An Efficient Sampling Technique for Off-line Quality Control

Jayant R. KALAGNANAM
IBM
T. J. Watson Research Center
Yorktown Heights, NY 10598
(jayant@watson.ibm.com)

Urmila M. DIWEKAR
Environmental Institute
Carnegie Mellon University
Pittsburgh, PA 15213
(urmila@cmu.edu)

The basic setting of this article is that of parameter-design studies using data from computer models. A general approach to parameter design is introduced by coupling an optimizer directly with the computer simulation model using stochastic descriptions of the noise factors. The computational burden of these approaches can be extreme, however, and depends on the sample size used for characterizing the parametric uncertainties. In this article, we present a new sampling technique that generates and inverts the Hammersley points (a low-discrepancy design for placing n points uniformly in a k-dimensional cube) to provide a representative sample for multivariate probability distributions. We compare the performance of this to a sample obtained from a Latin hypercube design by propagating it through a set of nonlinear functions. The number of samples required to converge to the mean and variance is used as a measure of performance. The sampling technique based on the Hammersley points requires far fewer samples to converge to the variance of the derived distributions. An application to off-line quality control of a continuous-stirred tank reactor illustrates that the Hammersley points require up to 40 times fewer samples to converge to the variance of the derived distribution.

KEY WORDS: Hammersley points; Sampling techniques; Stochastic optimization.

Parameter design is an off-line quality-control method for designing products and manufacturing processes that are robust in the face of uncontrollable variations (Taguchi and Wu 1980; Kackar 1985; Nair 1992). The goal of parameter design is to identify design settings that make the product's performance less sensitive to the effects of manufacturing and environmental variations, and deterioration. The variables that affect a product's performance are classified into *design parameters* whose nominal settings can be specified and *noise parameters* that represent uncontrollable variations both during a product's lifetime and across different units. A performance measure that is a function of the design parameters is chosen (different applications can lead to the choice of different measures) so that maximizing this measure minimizes the expected loss. To find these design settings, it is necessary to have available a characterization of how the noisy input variables affect the process response.

Two different approaches have been used to relate the noisy input parameters to the process output/s: (1) Physical experiments are conducted by varying the input parameters over the noise space to generate a response surface, or (2) a modeling approach is taken in which computational models are developed (based on physical principles) that are then used to study the impact of the noisy inputs on the process outputs. The basic setting of this article corresponds to the second case in which we conduct parameter-design studies using data from computer models. The evaluation of the important characteristics of process outputs requires a scheme to estimate the output distributions from the input distributions using data generated from computer experiments. Monte Carlo-type methods are used for propagating the effects of input variability through a computer model to generate and study the output variability. Let $\mathbf{u}$ denote the vector of noisy inputs; then a sample of n input vectors $\mathbf{u}_i, i = 1, \ldots, n$, which is representative of the uncertainty distribution, is generated and the outputs, $y_i = H(\mathbf{u}_i)$, are evaluated at each of these samples. H represents the computer code that corresponds to the process model. The samples, y_i, thus generated from the computer model are subsequently used to estimate the important characteristics of the process output. The scheme used to generate this sample from the computer model lies at the heart of the parameter design problem, and this issue forms the focus of this article.

A general computational approach to the parameter-design problem is obtained by coupling an optimizer directly with the computer simulation model using stochastic descriptions of the noise factors (Boudriga 1990; Diwekar and Rubin 1994) and formulating it as a stochastic optimization problem. Such an approach is most useful when the response of the model is not very smooth and it is hard to construct a response surface. It is also computationally more expensive, however. The stochastic optimization problem involves the evaluation of an aggregate measure (used as a performance statistic) derived from a multivariate probability distribution, which is estimated numerically using a representative sample from the multivariate space (as outlined in the previous paragraph) and has to be repeated at each optimization iteration. Therefore, an efficient sampling scheme that reduces the number of samples required for each iteration can significantly improve the computational efficacy of the stochastic optimization procedure.

In this article, we present a new and efficient sampling technique using the Hammersley points for uniformly sampling a k-dimensional unit hypercube. The Hammersley sequence is one of a class of number-theoretic approaches for constructing uniform sequences that are typically referred to as *low-discrepancy sequences.* This new sampling technique requires far fewer samples as compared to other conventional techniques (such as Latin hypercube sampling) to approximate the mean and variance of distributions derived by propagating a representative sample (for the inputs) over nonlinear functions. For off-line quality control posed in terms of stochastic optimization, the use of this efficient sampling technique can significantly alleviate the computational burden. We illustrate this by applying the technique for parameter design of a continuous-stirred tank reactor (CSTR) and report computational savings of up to a factor of 40.

The article is organized as follows: Section 1 introduces the off-line quality control of a CSTR to motivate the stochastic-optimization approach and the importance of an efficient sampling technique. Section 2 provides a discussion of the conventional sampling techniques used in the literature and introduces the new sampling technique based on Hammersley points. Section 3 presents the results of a large set of numerical experiments conducted to compare this new sampling technique to the conventional ones discussed in Section 2. Off-line quality control of a CSTR is revisited in Section 3 to complete the analysis initiated in Section 1. Section 4 provides the conclusions of this research.

1. OFF-LINE QUALITY CONTROL OF A CONTINUOUS-STIRRED TANK REACTOR

In this section we introduce the example of a CSTR to motivate the use of low-discrepancy sequences to perform stochastic optimization for parameter design. A brief description of the off-line quality-control problem in the context of the CSTR is presented. This problem is subsequently formulated in terms of stochastic optimization, and the underlying numerical integrations over the noise spaces are outlined. The sample size required to characterize the mean and the variance of output from the CSTR process is used to compare the efficacy of sampling techniques. The computational advantages gained by using Hammersley points (as compared to Latin hypercube sampling) are provided to motivate the use of low-discrepancy sequences for integrating over noise spaces.

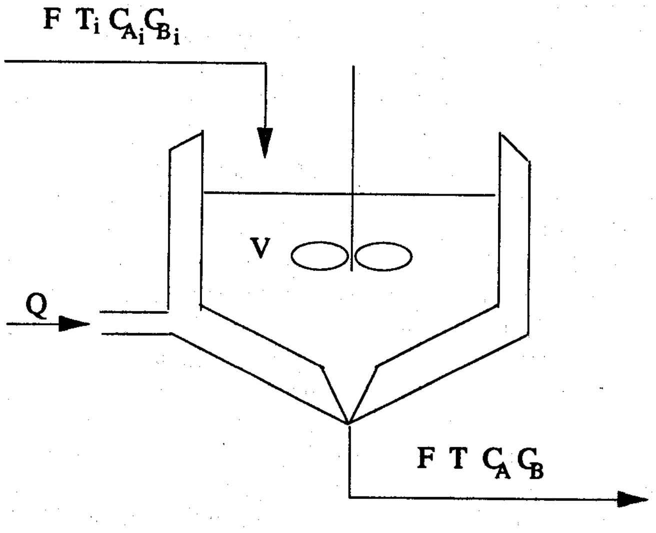

Figure 1. The Nonisothermal CSTR.

The system investigated in this study consists of a first-order sequential reaction, $A \rightarrow B \rightarrow C$, taking place in a nonisothermal CSTR. This is a common example used in the chemical-engineering design literature (Boudriga 1990; Diwekar and Rubin 1994) and is used here for illustrative purposes. Therefore, we have named the components A, B, and C and chosen inputs and parameters values for illustrative purposes. The process and the associated variables are illustrated in Figure 1. We are interested in designing this process such that the rate of production of species $B(r_B V)$ is 60 moles/minute (mol/min). As is apparent from the reaction pathway, however, species B degrades to species C if the conditions in the CSTR such as the temperature (T) and heat removal (Q) are conducive. The objective of parameter design is to produce species B at target levels with minimal fluctuations around the target in spite of continuous variation in the inputs. The inlet concentrations of A (C_{Ai}) and B (C_{Bi}), the inlet temperature (T_i), the heat added from the CSTR (Q), the bulk volume of the mixture in the CSTR (V), and the volumetric flow rate (F) are prone to continuous variations. (We have assumed here that the flow into the CSTR and the flow out of the CSTR are equal. The variations in the flow out of the CSTR are modeled via the variations in the bulk volume of mixture in the CSTR.) The objective of parameter design is to choose parameter settings for the design variables such that the variation in the production rate of $r_B V$ around the set point is kept to a minimum. The system parameters are also summarized in Table 1.

The five design equations that govern the production of species B (and the steady-state values of other variables) in the CSTR follow. The average residence time (τ) of each species in the reactor is given $\tau = V/F$:

$$Q = F\rho C_p(T - T_i) + V(r_A H_{RA} + r_B H_{RB}), \tag{1}$$

$$C_A = \frac{C_{A_i}}{1 + k_A^0 e^{-E_A/RT}\tau}, \tag{2}$$

$$C_B = \frac{C_{B_i} + k_A^0 e^{-E_A/RT}\tau C_A}{1 + k_B^0 e^{-E_B/RT}\tau}, \tag{3}$$

$$-r_A = k_A^0 e^{-E_A/RT} C_A, \tag{4}$$

and

$$-r_B = k_B^0 e^{-E_B/RT} C_B - k_A^0 e^{-E_A/RT} C_A. \tag{5}$$

C_A and C_B are the bulk concentrations of A and B, T is the bulk temperature of the material in the CSTR, and the rates of consumption of A and B are given by $-r_A$ and $-r_B$. These five variables are the state variables of the CSTR

Table 1. Parameters and Their Values Used in the Study

Parameter	*Values*	*Units*	*Description*
k_A^0	8.4×10^5	min^{-1}	Physical constant
k_B^0	7.6×10^4	min^{-1}	Physical constant
H_{RA}	-2.12×10^4	J/mol	Physical constant
H_{RB}	-6.36×10^4	J/mol	Physical constant
E_A	3.64×10^4	J/mol	Physical constant
E_B	3.46×10^4	J/mol	Physical constant
C_p	3.2×10^3	J/kg degree K	Physical constant
R	8.314	J/mol degree K	Physical constant
ρ	1,180.0	kg/m^3	Physical constant
C_{Ai}	3,118	mol/m^3	Input variable
C_{Bi}	342	mol/m^3	Input variable
T_i	314	°K	Input variable
Q	1.71×10^6	J/min	Input variable
V	.0391	m^3	Input variable
F	.0781	m^3/min	Input variable (calculated initially assuming $r_B V$ = 60)
C_A	2,275.7	mol/m^3	Output variable
C_B	1,110.2	mol/m^3	Output variable
r_A	−1,682.5	mol/m^3/min	Output variable
r_B	1,534.5	mol/m^3/min	Output variable
T	300	°K	Output variable

and can be estimated for a given set of values for the input variables ($C_{Ai}, C_{Bi}, T_i, Q, F,$ and V) and the following physical constants: k_A^0, k_B^0 and E_A, E_B, the pre-exponential Arrhenius constants and activation energies, respectively; H_{RA} and H_{RB}, the molar heats of the reactions that are assumed to be independent of temperature; and ρ and C_P, the density and specific heats of the system that are assumed to be the same for all processing streams. Once input variables are given, the state variables C_A, C_B, r_A, r_B, and T can be solved iteratively. An initial value for the bulk CSTR temperature T is chosen and Equations (2)–(5) are used to evaluate C_A, C_B, r_A, and r_B, respectively. Substituting these values in Equation (1), the bulk temperature T is solved iteratively using a secant method (Press, Flannery, Teukolsky, and Vetterling 1986). If the production rate is fixed as in this case, then one can free F as the input variable and calculate it iteratively to match the production rate $r_B V$.

The design objective is to produce 60 mol/min of component B; that is, $R_B = r_B V = 60$. The initial nominal set points for the input variables corresponding to this point are provided in Table 1. The continuous variations in the input variables ($C_{Ai}, C_{Bi}, T_i, Q,$ and F), however, result in continuous variations in the production rate, R_B. The variations in the inputs are described using two-parameter (mean and variance) normal distributions. These variations are assumed, for the purpose of illustration, to be kept at an error level for each input of $E_i = \sigma_i/\mu_i \times 100 = 10\%$. The variation in R_B (around the initial nominal point) due to the variations in the inputs is characterized as a probability distribution. The distribution is estimated by sampling the normal distributions for each of the six normal input uncertainty distributions (with an error level of 10%) and solving Equations (1)–(5) for R_B at each of these samples. The parameter design problem for the CSTR can now be described as a stochastic optimization problem in which the objective is to find parameter settings for the input variables that minimize the variance of the output distribution for R_B. This can be characterized in mathematical terms as a nonlinear stochastic optimization problem as follows:

$$\underset{\theta}{\text{optimize}} \quad J = P_1(R_B(\theta, x, u)) \tag{6}$$

subject to

$$P_2(h(\theta, x, u)) = 0 \tag{7}$$

and

$$P_3(g(\theta, x, u) \leq 0) \geq \alpha, \tag{8}$$

where u is the vector of uncertain input variables $(C_{Ai}, C_{Bi}, T_i, Q, F, V), \theta$ is the set of control variables $(C_{Ai}, C_{Bi}, T_i, Q, V)$, and x is the set of parameters including the physical constants. The P_i represent probabilistic functionals, which, for the case of variance minimization of R_B, can be represented by

$$\sigma_{R_B}^2 = \int_0^1 (R_B - \overline{R_B})^2 \, dF, \tag{9}$$

where F is the cumulative probability distribution of R_B. The mean ($\bar{R}_B$) and variance ($\bar{\sigma}_{R_B}^2$) are estimated as follows:

$$\overline{\sigma_{R_B}}^2 = \frac{\sum_1^{N_{\text{samp}}} (R_{Bi} - \overline{R_B})^2}{N_{\text{samp}}} \tag{10}$$

and

$$\overline{R_B} = \int_0^1 R_B(\theta, x, u) \, dF. \tag{11}$$

$$= \frac{\sum_1^{N_{\text{samp}}} R_{Bi}}{N_{\text{samp}}}, \tag{12}$$

where R_{Bi} are the outputs of the N_{samp} samples used for error propagation. The number of samples required for propagating uncertainty depends on the accuracy required for estimating the variance of the output distribution. The nominal values of the variables are calculated using the preceding objectives by solving the five equations [Eqs. (1)–(5)]. The variance in the designed output variable is estimated by propagating a sample set of points from the joint probability distribution of the inputs with errors. For the example of the CSTR, we do not have any probabilistic constraints; therefore Equations (7) and (8) can be omitted.

The nonlinear stochastic optimization problem is solved using successive quadratic programming (SQP). In SQP, at each iteration the problem is approximated as a quadratic program in which the objective function is quadratic and the constraints are linear. Similar to linear programming, the special features of a quadratic objective function are exploited to solve the problem more efficiently. The quadratic programming subproblem is solved for each step to obtain

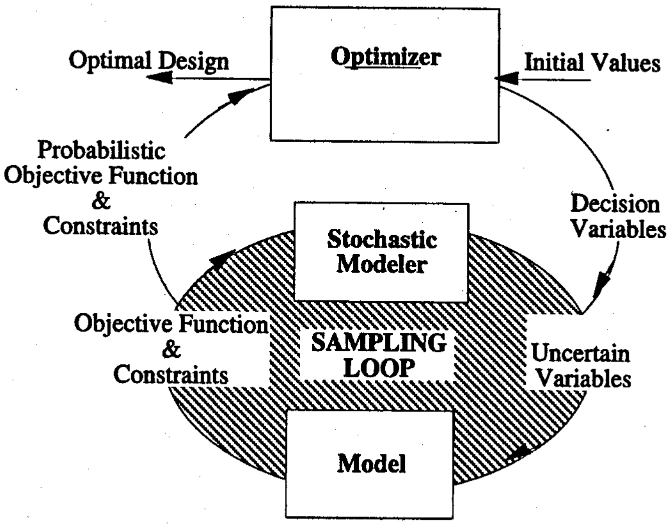

Figure 2. Pictorial Representation of the Stochastic Optimization Framework.

the next trial point. This cycle is repeated until the optimum is reached. The stochastic optimization procedure is presented diagrammatically in Figure 2.

The stochastic modeler assigns the probabilistic distribution to the model input parameters u, then uses a sampling technique to generate a specified number of samples (N_{samp}) and passes the sampled value of each parameter to the model. After each model run, the output variables of interest (in this case the objective function and constraints) are collected. The simulation is then repeated for a new set of samples selected from the probability distributions. After all the samples have gone through the cycle, the stochastic modeler analyzes the output and finds the expected value of the objective function and constraints, which is then passed to the optimizer.

One can easily envision the computational intensity of the stochastic optimization problem (Fig. 2) presented earlier. As in deterministic optimization, at each optimization iteration one needs to run the stochastic model with many samples to calculate the probabilistic functionals. The different sampling techniques currently available in the literature include the commonly used Monte Carlo technique and stratified sampling techniques such as Latin hypercube sampling (LHS). In our analysis of the CSTR, we found that the Monte Carlo method requires well over 10,000 samples to estimate the variance of R_B to within 1% of the true value. The Latin hypercube technique reduces the sample requirement to about 6,100 samples. In contrast, the new technique introduced in this article (based on quasi-random sequences) requires only about 150 samples to achieve the same level of accuracy as presented in Section 3. Such computational savings motivate the use of low-discrepancy sequences for multivariate integration over noise spaces. These techniques and the new one (introduced in this article) using quasi-random sequences are discussed in the following section.

2. THE NEW SAMPLING TECHNIQUE

Because most stochastic-optimization problems involve integrals of some probabilistic functional [Eqs. (6) and (9)], we need to design a sampling technique that provides a sample that is representative of the probability distribution. A common approach is to generate a sequence of points of size n on a k-dimensional unit hypercube, assuming a uniform distribution $U(0,1)$. The specific values for each input variable are selected by inverting the n samples over the cumulative distribution function. The convergence rate of the sampling technique depends in a critical way on how the sequence on the unit hypercube is chosen. In this section we discuss some of the commonly used sampling designs (on the unit hypercube) and introduce a quasi-Monte Carlo design based on the Hammersley sequence. Later we show how a sampling technique that uses the Hammersley sequence provides a faster convergence rate than other commonly used techniques.

Perhaps one of the best known methods for sampling a probability distribution is the Monte Carlo sampling (MCS) technique, which is based on the use of a pseudorandom-number generator to approximate a uniform distribution, $U(0,1)$ with n samples, on a k-dimensional unit hypercube. The specific values for each input variable are selected by inverting the n samples over the cumulative distribution function. On average, the error ε of approximation is of the order $O(N^{-1/2})$. The remarkable feature is that the bound is not dependent on the dimension k. One of the main disadvantages of the Monte Carlo method, however, is that the bound is probabilistic, and there is no methodical way for constructing the sample points to achieve the probabilistic bound (Papageoriou and Wasilkowski 1990; Niederreiter 1992). It is also important to note that the error of approximating an integrand by a finite sample depends on the equidistribution properties of the sample used for $U(0,1)$ rather than on its randomness. Once it is apparent that the uniformity properties are central to the design of sampling techniques, constrained or stratified sampling becomes appealing (Morgan and Henrion 1990).

LHS (Iman and Shortencarier 1984) is one form of stratified sampling that can reduce the variance in the Monte Carlo estimate of the integrand. The range of each input u_i is divided into nonoverlapping intervals of equal probability. One value from each interval is selected at random with respect to the probability density in the interval. The n values thus obtained for u_1 are paired in a random manner with the n values of u_2, and these n pairs are combined with n values of u_3 and so on to form n k-tuplets. The random pairing is based on a pseudorandom-number generator. The main shortcoming with this stratification scheme is that it is one-dimensional and does not provide good uniformity properties on a k-dimensional unit hypercube (for k input variables). This approach still only provides probabilistic bounds; however, it reduces the error of the estimate as compared to the Monte Carlo approach.

The quasi-Monte Carlo methods seek to construct a sequence of points that perform significantly better than Monte Carlo, which has an average case complexity of the order $1/\varepsilon^2$. For a suitably chosen set of samples, the quasi-Monte Carlo method provides a deterministic error bound of the order $N^{-1}(\log N)^{k-1}$ without any strong assumptions about the integrand. Some well-known constructions for quasi-Monte Carlo sequences are the ones due to Halton, Hammersley, Sobol, Faure, Korobov, and Niederreiter (Neiderreiter 1992). Fang, Wang, and Bentler (1994) examined the applications of some of these methods for statistical inference and regression analysis.

The basic idea in this article is to replace a Monte Carlo integration by a quasi-Monte Carlo scheme in the stochastic-optimization problem. In this section we describe a new sampling technique based on the use of the Hammersley points. We call this new technique the Hammersley sequence sampling (HSS) technique. It uses the Hammersley points to uniformly sample a unit hypercube and inverts these points over the joint cumulative probability distribution to provide a sample set for the variables of interest. In the following two subsections we describe an algorithm for generating the Hammersley points, the implementation of inversion, and the imposition of a correlation structure on the sample.

2.1 The Hammersley Points

The choice of an appropriate quasi-Monte Carlo sequence is based on the concept of discrepancy. The deterministic upper and lower error bounds of any sequence for integration are expressed in terms of the discrepancy measure. Discrepancy is a quantitative measure for the deviation of the sequence from the uniform distribution. Therefore, it is typically desirable to choose a low-discrepancy sequence. Some examples of low-discrepancy sequences are the Halton (1960) and Hammersley (1960) sequences. The constant terms on the error bounds for these sequences, however, are a strong function of the dimension k of the unit hypercube, and other sequences such as the Sobol sequences (Niederreiter 1978) and the Faure sequences (Fox 1986) have been developed to alleviate this problem. The other problem often encountered with the preceding sequences is that the error bounds are not adequately sensitive to the form of the integrand. Several designs using "good lattice" points were introduced by Korobov (Niederreiter 1978) and Niederreiter (1988) in the literature to address these issues. Without embarking on a detailed discussion of these issues (see Niederreiter 1992), it is apparent that we are faced with the issue of which sequence one should use for the design of a quasi-Monte Carlo sampling technique. We have chosen to examine the Hammersley points in this article. Once the advantages of using low-discrepancy sequences (as compared to pseudo-random numbers) is established, the optimal choice of a low-discrepancy sequence can be examined. This issue would be the topic of another article, however, and is not addressed in this one.

In this paragraph we provide a definition of the Hammersley points and explicate a procedure for it's design. Any integer n can be written in radix-R notation (R is an integer) as follows:

$$\begin{aligned} n &\equiv n_m n_{m-1} \cdots n_2 n_1 n_0 \\ &= n_0 + n_1 R + n_2 R^2 + \cdots + n_m R^m, \end{aligned}$$

where $m = [\log_R n] = [\ln n / \ln R]$ (the square brackets denote the integral part). A unique fraction between 0 and 1 called the *inverse radix number* can be constructed by reversing the order of the digits of n around the decimal point as follows:

$$\begin{aligned} \phi_R(n) &= .n_0 n_1 n_2 \ldots n_m \\ &= n_0 R^{-1} + n_1 R^{-2} + \cdots + n_m R^{-m-1} \end{aligned}$$

The Hammersley points on a k-dimensional cube are given by the following sequence:

$$\vec{z}_k(n) = \left(\frac{n}{N}, \phi_{R_1}(n), \phi_{R_2}(n), \ldots, \phi_{R_{k-1}}(n)\right), \qquad n = 1, 2, \ldots, N,$$

where $R_1, R_2, \ldots, R_{k-1}$ are the first $k-1$ prime numbers. The Hammersley points are $\vec{x}_k(n) = 1 - \vec{z}_k(n)$.

2.2 Implementation of Correlation Structures

The implementation of correlation structures is based on the use of rank correlations (Iman and Conover 1982). The method is very similar to the one used for Latin hypercube samples with one difference: LHS uses a matrix of independent permutations of arbitrary scores for generating a correlation structure, whereas for HSS we use the Hammersley points for the same purpose. In this subsection we outline the method based on rank correlations used for generating a correlation structure in LHS and highlight the main difference in the implementation for the HSS technique.

Let X be a matrix of uncorrelated random vectors, and let C be the desired rank correlation matrix of X. Then, because C is positive definite, $C = P \times P'$ (Cholesky factorization), where P is a lower triangular matrix. Then, for some matrix R of arbitrary scores, the transformed matrix $R^* = R \times P'$ has the desired rank correlation matrix C. R is chosen such that the correlation matrix and the rank correlation matrix of R^* are the same. Now, to introduce the desired rank correlation in X, the random vectors are arranged in the same rank order as R^*. For LHS, the matrix R is constructed from van der Warden scores (Iman and Conover 1982), whereas for HSS the matrix is the set of Hammersley points.

The main impact of using rank correlations for HSS is that the uniform structure of the Hammersley points is somewhat distorted; however, its effect on the transformed sample is not easily characterized analytically. Whether the distortions are large enough to completely negate the advantages of the Hammersley points is an empirical question that is investigated in Section 3 by comparing the convergence properties of HSS with LHS for correlated samples.

3. RESULTS AND DISCUSSION

In this section we examine the uniformity properties of the new sampling scheme—that is, the HSS technique—characterize its impacts on the computational intensity of the stochastic-optimization problem outlined in Section 1, and illustrate it with the CSTR example from process design. The results of this analysis are presented in the following way. First the uniformity properties of the Hammersley sequences are compared to sampling schemes graphically. Next a series of numerical experiments is described that

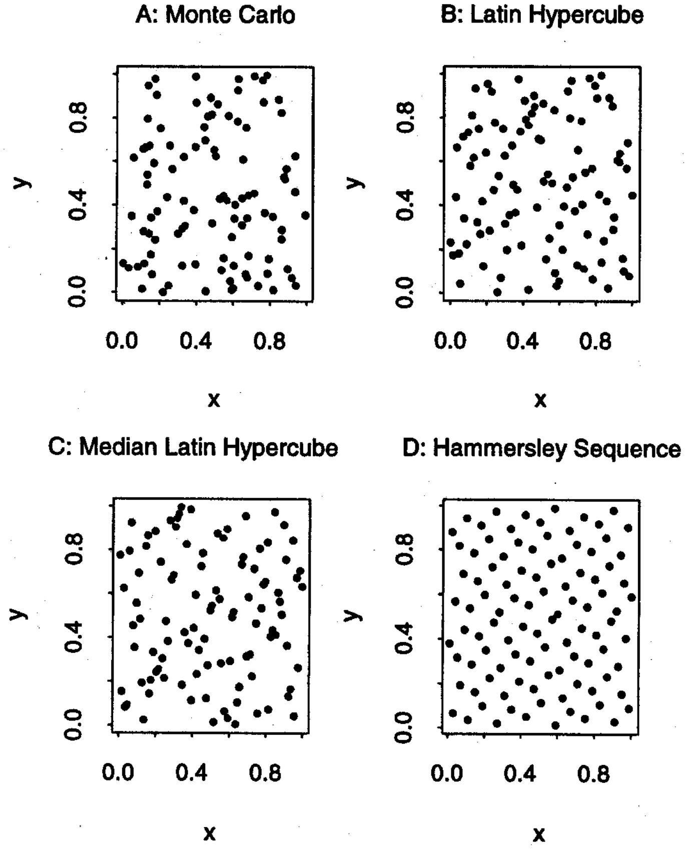

Figure 3. Sample Points (100) on a Unit Square Using (A) Linear Congruent Generator, (B) Random Latin Hypercube, (C) Median Latin Hypercube, and (D) the Hammersley Points.

examines the number of samples that are required to converge to the mean and variance for each sampling scheme. Finally, we apply the new sampling scheme to the design of a CSTR and illustrate the computational savings as compared to using the conventional techniques such as Monte Carlo or Latin hypercube.

3.1 Uniformity Properties of the Hammersley Points

In our discussion of different sampling techniques, we explicate the importance of the uniformity properties of a sampling technique when the sample is used for approximating a distribution by finite samples. Figure 3 graphs the samples generated by different techniques on a unit square. This provides a qualitative picture of the uniformity properties of the different techniques. It is clear from Figure 3 that the Hammersley points have better uniformity properties than other techniques. The main reason for this is that the Hammersley sequence is a low-discrepancy design for placing n points on a k-dimensional hypercube. In contrast, other stratified techniques such as the Latin hypercube are designed for uniformity along a single dimension and then randomly paired for placement on a k-dimensional cube. Therefore, the likelihood of such schemes providing good uniformity properties on high-dimensional cubes is extremely small. Figure 4 illustrates the effect of imposing a correlation structure on the sample sets. The approach used is described in Section 2.2, which uses rank correlations to preserve the stratified design along each dimension. Although this approach preserves the uniformity properties of the stratified schemes, the low-discrepancy design of the Hammersley sequence is perturbed by imposing the correlation structure. The effect of this on the uniformity properties is not apparent from Figures 3 and 4; however, we will examine this issue in detail in the following subsections.

3.2 Convergence Properties of Samplers

In this subsection, we provide a comparison of the performance of the HSS technique to that of LHS and MCS techniques. The comparison is performed by propagating samples derived from each of the techniques for a set of n-input variables (X_i), through various functions ($Y = f(X_1, X_2, \ldots, X_n)$) and measuring the number of samples required to settle down to within an error of $E\%$ (typically we use 1%) of the "true" mean and variance of the derived distribution for Y. We have adopted the following decision rule for determining the convergence of a sampling technique: (1) We estimate the "true" mean and variance for each test case by propagating a very large number of samples using Monte Carlo (50,000 samples), Latin hypercube, and Hammersley points (about 10,000 samples). When the techniques provide the same estimates for the mean and variance, we accept these values as "true" values (else we might have to increase the number of samples). (2) Once the "true" values have been established, the performance of different sampling techniques is compared by estimating the number of samples required to settle to within 1%

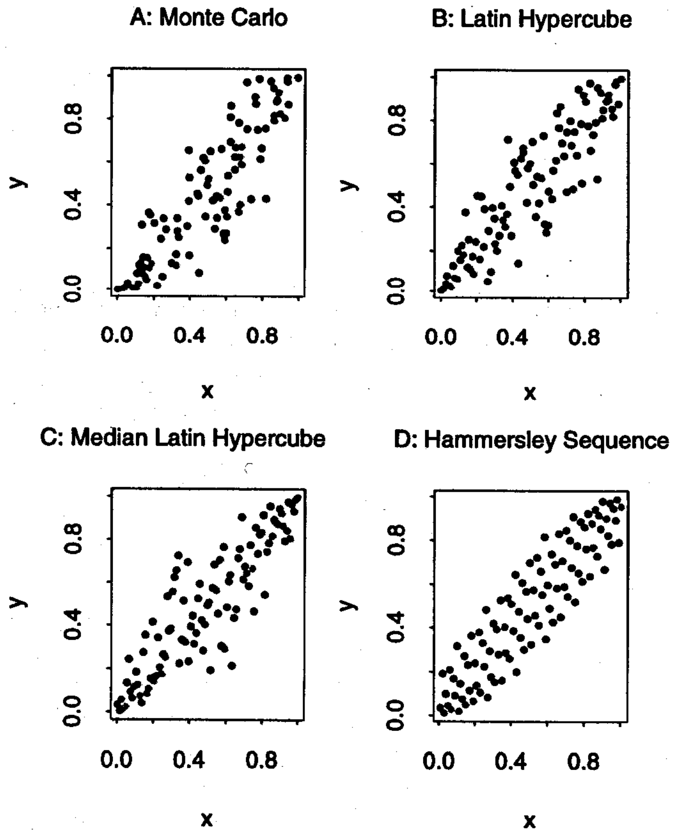

Figure 4. Sample Points (100) on a Unit Square With Correlation of .9 Using (A) Linear Congruent Generator, (B) Random Latin Hypercube, (C) Median Latin Hypercube, and (D) the Hammersley Points.

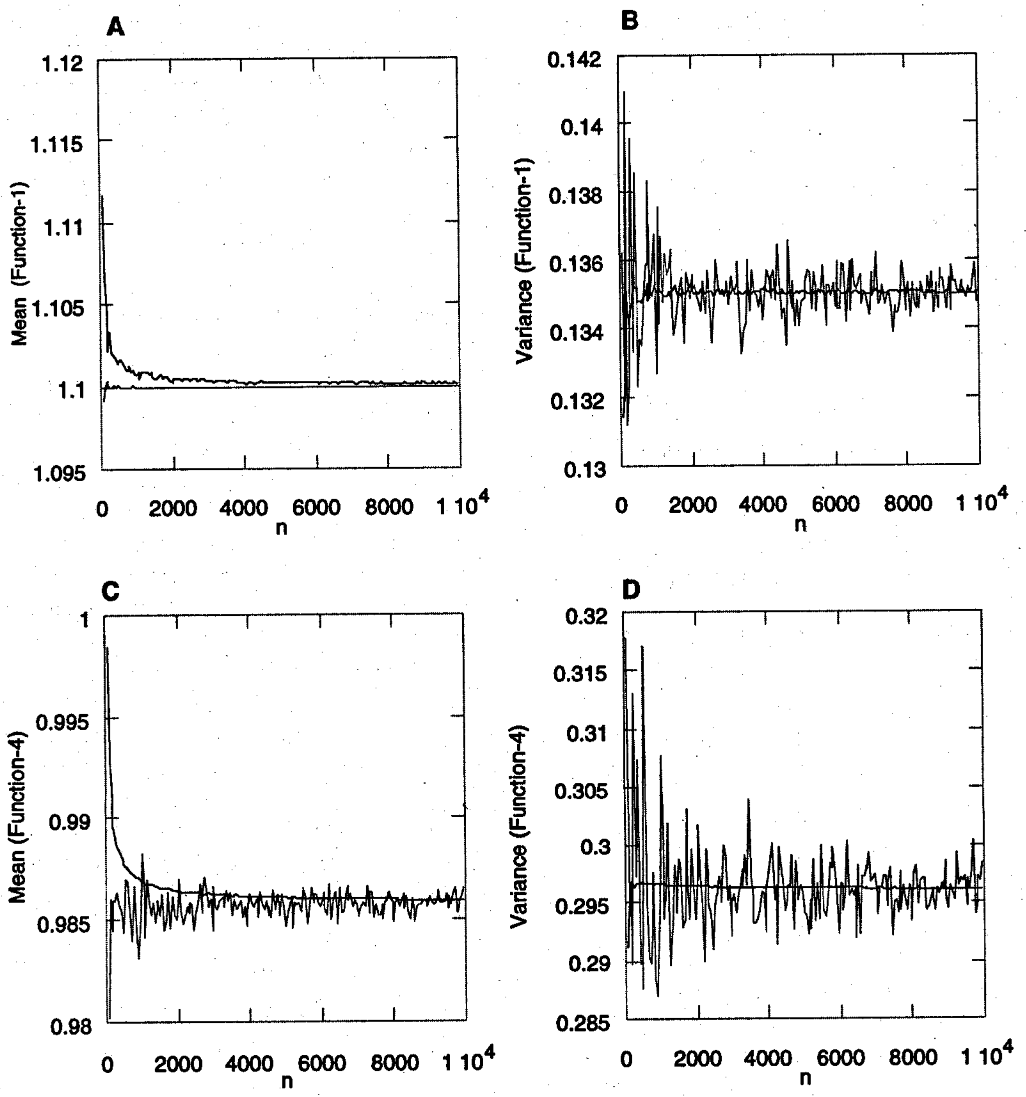

Figure 5. The Mean and Variance as a Function of Sample Size for LHS (dotted line) and HSS (solid line) for Two Input Variables Without Correlations: (A) Mean of Function 1, (B) Variance of Function 1, (C) Mean of Function 4, and (D) Variance of Function 4.

of the "true" values. This performance is also graphically presented by plotting the calculated value of mean and variance as a function of number of samples used in the calculation. Sampling schemes that add one point at a time (such as Monte Carlo and Hammersley sequences) typically have less fluctuations as compared to schemes that do not retain the original n points in generating the next one. As a result of this, one might be misled into thinking that "retaining schemes" have converged when in fact they are fluctuating slowly.

To address these issues, we have chosen some test functions that are analytically simple for which the "true" mean and variance can be estimated exactly. For these functions we will show in the following section that the calculated values for each sampling technique converge to the true values. This comparison also provides an estimate of the approximate number of samples required (order of magnitude) for convergence. Additionally, to guard against underestimating the number of samples required for convergence of "retaining schemes," we examined each of the test functions (and the CSTR example) using at least five times the number of samples required for the sequence to initially settle down. In other words, if the fluctuations in the variance estimate using Hammersley sequence settles down around N samples, we continued the simulation up to $\max(5 \times N, 10{,}000)$ to examine if the estimate slowly fluctuates out of the 1% band. This oversampling guards against underestimating the settling time of the "retaining schemes."

Extensive numerical comparisons of the Monte Carlo technique with the Latin hypercube (e.g., Diwekar and Rubin 1994) show that the effect of not retaining the original n samples in generating the next sample is actually quite negligible as compared to that of the uniformity properties of the sampling techniques. The Monte Carlo method retains the n samples while generating the next sample, whereas in LHS the $n+1$ samples are completely unconnected to n samples generated previously. Although one might expect this to exaggerate the fluctuations and convergence properties of the LHS scheme, numerical experiments indicate that the fluctuations in LHS are systematically smaller than MCS. This indicates that fluctuations in LHS are more sen-

Table 2. True Estimates

Function	*Mean*	*Variance*
Function 1	1.10	.135
Function 4	.98579	.296013

sitive to the uniformity properties of the sampling technique than to the retention of the original n points in generating the next sample. This empirical observation forms the basis of the decision rule used for estimating convergence.

Now we present results from a large matrix of numerical tests. The design of the test matrix included varying of the type of function, the number of input variables X_i, the type of input distribution, and the correlation structures between them. The details of the test matrix are as follows:

Sampling Techniques: A total of four sampling techniques have been compared: Monte Carlo, random Latin hypercube, median Latin hypercube (which is the same as Latin hypercube except that observations are taken at the median point within each of the n equiprobable intervals), and Hammersley.

Number of Variables: The number of input variables used was varied between 2 and 10.

Functions: Five different kinds of functions were used, as follows:

1. Function 1: Linear additive function: $Y = \sum_i X_i$
2. Function 2: Multiplicative function: $Y = \prod_i X_i$
3. Function 3: Quadratic function: $Y = \sum_i X_i^2$;
4. Function 4: Exponential function: $Y = \sum_i X_i \times \exp X_{i+1}$
5. Function 5: Logarithmic function: $Y = \sum_i X_i \times \log(X_{i+1})$

For functions 4 and 5, the variable X_{i+1} wraps around to X_1 if required.

Distributions: Three types of distributions have been used for the input variables X_i. Two of them, uniform and normal, are symmetric, and the third is a skewed distribution, lognormal.

Correlations: Three types of correlation structures have been used: The first is a zero correlation, and the other two sets use a correlation of .5 and .9 between the input variables.

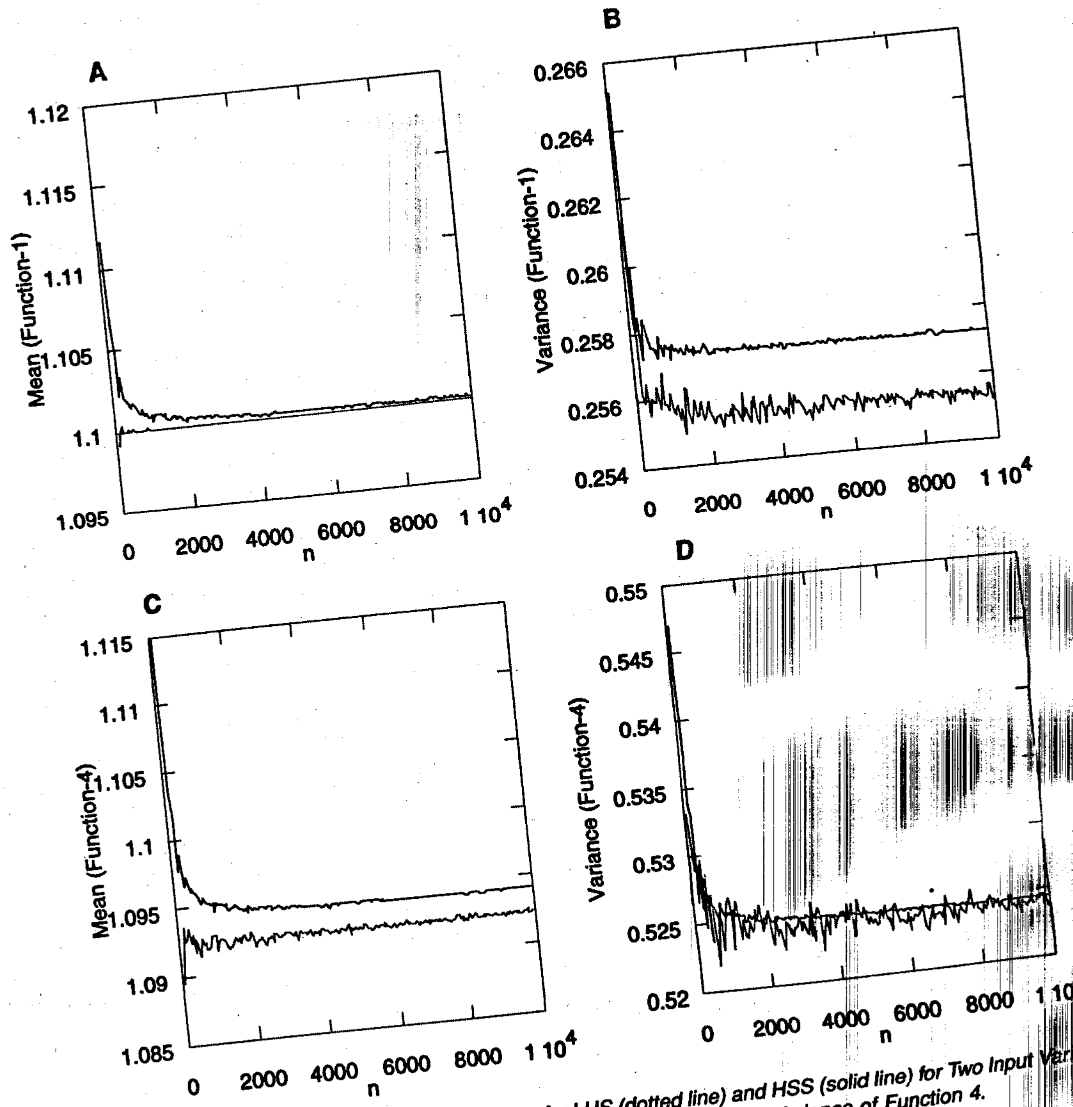

Figure 6. The Mean and Variance as a Function of Sample Size for LHS (dotted line) and HSS (solid line) for Two Input Variables With Correla of .9: (A) Mean of Function 1, (B) Variance of Function 1, (C) Mean of Function 4, and (D) Variance of Function 4.

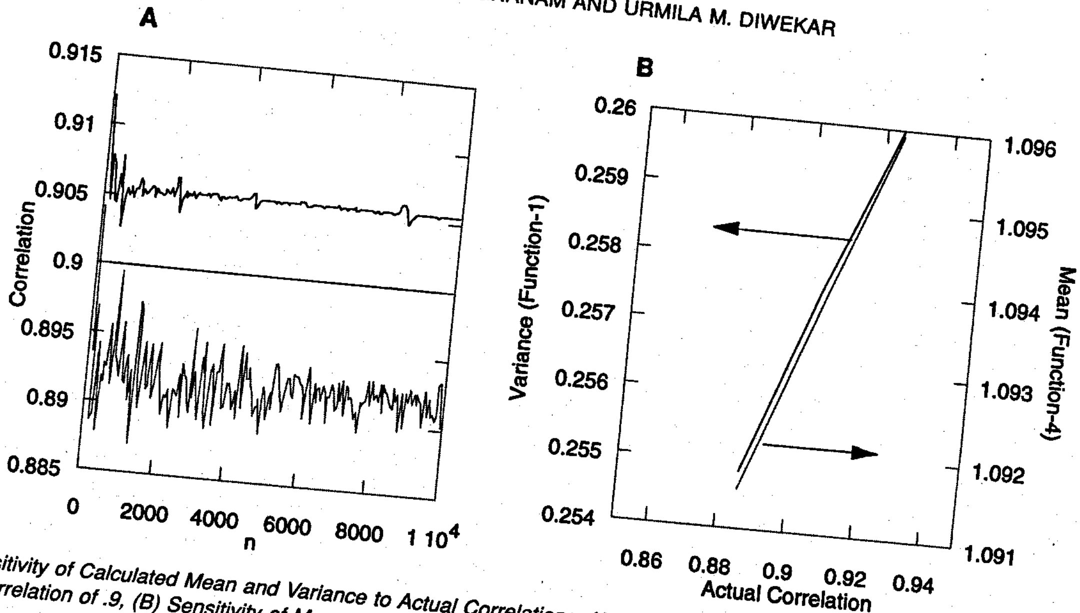

Figure 7. Sensitivity of Calculated Mean and Variance to Actual Correlations: (A) Actual Correlations for HSS (solid line) and LHS (dotted line) for a Specified Correlation of .9, (B) Sensitivity of Mean and Variance to Different Levels of Correlations in the Input Variables.

This matrix represents a total of 180 datasets (4 sampling techniques × 3 types of distributions × 3 correlation structures × 5 functions) for each set of input variables, X_i. As the number of input variables is varied from 2 to 10, it adds another factor of 9; that is, there are 1,620 datasets. In the interests of space and clarity, however, we will present only the results that highlight the main findings of this numerical experiment.

Initially we present two figures that illustrate the rate of convergence for both the LHS and HSS sampling technique. Figure 5 plots the mean and the variance for Functions 1 and 4 using two input variables that are uncorrelated. A uniform distribution $U(.1, 1)$ is used for both the inputs. The results are unequivocal—the HSS technique requires far fewer samples to converge to within 1% of the variance. Median Latin hypercube is, however, preferable for estimating the mean of linear functions (with symmetric distributions) because it provides exact estimates. This is illustrated by Figure 5A, which plots the mean for Function 1, which is linear. The HSS mean estimates for nonlinear functions usually converge faster, but the difference between the approaches is slight. Function 1 and Function 4 are also analytically simple enough that their "true" mean and variance can be estimated analytically and are reported in Table 2. The calculated estimates match the true estimates very accurately, which provides an acceptable level of calibration of the accuracy of the sampling techniques.

Figure 6 is similar except that a correlation of .9 has been introduced between the two input variables. Once again, the HSS technique requires far fewer samples to converge to within 1% of their respective converged values of mean and variance. The HSS sample and the LHS sample, however, sometimes converge to a slightly different mean (for Function 4) and variance (for Function 1). Figure 7 presents

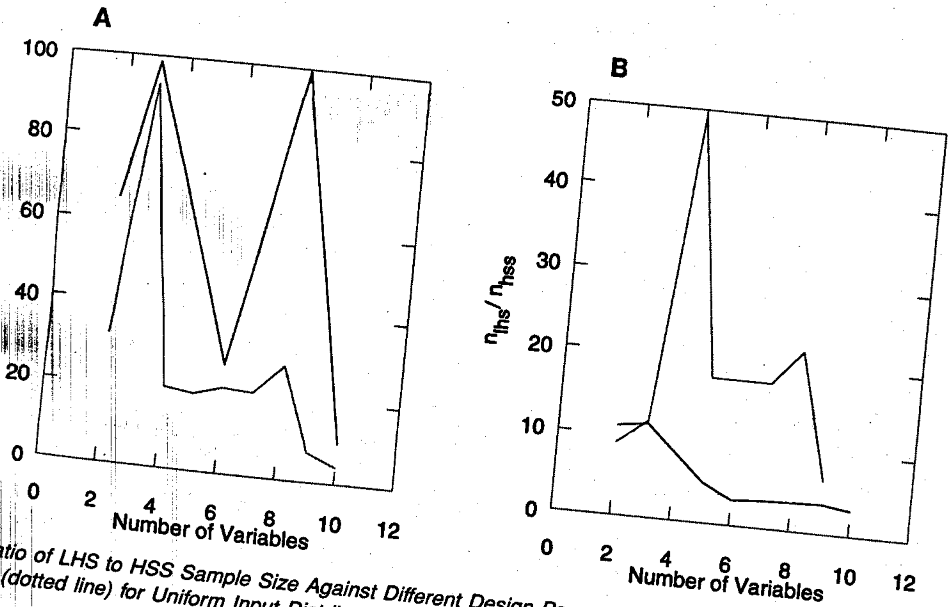

Figure 8. Plot of the Ratio of LHS to HSS Sample Size Against Different Design Parameters for 1% Convergence of Variance: (A) Function 1 (solid line) and Function 4 (dotted line) for Uniform Input Distributions, and (B) Function 1 (solid line) and Function 4 (dotted line) for Lognormal Input Distributions.

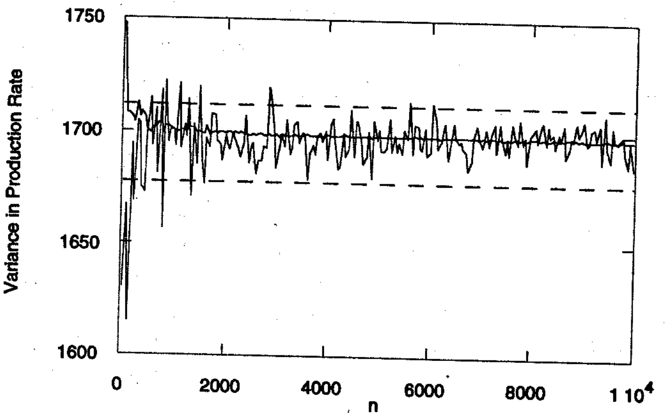

Figure 9. Variance of R_B Using LHS (dotted line) and HSS (solid line): 1% Error Band Is Shown by Broken Lines.

two graphs that examine these differences in some detail. Figure 7A shows the actual correlation for the LHS and HSS sample for a desired correlation of .9. It is observed that the actual correlation for HSS is .905 and for LHS is .895. We find that the Monte Carlo approach also introduces such a bias (not shown in figure). The appearance of a bias in the actual correlation has been consistently observed over different functions, seeds, and sample size for all sampling techniques. (One approach for adjusting these errors is based on trial-and-error methods in which the target correlation is perturbed to achieve desired correlations.) This indicates that numerical techniques used in Section 2.2 introduce a bias in the actual correlation for a desired level of correlation. The calculated mean and the variance estimates of the output variable for correlated input variables is extremely sensitive to small differences (less than 1%) in the actual correlations in the input samples. This is illustrated in Figure 7B, in which the calculated mean and variance are plotted as functions of actual correlation. Figure 7 illustrates how numerical errors introduced in actual correlations are propagated to the calculated mean and variance, and this in turn lies at the heart of the difference in the converged values for mean and variance in Figure 6.

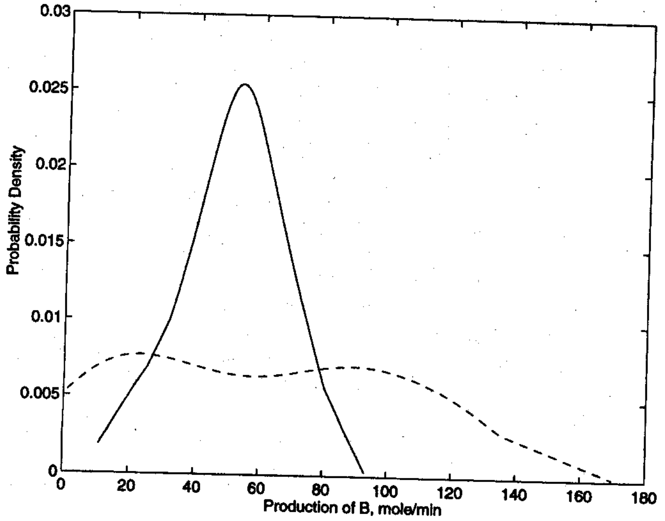

Figure 10. Variations in the Production Rate of Component B, Before (broken line) and After (solid line) Design for Quality.

Figure 8 presents a more comprehensive view of the comparisons conducted in the numerical experiment. This figure plots the ratio of the LHS to the HSS sample size as a function of the design parameters (outlined in the preceding matrix) of the numerical experiment. The convergence sample sizes for the LHS schemes were averaged over 10 different sample sets. The sample size used for this comparison is the number of samples required to converge to within 1% of the actual value of variance. Each subgraph plots the ratio of the sample size against the number of input variables for the two functions (Function 1 and Function 4) and for two types input distributions uniform (symmetric) and lognormal (asymmetric). Once again the results are encouraging—the HSS sampling technique has a much faster convergence rate, anywhere from a factor of 3 to 100 and larger! The results presented here are qualitatively representative of the general trends observed for all the datasets that were analyzed.

3.3 CSTR Revisited

In Section 1 we formulated the parameter-design problem for the CSTR. In this section we present the results of nonlinear stochastic optimization applied to this problem and subsequently compare the performance of the Hammersley samples and the Latin hypercube samples for this problem.

The design objective was to produce 60 mol/min of component B; that is, $R_B = 60$. The variance in the designed output variable is estimated by propagating a sample set of points from the joint probability distribution of the inputs with errors. Because the objective from a quality-control perspective is to minimize this variance, we characterize the number of samples required to estimate the variance to within 1% of its value using both Latin hypercube and shifted Hammersley points. The Hammersley points require about 150 points to converge as compared to 6,100 points required by the Latin hypercube sample. Figure 9 plots the variance as a function of number of samples. Figure 10 shows the variation in the production rate (r_B) before and after the design for quality control, where the variance is reduced from 1,638 to 232 by merely changing the nominal values of the parameters as shown in Table 3.

It is interesting to note some of the properties of the final setting found by the parameter-design methodology. First of all, the volume of the bulk mixture in the reactor is increased by more than 20%, thereby adding further damping to the system—this increased mass capacitance provides an increased buffer between the input variations in concentrations and the output production rate. Another interesting suggested change is that the temperature (T_i) of the inlet stream is increased, which requires that we add this heat to keep the CSTR bulk temperature (T) at about 309 K. This results in an increased operating temperature of the CSTR, which is locally less sensitive to inlet temperature and heat added. Both of these changes would have some economic impacts on the cost of running the CSTR. In a more practical setting it might become imperative to include some resource constraints on the optimization in terms of the dollars available for such improvements.

Table 3. Initial and Final Nominal Parameter Settings

Variables	Units	Initial	Final
C_{A_i}	mole/m^3	3,118	3,119.8
C_{B_i}	mole/m^3	342.0	342.24
T_i	°K	314	350
Q	J/min	1.71×10^6	5.0×10^6
V	m^3	.0391	.05
F	m^3/min	.078	.043
T	°K	300.0	309.5
$\sigma^2_{R_B}$		1,638	232

Finally, we characterize the contribution (locally) of the various noise factors to the output variation in the production rate, by conducting a sensitivity analysis of the integration data for the initial and final choices of the nominal parameter settings. Table 4 provides the contribution of each noise factor to the total output variance (in % terms). This table evaluates the explanatory power of each input variable using a measure that provides the reduction in total (original) variance by fixing one factor at a time at its nominal value. The total variance in the production of component R_B is reported in the first row. The contribution of each noise factor has been calculated by fixing it at its nominal value while retaining the noise in all the other factors. This analysis has been conducted at both the initial and final settings. The analysis shows that, when the variable T_i and/or Q is fixed to its nominal value, the variance is reduced significantly both at initial and final conditions. Therefore, it is apparent that changing the nominal value of these parameters can reduce the variance, which is also reflected in the results in which the optimal nominal value for these parameters goes to its bounds. Notice from Equation (1) that the bulk temperature of the CSTR (T) is related to both Q and T_i and hence equally sensitive to both. From Equations (2)–(5) it is clear that the production rate of B is critically dependent on T. Note, however, that the function is highly nonlinear and this analysis only provides a local analysis at the two points on what appears to be a convoluted surface. The change in volume, V, is due to this nonlinearity. The search for a region in the parameter space with minimal impact on the production variance necessitates this change in volume although this variable does not locally affect the variance in the production level.

Table 4. Sensitivity Analysis of the Noise Factors

	Δ Variance, %	
Variables	Initial	Final
Total	1,699.0	232.0
C_{A_i}	6%	15%
C_{B_i}	.2%	.1%
T_i	94%	52%
F	2%	23%
Q	94%	53%
V	.2%	.1%

4. CONCLUSIONS

This article presented a new sampling technique based on Hammersley points. This new sampling technique is shown to have better uniformity properties, which reduces the computational intensity of stochastic optimization problems considerably. Because Taguchi's parameter design method for off-line control essentially involves the solution of stochastic optimization problems, it was found that this sampling technique is always preferred for parameter-design problems. This is because of its high precision and consistent behavior coupled with great computational efficiency. This method was illustrated for off-line quality control of a continuous-stirred tank.

ACKNOWLEDGMENT

We acknowledge the useful discussions with Mitchell J. Small and Max Henrion. We also thank our anonymous referees for their useful comments and suggestions, which have greatly helped improve the quality of this article. This work was supported in part by National Science Foundation grant DMS-9523589 (J.R.K.) and Department of Energy grant DE-AC21-92MC29094 (U.M.D.).

[Received December 1994. Revised January 1997.]

REFERENCES

Boudriga, S. (1990), "Evaluation of Parameter Design Methodologies for the Design of Chemical Processing Units," unpublished Master's thesis, University of Ottawa, Dept. of Chemical Engineering.

Diwekar, U. M., and Rubin, E. S. (1994), "Parameter Design Method Using Stochastic Optimization With ASPEN," *Industrial & Engineering Chemistry Research*, 33, 292–298.

Fang, K.-T., Wang, Y., and Bentler, P. M. (1994), "Some Applications of Number-Theoretic Methods in Statistics," *Statistical Science*, 9, 416–428.

Fox, B. L. (1986), "Algorithm 647: Implementation and Relative Efficiency of Quasi-Random Sequence Generators," *Association for Computer Machinery Transactions on Mathematical Software*, 12, 362–376.

Halton, J. H. (1960), "On the Efficiency of Certain Quasi-Random Sequences of Points in Evaluating Multi-Dimensional Integrals," *Numerical Mathematics*, 2, 84–90.

Hammersley, J. M. (1960), "Monte Carlo Methods for Solving Multivariate Problems," *Annals of the New York Academy of Science*, 86, 844–874.

Iman, R. L., and Conover, W. J. (1982), "Small-sample Sensitivity Analysis Techniques for Computer Models, With an Application to Risk Assessment," *Communications in Statistics—Part A, Theory and Methods*, 17, 1749–1842.

Iman, R. J., and Shortencarier, M. J. (1984), "A FORTRAN77 Program and User's Guide for Generation of Latin Hypercube and Random Samples for Use With Computer Models," NUREG/CR-3624, SAND83-2365, Sandia National Laboratories, Albuquerque, NM.

Kacker, R. S. (1985), "Off-line Quality Control, Parameter Design, and the Taguchi Method," *Journal of Quality Technology*, 17, 176–210.

Morgan, G., and Henrion, M. (1990), *Uncertainty: A Guide to Dealing With Uncertainty in Quantitative Risk and Policy Analysis*, New York: Cambridge University Press.

Nair, V. N. (ed.) (1992), "Taguchi's Parameter Design: A Panel Discussion," *Technometrics*, 34, 127–161.

Niederreiter, H. (1978), "Quasi-Monte Carlo Methods and Pseudo-Random Numbers," *Bulletin of the American Mathematical Society*, 84, 957–1104.

——— (1988), "Multidimensional Numerical Integration Using Pseudorandom Numbers," in *Stochastic Programming* (Programming Study, Vol. 27), eds. A. Prekopa and R. J.-B. Wets, Amsterdam: North-Holland, pp. 17–38.

——— (1992), *Random Number Generation and Quasi-Monte Carlo Methods*, Philadelphia: Society for Industrial and Applied Mathematics.

Papageorgiou, A., and Wasilkowski, G. W. (1990), "On Average Case Complexity of Multivariate Problems," *Journal of Complexity*, 6, 1–6.

Press, W. H., Flannery, B. P., Teukolsky, S. A., and Vetterling, W. H. (1986), *Numerical Recipes: The Art of Numerical Computing*, New York: Cambridge University Press.

Taguchi, G., and Wu, Y. (1980), *Introduction to Off-line Quality Control*, Nagoya, Japan: Central Japan Quality Control Association.