Monte Carlo Simulation and Random Number Generation

RODNEY F. W. COATES, SENIOR MEMBER, IEEE, GARETH J. JANACEK, AND KENNETH V. LEVER, MEMBER, IEEE

Abstract—This paper discusses methods of generating pseudorandom number sequences which may have predetermined spectral and probability distribution functions. Such sequences are of potential value in Monte Carlo simulation of communication, radar, and allied systems. The methods described here are particularly suited to implementation on microcomputers, are machine portable, and have been subjected to exhaustive investigation by means of both statistical and theoretical tests.

I. Introduction

IMAGINE that it is required to determine the area of a piece of paper—a page torn raggedly from a book, for example. Fix the paper to a larger dart-board of known diameter and, blindfolded, commence to throw darts at the board. Keep the total of both the number of throws which hit the board and the number of throws which hit the paper. The area of the paper is then readily calculated as the product of the area of the dart-board and the ratio of the number of hits on the paper to the number of hits on the board.

This illustration is, of course, almost trivial. However, it highlights several features of the Monte Carlo approach. At the most fundamental level, we note that our actual problem, *finding an area*, was subsumed into a less specific class of problems involving the *evaluation of a statistical measure*, in this instance finding an *average*. Furthermore, the requirement that the darts be thrown "blind" carries with it some presumption concerning impartiality.

Several further questions then follow. How may we compute random deviates? What will be their distribution as computed? How may their distribution be shaped to serve particular needs? Which of their statistical properties are of significance and how may those properties be tested? Are the properties machine dependent? Indeed, how important is true randomness, anyway, when we note that a purely deterministic and regular sampling of the dart-board surface would serve also to allow us to compute area?

II. Tutorial on Random Number Generation

Whatever sophisticated extensions of method we may subsequently discover, Monte Carlo simulation is based upon the generation of nominally uniformly distributed, pseudorandom variates and on the weighting of such variates to establish other required distribution functions. Application of the Monte Carlo method to a wide range of physical problems, together with discussions of various approaches to enhancing computational efficiency, are to be found in, for example, the standard text by Hammersley and Handscombe [1].

One technique for substantially reducing computational time in Monte Carlo simulation is known as *importance sampling*. Papers by Shanmugan and Balaban [2], Jeruchim [3], [4], and Davis [5] review and extend the method of importance sampling in the estimation of error rate in communication systems. Similarly, papers by Lank [6] and Mitchell [7] discuss application of this technique to radar system false alarm rate evaluation. In all instances, however, the need remains for rapid, reliable, well-tested, and machine portable pseudorandom number generators and it is to this problem that this paper is addressed.

One commonly used technique for generating pseudorandom variates is the *congruential method* [8] which may be defined by the algorithm

$$x_i = a \cdot x_{i-1} + b \pmod{c}; \qquad u_i = x_i/c.$$

This algorithm generates pseudorandom variates u_i, over the interval $0 < u_i < 1$, which are substantially uniformly distributed; we say that the random process is $U(0, 1)$. The algorithm computes the ith value of the intermediate variable x_i from its predecessor by multiplying the value of the predecessor by a (large) number a, adding another (large) number b, and repeatedly subtracting a third (large) number c. Since the repeated subtraction means that the x_i must lie between zero and $c - 1$, division by c will produce the required u_i: if c is sufficiently large, $(c - 1)/c$ will be close to unity. Although u_i is strictly discrete, if a, b, and c are indeed large, the number sequence adequately models, on the digital computer, a continuous random variable. This may be denoted as u and is statistically described by its probability density (or distribution) function

Manuscript received March 25, 1987; revised September 18, 1987. This work was supported by the U.K. Science and Engineering Research Council under Contract GR/D 27211.

R. F. W. Coates and G. J. Janacek are with the University of East Anglia, Norwich, England.

K. V. Lever is with Brunel University.

IEEE Log Number 8718201.

$$\begin{aligned} q(u) &= 1; \qquad 0 \leq u \leq 1 \\ &= 0 \qquad \text{elsewhere.} \end{aligned}$$

The constant term b should be relatively prime to c. Both the multiplier a and the constant b are selected so that statistical properties of the generated sequence x_i are as desired, namely that the repetition period of the sequence is as long as possible and that the speed of generation is as high as possible. The choice of the modulus c is determined by the base and capacity (numerical range) of the computer being used. These criteria clearly tell us that the congruential algorithm is not *machine portable*, at least in the form that is given above. Portability and other aspects of random number generation are subjects to which we shall return in Section III.

Having obtained a method, however basic, for computing a uniformly-distributed sequence, we may inquire as to how this sequence may be used to generate pseudorandom variates with some other distribution function.

Suppose that we wish to establish a new random variable v with distribution $p(v)$; we shall assume that $p(v)$ is a continuous function of v. One method is to calculate v from u by a functional transformation $v = f(u)$. This can be obtained by means of an argument employing the cumulative distribution function (cdf)

$$\begin{aligned} P(c) &= \Pr\{v \leq c\} = \Pr\{f(u) \leq c\} \\ &= \Pr\{u \leq f^{-1}(c)\} = f^{-1}(c) \end{aligned}$$

since the cdf of a $U(0, 1)$ variable is just the identity on the interval $[0, 1]$. Thus, we have the *inverse probability integral transform*

$$v = f(u) = P^{-1}(u)$$

equivalent to the relationship

$$u = \int_{-\infty}^{v} p(v') \cdot dv' = P(v).$$

In some cases, this relationship will yield an expression which may be inverted explicitly in terms of elementary functions, thereby giving v as a function of u, as required.

By way of example, consider the one-sided exponentially distributed pseudorandom variate, which has been used in communication engineering for modeling the postdetection distribution of narrow-band noise. We may proceed directly from a knowledge of $P(v)$

$$\begin{aligned} u = P(v) &= 1 - \sigma^{-1} \exp(-v/\sigma) \qquad v > 0 \\ &= 0 \qquad\qquad\qquad\qquad\quad v < 0 \end{aligned}$$

where both the mean and rms signal levels are given by σ. We can invert this relation to find

$$v = \sigma \cdot \ln(1/(1 - u)) \qquad 0 \leq u < 1.$$

For less tractable probability distributions, we may need to resort to numerical evaluation of the integral and subsequent table lookup or a numerical approximation by curve fitting. Thus, computation of normally distributed (Gaussian) pseudorandom variates of zero mean and unity standard deviation may be achieved by applying the transformation obtained by the lambda distribution method [9]

$$v_i = 4.91\left(u_i^{0.14} - (1 - u_i)^{0.14}\right).$$

The accuracy of the method, which produces Gaussian variates in the range $|v_i| < 4.91$ becomes somewhat degraded for $|v_i| < 2.4$, and the execution is slowed down by the lengthy exponentiation process. The algorithm is therefore not a particularly good one and a number of improvements have been discussed in the statistical literature; we refer to these briefly in Section IV.

III. Design of Portable Uniform Random Number Generators

The development of uniform random number generators for mainframe computers, as chronicled, for example, by Knuth [10], appears to have been fraught with difficulties. Many generators—some still in current use—have been shown to be inadequate with respect to the various tests of randomness that can be applied. The same problems have been encountered in attempting to transfer Monte Carlo analysis from the mainframe to the desk-top minicomputer and the engineering workstation.

These problems were recently overcome, however, by the publication by Wichmann and Hill [11], [12] of an exceptionally effective algorithm suitable for 8- and 16-bit machines. The algorithm has two attractive properties. First, the algorithm is so simple and the code so short that it is easily transportable from machine to machine, whatever the language or operating system. The Fortran code is given in Appendix A. Furthermore, and equally importantly, the algorithm is very reliable in that it passes all of the large battery of *statistical* tests reportedly applied to it. We shall not, however, be concerned with such tests and we refer the interested reader again to Knuth [10]. We have good reason for rejecting statistical tests: Knuth has convincingly argued that statistical tests are not enough, and that *theoretical* tests should also be applied where possible. This argument has some force: for generators that appear to be satisfactory by passing statistical tests (on the sequence of numbers produced by the generator) can be shown to be poor by failing theoretical tests (on the *structure* of the algorithm). Particularly powerful in this respect is the *spectral* test, first developed by Coveyou and MacPherson [13] and we will give a brief account of its formulation and use.

Our own contribution [14] to this subject has been to show how the Wichmann–Hill algorithm, a combination of three not particularly good generators, achieves its high performance. We have two ways of modeling the behavior of the algorithm. The first makes use of a statistical central limit argument originally developed to analyze the flatness of quantizing noise spectra under very general conditions [15], [16]. This results in a convincing explanation of why the performance is so good with only three subgenerators. But the greater insight is obtained by means of an entirely different approach employing number theory. We show that the Wichmann-Hill composite generator is equivalent to a single linear congruential generator with a period very much longer than that of any of its constituent generators. The equivalent single generator would not be directly implementable on the 8- or 16-bit machines for which the Wichmann-Hill generator was designed, and we can regard this decomposition into "smaller" subgenerators as an example of the "divide-and-conquer" [17] strategy currently popular in computing science.

The increased length of the equivalent single generator in itself provides an intuitive understanding of the origin of the algorithm's excellent performance, but there is a much more important repercussion: we are able to apply the Coveyou-MacPherson spectral test to the Wichmann-Hill algorithm, where previously it was thought inapplicable. In this way we show that the algorithm is very good indeed, satisfying the criteria advocated by Knuth and passing the test with flying colors. Furthermore, our improved understanding of the structure of the algorithm allows us to design over 100 new generators of the Wichmann-Hill type, which exhibit much the same high performance. Thus, we can say that the original version of the algorithm is, for all practical purposes, optimal within this class. It is also rather easy to design Wichmann-Hill type algorithms for 32- and 64-bit machines; we give an example of one such design, together with the results of the spectral test confirming its high performance.

The Wichmann-Hill scheme makes use of three inferior linear, multiplicative, congruential generators

$$x_i = ax_{i-1} (\text{mod } p) \qquad y_i = by_{i-1} (\text{mod } q)$$
$$z_i = cz_{i-1} (\text{mod } r)$$

where p, q, and r are distinct primes. These are then combined to provide a normalized sequence confined to the interval $[0, 1]$ as follows:

$$u_i = (x_i/p + y_i/q + z_i/r) \, (\text{mod } 1).$$

It is well known (see, for example, [18], [19]) that the fractional part of the sum of K independent $U(0, 1)$ random variables is also $U(0, 1)$. It is also clear that if the constituent variables are not quite uniformly distributed, then the fractional part of their sum will be more nearly uniform. This is one way of looking at the efficacy of the Wichmann-Hill algorithm.

We model the pdf's, $p_n(v)$, of the constituent, nearly uniformly distributed sequences as linear combinations of a perfectly uniform component $u(v)$ and pertubation components $\epsilon_n(v)$ on the interval $[0, 1]$

$$p_n(v) = (1 - \lambda_n) \, u(v) + \lambda_n \epsilon_n(v)$$

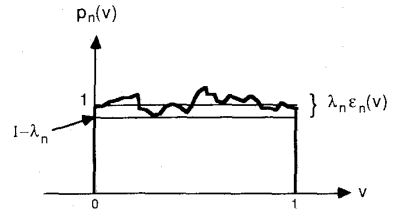


Fig. 1. Wichmann-Hill constituent pdf $p_n(v)$ and its perturbation pdf $\epsilon_n(v)$.

for $n = 1, 2, \cdots, K$. These relationships are depicted in Fig. 1.

By putting $\lambda_n = \min\{1 - p_n(v)\}$ in the interval $[0, 1]$ we ensure that $\epsilon_n(v)$ is also a pdf, and we are then able to show that the relevant measures of uniformity are logarithmic (due to the exponential-with-N approach to central-limit uniformity), so that the departure from uniformity of the constituent distributions is given by the logarithm (base 10) of the geometric mean of the λ_n

$$F = \left|\log (\Pi\lambda_n)^{1/K}\right| = \left|\frac{1}{K} \Sigma \log (\lambda_n)\right|.$$

A similar logarithmic measure is used to characterize the departure from uniformity of the resultant composite pdf, which is roughly sinusoidal, peaking near to $v = 1/2$ and falling to a minimum near the extremes $v = 0$ and $v = 1$. The ratio $r(K)$ of maximum to minimum, for K constituents, is very close to unity, so we define the departure from uniformity $F(K)$ as

$$F(K) = \left|\log (1 - r(K))\right|.$$

It turns out that we can find a very simple, general criterion relating K to these measures of uniformity, and to the average θ of the mean-square values of the perturbation distributions

$$\theta = \frac{1}{K} \sum_{n=1}^{K} (\sigma_n^2 + \mu_n^2).$$

Using the "ceiling" notation $\lceil x \rceil$ to indicate the nearest integer higher than x, we obtain the result that, taking

$$K = \left\lceil ((0.602 + F(K))/(8.57\theta + F)) \right\rceil$$

ensures that K nearly uniform distributions, flat within 10^{-F} on average, will provide a Wichmann-Hill resultant flat to within $10^{-F(K)}$, where θ is the mean-square perturbation value, averaged over the ensemble of perturbation components.

It is not difficult to show that the worst case occurs when each perturbation consists of a single spike or "glitch" at $v = b_n$; we have $\mu_n = b_n$ and $\sigma_n = 0$. The case where the expectation $E\{\mu_n\} = 0$ is certainly the worst case, as θ then takes its minimum value (zero), so that K takes its maximum value. Even with F as large as 2 (1 percent departure from uniformity) only $K = 4$ subgenerators are required to ensure $F(K) = 7$; that is, overall uniformity to within 0.00001 percent. Less extreme cases can be shown to achieve the same result with only three subgenerators—a result that appears to be virtually independent of the character of the perturbation distribution. Thus, we obtain a theoretical result in conformity with the choice $K = 3$ made on an empirical basis by Wichmann and Hill.

We now consider the alternative, number theoretic model of the Wichmann–Hill algorithm. It can be shown [14] that the Wichmann–Hill generator can be modeled as a single multiplicative, linear congruential generator

$$w_i = (qrM_1a + prM_2b + pqM_3c)\, w_{i-1} (\text{mod } M)$$

where M = pqr and M_1, M_2, and M_3 are chosen so that

$$qrM_1 \equiv 1(\text{mod } p) \qquad prM_2 \equiv 1(\text{mod } q)$$

$$pqM_3 \equiv 1(\text{mod } r).$$

For the original Wichmann–Hill generator we have

$$x_i = 171x_{i-1}(\text{mod } 30269) \qquad y_i = 172y_{i-1}(\text{mod } 30307)$$

$$z_i = 170z_{i-1}(\text{mod } 30323)$$

and

$$w_i = 16{,}555{,}425{,}264{,}690w_{i-1}$$

$$\cdot\ (\text{mod } 27{,}817{,}185{,}604{,}309).$$

Since M = pqr is composite, the period of this generator is not $M \cong 2.8 \times 10^{13}$, but $\lambda(M) = \text{lcm}(p-1, q-1, r-1)$. Since

$$p - 1 = 30268 = 2^2.7.23.47$$

$$q - 1 = 30306 = 2.3.5051$$

$$r - 1 = 30322 = 2.15161$$

we have

$$\lambda(M) = 2^2.3.7.23.47.5051.15161$$

$$\cong 7.0 \times 10^{12}$$

which should be long enough for most purposes. Notice the correction to the algorithm as presented in [11] to be found in [12] concerning the length of the period; this does not affect the validity of the algorithm.

The spectral test, as its name implies, is based on a k-dimensional Fourier transform of generated samples taken k at a time. The derivation of the theory behind the test is beyond the scope of this paper; details can be found in Knuth [10]. The basic idea is that all pseudorandom generators, being finite-state machines, exhibit periodicity in the space of k-tuples, while genuinely random generators would not. This periodicity is manifested by the k-tuples lying on hyperplanes given by the expression

$$x_1 + ax_2 + a^2x_3 + \cdots + a^{k-1}x_k = 0(\text{mod } \lambda(M)).$$

It is the lack of this rather subtle higher dimensional regularity that has been the downfall of many generators satisfactory with respect to lower dimensional criteria. It can be shown that the smallest nonzero wave number in the spectrum of k-tuples is given by

$$\nu_k = \min\left\{(x_1^2 + x_2^2 + \cdots + x_k^2)^{1/2}\right\}.$$

Since $1/\nu_k$ is the maximum distance between hyperplanes, a good generator would exhibit least granularity if it had very large values of ν_k. Knuth suggests the design criterion

$$\nu_k \geq 2^{30/k}.$$

He also suggests that the number of k-tuples falling in the k-dimensional ellipsoid surrounding the origin is a measure of the k-dimensional randomness. Since this is proportional to the volume of the ellipsoid

$$\mu_k = \frac{\pi^{k/2}\nu_k^k}{(k/2)!\ \lambda(M)}$$

the larger the value of μ_k, the better. Knuth recommends $\mu_k > 0.1$; the generator is very good if $\mu_k > 1$. (Note that we are using μ_k here in a different sense to the previous use as the mean of the perturbation distribution.)

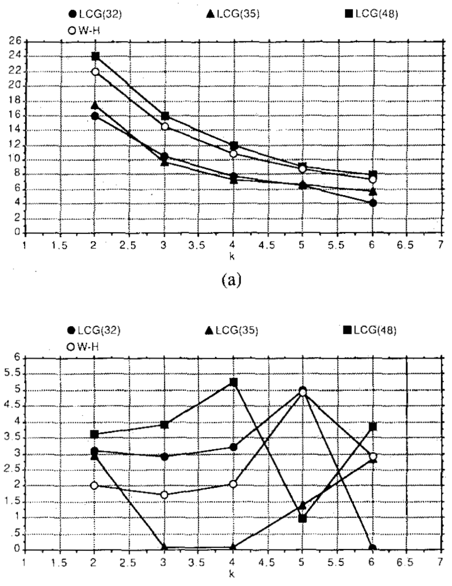
Fig. 2 gives a comparison of the Wichmann–Hill generator and selected linear congruential generators available on mainframes. As can be seen, the performance is excellent—almost as good as a mainframe generator with modulus 2^{48}. LCG(32) refers to the linear congruential generator with $a = 69060$, $M = 2^{32}$. Similarly, LCG(35) refers to the case $a = 19935388837$, $M = 2^{35}$; and LCG(48) to the case $a = 31167285$, $M = 2^{48}$. These data, for the linear congruential generators, are taken from Knuth [10]. The Wichmann–Hill case (W–H) is shown here, and elsewhere in what follows, as open circles.

Other generators of the Wichmann–Hill type can be found easily by restricting the primes p, q, and r to the form $2p^* + 1$, $2q^* + 1$, and $2r^* + 1$ where p^*, q^*, and r^* are also prime. This gives maximal period, as the lcm of $p - 1$, $q - 1$, and $r - 1$ is then $2p^*q^*r^*$. A search has revealed 16 suitable primes, together with the corresponding primitive elements used as multipliers. Of the

$$\binom{16}{3} = 560$$

possible cases about 80 percent can be dismissed as having multipliers which are factors of the moduli or as being defective in other ways, leaving over 100 good generators with $\mu_k > 1$ for $k = 2, 3, 4, 5, 6$ that convincingly pass the spectral test. All these cases have similar performance to the original Wichmann–Hill generator, which is therefore as good a representative of this class of good generators as any other.

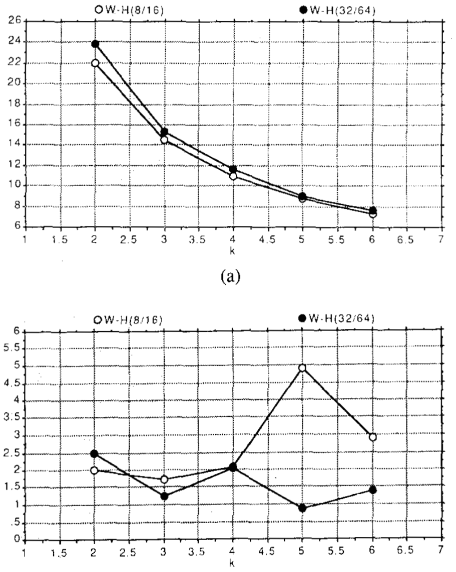
Similar ideas can be used to extend the design to generators suitable for 32- or 64-bit machines. The following example just outperforms the original Wichmann–Hill in respect to ν values and only falls a little below $\mu_k = 1$ for



(a)

(b)

Fig. 2. The spectral test: comparison between Wichmann-Hill and mainframe generators: (a) log ν_k versus k; (b) μ_k versus k.



(a)

(b)

Fig. 3. The spectral test: 8/16 bit and 32/64 bit Wichmann-Hill generators: (a) log ν_k versus k; (b) μ_k versus k.

the case $k = 5$:

$$x_i = 249x_{i-1}(\text{mod } 61967)$$

$$y_i = 251y_{i-1}(\text{mod } 63443)$$

$$z_i = 252z_{i-1}(\text{mod } 63599).$$

The period $\lambda(M) = 2.36983.31721.31799 \cong 7.5 \times 10^{13}$ should be satisfactory. Fig. 3 compares the performance of this generator to that of the original Wichmann-Hill generator.

IV. Nonuniform Random Number Generators

In this section we address two problems: how to design generators with specified first-order (marginal) probability distributions; and how to do the same thing when both the first-order distribution and the power spectrum are specified. The first problem has been intensively studied and a large number of solutions proposed. In general, there are three methods available:

1) inversion, exact or approximate
2) rejection techniques
3) composition techniques.

A good review of these methods can be found in [20], and we will not discuss the general case further. We shall concentrate instead on the example of the Gaussian case, for two reasons: it serves to illustrate the basic ideas and leads us to the starting point for tackling the second problem outlined above.

For the generation of Gaussian variates, there are several approximate methods available. We have already mentioned the lambda-function approximation

$$x = c\{u^\lambda - (1-u)^\lambda\}$$

with $c = 4.91$ and $\lambda = 0.14$, for generating approximately standard $N(0, 1)$ (zero mean, unit variance) Gaussian variates. Approximate inversion using a numerical algorithm to compute

$$x = \sqrt{2}\,\text{erf}^{-1}(2u - 1)$$

is sometimes faster, but better exact methods are available. The Box–Muller sine-cosine technique [22] for generating pairs of Gaussian variates (x_1, x_2) from pairs of uniform variates (u_1, u_2) is widely known but the simplified *polar method* referred to in [23] is faster

$$x_1 = \alpha u_1; \qquad x_2 = \alpha u_2 \quad \text{where } \alpha = \sqrt{\frac{-2 \ln W}{W}}$$

and $W = u_1^2 + u_2^2$ subject to $u_1^2 + u_2^2 \leq 1$.

Other methods are faster still [24], [25] and use a combination of the above and rejection and composition techniques. The fastest techniques exploit, in all or part, the use of machine language—but this clearly conflicts with the need to keep the algorithm reasonably portable. We have chosen, for the purposes of what follows, to use the polar algorithm.

We now consider the generation of variates having both a specified probability distribution and a specified power spectrum. It is clear that the transformation used to convert a $U(0, 1)$ variate to some other distribution, being a memoryless nonlinearity, might have a drastic effect on the originally white spectrum of the input process. It may or may not be possible to calculate the resultant spectrum. Even if it were possible, it is clear that one has no opportunity to design the spectrum of the output process; one must take what one gets.

Attempts to reshape the output spectrum by means of linear filtering are confounded, as filtering will perturb, perhaps ruin, the shape of the probability distribution that the nonlinear transformation was designed to provide. Successive approximation might be possible, by means of a series of nonlinear and linear transformations, but the prospect is not attractive. There is, in fact, no need to go to such lengths; the technique originated by Sondhi [26] provides a reasonably direct route to the desired result. His approach starts with the observation that there is one exception to the assertion that spectral shaping by linear filtering tends to distort the probability distribution: the Gaussian case. It is well known that a linearly filtered Gaussian is still Gaussian, and any changes in the mean or variance can be remedied, if required, by simple adjustments to the signal dc and rms levels. Sondhi's technique therefore starts with a white Gaussian input, subjects it to linear filtering (which leaves it Gaussian) to obtain a pre-emphasized spectrum. This is then followed by a memoryless nonlinear transformation, of (say) the inverse probability integral transform type, to obtain the desired marginal probability distribution. This nonlinearity will also change the spectrum of the filtered Gaussian process to the required output spectrum. Thus, the filter does not on its own provide the required spectrum but provides a predistortion shaping, so that when it is further distorted by the action of the nonlinearity, the required spectrum results, at least approximately.

Given that the Gaussian process taken as the starting point by Sondhi will, in practice, be obtained by means of a memoryless nonlinear operation on a $U(0, 1)$ process, we can recognize the Sondhi algorithm as a special case of Oppenheim's canonical form of a homomorphic signal processing system [27]. In such systems, the memory is concentrated in the central core processor, while the nonlinearity is confined to the input preprocessor and the output postprocessor. Since both nonlinearities are monotonically increasing, being derived (at least notionally) from cumulative distribution functions, they are indeed homomorphisms, preserving the algebraic structure of the space of input processes. The algorithm, or architecture, can be summarized as in Fig. 4.

Notice that there is an important point here: the uniform-to-Gaussian transformation $x = f(u)$ is assumed *not* to alter the white spectrum of the input $U(0, 1)$ process. Intuitively, it could be argued that as the nonlinearity is memoryless, an impulsive autocorrelation should remain impulsive after the transformation—but this scarcely constitutes a rigorous argument. We are uncertain whether this property has been shown to hold *theoretically*. Accordingly we subjected the polar Gaussian generator to stringent *statistical* tests of spectral whiteness; the generator appears to be satisfactory in this respect. It is interesting to speculate whether this property can be established, either exactly or approximately, by means of techniques similar to those in Section III.

Note also that Oppenheim has proved that the decomposition shown in Fig. 4 is canonical. We can interpret this to mean that, although a more convenient modification might be found, it is not likely to be simpler; Fig. 4 shows an architecture of minimum complexity.

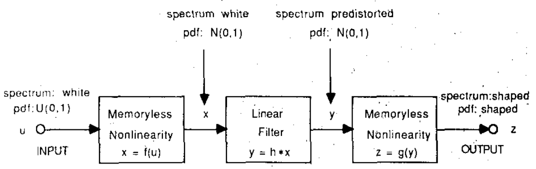


Fig. 4. Oppenheim canonical form of Sondhi algorithm.

The details of the Sondhi algorithm can be found in [26], but a simple example will clarify the idea. We may start with a $N(0, 1)$ white Gaussian generator derived by means of the polar transformation from the Wichmann–Hill $U(0, 1)$ white uniform generator. Our target is to design a generator having a one-sided negative-exponential probability distribution $p(z)$ (with $\mu = 2$, $\sigma^2 = 4$, and a single-pole spectrum $W_z(\omega)$ with $\alpha = 0.7$):

$$p(x) = \begin{cases} \frac{1}{2} \exp\left(-\frac{1}{2}x\right) & x > 0 \\ 0 & x < 0 \end{cases}$$

$$W_z(\omega) = \frac{1}{2\pi \left| 1 - \alpha^2 \exp\left(-j\omega\right) \right|^2}.$$

This case is sufficiently simple to be analytically tractable. We find that

$$z = g(y) = y^2$$

and that the Gaussian $N(0, 1)$ input must have a spectrum similar to $W_z(\omega)$, but with α^2 replaced by α. The filter characteristic turns out to be

$$\left| H(\omega) \right|^2 = \left| \frac{1 - \alpha \exp\left(-j\omega\right)}{1 - \alpha^2 \exp\left(-j\omega\right)} \right|^2.$$

Figs. 5 and 6 show the waveform, the histogram of the probability distribution function, and the estimated spectrum for (one realization of) the input and output, respectively. Apart from the tests on the flatness of the Gaussian process alluded to above, we have double-checked the shaping of the input spectrum. A maximum entropy estimate [28] confirms that $\alpha_{\text{est}} = 0.718 \pm 0.022$ with $\mu_{\text{est}} = -0.01 \pm 0.11$ ($\alpha = 0.7$, $\mu = 0$), for the input process. The corresponding values for the output process are $\alpha^2_{\text{est}} = 0.496 \pm 0.028$ with $\mu_{\text{est}} = 2.04 \pm 0.15$ ($\alpha^2 = 0.49$, $\mu = 2$).

For less tractable cases, the main problem resides in the computation of the required filtering characteristic. The computation of the output nonlinearity is easy enough; it is merely the composite of the inverse of the target cdf with the cdf of the Gaussian

$$z = g(y) = P_z^{-1}\left(\tfrac{1}{2}\left(1 + \operatorname{erf}\left(y/\sqrt{2}\right)\right)\right).$$

We next find the coefficients $\{g_n\}$ of the Hermite–Fourier expansion of $g(\cdot)$ [29], [30] and then determine (numer-

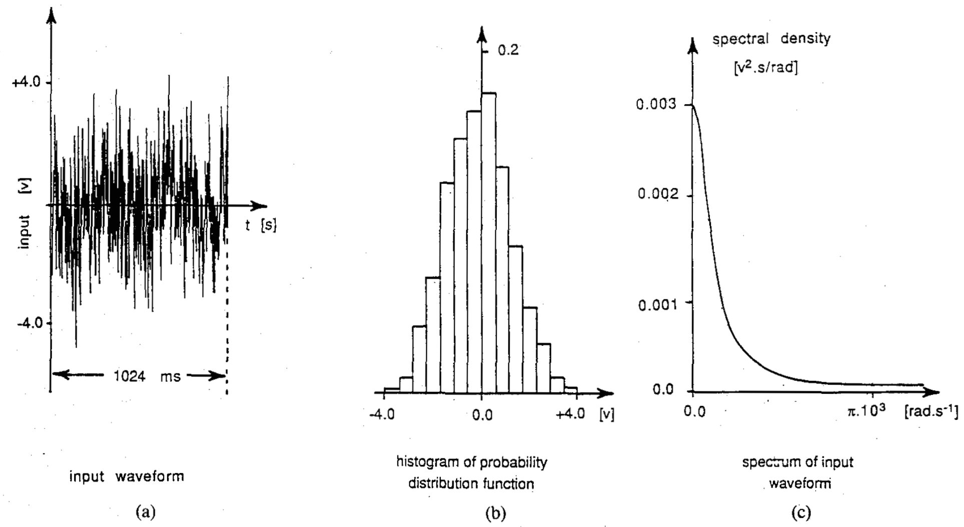


Fig. 5. Example of Sondhi transformation: input process (a) waveform, (b) histogram, (c) spectrum.

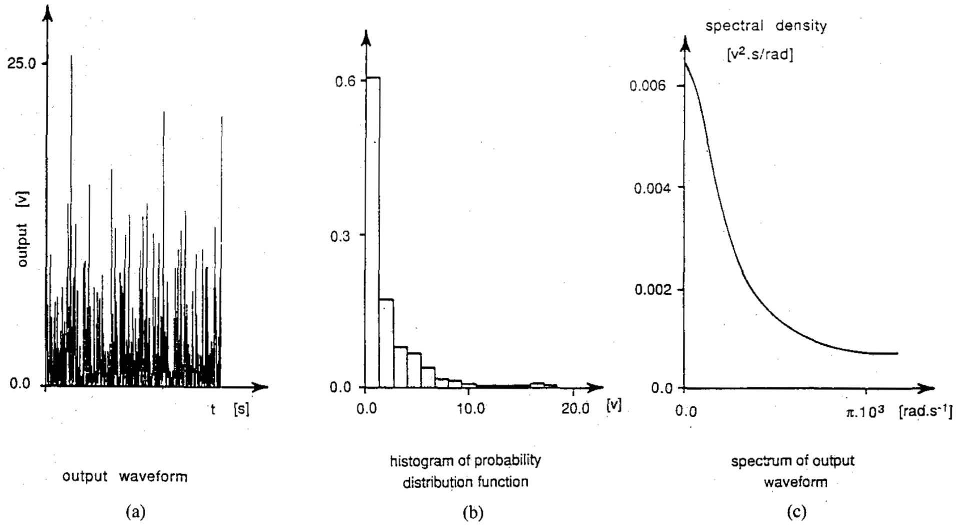


Fig. 6. Example of Sondhi transformation: output process (a) waveform, (b) histogram, (c) spectrum.

ically) the inverse $G^{-1}(\cdot)$ of

$$G(\rho) = \sum_{n=0}^{\infty} g_n^2 \rho^n.$$

Inverse Fourier transformation of the target spectrum (for example, by means of the FFT algorithm [31]) gives the target autocorrelation $R_z(\tau)$. The required filter impulse response autocorrelation $R_y(\tau)$ can then be shown to be

$$R_y(\tau) = G^{-1}(R_z(\tau)).$$

Fourier transformation of $R_y(\tau)$ will provide the specification of the power transfer characteristic of the predistortion filter. This can be regarded as a (digital) filtering problem in a signal processing setting, amenable to computer-aided design in the usual way [27]. Alternatively, it can be regarded as a model-fitting problem in a statistical setting, and ARMA (autoregressive moving average) techniques can be brought to bear upon it [32]. Either way it is clear that the Sondhi algorithm, unlike the Wichmann-Hill algorithm for the generation of uniform random variates, requires a substantial amount of numerical computation, some of it interactive. There is much work to be done before this process can be even partially automated, and portability seems far off at this time.

VII. Summary and Conclusions

Procedural methods for establishing pseudorandom sequences for use in, among other problem areas, the Monte Carlo simulation of communication radar, and other signal processing systems have been reviewed. The generation of pseudorandom variates of specifiable probability density distribution and power spectral density (or autocorrelation function) has been identified as an area of particular significance and difficulty.

The Wichmann–Hill approach to the generation of uniform random variates has been discussed. New evidence has been presented, first in statistical terms, by employing the concept of the perturbation distributions, and second in number theoretic terms, involving a spectral test on k-tuples sampled from the generated sequence, to explain the excellent performance of Wichmann–Hill type generators. The portability of random number generators in this family has also been noted. Finally, the difficulties in specifying both distribution function and power spectral density for a desired pseudorandom sequence have been discussed. The Sondhi approach to solving this problem has been described and the required algorithm illustrated by calculation of random variates of exponential distribution with a single-pole spectrum.

Appendix A
The Wichmann–Hill Algorithm

```
FUNCTION RANDOM(L)
    COMMON/RAND/IX, IY, IZ
    IX = 171 * MOD(IX, 177) - 2 * (IX/177)    |
    IY = 172 * MOD(IY, 176) - 2 * (IY/176)    |
    IZ = 170 * MOD(IZ, 178) - 2 * (IZ/178)    |
    IF (IX.LT.0) IX = IX + 30269              |
    IF (IY.LT.0) IY = IY + 30307              |
    IF (IZ.LT.0) IZ = IZ + 30323              |
    RANDOM = AMOD(FLOAT(IX)/30269.0
      + FLOAT(IY)/30307.0
      + FLOAT(IZ)/30323.0,1.0)
    RETURN
    END
```

Then the above version requires integer arithmetic up to 30323. Replacing the six instructions indicated with the three below produces identical results and may run slightly faster, provided integer arithmetic up to 5212632 is available.

```
IX = MOD(171 * IX, 30269)
IY = MOD(172 * IX, 30307)
IZ = MOD(170 * IX, 30323)
```

Both versions need to be seeded with IX, IY, and IZ (integer) values in the range 1 to 30000.

References

[1] J. M. Hammersley and D. C. Handscomb, *Monte Carlo Methods.* London: Methuen, 1964.

[2] K. S. Shanmugan and P. Balaban, "A modified Monte Carlo simulation technique for the evaluation of error rate in digital communication systems," *IEEE Trans. Commun.*, vol. COM-28, pp. 1916–1924, Nov. 1980.

[3] M. C. Jeruchim, "Techniques for estimating the bit error rate in the simulation of digital communication systems," *IEEE Trans. Select. Areas Commun.*, vol. SAC-2, pp. 153–170, Jan. 1984.

[4] ——, "On the application of importance sampling to the simulation of digital satellite and multihop links," in *Proc. IEEE Global Telecommun. Conf.*, New Orleans, LA, 1985, pp. 1088–1092.

[5] B. R. Davis, "An improved importance sampling method for digital communication system simulations," *IEEE Trans. Commun.*, vol. COM-34, pp. 715–719, July 1986.

[6] G. W. Lank, "Theoretical aspects of importance sampling applied to false alarms," *IEEE Trans. Inform. Theory*, vol. IT-29, pp. 73–82, Jan. 1983.

[7] R. L. Mitchell, "Importance sampling applied to simulation of false alarm statistics," *IEEE Trans. Aerospace Electron. Syst.*, vol. AES-17, pp. 15–24, Jan. 1981.

[8] D. H. Lehmer, "Mathematical methods in large-scale computing," in *Proc. 2nd Symp. Large-Scale Comput. Machinery*, Harvard University Press, 1951, pp. 142–145.

[9] B. L. Joiner and J. R. Rosenblatt, "Some properties of the range from Tukey's symmetric lambda distributions," *J. Amer. Statist. Ass.*, vol. 66, pp. 394–399, 1971.

[10] D. E. Knuth, "The art of computer programming," *Seminumerical Algorithms*, 2nd ed., Vol. 2. Reading, MA: Addison-Wesley, 1981.

[11] B. A. Wichmann and I. D. Hill, "An efficient and portable pseudorandom number generator," *Appl. Stat.*, Algorithm AS 183, pp. 188–190, 1982.

[12] ——, "A pseudorandom number generator," National Physical Lab. Rep. DITC 6/82, 1982.

[13] R. Coveyou and R. MacPherson, "Fourier analysis of uniform random number generators," *J. Assoc. Comput. Mach.*, vol. 1, pp. 100–119, 1967.

[14] G. J. Janacek and K. V. Lever, "High-performance easily-portable uniform random number generators: A 'divide-and-conquer' approach to complexity," presented at the 5th Int. Symp. on Data Analysis and Informatics, Versailles, Sept. 29–Oct. 2, 1987.

[15] K. V. Lever and K. W. Cattermole, "Quantizing noise spectra," *Proc. IEE.*, vol. 121, no. 9, pp. 945–954, 1974.

[16] K. V. Lever, "Quantizing noise spectra," in *Mathematical Topics in Telecommunications, Vol.* 2 (Problems of Randomness in Communication Engineering), K. W. Cattermole and J. J. O'Reilly, Eds. Pentech Press, 1984, pp. 199–217.

[17] D. F. Stanat and D. F. McAllister, *Discrete Mathematics in Computer Science.* Englewood Cliffs, NJ: Prentice-Hall, 1977, pp. 248–256.

[18] W. Feller, *An Introduction to Probability Theory and Its Applications*, Vol. 2. New York: Wiley, 1966, pp. 60–63.

[19] M. J. Lighthill, *Fourier Analysis and Generalized Functions.* Cambridge: Cambridge University Press, 1970, p. 21.

[20] A. C. Atkinson and M. C. Pearce, "The computer generation of Beta, Gamma and normal random variables," *J. Royal Statist. Soc.*, vol. 139, part 4, pp. 431–461, 1976.

[21] R. E. Odeh and J. O. Evans, "The percentage points of the normal distribution," *Appl. Stat.*, vol. 23, Algorithm AS 70, pp. 96–97, 1974.

[22] G. E. P. Box and M. E. Muller, "A note on the generation of random normal deviates," *Ann. Math. Statist.*, vol. 29, pp. 610–611, 1958.

[23] G. Marsaglia and T. A. Bray, "A convenient method for generating normal variables," *SIAM Rev.*, vol. 6, pp. 260–264, 1964.

[24] G. E. Forsythe, "Von Neumann's comparison method for sampling from the normal and other distributions," *Math. Comput.*, vol. 26, pp. 817–826, 1972.

[25] R. P. Brent, "A Gaussian pseudorandom number generator," *Commun. Ass. Comput. Mach.*, vol. 17, pp. 704–706, 1974.

[26] M. M. Sondhi, "Random processes with specified spectral density and first-order probability density," *Bell. Syst. Tech. J.*, vol. 62, pp. 679–700, Mar. 1983.

[27] A. V. Oppenheim and R. W. Shafer, *Digital Signal Processing.* Englewood Cliffs, NJ: Prentice-Hall, 1975.

[28] A. B. Baggeroer, "Confidence limits for regression (MEM) spectral estimates," *IEEE Trans. Inform. Theory*, vol. IT-22, no. 5, pp. 534–545, 1976.

[29] N. M. Blachman, "The uncorrelated outputs of a nonlinearity," *IEEE Trans. Inform. Theory*, vol. IT-14, pp. 250–255, 1968.

[30] ——, "The signal × signal, noise × noise and signal × noise output of a nonlinearity," *IEEE Trans. Inform. Theory*, vol. IT-14, pp. 21–27, 1968.

[31] E. O. Brigham, *The Fast Fourier Transform.* Englewood Cliffs, NJ: Prentice-Hall, 1974.

[32] R. L. Kashyap, "Optimal choice of AR and MA parts in ARMA model," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. PAMI-4, pp. 99–104, 1982.

Rodney F. W. Coates (M'81-SM'86) was born in Southhampton, England, on March 28, 1944. He received the B.S. degree from the University of Southampton, Southampton, England, in 1965 and the Ph.D. degree from the Queen's University, Belfast, Ireland, in 1968, where he specialized in computer modeling of communication systems.

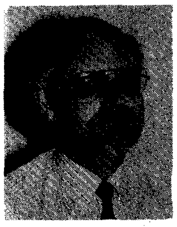
He began an academic career in 1968, as Lecturer in Electronics and a founding member of Staff of the Department of Electronics, Chelsea College, University of London. In 1971 he was appointed as a Lecturer to the Staff of the University College of North Wales, Bangor. In 1985 he was appointed Senior Lecturer in Electronic Systems Engineering at the University of East Anglia, Norwich, and in June 1987 was appointed to the Chair of Electronics in the School of Information Systems at the same university. He is the author of the book *Modern Communication Systems* (London: Macmillan), and numerous papers on communication engineering, oceanographic instrumentation, and underwater acoustics.

Dr. Coates is a member of the Communications, Ferroelectrics, and Frequency Control Society; the Acoustics, Speech, and Signal Processing Society; and the Oceanic Engineering Society. He is a Member of the Executive Committee of the United Kingdom and Republic of Ireland branch of the IEEE, for whom he is currently organizing a COMM/ASSP Chapter.

Gareth J. Janacek was born in Pontypool, Wales, in 1945. He received the B.S. degree in mathematics in 1966, the M.S. degree in applied statistics and random processing, both from the University of Manchester, Manchester, England, and the Ph.D. degree.

After working in medical statistics, he conducted postdoctoral work in time series analysis at the University of Nottingham, Nottingham, England. He was appointed Lecturer in Mathematics at the University of East Anglia, Norwich, England, in 1974. His main interests are in the study of the structure of multiple time series, mainly via state-space methods, and in statistical computing, especially the simulation of time series and the development of portable random number generators.

Dr. Janacek is a Fellow of the Royal Statistical Society.



Kenneth V. Lever (M'77) was born in London, England, on July 13, 1944. He received the B.S. degree in mathematics from the University of Liverpool, Liverpool, England, in 1966, and the M.S. degree in telecommunication systems from the University of Essex, Essex, England, in 1972.

He joined GEC at the Hirst Research Centre, Wembley, London, as Scientific Officer and went on to achieve the position of Principal Systems Engineer in 1976. He was appointed Lecturer in the Department of Electronics at Chelsea College, London, in 1977 and went on to lecture in Digital Communications at Brunel University in 1979. In 1982 he was a founding member of the Department of Electronic Systems Engineering, University of East Anglia, Norwich, England, where he lectured in communication engineering and signal processing in the School of Information Systems. He has recently been appointed to the GEC Readership in Telecommunications at Brunel University

Mr. Lever is the author of numerous research papers concerned with communication engineering and acts as a Consultant to several major U.K. companies and organizations.