Proceedings of the 1992 Winter Simulation Conference
ed. J. J. Swain, D. Goldsman, R. C. Crain, and J. R. Wilson

THE TES METHODOLOGY: MODELING EMPIRICAL STATIONARY TIME SERIES

Benjamin Melamed
Jon R. Hill

NEC USA, Inc.
4 Independence Way
Princeton, New Jersey 08540, U.S.A.

David Goldsman

School of Industrial and Systems Engineering
Georgia Institute of Technology
Atlanta, Georgia 30332, U.S.A.

ABSTRACT

Autocorrelated processes occur naturally in many domains. Typical examples include autocorrelated (bursty) job arrivals to a manufacturing shop or telecommunications network.

This paper presents a novel approach to input analysis of autocorrelated processes, called the TES (*Transform-Expand-Sample*) modeling methodology. TES is a versatile class of stochastic processes which can simultaneously capture both the marginal distribution and autocorrelation structure of a stationary (empirical) time series. In this paper we summarize the TES modeling methodology and briefly review a software environment, called TEStool, which supports this methodology through an interactive graphical user interface (GUI). The GUI greatly facilitates the process of fitting a TES model to empirical time series, by providing immediate feedback to modeling actions. We conclude the paper with a number of examples which demonstrate the efficacy of the TES methodology and the TEStool GUI by fitting TES models to empirical datasets obtained from actual field measurements.

1 INTRODUCTION

Autocorrelated processes occur naturally in many domains. For instance, the simulation practitioner might encounter autocorrelated job arrivals to a manufacturing shop or VBR (variable bitrate) compressed video in a telecommunications network (Lee et al. 1991, Melamed et al. 1992). This paper is concerned with the modeling and analysis of such autocorrelated processes.

A number of recent studies (Fendick et al. 1989, Heffes and Lucantoni 1986, Livny et al. 1993, and Patuwo et al. 1993) have shown that if autocorrelated customer interarrival times drive a queuing system, the resulting performance measures are much worse than those corresponding to renewal traffic (which ignores autocorrelations in interarrival times); for example, positively correlated interarrivals often cause an increase in the mean customer waiting time. The growing realization of the impact of dependence in traffic streams on system performance provides the motivation for devising Monte Carlo methods for modeling and generating autocorrelated variates with a wide variety of marginal distributions and autocorrelation structures (see, e.g., Bratley et al. 1987).

In practice, we aim to devise a model having:

1. a marginal distribution that matches its empirical counterpart.

2. an autocorrelation function that approximates its empirical counterpart.

3. sample paths that "resemble" the empirical data.

See Lewis and McKenzie (1991) and Schmeiser (1990) for some representative references.

In this paper we focus on a class of methods called TES (*Transform-Expand-Sample*), introduced in Melamed (1991), and further developed in Jagerman and Melamed (1992abc). TES methods can generate a variety of sample paths, and autocorrelation functions with a variety of functional forms (monotone, oscillating and alternating), while guaranteeing an exact match of the marginal distribution to its empirical counterpart. In addition, TES allows great latitude in approximating the empirical autocorrelations, even as it maintains the matching of the marginal distributions. Furthermore, the autocorrelation function of a TES model can be computed from fast numerical formulas without requiring simulation. This is important in interactive modeling, since the numerical computations are much faster than the corresponding simulation-based statistical calculations. Consequently, the modeling process can be implemented as a heuristic search on a large parameter space, and this search can be effectively speeded up by casting it into a visual interactive style.

This paper describes both the TES methodology, and a software environment, called TEStool, which supports this methodology. TEStool is written in C++ and runs on Sun workstation platforms under the OpenWindows display manager. It makes heavy use of visualization in order to provide a pleasant interactive environment supporting modeling of dependent time series. This environment speeds up the modeling search process, cuts down on modeling errors and relieves the tedium of repetitive search. In addition, it casts the search into an intuitive procedure, transforming it into a kind of video arcade game, and enabling its use by non-experts and experts alike.

2 TES METHODS FOR GENERATING AUTOCORRELATED SEQUENCES

TES processes are essentially autoregressive schemes with modulo-1 reduction. TES generation methods are easy to implement on a computer, enjoying a low computational complexity.

Let $\lfloor x \rfloor = \max\{n \text{ integer} : n \leq x\}$ be the integral part of x, and $\langle x \rangle = x - \lfloor x \rfloor$ be the fractional part of x. Let $\{V_n\}$ be a sequence of iid (independent identically distributed) random variables with a common, though arbitrary, density f_V. Further, let U_0 be uniform on $[0,1)$ and independent of each element of the sequence $\{V_n\}$. The random variables V_n are referred to as *innovations*.

Pure TES processes are stationary and come in two flavors: TES$^+$, giving rise to a sequence $\{U_n^+\}$, and TES$^-$, giving rise to a sequence $\{U_n^-\}$. The definition of $\{U_n^+\}$ is given recursively by

$$U_n^+ = \begin{cases} U_0 & \text{if } n = 0 \\ \langle U_{n-1}^+ + V_n \rangle & \text{if } n > 0 \end{cases} \quad (1)$$

and the sequence $\{U_n^-\}$ is defined in terms of $\{U_n^+\}$ by

$$U_n^- = \begin{cases} U_n^+ & n \text{ even} \\ 1 - U_n^+ & n \text{ odd} \end{cases} . \quad (2)$$

TES methods achieve coverage of the full range of feasible lag-1 autocorrelation; TES$^+$ methods cover the positive range $[0,1]$, while TES$^-$ methods cover the negative range $[-1,0]$. It can also be shown (Jagerman and Melamed 1992a) that (1) and (2) each give rise to a sequence of cid (correlated identically distributed) random variables with uniform marginals on [0,1). However, in practice, one is actually interested in transformed TES processes $\{X_n^+\}$ and $\{X_n^-\}$, obtained from (1) or (2) by some transformation D (called a *distortion*), i.e.,

$$X_n^+ = D(U_n^+), \quad X_n^- = D(U_n^-). \quad (3)$$

The transformation D, and the fact that the sequences $\{U_n^+\}$ and $\{U_n^-\}$ all have uniform marginals on [0,1), allow us to generate random sequences with essentially arbitrary marginals by using the inversion method; for a given distribution function F, the inversion method takes $D = F^{-1}$ (see, e.g., Bratley, Fox and Schrage 1987). The resultant sequences $\{X_n^+\}$ and $\{X_n^-\}$ are both cid with marginal distribution F. The corresponding autocorrelation functions are denoted by ρ_X^+ and ρ_X^-, respectively.

A Monte Carlo simulation of (1) and (2) can always provide an estimate of the model's autocorrelation function, if a sufficient sample size is generated. This approach, however, can be costly in terms of time complexity, especially when high autocorrelations necessitate large sample sizes for adequate statistical reliability. Fortunately, there are numerically computable formulas for the autocovariance functions and spectral densities of $\{X_n^+\}$ and $\{X_n^-\}$ which are fast, efficient and accurate (Jagerman and Melamed 1992abc). For a given lag τ, the corresponding autocorrelation functions are given, respectively, by

$$\rho_X^+(\tau) = \frac{2}{\sigma_X^2} \sum_{\nu=1}^{\infty} \Re[\tilde{f}_V^\tau(i2\pi\nu)] \, |\tilde{D}(i2\pi\nu)|^2 \quad (4)$$

and

$$\rho_X^-(\tau) = \begin{cases} \rho_X^+(\tau) & \tau \text{ even} \\ \frac{2}{\sigma_X^2} \sum_{\nu=1}^{\infty} \Re[\tilde{f}_V^\tau(i2\pi\nu)] \, \Re[\tilde{D}(i2\pi\nu)^2] & \tau \text{ odd} \end{cases} \quad (5)$$

where σ_X^2 is the common variance of $\{X_n^+\}$ and $\{X_n^-\}$, tilde denotes the Laplace Transform, f_V^τ is the τ-fold convolution of the innovation density f_V, $\Re$ denotes the real part operator, and $i = \sqrt{-1}$. It is worth noting that the effects of the innovation and distortion are conveniently separated in (4) and (5). More importantly, the series converge rapidly.

Since, in practice, marginal distributions of empirical data are represented by an empirical histogram H, we use the *histogram distortion*

$$D_H(x) = \sum_{k=1}^{N} 1_{[C_{k-1}, C_k)}(x) \, [\ell_k + (x - C_{k-1}) \frac{w_k}{p_k}] \quad (6)$$

where N is the number of histogram cells of the form $[\ell_k, r_k)$, $w_k = r_k - \ell_k$ is the width of cell k, 1_A is the indicator function of set A, p_k is the probability of cell

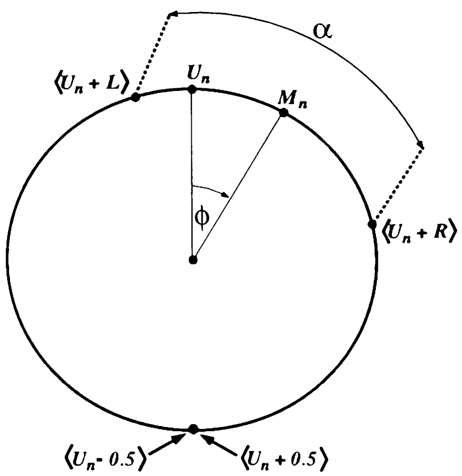

Figure 1: Recursive Generation of a Basic TES Process.

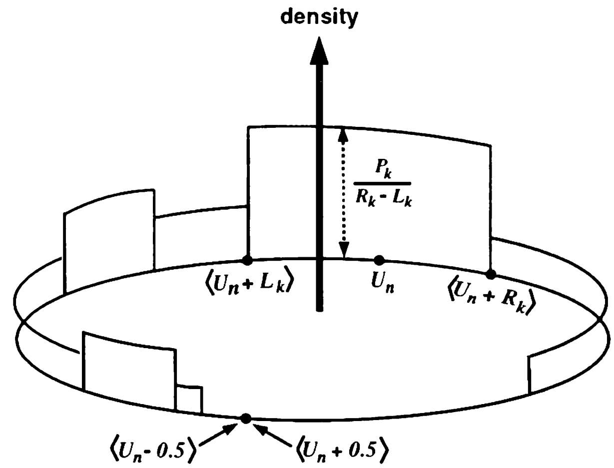

Figure 2: A 3-Dimensional Representation of a Step Function Density on the Unit Circle.

k, and $C_k = \sum_{j=1}^{k} p_k$ is the cumulative distribution of $\{p_k\}$ ($C_0 = 0$ and $C_N = 1$).

An important basic innovation, investigated in Melamed (1991), is given by

$$V_n = L + (R - L)Z_n, \quad -0.5 \leq L < R \leq 0.5, \quad (7)$$

where $\{Z_n\}$ is an iid sequence of random variables with uniform marginals on [0,1); that is, $\{Z_n\}$ is some pseudo-random number stream available on most computers, and the V_n are uniform on $[L, R)$. TES processes attendant to (7) are parameterized by pairs (L, R). It is often more convenient to use an equivalent parameterization (α, ϕ), where

$$\alpha = R - L, \quad \phi = \frac{R + L}{R - L} \quad (8)$$

where $0 < \alpha \leq 1$ and $0 \leq |\phi| \leq 1/\alpha - 1$.

Intuitively, the modulo-1 arithmetic used in defining TES processes in (1) has a simple geometric interpretation as a Markovian random walk on the unit circle (circumference 1), with random step size V_n. Consider the basic TES process defined by (7) and (8), and refer to Figure 1. Here L and R are measured as displacements from the current TES variate U_n (the relative origin), α is the length of the interval straddling U_n, and ϕ is an indication of the rotation of that interval from symmetric straddle (roughly speaking, ϕ is an indication of the angle between U_n and the interval midpoint M_n). If $\phi = 0$, the next TES iterate is equally likely to fall to the left or to the right of the current iterate. In this case, $E[V_n] = 0$, and the random walk has zero drift around the circle. Furthermore, $\rho_U^+(\tau)$ is monotone decreasing in the lag τ. If $\phi > 0$, we have $E[V_n] > 0$, and the drift is positive, resulting in cyclical sample paths with the attendant $\rho_U^+(\tau)$ oscillating about zero in the lag τ. The case $\phi < 0$ is analogous but with opposite drift.

A simple but considerable generalization of the basic innovation sequence $\{V_n\}$ given in (7) is attained by specifying f_V as a step function

$$f_V(x) = \sum_{k=1}^{K} 1_{[L_k, R_k)}(x) \frac{P_k}{R_k - L_k}, \quad (9)$$

so each V_n is a mixture of uniform variates on $[L_k, R_k)$ with probability P_k. A 3-dimensional representation of a step function density (9) appears in Figure 2. The generalized parameterization specifies a set of triplets $\{(L_k, R_k, P_k)\}_{k=1}^{K}$ where the step boundaries L_k and R_k satisfy $-0.5 \leq L_k < R_k \leq 0.5$, and $0 < P_k \leq 1$ is the probability of triplet k ($\sum_{k=1}^{K} P_k = 1$). For convenience, we also require that the triplets do not overlap (i.e., $R_k \leq L_{k+1}$, $k = 1, \ldots, K - 1$). Clearly, step function densities with support on $[-0.5, 0.5]$ can approximate arbitrarily closely any densities with the same support.

The sample paths of $\{X_n^+\}$ and $\{X_n^-\}$ often exhibit a visual "discontinuity" whenever the corresponding uniform random walks $\{U_n^+\}$ and $\{U_n^-\}$ "cross" point 0 on the unit circle in either direction (see Melamed 1991 and Jagerman and Melamed 1992a). In order to "smooth" TES sample paths or to skew them variously, one can use the so-called family of *stitching transformations* S_ξ, parameterized by $0 \leq \xi \leq 1$, and defined for all $y \in [0,1)$ by

$$S_\xi(y) = \begin{cases} \dfrac{y}{\xi}, & \text{if } 0 \le y < \xi \\[2ex] \dfrac{1-y}{1-\xi}, & \text{if } \xi \le y < 1 \end{cases} . \qquad (10)$$

Processes of the form $\{S_\xi(U_n^+)\}$ and $\{S_\xi(U_n^-)\}$ are called *stitched* TES processes. Note that for $0 < \xi < 1$, the S_ξ are continuous on the unit circle; in particular, $S_\xi(0) = S_\xi(1)$. It can be shown (Jagerman and Melamed 1992a) that all S_ξ, $0 \le \xi \le 1$, preserve uniformity; consequently, stitched TES processes can still be distorted to arbitrary marginals.

The reader is referred to Melamed (1991) and Jagerman and Melamed (1992a) for detailed discussions of the qualitative behavior of TES autocorrelation functions and sample paths as a function of the innovation density and the stitching parameter.

3 THE TES MODELING METHODOLOGY

The TES modeling methodology is supported by the TEStool software environment; consequently, it will be discussed from the vantage point of a user interacting with TEStool in order to fit a TES model to empirical data. A more detailed description of TEStool services is deferred until later.

A typical user would start out with a sample of some empirical time series data representing a partial process history (e.g., bitrates on a link in a communications network).

A complete specification of a TES model via the TES modeling methodology consists of two parts:

1. Specifying the distortion: As mentioned in Section 2, the distortion D takes a TES process, marginally uniform on $[0,1)$, and transforms each element in the time series via (3). TEStool provides distortions for uniform, exponential or geometric marginals. When modeling empirical data, D corresponds to the inverse distribution function of the empirical data, obtained from the empirical density function (itself modeled as a histogram); that is, $D = D_H$ as given in (6). TEStool automates the generation of D_H from empirical data. It enables the user to read in empirical datasets, to generate and display an empirical histogram of the data and to extract D_H from it.

2. Finding an innovation: The core activity of the modeling process in TEStool is an heuristic search for a suitable innovation. Recall that $\{X_n^+\}$ and $\{X_n^-\}$ always have the requisite distribution F *regardless* of the innovation $\{V_n\}$ selected. However, different innovations give rise to different stochastic processes with different autocorrelation functions, thereby providing great leeway in fitting autocorrelation functions. In TEStool, the user searches through innovations in the space of step function densities whose support is contained in $[-0.5, 0.5]$, in order to fit a prescribed autocorrelation function (usually, an empirical one). At the same time the user also looks for models whose simulated sample paths "resemble" their empirical counterparts. Additional details will be given in the sequel.

4 THE TEStool GRAPHICAL USER INTERFACE

TEStool is an interactive modeling environment (Geist and Melamed 1992) with a graphical user interface (GUI). It provides services to generate and modify TES models, and to examine their statistics. During a typical modeling session, the user conducts an heuristic search for innovations that give rise to TES models whose statistical characteristics approximate those of the empirical data.

TEStool classifies data into three dataset types. An *empirical* dataset contains data whose source is external to TEStool. The data consist of an empirical time series or statistics computed from it (histogram, autocorrelation function or spectral density). A *simulated* dataset contains data whose source is a Monte Carlo simulation of a TES model. The data consist of a sample path realization or statistics computed from it (histogram, autocorrelation function or spectral density). A *numerical* dataset contains the results of a numerical computation of the autocorrelation function, or the spectral density of a TES model, calculated via (4) and (5).

Figure 3 displays a TEStool screen as it might appear during a typical modeling session. The parent window is sub-divided into four canvas areas designed to display various types of color graphics and, at the bottom, two smaller text display panels that provide access to the host operating system and, in future versions, to display help messages. Located within the top margin of the screen is a row of buttons which create menus and forms that support operations not specific to any canvas:

1. The `View` button: controls the number of canvases displayed, ranging in number from one to four.

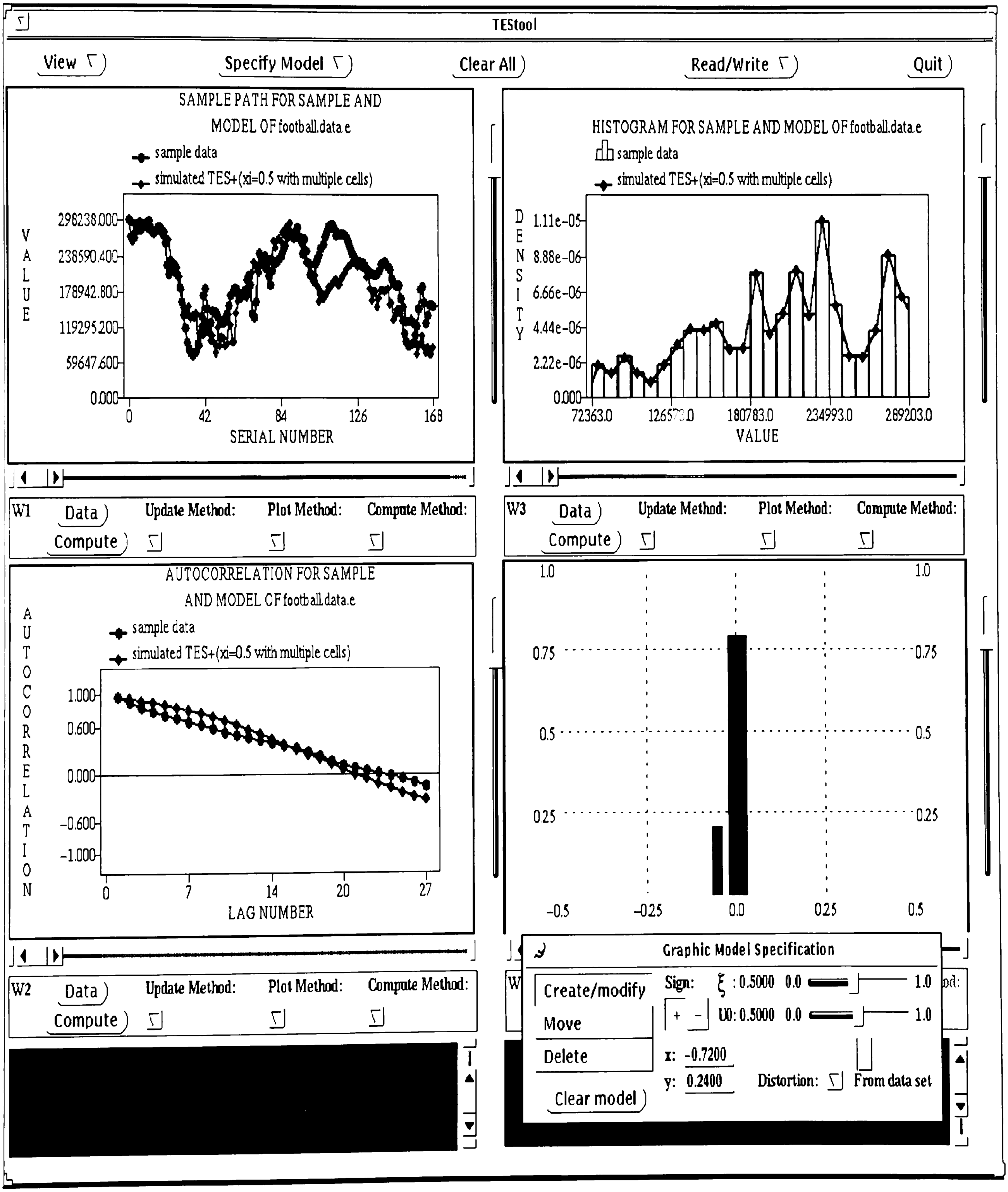

Figure 3: A Typical TEStool Screen in a Modeling Session of Empirical Time Series

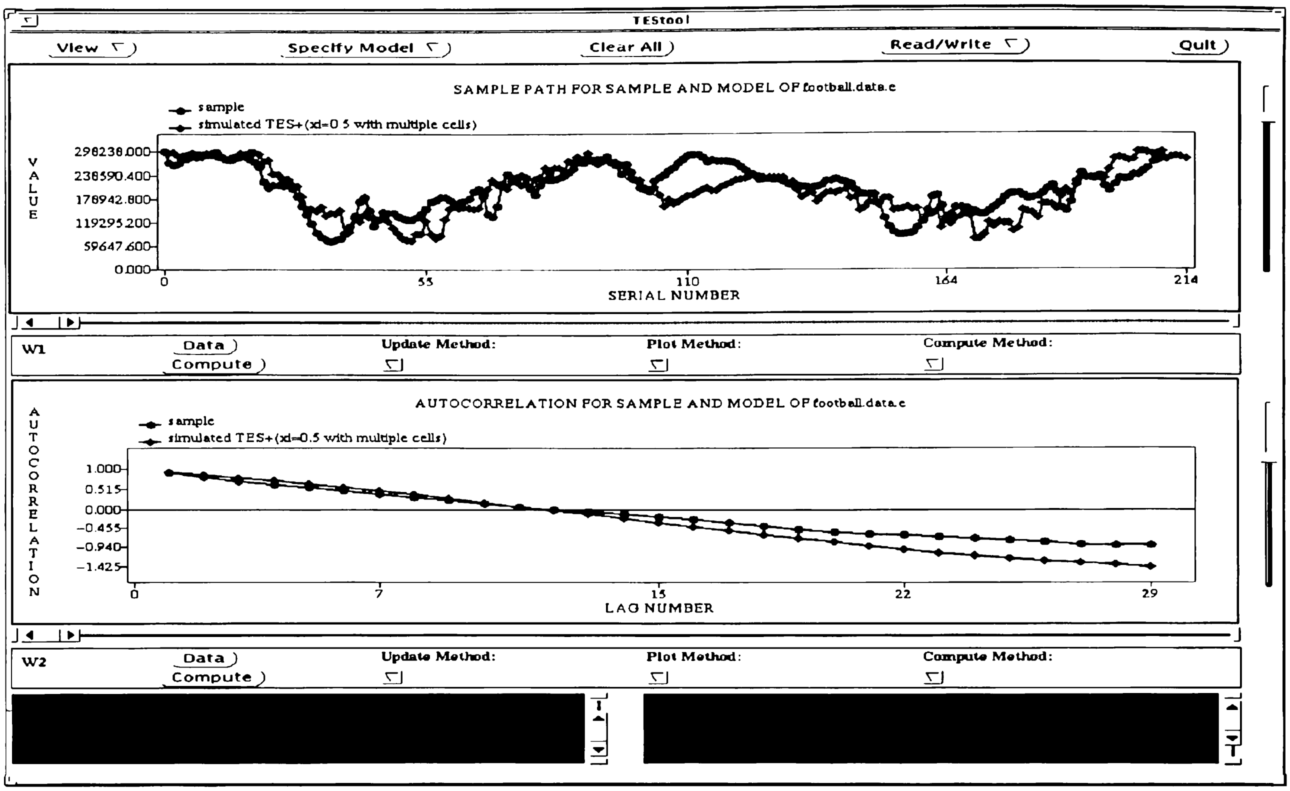

Figure 4: The Sample Path and Autocorrelation Plots Expanded Horizontally.

2. The `Specify Model` button: creates either text or graphic sub-windows to construct a TES model.
3. The `Clear All` button: deletes all data and models and clears all canvases.
4. The `Read/Write` button: creates a menu allowing the user to save the current modeling session to disk or to restore a previous session from disk.
5. The `Quit` button: requests user confirmation to exit from TEStool.

Directly beneath each of the four display canvases is a control panel containing menu-buttons and selection gadgets specific to each canvas; these allow a user to populate a canvas with a variety of graphic displays and to tailor their appearances. The buttons and selection gadgets in each control panel have the following functions:

1. The `Data` button: creates a menu allowing selection of various types of diskfile input/output operations.
2. The `Compute` button: creates a menu allowing assignment of the canvas to one of four classes of graphical displays supported by TEStool (time series, histogram, autocorrelation or spectral density).
3. The `Update Method` selector: determines whether computations are performed only on user command, or interactively in response to modifications to the current model.
4. The `Plot Method` selector: determines whether to create a new plot or redraw the existing plot for each computation. Multiple plots may be displayed on the same canvas as long as they are all of the same class.
5. The `Compute Method` selector: determines whether empirical, simulation or analytic methods are to be used to compute the graphical display class assigned to the canvas.

4.1 The Graphic Model Specification Window

The TEStool GUI provides both graphical and textual windows for the creation and modification of both the innovation density and the stitching parameter. A step function innovation density may be spec-

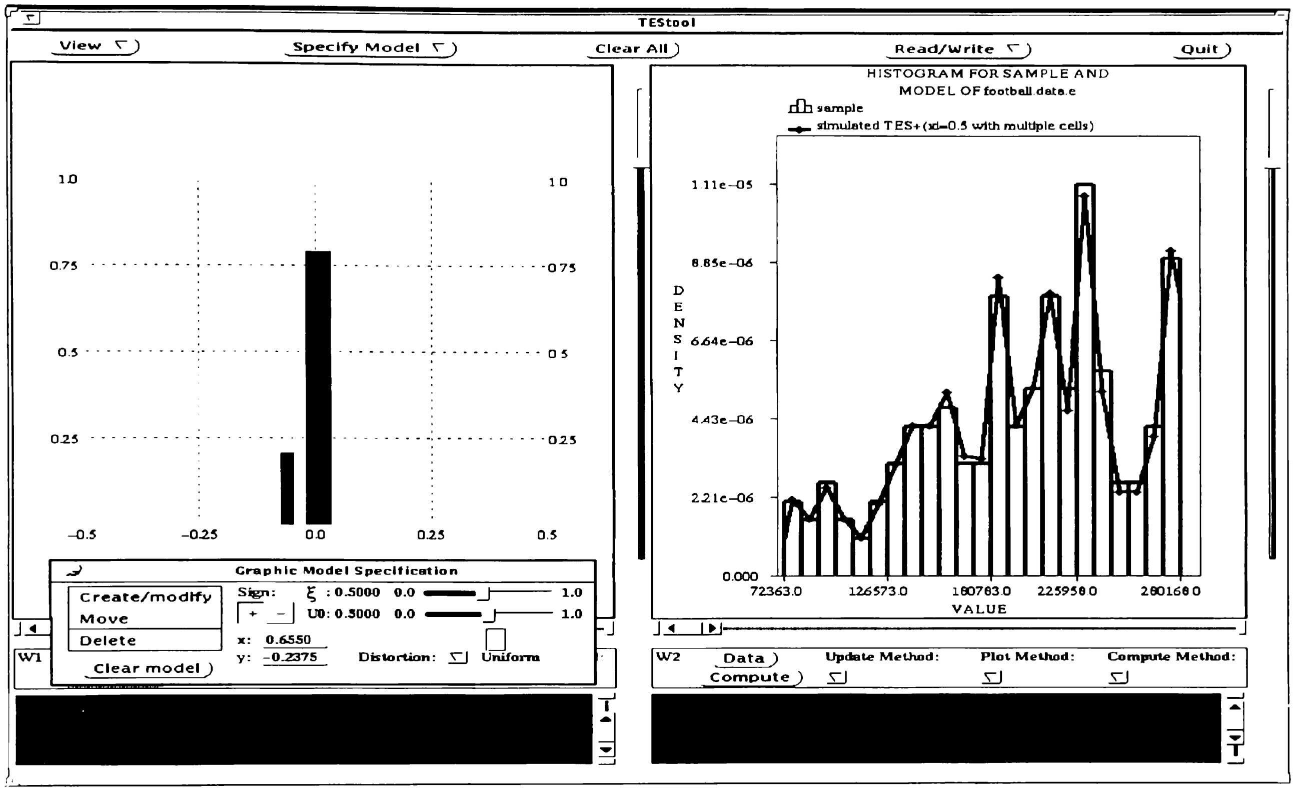

Figure 5: The Innovation Step Density (left); the Empirical and TES Simulation Histograms (right)

ified visually (see the left side of Figure 5) by graphically constructing a representation of Figure 2 as it would appear if it were unwrapped and mapped onto a 2-dimensional plane. This is accomplished by using the workstation mouse to sweep out a set of one or more non-overlapping rectangles, which represent the steps of an innovation density, on a rectangular planar grid, as shown in the left canvas of Figure 5. All points on the horizontal axis of the innovation canvas correspond to positions on the unit circle illustrated in Figure 2, and thus range from -0.5 to 0.5, while the vertical axis values represent probabilities and so range from 0.0 to 1.0. Each step may be represented by a (L_k, R_k, P_k) triplet defined in the following way: the base of any step k always rests on the X axis; the left and right step edges correspond to L_k and R_k, respectively, and the height of the step corresponds to P_k. This representation is not a valid step probability density function because the total area under the steps is not constrained to sum to one. However, this method of representing a step function density is more useful for the current application, since it has the advantage of being easier to manipulate graphically. The graphical advantage follows from the fact that the sum of the heights of the steps is bounded by one, whereas the height of a standard density is unbounded.

The TEStool innovation canvas allows the user to add and modify as many steps as desired, but rigidly enforces the constraint $\sum P_k \leq 1.0$. The innovation software continuously monitors the value of this sum and truncates any step at the height which exactly satisfies $\sum P_k = 1.0$. If $\sum P_k < 1.0$, the TEStool computation modules will refuse to perform any calculations, displaying instead an appropriate error message. Two slider gadgets are provided in the control panel appearing just below the innovation canvas (left hand side of Figure 5) to allow the user to modify the value of the stitching parameter ξ associated with (10), as well as the initial TES variate U_0.

4.2 The Interactive Mode

A major advantage of the TEStool GUI lies in its ability to rapidly relay updated visual statistics to the user in response to modifications to the model parameters displayed in the innovation canvas. An initial set of calculations must first be performed on an empirical time series dataset (which appears in the upper left display canvas of Figure 3) by computing and displaying the empirical histogram (upper right canvas) and autocorrelation function (lower left canvas). Following construction of a valid innovation model (lower right canvas) as described in Section 5, another round of simulation and/or analytical computations must be performed in the 'On Command' mode. A user may then elect to assign any canvas displaying dataplots to the interactive mode by means of the 'Update Method' selector gadget. Subsequently, whenever changes are made to the stitching parameter or to the probability steps in the innovation canvas resulting in a legal model (i.e., $\sum P_k = 1.0$), TEStool immediately re-computes and re-displays any graphs associated with those canvases which have been assigned to the interactive mode.

The time required for TEStool to respond in interactive mode depends on the particular computations requested, which may be simulated or analytical. In practice, if the autocorrelations are computed analytically and the sample path simulations generate a relatively small number of points (on the order of 1000), the computations result in a response time on the order of one second on an unloaded SPARCstation2. Autocorrelation statistics generated by a Monte Carlo simulation would increase the response time accordingly, as would an increase in the number of sample points generated.

The ability to continually change a TES model and obtain prompt visual feedback on the results serves to facilitate the assessment of progress in the task of simultaneously approximating both an empirical autocorrelation function and a simulated sample path. Thus, the TEStool GUI serves as an effective aid in helping to select subsequent modifications of the innovation steps and stitching parameter. The heuristic search process for an appropriate TES model becomes intuitive, since the actions performed in the model window bear strong similarities to the knobs and joysticks used in interactive video arcade games. TEStool users can quickly acquire a qualitative and quantitative understanding of the relationships between the model parameters and its corresponding sample paths and statistics, without having prior grounding in the abstract underlying mathematics of TES processes. A resultant benefit is that the task of constructing a TES model may be readily delegated to a non-expert. On the other hand, an expert user experimenting with TEStool could possibly extract rules that would facilitate the heuristic search for an appropriate model. In this direction, future versions of TEStool may incorporate an automated search module designed to perform multiple offline iterations of the modeling process, in an attempt to identify a candidate set of TES models.

5 EXAMPLE: TES MODELING OF COMPRESSED VBR VIDEO

Figures 3–5 are reproduced from an actual workstation display screen to illustrate an example in which TEStool has been used to model a sequence of variable bitrate (VBR) encoded video frames. Figure 3 depicts the TEStool screen sub-divided into four graphical display canvases which represent different stages of the modeling process. The upper left canvas plots the empirical sample path of compressed frame bitrates (round markers); superimposed on it is a model sample path generated by a Monte Carlo simulation of a TES model (diamond markers). The innovation density incorporated in that TES model is represented graphically in the lower right hand modeling window. The corresponding histogram and autocorrelation function are plotted against their empirical counterparts in the upper right and lower left hand canvases, respectively.

Recall that the TES methodology always guarantees that for any innovation density, the marginal distribution of a TES model will asymptotically approach that of its empirical counterpart, as the sample size approaches infinity; indeed, the right canvas in Figure 5 shows excellent agreement with the corresponding histograms. This is one way in which the TES methodology offers a distinct advantage over similar models, such as autoregressive schemes, for which it may be impossible to match an arbitrary empirical marginal distribution (histogram).

The selection of the innovation density parameters represented in Figures 3 and 5 was accomplished by an iterative heuristic search process. The object of the search is to find an "acceptable fit" of both the TES autocorrelation function and simulated sample path to their respective empirical counterparts (as stated previously, the fit of the marginal distribution is guaranteed.) The process of finding an "acceptable fit" to multiple plots is greatly facilitated and made practical by the TEStool graphic display canvases which will be detailed in this section.

In the current example, parameters K, L_k, R_k and ξ were constantly modified by use of the workstation mouse. The sequence of modifications was guided only by a qualitative understanding of the behavior of TES processes. The search could proceed rapidly, since the model autocorrelation function was computed analytically. These computations rely upon (4), with the specializations of the Laplace transform of the histogram distortion (6) and the step function density (9), both of which were computed in Jagerman and Melamed (1992b). Since an average video scene in this example lasts for only a few seconds and the complete dataset consisted of 214 sample points, we only attempted to match the first 20 lags of the empirical autocorrelation function; the statistical reliability of the higher lags is doubtful, anyway.

The heuristic TES modeling process described above settled on the 2-step innovation density appearing in Figure 3, and again in Figure 5. In addition, the symmetric shape of the empirical sample path in Figure 4 suggested a stitching transformation S_ξ, with $\xi = 0.5$. The entire search process that resulted in identification of the model reported here was performed in less than an hour.

The criterion for identification of an "acceptable fit" of TES-generated sample paths in the current implementation of TEStool remains a subjective one, requiring a level of pattern matching skills that is more readily attained by human operators than by the most sophisticated computer algorithms. This follows from the recognition that the modeling goal is not to reproduce, point for point, the original empirical sample path, but to construct a TES process that will capture the underlying characteristics of the empirical sample, so that something akin to a Turing Test might be successfully applied: a significant percentage of human observers would not be able to distinguish the empirical sample path from that generated by a TES process simulation. We believe it is worth some effort to achieve this matching, since similarity of sample paths would increase the level of confidence in the model. We emphasize that such qualitative matching is done in addition to, not instead of, the quantitative matching of marginal distributions and approximation of autocorrelation functions.

6 CONCLUSIONS

This paper reported on a new modeling methodology, TES, and on a supporting visual modeling environment, TEStool. The TEStool GUI facilitates an heuristic search for TES models that simultaneously capture both the marginal distribution and autocorrelation function of empirical times series data.

Autocorrelated time series abound in practical applications. A case in point is bursty (autocorrelated) traffic in high-speed telecommunications networks. When offered to queuing systems, autocorrelated traffic can give rise to far worse performance measures as compared to those predicted by classical queuing models (which ignore autocorrelation). Consequently, there exists a strong need for modeling utilities that can accurately characterize autocorrelated sample data (i.e., field measurements) and generate replicate stationary dependent time series. Such models may then used as input drivers (source generators) for Monte Carlo simulations.

The TES modeling methodology appears to provide more realistic models of actual dependent stochastic processes than do other methodologies, since it attempts to fit both marginals and autocorrelations. Broadly speaking, these other methodologies usually employ one of the following two strategies: The first stategy constructs a stochastic process with a known autocovariance structure (e.g., a first-order autoregressive scheme). However, with this approach the desired target marginals are difficult to obtain. The second strategy is the reverse of the first. It constructs a random variate having the prescribed marginal, and generates additional variates by employing a selected distribution-preserving transformation. However, this transformation is specific to each prescribed marginal, and cannot always be devised. (See Jagerman and Melamed 1992a for representative references). In contrast, the TES methodology often possesses sufficient leeway to fit both marginals and autocorrelation structure simultaneously.

Of course, TES methods do not provide an acceptable fit for every conceivable stochastic process. Fortunately, in many of these cases it is possible to perform additional transformations that render the data more amenable to TES modeling. A case in point is described in Melamed et al. (1992). Here the empirical data exhibited periodic aspects which could not be modeled by TES satisfactorily. However, removal of the periodic components revealed by periodogram analysis yielded residuals data that could be modeled by TES; the resulting composite model was in excellent agreement with the empirical data.

The TEStool visual environment was found to be indispensable to efficacious usage of the TES modeling methodology. Our experience shows that it increases both the speed and quality of the TES modeling process. TEStool transforms the potentially tedious search process required for constructing a TES model into something akin to a video arcade game, making it possible to rapidly investigate the behavior of TES autocorrelations as a function of the innovation and distortion parameters. By providing immediate visual feedback on the results of incremental changes to a TES model, a TEStool user is able to quickly scan through a large volume of candidate TES processes and often converge to an appropriate model. Some specific experiences resulting from the use TEStool have been described in this paper. Extensive additional experience with TEStool has shown that TES processes exhibit a rich behavioral "dynamic range," and that certain innovations can result in very complex autocorrelation functions and quite intricate sample paths.

REFERENCES

Bratley, P., B.L. Fox, and L.E. Schrage. 1987. *A guide to simulation*. 2nd edition. New York: Springer-Verlag.

Fendick, K.W., V.R. Saksena, and W. Whitt. 1989. Dependence in packet queues. *IEEE Transactions on Communications* 37:1173–1183.

Geist D., and B. Melamed. 1992. TEStool: An environment for visual interactive modeling of autocorrelated traffic. *Proceedings of ICC '92*, 1285–1289, Chicago, Illinois.

Heffes, H., and D.M. Lucantoni. 1986. A Markov modulated characterization of packetized voice and data traffic and related statistical multiplexer performance. *IEEE Journal on Selected Areas in Communications* SAC-4:856–868.

Jagerman, D., and B. Melamed. 1992a. The transition and autocorrelation structure of TES processes, part I: general theory. To appear in *Stochastic Models*.

Jagerman, D., and B. Melamed. 1992b. The transition and autocorrelation structure of TES processes, part II: special cases. To appear in *Stochastic Models*.

Jagerman, D., and B. Melamed. 1992c. Spectral analysis of basic TES processes. Preprint, NEC USA, Inc., Princeton, New Jersey.

Lee, D.-S., B. Melamed, A. Reibman, and B. Sengupta. 1991. Analysis of a video multiplexor using TES as a modeling methodology. *Proceedings of GLOBCOM '91*, 16–20, Phoenix, Arizona.

Lewis, P.A W., and E. McKenzie. 1991. Minification processes and their transformations. *Journal of Applied Probability* 28:45–57.

Livny, M., B. Melamed, and A.K. Tsiolis. 1993. The impact of autocorrelation on queuing systems. To appear in *Management Science*.

Melamed, B. 1991. TES: a class of methods for generating autocorrelated uniform variates. *ORSA Journal on Computing* 3:317–329.

Melamed, B., D. Raychaudhuri, B. Sengupta, and J. Zdepski. 1992. TES-based traffic modeling for performance evaluation of integrated networks. *Proceedings of INFOCOM '92*, 75–84, Florence, Italy.

Patuwo, B.E., R.L. Disney, and D.C. McNickle. 1993. The effects of correlated arrivals on queues. To appear in *IIE Transactions*.

Schmeiser, B.W. 1990. Simulation experiments. In *Handbook of Operations Research and Management Science: Vol. 2*, (ed. D. Heyman and M. Sobel), New York: North Holland.

AUTHOR BIOGRAPHIES

BENJAMIN MELAMED is Head of the Performance Analysis Department at the C&C Research Laboratories, NEC USA, Inc., in Princeton, NJ. His research interests include systems modeling and analysis, simulation, stochastic processes, and visual modeling environments. Melamed received his Ph.D. in Computer Science in 1976 from the University of Michigan. Before coming to NEC, he taught in the Department of Industrial Engineering and Management Science at Northwestern University. He later joined the Performance Analysis Department at Bell Laboratories, where he became an AT&T Bell Laboratories Fellow in 1988. He is a Senior Member of IEEE.

JON HILL is a member of the Performance Analysis Department at the C&C Research Laboratories, NEC USA, Inc., in Princeton, NJ. His interests include simulation output analysis, graphical interfaces to modeling tools, statistics, data analysis and graphical techniques for displaying data. After receiving a Ph.D. in Biophysics from the University of Utah in 1980, he joined AT&T Bell Laboratories at Holmdel, NJ, and participated in both the Microsystems Design Laboratory and the Operations Research Department there, until joining NEC in 1992.

DAVID GOLDSMAN is an Associate Professor in the School of ISyE at Georgia Tech. His research interests include simulation output analysis and ranking and selection. He is Vice President of the TIMS College on Simulation.