ALGORITHMIC MODELING OF TES PROCESSES

Predrag R. Jelenković

Department of Electrical Engineering
and CTR
Columbia University
New York, NY 10027

Benjamin Melamed *

RUTCOR
Rutgers University
P.O. Box 5062
New Brunswick, NJ 08903

Abstract

TES (Transform-Expand-Sample) is a versatile class of stationary stochastic processes which can model arbitrary marginals, a wide variety of autocorrelation functions, and a broad range of sample path behaviors. TES parameters are of two kinds: the first kind is used for the exact fitting of the empirical distribution (histogram), while the second kind is used for approximating the empirical autocorrelation function. Parameters of the first kind are easy to determine algorithmically, but those of the second kind require a hard heuristic search on a large parametric function space.

This paper describes an algorithmic procedure which can replace the heuristic search, thereby largely automating TES modeling. The algorithm is cast in nonlinear programming setting with the objective of minimizing a weighted sum of squared differences between the empirical autocorrelations and their candidate TES model counterparts. It combines a brute-force search with Steepest-Descent nonlinear programming using Zoutendijk's Feasible Direction Method. Finally, we illustrate the efficacy of our approach via three examples: two from the domain of VBR (Variable Bit Rate) compressed video, and one representing results from a laser intensity experiment.

Keywords and Phrases: TES Processes, Autoregressive Processes, Autocorrelation Function, Nonlinear Programming, Algorithmic Modeling.

1 Introduction

Stochastic dependence is quite common in real-world random phenomena, including bursty traffic in high-speed communications networks. Compressed video, also known as VBR (Variable Bit Rate) video, is a case in point. Intuitively, burstiness is present in a traffic process, if arrival points appear to form visual clusters on the time line. Strong positive short-term autocorrelations (second-order properties) are good indicators of traffic burstiness [3, 10], which is also affected by the marginal distributions (first-order properties). The autocorrelation function is a popular statistical proxy of dependence, especially in engineering disciplines, while the marginal distribution of a stationary time series is estimated in practice by the empirical histogram. Analytical models, however, tend to ignore dependence in order to gain analytical tractability; in particular, the bulk of queueing models is

*This work was done in part while the second author was with C&C Research Laboratories, NEC USA Inc., 4 Independence Way, Princeton, NJ 08540.

devoted to the study of queues with independent interarrival and service times. The impact of autocorrelation in traffic processes on queueing measures (e.g., mean queue length, mean waiting times and loss probabilities in finite buffers) can be very dramatic, even in light traffic regimes; worse still, ignoring correlations leads to over-optimistic predictions which are often off by orders of magnitude [6, 25, 12, 16, 22].

A natural idea is to capture first-order and second-order properties of empirical time series (assumed to be from a stationary probability law) by fitting simultaneously both the empirical distribution (histogram) and empirical autocorrelation function. This approach was used in [14], and more recently in the theoretical work reported in [7, 8, 9, 17], and the applied work described in [20, 21, 23, 19]. An extensive survey of modeling methods within this purview may be found in [11].

TES (Transform-Expand-Sample) is a versatile class of stationary stochastic processes with general marginals, a wide variety of autocorrelation functions (e.g., monotone, oscillatory and others), and broad range of sample path behaviors (e.g., directional and reversible) [7, 8, 17]. From a Monte Carlo simulation prospective, TES generation algorithms are fast and require little memory. In essence, TES is a first-order nonlinear autoregressive scheme with modulo-1 reduction and additional transformations. Its specification consists of two kinds of parameters from two distinct sets. The first set, which is algorithmically determined, guarantees an exact match to the empirical distribution (histogram). The second set largely determines the autocorrelation structure. In order to approximate the empirical autocorrelation function, the TES modeling methodology to-date employs a heuristic search approach on a large parametric space.

Effective TES modeling requires computer support. An interactive visual modeling environment, called TEStool, was designed and implemented to support heuristic searches for TES models under human control [4]. TEStool allows the user to read in empirical time series and calculate their statistics (histogram, autocorrelation function and spectral density) in textual and graphical representations. It provides services to construct and modify TES models and to superimpose the corresponding TES statistics on their empirical counterparts. The search proceeds in an interactive style, guided by visual feedback: each model modification triggers a recalculation and redisplay of the corresponding statistics. This approach has several drawbacks. First, effective TES modeling requires qualitative understanding of TES processes; second, the search scope and speed are fundamentally limited by the speed of the human response and the individual extents of human patience; and third, modeling precision is constrained by screen resolution as perceived by the human eye.

This paper presents a TES modeling algorithm, which largely automates the modeling process. The modeler is only asked to specify a few parameters which determine the extent of the algorithmic search, and consequently, the precision of the results, subject to the prevailing limits on the computational complexity. Finally, the algorithm produces multiple candidate TES models, and the final selection of a model is carried out by the modeler by inspecting the associated simulated sample paths and judging their "resemblance" to the empirical data. This last step is not automated, since in the absence of an agreed upon metric, the notion of sample path "resemblance" is necessarily subjective. We mention, though, that modelers are routinely called upon to make such subjective judgments. Our modeling approach centers on the so-called GSLO (Global Search Local Optimization) algorithm. As the name suggests it combines a global search with local nonlinear programming to minimize an objective function consisting of the distance between the empirical autocorrelation function and its candidate model counterpart. The notion of distance is taken as a weighted sum of squared differences between autocorrelations of corresponding lags. Key to this approach is the existence of *fast* and *numerically stable* analytical formulas for calculating the objective function and its partial derivatives, as well as the simplicity of the constraints in the ensuing nonlinear optimization problem. The algorithm has been incorporated into TEStool, which is used to illustrate the efficacy of this approach via three examples: two from the domain of compressed video traffic, and one from a physics laboratory experiment on laser intensity.

The rest of the paper is organized as follows. Section 2 presents an overview of TES processes germane to this paper. Section 3 states the problem in the framework of nonlinear programming. Section 4 discusses the calculation of the partial derivatives of the objective function. Section 5 presents an algorithmic solution to the problem, and Section 6 presents some examples illustrating its efficacy. Finally, Section 7 concludes the paper. Some key formulas utilized in the modeling algorithm are given in the Appendix.

2 TES Processes

This section provides a brief overview of TES processes, relevant to this paper; the reader is referred to Melamed [17], and Jagerman and Melamed [7, 8] for additional details, and to Melamed [18] for a comprehensive overview. Throughout the paper, the Laplace transform of a function f is denoted by $\tilde{f}$, and the indicator function of a set A is denoted by 1_A.

The construction of a TES process involves two random sequences in lockstep. The first sequence, called a *background* TES process (sequence), plays an auxiliary role. It is chosen as either $\{U_n^+\}_{n=0}^{\infty}$ or $\{U_n^-\}_{n=0}^{\infty}$, defined recursively by

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

Here, U_0 is distributed uniformly on $[0, 1)$; $V = \{V_n\}_{n=1}^{\infty}$ is a sequence of iid random variables, independent of U_0, called the *innovation sequence*; and angular brackets denote the modulo-1 (fractional part) operator $\langle x \rangle = x - \max\{\text{integer } n : \ n \le x\}$. The superscript notation in (1) is motivated by the fact that TES sequences, $\{U_n^+\}$ and $\{U_n^-\}$, can generate lag-1 autocorrelations in the ranges [0,1] and [-1,0], respectively. From now on, we will always append plus or minus superscripts to other mathematical objects associated with $\{U_n^+\}$ and $\{U_n^-\}$, but the superscript is omitted when the distinction is immaterial. Intuitively, the modulo-1 arithmetic, used in the definition of the background TES processes $\{U_n^+\}$ in (1), gives rise to a simple geometric interpretation as random walks on a circle of circumference 1 (unit circle), with mean step size $E[V_n]$.

The second sequence, called a *foreground* TES process (sequence), is the target TES model. Foreground TES sequences are denoted by $\{X_n^+\}_{n=0}^{\infty}$ or $\{X_n^-\}_{n=0}^{\infty}$, respectively, and given by

$$X_n^+ = D(U_n^+), \quad X_n^- = D(U_n^-), \tag{2}$$

where D is a real-valued measurable transformation from $[0, 1)$, called a *distortion.* Eq. (2) defines two classes of TES models, denoted by TES$^+$ and TES$^-$, respectively.

The autocorrelation functions of TES processes, with common variance, $0 < \sigma_X^2 < \infty$, can be calculated numerically from fast and accurate formulas [7, 8]. Specifically, for any $\tau \ge 0$, the corresponding autocorrelations of lag τ for $\{X_n^+\}$ and $\{X_n^-\}$, respectively, are given by

$$\rho_X^+(\tau) \;=\; \sum_{\nu=1}^{\infty} Re[\tilde f_V^\tau(i2\pi\nu)]\,\delta_\nu^+, \qquad \rho_X^-(\tau) \;=\; \begin{cases} \sum_{\nu=1}^{\infty} Re[\tilde f_V^\tau(i2\pi\nu)]\,\delta_\nu^+, & \tau \text{ even} \\ \\ \sum_{\nu=1}^{\infty} Re[\tilde f_V^\tau(i2\pi\nu)]\,\delta_\nu^-, & \tau \text{ odd} \end{cases} \tag{3}$$

where

$$\delta_\nu^+ \;=\; \frac{2\,|\tilde D(i2\pi\nu)|^2}{\sigma_X^2}, \quad \delta_\nu^- \;=\; \frac{2\,Re[\tilde D^2(i2\pi\nu)]}{\sigma_X^2}. \tag{4}$$

Analytical formulas for various $\tilde f_V(i2\pi\nu)$ and $\tilde D(i2\pi\nu)$ are given in [8].

Given an empirical time series, $\{Y_n\}_{n=0}^N$, one uses in practice a composite distortion of the form

$$D_{Y,\xi}(x) \;=\; \hat H_Y^{-1}(S_\xi(x)), \quad x \in [0,1). \tag{5}$$

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

$$S_\xi(y) \;=\; \begin{cases} y/\xi, & 0 \le y \le \xi \\ (1-y)/(1-\xi), & \xi \le y < 1. \end{cases} \tag{6}$$

The outer transformation, $\hat H_Y^{-1}$, is the inverse of the empirical (histogram) distribution function computed from $\{Y_n\}$ as

$$\hat H_Y^{-1}(x) \;=\; \sum_{j=1}^{J} 1_{[\hat C_{j-1},\hat C_j)}(x)\,[l_j + (x-\hat C_{j-1})\,\frac{w_j}{\hat p_j}], \quad x \in [0,1), \tag{7}$$

where J is the number of histogram cells, $[l_j, r_j)$ is the support of cell j with width $w_j = r_j - l_j > 0$, $\hat p_j > 0$ is the probability estimator of cell j and $\{\hat C_i\}_{i=0}^J$ is the cdf of $\{\hat p_j\}_{j=1}^J$, i.e., $\hat C_j = \sum_{i=1}^j \hat p_i$, $1 \le j \le J$ ($\hat C_0 = 0$ and $\hat C_J = 1$).

The rationale for TES processes stems from the following facts. First, all TES background sequences are stationary Markovian, and their marginal distribution is uniform on $[0,1)$, *regardless* of the probability law of the innovations $\{V_n\}$ selected, as a consequence of the *General Iterated Uniformity* Lemma in [7]. Second, The *inversion method* [2] permits us, in principle, to transform any uniform variate to others with arbitrary distributions as follows: if U is uniform on $[0,1)$ and F is a prescribed distribution, then $X = F^{-1}(U)$ has distribution F; the case $F = \hat H_Y$ is just a special case. And third, for $0 < \xi < 1$, the effect of S_ξ is to render the sample paths of background TES sequences more "continuous-looking". As stitching transformations preserve uniformity [17], the inversion method can still be applied to stitched background processes, $\{S_\xi(U_n)\}$, so that any foreground TES variate of the form $X_n = \hat H_Y^{-1}(S_\xi(U_n))$, obtained from any background sequence $\{U_n\}$, is always guaranteed to obey the empirical distribution (histogram), $\hat H_Y$, *regardless* of the innovation density f_V and stitching parameter ξ selected. The choice of (f_V, ξ) determines the dependence structure of the sequence $\{X_n\}$ 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, one can concentrate on the latter.

An important property of the autocorrelation functions in (3) is their uniform absolute summability in τ, which is an easy consequence of the facts

$$0 \;\le\; \sum_{\nu=0}^{\infty} |\delta_\nu^-| \;\le\; \sum_{\nu=0}^{\infty} \delta_\nu^+ \;=\; 1, \quad \text{and} \quad |\tilde f_V^\tau(i2\pi\nu)| \;\le\; 1, \;\; \tau \ge 0,\; \nu \ge 1. \tag{8}$$

For our purposes, this allows us to fix $D_{Y,\xi}$, and to use the same finite sum in calculating autocorrelations for all lags τ and all innovation densities f_V, thereby achieving a uniformly bounded error. These calculations are both *fast* and *accurate*. Experimentation has shown that just seven terms in the sums of (3) appear sufficient for keeping the error under 0.01, uniformly in τ.

3 Problem Formulation

We now proceed to formulate the TES fitting problem for TES processes of a specialized form. Specifically, we restrict the discussion to distortions, $D_{Y,\xi}$, from (5) (which represent TES parameters of the first kind), and to pairs, (f_V, ξ), of step-function innovation densities and stitching parameters (which represent TES parameters of the second kind.) These choices of $D_{Y,\xi}$ and f_V have the merit of simplicity, without loss of generality (every density can be approximated arbitrarily closely by step functions).

Recall that an exact match to the empirical histogram is guaranteed by (5). Thus, the problem reduces to one of approximating the empirical autocorrelation function, $\hat{\rho}_Y$, by some TES model autocorrelation function, $\rho_{f_V,\xi}$, to be determined through the choice of (f_V, ξ). To this end, we shall need a metric on the space of autocorrelation functions. This metric should reflect the fact that in most applications (e.g., queueing systems), it is more important to approximate the lower-lag autocorrelations than the higher-lag ones. This consideration leads us to employ an objective function whose general form is a weighted sum of squared differences between the empirical and modeled autocorrelations, namely,

$$g(f_V, \xi) \;=\; \sum_{\tau=1}^{T} a_\tau \left[\rho_{f_V,\xi}(\tau) - \hat{\rho}_Y(\tau)\right]^2, \tag{9}$$

where T is the maximal autocorrelation lag to be approximated, and the $0 < a_\tau \le 1$ are weight coefficients. We can now formally cast the search for a TES model into the following nonlinear optimization problem:

Problem 1 *For a fixed inverse histogram distribution (7), find an optimal innovation density and stitching parameter, (f_V^*, ξ^*), such that*

$$(f_V^*, \xi^*) \;=\; \underset{(f_V,\xi)}{\operatorname{argmin}}\{g(f_V, \xi)\}, \tag{10}$$

where $g(f_V, \xi)$ is given in (9). □

We next restrict the scope of innovation densities considered, again with no loss of generality. From a computational viewpoint, general step-function densities have the drawback that constituent functions have unbounded supports. Alluding to Eq. (1), observe that

$$U_n^+ \;=\; \langle U_{n-1}^+ \,+\, V_n\rangle \;=\; \langle U_{n-1}^+ \,+\, \langle V_n\rangle\rangle,$$

which implies that only innovation variates of the form $\langle V_n\rangle$ need be considered. Consequently, we may restrict consideration to step-function densities f_V, whose support is contained in $[0, 1)$; in fact, owing to the modulo-1 arithmetic in (1), any support interval of length one will do. The interval $[-0.5, 0.5)$ is chosen as a convenient particular case.

Next, to render the minimization procedure tractable, we further restrict admissible f_V to lie in the set $\mathcal{Q} = \cup_{k=1}^{\infty} \mathcal{Q}_k$, where $\mathcal{Q}_k$ is the set of step-function innovation densities over $[-0.5, 0.5)$ of the form

$$f_V(x) \;=\; \sum_{n=1}^{k} \frac{P_n}{1/k}\, 1_{[-0.5+(n-1)/k,-0.5+n/k)}(x), \tag{11}$$

parameterized by the set $\mathcal{P}_k$ of discrete densities $\mathbf{P} = (P_1, \ldots, P_k)$, $\sum_{n=1}^{k} P_n = 1$. In practice, we approximate the set $\mathcal{Q}$ by a subset $\mathcal{Q}_K$ for a large K (say, $K = 100$), and define the parameter space

$$\mathcal{G}_K = \{(\mathbf{P}, \xi) : \; \mathbf{P} \in \mathcal{P}_K, \; \xi \in [0, 1]\}. \tag{12}$$

The optimaization problem (10) then reduces to the following problem:

Problem 2 *For a fixed inverse histogram distribution (7) and a fixed $K > 0$, find optimal parameters $(\mathbf{P}^*, \xi^*) \in \mathcal{G}_K$, such that*

$$(\mathbf{P}^*, \xi^*) \;=\; \underset{(\mathbf{P},\xi)\in\mathcal{G}_K}{\operatorname{argmin}} \{g_K(\mathbf{P}, \xi)\}, \tag{13}$$

where

$$g_K(\mathbf{P}, \xi) \;=\; \sum_{\tau=1}^{T} a_\tau \left[\rho_{\mathbf{P},\xi}(\tau) - \hat{\rho}_Y(\tau)\right]^2, \quad (\mathbf{P}, \xi) \in \mathcal{G}_K, \tag{14}$$

while T and a_τ are as in Eq. (9). $\square$

Problem 2 is a reduction of Problem 1 to a finite-dimensional nonlinear optimization problem with two nice properties. First, it is subject to simple linear constraints; and second, there exist analytical formulas for the objective function and its partial derivatives with respect to every optimization variable. Consequently, Problem 2 is amenable to a variety of standard nonlinear programming techniques [1].

4 Partial Derivatives of the Objective Function

This section derives the partial derivatives of the objective function g_K in (14). To simplify the notation, we write $(P_1, \ldots, P_K, \xi)$, rather than $((P_1, \ldots, P_K), \xi)$, interchangeably with $(\mathbf{P}, \xi)$.

Clearly, $g_K(P_1, \ldots, P_K, \xi)$ has partial derivatives, if $\rho_{P_1,\ldots,P_K,\xi}(\tau)$ does. Recall that (8) ensures that the series in (3) are uniformly convergent, so that we may interchange there the order of differentiation and summation [24], leading to

$$\frac{\partial \rho^{\pm}_{\mathbf{P},\xi}(\tau)}{\partial P_n} \;=\; \sum_{\nu=1}^{\infty} \tau\, Re\!\left[\tilde{f}_V^{\tau-1}(i2\pi\nu)\, \frac{\partial \tilde{f}_V(i2\pi\nu)}{\partial P_n}\right] \delta^{\pm}_{\nu}, \tag{15}$$

$$\frac{\partial \rho^{\pm}_{\mathbf{P},\xi}(\tau)}{\partial \xi} \;=\; \sum_{\nu=1}^{\infty} Re\!\left[\tilde{f}_V^{\tau}(i2\pi\nu)\right] \frac{\partial \delta^{\pm}_{\nu}}{\partial \xi}. \tag{16}$$

The calculation of the partial derivatives above will be outlined next.

An inspection of Eqs. (15) – (16) reveals that the requisite partial derivatives call for the calculation of the transforms $\tilde{f}_V(i2\pi\nu)$, the quantities $\delta_\nu^\pm$, and their partial derivatives.

To calculate the $\tilde{f}_V(i2\pi\nu)$, we can either appeal to Proposition 2 in [8], or carry out a direct calculation of $\tilde{f}_V$, for $f_V \in \mathcal{Q}_K$, using Eq. (11) with $k = K$, yielding

$$\tilde{f}_V(i2\pi\nu) \;=\; \sum_{n=1}^{K} P_n \, \frac{e^{i\pi\nu(K-2n+1)/K}\,\sin(\pi\nu/K)}{\pi\nu/K}. \tag{17}$$

The differentiation of (17) with respect to the P_n is straightforward.

To calculate the $\delta_\nu^\pm$, we first represent

$$\tilde{D}_{Y,\xi}(i2\pi\nu) \;=\; a_{\xi,\nu} + i\, b_{\xi,\nu}, \quad \nu \geq 1. \tag{18}$$

Hence, Eq. (4) becomes

$$\delta_\nu^\pm \;=\; \frac{2}{\sigma_X^2}\left[a_{\xi,\nu}^2 \;\pm\; b_{\xi,\nu}^2\right], \tag{19}$$

since by Eq. (18), $|\tilde{D}_{Y,\xi}(i2\pi\nu)| = a_{\xi,\nu}^2 + b_{\xi,\nu}^2$ and $Re[\tilde{D}_{Y,\xi}^2(i2\pi\nu)] = a_{\xi,\nu}^2 - b_{\xi,\nu}^2$. The corresponding derivatives are readily found to be

$$\frac{\partial \delta_\nu^\pm}{\partial \xi} \;=\; \frac{4}{\sigma_X^2}\left[a_{\xi,\nu}\,\frac{\partial a_{\xi,\nu}}{\partial \xi} \;\pm\; b_{\xi,\nu}\,\frac{\partial b_{\xi,\nu}}{\partial \xi}\right], \tag{20}$$

It remains to calculate the quantities $a_{\xi,\nu}$ and $b_{\xi,\nu}$ in (19), and their partial derivatives in (20).

The formulas for $a_{\xi,\nu}$ and $b_{\xi,\nu}$ are given by Proposition 4 in [8] as follows (see Eq. (7) for notation). For $0 < \xi < 1$,

$$\begin{aligned}
a_{\xi,\nu} \;=\; & \sum_{j=1}^{J} \frac{r_j\left[\sin(2\pi\nu\xi\hat{C}_j) \;+\; \sin(2\pi\nu(1-\xi)\hat{C}_j)\right]}{2\pi\nu} \\
& - \sum_{j=1}^{J} \frac{l_j\left[\sin(2\pi\nu\xi\hat{C}_{j-1}) \;+\; \sin(2\pi\nu(1-\xi)\hat{C}_{j-1})\right]}{2\pi\nu} \\
& + \sum_{j=1}^{J} \frac{w_j}{\hat{p}_j}\left[\frac{\cos(2\pi\nu\xi\hat{C}_j) \;-\; \cos(2\pi\nu\xi\hat{C}_{j-1})}{\xi(2\pi\nu)^2}\right. \\
& \qquad\qquad \left. + \;\frac{\cos(2\pi\nu(1-\xi)\hat{C}_j) \;-\; \cos(2\pi\nu(1-\xi)\hat{C}_{j-1})}{(1-\xi)(2\pi\nu)^2}\right],
\end{aligned} \tag{21}$$

$$\begin{aligned}
b_{\xi,\nu} \;=\; & \sum_{j=1}^{J} \frac{r_j\left[\cos(2\pi\nu\xi\hat{C}_j) \;-\; \cos(2\pi\nu(1-\xi)\hat{C}_j)\right]}{2\pi\nu} \\
& - \sum_{j=1}^{J} \frac{l_j\left[\cos(2\pi\nu\xi\hat{C}_{j-1}) \;-\; \cos(2\pi\nu(1-\xi)\hat{C}_{j-1})\right]}{2\pi\nu} \\
& - \sum_{j=1}^{J} \frac{w_j}{\hat{p}_j}\left[\frac{\sin(2\pi\nu\xi\hat{C}_j) \;-\; \sin(2\pi\nu\xi\hat{C}_{j-1})}{\xi(2\pi\nu)^2}\right. \\
& \qquad\qquad \left. - \;\frac{\sin(2\pi\nu(1-\xi)\hat{C}_j) \;-\; \sin(2\pi\nu(1-\xi)\hat{C}_{j-1})}{(1-\xi)(2\pi\nu)^2}\right],
\end{aligned} \tag{22}$$

while for $\xi = 0$ or $\xi = 1$,

$$a_{0,\nu} \;=\; a_{1,\nu} \;=\; \sum_{j=1}^{J} \left[\frac{r_j \, \sin(2\pi\nu \hat{C}_j) \;-\; l_j \, \sin(2\pi\nu \hat{C}_{j-1})}{2\pi\nu} \right.$$
$$\left. + \; \frac{\cos(2\pi\nu \hat{C}_j) \;-\; \cos(2\pi\nu \hat{C}_{j-1})}{(2\pi\nu)^2} \times \frac{w_j}{\hat{p}_j} \right], \tag{23}$$

$$b_{0,\nu} \;=\; b_{1,\nu} \;=\; \sum_{j=1}^{J} \left[\frac{r_j \, \cos(2\pi\nu \hat{C}_j) \;-\; l_j \, \cos(2\pi\nu \hat{C}_{j-1})}{2\pi\nu} \right.$$
$$\left. - \; \frac{\sin(2\pi\nu \hat{C}_j) \;-\; \sin(2\pi\nu \hat{C}_{j-1})}{(2\pi\nu)^2} \times \frac{w_j}{\hat{p}_j} \right]. \tag{24}$$

The differentiation of (21) – (24) with respect to ξ is straightforward, though moderately tedious.

Some observations on the practical computation of the partial derivatives Eqs. (15) – (16) are warranted at this juncture. First, from (17),

$$\left| \tilde{f}_V(i2\pi\nu) \right| \;\leq\; \frac{1}{\pi\,\nu/K}, \qquad \left| \frac{\partial \tilde{f}_V(i2\pi\nu)}{\partial P_n} \right| \;\leq\; \frac{1}{\pi\,\nu/K}, \qquad \nu \geq 1. \tag{25}$$

From (25) we conclude, with the aid of (8), that (15) is uniformly summable in τ for all $1 \leq n \leq K$ and all $(\mathbf{P}, \xi) \in \mathcal{G}_K$. Hence, we can use the same number of terms to approximate (15), for all $1 \leq n \leq K$, all $\tau \geq 1$, and all $(\mathbf{P}, \xi) \in \mathcal{G}_K$, while retaining a uniformly bounded error. Second, the situation in (16) is more complex, due to the fact that (8) is no longer guaranteed to hold. A careful analysis of the formulas (21) – (24) reveals that $\dfrac{\partial \delta_\nu^\pm}{\partial \xi} = O(1/\nu)$, for $0 \leq \xi \leq 1$. This fact, together with the observation $|\tilde{f}_V(i2\pi\nu)| = O(1/\nu)$, allows us to conclude that the error incurred in approximating (16) by L summands is on the order of $O(1/L)$.

The foregoing discussion provides the basis for a numerical procedure for *fast* and *accurate* calculation of the autocorrelation function and its partial derivatives (with respect to all optimization variables) associated with the TES processes under consideration.

5 The GSLO Algorithm

We are now in a position to present an algorithmic solution for Problem 2, which we term the GSLO (Global Search Local Optimization) algorithm. It is sketched below for given integers, K and B.

GSLO Algorithm Outline

GS Discretize the parameter space, $\mathcal{G}_K$, into a finite number of points. Then, for each such point, evaluate the objective function g_K in (14), and keep the best B points (those points, $\mathbf{x} \in \mathcal{G}_K$, which give rise to the B smallest values of $g_K(\mathbf{x})$).

LO Using each of these B points as a starting point, find a local minimum of g_K via a nonlinear programming algorithm. Then, select among them that point, $\mathbf{x}^*$, which gives rise to the smallest local minimum, $g_K(\mathbf{x}^*)$. □

Note that the global search first selects the B most promising initial points for the local search, so as to increase the chance that the best local minimum found is relatively close to the global minimum. However, the analyst is free to select a less optimal model, if its simulated realizations are judged to bear a "better resemblance" to the empirical record.

In addition to K and B, the global search algorithm requires the specification of two additional parameters, N_P and N_ξ; these are the number of equidistant values that each P_n and ξ can assume, respectively. The total number, N_{tot}, of points, $\mathbf{x}$, at which the GSLO Algorithm needs to evaluate $g_K(\mathbf{x})$ in the **GS** step above, is

$$N_{tot} \;=\; 2\,N_\xi \binom{N_P + K - 2}{N_P - 1}, \tag{26}$$

where the factor 2 is due to the fact that we search both the TES^+ and TES^-classes of processes. Clearly, the parameters N_P, N_ξ and K must be moderate, since N_{tot} grows very fast in them.

A computer implementation of the **GS** step is straightforward. The objective function is calculated inside $K+1$ nested loops (K loops for the discretization of each P_n, and one for that of ξ); the computed value is then compared with the current best set, namely, the running set of the (at most) B best values in some sorting order. If the newly computed value improves on the worst value in the current best set, then the worst value is discarded and the new value is added in the sorting order. We continue this way until we search the whole discretized parameter space. We mention that since the number of loops, $K+1$, is a parameter of the algorithm, loop traversal is implemented by recursive calls.

The local optimization in the **LO** step above was implemented, using the nonlinear programming method, called *Zoutendijk Feasible Direction Method*; see [1], pp. 409. This method is an iterative procedure where at each iteration one determines (i) the optimal feasible direction for the choice of the next point, and (ii) the optimal step size in that direction. Let $\nabla g_K(\mathbf{x})$ denote the vector of the partial derivatives (gradient) of g_K evaluated at $\mathbf{x} \in \mathcal{G}_K$. A *direction* in $\mathcal{G}_K$ is any real vector, $\mathbf{d} = (d_1, \ldots, d_K, d_{K+1})$. Given a *feasible point*, $\mathbf{x} = (\mathbf{P}, \xi)$ in the *feasible space*, $\mathcal{G}_K$, a direction, $\mathbf{d}$, is *feasible*, if $\mathbf{x} + \lambda \mathbf{d} \in \mathcal{G}_K$, for some $\lambda > 0$. A feasible direction, $\mathbf{d}$, is an *improving feasible direction*, if in addition, $\nabla g_K(\mathbf{x})\,\mathbf{d}^t < 0$, t being the transpose operator; see Lemma 10.1.2 in [1]. Thus, the *optimal feasible direction*, $\mathbf{d}^*$, is a solution of the following linear subproblem (for given $\mathbf{x} \in \mathcal{G}_K$):

Subproblem 1 (Optimal Feasible Direction)

$$\begin{array}{ll} \text{Minimize} & \nabla g_K(\mathbf{x})\,\mathbf{d}^t \;\text{ over } \mathbf{d} = (d_1, \ldots, d_K, d_{K+1}), \\ \text{subject to} & \mathbf{d} \text{ is a feasible direction} \\ & \text{and } \; -1 \le d_j \le 1,\; 1 \le j \le K+1 \;\; \text{(normalization).} \end{array} \tag{27}$$

□

Once the optimal feasible direction, $\mathbf{d}^*$, is found, then $\lambda_{max}(\mathbf{x}) = \max\limits_{\lambda>0}\{\mathbf{x} + \lambda \mathbf{d}^* \in \mathcal{G}_K\}$ is the *maximal feasible step size*, and one may proceed to solve for the *optimal step size*, λ^*, in the following subproblem (for given $\mathbf{x} \in \mathcal{G}_K$ and $\mathbf{d}^*$):

Subproblem 2 (Optimal Step Size)

$$\begin{array}{ll} \text{Minimize} & g_K(\mathbf{x}+\lambda \mathbf{d}^*) \;\text{ over } \lambda \\ \text{subject to} & 0 \le \lambda \le \lambda_{max}(\mathbf{x}). \end{array} \tag{28}$$

□

Once the optimal value, λ^*, is found, replace $\mathbf{x}$ by $\mathbf{x} + \lambda^* \mathbf{d}^*$ and solve again for new optimal feasible direction and step size. The algorithm terminates when the optimal value of $\nabla g_{_K}(\mathbf{x})\,\mathbf{d}^t$ in (27) falls below a prescribed threshold. Clearly, it is essential to find efficient solutions for (27) and (28), since these are solved repeatedly.

Suppose for now that (27) in Subproblem 1 has been solved, and consider (28) in Subproblem 2. Since the line search is conducted on the finite interval $(0, \lambda_{max}(\mathbf{x}))$, it is possible, in principle, to find the global minimum on the line segment, with arbitrary precision. However, prescribing a high precision can render the solution overly time consuming. Note that the optimal feasible direction in (27) is not exact since it is based on a linear approximation of the objective function, using its first partial derivatives only. In the theory of nonlinear programming, this is known as the zigzagging effect [1]. Thus, there is no reason to invest heavily in a precise solution of (28). One widely-accepted practical solution to this problem is the so-called Armijo's Rule (see [1], pp. 281), which may be described briefly as follows. Suppose the line search is at point $\mathbf{x}$, and let $\mathbf{d}$ be an improving feasible direction. Let $\theta(\lambda) = g_{_K}(\mathbf{x} + \lambda\mathbf{d})$, $0 \le \lambda \le \lambda_{max}(\mathbf{x})$, and let $0 < \epsilon < 1$ and $\alpha > 1$ be two parameters (our implementation used $\alpha = 2, \epsilon = 0.2$). Define further, $\hat{\theta}(\lambda) = \theta(0) + \lambda\epsilon\theta'(0)$, where prime indicates differentiation. Armijo's Rule initially sets $\lambda = \lambda_{max}((\mathbf{x})$. Then, while $\theta(\lambda) > \hat{\theta}(\lambda)$, set $\lambda = \lambda/\alpha$ and repeat; and otherwise, set $\lambda^* = \lambda$ and exit.

Returning to the solution of (27), observe that a closed form solution can be obtained, owing to the relatively simple linear constraints involved. Since $P_K = 1 - \sum_{n=1}^{K-1} P_n$, a reduction in the problem dimension may be attained as follows. First, replace the original parameter space, $\mathcal{G}_K$, by the reduced parameter space,

$$\mathcal{H}_{K-1} \;=\; \{(P_1, \ldots, P_{K-1}, \xi) \colon P_n \ge 0,\ 1 \le n \le K-1,\ \sum_{n=1}^{K-1} P_n \le 1,\ \xi \in [0,1]\},$$

and second, replace the original objective function $g_{_K}$, by a new objective function $h_{_{K-1}}$, over $\mathcal{H}_{K-1}$, given by

$$h_{_{K-1}}(P_1, \ldots, P_{K-1}, \xi) \;=\; g_{_K}(P_1, \ldots, P_{K-1}, 1 - \sum_{j=1}^{K-1} P_j, \xi), \quad (P_1, \ldots, P_{K-1}, \xi) \in \mathcal{H}_{K-1}. \tag{29}$$

Consider a a variant of the optimization problem (27), but with $\mathcal{G}_K$ replaced by $\mathcal{H}_{K-1}$, $g_{_K}$ replaced by $h_{_{K-1}}$, and direction vectors of the form $\mathbf{d} = (d_1, \ldots, d_K)$. Assuming that the normalization constraints (but not the feasibility constraints) in (27) are satisfied for the new problem, it is clear that the minimum of $\nabla h_{_{K-1}}(\mathbf{y})\mathbf{d}^t$ is attained for

$$d_j \;=\; -\mathrm{sign}\left(\partial h_{_{K-1}}(\mathbf{y})/\partial y_j\right), \quad 1 \le j \le K, \tag{30}$$

where sign(z) is +1 or -1 according as z is non-negative or negative, respectively. Our goal is to change d_j in such a way that the feasibility constraints in (27) are satisfied, while $\nabla h_{_{K-1}}(\mathbf{y})\,\mathbf{d}^t$ is increased as little as possible. To this end, we first fix the boundary constraints for each coordinate, n. For example, if $P_n = 0$ and $d_n = -1$, then set $d_n = 0$; similar actions are taken for other boundary values, e.g., for $P_n = 1$ and $\xi = 0$ or $\xi = 1$. Finally, the only constraint left is $\sum_{n=1}^{K-1} P_n \le 1$ and an infeasible direction will ensue if $\sum_{n=1}^{K-1} P_n = 1$ and $\sum_{j=1}^{K-1} d_j > 0$. The best feasible d_j are obtained when their sum vanishes, coupled with a minimal increase of $\nabla h_{_{K-1}}(\mathbf{y})\,\mathbf{d}^t$. Let

$$I_{K-1}(\mathbf{d}) \;=\; \{1 \le j \le K-1\colon\ d_j = 1 \ \text{ or } \ (d_j = 0 \ \text{ and } \ P_j = 1)\}$$

be the set of indices j, for which d_j can be decreased without violating the normalization constraint in (27). It is readily seen that $\nabla h_{K-1}(\mathbf{y})\,\mathbf{d}^t$ would increase the least by decreasing that d_n, such that $-\partial h_{K-1}(\mathbf{y})/\partial P_n$ is minimized over $I_{K-1}(\mathbf{d})$. For such an index n, decrease d_n just enough to obtain $\sum_{j=1}^{K-1} d_j = 0$; if this is not possible, set $d_n = -1$, remove n from $I_{K-1}(\mathbf{d})$ and repeat. The optimal direction is obtained when the corresponding sum of d_j vanishes.

6 Examples

This section illustrates the efficacy of the algorithmic TES modeling methodology via three examples: two from the domain of compressed video traffic, and one from a laboratory experiment on an NH_3 laser intensity. All examples utilize $K = 100$, $N_P = 4$, $N_\xi = 11$, and $B = 150$, resulting in a total of 3.77×10^6 searches in the **GS** algorithm. Figures 1 – 3 display, for each respective example, a TEStool screen depicting the results of the corresponding algorithmic TES modeling.

6.1 H.261-Compressed Video

Data compression is extensively used to reduce the transmission bandwidth requirements of telecommunications traffic. The idea is to code the data at the source, thereby compressing it to a fraction of its original size, and then transport the compressed data over the network and decode it at its destination. Video service in emerging ISDN (Integrated Service Digital Networks) is a typical application, for which the exact reproduction of the original signal is not necessary. In fact, redundant visual information, to which the human eye is relatively insensitive, may be removed without degrading the perceptual quality of the decoded image. H.261 is a popular coding standard, which makes use of DCT (Discrete Cosine Transform) and other techniques to compress video spatial units (frames or subframes) [15]. Since such coded units have random (but highly autocorrelated) transmission requirements (say, in bits), the corresponding coding schemes are referred to as VBR (Variable Bit Rate).

Figure 1 displays a TEStool screen showing the results of the algorithmic TES modeling for an empirical sample path of VBR video (frame sizes), in which the coding scheme used was a variant of the H.261 standard [21]. The video scene content was a football sequence and the depicted results are for a TES$^+$ model. Note the excellent agreement of the TES model histogram and autocorrelation function with their empirical counterparts in the upper-right and lower-left canvases, respectively. Furthermore, the corresponding sample paths in the upper-left canvas are markedly “similar”. The GSLO-obtained innovation density is depicted in the lower-right canvas.

6.2 MPEG-Compressed VBR Video

MPEG (Moving Picture Expert Group) is an emerging family of compression standards designed to encode audio-visual signals over broadband transmission channels ([13]). The importance of MPEG derives from its planned central role in facilitating future delivery of multi-media services to customer premises. This section focuses on MPEG-based video, designed to compress a full-motion video stream to about 1.5 Mbits/second.

Coded picture sequences in MPEG are composed of cycles. A coded picture can be either an Intrapicture (I-frame), Predicted picture (P-frame) or Bidirectionally Predicted picture (B-frame). The sequence of picture (frame) types within each cycle is deterministic, though the corresponding bit rates are random. MPEG type sequences can be chosen as an MPEG parameter, depending on the application. The probability laws of frame types are markedly different. In particular, the marginal distributions of I-frames, P-frames, and B-frames are well separated, with descending means in this order. Consequently, MPEG-compressed sequences are non-stationary, due to the determinism of the frame type sequence. The particular type sequence chosen in the case study described here had a length-nine cycle of the form IBBPBBPBB... [23]. The modeling approach called for a composite TES model. First, each subsequence of MPEG frame types was modeled as a separate TES sequence, I-frames and B-frames each by a TES^+ model and P-frames by a TES^- model.

Figure 2 depicts a TEStool screen showing the algorithmic modeling results for the P-frame subsequence. The video scene content was a "busy" sequence of a toy train in motion, combined with a moving calendar [23]. The figure is similar in structure to the previous one, and the model statistics are again in excellent agreement with their empirical counterparts.

6.3 NH_3 Laser Data

The Santa Fe Institute conducts competitions in time series prediction, using neural net methods. An empirical set of partial random data is made available and competitors are asked to predict the rest of the time series. The data set in this example consisted of 1000 data points, representing a clean physics laboratory of the fluctuating intensity of an NH_3 laser experiment, reposited in `ftp.santafe.edu` (see [5]). The GSLO algorithm assumed a maximal autocorrelation lag of $T = 15$.

Figure 3 depicts the TEStool screen showing the algorithmic modeling results for the laser data. Again the figure is similar in structure to the previous ones, and the model statistics are in excellent agreement with their empirical counterparts. Furthermore, the model exhibits considerable predictive power in the sense that in the time interval $(0, 30)$, the sample path of the model is very close to its empirical counterpart.

7 Conclusion

This paper has developed an algorithmic methodology for TES modeling, thereby shifting the modeling burden from a human conducting a heuristic search over a large parametric space. The algorithmic search is largely automated; the user is only required to specify a few discretization parameters which determine the accuracy and the computational complexity of the search. The end-product of the algorithm is a set of models whose number is a user-supplied parameter. The final choice is made by a user perusing the results, based on the "resemblance" of the model (simulated) sample paths to their empirical counterparts.

The algorithm has been incorporated into the TEStool modeling environment to supplement its heuristic search support. Experimentation with the modeling algorithm, as implemented in TEStool, has yielded remarkably accurate TES models of empirical records in a relatively short period of time, typically on the order of minutes. The efficacy of the modeling algorithm was illustrated by three case study examples. Experience with a variety of additional empirical data sets supports our claim that the algorithmic TES modeling methodology presented here can serve as a powerful input analysis technique for simulation analysis in general, and broadband video traffic modeling, in particular. Finally, we remark that since user involvement is minimal, requiring little expertise in TES processes, this technique may be comfortably used by experts and non-experts alike.

Acknowledgments

Special thanks are due to D.L. Jagerman and B. Sengupta for many valuable suggestions. The second author wishes to thank RUTCOR and especially its director, Peter Hammer, for their uncommon hospitality.

References

[1] Bazaraa, M.S., Sherali, H.D. and Shetty, C.M. *Nonlinear Programming*, Wiley, 1993.

[2] Devroye, L. *Non-Uniform Random Variate Generation*, Springer-Verlag, 1986.

[3] Frost, V. and Melamed, B. "Overview of Simulation and Traffic Modeling for Telecommunications Networks", *IEEE Communications Magazine*, Vol. 32, No. 3, 70–81, 1994.

[4] Geist, D. and Melamed, B. "TEStool: An Environment for Visual Interactive Modeling of Autocorrelated Traffic", *Proceedings of the 1992 International Conference on Communications*, Vol. 3, 1285–1289, Chicago, Illinois,1992.

[5] Gershenfeld, N.A. and Weigend, A.S. "Results of the Time Series Prediction Competition at the Santa Fe Institute", *Proc. IEEE International Conference on Neural Networks*, Vol. 3, 1786–1793, San Francisco, California, 1993.

[6] Jacobs, P.A. "A Cyclic Queueing Network With Dependent Exponential Service Times", *J. Appl. Prob.*, Vol. 15, 573–589, 1978.

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

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

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

[10] Jagerman, D.L. and Melamed, B. "Burstiness Descriptors of Traffic Streams: Indices of Dispersion and Peakedness", *Proceedings of the Twenty Eighth Annual Conference on Information Sciences and Systems*, Vol. 1, 1–5, Princeton, New Jersey, March 1994.

[11] G.E. Johnson. Construction of Particular Random Processes. *Proc. of the IEEE*, Vol. 82, No. 2, 270–285, 1994.

[12] Latouche, G. "An Exponential Semi-Markov Process, With Applications to Queueing Theory", *Stochastic Models*, Vol. 1, No. 2, 137–169, 1985.

[13] Le Gall, D. "MPEG: A Video Compression Standard for Multimedia Applications", *Communications of the ACM* Vol. 34, 46–58, 1991.

[14] Liu, B. and Munson, D.C. "Generation of Random Sequence Having a Jointly Specified Marginal Distribution and Autocovariance", *IEEE Transaction on Acoustics, Speech and Signal Processing*, Vol. 30, No. 6, 973–983, 1982.

[15] Liou, M. "Overview of the px64 kbit/s Video Coding Standard", *Communications of the ACM*, Vol. 34, No. 4, 59–63, 1991.

[16] Livny, M., Melamed, B. and Tsiolis, A.K. "The Impact of Autocorrelation on Queuing Systems", *Management Science*, Vol. 39, 322–339, 1993.

[17] Melamed, B. "TES : A Class of Methods for Generating Autocorrelated Uniform Variates", *ORSA J. on Computing*, Vol. 3(4): 317–329, 1991.

[18] Melamed, B. "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.

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

[20] Melamed, B., Raychaudhuri, D., Sengupta, B. and Zdepski, J. "TES-Based Traffic Modeling for Performance Evaluation of Integrated Networks", *Proc. IEEE INFOCOM*, Florence, Italy, 1992. Also to appear in *IEEE Trans. in Comm.*, 1994.

[21] Melamed, B. and Sengupta, B. "TES Modeling of Video Traffic", *IEICE Transactions on Communications*, Vol. E75-B, No. 12, 1292–1300, 1992.

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

[23] Reininger, D., Melamed, B. and Raychaudhuri, D. "Variable Bit Rate MPEG Video: Characterization, Modeling and Multiplexing", *Proc. 14-th International Traffic Congress*, Antibes Juan-les-Pins, France, Vol. 1a, 295–306, 1994.

[24] Rudin, W. *Principles of Mathematical Analysis*, McGraw-Hill, 1964.

[25] Tin, P. "A Queueing System with Markov-Dependent Arrivals", *J. Appl. Prob.*, Vol. 22, 668–677, 1985.

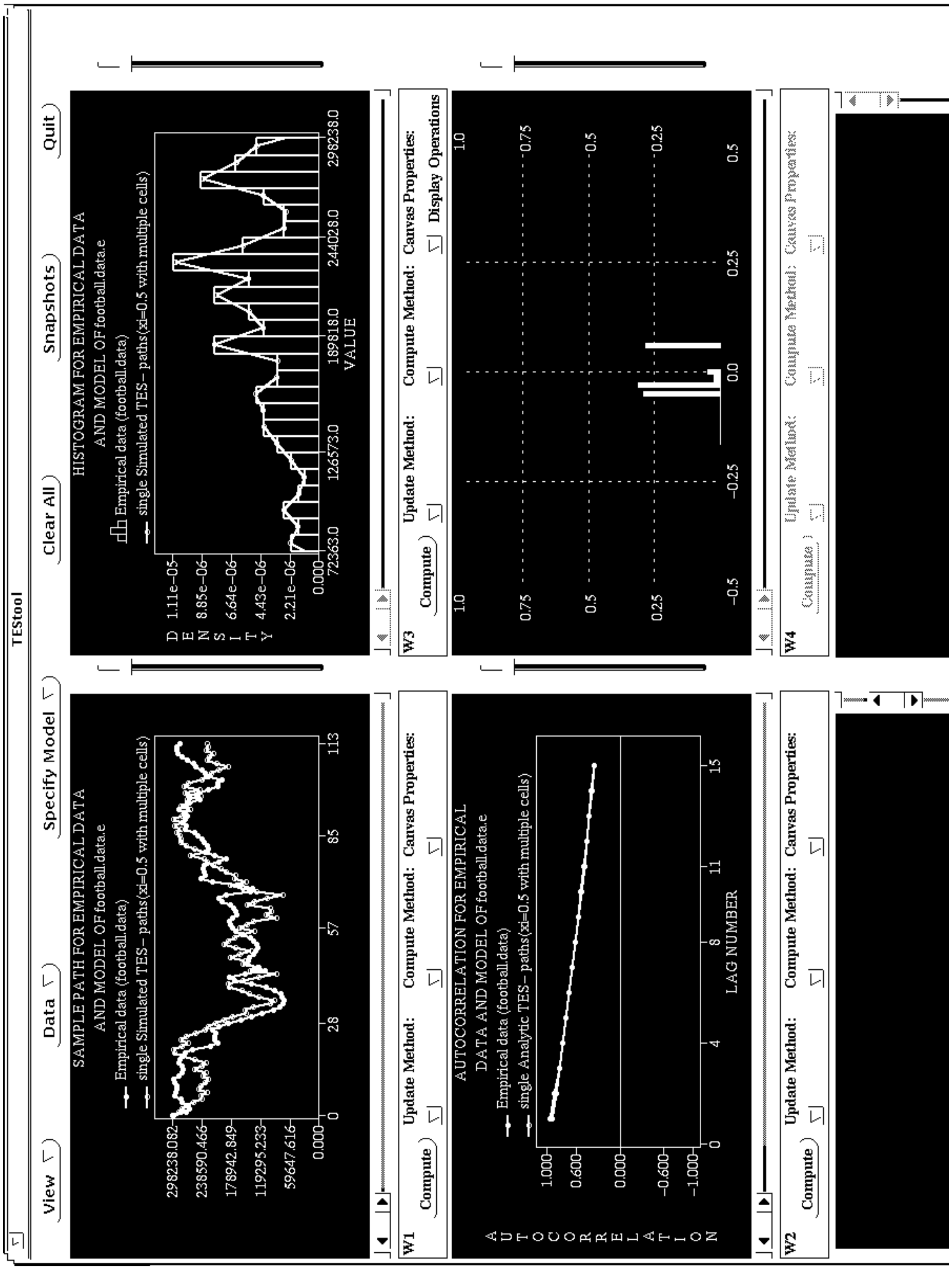

Figure 1: A TES$^+$ Model for a H.261-Compressed VBR video

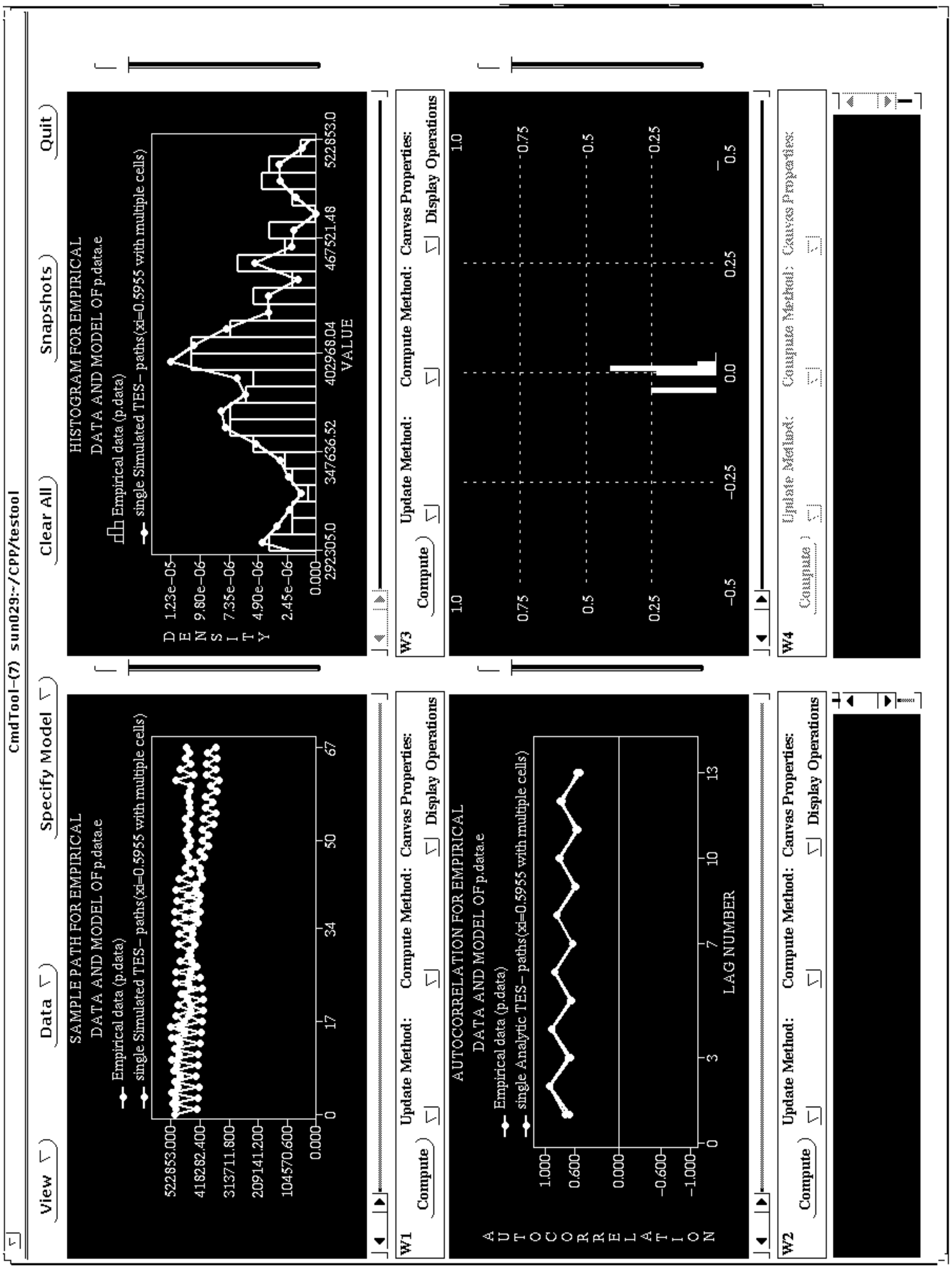

Figure 2: A TES$^-$ Model for P-Frame subsequence from an MPEG-Compressed VBR video

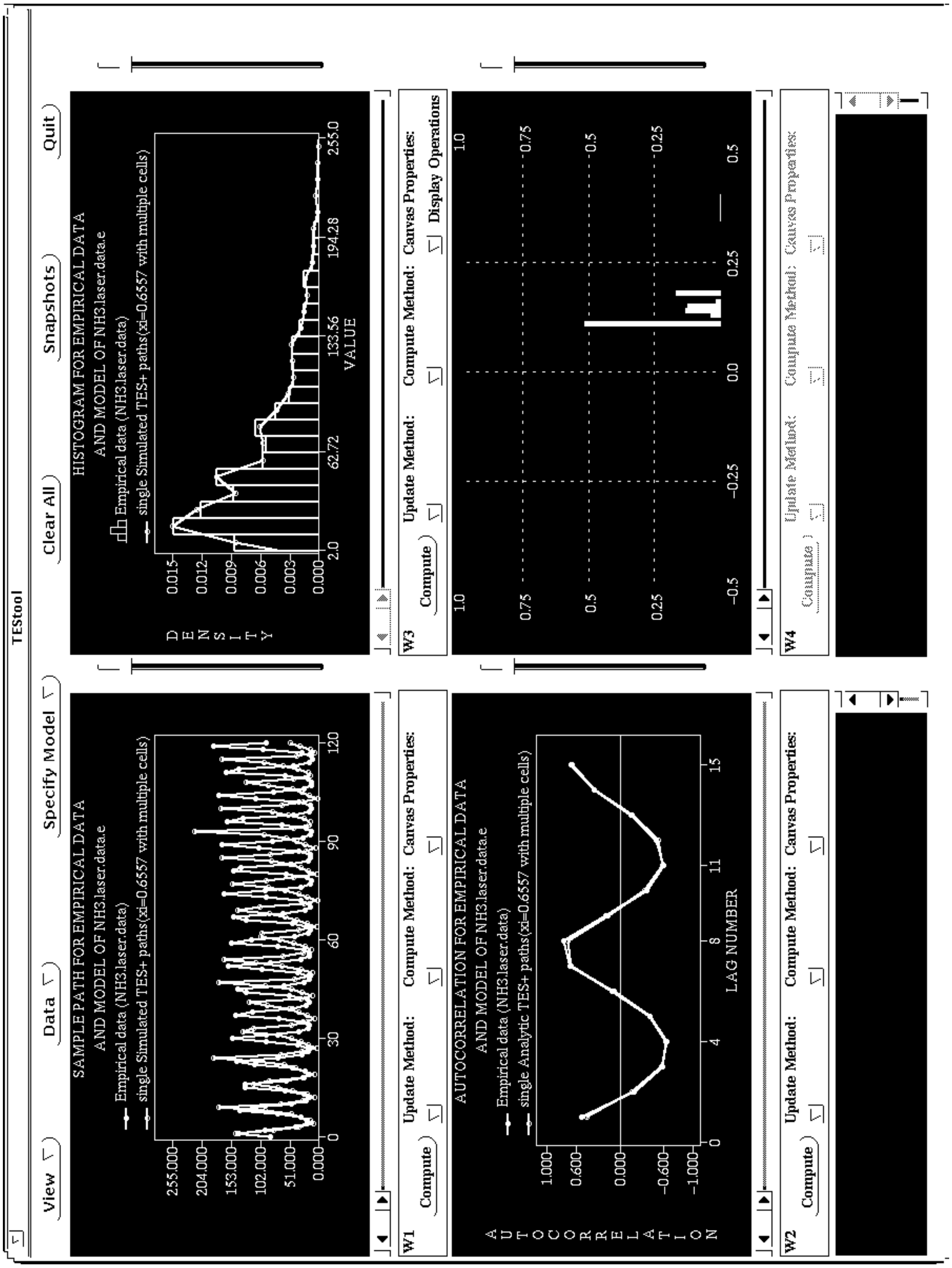

Figure 3: A TES$^+$ Model for the NH_3 Laser Data