') = \langle \mathbf{x}, \mathbf{x}' \rangle^d$
Gaussian	$k(\mathbf{x}, \mathbf{x}') = \exp\left(-\frac{\ \mathbf{x} - \mathbf{x}'\ ^2}{2\sigma^2}\right)$
Sigmoid	$k(\mathbf{x}, \mathbf{x}') = \tanh(\kappa \langle \mathbf{x}, \mathbf{x}' \rangle + \nu)$
Inhomogeneous Polynomial	$k(\mathbf{x}, \mathbf{x}') = (\langle \mathbf{x}, \mathbf{x}' \rangle + c)^d$

Eq. (8). Replacing the dot product in Eq. (7), the SVR model becomes Eq. (9). In this study, the SVR model is constructed by using the Gaussian kernel function, which is frequently used in the SVR technique.

$$\begin{aligned} \text{Maximize } & -\frac{1}{2} \sum_{i,j=1}^l (\alpha_i - \alpha_i^*)(\alpha_j - \alpha_j^*) k \langle \mathbf{x}_i, \mathbf{x}_j \rangle \\ & - \varepsilon \sum_{i=1}^l (\alpha_i + \alpha_i^*) + \sum_{i=1}^l y_i (\alpha_i - \alpha_i^*) \\ \text{subject to } & \sum_{i=1}^l (\alpha_i - \alpha_i^*) = 0 \\ & \alpha_i, \alpha_i^* \in [0, C] \end{aligned} \tag{8}$$

$$f(\mathbf{x}) = \sum_{i=1}^l (\alpha_i^* - \alpha_i) k \langle \mathbf{x}_i, \mathbf{x} \rangle + b \tag{9}$$

### 2.2.2 Determination of user-defined parameters in SVR

A problem with SVR is that its performance depends on the user’s knowledge of specific parameters. Most existing software implementations treat these parameters as user-defined inputs. There are no accepted criteria to decide their values, even though they have such a large effect on a model’s accuracy. In this study, three parameters ( $C$ ,  $\varepsilon$  and  $\sigma$ ) are used. First,  $\sigma$  represents the kernel width. With large  $\sigma$  values, SVR behaves similarly to a second order polynomial regression. Second,  $\varepsilon$  is the error range allowed by the user. The points lying outside the  $\varepsilon$  tube are called support vectors, because these establish the fundamentals of the estimated regression model. The parameter  $\varepsilon$  regulates the radius of the  $\varepsilon$  tube around the regression model, and thus the number of support vectors that will be selected to

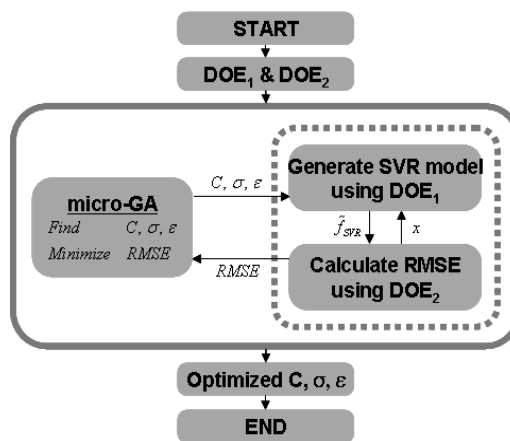


Fig. 1. Algorithm for parameter optimization.

construct the regression function. Choosing a value of  $\varepsilon$  that is too large results in fewer support vectors and a smoother regression model. Third,  $C$  determines the trade-off between the smoothness and the deviations of the regression model.

The algorithm used in the optimization of these parameters is shown in Fig. 1. To construct the SVR model, the dual form optimization problem was solved using DOT (SQP) [13], and the parameter optimization problem was solved by the *micro-Genetic Algorithm* ( $\mu$ -GA). In Fig. 1,  $DOE_1$  is the set of sampling points used to construct the SVR model, and  $DOE_2$  is the set of sampling points used to estimate the accuracy of the model. The design variables in the parameter optimization problem are  $\sigma$ ,  $\varepsilon$  and  $C$ . The objective is to minimize the *RMSE* (*Root Mean Square Error*) value, which can be obtained by Eq. (10):

$$RMSE = \sqrt{\frac{\sum_{i=1}^{n_{DOE_2}} (f_i - \tilde{f}_i)^2}{n_{DOE_2}}} \quad (10)$$

It is difficult to use practical problems to perform the optimization because  $DOE_2$  needs many design points to estimate the accuracy of the SVR model. To overcome this difficulty, we used mathematical problems. Table 2 shows the functions used, and Table 3 shows the optimum parameter values of each mathematical function found by applying the algorithm depicted in Fig. 1.

Even though the parameter values found in each example differ, Table 3 shows a trend in the optimized parameters. Each parameter has a tendency to take a specified value:  $\sigma$  takes values near 0.5,  $\varepsilon$  takes values near 0.0, and  $C$  takes values between  $10e+7$  and  $10e+8$ . In light of these results, we chose fixed parameters as follows:  $\sigma = 0.5$ ,  $\varepsilon = 0$ ,  $C = 10e+8$ .

Table 2. Mathematical functions used in parameter optimization

A	2D Multi-modal Problem	$f(x) = x_1 x_2 \sin(x_1) + 0.1x_1^2 + x_1 - 1.5x_2$
B	Six hump	$f(x) = 4x_1^2 - 2.1x_1^4 + \frac{1}{3}x_1^6 + x_1 x_2 - 4x_2^2 + 4x_2^4$
C	Rosenbrock	$f(x_1, x_2) = (x_2 - x_1 + 1)^2 + (1 - x_1)^2$
D	Goldstein and Price Function	$f(x) = [1 + (x_1 + x_2 + 1)^2(19 - 14x_1 + 3x_1^2 - 14x_2 + 6x_1 x_2 + 3x_2^2)] \times [30 + (2x_1 - 3x_2)^2(18 - 32x_1 + 12x_1^2 - 48x_2 - 36x_1 x_2 + 27x_2^2)]$
E	Branin Function	$f(x) = a(x_2 - b x_1^2 + c x_1 - d)^2 + e(1 - g) \cos(x_1) + e$ where $a = 1, b = \frac{5.1}{4\pi^2}, c = \frac{5}{\pi}, d = 6, e = 10, g = \frac{1}{8\pi}$
F	2D- Problem	$f(x) = 10^6 x_1^4 + x_2^2 - (x_1^4 + x_2^2)^2 + 10^6 (x_1^4 + x_2^2)^4$ where $k = 5, l = -5$

Fig. 2 compares the *RMSE* values of the SVR model using the optimal parameters shown in Table 3 with the proposed, fixed parameters. We can see that using the optimized parameters results in lower *RMSE* values. This means that using the optimal parameters allows us to construct a better SVR model. The differences in the *RMSE* values found with the fixed parameters and those found with the optimized parameters are relatively small, so the accuracy of the SVR model using the fixed parameters is acceptable in the SAO process. It makes sense to adopt an SVR with fixed parameters because the process of optimizing the parameter values at each iteration is time-consuming.

### 2.3 Model management technique

The proposed SAO method uses the trust region algorithm as a model management technique.

Newton's method cannot be applied when a function with many local minima is approximated by Taylor series to the quadratic function,  $q(\delta_k)$ , and then optimized, but the trust region algorithm can be used. If the region where  $q(\delta_k)$  can be approximated by  $f(x_k + \delta_k)$  is  $\Gamma^k = \{\delta_k : \|\delta_k\| \leq h^k\}$ , the new design point,  $x_{k+1} (x_{k+1} = x_k + \delta_k)$ , is determined by solving the optimization problem as in Eq. (11):

Table 3. Results of parameter optimization.

	A	B	C	D	E	F
<i>Sigma</i>	0.5083	0.4872	0.4831	0.4843	0.5160	0.5123
<i>C</i>	5.65E+07	8.20E+07	3.18E+07	6.80E+07	1.18E+07	7.32E+07
<i>Epsilon</i>	0.00021	0.00016	0.00007	0.00087	0.00082	0.00089

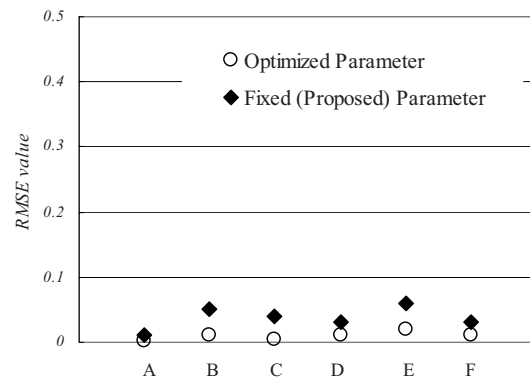


Fig. 2. Comparison of *RMSE* values between fixed parameters and optimized parameters.

$$\begin{aligned} \text{Minimize } q(\delta_k) &= \nabla f(\mathbf{x}_k)^T \delta_k + \frac{1}{2} \delta_k^T \mathbf{G}_k \delta_k \quad (11) \\ \text{subject to } \|\delta_k\| &\leq h^k \end{aligned}$$

The trust region algorithm is used to solve the above optimization problem iteratively. The algorithm controls  $h^k$  to prevent a slow convergence rate. The true decrement of  $f$  at the  $k^{\text{th}}$  iteration is defined as Eq. (12) and the predicted decrement is defined as Eq. (13). Then the trust region ratio is defined as Eq. (14).

$$\Delta f^k = f(\mathbf{x}_k) - f(\mathbf{x}_k + \delta_k) \quad (12)$$

$$\Delta q^k = q(0) - q(\delta_k) = f(\mathbf{x}_k) - q(\delta_k) \quad (13)$$

$$r^k = \Delta f^k / \Delta q^k \quad (14)$$

The trust region ratio,  $r^k$ , is the measure of the fidelity of  $q(\delta_k)$ . If  $r^k$  is close to ‘1’,  $q(\delta_k)$  is a high-fidelity model. The trust region algorithm decreases or increases  $h^k$  to maintain the fidelity of  $q(\delta_k)$ . This feature makes the trust region algorithm superior to other model management techniques, which exponentially limits the change in design variables. Fletcher [4] and Powell [5] proved that the trust region algorithm guarantees a global convergence.

### 3. Computational procedure of the proposed method

The computational procedure of the proposed method follows:

*Step 0.* Set the initial trust region to include the initial design. Set  $k=1$ . Select  $2n+1$  sampling points by OLHD, where  $n$  is the number of design variables.

*Step 1.* Evaluate the real function values at each sampling point. Then construct the approximate objective and constraints using the SVR model with the fixed parameters.

*Step 2.* Perform the approximate optimization using the approximate objective and constraints in the trust region.

*Step 3.* Evaluate the actual and the predicted reductions such as  $\Delta f^k$  and  $\Delta q^k$  using Eq. (12) and Eq. (13), and estimate the reduction ratio,  $r^k$ , using Eq. (14). If  $r^k < 0.25$  or  $1/0.25 < r^k < \infty$ , set  $\Gamma^{k+1} = \Gamma^k / 4$ . If  $0.25 < r^k < 0.75$  or  $1/0.75 < r^k < 1/0.25$ , set  $\Gamma^{k+1} = \Gamma^k$ . If  $0.75 < r^k < 1/0.75$  and  $\|x_k^0 - x_k^*\| = \Gamma^k$ , then set  $\Gamma^{k+1} = 2\Gamma^k$ , otherwise set  $\Gamma^{k+1} = \Gamma^k$ . If  $r^k > 0$ , then update  $x_{k+1}^0 = x_k^*$  and go to *Step 4* with  $k=k+1$ . Oth-

erwise, select new sampling points using ILHD around  $x_k$  in the reduced trust region and go back to *Step 1*.

*Step 4.* If  $\Delta f^k \leq \varepsilon_1$ ,  $\Delta f^k / f^k \leq \varepsilon_2$  or  $\|x_k^0 - x_k^*\| / x_k^* \leq \varepsilon_3$ , then stop. Otherwise, select new sampling points by ILHD around  $x_{k+1}$  in the new trust region  $\Gamma^{k+1}$ , and go back to *Step 1*.

In *Step 2*, the constrained optimization problem was solved using the *micro-Genetic Algorithm* and the *MMFD (Modified Method of Feasible Directions)* in DOT version 5.0 [13]. The constants 0.25 and 0.75 (*Step 3*) are commonly used in trust region algorithms, which are known to be insensitive to changes in their values, so the values were not changed in this study.

### 4. Applications

To show the performance of the proposed method, three mathematical problems and two engineering design problems are solved in this section. In all examples, the results of the proposed method are compared with the results of SAO methods using other meta-models such as kriging, RBF, and PR. Here, kriging used the constant global model and the Gaussian correlation function. RBF used the Gaussian basis function as follows:

$$\phi(z) = \exp\left(\frac{-z^2}{c}\right), \quad c = \frac{2.5}{\sqrt[n]{N}}$$

where  $n$  is the number of design variables and  $N$  is the number of sampling points. PR used the least square method with full quadratic coefficients.

#### 4.1 Mathematical problems

Three mathematical optimization problems and their results are represented in Table 4 and Table 5, respectively. In two of the problems, all methods found the optimum value as shown in Table 5, but in the example of the Haupt function, SAO using RBF and PR converged to a local optimum. The proposed method showed the smallest number of function calls in all cases, as shown in Table 5. These results demonstrate both the accuracy and efficiency of the proposed method.

#### 4.2 Coil spring design problem

In this problem, the objective is to minimize the

weight of the coil spring under tensile and compressive forces. The coil spring is shown in Fig. 3. The constraints on the system are minimum displacement, transfer stress, surge frequency, and the limit of the spring's external diameter. The design variables are the wire's diameter, the coil diameter, and the number of coils. The optimization formula is Eq. (15).

Table 4. Mathematical problems.

Goldstein and Price Function	$\min. f(\mathbf{x}) = [1 + (x_1 + x_2 + 1)^2(19 - 14x_1 + 3x_1^2 - 14x_2 + 6x_1x_2 + 3x_2^2)] \times [30 + (2x_1 - 3x_2)^2(18 - 32x_1 + 12x_1^2 - 48x_2 - 36x_1x_2 + 27x_2^2)]$ $-2 \leq x_1, x_2 \leq 2$
Haupt Function	$\min. f(\mathbf{x}) = x_1 \sin(4x_2) + 1.1x_2 \sin(2x_1)$ $-4 \leq x_1, x_2 \leq 4$
Fenton and Eason's Function	$\min. f(\mathbf{x}) = \left( \frac{1}{10} \right) \left( 12 + x_1^2 + \frac{1 + x_2^2}{x_1^2} + \frac{x_1^2 x_2^2 + 100}{(x_1 x_2)^4} \right)$ $\text{s.t. } g_1 = -\frac{5}{3}x_1 - x_2 + 5 \leq 0$ $0.01 \leq x_1, x_2 \leq 10$

Table 5. Results of mathematical examples.

Goldstein and Price	initial	Kriging	RBF	PR	SVR
x1	0.0000	-0.0081	-0.0002	0.0000	0.0000
x2	0.0000	-1.0030	-1.0000	-1.0000	-0.9997
Obj.	600.0000	3.0152	3.0000	3.0000	3.0002
No. of function call		69	108	55	41

Fenton & Eason	initial	Kriging	RBF	PR	SVR
x1	10.0000	1.8140	1.7703	1.7638	1.7750
x2	10.0000	1.9745	2.0475	2.0620	2.0670
Obj.	113.0100	17.4650	17.4460	17.4460	17.4500
g1	-21.6000	0.0021	0.0020	0.0000	-0.0260
No. of function call		73	43	38	26

Haupt function	initial	Kriging	RBF	PR	SVR
x1	0.0000	-2.7700	-1.2280	-1.2260	-2.7650
x2	0.0000	-2.4590	2.4545	2.4500	-2.4580
Obj.	0.0000	-5.4080	-3.8516	-3.8512	-5.4071
No. of function call		38	46	69	38

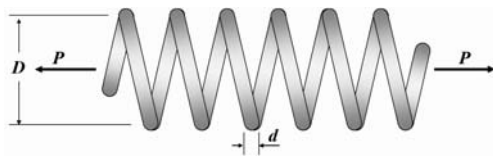


Fig. 3. Coil spring problem.

$$\text{Minimize } f(\mathbf{x}) = (x_3 + 2)x_2x_1^2$$

$$\text{subject to } g_1(\mathbf{x}) = 1 - \frac{x_2^3x_3}{71875x_1^4} \leq 0$$

$$g_2(\mathbf{x}) = \frac{4x_2^2 - x_1x_2}{12566(x_2x_1^3 - x_1^4)} + \frac{1}{5108x_1^2} - 1 \leq 0$$

$$g_3(\mathbf{x}) = 1 - \frac{140.45x_1}{x_2^2x_3} \leq 0$$

$$g_4(\mathbf{x}) = \frac{(x_1 + x_2)}{1.5} - 1 \leq 0$$

$$0.005 \leq x_1 \leq 5, \quad 0.005 \leq x_2 \leq 5, \quad 1 \leq x_3 \leq 15$$

(15)

The results are shown in Table 6 and Fig. 4. The proposed method finds that the best solution (*the objective*) and the efficiency (*the number of function calls*) of the proposed method are better than those of the SAO method using other meta-models.

Table 6. Results of Coil Spring design problem.

Coil Spring design	initial	Kriging	RBF	PR	SVR
x1	0.2000	0.0730	0.0686	0.0638	0.0560
x2	1.3000	0.9749	0.9172	0.7238	0.4660
x3	2.0000	3.1160	10.9100	12.5712	8.6000
Obj.	0.2080	0.0265	0.0557	0.0429	0.0155
g1	0.9620	-0.4170	-4.3000	-3.0200	-0.2320
g2	-0.9360	-0.1160	0.0020	0.0021	-0.0069
g3	-7.3100	-2.4600	-0.0492	-0.3559	-3.2100
g4	0.0000	-0.3010	-0.3430	-0.4750	-0.6520
No. of function call		116	236	236	79

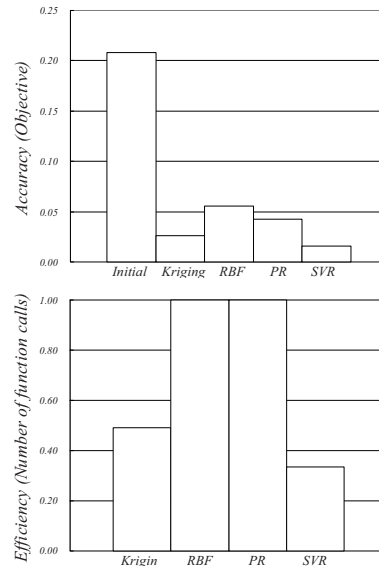


Fig. 4. Comparison of accuracy and efficiency.

Table 7. Results of SRM design problems.

SRMdesign	initial	Kriging	RBF	PR	SVR
$x1$	25.2000	4.8425	20.0400	17.0210	13.5600
$x2$	45.1000	51.7500	53.9400	53.7850	49.3400
$x3$	30.0000	30.9250	37.2100	44.4400	50.7300
Obj.	-17.1000	-0.5540	-0.2910	-0.2833	-0.3040
$g1$	3.9300	2.2400	-0.0001	-0.4600	-0.0940
$g2$	-0.3300	0.6630	-0.2800	-0.2510	-0.2570
No. of function call		183	130	222	64

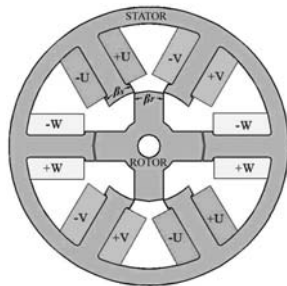


Fig. 5. Cross section of SRM.

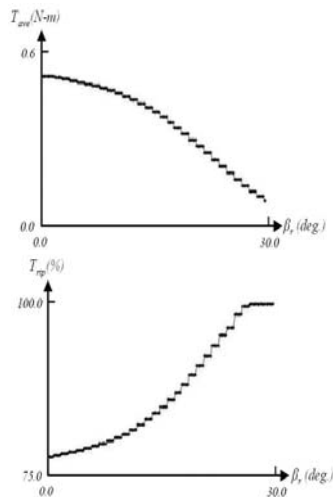


Fig. 6. Torque curve with respect to the rotor pole arc.

### 4.3 Switched reluctance motor (SRM) design problem

The SRM design problem [14] is solved in this section. The shape of SRM is shown in Fig. 5. The design variables are the switching-on angle, the switching-off angle, and the rotor pole arc. The goal is to **maximize** the average torque, which is subject to torque ripple and the maximum current phase. The initial design is given as  $\mathbf{x} = [25.2 \ 45.1 \ 30.0]^T$ .

Eight minutes of computation time were required to perform each analysis (Pentium 4 CPU 3.2 GHz).

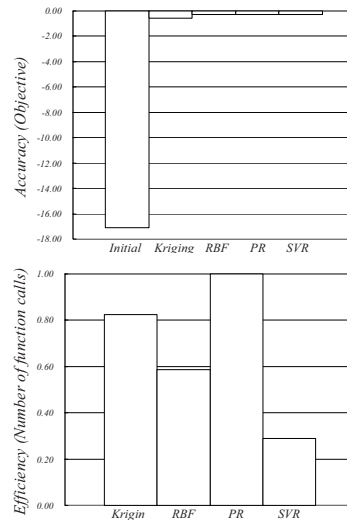


Fig. 7. Comparison of accuracy and efficiency.

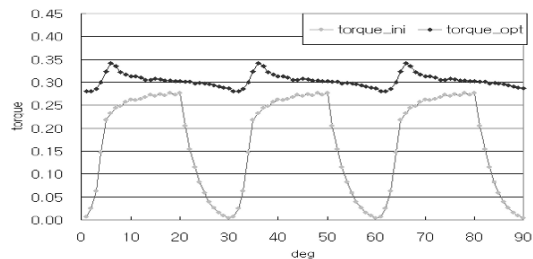


Fig. 8. Performance of the SRM for the initial and optimal design.

Fig. 6 shows the average torque,  $T_{ave}$ , and the torque ripple curve,  $T_{rip}$ , with respect to the rotor pole arc,  $\beta_r$ , given a fixed switching-on angle,  $\theta_{on}$ , and switching off angle,  $\theta_{off}$ . As shown in Fig. 6, this problem cannot be solved with the gradient-based optimization methods because each curve is a step function.

Our results are shown in Table 7 and Fig. 7. They demonstrate the superiority of the proposed method in both accuracy (*the objective*) and efficiency (*the number of function calls*). Fig. 8 shows the torque curve for the initial design and the optimal design. We can see that the average torque increased and torque ripple decreased considerably.

## 5. Conclusions

In this study, we suggested a sequential approximate optimization method using support vector regression, a new meta-model technique. We overcame the SVR method’s dependency on a user’s specific knowledge by fixing the user-defined parameters. We

adopted the inherited latin-hypercube design, which satisfies the space-filling concept and reuses a previous data set to increase the efficiency of the optimization process. In addition, we proposed a method which can guarantee the global convergence of SAO by using the trust region algorithm. Finally, we developed a program to embody the proposed method. The superiority of the proposed method is demonstrated in three mathematical problems and two engineering design problems. The results show that in all problems, the proposed method found the exact solution and performed about 26~63% more efficiently than the other SAO methods.

In the future, we would like to show that the DOE and model management technique is appropriate to each meta-model by conducting a comparative study of these components of the SAO.

### Acknowledgments

This work was supported by the center of innovative Design Optimization Technology (iDOT), Korea Science and Engineering Foundation, and the Second Brain Korea 21 Project in 2006.

### References

- [1] T. W. Simpson, J. Peplinski, P. N. Koch and J. K. Allen, Metamodels for computer based engineering design: survey and recommendations, *Engineering with Computers* 17 (2) (2001) 129-150.
- [2] Alex J. Smola and Bernhard Scholkopf, A tutorial on support vector regression, *Statistics and computing* 14 (3) (2004) 199-222.
- [3] Stella M. Clarke, an H. Griebisch, Timothy W. Simpson, Analysis of Support Vector Regression For Approximation of Complex Engineering Analyses, *Journal of mechanical design* 127 (6) (2005) 1077-1087.
- [4] R. Fletcher, An Algorithm for Solving Linearly Constrained Optimization Problems, *Mathematical Programming* 2 (1) (1972) 133-165.
- [5] M. J. D. Powell, Convergence Properties of a Class of Minimization Algorithms, *Nonlinear Programming* 2, Academic Press, New York, USA, (1975) 1-27.
- [6] G. G. Wang, Adaptive Response Surface Method Using Inherited Latin Hypercube Design Points, *Transactions of the ASME* 125 (2) (2003) 210-220.
- [7] J. S. Park, Optimal Latin-hypercube Designs for Computer Experiments, *Journal of Statistical Planning and Inference* 39 (1) (1994) 95-111.
- [8] M. H. McKay, R. J. Beckman and W. J. Conover, A Comparison of Three Methods for Selecting Values of Input Variables in the Analysis of Output from a Computer Code, *Technometrics* 21 (2) (1979) 239-245.
- [9] M. C. Shewry, Maximum Entropy Sampling, *Journal of Applied Statistics* 14 (1987) 165-170.
- [10] J. Sacks, W. J. Welch, T. J. Mitchell and H. P. Wynn, Design and Analysis of Computer Experiments, *Statistical Science* 4 (4) (1989) 409-435.
- [11] M. E. Johnson, L. M. Moore and D. Ylvisaker, Minimax and Maximin Distance Design, *Journal of Statistical Planning and Inference* 26 (2) (1990) 131-148.
- [12] K. T. Fang, C. X. MA and P. Winker, Centered  $L_2$ -Discrepancy of Random Sampling and Latin Hypercube Design, and Construction of Uniform Designs, *Mathematics of Computation* 71 (237) (2000) 275-296.
- [13] G. N. Vanderplaats, Dot Users Manual Version 5.0, VR&D, Colorado, USA, (1999) 12-111.
- [14] Youn-Hyun Kim, Kyung-Jin Hong and Dong-Hoon Choi, Optimal design of switched reluctance motor using two-dimensional finite element method, *Journal of Applied Physics* 91 (10) (2002) 6967-6969.