

## 공간선형모형을 이용한 전산실험의 분석\*

박정수<sup>1</sup>

### 요 약

오늘날 과학 및 공학연구를 위하여 복잡한 컴퓨터 시뮬레이션을 자주 사용하고 있다. 전통적 물리적 실험이 입력수의 한계와 출력의 측정오차를 가지는 반면 시뮬레이션 실험 또는 전산실험은 입력수에 제한이 거의 없고 대부분 출력 또한 측정오차가 없거나 매우 작다. 이러한 전산 실험의 분석을 위해 비교적 새롭게 등장한 통계적 방법이 DACE(Design and Analysis of Computer Experiments)이다. 본 발표에서는 이 DACE의 기본 개념과 원리를 간단히 소개하고, 실제로 기계설계에 효과적으로 적용된 사례를 보이고, 향후 연구방향 등을 논의하겠다. 특히 시뮬레이션 코드에 존재하는 절대상수의 추정문제(calibration)와 공간선형 모형의 최적선택을 위한 알고리즘을 소개한다.

### 1. 공간선형모형과 전산실험

오늘날 과학 및 공학연구를 위하여 복잡한 컴퓨터 시뮬레이션을 자주 사용하고 있다. 전통적 물리적 실험이 입력수의 한계와 출력의 측정오차를 가지는 반면 시뮬레이션 실험 또는 전산실험은 입력 수에 제한이 거의 없고 대부분 출력 또한 측정오차가 없거나 매우 작다. 이러한 전산 실험의 분석을 위해 비교적 새롭게 등장한 통계적 방법이 DACE(Design and Analysis of Computer Experiments) 이다. 본 논문에서는 이 DACE의 기본 개념과 원리를 소개하고, 실제로 기계설계에 효과적으로 적용된 사례를 보이고, 모형의 효과적인 선택문제와 절대상수의 추정문제에 대해 논의한다.

DACE는 기존의 물리적 실험에 사용된 통계적 방법인 반응표면방법(response surface methodology; RSM)을 전산실험에 적용하기 위해 고안된 방법으로서 크게 실험계획과 회귀분석이라는 두 단계로 정리된다. 전산실험에서는 많게는 수백 개의 입력변수가 필요하며 출력에 측정오차가 없거나 매우 작기 때문에, 기존의 실험계획에서 필요한 반복에 의한 오차추정이 필요 없게 되어서, 라틴 하이퍼큐브 실험계획(Latin-hypercube Design; LhD)이 널리 쓰이게 된다. LhD 는 생성하기 쉽고 적용할 수 있는 상황이 넓기 때문에 대부분의 전산실험에 매우 널리 사용되고 있다. LhD 외에도 maximin 거

\*This work was supported by the Korea Research Foundation Grant funded by the Korean Government(MOEHRD) (KRF-2005-202-C00072).

<sup>1</sup>500-757 광주광역시 북구 용봉동 300번지, 전남대학교 자연대 통계학과 교수.  
전화: 062-530-3445, E-mail: jspark@chonnam.ac.kr

리 계획[16]이나 최적실험계획들이 고안되었고, 특히 최적 LhD의 생성과 사용에 관한 연구가 보고되고 있다[1][2][8][17]. 이러한 실험계획에 바탕하여 전산실험이 이루어진 다음 얻게 되는 출력에 대한 통계적 분석에는 다음의 가우시안 회귀모형 또는 공간선형모형의 사용이 주장되었다[11].

$$Y(x) = \sum_{k=0}^d \beta_k f_k(x) + Z(x)$$

앞부분은 전통적인 회귀모형이고  $Z(x)$  는 가우시안 확률과정인데, 공분산 함수는

$$Cov(x_u, x_v) = \sigma^2 \exp - \sum_{i=1}^d \theta_i (x_{ui} - x_{vi})^2$$

형태를 취한다. 이 모형의 장점은 관찰값들을 내삽(interpolation)하기 때문에 측정오차없이 관찰된 출력을 분석하는데 매우 적절하다. 일단 자료가 관찰되면 이 모형에 사용된 모수들( $\beta_k, \theta_i, \sigma^2$ )을 최우추정법(MLE)으로 추정하고 이를 위 모형에 대입한 다음 미관측점에서의 최적예측  $Y_p(x)$  을 사용한다. 이 통계적 예측을 보통 크리깅(Kriging)이라고 부르는데, 그래서 DACE 방법을 Kriging 모델링이라고도 한다.

충분히 많은 실험으로부터 자료가 얻어진 경우 이 최적예측모형이 기존의 시뮬레이터를 대체할 수 있는 에뮬레이터 또는 metamodel 로 사용될 수 있다. 이 메타모델(또는 compact 모델, Kriging 모델, DACE 모델, surrogate 라고 불림) 을 이용하여 시뮬레이터에 대한 민감도분석, 최적점 파악, 실제자료와의 타당성 검사, 신뢰성분석 등 다양한 목적을 비교적 쉽게 수행할 수 있게 된다. 최근 출간된 단행본[12]에 DACE 에 대한 전반적인 소개와 구체적 공식 및 방법들을 볼 수 있다. 또 Kriging 모델 또는 DACE 방법이 실제로 MDO(Multidisciplinary Design Optimization)를 포함한 공학 연구 분야에 어떻게 적용되는지 그 조사와 추천 등에 관해서 Fang[18]과 Simpson[19][20]에서 볼 수 있다. 본 연구자 이외의 국내 관련 연구자의 연구는 [3][9][10][17]을 들 수 있다.

## 2. Dace 모형과 단계적 모형선택 알고리즘

Dace 모형에서 공분산함수의  $\theta$  들이 모두 같다는 제약조건을 “공통  $\theta$ ”라고 하자. 모두 다른  $\underline{\theta} = \theta_1, \theta_2, \dots, \theta_d$  인 경우를 “모든  $\theta$ ” 라고 하자. 또 모두 다른  $\underline{\beta} = \beta_0, \beta_1, \dots, \beta_d$  인 경우를 “모든  $\beta$ ” 라고 하자. 일반적으로 생각할 수 있는 전형적인 모형은 아래와 같다.

모형 1:  $\beta_0$  와 공통  $\theta$  로 된 모형

모형 2:  $\beta_0$  와 모든  $\theta$  로 된 모형

모형 3: 모든  $\beta$  와 공통  $\theta$  로 된 모형

모형 4: 모든  $\beta$  와 모든  $\theta$  로 된 모형

이러한 모형을 본 연구자는 이미 이용해 보았으나 이들 모형의 한계를 알고 최적 Dace 모형의 선택 필요성을 제기한바 있다[4]. 실제로 [5]에서는 어떤 모형을 선택해서 사용하였으나 최적 선택 알고리즘에 의한 것이 아니었다. 위의 모형들이 독립변수들과 종속변수간의 관계를 최적으로 설명하는 Dace 회귀모형이 아니기 때문에 우리는  $\beta$  와  $\underline{\theta} = \theta_1, \theta_2, \dots, \theta_d$  에서 선택된 일부의 조합으로 이루어지는 최적 Dace 모형을 구축하고자 한다. 그러면 최적 조합을 어떻게 찾을 것인가? 우리는 이를 위한 효과적인 알고리즘을 개발해야 한다.

전통적 회귀모형에서처럼 모든 가능한 조합이나 전진선택법이나 후진제거법 또는 단계적선택법이 가능하겠으나, 이 방법들을 Dace 모형에 적용하는데는 계산시간이 매우 오래 걸리는 매우 큰 단점이 생긴다. 왜냐하면  $\beta$  와  $\underline{\theta}$  를 추정하는데 사용되는 최우추정법은 수치적 반복 알고리즘을 사용하기 때문인데, 이는  $d^3 n^3$  차수( $n$ =표본의 크기,  $d$ =최적화의 차원)에 비례하는 계산속도를 따른다[3][7]. 한편 Welch 등[14]에 의해 전진선택법이 제안되었다. 이들은 우도비검정을 이용하여  $\theta$  를 하나씩 선택해나가는 알고리즘을 적용하여, 독립변수들을 선별하는 것에 목적을 두었다. 그러나 이들의 방법은 낮은 차원의  $d$  경우에만 적용가능하고 높은 차원의  $d$  에의 적용에는 계산상의 한계가 생긴다. 우리는 본 논문에서  $d$  가 30 이상인 높은 차원의 경우에 최적 모형을 선택하는 알고리즘을 개발한다. 최우추정치를 이용하는데 있어서  $\underline{\theta}$  가 주어지면  $\widehat{\beta}$  는 쉽게 구해지나 그 반대는 성립하지 않으므로, 최적화의 차원은  $\beta$  의 개수는 거의 영향을 미치지 않으며 영이 아닌  $\underline{\theta}$  의 개수로 정해진다. 따라서  $\underline{\theta}$  의 개수를 최소로 조절하면서 최적모형을 찾아가는 알고리즘이 효율적이다.

아래 알고리즘의 주요개념은 공통  $\theta$ 와 유사 t-검정의 사용이다. 또한 최적선형불편예측의 계산을 위해서 공분산  $V$  행렬의 역행렬을 계산하는데, 이때  $V$  는 자료의 수  $n$ 에 대해서  $n*n$  차원의 양정치 행렬이 된다. 이의 계산은 보통 출레스키 분해를 이용하고, 최적예측에는 최소제곱 추정치가 일부 쓰이는데, 이때에는 QR 분해가 사용된다[13]. 유사 t-검정(또는 Wald 검정)에서 사용되는 분모항 즉,  $\widehat{\beta}_j$  및  $\widehat{\theta}_k$  의 표준편차는 피트정보행렬의 역행렬의 대각선항의 제곱근을 사용한다. 본 논문에서는 우선 다음 두 개의 알고리즘(A, B)을 제안한다.

#### 알고리즘 A

[초기화]:  $J_0 = 0, 1, \dots, d$

[1단계]: 공통  $\theta$  제약조건과 모든  $\beta$  상태에서, 최우추정법으로  $\widehat{\beta}_j, j \in J_0$  를 구한다. 유사 t-검정으로  $\beta_j = 0$  이라는 가설을 채택하는  $j$  들을  $J_0$  로부터 제거한다.  $\beta_0$  는 제거하지 않는다.

[2단계]: [1단계]를 반복하여  $\beta_j = 0$  이라는 가설을 기각하는  $j$  들만 남을 때까지 계속한다. 남겨진  $j$  들의 집합을  $J$  라고 하고, 제거된  $j$  들의 집합을  $J^C$  라고 하자.

[3단계]: 모든  $\beta$  상태에서,  $J$  에 들어있는  $j$  에 대응되는  $\theta$  들에 대해서는 공통  $\theta$  라는 제약조

건을 주고,  $\widehat{\theta}_j, j \in J^C$  를 구한다. 유사 t-검정으로  $\theta_j = 0$  라는 가설을 채택하는  $\theta_j$  들을 제거한다. 이제  $K_1 = J^C - j: \theta_j = 0$  라고 한다.

[4단계]: 모든  $\beta$  상태에서,  $K_1$  에 들어있는  $j$  에 대응되는  $\theta$  들에 대해서는 공통  $\theta$  라는 제약 조건을 주고,  $\widehat{\theta}_j, j \in J$  를 구한다. 유사 t-검정으로  $\theta_j = 0$  라는 가설을 채택하는  $\theta_j$  들을 제거한다. 이제  $K_2 = J - : \theta_j = 0$ ,  $K = K_1 \cup K_2$  라고 한다.

[5단계]: 최우추정법으로  $\widehat{\beta}_j, j \in J_0$  와  $\widehat{\theta}_k, k \in K$  를 구한다. 유사 t-검정으로  $\theta_k = 0$  와  $\beta_j = 0$  이라는 가설을 채택하는  $k$  와  $j$  들을 제거한다. 이를 반복하여 더 이상 제거되지 않을 때까지 계속한다. 마지막 남은 모형을 선택한다.

### 알고리즘 B

[1단계]: 전통적 다중 회귀분석 모형  $y = \beta_0 + \beta_1 x_1 + \dots + \beta_d x_d + \gamma_1 x_1^2 + \dots + \gamma_d x_d^2$  에서 적절한 방법(예를 들어 단계적 선택법)으로 모형선택을 한다. 이때  $\beta_j$  와  $\gamma_j$  모두에서 선택되지 않은  $j$  들의 집합을  $L$  이라고 하자. 선택된  $\gamma_j$  들의  $j$  집합을  $J$  라고 하고 제거된  $\gamma_j$  들의  $j$  집합에서  $L$  을 뺀 집합을  $J^C$  라고 하자.

[2단계]:  $\theta_j = 0, j \in L$  으로 놓은 상태에서, 알고리즘 A의 [3단계]와 그 이후를 따른다.

알고리즘 B는 상대적으로 계산속도가 빠른, 전통적 회귀모형의 최적 선택방법을 이용하여 일부  $\theta$  를 제거하고 공통  $\theta$  로 묶을  $j$  들을 골랐다.

## 3. 일반화 비선형 최소제곱법에 의한 절대상수의 추정

원칙적으로 시뮬레이션 내에 존재하는 절대상수 (또는 매개변수,  $c^*$  라고 표기 함)는 비선형 최적화 기법으로 해결될 수 있다. 즉,  $Y(x)$ 를 시뮬레이션의 출력이라고 하고  $z(x)$  를 실제 실험자료라 하면 자료의 개수  $m$  에 대해서,

$$RSS = \sum_{i=1}^m (z(x_i) - Y(c^*, x_i))^2$$

을 최소로 하는  $c^*$  를 선택하므로써 가능하다. 그런데 문제는 우리가 사용할 시뮬레이터는 슈퍼 컴퓨터의 cpu 시간 만도 약 5 분이 걸리는 것으로서 위의 비선형 최적화를 위해서는 엄청난 시간이 소요됨으로, 이 방법은 현실적으로 매우 어려운 일이다. 따라서 우리는 위에서 선택된 Dace 모형을 사용하여 통계적 방법으로  $c^*$  를 추정할 것이다. Dace 모형에 기초하여 실험점이 아닌 부분에 대해 예측을 할 수 있는데 이를  $\widehat{Y}(x)$  라고 하자. 이  $\widehat{Y}(x)$  는 전산실험 자료로부터 쉽게 구해지고 그 컴퓨터 계산도 매우 빠르다. 따라서 우리는 이것을 시간이 많이 걸리는 시뮬레이션의

출력으로 대신하여 사용할 것이다. 즉, 우리의 새로운  $c^*$  추정의 기준은, 자료의 크기  $m$  대해서

$$RSS_p = \sum_{i=1}^m (z(x_i) - \widehat{Y}(c^*, x_i))^2$$

이 된다. 이 기준은 그 1차 미분치가 쉽게 구해지지 않으므로 수치적 미분에 의해서 1차 미분값을 근사시키는 경우의 Quasi-Newton 최적화 방법을 통하여 그 최소화가 수행되고, 이때 구해진 추정치를 “비선형 최소제곱 추정치”(nonlinear least squares estimator: NLSE)라고 부른다. 이렇게 절대상수의 통계적 추정을 통하여 시뮬레이션 프로그램 자체를 가장 현실적 상황과 맞게 최적화시키는 방법이다. 따라서 본 연구자는 이런 과정을 code tuning 이라고 불렀다[4][5][6].

이를 벡터와 행렬로 다시 표현하면 다음과 같아진다. 즉  $(z - \widehat{Y}(X))'(z - \widehat{Y}(X))$  를 최소로 하는  $c^*$ 를 구하는 것이다. 여기서  $\widehat{Y}$  는 위에서 선택한 최적 공간회귀모형을 이용하므로 결국 “비선형”이 된다.  $X$  는 실제 실험에 사용된 실험점으로  $c^*$ 를 포함한다. 이 방법은 이미 본 연구자에 의해서 연구된 바 있다[4][6]. O'Hagan과 Kennedy (2001)는 베이지안 접근법을 사용하여  $c^*$ 를 추정하였는데, 그들은 이런  $c^*$  추정법을 calibration 이라고 불렀다[15]. 절대상수  $c^*$ 의 추정을 위한 또 하나의 방법으로써  $(z - \widehat{Y}(X))'K^{-1}(z - \widehat{Y}(X))$  를 최소로 하는  $c^*$ 를 구하는 것이다. 이를 최소로 하는  $c^*$ 는 해석학적인 해를 구할 수 없으므로, 수치적 최적화 알고리즘을 이용하여 구한다. 이 방법은 “일반화된 비선형 최소제곱추정법( GNLSE)” 이다. 여기서  $K$  는 예측오차  $z - \widehat{Y}(X)$ 의 공분산 행렬로서  $m \times m$  행렬로서, 기존의 NLSE 에 “일반화” 라는 말을 붙게 한다. 이것은 몇 번의 계산을 거쳐(과정 생략) 결국  $K = \Sigma(X) + \sigma_e^2$  형태로 쓰지는데, 이때  $\sigma_e^2$  은 실제 실험의 관측자료에 있는 관측오차의 분산이고,

$$\Sigma(X) = E[(Y(X) - \widehat{Y}(X))(Y(X) - \widehat{Y}(X))|y] \quad (4.1)$$

이다. 여기서  $y$  는 전산실험에서의 반응값의 벡터이다. 이것을 실제 계산에 활용할 수 있도록 구체화된 공식으로 구해서 이용한다. 이를 위해  $\widehat{Y}(X) = (U, L)A^{-1}(y, 0)'$  을 이용한다. 그런데 여기서  $U = (v'_{x_1}, \dots, v'_{x_m})'$ ,  $L = (f'_{x_1}, \dots, f'_{x_m})'$  이다. 여기서  $U$  는 실제 자료의 실험계획점과 다른 점들과의 공분산 벡터로 된  $m \times n$  행렬이고,  $L$  은 실제 자료의 실험계획점에 대응하는 선형모형 벡터로 된  $m \times k$  행렬이다. 또

$$A = \begin{pmatrix} V & F \\ F' & O \end{pmatrix}$$

인  $(n+k) \times (n+k)$  행렬이다. 이제 식 (4.1)을 전개하면

$$E \widehat{Y}(X) \widehat{Y}(X) - 2E \widehat{Y}(X) Y'(X) + E Y(X) Y'(X)$$

인데, 여기서 세 번째 항이  $L\beta\beta'L' + \sigma_z^2 R(X, X)$  이 된다. 또한 첫 번째 항과 두 번째 항은 각각,  $Eyy' = \sigma_z^2 V + F\beta\beta'F'$ ,  $E[yY'(X)] = F\beta\beta'F' + \sigma_z^2 U$  이기 때문에, 블록행렬의 역행렬 공식을 이용하여, 몇 차례의 계산을 거쳐 결국  $\sigma_z^{-2}\Sigma(X) = R(X, X) - (U, L)A^{-1}(U', L')$  을 얻어서, 실제로 계산 가능한 형태가 되었다. 여기에 위에서 전산실험자료에 대해 구한 최적 회귀모형을 적용하며, 그때 구한  $\sigma_z^2$  와  $\sigma_{\theta}^2, \underline{\theta}$  의 최우추정치들 및 식과 K에 대입하여 사용한다.

#### 4. 실제 기계설계에 적용된 사례

실례로서 TV나 컴퓨터 모니터의 중요부품인 color picture tube 의 여러 부분의 최적설계에 DACE 가 어떻게 쓰였는가를 보이고자 한다[21]. 이 연구는 Philips 회사에서 개발한 시뮬레이션 모델에 기초하는데, 문제는 최적화하려고 하는 함수가 정확히 주어지지 않으며, 미분벡터에 대한 정보도 주어지지 않으며 컴퓨팅 시간도 매우 오래 걸린다. 이런 상황에서 몇 개의 설계 파라미터를 주어진 제약조건 하에서 최적화한다. 단계별 진행과정은 다음과 같다.

1단계: 문제 설정

2단계: 전산실험을 위한 좋은 LhD를 생성하여 실험을 실시 (실제로 제약조건 하에서 miximin 거리 LhD[16]를 사용함)

3단계: DACE 모델 설정 (이 DACE 모델은 원 시뮬레이션의 근사모델이며 계산시간이 매우 빠르고 다루기 쉬워서 모형의 속성과 및 통합이나 최적화 문제 해결 등이 수월하다.)

4단계: 위의 DACE 모델을 이용하여 최적화 실행 (실제로 pattern search 최적화 알고리즘을 사용)

5단계: 최적 파라미터들을 원래의 시뮬레이션 코드에 대입하여 추가적으로 몇 번 더 실행 후, 사전에 설정한 기준에 적합한지 확인. 만약 적합하면 정지하고, 미흡하면 2단계에서 추가적인 LhD를 생성하여 실험 실시 후 3, 4단계 진행.

이러한 접근법의 장점은 1) 개발시간을 단축시킬 수 있고, 2) 시뮬레이션 전문가가 아니라도 일단 만들어진 DACE 모델을 쉽게 이용할 수 있고, 3) 민감도 분석이나 파라미터의 영향력 등에 관한 연구를 더 쉽게 할 수 있고, 4) engineer가 designer에게 덜 의존적이게 되며, 그래서 결국 5) 향상된 설계가 이루어진다[21]. 실제로 color picture tube 의 최적설계에 있어 기존방법에 비하여 10-50% 정도의 설계향상이 있었고, 일부 프로젝트에서 50% 정도 개발시간이 단축됐다.

#### 5. 토의 및 향후과제

DACE 모델링을 시뮬레이션 전체에 적용하기보다는 모형 내에서 컴퓨팅 시간이 오래 걸리는 일부분에 적용하여 시뮬레이션의 정확도를 유지하면서도 계산시간을 매우 단축시키는 방법을 택할

수도 있다. 향후연구 또는 현재 진행되는 통계학 연구로는 자료에 적절한 최적의 가우시안 회귀모형 선택, 다양한 전산실험계획법[18]과 최적실험계획 또는 최적 LhD 의 효과적인 구축 알고리즘의 개발, 시뮬레이터와 실제자료가 잘 맞도록 하는 code tuning 또는 calibration 기법 연구, 시계열이거나 다반응의 출력에 대한 분석, DACE 기법을 활용한 신뢰성 연구, 이산형 입력변수 등 다양한 상황에서의 DACE 모형수립, 그리고 DACE 모형과 신경망 모형, 다차원 스프라인과의 비교연구 등을 들 수 있다.

### 참고문헌

- [1] 심정욱, 박정수, 배종성 (1994). Design and Analysis of Computer Experiments with Applications to Quality Improvement, *응용통계연구*, 7, 83-102.
- [2] 박정수, 황현식 (2000). 라틴-하이퍼큐브 실험계획 간의 거리 계산과 비교, *응용통계연구*, 13 (2) 477-488.
- [3] Kim T. Y., Park J. S., Cox D. D. (2002). Fast algorithm for cross-validation of the best linear unbiased predictor, *J COMPUT GRAPH STAT*, 11 (4): 823-835 DEC.
- [4] Cox D. D., Park J. S., Singer C. E. (2001). A statistical method for tuning a computer code to a data base, *COMPUT STAT DATA AN*, 37 (1): 77-92 JUL 28.
- [5] Cox D. D., Park J. S., Singer C. E. (1991). Sacks J, Tuning Complex computer Codes to Data, *Proceedings 23rd Interface Conference*, 23, 266-271.
- [6] Park J. S., Jeon J. (2002). Estimation of input parameters in complex simulation using a Gaussian process metamodel, *PROBABILIST ENG MECH*, 17 (3): 219-225.
- [7] Park J. S., Baek J. S. (2001). Efficient computation of maximum likelihood estimators in a spatial linear model with power exponential covariogram, *COMPUT GEOSCI-UK*, 27 (1): 1-7 FEB.
- [8] Park J. S. (1994). Optimal Latin-hypercube designs for computer experiments, *J STAT PLAN INFER*, 39: 95-111.
- [9] Lim Y. B., Sacks J., Studden W. J., et al. (2002). Design and analysis of computer experiments when the output is highly correlated over the input space, *CAN J STAT*, 30 (1): 109-126 MAR.
- [10] Ryu J. S., Kim M. S., Cha K.J., et al. (2002). Kriging interpolation methods in geostatistics and DACE model, *KSME INT J*, 16 (5): 619-632 MAY.
- [11] Sacks J., Welch W. J., Mitchell T., et al. (1989). Design and analysis of computer experiments, *STAT SCI*, 4 (4): 409-435.
- [12] Santer T. J., Williams B. J., Notz W. I. (2003). *The Design and Analysis of Computer Experiments*, Springer, NY.
- [13] Golub G. H., van Loan C. F. (1996). *Matrix Computations* (3rd ed.). The Johns Hopkins University Press, Baltimore.
- [14] Welch W. J., Buck R. J., Sacks J., et al. (1992). Screening, predicting, and computer experiments,

- Technometrics*, 34, 15-25.
- [15] Kennedy M. C., O'Hagan A. (2001). Bayesian calibration of computer models, *J ROY STAT SOC*, B 63: 425-450, Part 3.
- [16] Johnson, M., Moore, L., and Ylvisaker, D. (1990). Minimax and Maximin Distance Designs, *Jour Stat Plan Infer*, 26, 131-148.
- [17] Lee, J. H. (1999). Asymptotic comparison of Latin hypercube sampling and its stratified version, *Jour Korean Stat Soc*, 28, 135-150.
- [18] Fang, K. T. (2002). Experimental designs for computer experiments and for industrial experiments with model unknown, *Jour Korean Stat Soc*, 30, 277-300.
- [19] Simpson T., et. al. (2001). Kriging models for global approximation in simulation-based multidisciplinary design optimization, *AIAA Journal*, 39, 2233-2241.
- [20] Simpson T. et. al. (2001). Metamodels for computer-based engineering design: survey and recommendations, *Engineering with Computers*, 17, 129-150.
- [21] Hertog, D., Stehouwer, P. (2002). Optimizing color picture tube by high-cost nonlinear programming, *European Journal of Operational Research*, 140, 197-211.