INFORMATION TO USERS

This manuscript has been reproduced from the microfilm master. UMI films the text directly from the original or copy submitted. Thus, some thesis and dissertation copies are in typewriter face, while others may be from any type of computer printer.

The quality of this reproduction is dependent upon the quality of the copy submitted. Broken or indistinct print, colored or poor quality illustrations and photographs, print bleedthrough, substandard margins, and improper alignment can adversely affect reproduction.

In the unlikely event that the author did not send UMI a complete manuscript and there are missing pages, these will be noted. Also, if unauthorized copyright material had to be removed, a note will indicate the deletion.

Oversize materials (e.g., maps, drawings, charts) are reproduced by sectioning the original, beginning at the upper left-hand corner and continuing from left to right in equal sections with small overlaps. Each original is also photographed in one exposure and is included in reduced form at the back of the book.

Photographs included in the original manuscript have been reproduced xerographically in this copy. Higher quality 6" x 9" black and white photographic prints are available for any photographs or illustrations appearing in this copy for an additional charge. Contact UMI directly to order.

UMI

A Bell & Howell Information Company
300 North Zeeb Road, Ann Arbor MI 48106-1346 USA
313/761-4700 800/521-0600

Automated Modeling of Broadband Network Data Using the QTES Methodology

by

G. Klaoudatos, BEng.

Electrical Engineering, University of Patras. Greece

A thesis submitted to
the Faculty of Graduate Studies and Research
in partial fulfillment of
the requirements for the degree of

Master of Engineering

Ottawa-Carleton Institute for Electrical Engineering
Department of Systems and Computer Engineering

Carleton University
Ottawa, Ontario
October 8, 1997
© copyright
1997, G. Klaoudatos

National Library
of Canada

Acquisitions and
Bibliographic Services

395 Wellington Street
Ottawa ON K1A 0N4
Canada

Bibliothèque nationale
du Canada

Acquisitions et
services bibliographiques

395, rue Wellington
Ottawa ON K1A 0N4
Canada

Your file   Votre référence

Our file   Notre référence

The author has granted a non-exclusive licence allowing the National Library of Canada to reproduce, loan, distribute or sell copies of this thesis in microform, paper or electronic formats.

The author retains ownership of the copyright in this thesis. Neither the thesis nor substantial extracts from it may be printed or otherwise reproduced without the author's permission.

L'auteur a accordé une licence non exclusive permettant à la Bibliothèque nationale du Canada de reproduire, prêter, distribuer ou vendre des copies de cette thèse sous la forme de microfiche/film, de reproduction sur papier ou sur format électronique.

L'auteur conserve la propriété du droit d'auteur qui protège cette thèse. Ni la thèse ni des extraits substantiels de celle-ci ne doivent être imprimés ou autrement reproduits sans son autorisation.

0-612-26998-1

The undersigned recommend to the Faculty of Graduate Studies
and Research acceptance of the thesis

**Automated Modeling of Broadband Network Data Using the QTES
Methodology**

submitted by G. Klaoudatos. BEng.

in partial fulfilment of the requirements for
the degree of Master of Engineering

Thesis Supervisors

Chair, Department of Systems and
Computer Engineering

Carleton University

October 8, 1997

Abstract

As new and more advanced communication services evolve, the demand for better models which will predict system performance more accurately becomes larger. A number of these services supported by high speed networks give rise to bursty (auto-correlated) traffic streams. A typical example is VBR (variable bit rate) compressed video.

The objective of this Thesis is to investigate a new modeling methodology called QTES (Quantized Transform-Expand-Sample) which can be used to model network traffic taking into consideration both the marginal and the autocorrelation function of the empirical data. An effort is made towards an algorithmic procedure rather than a heuristic search, thereby largely automating QTES modeling.

Acknowledgements

I would like to express my sincere thanks to my thesis supervisors. Prof. I. Lambadaris and Prof. A. R. Kaye as well as Prof. M. Devetsikiotis for helping me achieve a focused and organized approach to research and for their guidance throughout the course of this thesis. I wish also to profusely thank Prof. Makios from the Electrical Engineering department at the University of Patras for his help and support. His encouragement has been deeply appreciated.

I wish to profusely thank my family, for their enduring patience, support and encouragement during this research.

I gratefully acknowledge the funding support provided by the Telecommunications Research Institute of Ontario.

Table of Contents

Acceptance Sheet	ii
Abstract	iii
Acknowledgements	iv
Table of Contents	v
List of Tables	viii
List of Figures	xi
List of Acronyms	xii
1 Introduction	1
1.1 Background	1
1.2 Previous Research	4
1.3 Motivation and Objectives of this Thesis	7
1.4 Thesis Organization	10
2 Transform Expand Sample Models	12
2.1 TES Processes	12
2.2 Discretized and Quantized TES Processes	17
2.2.1 Discretized TES Processes (DTES)	17
2.2.2 Quantized TES Processes (QTES)	19

2.3	Summary	21
3	Automated Modeling of Broadband Network Data Using the QTES	
	Methodology	22
3.1	Problem Formulation	23
3.2	Algorithmic Approach	25
3.3	Implementation of the Exhaustive Search	25
3.3.1	Next Composition	26
3.3.2	Global Search	29
3.4	Local Search	31
3.4.1	Constraint Optimization	31
3.4.2	Sequential Quadratic Programming (SQP)	32
3.4.3	Partial Derivatives of the objective function	34
3.5	Examples and Results	36
3.6	Summary	41
4	Random Search	43
4.1	Random Search Algorithm	44
4.1.1	Random k-subset of an n-Set	45
4.1.2	Random Composition of n into k parts (rancom)	48
4.2	Examples and Results	48
4.3	Summary	52
5	Comparison Between Exhaustive Search and Random Search	53
5.1	Summary	58

6	QTES Generator	60
6.1	Inverse Transform Method	61
6.2	QTES algorithm	63
6.2.1	Discrete number generator (<i>dis_genn</i>)	63
6.2.2	Background QTES generator (<i>qttes</i>)	64
6.2.3	Foreground QTES generator	64
6.3	Results	66
6.4	Summary	70
7	Conclusions and Recommendations for further Study	71
7.1	Conclusions	71
7.2	Recommendations for Further Study	75
7.2.1	Experimental Design and Design Optimization	75
7.2.2	Other Recommendations	79

List of Tables

5.1	Required running time (hours) for obtaining the traffic models from the empirical VBR video samples	58
7.1	Generic 3 factor design analysis	77

List of Figures

2.1	Probability Density of Innovation Sequence V_n	14
2.2	Migration of correlated background sequence on the unit circle	14
2.3	Probability density of Innovation Sequence V_m	15
3.1	Flowchart for the Next k -Subset of an n -Set routine	28
3.2	Flowchart for the Next Composition routine	30
3.3	Autocorrelation Coefficients for the BBC news empirical traffic sample and the QTES models for $K=10$ and $K=20$ using the exhaustive search	40
3.4	Autocorrelation Coefficients for the Last Action Hero empirical traffic sample and the QTES models for $K=10$ and $K=20$ using the exhaustive search	40
3.5	Autocorrelation Coefficients for the Football empirical traffic sample and the QTES models for $K=10$ and $K=20$ using the exhaustive search	41
4.1	balls in cells model	45
4.2	balls in cells model example	45
4.3	Flowchart for the ranksb algorithm	49
4.4	Autocorrelation Coefficients for the BBC News empirical traffic sample and the QTES models for $K=10$ and $K=20$ using random search . . .	50

4.5	Autocorrelation Coefficients for the Last Action Hero empirical traffic sample and the QTES models for $K=10$ and $K=20$ using random search	51
4.6	Autocorrelation Coefficients for the Football empirical traffic sample and the QTES models for $K=10$ and $K=20$ using random search . . .	51
5.1	Autocorrelation Coefficients for the empirical data stream and the QTES ($K=20$) models of example1 (BBC video sample) using exhaustive and random search	54
5.2	Autocorrelation Coefficients for the empirical data stream and the QTES ($K=20$) models of example1 (Last Action Hero video sample) using exhaustive and random search	54
5.3	Autocorrelation Coefficients for the empirical data stream and the QTES ($K=20$) models of example3 (Last Action Hero video sample) using exhaustive and random search	55
5.4	Autocorrelation Coefficients for the empirical data stream and the QTES ($K=20$) models of example1 (football video sample) using exhaustive and random search	56
5.5	Autocorrelation Coefficients for the QTES ($K=10$) models of example3 (football video sample) for $N=30$ and $N=70$ using random search . .	57
5.6	Autocorrelation Coefficients for the QTES ($K=20$) models of example3 (football video sample) for $N=100$ and $N=500$ using random search .	57
6.1	QTES generator	62
6.2	Inverse-transform method for continuous random variables	63

6.3	Continuous, piecewise-linear empirical distribution function from grouped data	65
6.4	QTES models ($K = 10$ and $K = 20$) of the BBC news video traffic sample	67
6.5	QTES models($K = 10$ and $K = 20$) of the Last Action Hero video traffic sample	68
6.6	QTES models($K = 10$ and $K = 20$) of the football video traffic sample	69

List of Acronyms

ATM	Asynchronous Transfer Mode
BISDN	Broadband Integrated Services Digital Network
DCT	Discrete Cosine Transform
DES	Discrete Event Simulation
DTES	Discrete Transform Expand Sample
FR	Frame Relay
FDDI	Fiber Distributed Data Interface
IEEE	Institute of Electrical and Electronic Engineers
IID	Independent Identically Distributed
ISDN	Integrated Services Digital Network
KT	Kuhn-Tucker
LAN	Local Area Network
MMPP	Markov-Modulated Poisson Processes
MPEG	Motion Picture Experts Group
QoS	Quality of Service
QP	Quadratic Programming
QTES	Quantized Transform Expand Sample
SQP	Sequential Quadratic Programming
TES	Transform Expand Sample
VBR	Variable Bit Rate
WAN	Wide Area Network

QTES	Quantized Transform Expand Sample
SQP	Sequential Quadratic Programming
TES	Transform Expand Sample
VBR	Variable Bit Rate
WAN	Wide Area Network

Chapter 1

Introduction

1.1 Background

One of the vital elements in today's technological environment is the efficient and fast flow of information through telecommunications networks. This flow is supported by complex computer and communications structures that, if properly designed and operated, are invisible to the end users. High-speed network transport mechanisms, such as the asynchronous transfer mode (ATM), cell relay and the frame relay (FR) allow transmission speeds of 1.5 Mb/s to 150 Mb/s as seen by the end user, while they serve also as enabling technologies for new classes of communication services, such as multimedia and video on demand that are typically grouped under the heading of B-ISDN [1].

As these new communications services evolve and the needs of users change, the enterprise must respond by modifying existing communications systems or by implementing entirely new ones. To this end, telecommunications professionals are being called upon to design and manage these systems in the face of fast-moving technology and a climate of increasing customer expectations. Design and management decisions require predictions of network performance. Quality of Service (QoS)

considerations have to take into account the fact that such services are both delay-sensitive and loss-sensitive as well. In the face of all this, network and traffic designers must decide how to set a number of parameters, for instance the threshold values, and evaluate the expected performance under a wide variety of traffic conditions.

All this bring us to two issues that constitute the core subject of emerging networks: performance evaluation and traffic modeling. Monte Carlo computer simulation [1] is a flexible performance prediction tool used widely in sciences and engineering. Its flexibility comes from the fact that it consists of a computer program that “behaves” like the system under study. Analytical methods are also used as an alternative to avoid the huge computing load associated with conventional Monte Carlo simulation. Unfortunately, analytical models require many assumptions and are too restrictive for most real-world systems. Most simulation tools for telecommunications networks though are based on Discrete Event Simulation (DES) [2, 3]. The most important feature of DES models is that they keep track of time via simulation clocks, which change by random increments. The basic executable unit in DES models is an event (a program that is executed at discrete simulation times).

Traffic modeling is a key element in simulating communications networks. A clear understanding of the nature of traffic which goes through a particular system, its statistical properties and the selection of an appropriate traffic model are vital issues for the success of the modeling enterprise. In ATM and FR networks, buffer overflow is the major cause of losses. Therefore, the probability and the correlations of such losses, together with the queuing delays constitute major issues of the QoS criteria. Such criteria are extremely sensitive to the accuracy of the traffic models used in their prediction.

Many the existing traffic models do not represent the traffic in networks with sufficient accuracy. In particular, they may not fit either the first order statistics (marginal probability distribution) or the second order statistics (autocorrelation function) of the empirical time series. Temporal dependence which is proxied by the autocorrelation function is a major cause of burstiness in telecommunications traffic, especially in emerging high-speed communications networks: typical examples are file transfers and compressed variable bit rate video (VBR). The autocorrelation function is a convenient measure of temporal dependence in real-valued stochastic processes, frequently used by engineers. For a discrete-time, real-valued stationary stochastic process $\{X_n\}_{n=0}^{\infty}$, the autocorrelation function $\rho_x(\tau)$ consists of the lagged correlation coefficients

$$\rho_x(\tau) = \frac{E[X_n X_{n+\tau}] - \mu_X^2}{\sigma_X^2} \quad \tau = 1, 2, \dots \quad (1.1)$$

where $\mu_X < \infty$ and $\sigma_X^2 < \infty$ are the mean and variance, respectively, of the X_n . Burstiness which more commonly is caused by the presence of significant positive autocorrelations in the interarrival process can make waiting times arbitrarily high, without increasing the arrival rate. Various studies [4] have shown that when auto-correlated traffic is offered to a queueing system, the resulting performance measures are considerably worse than those corresponding to renewal traffic. In fact, mean waiting times can differ by orders of magnitude.

As a solution to this type of problem, we describe and investigate the use of the Quantized Transform-Expand -Sample (QTES) methodology to capture both the first and second order statistics of empirical time series simultaneously and accurately. The criteria for a good traffic model should be the following [5]:

- The marginal distribution of the model should match its empirical counterpart
- The autocorrelation function of the model should approximate its empirical counterpart. Since the empirical data is finite, the model needs to approximate the significant leading autocorrelations.
- Sample paths generated by a simulation of the model should “resemble” the empirical data.

We should mention that the first two requirements constitute quantitative criteria, requiring that the first-order and second-order properties of the empirical data are adequately captured. The third requirement though, is a qualitative requirement which can not be defined with mathematical accuracy. It is mainly a heuristic procedure and if a model can imitate the qualitative character of the empirical data, so much the better, as our confidence in the model is increased. It is important to realize that the qualitative similarity should not substitute for the first two requirements which approximate the statistical signature of the empirical data [5].

1.2 Previous Research

There are many traffic models which are commonly used in traffic modeling [1]. Such models are used either as a part of an analytical model, or to drive discrete-event simulations. Among the quantities that are of particular interest to the modeler, interarrival times, batch sizes and work-load processes, in either continuous or discrete (slotted) time are the most important.

- *Renewal Traffic Models* Renewal models have been used a lot for traffic modeling, because of their mathematical simplicity. In a renewal traffic process, the interarrival times $\{A_n\}$ are independent, identically distributed (IID), but their distribution is allowed to be general. Renewal processes, while mathematically simple, they have a major disadvantage: the autocorrelation function of $\{A_n\}$ vanishes identically for all nonzero lags. This poses a serious problem to the performance evaluation of broadband networks which are mainly dominated by bursty traffic. When this traffic is offered to a queueing system, it gives rise to much worse performance (such as mean waiting times) as compared to renewal traffic which lacks temporal dependence [4]. Poisson processes are the oldest renewal processes used in traffic modeling.

Poisson models are the oldest traffic models, dating back to the advent of telephony and the renowned pioneering telephone engineer A. K. Erlang. A Poisson process can be characterized as a renewal process whose interarrival times $\{A_n\}$ are exponentially distributed with rate parameter λ : $P\{A_n \leq t\} = 1 - \exp(-\lambda t)$ [6]. Equivalently, it is a counting process, satisfying $P\{N(t) = n\} = \exp(-\lambda t)(\lambda t)^n/n!$, and the number of arrivals in disjoint intervals is statistically independent (a property known as independent increments).

Poisson processes enjoy some elegant analytical properties. First, the superposition of independent Poisson processes results in a new Poisson process whose rate is the sum of the component rates. Second, the independent increments property renders Poisson a memoryless process. This, in turn, greatly simplifies queueing problems involving Poisson arrivals. Third, Poisson processes are

fairly common in traffic applications that physically comprise a large number of independent traffic streams, each of which may be quite general.

Time-dependent Poisson processes are defined by letting the rate parameter λ depend on time.

The discrete-time analog of Poisson processes are Bernoulli models. Here the probability of an arrival in any time slot is p , independent of any other one. It follows that for slot k , the corresponding number of arrivals is binomial. The time between arrivals is geometric with parameter p .

- *Markov Traffic Models* Markov processes are more general than renewal processes, since they introduce dependence into the random sequence $\{A_n\}$ [6]. Despite the fact that Markov processes may have different parameters, the interarrival times are still exponentially distributed. Due to the memoryless property of the exponential distribution, Markov traffic models have good analytical properties and they are widely known. In order to build a Markov process though, we need to know the transition matrix which is very difficult to get when too many states are assumed. A way to reduce the complexity of Markov processes, while maintaining their flexibility, is to integrate Markov processes with few states along with other processes to form a new process. Markov-Modulated Poisson Processes (MMPP) constitute an extremely important class of such traffic models. These models have been widely used to model voice traffic sources [7].
- *Autoregressive Traffic Models* In autoregressive models the next random variable is defined in the sequence as a function of previous ones within a time window

stretching from the present into the past. Such models are particularly suitable for modeling VBR-coded video, a major consumer of bandwidth in today's high-speed networks. The nature of video frames is such that successive frames within a video scene vary visually very little (there are 30 frames per second in a high quality video). Only scene changes can cause abrupt changes in frame bit rate. Thus the sequence of bit rates (frame sizes) may be modeled by an autoregressive scheme, while scene changes can be modeled by some modulating mechanism, such as a Markov chain. A linear autoregressive model has been used to model variable bit rate (VBR) coded video [8]. There are many different kinds of autoregressive traffic models such as the linear autoregressive models, MA, ARMA, and ARIMA which are outside the scope of this thesis. One type of the autoregressive traffic models though is of particular interest to us, since the whole modeling methodology that we will use in this paper is based on these traffic models: Transform-Expand-Sample models. These models will be described in detail in the sequel.

1.3 Motivation and Objectives of this Thesis

This thesis presents a consolidated study that is based on the generation of traffic models from empirical traffic streams which go through high speed networks. The aim is to produce traffic models in an automated way using the QTES methodology [9] which will capture both the autocorrelation structure and the marginal distribution of the empirical traffic streams simultaneously.

The main objective of this thesis is to develop an algorithmic procedure which will automate traffic modeling making use of the QTES modeling methodology. The

algorithm will use a nonlinear programming setting with the objective of minimizing a weighted sum of squared differences between the empirical autocorrelations and their candidate QTES model counterparts.

QTES processes is a class of stochastic processes which can be used to model a large class of stochastic processes in an accurate way. The QTES modelling methodology approximates a continuous-state TES process by a discrete-state QTES counterpart. TES is a class of models [5, 11, 12, 13] which can accurately model empirical sequences, and in particular, can effectively capture traffic burstiness [10]. The TES modelling methodology (to be reviewed in chapter 2) can accurately capture both first-order and second-order statistics of empirical data: more specifically, the model exactly captures the empirical marginal distribution (histogram) and approximately captures the empirical autocorrelation function, simultaneously. One problem which is associated with the use of TES processes in traffic modelling is that the TES-based queueing models have been analytically intractable, mainly due to the fact that TES processes are essentially transformed Markov sequences, over an uncountable state space. In this thesis, we use another class of stochastic processes, the QTES processes (Quantized TES) which approximates a continuous-state TES process by a discrete-state variant QTES counterpart. This way we can have a tractable queueing model which is still quite accurate in the sense that the traffic model captures first-order and second order statistics of the empirical traffic data. A detailed analysis of both TES and QTES processes is presented in chapter 2.

The specific objectives of this thesis are to :

1. Develop an algorithmic approach for capturing the correlation structure of the empirical traffic using the Exhaustive Search Local Optimization algorithm. As

the name suggests this algorithm combines an exhaustive search with a local nonlinear programming technique to minimize an objective function consisting of the distance between the empirical autocorrelation function and its candidate model counterpart.

2. Validate the efficiency of the Exhaustive Search Local Optimization algorithm by results using MPEG-compressed VBR video data streams. Compare the accuracy of the approximation of the empirical autocorrelation structure with its QTES counterpart for different quantization levels.
3. Modify the automated algorithm by replacing the exhaustive search with a random search.
4. Validate the efficiency of the algorithm, when a random search is used by results using the same MPEG-compressed VBR video data streams as in the case of the exhaustive search.
5. Compare the accuracy of the approximation of the empirical autocorrelation structure with its QTES counterpart between the random and the exhaustive search.
6. Develop a QTES traffic generator which will accept empirical traffic data as an input and it will return a traffic model which captures both the autocorrelation structure and the marginal distribution of the empirical data. The parameters which ensure the match in the autocorrelation structure between the empirical data stream and the model are calculated by the automated algorithm which was described in a nutshell above. The match in the marginal distribution is

achieved by developing an algorithm which makes use of the Inversion Transform method.

7. Validate the accuracy of the QTES generator by results using the same MPEG-compressed VBR video data streams as above.

1.4 Thesis Organization

The remaining chapters of the thesis are organized as follows:

Chapter 2 : Reviews the fundamentals of the Transform Expand Models (TES, DTES, QTES), their mathematical properties, characteristics, reasons for creating the DTES and QTES processes from the TES processes.

Chapter 3: The optimization problem is formulated by introducing the objective function consisting of the distance between the empirical autocorrelation function and its candidate model counterpart. The optimization parameters and the parametric space are defined. This chapter presents the Exhaustive Search Local Optimization algorithm. The implementation of the exhaustive search and the local optimization are explained in a detailed manner starting from the mathematical prerequisites and going further into investigating the implementation from a programming point of view. The chapter ends with validating the efficiency of the Exhaustive Search Local Optimization algorithm by using three examples from the domain of the compressed VBR video traffic.

Chapter 4: Replaces the exhaustive search with a random search over the parametric space. The chapter explains first the mathematical background that is needed to develop a random search algorithm and then it proceeds with explaining in

great detail the implementation of the random search algorithm from the programming point of view. Finally the efficiency of this approach is validated by using the same three VBR video data streams as in the previous chapter.

Chapter 5: Compares the efficiency of the exhaustive and the random search with respect to the approximation of the empirical autocorrelation with its QTES counterpart. This chapter investigates the risks which are associated with the random sampling procedure and shows through an example that by increasing the number of the randomly chosen points, we have better chances of getting “good” initial points. The price we pay for this increase is that the random search takes much longer.

Chapter 6: Implements a QTES generator whose purpose is to generate sample path realizations which capture the first order statistics (marginal probability distribution) and the second order statistics (autocorrelation function) of a given empirical data sample. The QTES generator is presented as a foreground/background scheme. This chapter explains in detail how an exact match to the empirical histogram is achieved by the use of the Inverse Transform method. Finally the accuracy of the QTES generator is validated by using the same three examples from the domain of the compressed VBR video traffic.

Chapter 7: Conclusions and Recommendations for further study.

Chapter 2

Transform Expand Sample Models

2.1 TES Processes

This subsection provides a brief overview of TES processes and the TES modeling methodology. For detailed discussion see [11, 12, 13]. TES modeling constitutes a different input analysis approach whose principal merit is its potential to capture both the empirical marginal distribution (histogram) and to simultaneously approximate the leading empirical autocorrelations via a stationary stochastic process.

The construction of TES models involves two random sequences in lockstep. The first sequence, called a *background* TES process, plays an auxiliary role, and is defined either as $\{U_n^+\}_{n=0}^\infty$ or $\{U_n^-\}_{n=0}^\infty$; these have the form

$$U_n^+ = \begin{cases} U_0 & \text{if } n = 0 \\ \langle U_{n-1}^+ + V_n \rangle & \text{if } n > 0 \end{cases} \quad (2.1)$$

and

$$U_n^- = \begin{cases} U_n^+ & \text{if } n \text{ even} \\ 1 - U_n^+ & \text{if } n \text{ odd} \end{cases} \quad (2.2)$$

respectively, where U_0 is distributed uniformly on $[0,1)$, $\{V_n\}_{n=1}^{\infty}$ is a sequence of iid random variables, independent of U_0 , called the *innovation sequence*, and the angular brackets denote the modulo-1 (fractional part) operator $\langle x \rangle = x - \max \{ \text{integer } n: n \leq x \}$. The superscript notation in Eq 2.1 and 2.2 is due to the fact that $\{U_n^+\}_{n=0}^{\infty}$ and $\{U_n^-\}_{n=0}^{\infty}$ can generate lag-1 autocorrelations in the range $[0,1]$ and $[-1,0]$, respectively. For example, the innovation density function illustrated in figure innovation can be viewed causing the background sequence U_n^+ to migrate slowly and consistently around the circle in a clockwise direction (figure 2.2). A background sequence of this type will exhibit strong positive autocorrelations over several lags. as well, a periodic component can be introduced into the background sequence autocorrelation by shifting the density function in figure 2.1 away from the y-axis. Figure 2.3 illustrates a different innovation density function, such that the value of V_n varies uniformly on $[-0.5, 0.5]$. In this case, the corresponding background sequence moves arbitrarily on the circle and is therefore uncorrelated.

The second sequence, called a *foreground TES Process* and denoted by $\{X_n^+\}_{n=0}^{\infty}$ or $\{X_n^-\}_{n=0}^{\infty}$, respectively, is a transformed version of a background TES process,

$$X_n^+ = D(U_n^+), \quad X_n^- = D(U_n^-) \quad (2.3)$$

where D is a measurable transformation from $[0,1)$ to the reals, called a *distortion*. Eq 2.3 defines two classes of TES models, denoted TES^+ and TES^- , respectively, and these foreground sequences are the end-product TES models.

An important family of distortions is based on an empirical time series, $\{Y_n\}_{n=0}^N$; these distortions consist of composite transformations of the form

Figure 2.1: Probability Density of Innovation Sequence V_n

Figure 2.2: Migration of correlated background sequence on the unit circle

Figure 2.3: Probability density of Innovation Sequence V_m

$$D_{Y,\xi}(x) = \hat{H}_Y^{-1}(S_\xi(x)), \quad x \in [0, 1]. \quad (2.4)$$

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

$$U_n^- = \begin{cases} y/\xi, & 0 \leq y \leq \xi \\ (1-y)/(1-\xi), & \xi \leq y \leq 1 \end{cases} \quad (2.5)$$

The outer transformation $\hat{H}_Y^{-1}$, is the inverse of the empirical (histogram) distribution function computed from an empirical time series, $Y = \{Y_n\}$, as

$$\hat{H}_Y^{-1}(x) = \sum_{j=1}^J 1_{[\hat{C}_{j-1}, \hat{C}_j]}(x) [l_j + (x - \hat{C}_{j-1}) \omega_j / \hat{p}_j], \quad 0 \leq x \leq 1, \quad (2.6)$$

where J is the number of histogram cells, $[l_j, r_j)$ is the support of cell j with width $\omega_j = r_j - l_j \geq 0$, $\hat{p}_j$ is the probability estimator of cell j and $\{\hat{C}_i\}_{i=0}^J$ is the cdf of

$\{p_j\}_{j=1}^J$.

The rationale for TES processes stems from the following facts: First, all background TES sequences are stationary Markovian, and their marginal distribution is uniform on $[0,1)$, regardless of the probability law of the innovations $\{V_n\}$ selected. It can be shown that the τ -step transition density of $\{U_n^+\}$ is

$$g_\tau^+(y|x) = \begin{cases} \sum_{\nu=-\infty}^{\infty} \tilde{f}_V^\tau(i2\pi\nu) e^{2i\pi\nu(y-x)}, & 0 \leq y, x < 1 \\ 0, & \text{otherwise} \end{cases} \quad (2.7)$$

and that of $\{U_n^-\}$ is

$$g_\tau^-(y|x) = \begin{cases} g_\tau^+(y|x) & 0 \leq y, x < 1, \tau \text{ even} \\ \sum_{\nu=-\infty}^{\infty} \tilde{f}_V^\tau(i2\pi\nu) e^{-2i\pi\nu(y-x)}, & 0 \leq y, x < 1, \tau \text{ odd} \\ 0, & \text{otherwise} \end{cases} \quad (2.8)$$

The tilde denotes the Laplace transform. Second the *inversion method* enables us to transform any uniform variable to one with an arbitrary prescribed distribution as follows: If F is of any distribution function and U is uniform in $(0,1)$, then the random variable $X = F^{-1}(U)$ satisfies $X \sim F$. For detailed discussion see [2]. And third, for $0 < \xi < 1$, the sample paths of background TES sequences are forced to be more “continuous looking” due to the S_ξ transformation. The Inversion method can still be applied to stitched background processes $\{S_\xi(U_n)\}$ since stitching transformations preserve uniformity [11]. Therefore, any foreground TES sequence of the form

$$X_n = \hat{H}_Y^{-1}(S_\xi(U_n)) \quad (2.9)$$

is always guaranteed to have the empirical distribution $\hat{H}_Y$, regardless of the innovation density f_v and the stitching parameter ξ selected. The choice of a pair (f_v, ξ)

will determine the autocorrelation function of 2.9. Therefore, TES modeling manages to capture both the marginal distribution and the autocorrelation function of the empirical data simultaneously.

2.2 Discretized and Quantized TES Processes

Despite the fact that TES-based models have proven to be accurate in a variety of application domains [14], TES based queueing models, however, have been analytically intractable. Queueing models with TES-based traffic have only been studied via Monte Carlo simulation. The analysis of such models is difficult, due to the fact that TES processes are transformed Markov sequences, over an uncountable state space. The reason, therefore, for introducing discretized and quantized TES processes is to reduce the continuous state space of background TES processes (the unit circle) to a finite state space.

2.2.1 Discretized TES Processes (DTES)

A DTES process is a discrete variant of a TES process over a finite state space. This is achieved by partitioning the unit circle into a finite number of subintervals (called slices), and “collapsing” each slice subinterval into a new state. Intuitively, if the slice subintervals are “small enough”, then the exact location of the original process within a slice is “well approximated” by the slice index. Let $Z_M = \{0, 1, \dots, M-1\}$, for some integer $M > 0$. A quantization of the unit circle, $[0,1)$, is a partition,

$$\Gamma(M) = \{\Gamma_m = [m\delta, (m+1)\delta] : m \in Z_M\}. \quad (2.10)$$

of the interval $[0,1)$ into M subintervals (slices). Γ_m , each of length $\delta = \delta_M = 1/M$. The integer-valued modulo- M operator from the integers onto Z_M is denoted by

$$\langle m \rangle_M = m - M \lfloor m/M \rfloor \quad (2.11)$$

where $\lfloor x \rfloor = \max\{n: n \text{ is an integer and } n \leq x\}$.

Let C_0 be an integer-valued random variable, distributed uniformly over Z_M . Let, further, $\{J_n\}_{n=1}^\infty$ be an iid sequence of integer-valued random variables, independent of C_0 , called the *innovation sequence*, analogously to the setting of TES processes. The probability distribution of the J_n is denoted by

$$f_J(j) = P\{J_n = j\} = \phi_j \quad j \in Z_M \quad (2.12)$$

The background DTES processes are defined by

$$C_n^+ = \begin{cases} C_0 & \text{if } n = 0 \\ \langle C_{n-1} + J_n \rangle_M & \text{if } n > 0 \end{cases} \quad (2.13)$$

and

$$C_n^- = \begin{cases} C_0^+ & \text{if } n \text{ even} \\ M - 1 - C_n^+ & \text{if } n \text{ odd} \end{cases} \quad (2.14)$$

Like TES background processes, the DTES background processes are also stationary and their marginal distributions are all uniform on Z_M . Their τ -step transition probabilities are given by

$$h_\tau^+(k|j) = 1/M \sum_{\nu=0}^{M-1} \tilde{f}_j^\tau(i2\pi\nu) e^{2i\pi\nu(k-j)/M}, \quad (2.15)$$

and

$$h_{\tau}^{-}(k|j) = \begin{cases} h_{\tau}^{+}(k|j) & \tau \text{ even} \\ 1/M \sum_{\nu=0}^{M-1} \tilde{f}_{\tau}^{\tau}(i2\pi\nu) e^{-2\pi\nu(k+j+1)/M}, & \tau \text{ odd} \end{cases} \quad (2.16)$$

where $\tilde{f}_{\tau}^{\tau}(i2\pi\nu)$ is the discrete Fourier transform of the discrete probability distribution function, f_J for $\nu = 0, 1, \dots, K-1$. For more details, refer to [11].

2.2.2 Quantized TES Processes (QTES)

Quantized TES processes (referred to as QTES processes) are obtained from DTES processes by extending the latter to the unit circle, in such a way that the corresponding transition density has a quantized domain of conditioning.

We have mentioned in the previous section that the background DTES processes, $\{C_n\}$ are marginally uniform on Z_M , i.e.,

$$P\{C_n = m\} = 1/M = \delta, \quad m \in Z_M. \quad (2.17)$$

The corresponding QTES process, $\{Q_n\}$, is a continuous extension of $\{C_n\}$, defined by

$$\{Q_n\} = \delta[C_n + W_n], \quad (2.18)$$

where the sequence $\{W_n\}$ is iid with uniform marginals on $[0,1]$. In other words, Q_n is conditionally uniform on Γ_m , given that $C_n = m$. From 2.17 and 2.18, it can be concluded that any QTES background sequence is uniform in $[0,1]$. The τ -step transition density of $\{Q_n\}$ is given by

$$q_{\tau}(y|x) = h_{\tau}(I_{\Gamma}(y)|I_{\Gamma}(x))M, \quad (2.19)$$

where $I_\Gamma(z)$ is the unique slice index, $k = k(z)$, in the partition, Γ , such that $z \in \Gamma_k$. Therefore, $q_\tau(y|x)$ is only a function of the partition Γ , in the sense that it does not depend on x and y , but only on the partition indexes indices in which they lie. In other words, q_τ is constant in each variable over partition slices, implying that it maps a continuous domain to a finite range. As a result, g_τ only requires a finite number of values for its representation, while in a TES background process the transition function g_τ requires two continuous variables. It is this reduction in complexity that motivated the QTES processes. As it can be seen by 2.7 and 2.15 and from 2.8 and 2.16,

$$\lim_{\varepsilon \rightarrow 0} q_\tau^+ = g_\tau^+, \quad \lim_{\delta \rightarrow 0} q_\tau^- = g_\tau^- \quad (2.20)$$

Let D be a distortion, and let $\{Z_n\}$ be the corresponding foreground sequence, with elements $Z_n = D(Q_n)$. Thus, the QTES foreground sequence, $\{Z_n\}$, has the same marginal distribution, $\hat{H}_Y$, as the TES foreground sequence $\{X_n\}$, with $X_n = \hat{H}_Y^{-1}(S_\xi(U_n))$.

Let further $\overline{D}_m$ be the partial mean of Z_n on each slice Γ_m , namely,

$$\overline{D}_m = \frac{1}{\delta} \int_{m\varepsilon}^{(m+1)\varepsilon} D(x) dx \quad (2.21)$$

and notice that

$$\mu_Z = \int_0^1 D(x) dx = \hat{D}(0) = \sum_{\nu=0}^{M-1} \overline{D}_j \quad (2.22)$$

The autocorrelation function $\rho_Z(\tau)$ of Z_n for $\tau = 0, 1, \dots$, is given by

$$\rho_Z(\tau) = \frac{\delta^2}{\sigma_Z^2} \sum_{\nu=1}^{M-1} \tilde{f}_J^*(i2\pi\nu) \|\tilde{D}(i2\pi\nu)\|^2. \quad (2.23)$$

and

$$\rho_{\tilde{Z}}(\tau) = \begin{cases} \rho_{\tilde{Z}}^*(\tau) & \tau \text{ even} \\ \frac{\delta^2}{\sigma_{\tilde{Z}}^2} \sum_{\nu=1}^{M-1} \tilde{f}_J^*(i2\pi\nu) \tilde{D}^2(i2\pi\nu) e^{-i2\pi\nu\tau} & \tau \text{ odd} \end{cases} \quad (2.24)$$

2.3 Summary

The TES modeling methodology has an attractive property of decoupling the distribution of the foreground process from the autocorrelation function. Because the distribution of X_n is guaranteed to match the empirical distribution, the model needs only to be optimized to fit the empirical autocorrelation function. Fitting the TES model to the empirical autocorrelation is accomplished by varying the innovation density, f_t , and the stitching parameter, ξ .

Although TES-based models have proven to be accurate in a variety of application domains, TES-based queueing models, however, have been analytically intractable, mainly because TES processes are transformed Markov sequences, over an uncountable state space. This is the reason for introducing the quantized TES processes (DTES). By using DTES processes, we manage to reduce the continuous state space of background TES processes to a finite state space. Finally, we obtain quantized TES processes (QTES) by extending DTES processes to the unit circle, in such a way that the corresponding transition density has a quantized domain of conditioning.

Chapter 3

Automated Modeling of Broadband Network Data Using the QTES Methodology

Despite the fact that TES modeling has been proven effective in many application domains, it has several disadvantages. Firstly, someone has to understand in depth the qualitative nature of TES processes in order to use the TES modeling methodology effectively and get accurate results. Secondly, the scope and the speed of the search are fundamentally subject to the human response and thirdly, modeling precision is constrained by screen resolution as perceived by human eye. By trying to shift the modeling burden from the modeler to the computer, we can largely automate the modeling process and alleviate the above problems. As we have already mentioned in section 2.2, TES based queueing models have been analytically intractable, due to the fact that TES processes are transformed Markov processes, over an uncountable state space. Therefore we are going to use QTES processes in this algorithmic modeling, achieving in that way a reduction in the complexity of the algorithm. An algorithmic procedure using TES processes is discussed in [15].

Our modeling approach combines a global search with a local nonlinear technique to minimize the objective function consisting of the distance between the empirical autocorrelation function and its candidate model counterpart. By distance, we mean the weighted sum of squared differences between autocorrelation coefficients of corresponding lags. This approach was mainly motivated by the fact that there are fast and numerically stable analytical formulas for the calculation of the objective function and its partial derivatives. More than that, the constraints in the local nonlinear optimization problem are very simple.

3.1 Problem Formulation

As we have already mentioned, QTES modeling methodology tries to produce accurate traffic models from empirical data by capturing first-order and second-order statistics of empirical time series simultaneously. The first order parameters are represented by the distortions $D_{Y,\xi}$ which are given by equation 2.4, while the second order parameters are represented by the pairs (f_v, ξ) of step-function innovation densities and stitching parameters.

Since an exact match to the empirical histogram is guaranteed by using the inversion method, the problem reduces to one of approximating the empirical autocorrelation function, $\hat{\rho}_Y$, by some QTES model autocorrelation function, $\rho_{f_v, \xi}$, which will be determined by the choice of (f_v, ξ) . f_v is the discrete innovation used in the QTES model and ξ is the stitching factor.

We have discrete innovation densities of the form $f_V = \mathbf{P} = \{P_1, P_2, \dots, P_k\}$, with the restrictions

$$\sum_{n=1}^{n=k} P_n = 1 \quad \text{and} \quad 0 \leq P_n \leq 1 \quad (3.1)$$

and the stitching factor. ξ lies in the interval $[0,1]$. Therefore, we can define the parameter space as

$$G_K = \{(\mathbf{P}, \xi) : \mathbf{P} = (P_1, \dots, P_k) \quad \xi \in [0,1]\} \quad (3.2)$$

Now, we need a metric on the space of autocorrelation functions. The metric which will be used is

$$g(f_v, \xi) = \sum_{\tau=1}^T \alpha_{\tau} [\rho_{f_v, \xi}(\tau) - \rho_Y(\tau)]^2 \quad (3.3)$$

where T is the maximal autocorrelation lag to be approximated. The metric should also take into consideration the fact that it is more important to approximate the lower-lag autocorrelations than the higher-lag ones. The use of the weight sum of the squared differences between the empirical and modeled autocorrelations in 3.3 is due to this reason. In the above metric (3.3), the $0 < \alpha_{\tau} \leq 1$ are weight coefficients.

Our task now is to search for a pair (P, ξ) , which minimizes the autocorrelation distance in 3.3. Formally, we seek to solve the following nonlinear optimization problem:

For a fixed inverse histogram distribution, find an optimal DTES innovation density and stitching parameter, (f^*_v, ξ^*) , such that

$$(f^*_v, \xi^*) = \operatorname{argmin}\{g(f_v, \xi)\} \quad (3.4)$$

where $g(f_v, \xi)$ is given in equation 3.3.

As we can observe from the above analysis, the optimization problem that we have to tackle has a finite dimensionality due to the use of the QTES proceses. it is subject to simple linear constraints and its objective function has an analytical. closed formula as it can be seen from equations 2.23 and 2.24.

3.2 Algorithmic Approach

The algorithmic approach which will be used for the solution of the optimization problem consists of two parts.

Exhaustive Search: We select the B most promising points of the parametric space G_K . In order to do that, we discretize the parameter space into a finite number of points K, evaluate the objective function g_K at those K points and keep those points which give the B smallest values of g_K .

Local Search: We use each of these B points as initial points to find a minimum of the objective function g_K . Then, we select among them that point x^* that gives the smallest value for g_K .

By using the exhaustive search as the first step for the optimization procedure, we get rid of all the obviously “bad” points right from the beginning, while we increase the chance that the best local minimum is close to the global minimum.

3.3 Implementation of the Exhaustive Search

As we have already pointed out, the Exhaustive Search selects the B most promising initial points for the local search. This means that the algorithm finds the B points $(\mathbf{P}, \xi)$ which give the smallest value of the objective function. In order to simplify

the notation. we write $(P_1, \dots, P_K, \xi)$ rather than $((P_1, \dots, P_K), \xi)$, interchangeably with $(\mathbf{P}, \xi)$. Therefore, what we have to do first, is to find a way to produce all the possible combinations of $(P_1, \dots, P_K, \xi)$ where $P_1 + \dots + P_K = 1$, $0 \leq P_n \leq 1$ and $\xi \in [0, 1]$.

In addition to K and B , the exhaustive search algorithm requires two additional parameters. N_P and N_ξ . These are the number of equidistant values that each P_n and ξ can assume respectively. For example, if we have a $N_P = 4$, it means that each one of the $P_1, \dots, P_K$ can be equal to 0, 0.25, 0.50, 0.75 or 1. There is of course the restriction that $P_1 + \dots + P_K = 1$. The same rule applies to ξ also. If, for example, we have $N_\xi = 10$, it means that the allowable range for the smoothing transform is $[0, 1]$ in increments of 0.1. The reason for establishing these specifications is obvious: First, we decrease the number of searches for both the discrete innovation and the stitching factor and second, given that we have an initial “good” innovation and a stitching factor, we get more exact values in the local optimization step of the algorithm.

We used two routines in order to implement the Global Search Algorithm: the *next composition* and the *Global Search*. Both of them are explained in the sequel.

3.3.1 Next Composition

We observed that the way that each initial discrete innovation is formulated is the same as the composition of n into k parts. Let n and k be fixed positive integers. By a *composition* of n into k parts, we mean a representation of the form

$$n = r_1 + r_2 + r_3 + \dots + r_k \quad (3.5)$$

in which $r_i \geq 0$ ($i = 1, \dots, k$) and the order of summands is important. By using the same idea in our case, we can say that if we have a discretization level K and N_P is

the number of the equidistant values that each one of the $P_1, \dots, P_K$ can take, then this is the same as if we have to accomodate N_P balls in K cells. In [16] it is proven that there are

$$\binom{n+k-1}{n} \quad (3.6)$$

ways of arranging n balls in k cells.

The algorithm for generating compositions of n into k parts is an extension of another algorithm that we made which returns all the combinations of n things taken k at a time. Therefore, we will explain this algorithm first.

In a lexicographic sequence, we can obtain the successor of a given k -subset $\{a(1), \dots, a(k)\}$ as follows: search for the smallest h such that $a(k+1-h) < n+1-h$, then increase $a(k+1-h)$ by 1 and set $a(j) \leftarrow a(j-1) + 1$ ($j = k+2-h, k$).

In figure 3.1, we can see how we obtain a successor of a given k -subset of an n -Set. It is interesting to notice that the index h can be found without searching, since at each transition from a subset to its successor, h increases by 1 unless $a(k+1-h) < n-h$, where h is reset to 1 on the next transition.

We can generate compositions of n into k parts by going back to our lexicographic algorithm and translating it into a direct algorithm for compositions. Recall that in that algorithm, if $\{a(1), \dots, a(k)\}$ is a $(k-1)$ subset, we go to the next one by finding the smallest h for which

$$a(k+1) = n, a(k-2) = n-1, \dots, a(k-h) = n-h+1: a(k-h-1) < n-h \quad (3.7)$$

Figure 3.1: Flowchart for the Next k -Subset of an n -Set routine

We then increase $a(k-h-1)$ by 1 and set each succeeding $a(r+1)$ equal to one more than its predecessor $a(r)$.

As far as the composition which is associated with the subset (3.8) is concerned.

$$n = r(1) + r(2) + \dots + r(k) \quad (3.8)$$

the relations (3.7) tells us that $r(k) = r(k+1) = \dots = r(k+1-h) = 0$ and $r(k-h) > 0$. By increasing $a(k-h-1)$ by 1 and setting each following $a(r)$ equal to one more than its predecessor, will increase $r(k-h-1)$ by 1. set $r(k) = r(k-h) - 1$ and set $r(k-h) = 0$.

The notion of subsets can then be eliminated and the algorithm can be stated directly in terms of compositions. When all of this is done, what remains is to search the last composition $r_1, \dots, r_k$ to find the first nonzero part r_h . We then put $t \leftarrow r_h$, $r_h \leftarrow 0$, $r_1 \leftarrow t - 1$, $r_{h+1} \leftarrow r_{h+1} + 1$

In figure 3.2, we can see the flowchart for the complete algorithm. If we divide all the elements of each different combination with the discretization level K , we can have the DTES innovation.

3.3.2 Global Search

The objective function is calculated for each one of the B initial DTES innovations and stitching factors ξ . The computed value is then compared with the current best set, namely the running set of best (at most B) values in a 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 sorted order. We continue this way until we search the whole discretized space.

Figure 3.2: Flowchart for the Next Composition routine

3.4 Local Search

The local optimization of the objective function in 3.4 can be solved by many standard nonlinear programming techniques. We implemented it using the sequential quadratic programming technique (SQP) which is considered state-of-the-art in nonlinear constrained optimization methods [17, 18, 19].

3.4.1 Constraint Optimization

The optimization problem that we have can be described as a constraint problem as follows:

$$\begin{aligned}
 & \text{minimize } g(\mathbf{x}), \quad \mathbf{x} \in R^{K+1} \\
 & \text{subject to : } r_1(\mathbf{x}) = 0. \\
 & r_i(\mathbf{x}) \leq 0, \quad i = 2, \dots, 2K + 3
 \end{aligned} \tag{3.9}$$

where $\mathbf{x}$ is the vector of the parameters which will be optimized, g is the objective function in eq.3.3, $(g : R^{K+1} \rightarrow R)$, $\mathbf{r}$ is the vector of quality and equality constraints (lower-case boldface letters indicate vectors and capital-case boldface lettere indicate matrices).

In constraint optimization, we want to transform the problem into an easier subproblem which can be solved and used as the basis of an iterative process. Most of the methods that are used today in nonlinear optimization depend heavily on the solution of the Kuhn-Tucker (KT) equations. The KT equations are necessary conditions for the optimality of a constrained optimization problem [20, 21].

The Kuhn-Tucker equations can be described as follows:

$$\begin{aligned} \nabla g(\mathbf{x}^*) + \sum_{i=1}^{2K+1} \lambda_i^* \nabla r_i(\mathbf{x}^*) &= 0 \\ \lambda_1^* r_1(\mathbf{x}^*) &= 0 \\ \lambda_i^* &\geq 0 \quad i = 2, \dots, 2K+3 \end{aligned} \tag{3.10}$$

The first equation indicates the fact that at the solution point the gradients between the objective function and the active constraints are cancelled. The Langranian multipliers serve this purpose. i.e. they balance the deviations in magnitude of the objective function and the gradients of the constraints. Non-active constraints have Langranian multipliers equal to zero, since they do not contribute to the canceling process. This is stated in the last two equations of Eq. 3.10. A more detailed analysis of the constraint optimization and the Kuhn-Tucker equations can be found in [20, 21].

Many algorithms are based on the direct computation of the Langranian multipliers. There are some more efficient methods however which make use of second order information, as far as the KT equation is concerned using updating procedures. These methods are known as Sequential Quadratic Programming (SQP) methods, because a quadratic subproblem (QP) is solved at each iteration.

3.4.2 Sequential Quadratic Programming (SQP)

In SQP methods, at each iteration, we approximate the Hessian of the Langranian function

$$L(\mathbf{x}, \lambda) = g(\mathbf{x}) + \sum_{i=1}^{2K+1} \lambda_i r_i(\mathbf{x}) \tag{3.11}$$

using a quasi-Newton updating procedure. Then, we use this approximation to find a feasible direction $\mathbf{d} = (d_1, \dots, d_K, d_{K+1})$ through a quadratic problem and finally,

once a feasible direction has been obtained, we apply a line search procedure to find the optimal feasible step size. A more detailed description of SQP can be found in [17, 18, 19].

Given the problem in eq. 3.4, the SQP implementation consists of three steps:

1) Update the Hessian Matrix

At each iteration, $\mathbf{H}$, a positive-definite quasi-Newton approximation of the Hessian of the Langranian function, is calculated using the Broyden [22], Fletcher [23], Goldfarb [24], Shanno [25] (BFGS) formula which is given by

$$\begin{aligned} \mathbf{H}_{l+1} &= \mathbf{H}_l + \frac{\mathbf{q}_l \mathbf{q}_l^T}{\mathbf{q}_l^T \mathbf{s}_l} - \frac{\mathbf{H}_l^T \mathbf{H}_l}{\mathbf{s}_l^T \mathbf{H}_l \mathbf{s}_l} \\ \mathbf{s}_l &= \mathbf{x}_{l+1} - \mathbf{s}_l \\ \mathbf{q}_l &= \nabla g(\mathbf{x}_{l+1}) + \sum_{i=1}^{2K+1} \lambda_i \nabla r_i(\mathbf{x})_{l+1} - (\nabla g(\mathbf{x}_l) + \sum_{i=1}^{2K+1} \lambda_i \nabla r_i(\mathbf{x})_l) \end{aligned} \quad (3.12)$$

2) Quadratic Problem

At each iteration a QP problem is solved to yield a search direction in which the solution is estimated to lie.

$$\begin{aligned} \text{minimize} \quad & \frac{1}{2} \mathbf{d}^T \mathbf{H}_k \mathbf{d} + \nabla g(\mathbf{x}_l)^T \mathbf{d} \\ & \nabla r_1(\mathbf{x})^T \mathbf{d} + r_1(\mathbf{x}) = 0 \\ & \nabla r_i(\mathbf{x})^T \mathbf{d} + r_i(\mathbf{x}) \leq 0 \quad i = 2, \dots, 2K+1 \end{aligned} \quad (3.13)$$

The QP sub-problem can be solved using any QP algorithm.

3) Line Search

The solution of the QP sub-problem produces a vector $\mathbf{d}_l$. Now, we have to find the minimum along the line formed from this search direction. This minimum is

generally approximated using a search procedure (e.g., Fibonacci, Golden Section) or by a polynomial method involving interpolation or extrapolation (e.g., quadratic, cubic). The problem is to find a new iterate $\mathbf{x}_{l+1}$ of the form

$$\mathbf{x}_{l+1} = \mathbf{x}_l + a_l \mathbf{d}_l \quad (3.14)$$

where $\mathbf{x}_l$ is the current iterate, $\mathbf{d}$ the search direction obtained by the QP subproblem and a is a scalar step length parameter which is the distance to the minimum.

The algorithm terminates when the optimal value of the objective function falls below a prescribed threshold.

3.4.3 Partial Derivatives of the objective function

This section derives the partial derivatives of the objective function g_K in (3.3) which are used by the SQP optimization algorithm for the local search. To simplify the notation, we write $(P_1, \dots, P_K, \xi)$ rather than $((P_1, \dots, P_K), \xi)$, interchangeably with $(\mathbf{P}, \xi)$. By differentiating the objective function with respect to all optimization variables, we obtain

$$\frac{\partial g(f_v, \xi)}{\partial P_K} = \sum_{\tau=1}^T 2\alpha_{\tau} [\rho_{f_v, \xi}(\tau) - \rho_Y(\tau)] \frac{\partial \rho_{P, \xi}}{\partial P_K} \quad (3.15)$$

$$\frac{\partial g(f_v, \xi)}{\partial \xi} = \sum_{\tau=1}^T 2\alpha_{\tau} [\rho_{f_v, \xi}(\tau) - \rho_Y(\tau)] \frac{\partial \rho_{P, \xi}}{\partial \xi} \quad (3.16)$$

By differentiating the autocorrelation function in (2.23) with respect to all optimization variables, we obtain

$$\frac{\partial \rho_{(P, \xi)}}{\partial P_K} = \frac{\delta^2}{\sigma_Z^2} \sum_{\nu=1}^{M-1} \tau \tilde{f}_J^{-1}(i2\pi\nu) \|\tilde{D}(i2\pi\nu)\|^2 \frac{\partial \tilde{f}_J(i2\pi\nu)}{\partial P_K} \quad (3.17)$$

$$\frac{\partial \rho_{(P,\xi)}}{\partial \xi} = \frac{\delta^2}{\sigma_Z^2} \sum_{\nu=1}^{M-1} \tilde{f}_J(i2\pi\nu) \frac{\partial \|\tilde{D}(i2\pi\nu)\|^2}{\partial \xi} \quad (3.18)$$

As it can be seen from equations (3.17) and (3.18), we need to calculate the partial derivatives $\frac{\partial \tilde{f}_J(i2\pi\nu)}{\partial P_K}$ and $\frac{\partial \|\tilde{D}(i2\pi\nu)\|^2}{\partial \xi}$ in order to find the gradient of the objective function with respect to the optimization variables.

By using symbolic differentiation rather than a numerical one, we make the optimization algorithm faster.

In order to find the partial derivative $\frac{\partial \tilde{f}_J(i2\pi\nu)}{\partial P_K}$, we can express the DTES innovation as follows:

$$f_j(k) = \sum_{k=1}^K P_k 1_{[\frac{n-1}{K}, \frac{n}{K}]} \quad (3.19)$$

By taking the Discrete Fourier Transform of the above DTES innovation, we have

$$\tilde{f}_J(i2\pi\nu) = \sum_{k=1}^K P_k \exp \frac{-i2\pi(k-1)(\nu-1)}{K} \quad \nu = 1, 2, \dots, K \quad (3.20)$$

It can be proved by simple mathematics that the partial derivatives of the $\tilde{f}_J(i2\pi\nu)$ with respect to $P_1, P_2, \dots, P_K$ are given by the vectors:

$$\begin{aligned} \frac{\partial \tilde{f}_J(i2\pi\nu)}{\partial P_1} &= [1, 1, \dots, 1], \\ \frac{\partial \tilde{f}_J(i2\pi\nu)}{\partial P_2} &= [1, \exp \frac{-i2\pi(2-1)(2-1)}{K}, \dots, \exp \frac{-i2\pi(2-1)(K-1)}{K}], \dots, \\ \frac{\partial \tilde{f}_J(i2\pi\nu)}{\partial P_K} &= [1, \exp \frac{-i2\pi(K-1)(2-1)}{K}, \dots, \exp \frac{-i2\pi(K-1)(K-1)}{K}] \end{aligned}$$

In order to calculate the partial derivative $\frac{\partial \|\tilde{D}(i2\pi\nu)\|}{\partial \xi}$, we have to take into consideration the fact that every complex number can be expressed as a sum of real

and an imaginary part. In that respect we can express the Fourier transform of the distortion $D(i2\pi\nu)$ as follows:

$$D(i2\pi\nu) = a_{\xi,\nu} + b_{\xi,\nu} \Rightarrow$$

$$\frac{\partial \|\tilde{D}(i2\pi\nu)\|^2}{\partial \xi} = 2a_{\xi,\nu} \frac{\partial a_{\xi,\nu}}{\partial \xi} + 2b_{\xi,\nu} \frac{\partial b_{\xi,\nu}}{\partial \xi}$$

where $D(i2\pi\nu)$ is the discrete Fourier Transform the distortion. We have to recall that the distortion is a column vector whose elements are the partial means of the corresponding foreground sequence given by 2.21.

3.5 Examples and Results

This section illustrates the efficacy of the automated QTES modeling methodology via some examples from the domain of the compressed video traffic. More specifically, we used MPEG-Compressed VBR video data streams.

Video traffic is inherently variable bit-rate (VBR). The information associated with each frame is a function of the contents of the scene, the motion of objects in the frame and the motion of the camera itself. Data compression is extensively used to reduce the transmission bandwidth requirements of telecommunication traffic. Data is coded at the source, therefore it is compressed to a fraction of its original size, and then it is transported over the network. Finally it is decoded at its destination. Coding standards like H.261 and MPEG compress the raw image data of each frame by taking advantage of the correlation in the temporal as well as spatial domain of a video sequence [15]. The compression achieved for each frame is not of same magnitude and this leads to an even greater burstiness in the compressed video traffic.

Compression in MPEG (Moving Picture Expert Group) is achieved by using three frame encoding types: intraframe frame (I frame), predictive frame (P frame) and bi-directionally interpolated frames (B frames). I frames are self contained and can be decoded independently, whereas. P frames need a past I or P frame for decoding. B frames are encoded with motion vectors from a past (I or P) and a future (I or P) frame, hence both are needed for decoding. B frames are not used for prediction of other frames hence the errors in them do not propagate, as in I and P frames. This makes the B frames the least critical of the three frames. Errors in I and P frames propagate until the arrival of another I frame when the whole frame is effectively refreshed. The MPEG coding standard makes heavy use of Discrete Cosine Transform (DCT) in order to achieve the compression of video units.

We used three different video sequences as an input to our automated algorithm: BBC video. Last Action Hero video and football video whose qualitative and visual aspects are explained in the sequel.

Example 1: BBC Video

The BBC Video sample consists of 25000 frames played at 30 frames/sec showing a human figure talking into the camera. There is almost no video movement in the scene, but picture brightness is high implying high picture detail. The background consists of smoothly varying brightness and colour information without sharp visual boundaries.

The marginal distribution of the traffic stream is exponential with parameter $l = 1$. The number of equidistant values for each P_n and ξ were set to 4 and 10 respectively. The autocorrelation coefficients were calculated for 50 lags. We kept the $B = 30$ best points out of all points that we got from the exhaustive search and

we optimized the cost function only for those 30 points. We then took the point which gives the smallest value of the objective function.

Figure 3.3 displays the results of the algorithmic modelling of a P-frame empirical video sequence for a QTES⁺ model. Note the very good agreement of the QTES model autocorrelation function with its empirical counterpart. As expected, when we increase the discretization level from $K = 10$ to $K = 20$, the empirical autocorrelation is more accurately approximated by the autocorrelation of the QTES model.

Example 2: Last Action Hero Video

The Last Action Hero sample is encoded into a rather compact MPEG stream, in part due to the wide-screen format employed on the source video tape. This wide-screen format effectively uses 70% of the middle range of the picture area with the remaining 30% contributing nothing to the information content. The Last Action Hero Video consists of 25000 frames.

The number of equidistant values for each P_n and ξ were set to 4 and 10 respectively. The autocorrelation coefficients were calculated for 50 lags. We kept the $B = 30$ best points out of all points that we got from the exhaustive search and we optimized the cost function only for those 30 points. We then took the point which gives the smallest value of the objective function.

In figure 3.5, we can see how well the empirical autocorrelation of a P frame video sequence is approximated by its QTES counterpart. By a closer look, we can observe again that the approximation is better for $K = 20$ than for $K = 10$.

Example 3: Football Video

The football segment is a high action segment with very saturated monotonic areas of colour. The video areas which are subject to movement get larger as the scene progresses. and this corresponds nicely to an increasing function of frame sizes. The Football Video consists of 25000 frames.

We kept the parameters of the optimization algorithm the same as in the two previous examples, i.e. the number of equidistant values for each P_n and ξ were set to 4 and 10 respectively, the autocorrelation coefficients were calculated for 50 lags, we kept the $B = 30$ best points out of all points that we got from the exhaustive search and we optimized the cost function only for those 30 points. We then took the point which gives the smallest value of the objective function.

By comparing the accuracy of the approximation in the three above examples, we can see that in the case of the BBC and the Last Action Hero video samples the approximation of the empirical autocorrelation with its QTES counterpart is very good. In the case of the football video sample though, we notice that as we move to higher lags the approximation becomes less accurate. This happens mainly because the football video stream is a high action segment and the video areas which are subject to movement get larger as the scene progresses. This increases the burstiness of the video data stream and leads to a less accurate approximation as we move to higher lags. More than that, the presence of the exponentially distributed weight coefficients in the objective function forces the optimization to focus on the lower-lag autocorrelation coefficients.

Figure 3.3: Autocorrelation Coefficients for the BBC news empirical traffic sample and the QTES models for $K=10$ and $K=20$ using the exhaustive search

Figure 3.4: Autocorrelation Coefficients for the Last Action Hero empirical traffic sample and the QTES models for $K=10$ and $K=20$ using the exhaustive search

Figure 3.5: Autocorrelation Coefficients for the Football empirical traffic sample and the QTES models for $K=10$ and $K=20$ using the exhaustive search

3.6 Summary

The design of an automated algorithm which approximates an empirical autocorrelation as closely as possible using the QTES modeling methodology was investigated in this chapter. This was accomplished by finding an optimal innovation density, f_v and an optimal stitching parameter, ξ .

Our automated modeling approach combines an exhaustive search with a non-linear technique to minimize the objective function consisting of the distance between the empirical autocorrelation function and its candidate QTES model counterpart.

In the exhaustive search we select the B most promising points of the parametric space G_K by discretizing the parameter space into a finite number of points K , evaluating the cost function g_K at those points and keeping those points which give the B smallest values of g_K . Since the exhaustive search is of a combinatoric

nature, the exhaustive search is very time consuming.

In the Local search we use each one of the B points as initial points to find a minimum of the objective function g_K . Then we select among them the point x^* which gives the smallest value of the g_K . We implemented the local optimization by using the sequential programming technique (SQP). The choice of this method was based mainly on the following facts: there are fast and numerically stable analytical formulas for the calculation of the cost function and its partial derivatives, the constraints imposed on the objective function are linear and very simple and finally the SQP technique makes use of second order information which leads to more accurate results.

Finally, the efficiency of the automated algorithm was verified by using three empirical traffic streams from the domain of the MPEG compressed VBR video traffic. In all cases the approximation of the empirical autocorrelation with its QTES model counterpart was very accurate. As expected, it was verified that when we increase the discretization level K of the QTES model, the accuracy is better.

Chapter 4

Random Search

So far, we have used an exhaustive search in the implementation of the automated algorithm in order to find all the initial points for the local search.

In algorithms of exhaustive search type, we have often a list of combinatorial objects and we want to search the entire list, or perhaps to search sequentially until we find an object which meets certain conditions. Random sampling, on the other hand, is done when the exact determination of the quantity by exhaustive search would be so time consuming as to be impracticable.

For the exhaustive search of our optimization problem, we have used the “next composition” subroutine which, each time we call upon it, returns one object on our list. We process the object and call the subroutine again to obtain the next object. Therefore our subroutine (a) realizes, when it is being called for the first time, (b) remembers enough about its previous output so that it can construct the next member of the list, (c) realizes, at the end, that there are not objects left and (d) informs the calling program that the end has been reached.

The logic of a random sampling is much simpler. Given the input parameters, the subroutine is expected to select at random just one object from the list specified by

the input parameter. Of course, each object on the list has equal a priory probability of being selected.

Therefore, in the random search approach, we first specify how many points N of the parametric space G_K we want to have randomly chosen. we evaluate the objective function g_K at those N points and keep those which give the B smallest values of g_K

4.1 Random Search Algorithm

The random search algorithm is also based on the “next composition” idea that was described in section 3.3.1. We will elaborate a bit more on this idea, in order to understand fully, how the random search algorithm works. In section 3.3.1, we had mentioned that there are

$$J(n, k) = \binom{n+k-1}{n} \quad (4.1)$$

ways of arranging n balls in k cells.

Let $n + k + 1$ spaces be marked on a sheet of paper, and suppose that in the first space and the last space we mark a vertical bar, as shown in figure 4.1. In the remaining $n + k - 1$ spaces, distribute the n balls with no more than one ball occupying any space. There are obviously $J(n, k)$ ways of doing this. In each of the other $k-1$ spaces place a vertical bar. We now have a pattern like the one shown in figure 4.2.

Now we can think of the vertical bars as representing cell boundaries. Hence, in figure 4.2 there are 5 cells containing respectively, 2, 0, 1, 3, 1 balls. It is now clear that the number of compositions of n into k parts is given by equation 4.1.

Figure 4.1: balls in cells model

Figure 4.2: balls in cells model example

The algorithm for generating compositions of n into k parts is based on the “balls in cells” model [16] which was described above. What we must do in order to generate compositions of n into k parts is to generate $(k-1)$ subsets of $(n+k-1)$ objects and to interpret each such subset as the set of locations of the interior vertical bars in figure 4.2. From the bar locations we can, by subtraction, find the number of balls between each consecutive pair of bars and therefore determine the composition which corresponds to the given subset.

We are interested in random compositions of n into k parts. Therefore, we can choose the positions of the cell boundaries at random, then by differencing we find out how many balls are in each cell. In other words, we have to find an algorithm which can find random k -subsets of an n -set.

4.1.1 Random k -subset of an n -Set

Suppose that integers k, n are given, $1 \leq k \leq n$ and we want to select k distinct elements $a_1, a_2, \dots, a_k$ from $\{1, 2, \dots, n\}$, at random. In our case, the n set can be a very large one, since it represents the discretization level for the discrete innovation, while k is the number of equidistant probability values. We tried therefore to produce an algorithm that takes the minimum memory storage and the minimum labour. The

output set contain k words, and so it seems reasonable to impose the requirement that

(i) *no more than k words of array storage should be used by the algorithm, each word to hold an integer between 0 and n*

Next, it seems that most of the effort is needed for the selection of k integers and perhaps some removal of duplicates, and so our second requirement is

(ii) *the average labor required should be $O(k)$.*

The main problem is that we select, one at a time, integers at random between 1 and n , and we need to know if the integer just chosen has been chosen before. If so, the new integer is discarded, otherwise is kept.

But how shall we discover if the latest integer is “new”? If we examine all integers so far chosen, we end with $O(k^2)$ labor, in violation of (ii). If we arrange the integers in a linked tree, the labor may drop, still not in $O(k)$, but the links force a violation of (i). If we keep an array whose i th entry tells us whether i has been chosen, then (i) and (ii) will be violated.

Our algorithm, which meets all the requirements, is, in summary, to divide the range $[1, n]$ into k subintervals R_l ($l = 1, 2, \dots, k$) of approximately equal sizes, and choose the cardinalities $|B_l|$ of the sets B_l of elements to be chosen from each subinterval. We determine them by a rejection method which simulates the choosing and recording of the members of the B_l . When this step is finished, we now must choose the actual members of the B_l from the interval R_l , again uniformly at random. The algorithm, in more detail, is the following.

- Divide the range $[1, n]$ into k subranges

$$R_l = \{m | \lfloor \frac{(l-1)n}{k} \rfloor < m \leq \lfloor \frac{ln}{k} \rfloor\} \quad (l = 1, \dots, k) \quad (4.2)$$

The random k -set to be chosen will consist of members B_l of the subranges R_l . As the number of sets B_l equals k , they will contain very few elements; some will be empty.

- First determine the cardinalities $|B_l|$, without worrying about exactly which elements of R_l will be members of B_l . In doing this, the k storage locations a_i can be used, one for each B_l . We draw a random number x in the range $[1, n]$, determine the R_l to which it belongs, by

$$l = 1 + \lfloor \frac{(xk-1)}{n} \rfloor \quad (4.3)$$

and accept or reject the x depending on whether it “duplicates” an element already accepted. Suppose m members of R_l have already been accepted, while the total number of members of R_l is q , then the probability that x is rejected, is m/q . With this in mind, we may simply reject x with probability m/q , without even checking x against any element that has been accepted. In fact we only maintain a count of the elements that have been accepted. All this is accomplished as follows: Initially, we store in a_l the number $\lfloor \frac{(l-1)n}{k} \rfloor$. This is one unit less than the smallest element in R_l . When an x has been chosen, and R_l is determined by 4.3, we accept x if $x > a_l$, reject if $x \leq a_l$. If x is accepted, we increase a_l by one unit, but drop x .

- When k such x have been accepted, we scan the a_l ($l = 1, \dots, k$) and move those a_l that no longer have their initial values (they represent the nonempty B_l to

the leftmost positions the array, say $(a_1, \dots, a_p)$. Next, for $j = p, p-1, \dots, 1$ we reserve space for each of the B_l in $(a_1, \dots, a_k)$, starting from the right, store the value of l in the rightmost of these spaces, the others are zeros.

- Again, scanning from the right, we now place random elements of R_l into the space reserved for B_l , listing them in order. Duplications are avoided by choosing at random. $m' = 1 + \lfloor hm \rfloor$, where m is the number of members of R_l not yet chosen. and x is the m' th member of this list of unchosen elements.

4.1.2 Random Composition of n into k parts (rancom)

Our algorithm for random compositions is based on the “balls in cells” model which was described above. Briefly, we choose the positions of the cell boundaries at random using the routine which was described in the previous section, then by differencing we find out how many balls are in each cell. The flowchart of the algorithm is shown in figure 4.3.

4.2 Examples and Results

This section illustrates that our automated QTES algorithm is equally efficient when we replace the exhaustive search with the random search which was described in detail in section 4.1

Figure 4.4 and 4.5 displays the results of the random algorithmic modeling for the BBC and Last Action Hero video samples. The number of equidistant values for each P_n and ξ were set to 4 and 10 respectively. The autocorrelation coefficients were calculated for 50 lags. For a discretization level $K = 10$, we chose randomly

Figure 4.3: Flowchart for the ranksb algorithm

Figure 4.4: Autocorrelation Coefficients for the BBC News empirical traffic sample and the QTES models for $K=10$ and $K=20$ using random search

$N = 30$ points of the parametric space G_K and we kept the $B = 10$ best points out of all those 30 points which were randomly chosen. Finally, we optimized the cost function for those B points and took the point which gives the smallest value of the objective function. For a discretization level $K = 20$, we used $N = 70$ and $B = 30$. Figure 4.6 illustrates the results of the random algorithmic modeling for the football video sample. For $K = 10$, we used $N = 70$ and $B = 30$, while for $K = 20$, we used $N = 500$ and $B = 50$.

In most cases the agreement between the QTES model autocorrelation functions and their empirical counterparts is quite good. As expected, when we increase the discretization level from $K = 10$ to $K = 20$, the empirical autocorrelations for both video sequences are more accurately approximated by the autocorrelation of the QTES models.

Figure 4.5: Autocorrelation Coefficients for the Last Action Hero empirical traffic sample and the QTES models for $K=10$ and $K=20$ using random search

Figure 4.6: Autocorrelation Coefficients for the Football empirical traffic sample and the QTES models for $K=10$ and $K=20$ using random search

4.3 Summary

The automated algorithm was modified by replacing the exhaustive search with a random sampling algorithm. Random sampling is done when the exact determination of the quantity by exhaustive search is so time consuming as to be impracticable.

In the random search algorithm we first specify how many points N of the parametric space G_K we want to have randomly chosen. Then we keep those points which give the B smallest values of g_K . Finally, we evaluate the objective function g_K at those B points and we select among them the point x^* which gives the smallest value of the g_K .

The efficiency of the automated algorithm was verified by using the same three MPEG compressed VBR video traffic samples as in the previous chapter.

Chapter 5

Comparison Between Exhaustive Search and Random Search

As we have already mentioned, random sampling is often preferred when the exact determination of the quantity by exhaustive search is time consuming. Each point has equal a priori probability of being selected. Since random sampling selects at random a number of points N from the parametric space G_K and the automated algorithm chooses the best B of those as initial points for the local search, it is reasonable to assume that random sampling will not bring so accurate results as the exhaustive search which uses indiscriminately all the points of the parametric space G_K . Figures 5.1, 5.2 and 5.3 compare the efficacy of the automated algorithm, when random and exhaustive search are used respectively. In all cases, we used a discretization level $K = 20$. For the random search, we used $N = 70$ and we chose the $B = 30$ points as initial points for the local search. In the exhaustive search, we also used $B = 70$. As it can be seen, the results that we got from the automated modeling using the random search are very close to those we got by using the exhaustive search.

The risk that is associated with the random sampling in the automated algorithm is obvious: The random sampling algorithm may not select N “good” points,

Figure 5.1: Autocorrelation Coefficients for the empirical data stream and the QTES ($K=20$) models of example1 (BBC video sample) using exhaustive and random search

Figure 5.2: Autocorrelation Coefficients for the empirical data stream and the QTES ($K=20$) models of example1 (Last Action Hero video sample) using exhaustive and random search

Figure 5.3: Autocorrelation Coefficients for the empirical data stream and the QTES ($K=20$) models of example3 (Last Action Hero video sample) using exhaustive and random search

which means that the best B of those N points which will be used as initial points for the local search will not be “good” enough either. Figure 5.4 illustrates this for the football video sample. In this example, we used a discretization level of $K = 10$. For the random search, we used $N = 70$ and we optimized the best $B = 30$ points. For the exhaustive search, we used $B = 30$.

Since random sampling selects at random points from the parametric space G_K , it is reasonable to assume that the greater the number N of the randomly chosen points we specify from the beginning is, the greater the chances of getting “good” initial points are. Figures 5.5 and 5.6 illustrate this in the case of the football video sample. For a discretization level of $K = 10$, the results are more accurate, when we use $N = 70$ and we optimize locally the best $B = 30$ points of those, rather than when we use $N = 30$ and we optimize locally the best $B = 10$ points of those. For a

Figure 5.4: Autocorrelation Coefficients for the empirical data stream and the QTES ($K=20$) models of example1 (football video sample) using exhaustive and random search

discretization level $K = 20$, we derive the same conclusion as it can be seen in 5.6.

From the above analysis, we can conclude that in the case of random search, the greater the parameters B and N are, the more accurate the traffic model will be. The parameter B represents the number of the points that are given to the local search as initial points for the non-linear local optimization to start. Therefore by increasing the number B , we increase the number of the local searches that the automated algorithm is going to perform. This increases the required running time for the automated algorithm. The running time is also increased when we increase the parameter N because we increase the number of the random searches in the first phase of the optimization.

Figure 5.5: Autocorrelation Coefficients for the QTES ($K=10$) models of example3 (football video sample) for $N=30$ and $N=70$ using random search

Figure 5.6: Autocorrelation Coefficients for the QTES ($K=20$) models of example3 (football video sample) for $N=100$ and $N=500$ using random search

Video Sample	Exhaustive Search, $K=10$	Random Search, $K=10$	Exhaustive Search, $K=20$	Random Search, $K=20$
BBC	3.52	0.86	55.13	18.63
Last Action Hero	5.105	1.51	95.37	53.74
Football	2.46	1.05	50.17	46.15

Table 5.1: Required running time (hours) for obtaining the traffic models from the empirical VBR video samples

5.1 Summary

The comparison between the exhaustive search and the random search showed that in most cases the random search is equally efficient as the exhaustive search. More than that the random search is much more faster than the heuristic search. In table 5.1, we can see the time it took us in hours to obtain the QTES traffic models for both the exhaustive and random search and for discretization levels of $K = 10$ and $K = 20$.

The comparison also revealed the risk that is associated with the random sampling: The random sampling algorithm may not select a N number of “good” points. Of course, the greater the number N of the randomly chosen points is, the higher the chances of getting “good” initial points are and the longer the automated algorithm is going to take. Therefore, the accuracy of the approximation of the empirical autocorrelation with its QTES model counterpart is a tradeoff between the number N of the randomly chosen points and the time that the algorithm is going to take.

From the table 5.1 it is also obvious that the burstiness which is involved in

a video traffic stream can increase the running time of the algorithm significantly. while at the same time it can reduce the positive impact of the use of the random search. In the case of the BBC and Last Action Hero video samples the difference in the running time between the random and exhaustive search is significant mainly due to the fact that the random search is faster than the exhaustive search. In the case of the Football video stream though the random search is still faster than the exhaustive search, but the difference is not so significant as it is in the case of the two other video samples. This happens because the Football traffic stream contains more burstiness than the other two and this means that the local search needs more time to find a local minimum which will be as close as possible to the global minimum of the objective function.

Chapter 6

QTES Generator

Traffic generators are a key element in studying communication networks and evaluating their performance. Traffic generators are most often employed in two fundamental ways: either as part of an analytical model, or to drive a discrete event simulation. The most common modeling context is queueing, where traffic generators generate the traffic which goes to a queue or a network of queues and various performance measures are calculated.

The purpose of the QTES generator is to generate sample path realizations which capture the first order statistics (marginal probability distribution) and the second order statistics (autocorrelation function) of the given empirical data sample simultaneously using the QTES methodology. In chapter 3, we developed an algorithm which calculates in an automated way the parameters which largely determine the autocorrelation structure. These two parameters, namely the DTES innovation and the stitching factor, are given as inputs to the QTES generator algorithm which is described later in this chapter to ensure that the autocorrelation structure of the synthetic QTES traffic is as close as possible to the empirical autocorrelation structure. As we will see in the sequel, an exact match to the empirical distribution (histogram)

can be guaranteed algorithmically through the inversion method [2, 26].

A QTES generator which can generate autocorrelated stochastic sequences is comprised of a foreground/background scheme like the one which is shown in figure 6.1 [12]. The generator generates two autocorrelated sequences in lockstep: an auxiliary sequence $\{Q_n\}$ and the target sequence $\{Z_n\}$. The background sequence $\{Q_n\}$, is a QTES background sequence generated recursively by the transition function $Q_n = T(Q_{n-1}, J_{n-1})$, where the transition function T is given by 2.18 and $\{J_n\}$ is a sequence of independent identically distributed (i.i.d) variates which have the same marginal distribution with that of the DTES innovation sequence given by equation 2.12. All the parameters which are needed to produce the background sequence Q_n , namely the DTES innovation sequence V_n and the stitching factor ξ , are determined by the automated algorithm which was described in detail in chapter 3. The sequence $\{Z_n\}$ is obtained in lockstep from the sequence $\{Q_n\}$ via $Z_n = D(Q_n)$, where the mapping D is the distortion given by equations 2.4 and 2.9.

6.1 Inverse Transform Method

The Inverse method is one of the widely-used techniques for generating variates which follow the marginal distribution of a given data sample [2, 26]. It can be used to sample from the exponential, the uniform, the Weibull and the triangular distributions, but most often it is used, when we want to sample from empirical distributions, which is what we have in our case. Additionally, it is the underlying principle for sampling from discrete distributions as we will see in the sequel.

Suppose that we wish to generate a random variate X that is continuous and has distribution function F that is continuous and increasing when $0 < F(x) < 1$.

Figure 6.1: QTES generator

Let F^{-1} denote the inverse of the function F . Then the algorithm for generating a random variable X having distribution function F is as follows ($\sim$ is read “distributed as”):

1. Generate $U \sim U(0,1)$
2. Return $X = F^{-1}(U)$

where U denotes the uniform distribution. Note that $F^{-1}(U)$ will always be defined, since $0 \leq U \leq 1$ and the range of F is $[0,1]$. Figure 8.1 illustrates the algorithm graphically, where the random variable corresponding to this distribution function can take either positive or negative values; the particular value of U determines which will be the case. In the figure, the random number U_1 results in the positive random variate X_1 , while the random number U_2 leads to the negative variate X_2 .

A more detailed description of the Inverse Transform Method can be found in [2, 26]

Figure 6.2: Inverse-transform method for continuous random variables

6.2 QTES algorithm

The QTES generator algorithm consists of three components: A discrete number generator, *dis_genn*, which generates a sequence of i.i.d variates which have the same marginal distribution with that of the DTES innovation, a background QTES generator, *qtes*, which generates the uniform QTES background sequence and finally a foreground QTES generator *qtesgen* which generates the foreground QTES sequence which captures both the autocorrelation structure and the marginal distribution of the empirical data set. In figure 6.1, the three generators are denoted by J_n , Q_n and Z_n respectively.

6.2.1 Discrete number generator (*dis_genn*)

Our situation here is the following: We have a probability mass function $P_1, P_2, \dots, P_K$ on the nonnegative integers which represents the innovation sequence we got from

the automated algorithm which was described in chapter 3, and we want to generate discrete random variates J with the corresponding distribution.

The direct inverse transform method, is as follows:

1. Generate $U \sim U(0, 1)$
2. Return the nonnegative integer $J = I$ satisfying

$$\sum_{n=1}^{I-1} P(n) \leq U < \sum_{n=0}^{n=I} P_n \quad (6.1)$$

6.2.2 Background QTES generator (*qtes*)

This routine generates the background QTES sequence Q_n in a recursive way by using the transition function which is given by equation 2.18. The innovation sequence and the stitching factor that are used for the generation of the background QTES variates have been calculated by the automated algorithm which was described in chapter 3. The background sequence Q_n which is generated has the same correlation structure of the empirical data stream. We have to recall the important fact the the background QTES variates are uniformly distributed as we mentioned in chapter 2.

6.2.3 Foreground QTES generator

The situation here is the following: We have an empirical data sample in the form of a histogram and we want to generate a sequence of numbers which has the same distribution with that of the empirical data set. In order to do that we will use the inverse transform method.

As we have mentioned, we have the empirical data sample in the form of a histogram. That means that the empirical data is grouped into k adjacent intervals

Figure 6.3: Continuous, piecewise-linear empirical distribution function from grouped data

$[a_0, a_1), [a_1, a_2), \dots, [a_{k-1}, a_k)$, so that the j th interval contains s_j observations, where s is the size of the empirical sample and $s_1 + s_2 + \dots + s_k = s$. A reasonable piecewise-linear empirical distribution function G could be specified by first letting $G(a_0) = 0$ and $G(a_j) = (s_1 + s_2 + \dots + s_k)/s$ for $j = 1, 2, \dots, k$. Then, interpolating linearly between the a_j 's, we can define

$$G(x) = \begin{cases} 0, & \text{if } x < a_0 \\ G(a_{j-1}) + \frac{x-a_{j-1}}{a_j-a_{j-1}}[G(a_j) - G(a_{j-1})] & \text{if } a_{j-1} \leq x < a_j \\ 1, & \text{if } a_k \leq x \end{cases} \quad (6.2)$$

Figure 6.3 illustrates this specification of an empirical distribution for $k = 4$.

The following inverse-transform algorithm generates a random variate which has the same marginal distribution as the empirical data set:

1. Generate a background QTES variate Q . Recall that $Q \sim U(0, 1)$

2. Find the nonnegative integer I ($0 \leq I \leq k - 1$) such that $G(a_I) \leq Q \leq G(a_{I+1})$.
and return $Z = a_I + [Q - G(a_I)](a_{I+1} - a_I) / [G(a_{I+1}) - G(a_I)]$

In figure 6.1, the inverse transform algorithm is denoted with the mapping D . The foreground sequence Z_n that is generated has the same marginal distribution with the empirical data sample and simultaneously captures the autocorrelation structure of the empirical data stream.

6.3 Results

This section illustrates the efficacy of the QTES generator algorithm via the same three examples that we used in the previous sections, namely the BBC news, Last Action Hero and football MPEG compressed VBR video samples.

In all cases, as it can be seen in the figures 6.4, 6.5 and 6.6, the Inverse Transform Method guarantees an exact match of the QTES traffic model histograms to their empirical counterparts. As far as the match to the empirical autocorrelation function is concerned, in the case of the BBC news and Last Action Hero video samples, we used the DTES innovation sequence and stitching factor we got from the automated algorithm using the Global search algorithm, while in the case of the football video sample, we used the results that the automated algorithm gave us using the random search algorithm. We have used a line with “+” to represent the empirical autocorrelation function, a dashed line to represent the autocorrelation function of the QTES model with a discretization level of $K = 10$ and a line with “*” for a discretization level of $K = 20$. In all cases, we calculated the autocorrelation coefficients for 50 lags.

Figure 6.4: QTES models ($K = 10$ and $K = 20$) of the BBC news video traffic sample

As it can be inferred from the figures 6.4, 6.5 and 6.6, the match of the QTES models generated by the QTES generator algorithm with their empirical counterparts is very accurate. It can also be seen that the approximation is more accurate for a discretization level of $K = 20$ than it is for a discretization level of $K = 10$. In the case of the football video sample, this is more visible, since there is long term dependence in the autocorrelation structure.

Figure 6.5: QTES models($K = 10$ and $K = 20$) of the Last Action Hero video traffic sample

Figure 6.6: QTES models($K = 10$ and $K = 20$) of the football video traffic sample

6.4 Summary

The QTES traffic generator generates traffic models from empirical data streams which capture both the marginal probability distribution and the autocorrelation function of the given empirical data sample simultaneously.

In chapters 3 and 4 we saw how we can produce a QTES autocorrelation function which approximates the empirical one as closely as possible. We achieved that by finding an optimal DTES innovation sequence and an optimal stitching parameter. These two parameters are given as inputs to the QTES generator algorithm and they are responsible for capturing the empirical autocorrelation. The Inverse Transform method which is the major component of the QTES generator algorithm guarantees an exact match of the QTES marginal distribution to the empirical one.

The efficiency and accuracy of the QTES generator was verified by using the same three MPEG compressed VBR video traffic samples as in the previous chapters.

Chapter 7

Conclusions and Recommendations for further Study

7.1 Conclusions

The ATM transport concept of B-ISDN allows for an efficient and flexible use of network resources for many different services with high transmission rates. One major traffic source which will make use of this capability will be video and multimedia applications. The changes in the cell rate of such sources are caused by the compression of the digitized video frames. The MPEG (ISO Moving Picture Expert Group) coding scheme is expected to be the major compression algorithm for the first ATM video applications. Thus there is a need to find source traffic descriptors which can be used efficiently for connection admission control and usage parameter control for sources like MPEG video sources where burstiness involved.

This thesis presented a consolidated study of modeling bursty traffic, over high speed networks, having investigated two of the principal challenging statistical properties of an empirical traffic sample that have to be captured so that the traffic

model can be an accurate approximation of the empirical data stream: the marginal distribution and the autocorrelation structure of the empirical traffic.

Leaving the autocorrelation structure out of the traffic model can be dangerous, since temporal dependence is a major cause of burstiness in telecommunication traffic. The modeling was done by using the QTES methodology in an automated approach (heuristic/random search local search algorithm). Although the evaluation of the methodology and the automated algorithm was limited to MPEG video traffic sources, the proposed approach is generic and can be applied to any traffic stream which goes through a network.

Previous research [5] specified a first approach to capturing both the autocorrelation structure and the marginal distribution of the empirical traffic stream, namely the Global Search Local Optimization (GSLO) algorithm [15] which is an automated procedure for modeling an empirical traffic source using TES processes. The problem with that approach as we have already mentioned in chapter 2 is that TES based queueing models have been analytically intractable, since TES processes are transformed Markov sequences over an uncountable state space. Queueing models with TES-based traffic have only been studied via Monte Carlo simulation. More than that this approach uses an exhaustive search over the parameter space to get the most promising points which will be given to the local optimization algorithm in order to get the optimal innovation and stitching factor. This makes the whole approach very time consuming, since the exhaustive search algorithm is of combinatorial nature. This thesis on the other hand outlines an algorithmic automated procedure for modeling an empirical traffic source using QTES processes. A QTES process approximates a continuous-state TES process by a discrete-state variant QTES counterpart.

In this way we can have a tractable queueing model which is still quite accurate in the sense that the traffic model captures first-order and second-order statistics of the empirical traffic data. The algorithm combines an initial search over the whole parameter space to get the best initial points with a local nonlinear programming technique to minimize an objective function consisting of the distance between the empirical autocorrelation function and its candidate model counterpart. As far as the initial search over the whole parameter space is concerned, we used an exhaustive search at the beginning and then we developed an algorithm which uses a random search to get the initial points for the local non-linear optimization. The use of the random search reduced significantly the total amount of time that the algorithm needs to bring results as it can be shown in table 5.1.

An exhaustive/random search local optimization algorithm was chosen to be used in this automated modeling of empirical data due to the existence of fast, analytical formulas for the autocorrelation function and the partial derivatives and also due to the simplicity of constraints imposed on the objective function. More specifically, the sequential quadrating programming (SQP) was chosen for the implementation of the local search optimization. This choice was made on the basis of the efficiency of the optimization algorithm which makes use of second order information, as far as the KT equation is concerned using updating procedures. This second order information was easy to obtain again due to the reasons that we mentioned at the beginning of the current paragraph.

The approximation of the empirical autocorrelation with its QTES model counterpart is shown to be very good when exhaustive search is used. It is also shown that by increasing the discretization level K of the model the approximation is

more accurate. As expected, the exhaustive search is very time consuming due to the fact that the algorithm which implements it, namely the next composition routine, is of combinatorial nature as it has been shown in section 3.3.1. From expression 3.6, it can be concluded that any increase in the discretization level K of the QTES model leads to a combinatorial increase in the total number of searches in the exhaustive algorithm.

We replaced the exhaustive search with a random sampling algorithm and we managed to reduce significantly the runtime of the whole algorithm as it can be seen from table /reftimetable. It was shown in chapter 5 that the use of a random search generates models almost as accurate as those we get from the exhaustive search over the parametric space. Since random sampling selects at random points from the parametric space, it is reasonable to assume that the greater the number of the randomly chosen points are, the higher the chances of getting “good” initial points are. This intuitive assumption was confirmed in the case of the football video sample. The approximation, as far as the empirical autocorrelation is concerned, was much more accurate, when we produced 70 initial randomly chosen points than 30 initial randomly chosen points. Of course the more randomly chosen initial points one generates, the more time consuming the automated algorithm is going to be.

On the basis of the general efficacy of the automated algorithm, it is inferred that many parameters can play an important role in the accuracy of the approximation and in the time that the automated algorithm needs in order to give results. Firstly, the discretization level K of the QTES model seems to be a determining factor of the accuracy of the model. The more we increase the discretization level, the more accurate the traffic QTES model is. Also the parameters N_P and ξ , namely the

number of equidistant values that each P_n and ξ can assume respectively, are of much importance, since they determine the accuracy of the approximation of the innovation sequence and the stitching factor in the local optimization. These three parameters, although they are the most crucial ones in determining the accuracy of the approximation, they are also the ones which mainly determine how fast the automated algorithm is going to be. Since they are the only participants in the expression 3.6, every slight increase in either of those will increase the total number of searches in a combinatorial way making the automated algorithm time consuming. The number of the best initial points B which will be optimized locally also plays an important role in the fastness of the algorithm, since it determines how many times the local optimization routine is going to run. As it can be seen, the efficacy of the algorithm, as far as the speed and the accuracy of the approximation is concerned, is a tradeoff among all the above parameters.

7.2 Recommendations for Further Study

7.2.1 Experimental Design and Design Optimization

The matching accuracy of the QTES models obtained from empirical traffic samples through the QTES modeling methodology was based on informal ad-hoc testing. Experimentation with the traffic models provided evidence that changes to the discretization level K , the number N_P and N_ξ of equidistant values that each P_n and ξ can assume, the number B of the best initial points which are optimized locally and the number of the randomly selected points, when random sampling is used, would affect the accuracy of the approximation of the empirical autocorrelation function, as measured by the cost function 3.3 and the time that the automated algorithm needs

to bring results. A more formal qualitative analysis could be based on determining experimentally which of the above parameters (or combinations of parameters) have the most significant influence on the matching accuracy of the QTES traffic model and the time that it takes for such a traffic model to be produced. This could be done by performing a 2^k factorial analysis [26]. After having determined which parameters play the most important role in the matching accuracy of the QTES traffic model and the duration of the automated algorithm, we can use these parameters as potential candidates for an optimization of the cost function with respect to those parameters. A more detailed description of the procedure is given in the sequel.

Experimental Design

Factorial analysis supports the characterization of several design parameters (factors) with reference to a standard system response measure, the cost function. Two levels (discrete values or ranges) are selected for each of the k factors. Simulations are run for each of the 2^k possible factor-level combinations, from which the effect of each factor and combinations of factors can be determined. Table 7.1 illustrates a notional experiment involving three factors (F_1, F_2, F_3).

The two levels associated with each factor are denoted “+” (high) and “-” (low). Eight different combinations of factors and levels are therefore possible and each combination defines a design point. The simulation result for each design point is expressed as a cost function value (C_1, C_2, C_3). The main effect for a particular factor is determined by computing the average difference in simulation results when the factor level is high and low, and the levels for all other factors remain fixed. Combined effects of two factors can also be defined.

DESIGN POINT	FACTOR			SIMULATION RESULT
	1	2	3	
1	-	+	-	C1
2	-	-	+	C2
3	-	-	+	C3
4	-	-	+	C4
5	-	+	-	C5
6	-	+	-	C6
7	-	-	-	C7
8	-	-	-	C8

Table 7.1: Generic 3 factor design analysis

When several simulation results are collected for each design point, the main effects ($e_1, e_2, \dots, e_k$) and combined effects (e_{12}, e_{13}, etc) can be calculated as confidence intervals, which are influenced by the variability of the measured effect and the number of simulations performed for each factor-level combination. A completely positive or negative confidence interval indicates that the factor, or combination of factors, has a statistically significant influence on the accuracy of the model and the duration of the algorithm and is a good candidate for optimization efforts.

Therefore, we can use the above procedure for our case and determine which factors play the most important role in the matching accuracy of the autocorrelation function and in the time that the automated algorithm needs to bring results.

Design Optimization

Optimization of a cost function can be achieved by locating the global minimum of the cost function, which has a multidimensional solution surface. The optimization effort involves computing the value of the cost function at several points on the solution surface and then applying some methodology, such as gradient descent, to locate the minimum value. For this particular optimization problem though, the Mean Field Annealing seems to be a nice choice, although a lot of the previously mentioned techniques can be applied,

Mean Field Annealing (MFA), a variant of standard simulated annealing, is a statistically based technique which is well suited to the general problem of optimizing stochastic cost functions over a multi-dimensional parameter space [27]. In particular, MFA is resistant to variation in the cost estimate from simulation, is not susceptible to local minima in the solution surface and converges rapidly to near-optimal solutions. The MFA algorithm randomly selects a dimension N , in the parameter space and generates estimates of the cost function C , at evenly spaced increments over the interval $[N_{min}, N_{max}]$. This process is repeated iteratively by selecting new values for the N axis, while the value for all other dimensions remain fixed, and until equilibrium is reached.

A parameter T , which is commonly referred to as the temperature, is involved in all those calculations, and provides a mechanism for escaping from local minima in the solution surface. By gradually lowering the temperature over time, we can avoid local minima and relax to an optimal or near optimal solution.

Therefore, after having determined which parameters are the most significant ones, we can follow the MFA algorithm and estimate the optimal values of those

parameters so that we can have a very good accuracy in the approximation of the empirical autocorrelation and an optimal duration for the automated algorithm.

7.2.2 Other Recommendations

The thesis demonstrated the efficacy and suitability of the QTES models and of the QTES automated algorithm for producing traffic models from empirical traffic streams which capture both the empirical marginal distribution and the empirical autocorrelation function. This research was focused only on modeling the traffic that is given as an input to a network or to a queueing system. Taking into consideration the fact that traffic models are mainly used to predict accurately different aspects of network performance such as cell loss statistics, further study could be done in the analysis of the performance of queueing systems and networks, when this kind of input is used. Such a study could be very useful, since it would provide us with information about the accuracy of a traffic model that is needed to lead to an improved network performance. In our case for example, let us assume that we have a queueing system and we want to compare its output performance when we use a QTES traffic input with a discretization level of $K = 10$ and when we use a discretization level of $K = 20$. If there is not significant improvement in the output performance between the two cases, then, maybe, there is not practical reason to use $K = 20$, given the fact that even a small increase in the discretization level of the QTES model increases the number of initial searches that the algorithm performs in a combinatorial way.

Many different traffic models have been proposed for the data generated by variable bit rate (VBR) coders. It has been observed in simulation studies that the use

of different coders results in a wide variation in predicted cell loss statistics. Therefore, there is a need to study several traffic models (such as TES models, QTES models, self similar traffic models) and compare their accuracy in capturing the statistical signature of the empirical data. Then their predictions of network performance can be examined, so that we can determine which models give a better prediction. Of course, the same procedure can be followed for other types of data as well, given the fact that certain traffic models are more appropriate for certain types of data than others.

In this thesis, we used an exhaustive algorithm in order to find the best initial points which will be used for local optimization. We also developed a random search algorithm in our effort to reduce the duration of the automated algorithm. Except for these two approaches, someone could investigate the possibility of using more sophisticated sampling methods to replace the exhaustive or random search. Methods such as important sampling and mean field annealing for example can be much more faster and more accurate at the same time.

References

- [1] V.S.Frost and B. Melamed. "Overview of Simulation and Traffic Modeling for Telecommunications Networks". *IEEE Comm.Mag.* Vol. 32(3):70–81, Mar. 1994.
- [2] J. Banks and J. S. Carson II. "*Discrete-Event System Simulation*". Prentice-Hall, Inc, 1984. pages 294-316.
- [3] G.S.Fishman. "*Principles of Discrete Event Simulation*". John Wiley and Sons, Inc, 1978.
- [4] B.Melamed M.Livni and A.K.Tsiolis. "The impact of Autocorrelation on Queuing".
- [5] B.Melamed. "An overview of TES processes and modeling methodology". *IEICE TRANS.COMMUN.* E75-B(12):359–393, 1993.
- [6] E. Cinlar. "*Introduction to Stochastic Processes*". Englewood Cliffs, NJ: Prentice-Hall, 1975.
- [7] H.Heffers and D.M.Lucantoni. "A Markov Modulated Characterization of Packetized Voice and Data and Related Statistical Multiplexer Performance" . *IEEE J. on Selected Areas in Commun. SAC-4*, pages 856–886. 1986.

- [8] A.Tabatabai D.Heyman and T.V. Lakshman. "Statistical Analysis and Simulation Study of Video Teletraffic in ATM Networks" . *IEEE Trans. Circuits and Systems for Video Technology*, 2.
- [9] Ren Q. Melamed. B. and B. Sengupta. "The QTES/PH/1 Queue". *Performance Evaluation*. Vol. 26(1):1-20, Jul. 1996.
- [10] D. L. Jagerman and B. Melamed. "Burstiness Descriptors of Traffic Streams: Indices of Dispersion and Peakedness". *Proceedings of the Twenty Eighth Annual Conference on Information Sciences and Systems*, Mar. 1992.
- [11] B. Melamed. "TES: A Class of Methods for Generating Autocorrelated Uniform Variates". *ORSA J. on Computing*, Vol. 3(4):317-329, 1991.
- [12] D. L. Jagerman and B. Melamed. "The Transition and Autocorrelation structure of TES Processes Part I: General Theory". *Stochastic Models*. Vol. 2(8):193-219. 1992.
- [13] D. L. Jagerman and B. Melamed. "The Transition and Autocorrelation structure of TES Processes Part II: Special Cases". *Stochastic Models*, Vol. 3(8):499-527, 1992.
- [14] B. Melamed and J. Hill. "Applications of the TES Modeling Methodology". *Proceedings of WSC '93*, pages 1330-1338, 1993.
- [15] Predrag R. Jelencovic and Benjamin Melamed. "Automated TES Modeling of Compressed Video". *IEEE Infocom '95*, pages 746-752, 1995.

- [16] W.Feller. *"An introduction to Probability Theory and its Applications "*. John Wiley and Sons Inc. New York. 1951.
- [17] R. Fletcher. *"Practical Methods of Optimization "*. John Wiley and Sons. Inc. 1980.
- [18] W. Murray P.E. Gill and M.H. Wright. *"Practical Optimization "*. Academic Press. London. 1981.
- [19] M.J.D.Powell. *"Variable Metric Methods for Constraint Optimization "*. Springer Verlag. 1983.
- [20] H.D. Sherali M.S Bazaraa and C.M. Shetty. *"Nonlinear Programming "*. John Wiley and Sons, Inc. New York. 1993.
- [21] Wayne L. Winston. *"Introduction to Mathematical Programming: Applications and Algorithms "*. International Thomson Publishing, California. 1995.
- [22] C.G. Broyden. "The Convergence of a Class of Double-rank Minimization Algorithms". *J.Inst.Maths.Applics*, Vol. 6:76–90, 1970.
- [23] R. Fletcher. "A New Approach to Variable Metric Algorithms ". *Computer Journal*, Vol. 13:23–26, 1970.
- [24] D. Goldfarb. "A family of Variable Metric Updates Derived by Variational Means". *Mathematics of Computing*. Vol. 24:23–26, 1970.
- [25] D.F. Shanno. "Conditioning of Quasi-Newton Methods for Function Minimization". *Mathematics of Computing*, Vol. 24:647–656, 1970.

- [26] A. Law and D. Kelton. "*Simulation Modeling and Analysis*". McGraw-Hill Inc. New York, 1991.
- [27] M.Devetsikiotis and K. Townsend. "Statistical Optimization of Dynamic Importance Sampling Parameteres for Efficient Simulation of Communication Networks". *IEEE ACM Trans. Net.*, Vol. 1(3):293-305, 1993.

IMAGE EVALUATION TEST TARGET (QA-3)

APPLIED IMAGE, Inc.
1653 East Main Street
Rochester, NY 14609 USA
Phone: 716/482-0300
Fax: 716/288-5989

© 1993, Applied Image, Inc., All Rights Reserved