TEStool : A Visual Interactive Environment for Modeling Autocorrelated Time Series

Jon R. Hill *
Benjamin Melamed †

NEC USA, Inc.
4 Independence Way
Princeton, NJ 08540

March 8, 1995

Abstract

TEStool is a visual interactive software environment for modeling autocorrelated time series, using a versatile class of stochastic processes called TES (*Transform-Expand-Sample*). The novel feature of the TES modeling approach is that it strives to fit a model to empirical records by simultaneously capturing both the empirical distribution and the leading empirical autocorrelations. Thus, TES models can have a high degree of fidelity, since both first-order and second-order aspects of the statistical signature of empirical time series are targeted for capture. TEStool has been used extensively to model empirical data from a variety of application domains, including compressed video traffic in high-speed communications networks, financial time series and machine reliability.

This paper explains the TES approach to modeling empirical autocorrelated time series adopted in TEStool, as well as TEStool's graphical user interface (GUI). Special emphasis is placed on features that increase the efficiency and quality of modeling based on visualization and real-time interaction. In particular, TEStool casts a heuristic search activity into an intuitive process which is guided by visual feedback. Consequently, the software makes TES modeling accessible to modelers who have a basic understanding of time series, but lack expert knowledge of TES theory.

Keywords and Phrases: autocorrelated processes, graphical user interface, input analysis, simulation, software environment, source modeling, stochastic processes, temporal dependence, TES processes, TES sequences

*Current address: Smith Barney, Inc., 1345 Avenue of the Americas, New York, NY 10105.
†Current address: RUTCOR, Rutgers University, P.O. Box 5062, New Brunswick, NJ 08903

1 Introduction

Monte Carlo simulation-based modeling is a fundamental tool in widespread use for assessing performance measures in stochastic systems [2, 13]. Because simulation is computationally intensive, it has been implemented on digital computers since their earliest days. An essential feature of simulation is the need to generate and manipulate random sequences, such as arrival and service processes in queueing systems.

The suite of currently available simulation languages provides facilities to sample independent variates only, via pseudo-random number generators. However, if sequences with temporal dependencies are required, they must be provided by the modeler. Such sequences are frequently encountered in computer and communications applications. Locality of reference in caches and data base accesses are typical computer examples, while bursty traffic and locality of switching typify communications examples. An important case in point is traffic in high-speed communication networks and the queueing behavior it engenders — a particularly rich domain of simulation applications [22]. Standard queueing models assume that customer interarrival times and service times are renewal processes; in queueing terminology, they are denoted by GI (General Independent). Various studies have shown that standard models predict overly optimistic performance measures, such as queue lengths and waiting times, resulting in errors of considerable magnitude [5, 7, 17, 27]. These studies convincingly demonstrate the modeling risk inherent in many models which ignore temporal dependencies for the sake of conceptual and mathematical simplification. Both simplicity and complexity must always be carefully balanced.

The growing realization of the importance of dependence in time series provides the motivation for devising computer methods that are capable of generating temporally-dependent time series with a wide variety of marginal distributions and autocorrelation functions. In order to obtain high-fidelity models of empirical records, we stipulate three requirements for any prospective model:

(1) the model's marginal distribution should match its empirical counterpart (histogram),

(2) the model's leading autocorrelations should approximate their empirical counterparts,

(3) sample paths generated by a Monte Carlo simulation of the model should "resemble" the empirical record.

We remark that the term "leading autocorrelations" refers to those with lags in some range of the form $1 \leq n \leq n^*$, which can be reliably estimated from sufficient data (for small samples, on the order of hundreds, one often uses the square root of the sample size as the cut-off lag, n^* [4]).

Most popular standard fitting methods are geared to satisfy either Requirement (1) alone (e.g., the inversion method [2, 13]) or Requirement (2) alone (e.g., autoregressive/moving average schemes [3]). Only a few fitting methods aiming to satisfy both Requirement (1) *and* (2), simultaneously, have been proposed in the literature. Schmeiser and co-workers studied correlation-induction methods based on Gaussian variates [28]. Lewis and co-workers proposed a class of generation methods called Minification/Maxification which generate marginally uniform variates, and those can be further transformed to ones with arbitrary marginal distributions [16].

The subject of this paper is a versatile class of generation methods of autocorrelated sequences, called TES (*Transform-Expand-Sample*) [19, 8, 9, 10], motivated by the three modeling requirements stipulated above. TES processes are versatile in the sense that they give rise to qualitatively diverse sample paths (including cyclical and non-directional) and to an assortment of autocorrelation functions with a variety of functional forms (monotone, oscillating, alternating and others). Furthermore, the TES modeling methodology (to be described later) guarantees an exact match to any empirical distribution, and simultaneously offers a great deal of flexibility in approximating the leading empirical autocorrelations. Thus, the TES modeling approach expressly attempts to conform to the three aforementioned modeling requirements. In addition, TES enjoys favorable computational properties. First, generation algorithms of TES sequences have a low computational complexity in both time and space. Second, the autocorrelation function of a TES model can be computed from fast and accurate analytical formulas without requiring simulation. This fact makes it feasible to adopt an interactive style in computer-assisted TES modeling, since the analytical 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 is greatly facilitated by casting it into a visual interactive style.

This paper describes both the TES modeling methodology and a visual interactive software environment, called TEStool, which implements this methodology. TEStool has been designed to support generic TES modeling, but places particular emphasis upon the fitting of TES models to empirical records. It makes extensive use of visualization to create an environment that relies on visual feedback and real-time interaction to support efficient heuristic searches for high-fidelity TES models. A visual modeling environment facilitates the search process and reduces both modeling errors and the tedium of repetition. TEStool effectively transforms the search process into an intuitive activity that does not require a deep understanding of the underlying mathematics. Consequently, TES modeling is made accessible to TES experts and non-experts alike.

The rest of this paper is organized as follows. Section 2 presents a brief overview of TES processes and Section 3 describes the particular implementation of TES theory incorporated into TEStool. Section 4 describes the TEStool graphical user interface (GUI) with special emphasis on features that increase the efficiency and quality of modeling based on visualization and real-time interaction. Section 5 describes actual experiences with TEStool modeling, while Section 6 concludes with a summary.

2 Overview of TES Processes

TES is a class of autocorrelated random sequences with arbitrary marginal distributions and a broad range of autocorrelation functions. Generation methods of TES sequences are essentially autoregressive schemes with modulo-1 reduction, followed by additional transformations. A detailed presentation of TES process theory may be found in [19, 8, 9, 10]. An extensive overview of TES processes and the corresponding modeling methodology appears in [20], and a more condensed version in [21]. This section contains a brief review.

For any real x, let $\lfloor x \rfloor = \max\{n \text{ integer} : \ n \leq x\}$ and $\langle x \rangle = x - \lfloor x \rfloor$ be the integral part and fractional part of x, respectively. Let U_0 be uniform on [0,1), and let $\{V_n\}_{n=1}^{\infty}$ be a sequence of iid (independent identically distributed) random variables with a common density f_V, independent of U_0. The random variables V_n are referred to as *innovations*, since they introduce the "new" randomness at each time step. The construction of general TES sequences employs two sequences in lockstep; these are called *background* and *fore-*

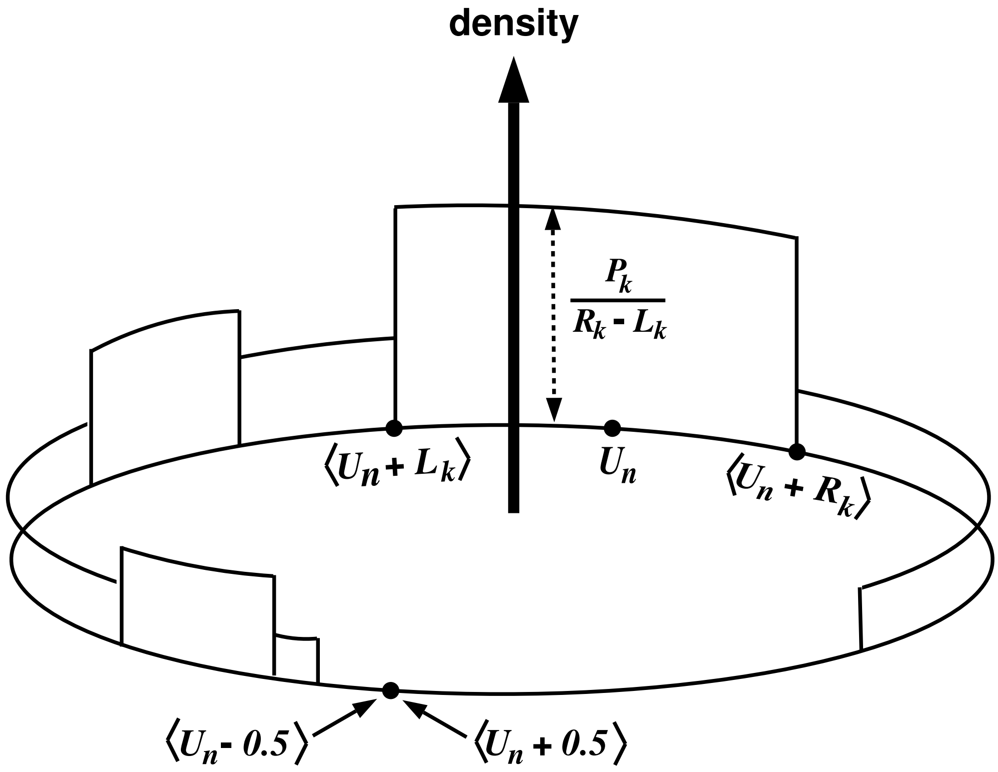

Figure 1: Geometric Interpretation of Background TES Sequences With Step-Function Innovation Densities

ground sequences, respectively. TES sequences fall into two classes: TES^+ and TES^-. The superscripts are suggestive of the fact that background TES sequences achieve coverage of the full range of feasible lag-1 autocorrelation [19]; TES^+ sequences cover the positive range $[0, 1]$, while TES^- sequences cover the negative range $[-1, 0]$. Throughout this paper, we routinely append such superscripts to various objects associated with these classes, but omit them if the distinction is immaterial. In particular, TES background sequences are denoted $\{U_n^+\}$ and $\{U_n^-\}$, and have the form

$$U_n^+ = \begin{cases} U_0, & n = 0 \\ \langle U_{n-1}^+ + V_n \rangle, & n > 0 \end{cases}, \qquad U_n^- = \begin{cases} U_n^+, & n \text{ even} \\ 1 - U_n^+, & n \text{ odd} \end{cases} \tag{2.1}$$

It can be shown that the sequences defined by Eq. (2.1) are stationary Markovian, and their marginal distribution is always uniform on [0,1), *regardless* of the choice of an innovation density f_V [8]. In fact, the choice of f_V provides a broad range of freedom in selecting a dependence structure for a TES sequence.

Figure 1 depicts a 3-dimensional geometric interpretation of the construction of background TES sequences with step-function innovation densities. Here the unit circle (circumference 1) lies at the bottom, in the 2-dimensional plane, with the current TES variate, U_n, in the north/north-easterly sector. A step-function innovation density, f_V, has been erected over the unit circle, with values in the third dimension (the "up" direction). The origin for the support of f_V is set at the current TES variate, U_n, implying that the support of f_V is the interval $[-0.5, 0.5)$; step k of f_V is specified by a triple, (L_k, R_k, P_k), and corresponds to a uniform selection of a value in the interval $[\langle U_n + L_k \rangle, \langle U_n + R_k \rangle)$ with probability P_k

(i.e., the height of step k is $\dfrac{P_k}{R_k - L_k}$). Intuitively, the modulo-1 arithmetic used in defining TES processes in (2.1) has a simple geometric interpretation as a Markovian random walk on the unit circle, with random step size V_n; for example, the next TES$^+$ variate, U_{n+1}, is obtained by sampling a value on the unit circle from the innovation density f_V.

While background TES sequences are auxiliary in nature, the real modeling interest normally lies in foreground TES sequences, $\{X_n^+\}$ and $\{X_n^-\}$, of the form

$$X_n^+ \;=\; D(U_n^+), \qquad X_n^- \;=\; D(U_n^-). \tag{2.2}$$

obtained from Eq. (2.1) via some transformation D from $[0,1)$ to the reals, called a *distortion*. Since background TES sequences are always marginally uniform on $[0,1)$, the generality of D allows us to generate random sequences with essentially arbitrary marginal distributions, by invoking the inversion method: if U is uniform on $[0,1)$ and F is any prescribed distribution function, then the random variable $F^{-1}(U)$ has distribution F; see, e.g., [2, 13] for details. In particular, this allows us to match any marginal distribution (histogram) of empirical data.

Our goal is to construct a model whose autocorrelation function approximates that of the empirical data. Naturally, a Monte Carlo simulation of the random sequences of Eq. (2.1) can provide a reasonable statistical estimate of the model's autocorrelation function, provided that the estimate is calculated from a sufficiently large sample. This approach, however, can be costly in terms of time complexity, especially when high autocorrelations necessitate large sample sizes to ensure adequate statistical reliability. It is, therefore, essential to develop fast and efficient computational techniques for these calculations. Reference [8] provides analytical formulas for the autocorrelation functions $\rho_X^+(\tau)$ and $\rho_X^-(\tau)$ of $\{X_n^+\}$ and $\{X_n^-\}$, respectively, for any lag $\tau > 0$,

$$\rho_X^+(\tau) \;=\; \frac{2}{\sigma_X^2} \sum_{\nu=1}^{\infty} \Re[\tilde{f}_V^{\tau}(i2\pi\nu)]\, |\tilde{D}(i2\pi\nu)|^2 \tag{2.3}$$

$$\rho_X^-(\tau) \;=\; \begin{cases} \rho_X^+(\tau), & \tau \text{ even} \\[2ex] \dfrac{2}{\sigma_X^2} \displaystyle\sum_{\nu=1}^{\infty} \Re[\tilde{f}_V^{\tau}(i2\pi\nu)]\, \Re[\tilde{D}^2(i2\pi\nu)], & \tau \text{ odd} \end{cases} \tag{2.4}$$

where σ_X^2 is the common variance of $\{X_n^+\}$ and $\{X_n^-\}$, the tilde symbol denotes the Laplace Transform, f_V^{τ} is the τ-fold convolution of the innovation density f_V, the $\Re$ symbol denotes the real part operator, and $i = \sqrt{-1}$. The series in Eqs. (2.3) and (2.4) converge rapidly, and are suitable for numerical calculation. Formulas for the quantities $\tilde{f}_V(i2\pi\nu)$, required in Eqs. (2.3) and (2.4), are given in [9].

3 The TEStool Modeling Methodology

In order to facilitate the modeling of empirical records, the TES modeling methodology, as implemented by TEStool, adopts several specializations of the TES theory reviewed in Section 2.

First, TEStool restricts innovation densities to the class of step-function densities, since this class is able to approximate any other density function arbitrarily closely. The conceptual and operational simplicity that ensues is gained with virtually no loss of generality.

Notice that a step-function density corresponds to a probabilistic mixture of uniform densities, that is,

$$f_V(x) \;=\; \sum_{k=1}^{K} 1_{[L_k,R_k)}(x)\,\frac{P_k}{R_k - L_k}, \quad -0.5 \le x < 0.5, \tag{3.1}$$

where 1_A is the indicator function of set A, K is the number of steps, $[L_k, R_k)$ is the support of step k with $-0.5 \le L_k < R_k < 0.5$, and $0 < P_k \le 1$ is the mixing probability of step k, subject to $\sum_{k=1}^{K} P_k = 1$; for convenience, it is also required that steps do not overlap (i.e., $R_k \le L_{k+1}$, $1 \le k \le K-1$). The support $[-0.5, 0.5)$ is mathematically equivalent to $[0, 1)$ due to the modulo-1 reduction in Eq. (2.1), but is more intuitive visually (recall the discussion of Figure 1 in Section 2). An equivalent parameterization, (α_k, ϕ_k), may be used instead of (L_k, R_k), where

$$\alpha_k = R_k - L_k, \quad \phi_k = \frac{R_k + L_k}{R_k - L_k}, \quad 0 < \alpha_k \le 1,\; 0 \le |\phi_k| \le \frac{1}{\alpha_k} - 1. \tag{3.2}$$

For step k, α_k is its support length, and ϕ_k measures the support's distance from the origin, somewhat analogously to polar coordinates; see [8, 20] for motivation and further details.

Second, TEStool utilizes a family of composite distortions of the form

$$D_{\xi,\hat{H}}(x) \;=\; \hat{H}^{-1}(S_\xi(x)), \quad x \in [0, 1). \tag{3.3}$$

Here, the inner transformation, S_ξ, is a "smoothing" operation, called a *stitching transformation*, parameterized by $0 \le \xi \le 1$, and given by

$$S_\xi(y) \;=\; \begin{cases} y/\xi, & 0 \le y \le \xi \\ (1-y)/(1-\xi), & \xi \le y < 1. \end{cases} \tag{3.4}$$

The outer transformation, $\hat{H}^{-1}$, is the inverse of the empirical (histogram) distribution function, computed from an empirical record $\{Y_n\}_{n=0}^{N}$ as

$$\hat{H}^{-1}(x) \;=\; \sum_{j=1}^{J} 1_{[\hat{C}_{j-1},\hat{C}_j)}(x)\,[l_j + (x - \hat{C}_{j-1})\,\frac{w_j}{\hat{p}_j}], \quad 0 \le x \le 1, \tag{3.5}$$

where J is the number of histogram cells, $[l_j, r_j)$ is the support of cell j with width $w_j = r_j - l_j > 0$, $\hat{p}_j$ is the probability estimator of cell j and $\{\hat{C}_i\}_{i=0}^{J}$ is the cumulative distribution function of $\{\hat{p}_j\}_{j=1}^{J}$, i.e., $\hat{C}_j = \sum_{i=1}^{j} \hat{p}_i$, $1 \le j \le J$, so that $\hat{C}_0 = 0$ and $\hat{C}_J = 1$. We mention that the hat symbol in Eq. (3.5) is appended to indicate that the corresponding variables are statistical estimates. Formulas for the quantities $\tilde{D}_{\xi,\hat{H}}(i2\pi\nu)$, required in Eqs. (2.3) and (2.4), are given in [9].

TES sequences of the form (2.1) give rise to sample paths that exhibit characteristic visual "discontinuities" whenever the corresponding random walk, $\{U_n\}$, crosses point 0 in either direction. The merit of the family of composite distortions (3.3) stems from the fact that for $0 < \xi < 1$, the stitched sequence $\{S_\xi(U_n)\}$ is far "smoother" (more "continuous-looking") than the sequence $\{U_n\}$ (see, e.g., Figures 6 and 17 in [20]), while simultaneously preserving the uniformity of $\{U_n\}$ (see Lemma 2 in [19]). It follows that any foreground TES sequence of the form $\{X_n\} = \{\hat{H}^{-1}(S_\xi(U_n))\}$, obtained from any background sequence $\{U_n\}$, is always guaranteed to have the empirical (histogram) distribution $\hat{H}$, *regardless* of the stitching parameter ξ and innovation density f_V selected. The choice of a pair

(ξ, f_V) determines the dependence structure of the foreground sequence, and in particular, its autocorrelation function. Thus, TES modeling decouples the fitting of the empirical distribution from the fitting of the empirical autocorrelation function. Since the former is guaranteed, the modeler need only concentrate on the latter.

An interactive modeling session would typically begin with an empirical data record $\{Y_n\}_{n=0}^N$ representing a partial process history. The activities involved in finding a complete specification of a TES model that captures first-order and second-order statistics of the empirical data are outlined below.

Selecting a TES Sign: The sign of the TES class (TES^+ or TES^-) is selected. The selection of a TES sign is based on the modeler's experience and knowledge of TES processes. Experience shows that TES^+ models are most prevalent. TES^- models are more appropriate for empirical sample paths and autocorrelation functions that have a sawtooth (zigzag) appearance.

Constructing the Inverse Distribution Function: In principle, the inverse of every distribution is available, due to its monotonicity, and TEStool has a set of analytical inverse distributions (e.g., geometric, exponential etc.) that may be selected from a menu. In particular, when modeling empirical records, the inverse distribution function of the empirical data is the histogram inverse $\hat{H}^{-1}$, given in Eq. (3.5). TEStool permits its determination by modeler-provided histogram parameters (number of cells and cell width) and by the empirical record, $\{Y_n\}_{n=0}^N$.

Selecting an Initial Stitching Parameter: A initial stitching parameter is selected in the range $0 \leq \xi \leq 1$. The choice of ξ (in conjunction with $\hat{H}^{-1}$) determines a composite distortion of the form (3.3)).

Selecting an Initial Innovation Density: An initial step-function innovation density, f_V, is selected. To this end, it suffices to consider only those with support in $[-0.5, 0.5)$, since the modulo-1 operator may be applied to the innovation variate without affecting Eq. (2.1). This determines a marginally uniform TES process, $\{U_n\}$, which can be either $\{U_n^+\}$ or $\{U_n^-\}$.

Searching for a Stitching Parameter and Innovation Density: Once the selections above have been made, a TES process, $\{X_n\}$, of the form (3.3) is completely determined, and its autocorrelation function can be computed from Eq. (2.3) or (2.4). In addition, a simulated sample path of $\{X_n\}$ is generated with the initial simulated variate typically set to the corresponding initial empirical observation. If the leading autocorrelations of ρ_X approximate well their empirical counterparts, and the simulated sample path "resembles" qualitatively its empirical counterpart, then the search is terminated; otherwise, the search for a suitable stitching parameter and step-function innovation density is continued. This heuristic search is the core activity of TES modeling. the modeler attempts various combinations of (ξ, f_V) and observes the outcome through visual feedback; the TEStool GUI allows the modeler to superimpose the graphs of TES statistics on their empirical counterparts (see Figure 2). The search proceeds in an interactive style: each model modification triggers a recalculation and redisplay of the corresponding statistics. The modeler may loop back to any step based on the quantitative fit of the current model's autocorrelation function or spectral density (recall that the histograms are guaranteed to match) and the

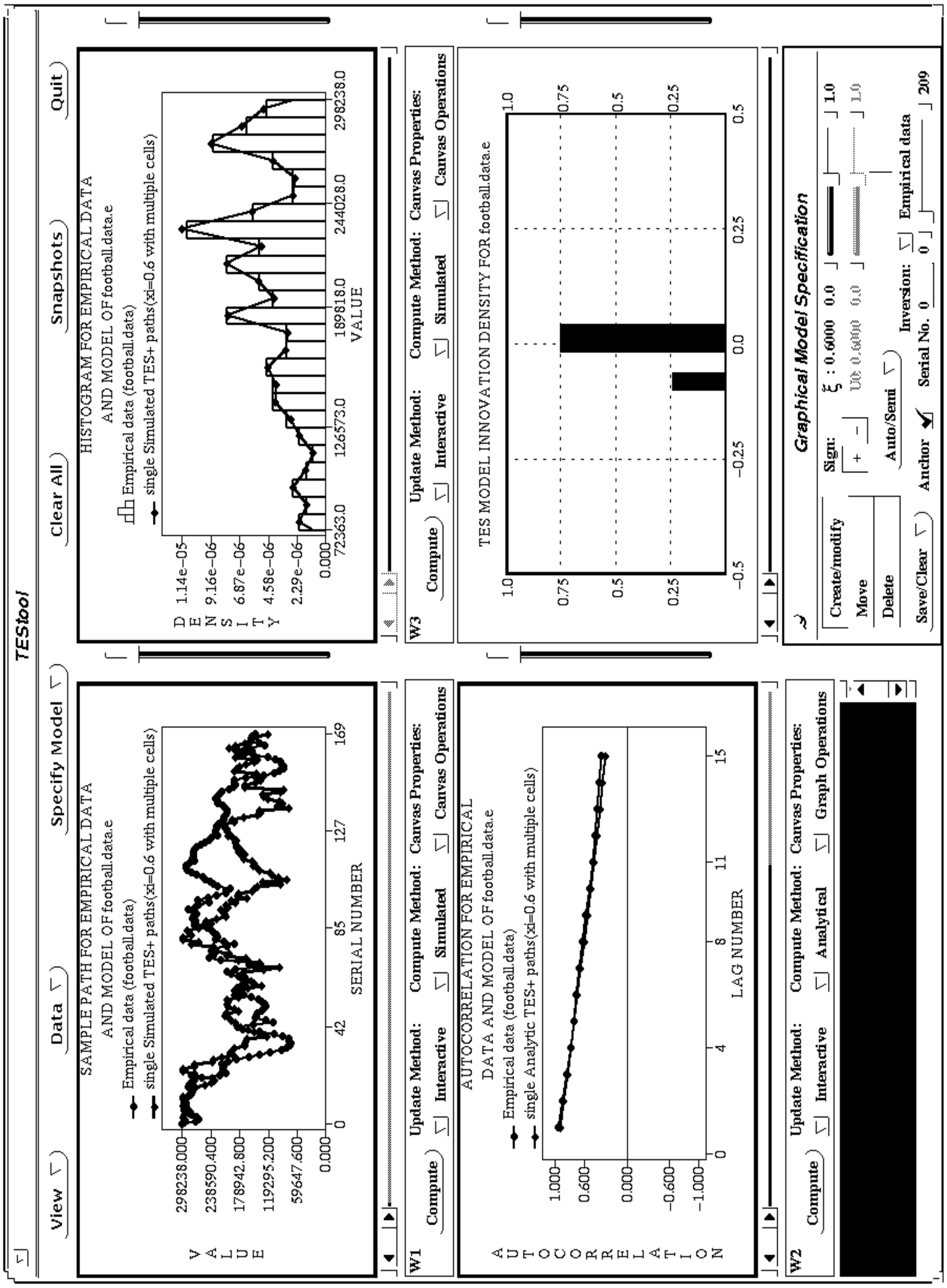

Figure 2: A TEStool display (bit rate process of compressed video).

qualitative fit of Monte Carlo sample paths to their respective empirical counterparts. While the search is heuristic (trial-and-error), it is not a blind search. A great deal is known about the qualitative behavior of sample paths and autocorrelation functions generated by TES processes, as a function of the stitching parameter and innovation density parameters. These are discussed in some detail in references [19, 8, 20]. Additional discussion of interactive TES modeling may be found in Section 4.5.

Figure 2 depicts a TEStool display after a successful interactive heuristic search has been completed. to the satisfaction of the modeler. The lower-right canvas displays the model specification (step-function innovation density, stitching parameter, etc.), while the other three canvases superimpose model statistics (diamonds) on their empirical counterparts (bullets). The empirical data modeled is a field measurement of a sequence of bit rates (the frame sizes of compressed video pictures), obtained from a video clip of a football scene. Observe that (i) the empirical histogram (upper-right canvas) is indeed matched by its model counterpart, as guaranteed by TES theory; (ii) the leading empirical autocorrelations are well-approximated by their model counterparts (there is an increasing, though slight, visible discrepancy in the vicinity of lags 12–15); and (iii) the empirical and model-generated sample paths appear remarkably "similar" in appearance. Thus, the model in Figure 2 complies with the three modeling requirements of Section 1, and as such it constitutes an adequate model from the standpoint of the TES modeling methodology.

4 The TEStool GUI

The TEStool visual interactive software environment provides utilities for constructing TES models and examining their statistics, or predictions, resulting from both heuristic searches and algorithmic ones (see discussion at the end of Section 6). It is written in C++ and runs on UNIX-based workstations with X-windows displays, though contemporary PC's would be adequate platforms.

This section describes the TEStool GUI in terms of the active objects that mediate interaction, such as menus, selection icons, buttons, sliders and graphics canvases. Special emphasis is placed on visual interactive aspects that facilitate effective modeling interactions.

4.1 The Window System

Figure 2 displays a TEStool screen (converted to black and white) as it might appear during a typical modeling session. Located within the top border of the parent window is a row of buttons that create menus and popup forms that mediate global operations which are not specific to any single canvas. The upper portion of the parent window is subdivided into four tiled (non-overlapping) canvas areas designed to display various types of color graphics; in the bottom portion of the parent window appear two smaller text display panels that provide access to the host operating system.

Two pulldown menus, shown in the lower-left corner of Figure 3, allow the modeler to individually tailor the graphical display for each canvas and to perform various graphical operations; see also the two subwindows in the upper part of Figure 3. They support the modification of display attributes (e.g., marker and connector style, width and color, zoom factor, etc.), as well as graph operations (e.g., cut and paste) and examination of their numerical values.

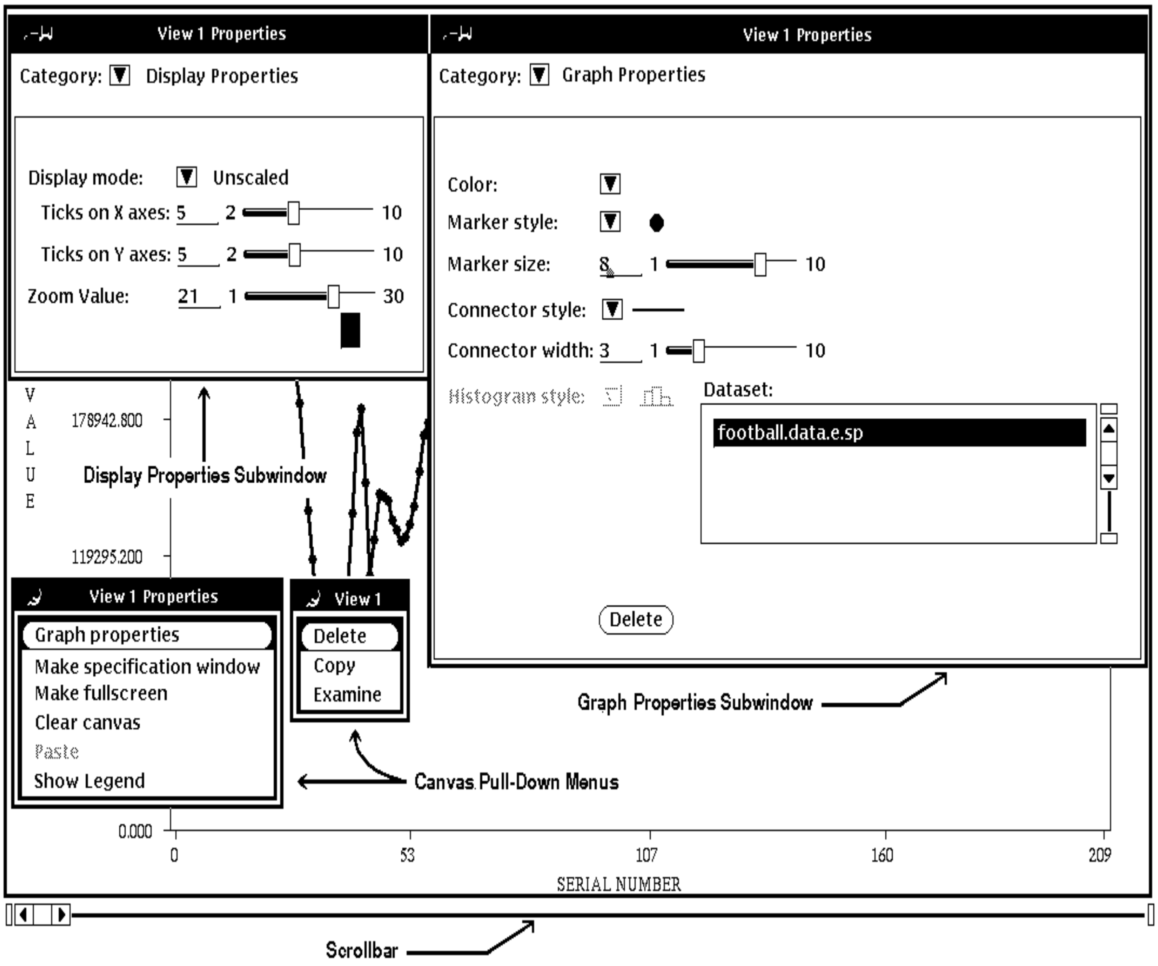

Figure 3: The canvas hidden menus and display popup form.

4.2 Operations on Datasets

TEStool statistical datasets are classified into three types:

Empirical: A dataset whose origin is external to TEStool. An empirical dataset would typically constitute a record of an empirical time series obtained by measurements from a "real-life" system, or statistical estimates calculated from it (histogram, autocorrelation function or spectral density).

Simulated: A dataset generated from a Monte Carlo simulation of a TES model specification. The dataset is comprised of either points of simulated sample paths, or of statistical estimates, calculated from them (histogram, autocorrelation function or spectral density).

Analytical: A dataset created by computing the autocorrelation functions (2.3) and (2.4), the corresponding spectral densities, or predicted TES values. These are computed numerically (not by simulation) using fast and accurate analytical formulas developed in [8, 9, 10, 11].

Note that while autocorrelation functions and spectral densities can be calculated for any dataset type (though by different methods), sample paths and histograms can only be associated with empirical and simulated datasets, though predicted values may be viewed as an "analytical sample path". On the other hand, different types of datasets may share a logical association within the TEStool environment. For example, an empirical dataset may be associated with a simulated or analytical dataset, since the empirical histogram may have been selected as a component of the composite distortion (3.3).

Dataset operations (e.g., read, save, examine) are supported by buttons within the upper border shown in Figure 2. TEStool allows the modeler to save the state of a TEStool session to a diskfile and to restore the state on request (`Snapshots` button).

4.3 Textual and Graphical Specifications of TES Models

Selection of the `Specify Model` button at the display top of Figure 2 results in the appearance of a pull-down menu that provides a choice between two methods, textual (see Figure 4) or graphical (see Figure 5), for specifying a TES model.

The default selection is textual. It creates a text form (Figure 4) displaying alphanumeric fields, a scrolling panel, and a selection gadget that allow the modeler to completely define a TES model by supplying the following information:

1. A TES model class sign (TES^+ or TES^-).
2. An initial value, U_0, on the unit circle for the background TES process.
3. A value for the stitching parameter, ξ.
4. An inversion transformation to a prescribed marginal distribution. The selection gadget at the bottom of the form creates a pull-down menu that allows the modeler to choose between a uniform, exponential, geometric or discrete distribution, as well as an empirical histogram distribution based on an empirical record. Recall that a histogram distribution is a function $\hat{H}$, whose inverse has the form of Eq. (3.5) in terms of a triplet set, $\{(l_j, r_j, \hat{p}_j)\}_{j=1}^{J}$.

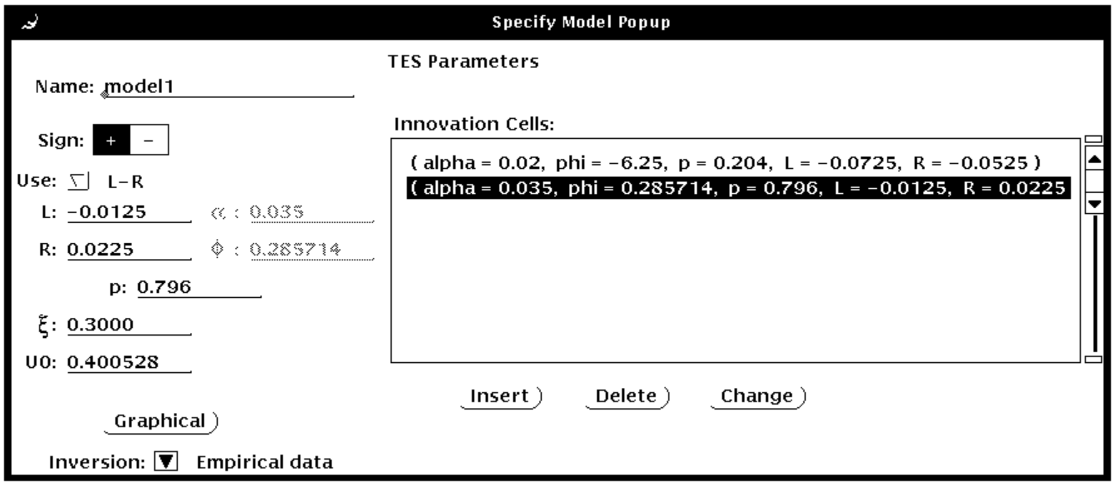

Figure 4: The textual model specification form.

5. An innovation step-function density of the form (3.1), whose steps are specified by a triplet set, $\{(L_k, R_k, P_k)\}_{k=1}^{K}$, or equivalently, $\{(\alpha_k, \phi_k, P_k)\}_{k=1}^{K}$ (see Section 3). Innovation steps appear on separate rows in a scrolling list whose items can be inserted, deleted or changed via selection of appropriate buttons (see Figure 4).

The graphical version of the model specification subwindow (see Figure 5) is intended to render user interaction during model specification as visually intuitive as possible. The graphical modeling canvas provides an interactive drawing area as well as a set of sliders and selection gadgets to enable the modeler to provide the same information as in the textual specification by using only the workstation mouse.

The steps of an innovation density are specified visually by constructing a graphical representation of the imaginary cylinder in Figure 1 as it would appear if it were unwrapped and mapped onto a two-dimensional plane. Three selection gadgets on the left side of the control panel allow the modeler to choose the action of the mouse within the canvas:

1. `Create/modify`: the mouse cursor is used to sweep out steps, visualized as rectangles.

2. `Move`: the mouse cursor is used to move existing steps horizontally, but only to positions that are non-overlapping with other steps.

3. `Delete`: a step pointed at by the mouse cursor is deleted.

Innovation density steps are created inside the customized modeling canvas by sweeping out one or more non-overlapping rectangles (representing steps of the innovation density, f_V) with the workstation mouse within a planar gridded canvas whose x values are between -0.5 and 0.5 and whose y values are between 0 and 1 (Figure 5). As an aid in constructing the steps graphically, the mouse cursor icon automatically changes to a crosshair icon whenever the cursor appears inside the innovation density canvas area. As an additional construction aid, a pair of text display fields are continuously updated with the x and y position values of the cursor within the innovation canvas. Each step represents an (L_k, R_k, P_k) triplet in the following manner: The base of a step k always rests on the (horizontal) x axis. The left and right base edges correspond to L_k and R_k, respectively, while the height is P_k. Since the plane actually represents an unwrapped cylinder, stretching a step to the right of $x = 0.5$

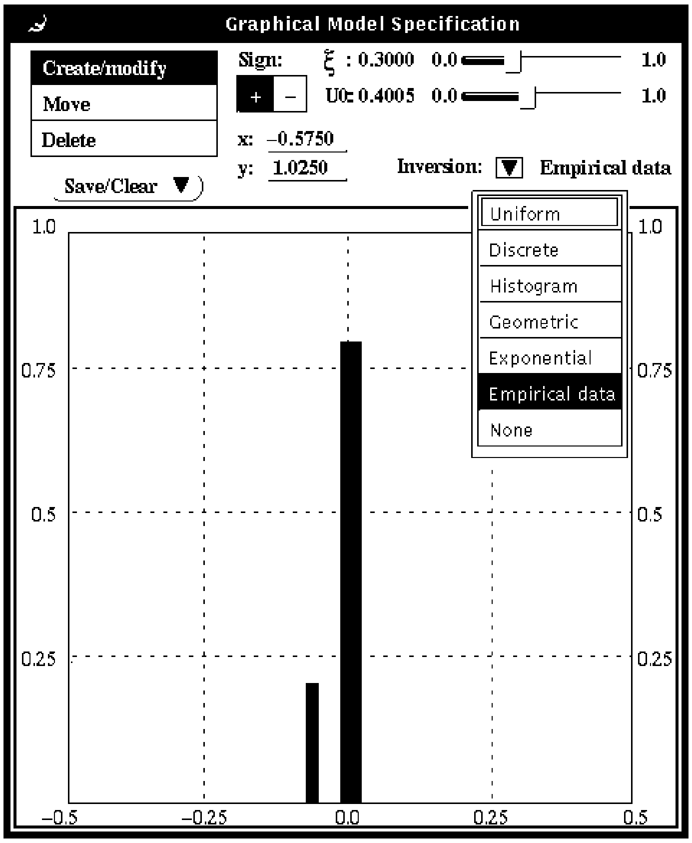

Figure 5: The graphical model specification subwindow.

will cause the cursor to wrap around to the right of $x = -0.5$, and to continue to stretch there; similarly, stretching to the left of $x = -0.5$ will cause it to wrap around to the left of $x = 0.5$.

Note that TEStool's graphical representation of a step-function density is not standard, since the total area under the steps is not constrained to sum to one. However, this method of representation is more useful for the current application, since it has the advantage of being distinctly easier to manipulate graphically. The graphical advantage derives from the fact that the sum of the heights of the steps is bounded by one, whereas the height of a density function may be unbounded.

The TEStool graphical specification canvas, depicted in Figure 5, allows the modeler to add and modify as many steps as desired, but rigidly enforces the constraint $\sum P_k \leq 1.0$. TEStool continuously monitors the value of this summation and truncates any step at the height which exactly satisfies $\sum P_K = 1.0$. As an additional visual aid, the border of the innovation density canvas changes from red to green if $\sum P_k$ is determined to be equal to one. Finally, if $\sum P_K < 1.0$, the border remains red, and the TEStool computation modules will refuse to perform any calculations that require a valid innovation density, displaying instead an appropriate error message. The upper slider allows the modeler to assign a value to the stitching parameter, ξ, while the lower slider controls the starting value, U_0, on the unit circle for a simulated sample path (see Section 4.5 for a discussion of the choice of U_0). A selection gadget on this panel provides a means to specify the inversion transformation by selecting a requisite distribution (including the empirical histogram) from the resulting pulldown menu, while another gadget allows the modeler to choose between TES$^+$ and

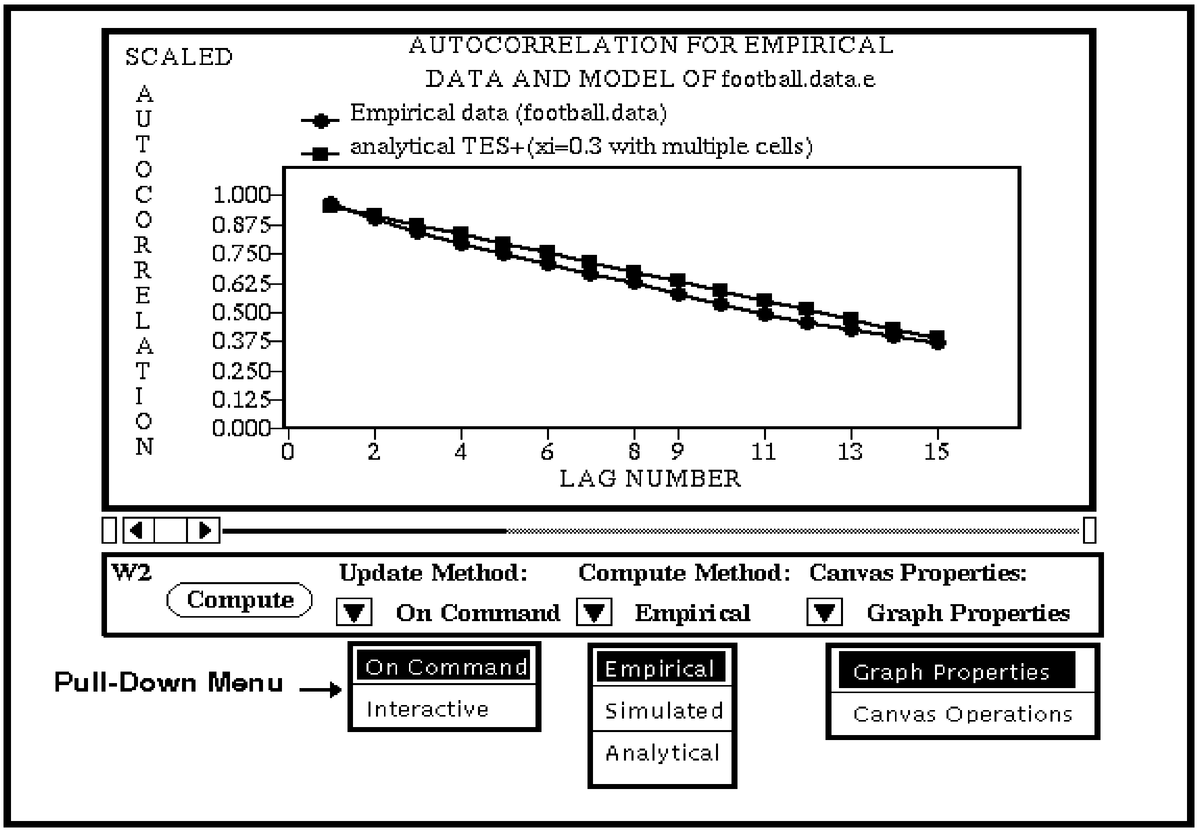

Figure 6: The control panel button pull-down menus.

TES$^-$ model classes (see Section 4.5 for more details).

4.4 Statistical Computations

Directly beneath each of the four display canvases is a control panel containing a menu-button and various selection gadgets (see Figure 6); these allow the modeler to populate the associated canvas with a variety of graphical displays and to tailor their appearances. The `Canvas Properties` Selector, which allows the modeler to perform canvas and graph operations via the popup forms depicted in Figure 3, has already been described in Section 4.1. The remaining button and selection gadgets are described next.

4.4.1 The `Compute` Button

The `Compute` button creates a menu supporting assignment of the canvas to one of four classes of graphical displays supported by TEStool (sample path, histogram, autocorrelation function or spectral density plots). Once a canvas is assigned to a display class, following a dataset load or computation operation, additional graphs may be displayed on the same canvas only if they are of the same class, but without regard to the method of computation. Selecting the `Compute` button creates a selection-menu displaying computational options: sample path, histogram, autocorrelation function or spectral density. If one of these options is selected, TEStool initiates three actions:

1. TEStool verifies that it has complete and valid data on hand to perform the computation. If any required input data is determined to be missing or incorrect, TEStool

informs the modeler via a popup message window and creates the appropriate forms for the modeler to provide the missing information.

2. TEStool first tests whether or not the selected computations have already been performed; if they have, the computation request is ignored.

3. TEStool attempts to perform the requested computations, and if successful, displays the results in graphical form. If the computation requires information found to be missing or incorrect, the modeler is offered the option of supplying the missing data in popup forms or accepting the pre-defined defaults. For example, requesting a simulated histogram graph before the associated sample path has been created makes it necessary to first provide a sample size (via a popup computation form) and perform a Monte-Carlo simulation.

4.4.2 The `Update Method` Selector

The `Update Method` selector determines whether computations are performed only on request, via a compute popup subwindow (not shown), or interactively, in response to modifications to the current TES model. A detailed discussion of TES modeling in the interactive mode is deferred until Section 4.5.

4.4.3 The `Compute Method` Selector

The `Compute Method` selector allows the modeler to choose among the three computation methods (`Empirical`, `Simulation`, and `Analytical`), via a pull-down menu (see Figure 6). Empirical computations are those performed exclusively on externally-loaded sample path datasets (empirical records) to calculate empirical statistics (histogram, autocorrelation function, or spectral density). Simulation computations require first a Monte Carlo generation of a TES model sample path; the calculation of simulation statistics proceeds in the same way as in the empirical case. The analytical mode is used only for the calculation of the exact theoretical autocorrelation function and spectral density of TES models, as well as predicted values and confidence intervals [11].

4.5 Modeling in the Interactive Mode

A major feature of TEStool is its ability to react interactively to changes made to the model displayed in a specification subwindow. However, a number of preliminary operations must be performed before the TEStool interactive mode may be employed.

If an empirical time series is to be modeled, it is of course necessary to first load a dataset of empirical data and to display it as a graph. Recall from Section 4.4.2 that statistical estimates of the marginal distribution, autocorrelation function and spectral density may be computed in the `On Command` mode of the gadget labeled `Update Method` (refer to Figure 6), and displayed on separate canvases. Following these initial steps, a TES model specification may be constructed as described in Section 4.3. Recall also that the modeler may assign any canvases displaying graphs (but not a specification subwindow) to the *interactive mode* by selecting the `Interactive` option of the gadget labeled `Update Method` (refer to Figure 6). Thereafter, any change in the TES model specification will cause TEStool to automatically recompute and redraw all simulated or analytical graphs in all canvases that have been assigned to the interactive mode. Any selection of a TES sign (plus or minus), stitching parameter ξ, initial background sequence value U_0, or a legal innovation density (i.e., $\sum P_k = 1$) qualifies as a specification change. In the TEStool GUI (Figure 5), the ξ and U_0 parameters can be modified by sliders in the range $[0, 1]$, a TES sign choice is made with a pair of pushbuttons to the left of the sliders, and innovation density steps are manipulated by the mouse after selecting a manipulation mode (create/modify, move or delete a step).

The response time required for each update depends upon the computations requested, which may be simulated or analytical; for simulation-based statistics, computation times depend heavily on the size of the underlying sample path. In practice, if the autocorrelations are computed analytically and the sample path simulations generate a relatively small number of points (on the order of $10^3 - 10^4$), both the computations and plot generation will require less than a second of system time on an unloaded Sun Microsystems SPARCstation 2. Statistical estimate calculations of an autocorrelation function and spectral density (as opposed to analytical calculations) will dramatically increase the response time as a function of the size of the sample path involved.

Choosing the initial point U_0 for a simulated sample path is of special interest when the inversion selected is the empirical inverse histogram (3.5) of an empirical record. In this case, it is reasonable to start the simulated path from the same initial value as the empirical path. In the TEStool computation popup form for generating simulated sample paths, alluded to in Section 4.4.2, this is referred to as "anchoring". In order to anchor the initial foreground process value, X_0, to its empirical counterpart, Y_0, one needs to formally invert Eq. (3.3), yielding $U_0 = S_\xi^{-1}(\hat{H}(Y_0))$. From Eq. (3.4), the formal inverse, S_ξ^{-1}, is one-valued only for $\xi = 0$ or $\xi = 1$. For $0 < \xi < 1$, an inverse $S_\xi^{-1}(y)$ is one-valued for $y = \xi$, but two-valued for $y \neq \xi$, so two values of U_0 are generally feasible for a given value of Y_0. To facilitate the selection of a particular anchoring initial value, U_0, the feasible values are automatically computed and stored by TEStool and may be acquired via the buttons (one for each feasible U_0 value) that appear dynamically just below the slider used to assign the value of U_0 (see Section 4.3). These buttons are similar in concept and function to the preset tuning buttons often provided on a radio receiver, that is, their selection will move the slider to the requisite value of an anchoring U_0. Alternatively, selecting the 'Anchor' check-box gadget in the aforementioned compute popup form, before the initial computation of a simulated sample path, will also ensure that anchoring will take place.

In TEStool-aided modeling, the modeler is repeatedly called upon to judge at each iteration whether or not an "acceptable fit" to the empirical data has been attained by a TES model. While quantitative goodness-of-fit criteria of model statistics to their empirical counterparts are straightforward, the corresponding judgment of sample path "resemblance" is highly subjective. Criteria for determining that model-generated sample paths are "similar" to a given empirical record cannot be defined with mathematical precision; rather, an informed judgment requires a level of pattern-matching skills that is more readily attained by human operators than by the most sophisticated computer algorithms. On the other hand, the objective of Monte Carlo modeling is not to reproduce the original empirical sample path, point for point, but to construct stochastic models that can capture the quantitative statistical signature and some qualitative visual patterns of the empirical data. Although the notion of "visual similarity" is largely intuitive, a heuristic approach akin to a Turing Test may be applied as follows: two sample paths would be pronounced "similar" if a statistically significant sample of human observers, familiar with the data, would not be able to distinguish the empirical sample path from those generated by a Monte Carlo simulation of the model. Experience has shown that it is worth some effort to pass a Turing Test, since resemblance of sample paths serves to increase the modeler's level of confidence in the model. It is emphasized that such qualitative matching is done in addition to, not instead of, the quantitative fitting of first-order statistics (marginal distributions) and second-order statistics (autocorrelation functions or spectral densities).

5 Experience With TEStool

Using theTES modeling approach, we have applied TEStool to a variety of empirical data from various disciplines. These are described in detail in [14, 22, 23, 24, 26]. Some specific experiences are briefly summarized below.

1. **MPEG-compressed video bit rates:** the Moving Picture Expert Group (MPEG) is an emerging family of compression standards designed to encode audio-visual signals over broadband transmission channels. Coded picture sequences in MPEG are composed of repeating cycles of three types of frames. The sequence of picture (frame) types within each cycle is deterministic, though their individual bit rates (compressed frame sizes) are random. The modeling approach developed for MPEG-encoded empirical sequences of bit rates relied on creating a set of three TES models, one for each type of picture sequence. These were interleaved in the appropriate (deterministic) order, resulting in an excellent match of the empirical and simulated autocorrelations, marginal distributions and spectral densities [26]. TES models for other compressions are described in [14, 24].

2. **Machine Fault Arrivals:** data representing several months of up-time intervals for a machine engaged in a semiconductor wafer fabrication facility were modeled using TEStool. The less common TES$^-$ model was required to capture the S-shaped functional form of the empirical autocorrelation. The qualitative "resemblance" of the empirical and simulated sample paths turned out to be quite good [22].

3. **Financial Time Series:** TEStool was used to model a sequence of quarterly returns from Treasury Bills (nominal interest rates minus inflation rate) for the period 1950-1990. The TES model was able to approximate the leading empirical autocorrelations (in addition to the empirical histogram), and gave rise to sample paths with considerable qualitative similarity to the observed historical record [22].

4. **Composite Stationary TES Models:** TES models are valid only for stationary sequences, yet many time series, such as compressed video of entire movies represent a collection of contiguous, but statistically distinct, sequences. A composite TES model for such sequences can be constructed by combining multiple TES models (each modeling a distinct stationary segment). The MPEG application [26] described above is an example of a deterministic composition rule. A Markov-modulated TES model is an example of a random composition rule. The latter has been used effectively to model a coded "Star Wars" movie with just four classes of scenes [23].

The visual-interactive facilities of TEStool have proven to be highly effective in accelerating modelers' progress along the "learning curve" of TES modeling. TES methods exhibit a rich behavioral "dynamic range" as a function of TES model parameters. In particular, the TEStool environment facilitates the investigation of the behavior of TES autocorrelations as a function of the innovation density, stitching parameter and inversion transformation, in that it allows a rapid search through a large number of TES models, guided by visual feedback. This makes TEStool suitable for mathematical "experimentation".

In a similar vein, TEStool has proved to be a useful platform for demonstrating statistical concepts associated with stochastic processes to students and laymen. The ability to examine sample paths and their statistics, to create and modify models interactively, and especially, to view the effects of changes to the models in real-time are particularly conducive to learning, since the need for mental abstraction is greatly reduced.

Naturally, the TES modeling methodology cannot be expected to successfully model all stationary time series. For example, Eqs. (2.3) and (2.4) show that any TES autocorrelation function decays geometrically in the lag. Long-range dependent (LRD) processes [1], and particularly the self-similar variety [1, 18], are a case in point. Such processes have hyperbolically-decaying autocorrelation functions, and appear to be good models for some compressed video traffic [6], and ethernet traffic [15]. However, while LRD processes fall outside the domain of TES processes, in practice, one may approximate the former by the latter. As another example we cite empirical time series with additional structure which does not lend itself to faithful modeling by means of a single TES model. The composite models described above are cases in point, showing how this problem might be handled.

6 Conclusion

Recent studies have shown a growing awareness that many modeling applications cannot ignore autocorrelations in random sequences used as source models for stochastic systems. In particular, offered traffic in high-speed communications networks can often be autocorrelated (bursty), resulting in substantially degraded queueing performance compared to that predicted by classical queueing models which ignore autocorrelations in arrival streams and service demands [6, 15, 25]. Consequently, Monte Carlo modeling will increasingly require facilities for modeling autocorrelated time series and generating dependent pseudo-random numbers as input sequences to stochastic systems. The demand for such facilities can be expected to grow: analytical methods can solve only very simple queueing models with autocorrelations, while in reality, modelers are facing stochastic systems with ever-increasing complexity.

This paper has described TEStool — a visual interactive software environment that supports a heuristic approach to modeling autocorrelated empirical records, based on the TES modeling methodology. The salient feature of this methodology is that any prescribed marginal distribution can be automatically matched exactly, while fitting a prescribed autocorrelation function proceeds as a heuristic search. TEStool casts this search into a process of modifying a visual representation of a TES model specification in small incremental perturbations, observing the-goodness-of-fit of the attendant statistics, and then proceeding with the next iteration. Thus, the TEStool GUI serves as an effective aid in the heuristic search activity by facilitating the selection of subsequent perturbations of TES model parameters. TEStool modelers may quickly acquire both a qualitative and quantitative understanding of the relationships between model parameters and their corresponding sample paths and statistics, without having prior grounding in the abstract underlying mathematics of TES processes, as described in Section 2. Because the heuristic search activity more nearly resembles the interactive paradigm of a video arcade game, modeling with TEStool is accessible to both experts and non-experts alike. Practical experience with TEStool has been highly favorable, resulting in high-fidelity TES models of empirical records, from a wide variety of disciplines. Our cumulative experience has confirmed that TEStool speeds up the modeling process, while alleviating the tedium of repetitive search.

TEStool has been recently enhanced by two new modeling features. We conclude this paper with a brief description, since these features are not yet mature and their details are out of the scope of this paper.

A new algorithmic approach has been devised and implemented in TEStool for the TES modeling methodology [12]. The algorithm first carries out a brute-force computation over a subspace of step-function innovation densities and various stitching parameters; recall that the inversion transformation is completely determined by the empirical record and modeler-supplied histogram parameters. Of those, the algorithm selects the best n combinations of pairs, (f_V, ξ), in the sense that the associated TES model autocorrelation functions have the smallest weighted mean-square error with respect to the empirical autocorrelation function. The algorithm then performs non-linear optimization on each model candidate to further reduce the aforementioned mean-square error. Finally, the analyst peruses the results and selects from the n optimized candidate models the one whose Monte Carlo sample path bears the "most resemblance" to the empirical record, in addition to having a small mean-square error. Preliminary experience shows that it produces better and faster results than its heuristic counterpart.

More recently, forecasting capabilities for TES processes have been added to TEStool [11]. The modeler can calculate forward and backward conditional expectations of TES processes, for any lag, given any "current" value; these may be used as point estimates of a TES process and its time-reversed version. Confidence regions straddling these point estimates, can also be computed and displayed graphically and numerically.

Acknowledgement

The authors are indebted to Danny Geist who implemented the first software version of TEStool and to Predrag Jelenkovic who implemented the first version of the algorithmic TES modeling software. We also thank Walter Willinger for providing information on traffic modeling by means of long-range dependent and self-similar processes.

References

[1] J. Beran. *Statistics for Long-Memory Processes.* Chapman and Hall, 1994.

[2] P. Bratley, B.L. Fox and L.E. Schrage. *A Guide to Simulation.* Springer-Verlag, 1987.

[3] G.E.P. Box and G.M. Jenkins. *Time Series Analysis: Forecasting and Control.* Prentice Hall, 1976.

[4] D.R. Cox and P.A.W. Lewis. *The Statistical Analysis of Series of Events.* Methuen and Co., London, 1968.

[5] K.W. Fendick, V.R. Saksena and W. Whitt. Dependence in Packet Queues. *IEEE Trans. on Comm.*, 37:1173–1183, 1989.

[6] M.W. Garrett and W. Willinger. Analysis, Modeling and Generation of Self-Similar VBR Video Traffic. *Proc. of the ACM Sigcomm '94*, London, UK, 269–280, 1994.

[7] H. Heffes and D.M. Lucantoni. 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, 1986.

[8] D.L. Jagerman and B. Melamed. The Transition and Autocorrelation Structure of TES Processes Part I: General Theory. *Stochastic Models*, Vol. 8, No. 2, 193–219, 1992.

[9] D.L. Jagerman and B. Melamed. The Transition and Autocorrelation Structure of TES Processes Part II: Special Cases. *Stochastic Models*, Vol. 8, No. 3, 499–527, 1992.

[10] D.L. Jagerman and B. Melamed. The Spectral Structure of TES Processes. *Stochastic Models*, Vol. 10, No. 3, 599–618, 1994.

[11] D.L. Jagerman and B. Melamed. Bidirectional Estimation and Confidence Regions for TES Processes". *Proceedings of MASCOTS '95*, Durham, North Carolina, 94–98, 1995.

[12] P. Jelenkovic and B. Melamed. Algorithmic Modeling of TES Processes. *Proceedings of INFOCOM '95*, Boston, Massachusetts, 746–752, 1995.

[13] A.M. Law and W.D. Kelton. *Simulation Modeling & Analysis*, (second edition). McGraw-Hill, 1991.

[14] D.-S. Lee, B. Melamed, A. Reibman and B. Sengupta. TES Modeling for Analysis of a Video Multiplexor. *Performance Evaluation*, Vol. 16, 21–34, 1992.

[15] W.E. Leland, M.S. Taqqu, W. Willinger and D.V. Wilson. On the Self-Similar Nature of Ethernet Traffic (Extended Version). *IEEE/ACM Trans. on Networking*, Vol. 2, No. 1, 1–15, 1994.

[16] P.A.W. Lewis and E. McKenzie. Minification Processes and Their Transformations. *J. Appl. Prob.*, Vol. 28, 45–57, 1991.

[17] M. Livny, B. Melamed and A.K. Tsiolis. The Impact of Autocorrelation on Queuing Systems. *Management Science*, Vol. 39, No. 3, 322–339, 1993.

[18] Mandelbrot, B.B., *The Fractal Geometry of Nature.* Freeman, New York, 1983.

[19] B. Melamed. TES: A Class of Methods for Generating Autocorrelated Uniform Variates. *ORSA J. on Computing*, 3:317–329, 1991.

[20] B. Melamed. An Overview of TES Processes and Modeling Methodology. In *Performance Evaluation of Computer and Communications Systems*, (L. Donatiello and R. Nelson, Eds.), 359–393, Springer-Verlag Lecture Notes in Computer Science, 1993.

[21] B. Melamed, D. Goldsman and J.R. Hill. The TES Methodology: Nonparametric Modeling of Stationary Time Series. *Proceedings of Winter Simulation Conference '92*, Arlington, Virginia, 135–144, 1992.

[22] B. Melamed and J. Hill. Applications of the TES Modeling Methodology. *Proceedings of Winter Simulation Conference '93*, Los Angeles, California, 1330–1338, 1993.

[23] B. Melamed and D. Pendarakis. A TES-Based Model for Compressed "Star Wars" Video. *Proceedings of GLOBECOM Communications Mini Conference*, San Francisco, California, 120–126, November, 1994.

[24] B. Melamed, D. Raychaudhuri, B. Sengupta and J. Zdepski. TES-Based Traffic Modeling for Performance Evaluation of Integrated Networks. *Proceedings of INFOCOM '92*, Florence, Italy, 1992.

[25] I. Norros. A Storage Model with Self-Similar Input. *Queueing Systems*, Vol. 16, 387–396, 1994.

[26] D. Reininger, B. Melamed and D. Raychaudhuri. Variable Bit Rate MPEG Video: Characterization, Modeling and Multiplexing. *Proceedings of the 14-th International Teletraffic Congress*, Antibes Juan-les-Pins, France, Vol. 1a, 295–306, June 1994.

[27] B.E. Patuwo, R.L. Disney and D.C. McNickle. The Effects of Correlated Arrivals on Queues. *IIE Transactions*, Vol. 25, No. 3, 105–110, 1993.

[28] B. W. Schmeiser. Simulation Experiments. In *Handbook of Operations Research and Management Science: Volume on Stochastic Models* (D. Heyman and M. Sobel, editors). North Holland, 1990.