Spectral Estimation for Congestion Control in ATM Networks

Jean-François Huard

A Thesis

in

The Department

of

Electrical and Computer Engineering

Presented in Partial Fulfillment of the Requirements

for the Degree of Master of Applied Science at

Concordia University

Montréal, Québec, Canada

June 1992

ISBN 0-315 81017-3

ABSTRACT

Spectral Estimation for Congestion Control
in ATM Networks

Jean-François Huard

This work represents the first step in a new approach to congestion control in Asynchronous Transfer Mode (ATM) networks. By monitoring the traffic stream at any point in the network, spectral analysis is performed to estimate the traffic characteristics. Traffic is segmented into a number of distinct classes which can be identified by their power density spectrum. Limited buffering is allowed at the input ports of the ATM multiplexer to resolve short-term conflicts. Using the correlation properties and spectral characteristics of the variable bit rate sources the estimation can be performed. The voice source model is a binary Markov process which is widely used. The video source model is an autoregressive Markov model which is gaining the most attention. Data sources are modeled as Bernoulli and Poisson processes. On the basis of the simulations and real video sequences analysis, spectral analysis seems to be an appropriate technique to identify the sources of traffic on an ATM link.

SOMMAIRE

Spectral Estimation for Congestion Control
in ATM Networks

Jean-François Huard

Ce travail représente la première étape d'une approche nouvelle pour le contrôle de la congestion dans les réseaux avec mode de transfert asynchrone. L'analyse spectrale est employée pour estimer le nombre de communications en cours pour divers types de sources tel la voix, la vidéo et les données. L'étude de la composition du trafic peut être faite à n'importe quel endroit à l'intérieur du réseau grâce à l'utilisation de l'analyse spectrale. En effet, en utilisant les caractéristiques statistiques de chaque type de source—moyenne, fonction d'autocorrélation et spectre de densité de puissance, l'estimation du nombre de communications en cours pour chaque classe de source est rendue possible. Le modèle de source utilisé pour la voix est un processus de Markov binaire très connu. Pour la vidéo, un processus autorégressif très populaire est employé. Finalement, les sources de données sont modélisées à l'aide de processus de Poisson et de Bernoulli. En se basant sur des simulations à l'ordinateur et en utilisant des séquences expérimentales de vidéo, l'analyse spectrale apparaît être une technique appropriée pour l'estimation de la composition du trafic dans les réseaux avec mode de transfert asynchrone.

To Nathalie,

Louis-Alexandre, Jean-Philippe and my parents

ACKNOWLEDGEMENTS

I would like first and foremost to express my sincerest gratitude to my thesis supervisor, Dr. Jeremiah F. Hayes for his superior guidance and insight throughout the span of this research. It is my belief that it is a rare occasion for a student to encounter a supervisor who so unselfishly devotes so much time and effort to a student's education as does Professor Hayes.

I also take pleasure in thanking Dr. Jeffrey Krolik for stimulating my interest in digital signal processing and spectral analysis.

I would especially like to thank my friends, François Gagnon for the time he spent with me in many discussions, and Jean Belzile for reviewing my thesis.

I would also like to thank the Natural Sciences and Engineering Research Council of Canada for the Centennial scholarship they awarded me, the Centre de recherche en informatique de Montréal for its télématique scholarship for the period beginning September 1990 to December 1991, and Bell-Northern Research for having supported this project.

I also wish to express my gratitude to my wife, Nathalie, for her support, patience and care over the many weekends and nights that I spent at school working on my thesis, and my sons Louis-Alexandre and Jean-Philippe for enduring my absence.

Finally, I would like to thank all the persons, although not individually specified, who lent their support and helped make this thesis a reality.

"*Mens sana in corpore sano*"

Sound mind in a sound body

Maxime de Juvénal (Satires, X, 356)

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

Chapter 1

Introduction

"Imagination is more important than knowledge"

Albert Einstein

1.1 Overview

The Asynchronous Transfer Mode (ATM) seems to be the preferred transport mode for the future Broadband Integrated Services Digital Networks (B-ISDN) as described in Recommendation I.121 of CCITT [1, 2]. These networks will have to support a large variety of services such as voice, several kinds of video types: conversational, entertainment and instruction video, as well as low- and high-rate data transfer. An important feature of the B-ISDN is the ability to provide variable transmission rates to achieve a constant quality of service. Since each source may generate information at a variable bit rate (VBR), an improvement in bandwidth efficiency can be achieved by the means of statistical multiplexing. Thus, the ATM links will carry a statistical mixture of traffic. To prevent congestion, call admission control and policing can be carried out at the input nodes. These congestion control policies are preventive and cannot be applied at an interior point in the network, at the output of a switch or multiplexer for example. At that point, all that is available to predict the onset of congestion is the traffic stream. Using this information, Hayes[3] suggested an approach to congestion control using spectral analysis. This three steps proposed approach can be applied anywhere in the network:

1. Estimation of the number of off-hook calls of each class (calls in progress).

2. Estimation of the state activity of the calls in progress.

3. Prediction of the time until onset of congestion based on the state activity estimated in 2.

This thesis concerns the first step.

A basic premise of the study is the limited role of buffering in the network. Buffers are important to resolve short-term conflicts at the output of the ATM multiplexer, however they cannot be large enough to store packets for any length of time. The role of routing is also very important. The multiplexer (or switch) must route all the packets of a source-destination pair to the same output port. The limited buffers and the routing procedures allow one to assume that the correlation properties of the VBR sources are carried at the output of the multiplexer. The estimation of the number of off-hook calls is done by means of spectral analysis using the correlation properties of the sources.

In the process of the development of congestion control schemes for high-speed networks, many avenues have been investigated [4, 5, 6, 7]. Spectral analysis of the mixed traffic appears to be a relatively new approach. It is understandable that this approach has not yet been widely investigated since the new algorithms for compression of digital video are not well understood [2]. This leads to a lack of information on the basic statistical characteristics of the VBR sources that the ATM networks will have to support. The new congestion control schemes must be flexible enough to handle the future types of sources and be adaptive such that as the research in teletraffic characterization and source modeling will progress, the schemes can be modified to be more efficient.

1.2 Research Objectives

The primary goal of this research project is to estimate the number of off-hook calls in progress on an ATM link. What does this mean exactly? It means that we want to monitor the traffic stream anywhere in the network, and somehow estimate the number of calls that are in progress on that link, and that, for each type of source.

Globally, two specific questions have to be answered:

1. *What characterizes each type of source? For example the characteristic features*

can be obtained from the autocorrelation function and power spectral density of the source process.

2. *How can the desired characteristics be retrieved from the ATM traffic stream? For example, power spectral estimation or filtering can be achieved.*

The second part of the project is very broad. Although several strategies have been studied, all possible approaches have not been explored. In this thesis, only certain fundamental techniques to estimate the traffic composition are investigated through simulation.

1.3 Contributions

The estimation of the number of off-hook sources on ATM links introduces new challenges if all the aspects of the real systems are to be taken into account. This research tries to deepen the comprehension of the process arising from the multiplexing of many types of VBR sources. The following contributions are the results of this work:

1. The characterization of voice and video sources packet processes by their autocorrelation function and power spectral density.

2. Assessment of the suitability of the various techniques to estimate the characteristics of the traffic.

3. A comparative study of these strategies by computer simulation on simulated streams.

The results of simulation show that different classes of sources can be discriminated; furthermore, with Monte-Carlo simulations, the number of sources of each class can be estimated. Unfortunately, when sampling a physical system, the sampling duration is not enough long to get rid of the phase effect between the

sources, yielding nonergodic processes for the observation period and distording the spectrum.

1.4 Thesis Outline

This thesis is divided into six chapters. The chapter 2 presents the basic concepts of ATM networks: statistical multiplexing, buffering, packet contention, time slotting and cells. These notions enable us to define the packet process arising from the multiplexing of several sources.

The third chapter describes the source models used in the study. The *binary Markov process* for the voice source is presented first. The *autoregressive model* for the average bit rate process of a video source follows. The data models are briefly presented in the last section of the chapter. The theoretical autocorrelation functions and power spectral densities are derived for each model. From this, the characteristic features of voice and video sources are obtained.

Chapter 4 describes the techniques which can be applied to estimate the wanted characteristics. The classical estimators of power spectrum such as periodogram and correlogram are presented first. The model-based approach immediately follows. Filtering strategies such as decimation, notch or comb filtering to extract the power of characteristic harmonics conclude this chapter.

In chapter 5 the simulation results obtained from simulated and real traffic streams are presented. Comparisons between theoretical, simulated and experimental autocorrelation function and power spectral density are presented. This chapter also provides a comparative study of the strategies elaborated to estimate the number of off-hook sources of type which are in progress.

Finally the conclusions are given in the sixth and last chapter. First the results of this project are summarized, then recommendations for future research projects are enumerated.

Chapter 2

ATM Networks

The asynchronous transfer mode is considered to be the preferred transport mode for the B-ISDN because unlike traditional synchronous transfer mode system (e.g. switched TDM) it is able to support a wide mix of services over a common network. The ATM provides a nonhierarchical structure in which short fixed length packets termed as cells, from different connections (source-destination pair) are multiplexed independently of their bit rates and burstiness. To achieve good performance, all connections may be statistically multiplexed to take advantage of the variable information transfer rate of each type of application.

2.1 Variable Bit Rate Sources

A variable bit rate source is a source whose bit rate depends on several factors such as information content, coding algorithm and quality of service (QOS) requirements [8]. For example, consider a PCM voice source that would continuously switch between a silent state where no packets are generated and an active state where packets are generated deterministically at a constant time intervals. Other examples would be subband coded video, discrete-cosine transform and DPCM coded video.

According to the coding algorithms selected, the information stream will be more or less bursty and correlated. It is these properties, the burstiness and correlation that we will use to characterize the sources in chapter 3. All the types of sources which are studied in this thesis can be modeled at different levels which are encapsulated in each other: the call level, the burst level and the cell level. At the call level, the source can be on-hook or off-hook. At the burst level, which is encapsulated in the off-hook state of the call level, the source can either be in idle or active state. When the source is in the active state, it packetizes the information and sends it according to its procedure; this layer is called the cell level. The Figure 2.1 is a good illustration of this simple model. In the foregoing, it will be assumed that at the call level, all sources are in the off-hook state (all calls are in progress).

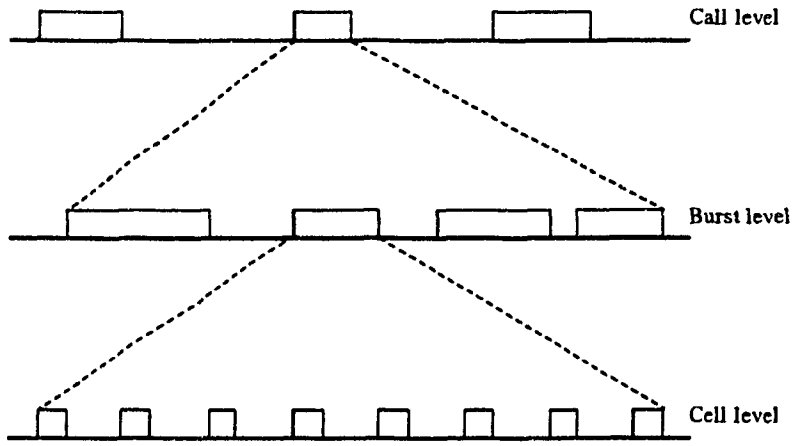


Figure 2.1: Simple ATM source model.

2.2 ATM Statistical Multiplexing and Buffering

Statistical multiplexing occurs when the capacity of a channel is less than the sum of the peak bit rates of all the VBR sources which are in progress. The statistical gain in bandwidth efficiency is the excess of the sum of the peak rates of all sources compared to the channel capacity. It should be noted that the sum of the average bit rates of all sources has to be less than the channel capacity for the system to be stable. When achieving any statistical gain, a collision phenomenon arises and a congestion control scheme must be applied. A collision occurs when several sources want to use the same resources, in this case a cell or time slot. To achieve a given QOS, buffering must be applied to reduce cell loss. On the other hand, to reduce delay of real-time traffic such as voice and video, buffering should be minimized. For example, if 10 video sources characterized by an average bit rate of 5 Mbits/sec and peak rate of 15 Mbits/sec, would share a channel with a capacity of 150 Mbits/sec, buffers for 10 cells would be enough to ensure no cell loss and maximum delay of 9 cells. However, if 50 sources with peak rates of 1, 2 or 3 Mbits/sec would share the same resources, it is not yet clear if an acceptable QOS would be obtained.

A basic premise of this study is the limited role of buffering to resolve short-term contention. The assumption of the limited buffering enables us to assume that

the correlation properties of the VBR sources will be carried out on the ATM links. These properties will be described in detail in chapter 3.

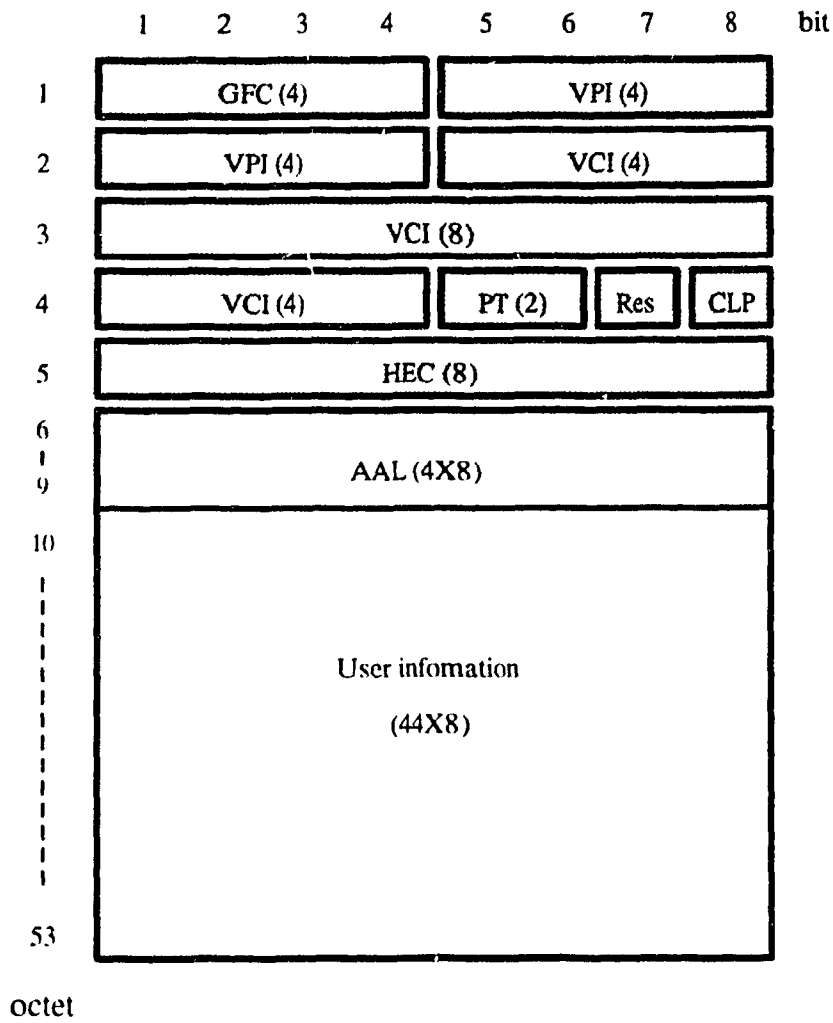
2.3 Time Slotting and Cells

The ATM channels are time slotted into fixed intervals termed as cells. The term asynchronous is related to the sources which are asynchronous in the sense that they have no common time reference other than the common channel cell. Each cell can accommodate one and only one packet of fixed size. As defined by CCITT, a cell consists of 53 octets, 5 of which are called overhead and are reserved for the network and 48 which are for the user information with respect to the ATM layer and are termed as the payload. The header of each cell contains, among others, the virtual channel and virtual path identifiers. The payload contains all the user data and ATM adaptation layer information [9]. The ATM adaptation layer is the one that has to integrate all the connection-oriented services such as VBR services. Figure 2.2 is an illustration of the cell format. For example, the PCM voice source will be packetized to form a 53 octets packets, five of which will be the ATM cell header. In the remaining 48 octets, four will be used for the ATM adaptation layer header, leaving the voice information field to be of 44 octets. Consequently, cell formation would take about 5.5 ms for a 64 kbits/sec PCM coding scheme.

In the future B-ISDN environment, the services at the user network interface are to be offered first at the basic rate of 155.52 Mbits/sec, and later at 622.08 Mbits/sec and more. Let T denote the duration of an ATM cell. Considering the basic ATM rate, the duration of a cell is

$$T = (O + P) \text{ bits } \times \frac{1}{L\frac{\text{bits}}{\text{sec}}} = \frac{(O + P)}{L} \text{ sec}, \tag{2.1}$$

where P denotes the payload size (in bits), O the overhead size and L the bit rate of the ATM link.



VCI	Virtual Channel Identifier	PT	Payload Type
VPI	Virtual Path Identifier	HEC	Header Error Control
CLP	Cell Loss Priority	Res	Reserved
GFC	Generic Flow Control	AAL	ATM Adaptation Layer

Figure 2.2: ATM cell format.

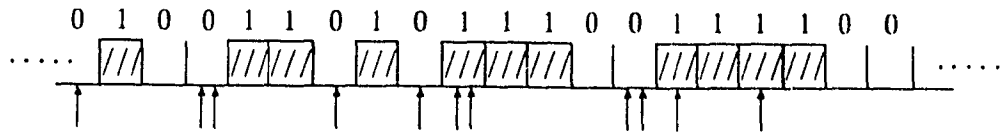


Figure 2.3: Example of ATM traffic stream.

2.4 Multiplexing of VBR Sources

In the process of estimating the number of off-hook sources from the traffic stream, all that is available is the presence or absence of a packet in a cell. From that, we can form a binary random point process denoted by

$$T(i) = \begin{cases} 0 & \text{if the cell is empty} \\ 1 & \text{if the cell is full} \end{cases} \quad i = 1, 2, 3 \ldots \tag{2.2}$$

The random process $T(i)$ is formed by the multiplexing of several VBR sources. When contention occurs, one of the colliding packet is sent to the output port while the others are kept in a contention buffer to be sent during the succeeding cells. Although the role of buffering is limited, throughout this study it is assumed that the contention buffers are large enough so that overflow can be neglected. Figure 2.3 is an illustration of the multiplexed traffic. Arrivals to the multiplexer are denoted by up-arrows and output packets (or filled cells) by hachured boxes. The sequence of zeros and ones represents the binary random process $T(i)$ for this example.

The operation of the multiplexer is non-linear and difficult to express for a signal processing approach. If no contention occurs, the multiplexer can be viewed as a simple adder. The multiplexing of several sources causes contention that are resolved by adding delays to the packets. As a first cut, the multiplexer will be considered to act as an adding device, neglecting this non-linear phenomenon. This delay phenomenon that is caused by the multiplexer buffer is often called jitter. For the simplicity of the theoretical analysis, it will be assumed that the resulting traffic process $T(i)$ is the sum of the traffic of the multiplexed sources.

Chapter 3

Source Models

In this chapter, the models of sources are presented. First, the binary Markov process for the voice source is presented. The Section 3.2 presents the VBR video source model which is gaining the most attention [10, 11, 12, 13, 14, 15, 16]. The model used is an autoregressive process for the average bit rate of a video source using a DPCM codec. The models used for the data sources are presented in Section 3.3. Bernoulli and Poisson processes are used for the data modeling. For each model presented in this chapter, theoretical cell autocorrelation function and power spectral density are derived.

3.1 Voice Modeling

The voice source modeling can be done at different levels: the call level, the burst level and the cell level as shown in Figure 2.1. At the call level, the statistics are well-known and continually used by the telephone companies to evaluate their equipment needs and to plan the expansion of their networks. We will suppose that the sources are off-hook at the call level. At the burst level, work was done in the sixties to develop Time Assigned Speech Interpolation (TASI) system for transatlantic communications. Now, this system uses digital speech interpolation (DSI). DSI means that the voice packets are generated only during active periods or talkspurts and not during idle or silence periods. This model is widely used for telephone speech modeling [15, 17, 18, 19] and is often referred as the ON-OFF source model. The ON-OFF model is the starting point to develop the voice model of this study.

At the burst level, the voice model for telephone speech has two states 0 and 1, which are respectively called idle and active state. These states represent the silences and talkspurts of the source and are assumed to be exponentially distributed with parameters λ and μ. In the idle state, no information is generated, thus no packet is sent to the network. During active periods, packets are generated at a constant

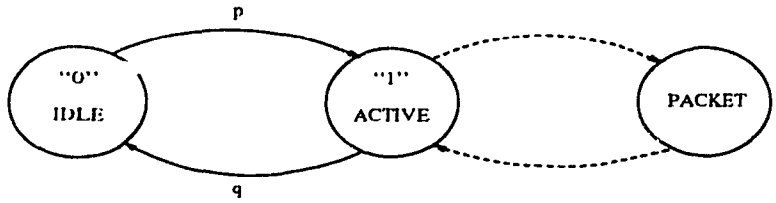


Figure 3.1: Binary Markov Process for voice source model.

rate whose value depends on the coding scheme. Therefore, the model shown in Figure 3.1 is used for the voice modeling and will be called the *binary Markov model.*

Considering a time sequence of equally-spaced embedded points, the state of the process can be observed and a Markov chain formed. Between two consecutive embedded points, there is a probability

$$p = 1 - \exp(-\lambda \Delta t) \simeq \lambda \Delta t \tag{3.1}$$

of going from idle to active state and from active to idle state of

$$q = 1 - \exp(-\mu \Delta t) \simeq \mu \Delta t. \tag{3.2}$$

Obviously, Δt is taken small enough to ensure that the probability of more than one transition is vanishingly small between two embedded points. The cell interval is considered to be a natural embedding point.

3.1.1 Autocorrelation and Power Spectral Density

In this section the autocorrelation function and power spectral density of the voice source at the cell level are derived.

Let the random point process of one source be denoted by

$$X(i) = \begin{cases} 0 & \text{if the source is idle} \\ 1 & \text{if the source is active} \end{cases} \quad \text{for } i = 1, 2, 3 \ldots \tag{3.3}$$

The process $X(i)$ is assumed to be stationary for the observation interval. The transition probability matrix of this two-state process is

$$\mathbf{Q} = \begin{pmatrix} 1-p & p \\ q & 1-q \end{pmatrix}, \tag{3.4}$$

where p and q can be obtained from Equations (3.1) and (3.2). The transient behavior of the process can be found easily,

$$\mathbf{Q}^n = \frac{1}{p+q} \cdot \begin{pmatrix} q & p \\ q & p \end{pmatrix} + \frac{(1-p-q)^n}{p+q} \cdot \begin{pmatrix} p & -p \\ -q & q \end{pmatrix}. \tag{3.5}$$

The burst autocorrelation function $r(k)$ is given by

$$\begin{aligned} r(k) &= E\left[X(i) \cdot X(i+k)\right], \\ &= \Pr\left\{X(i+k) = 1 \mid X(i) = 1\right\} \cdot \Pr\left\{X(i) = 1\right\}. \end{aligned} \tag{3.6}$$

The probabilities are easily determined with the transient behavior matrix $\mathbf{Q}^n$

$$\Pr\left\{X(i) = 1\right\} = q_{11}^{\infty} = \frac{p}{p+q}, \tag{3.7}$$

$$\Pr\left\{X(i+k) = 1 \mid X(i) = 1\right\} = q_{11}^{k}. \tag{3.8}$$

Substituting Equations (3.7) and (3.8) in Equation (3.6), the burst autocorrelation function becomes

$$r(k) = \left(\frac{p}{p+q}\right)^2 \cdot \left[1 + \frac{q}{p}(1-p-q)^{|k|}\right], \text{ for } k = 0, \pm 1, \pm 2 \ldots \tag{3.9}$$

Equation (3.9) gives the correlation between points set at equal intervals along the time axis. A value of one indicates activity, i.e. input of data to the system. This data is packaged into fixed length packets and transmitted over the line. In general, the rate at which the data is generated is slower than the line rate; accordingly, collecting data into packets involves delay, called the packetization delay.

We are interested in obtaining the autocorrelation and the spectrum for the packetized data process. As a first cut at the problem, we make a simple approximation which shows the correct general trends. A more refined analysis will be the goal of further work.

Let I denote the number of cells needed to obtain a full packet. If B is the source bit rate, P the number of information or payload and O the total overhead (cell header and AAL header) then we may write

$$I = \left\lfloor \frac{P}{B \cdot \mathcal{T}} \right\rfloor = \left\lfloor \frac{L \cdot P}{B \cdot (O + P)} \right\rfloor \text{ cells,} \tag{3.10}$$

where $\lfloor x \rfloor$ denotes the largest integer not greater than x. In general, a burst will not make up an integer number of packets; consequently, there are partially filled packets. We shall assume that these packets are transmitted at intervals of I cells. We also assume that the idle periods are long compared to I so that approximating them as being composed of units of I cells will lead to small error.

With these approximations we write the cell correlation function as

$$R(k) = \begin{cases} 0 & \text{if } k \neq n \cdot I \\ r(k)/I & \text{if } k = n \cdot I \end{cases} \quad \text{for } n = 0, \pm 1, \pm 2 \ldots \tag{3.11}$$

The factor I in (3.11) is a normalization.

The power spectral density of a point process is by definition [20]

$$P(f) = \sum_{k=-\infty}^{\infty} R(k) \cdot \exp(-j2\pi \cdot k \cdot \mathcal{T} \cdot f), \tag{3.12}$$

for $|f| \leq \frac{1}{2I}$, and where $R(k)$ is the discrete-time autocorrelation function of the process, sampled every $\mathcal{T}$ second. From Equations (3.9), (3.11) and (3.12) the power spectral density of one binary Markov process follows

$$\begin{aligned} P(f) &= \left(\frac{p}{p+q}\right)^2 \cdot \left\{ \frac{1}{I^2 \mathcal{T}} \cdot \sum_{n=-\infty}^{\infty} \delta(f - \frac{n}{I\mathcal{T}}) + \right. \\ &\quad \left. \frac{q}{Ip} \cdot \frac{1-\gamma^2}{1+\gamma^2 - 2\gamma \cos(2\pi I \mathcal{T} f)} \right\}, \end{aligned} \tag{3.13}$$

where $\gamma = (1-p-q)^I$, $p = \lambda \mathcal{T}$, $q = \mu \mathcal{T}$. $P(f)$ is periodic and should be considered only for $|f| < \frac{1}{2I\mathcal{T}}$. As one can see, the spectrum depends only on the bit rate B and the transition probabilities p and q. Another approach to compute the autocorrelation function and power spectral density using z-transform techniques is provided in Appendix A.

3.1.2 Results for M Sources

For M independent and uncorrelated sources, one can define the aggregate process

$$X_M(i) = \sum_{j=1}^{M} X_j(i), \tag{3.14}$$

where $X_j(i)$ is the cell process of a voice source, i.e. it takes the value 1 if a cell is full and 0 if the cell is empty. The autocorrelation function of $X_M(i)$ is

$$\begin{aligned} R_M(k) &= E\left[X_M(i) \cdot X_M(i+k)\right], \\ &= M \cdot R(k) + M(M-1) \cdot (E[Xj(i)])^2 . \end{aligned} \tag{3.15}$$

and the power spectral density

$$\begin{aligned} P_M(f) &= MP(f) + M(M-1)\,(E[Xj(i)])^2 \frac{1}{T} \cdot \sum_{n=-\infty}^{\infty} \delta(f - \frac{n}{T}) \\ &= M \cdot \left(\frac{p}{p+q}\right)^2 \cdot \left\{\frac{q}{Ip} \cdot \frac{1-\gamma^2}{1+\gamma^2 - 2\gamma\cos(2\pi IT f)}\right\} + \\ & M \cdot \left(\frac{p}{p+q}\right)^2 \cdot \left\{\frac{1}{I^2T} \cdot \sum_{n=-\infty}^{\infty} \delta(f - \frac{n}{IT})\right\} + \\ & M(M-1) \cdot \left(\frac{p}{p+q}\right)^2 \cdot \left\{\frac{1}{I^2T} \cdot \sum_{n=-\infty}^{\infty} \delta(f - \frac{n}{T})\right\}. \end{aligned} \tag{3.16}$$

These results are directly applicable to the voice source model. Let the transitions rates $\lambda = 1.3025$ and $\mu = 2.295$ [18] and use the parameters of the basic ATM level. The Table 3.1 summarizes all the parameters. For the set of values of Table 3.1, the theoretical autocorrelation function and power spectral density are plotted in Figures 3.2 to 3.5. Figure 3.2 is the autocorrelation function for a single voice source, and Figure 3.3 the power spectral density. The same is plot in Figures 3.4 and 3.5 for 100 sources.

We recall that the spectrum should only be considered for the frequency range bounded by $|f| < \frac{1}{2IT}$. Outside this range, a periodic repetition of this *fundamental* period, $\frac{1}{IT}$, appears, and is due to the packetization delay. In fact, the packetization delay produces correlation only between packets separated by I cells, leading to an

B	*PCM bit rate*	64 kbits/sec
O	*AAL and ATM cell header*	9 octets
P	*Voice Payload*	44 octets
L	*Basic ATM link speed*	155.52 Mbits/sec
T	*Cell duration*	2.72634 μsec
λ	*Transition rate* $(0 \rightarrow 1)$	1.3025 sec^{-1}
μ	*Transition rate* $(1 \rightarrow 0)$	2.295 sec^{-1}
p	*Transition probability*	$3.55105 \cdot 10^{-6}$
q	*Transition probability*	$6.25694 \cdot 10^{-6}$
I	*Packet interarrival*	2017 cells

Table 3.1: Set of parameters for a PCM voice source at the basic ATM level.

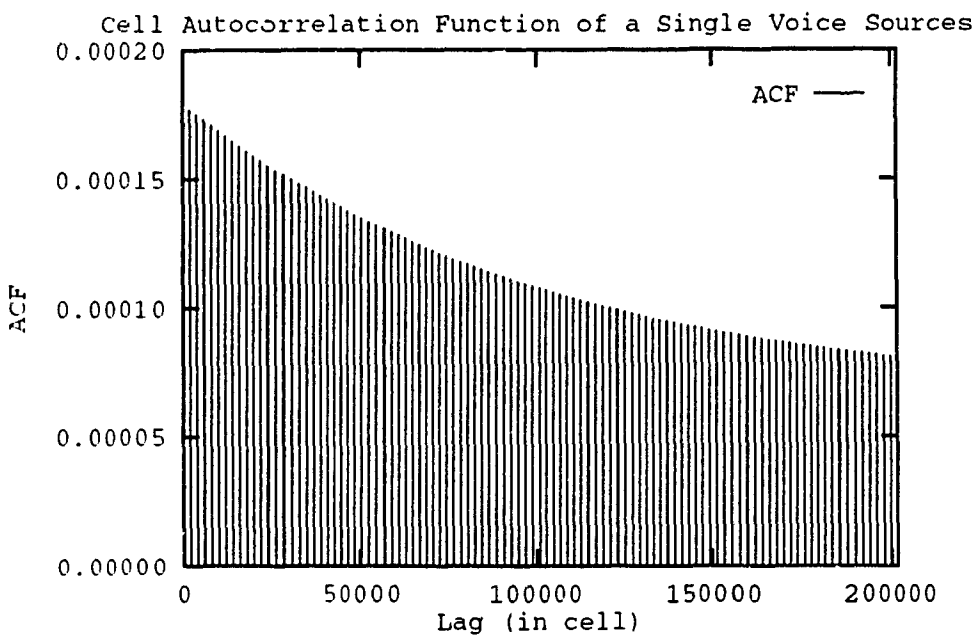


Figure 3.2: Autocorrelation function of a single voice source.

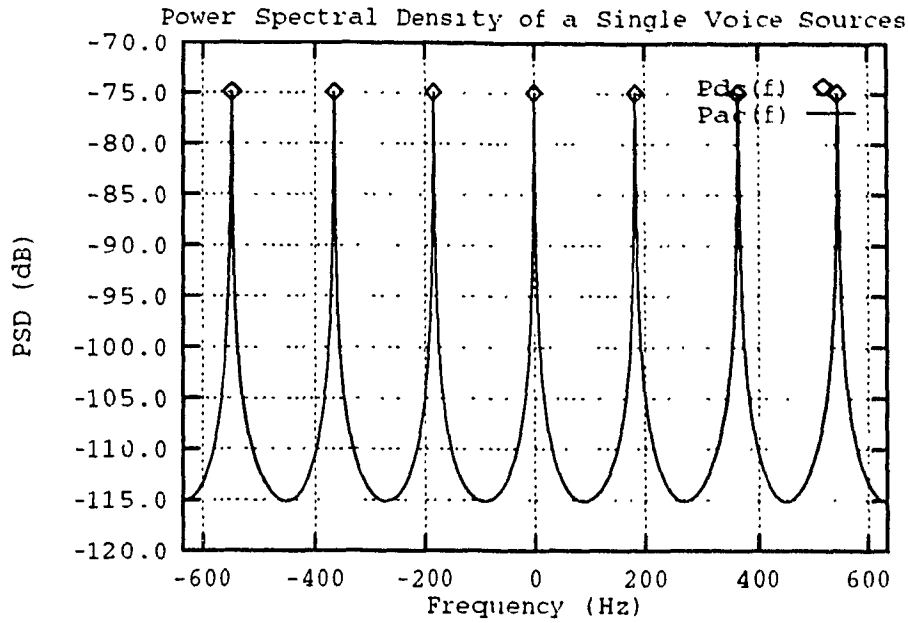


Figure 3.3: Power spectral density of single voice source.

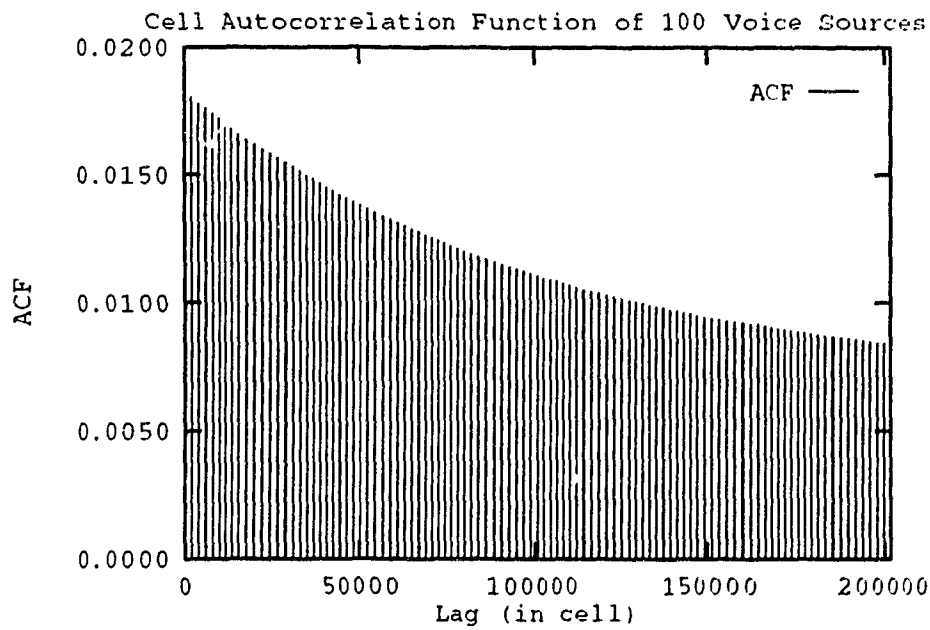


Figure 3.4: Autocorrelation function of 100 voice sources.

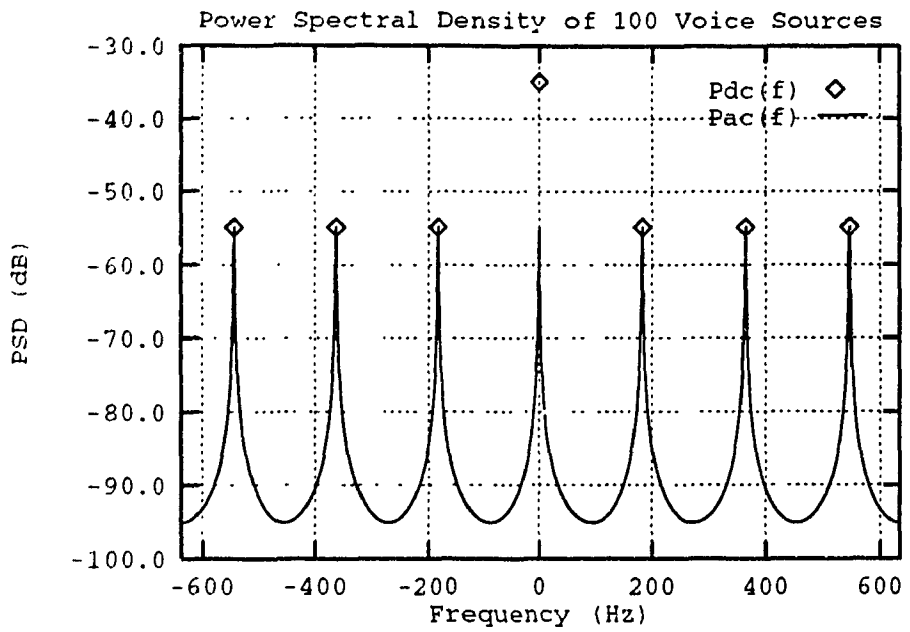


Figure 3.5: Power spectral density of 100 voice sources.

autocorrelation function with null values for all lags which are not integer multiple of I, introducing repetition of the spectrum when sampling at the cell rate. The position of the repeated spectrum, depends only on the factor I. which turns out to be independent of the transition rates λ and μ, but is a function of the source bit rate B and the transport layer specifications (cell size and link speed). Furthermore, the magnitude and the shape of the power spectral density change according to λ and μ, but the periodicity of the spectrum remains the same. In conclusion, this periodicity of the spectrum will be used to identify the voice sources.

3.2 Video Modeling

There exists several models for VBR video with the autoregressive Markov process gaining the most attention [10, 11, 12, 13, 14, 15, 16]. Section 3.2.2 presents this model. A Discrete-State, Continuous-Time Markov process proposed in [11] is also used for performance analysis of ATM networks [21]. This model is presented in section 3.2.5.

3.2.1 Parametric Modeling

Many random sequences encountered in engineering can be approximated by a linear difference equation of the form

$$x(n) = \sum_{i=1}^{p} a_i x(n-i) + \sum_{j=0}^{q} b_j \omega(n-j), \tag{3.17}$$

where $a_p \neq 0$, $b_q \neq 0$ and $\omega(n)$ is a white Gaussian noise sequence. This general linear model is termed as an autoregressive moving average process of order p and q (ARMA(p,q)). The first part of Equation (3.17) is the autoregressive branch. The second part is the moving-average branch which gives rise to the random nature of the observed process $x(n)$. For the issue of parametric modeling, the reader is referred to the texts of Box and Jenkins [22] and Priestley [23].

3.2.2 Autoregressive Markov Model

An autoregressive (AR) Markov process is a first-order AR process, or AR(1). A discrete AR(1) process $x(n)$ is governed by an equation of the form

$$x(n) = a\,x(n-1) + \omega(n), \tag{3.18}$$

where a is the AR parameter which is constant, and $\omega(n)$ is an independent Gaussian random variable.

Let $\lambda_{br}(n)$ denote the average bit rate of the nth video frame [11]

$$\lambda_{br}(n) = a\,\lambda_{br}(n-1) + \omega_{br}(n), \tag{3.19}$$

where $\omega_{br}(n)$ is a Gaussian random sequence ($\mathcal{N}(\mu_{br}, \sigma_{br}^2)$). The steady-state of $\lambda_{br}(n)$ is Gaussian and achieved when n is sufficiently large. The frame-to-frame average bit rate autocorrelation function is given by $r(k) = E\left[\lambda_{br}(n)\lambda_{br}(n+k)\right]$, and the autocovariance function by $c(k) = E\left[(\lambda_{br}(n) - \mu)(\lambda_{br}(n+k) - \mu)\right] = r(k) - \mu^2$, where $\mu = E[\lambda_{br}(n)]$ is the mean bit rate, and $\sigma^2 = E\left[(\lambda_{br}(n) - \mu)^2\right] = c(0)$, the variance of the bit rate.

$$\mu = E\left[\lambda_{br}(n)\right] = \frac{\mu_{br}}{1-a}, \tag{3.20}$$

$$\sigma^2 = V\left[\lambda_{br}(n)\right] = \frac{\sigma_{br}^2}{1-a^2}, \tag{3.21}$$

$$c(k) = \frac{\sigma_{br}^2}{1-a^2}\, a^{|k|}, \text{ for } k = 0, \pm 1, \pm 2 \ldots \tag{3.22}$$

$$r(k) = \frac{\sigma_{br}^2}{1-a^2}\, a^{|k|} + \frac{\mu_{br}^2}{(1-a)^2}. \tag{3.23}$$

The AR model was inferred using the autocovariance function of experimental setup [11]. The average bit rate $\lambda_{br}(t)$ was computed for each frame, considering the result as a continuous-time function since the frame period is small compared to the time scale. Assuming continuity, $\lambda_{br}(t)$ is considered as the instantaneous bit rate from which the autocovariance function $c(\tau) = E\left[\lambda_{br}(t)\lambda_{br}(t+\tau)\right] - \mu^2$ is computed. The discrete autocovariance function $c(k)$ is obtained by sampling the autocovariance function $c(\tau)$ at the frame interval, which means for $\frac{n}{T} = 30$ samples/sec. Parameters a, μ_{br} and σ_{br} are then obtained from the experimental mean ($\mu = 0.52$), variance ($\sigma^2 = 0.0532$), and autocovariance function ($c(\tau) = \sigma^2 e^{-3.9\tau}$). Table 3.2 gives the AR parameters for a sampling rate equal to the frame interval. Note that the parameters μ_{br} and σ_{br} in Table 3.2 are expressed in bit per pixel (bpp) and that 1 bpp represents 7.5 Mbits/sec.

The AR model does not take into account an important factor which is the

τ (sec^{-1})	a	σ_{br}	μ_{br}
$n/30$	0.8781	0.1108	0.063378

Table 3.2: AR parameters at the video frame interval [11].

variation of the bit rate within a frame, which is in fact the packet distribution. Resulting from this situation, two limit cases must be studied:

1. All the packets are sent as a burst at the beginning of each video frame interval.
2. The packets are sent uniformly at the mean packet rate during all the video frame duration.

The actual intraframe behavior is unknown but will be somewhere in between these two extreme cases [15]. The first model will be called as the *Burst AR* or BAR model and the second as *Uniform AR* or UAR model.

3.2.3 Burst AR model

The burst AR model for video assumes that all the information which is generated for a video frame is sent as a burst, at fixed intervals equal to the frame duration. The starting point to develop this model is Equation (3.19) for the average bit rate of the nth video frame.

Multiplying both sides of Equation (3.19) by the video frame duration T_f, the total number of bits for the nth frame is obtained. Dividing this product by the video cell payload P, the number of packets generated for the nth frame $\lambda_p(n)$ is obtained.

$$\lambda_p(n) = \frac{T_f}{P} \cdot \lambda_{br}(n) = a\,\lambda_p(n-1) + \omega_p(n). \tag{3.24}$$

Assuming that all the $\lambda_p(n)$ packets are sent as a burst, at the beginning of the frame interval, at a constant packet rate of I^{-1} packets per cell, the BAR packet process is formed. The symbol I^{-1} is used because of its inverse, I, which is the packetization

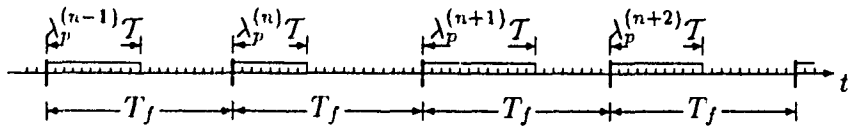


Figure 3.6: Illustration of the BAR process.

delay or packet interarrival for the voice source. Figure 3.6 is an illustration of this type process.

If the time origin is fixed at the beginning of the burst zero, then the process at the burst level can be expressed as

$$X(t) = \sum_{n=-\infty}^{\infty} f(n,t), \tag{3.25}$$

where $f(n,t)$ is the envelope of the burst of the nth frame. The function $f(n,t)$ is given by

$$f(n,t) = u_{-1}(t - nT_f) - u_{-1}(t - nT_f - \lceil \lambda_p(n) \rceil \mathcal{T}), \tag{3.26}$$

where $u_{-1}(t)$ is the unit step function, $\lceil x \rceil$ denotes the smallest integer not less than x and $\lambda_p(n)$ is given by Equation (3.24).

The process $X(t)$ will be called the envelope of the process. During a burst, the packets are sent at a constant packet rate denoted by I^{-1}. If I^{-1} is chosen equal to unity, then the burst process is identical to the cell process—once sampled. If the packet rate is chosen less than unity, then the bursts are expanded in duration by the same factor that the packet rate is reduced. Consequently, the variation of the packet rate is equivalent to a time scaling of the envelope of each burst. Therefore, the analysis is carried out with the packet rate equal to unity and rescaling is done at the end according to the video codec throughput.

Before starting the mathematical analysis of the BAR model, some details must be mentioned. Usually in the literature, autoregressive processes are driven by zero-mean Gaussian processes. In the present case, the driving random variable $\omega_p(n)$ has a nonzero mean. Moreover, there is a nonnull probability that the number of packets to be sent becomes negative; to circumvent this problem, a truncated

Gaussian distribution is used such that the burst size $\lambda_p(n)$ always remains in the closed interval $[S_-, S_+]$, where S_- and S_+ denote respectively the minimum and maximum burst size in cell.

The burst autocorrelation function is $r(t_1, t_2) = E[X(t_1)\,X(t_2)]$. Solving the problem this way is difficult due to the truncation of the Gaussian distribution, to the correlation between the bursts and to the inherent periodicity of the process. Another way to attack the problem is to take advantage of the Markov property of the AR(1) process. First let us fix the time origin t_o at the beginning the nth burst and condition on the number of video frames, k, between two bursts. Let $r(t_1, t_2 \mid t_o, k)$ denote the conditional autocorrelation function between the two bursts.

$$r(t_1, t_2 \mid t_o, k) = E[X(t_1)\,X(t_2) \mid t_o, k], \tag{3.27}$$

where

$$t_1 \in [t_o,\, t_o + T_f] \text{ and} \tag{3.28}$$

$$t_2 \in [t_o + k\,T_f,\, t_o + (k+1)\,T_f]. \tag{3.29}$$

The conditional autocorrelation function expressed by Equation (3.27) is actually the cross-correlation function of two random size bursts, k video frames apart. Further, the process is cyclostationary since its properties are identical for any translation of t_o by any integer multiple of video frame T_f. If the time origin t_o is random, this is actually the case, it can be shown that the process is stationary. Let t_1 be uniformly distributed in the closed interval $[t_o,\, t_o + T_f]$; therefore, the conditional cross-correlation function can be expressed as

$$r(\tau \mid k) = E_t[X(t)\,X(t + k\,T_f + \tau) \mid k], \tag{3.30}$$

where t is uniformly distributed in the interval $[t_o,\, t_o + T_f]$, $\tau = t_2 - t_1$, and for $|\tau| \leq T_f$.

Conditioning the Equation (3.30) on both the burst sizes at time t and at time $t + k\,T_f + \tau$, the cross-correlation function is easy to evaluate. Let $r(\tau \mid i, j, k)$ denote

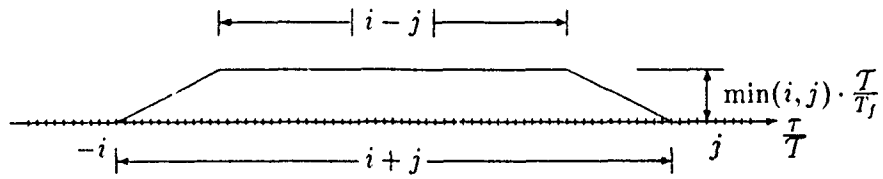


Figure 3.7: Example of conditional cross-correlation function $r(\tau \mid i,j,k)$

the conditional cross-correlation function between two bursts, the first one of size i at time t and the second one of size j at time $t + kT_f + \tau$; and, let $P_{ij}^{(k)}$ denote the transition probability of going from a burst of size i to a burst of size j after k video frames and P_i denote the steady-state probability of a burst of size i cells. Therefore,

$$r(\tau \mid k) = \sum_{i=S_-}^{S_+} \sum_{j=S_-}^{S_+} E_t\left[X(t)\, X(t + kT_f + \tau) \mid i,j,k\right] P_{ij}^{(k)} P_i, \tag{3.31}$$

$$= \sum_{i=S_-}^{S_+} \sum_{j=S_-}^{S_+} r(\tau \mid i,j,k)\, P_{ij}^{(k)} P_i, \tag{3.32}$$

where

$$P_{ij}^{(k)} = \Pr\{\lceil\lambda_p(n)\rceil = j \mid \lceil\lambda_p(n-k)\rceil = i\}, \tag{3.33}$$

$$P_i = \lim_{n\to\infty} \Pr\{\lceil\lambda_p(n)\rceil = i\}. \tag{3.34}$$

An illustration of the conditional cross-correlation function $r(\tau \mid i,j,k)$ is given in Figure 3.7. The Appendix B contains the details for the computation of the conditional cross-correlation function $r(\tau \mid i,j,k)$.

The transition probabilities $P_{ij}^{(k)}$ and the steady-state probabilities P_i are obtained from the transition matrix $\mathbf{P}$. The elements P_{ij} of $\mathbf{P}$ are obtained from the distribution of the driving noise process in Equation (3.24).

$$P_{ij} = \Pr\{\lceil\lambda_p(n)\rceil = j \mid \lceil\lambda_p(n-1)\rceil = i\}, \tag{3.35}$$

$$= \Pr\{j - a\,i - 1 < \omega_p(n) \le j - a\,i\}, \tag{3.36}$$

$$= \mathcal{Q}\left(\frac{j-1-a\,i-\mu_p}{\sigma_p}\right) - \mathcal{Q}\left(\frac{j-a\,i-\mu_p}{\sigma_p}\right). \tag{3.37}$$

where $\mathcal{Q}(\alpha)$ [24, p. 49] is a complementary error function defined by

$$\mathcal{Q}(\alpha) = \frac{1}{\sqrt{2\pi}} \int_{\alpha}^{\infty} \exp\left(\frac{-t^2}{2}\right) dt. \tag{3.38}$$

Furthermore, since $\lambda_p(n)$ is finite and limited by S_- and S_+, each element P_{ij} is normalized by dividing it by $\sum_{j=S_-}^{S_+} P_{ij}$, such that $\mathbf{P}$ is stochastic. The transition probabilities $P_{ij}^{(k)}$ can be obtained from the kth power of the matrix $\mathbf{P}$.

$$\mathbf{P}^k = \underbrace{\mathbf{P} \cdot \mathbf{P} \cdot \mathbf{P} \cdots \mathbf{P}}_{k} = \left[P_{ij}^{(k)}\right], \tag{3.39}$$

and the steady-state probabilities P_i are obtained by solving

$$\pi \mathbf{P} = \pi = [P_i]. \tag{3.40}$$

From the above equations, the cell autocorrelation function can now be derived. First lets rescale and sample the conditional cross-correlation function $r(\tau \mid k)$ according to the packet interarrival rate I^{-1}.

$$R(\tau \mid k) = \sum_{\ell=-\infty}^{\infty} r\left(\left|\frac{\tau}{I}\right| \mid k\right) \cdot \delta(\tau - \ell I T), \tag{3.41}$$

$$= \sum_{\ell=-\infty}^{\infty} r\left(|\ell T| \mid k\right) \cdot \delta(\tau - \ell I T), \tag{3.42}$$

where $\delta(\tau)$ is the dirac function or unit impulse function. Recall that $r(|\tau| \mid k)$ is time limited by $|\tau| \leq T_f$. The Equation (3.42) is simply the continuous-time representation of the point process.

Finally, to obtain the cell autocorrelation function of the BAR model, we must uncondition on k by time shifting all the cross-correlation function to their position (center at the lattice point) and add them together.

$$R(\tau) = \sum_{k=-\infty}^{\infty} R(\tau \mid k) * \delta(\tau - k T_f), \tag{3.43}$$

where the asterisk $(*)$ denotes the convolution operator.

Due to the form of the P_{ij}'s (difference of complementary error functions which are integrals), a closed form solution cannot be provided for the theoretical autocorrelation function of the BAR model. Therefore, a numerical solution has been

a	*AR(1) parameter*	0.8781
μ_p	$\omega(n)$ *mean*	45.01 packets per frame
σ_p	$\omega(n)$ *std. deviation*	78.693 packets per frame
T_f	*video frame size*	1/30 sec.
$\mathcal{T}$	*approx. ATM cell size*	1/12226 of T_f
P	*payload size*	44 octets

Table 3.3: BAR and ATM link parameters.

computed. The Figures 3.8 to 3.11 show the plots for two values of the number of cells between two consecutive packets, $I = 1$ and $I = 12$. The power spectral density has also to be computed numerically using Discrete-Fourier Transform, leading to a correlogram[1] estimator. Figures 3.12 to 3.15 show the power spectral density of the two autocorrelation functions. The parameters used for the computation are given in Table 3.3.

3.2.4 Uniform AR model

Unlike the BAR model which is sending all the packets as a burst, the uniform AR model for video sends uniformly all the information which is generated for a video frame over a fixed interval equal to the frame duration. As for the BAR model, the starting point to develop this model is the average bit rate of the nth video frame given by Equation (3.19). The method used to obtain the autocorrelation function is similar to the one for BAR model. The only difference is in the computation of the conditional cross-correlation function. Furthermore, no burst rescaling has to be done since the packets are always sent over the entire frame duration which is the maximum size that a burst can have.

Because of the nature of the packet process, the cross-correlation between two different frames is very low. Although the correlation between different frames exist, it will be assumed neglectable, hence set equal to zero. Therefore, only the

[1]See Section 1.2.2 for theory on correlogram spectral estimator.

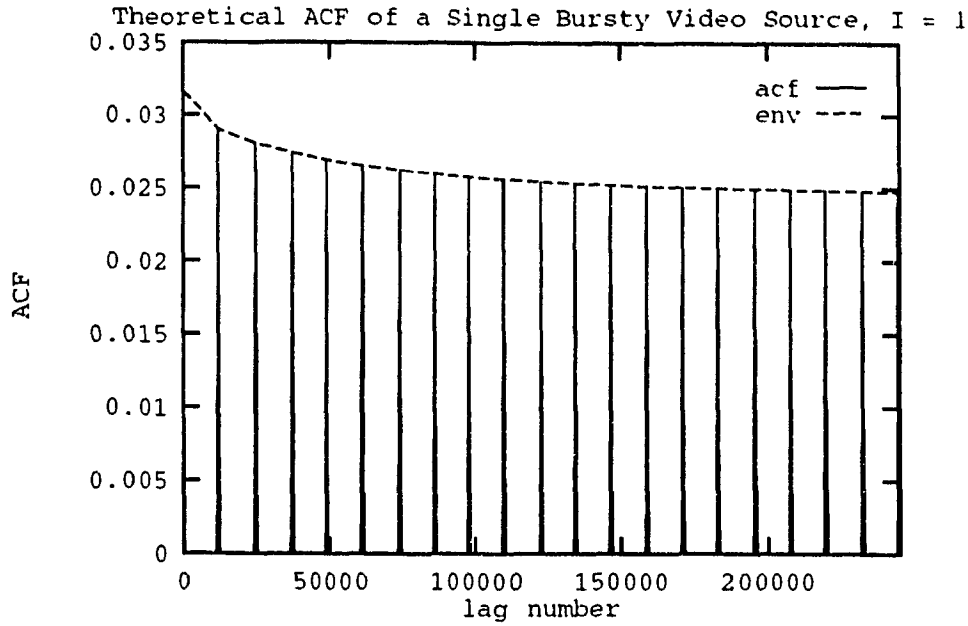


Figure 3.8: Autocorrelation function for $I = 1$.

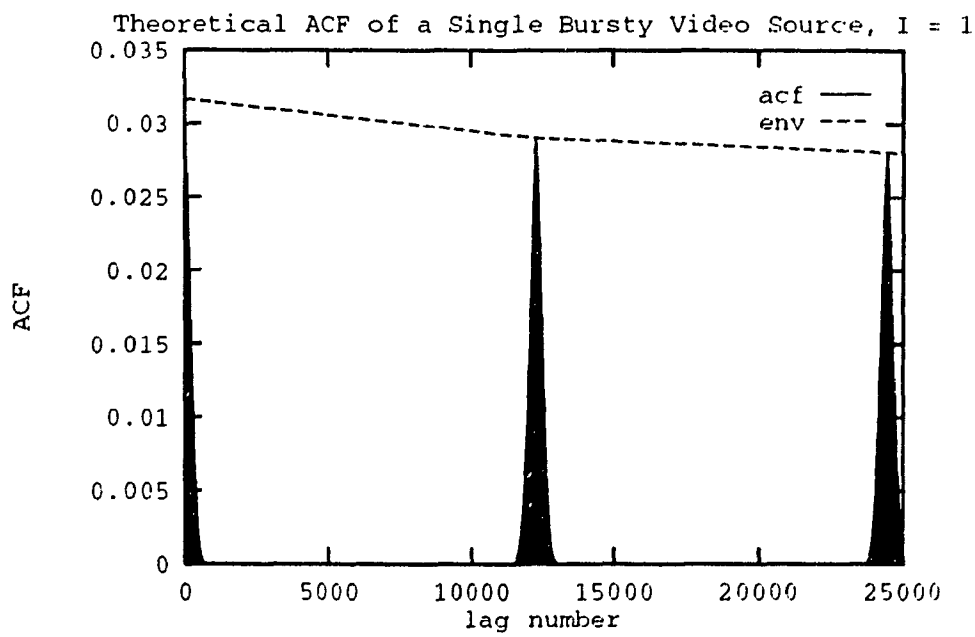


Figure 3.9: Enlargement of Figure 3.8.

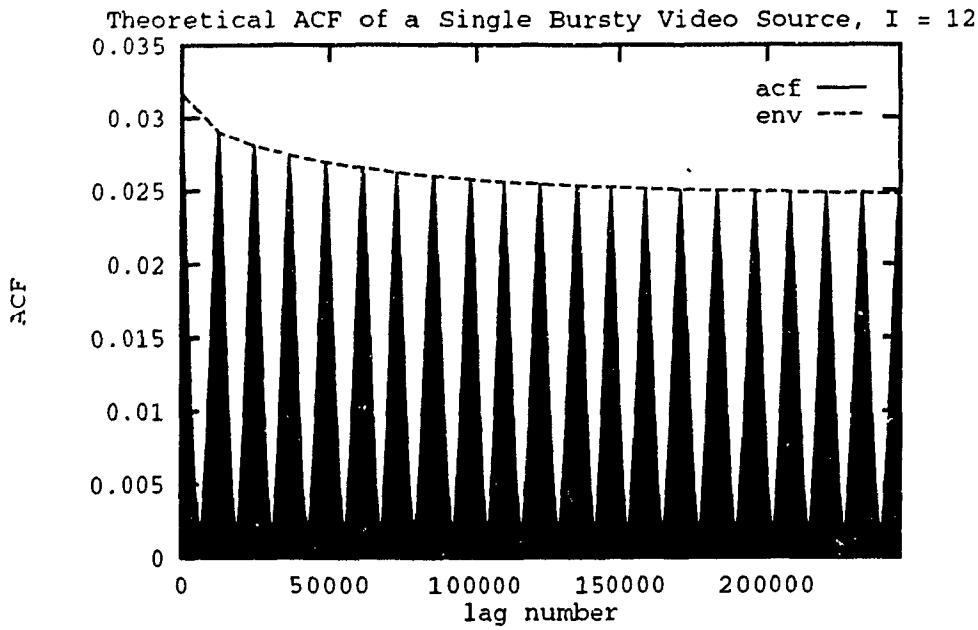


Figure 3.10: Autocorrelation function for $I = 12$.

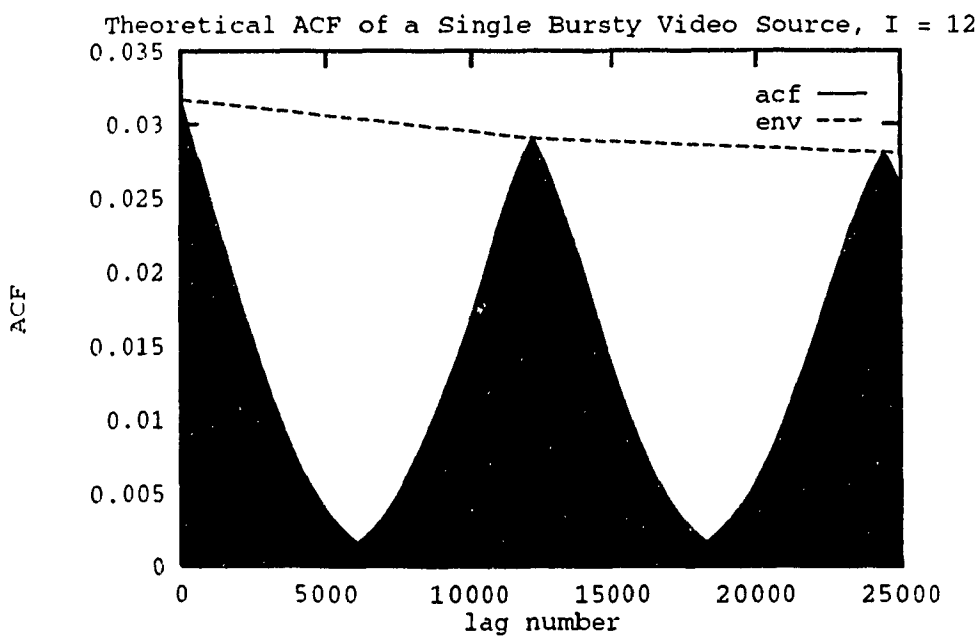


Figure 3.11: Enlargement of Figure 3.10.

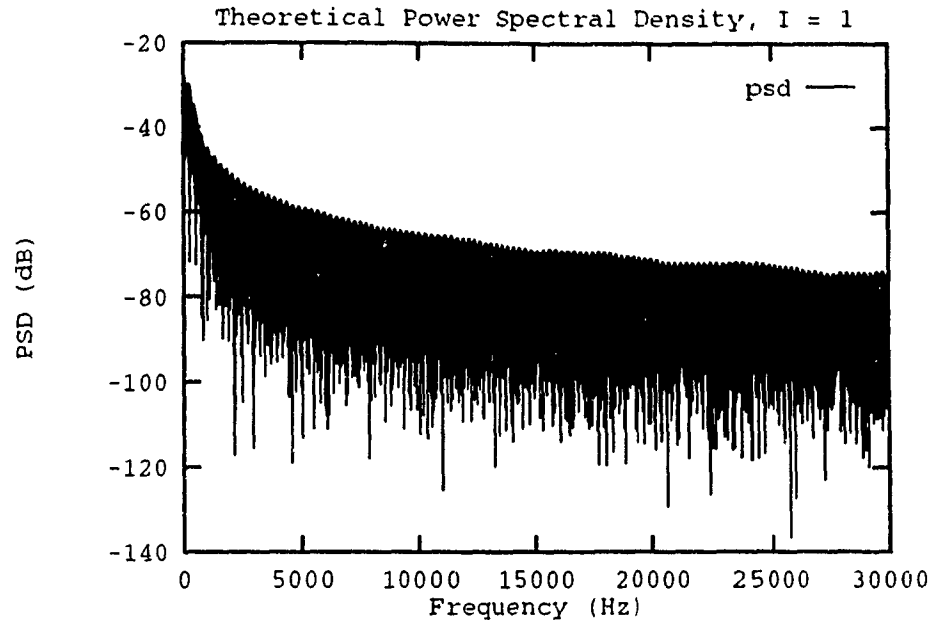


Figure 3.12: Power spectral density of the BAR model for $I = 1$.

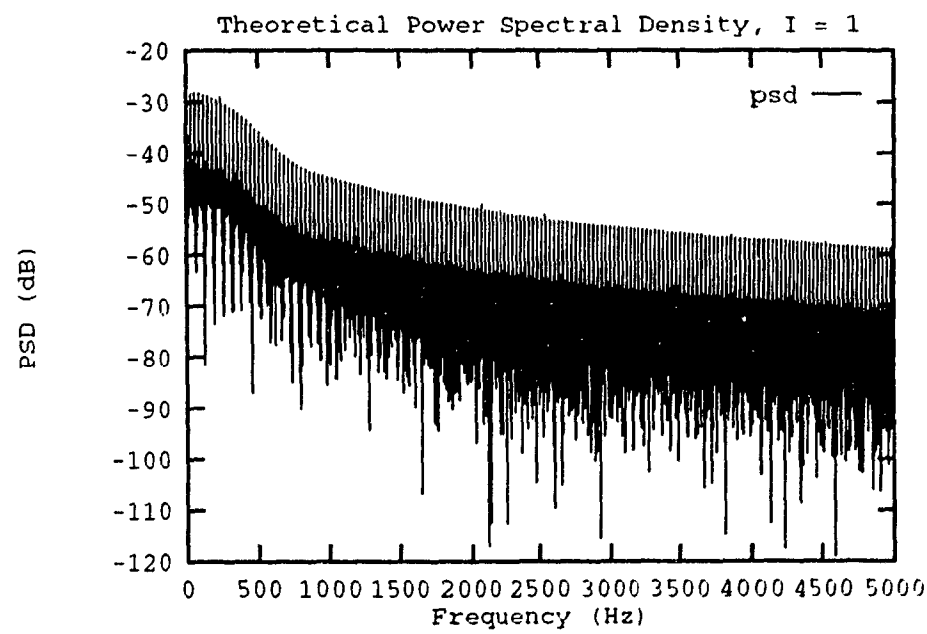


Figure 3.13: Enlargement of Figure 3.12.

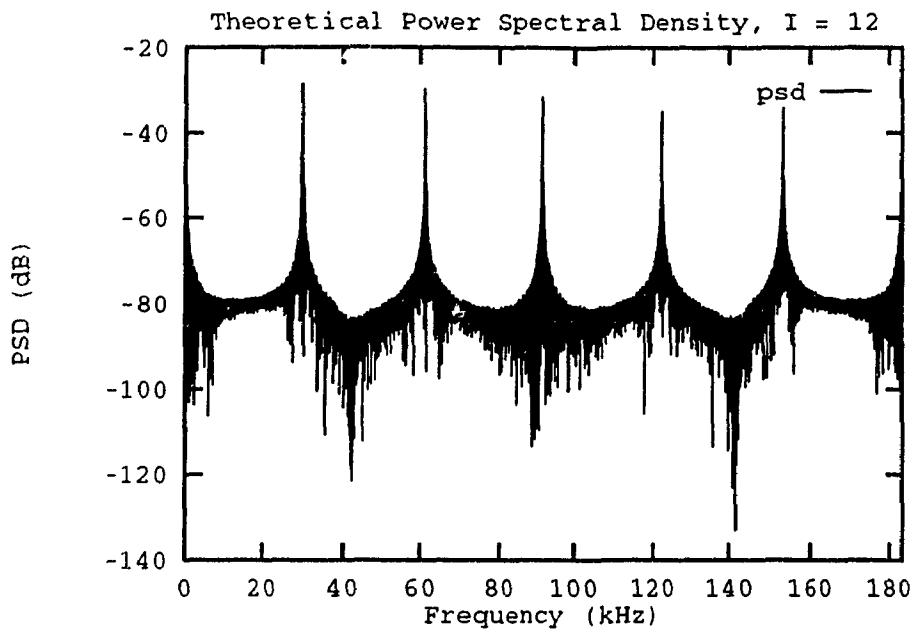


Figure 3.14: Power spectral density of the BAR model for $I = 12$.

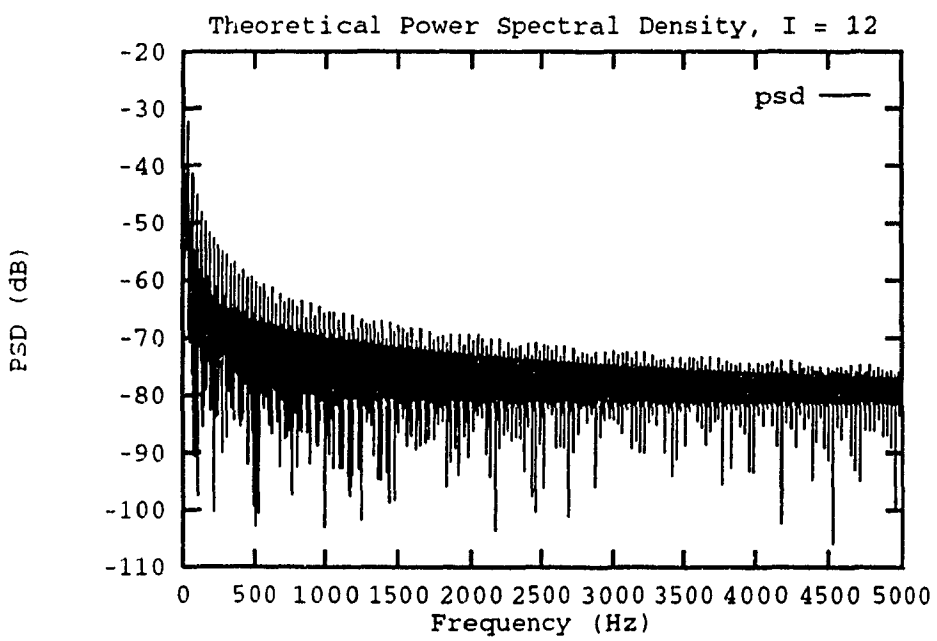


Figure 3.15: Enlargement of Figure 3.14.

autocorrelation within the interval $[0, T_f]$ will be evaluated.

The burst autocorrelation function is simply the triangle defined by

$$r(\tau) = 1 - \frac{|\tau|}{T_f}, \quad \text{for } |\tau| \leq T_f. \tag{3.44}$$

The cell autocorrelation function is easily obtained by sampling and weighting.

Let $R(\tau|i)$ denote the conditional cell autocorrelation function when there are i packets to send.

$$R(\tau|i) = \sum_{n=-i}^{i} r(\tau) \cdot \delta(\tau - \frac{n\,T_f}{i}) \cdot \frac{i\,T}{T_f}, \tag{3.45}$$

where $r(\tau)$ is the triangular burst autocorrelation function, P_i the steady-state probability of the number of packets to send and $\delta(\cdot)$ the dirac delta function.

Unconditioning on i, we obtain

$$R(\tau) = \sum_{i=S_-}^{S_+} \sum_{n=-i}^{i} r(\tau) \cdot \delta(\tau - \frac{n\,T_f}{i}) \cdot \frac{i\,T}{T_f} \cdot P_i, \tag{3.46}$$

$$= \sum_{i=S_-}^{S_+} \sum_{n=-i}^{i} \left(1 - \frac{|n|}{i}\right) \delta(\tau - \frac{n\,T_f}{i}) \cdot \frac{i\,T}{T_f} \cdot P_i, \tag{3.47}$$

where P_i is the steady-state probability of i packets to send.

The Figure 3.16 is the autocorrelation function for the parameters given in Table 3.3. Figure 3.17 is the associated power spectral density.

3.2.5 Minisources Model

The minisource model or discrete-state, continuous-time Markov model is used to model N independent video sources multiplexed on a high-speed link-- in the current case an ATM link. This model is inferred from the same experimental setup that the AR(1) model of Section 3.2.2 for the frame-to-frame average bit rate. The aggregate bit rate of the N video sources is quantized into a finite number of discrete levels (M) which represent the states of the process. Transitions between levels are assumed to occur with an exponential distribution and are state dependent. Without lost of generality, the steps between levels are chosen to be equally-spaced with increments

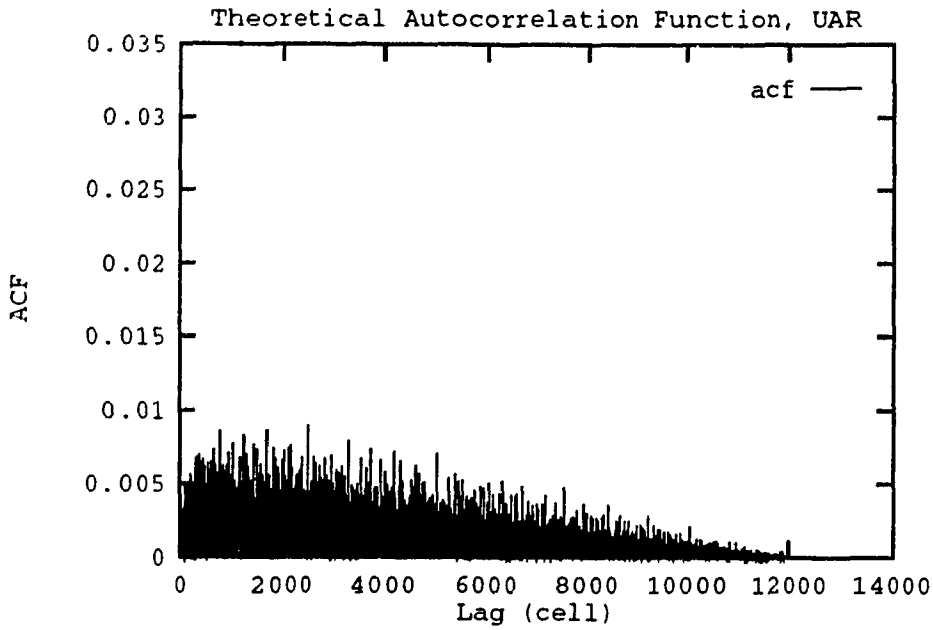


Figure 3.16: UAR model autocorrelation function.

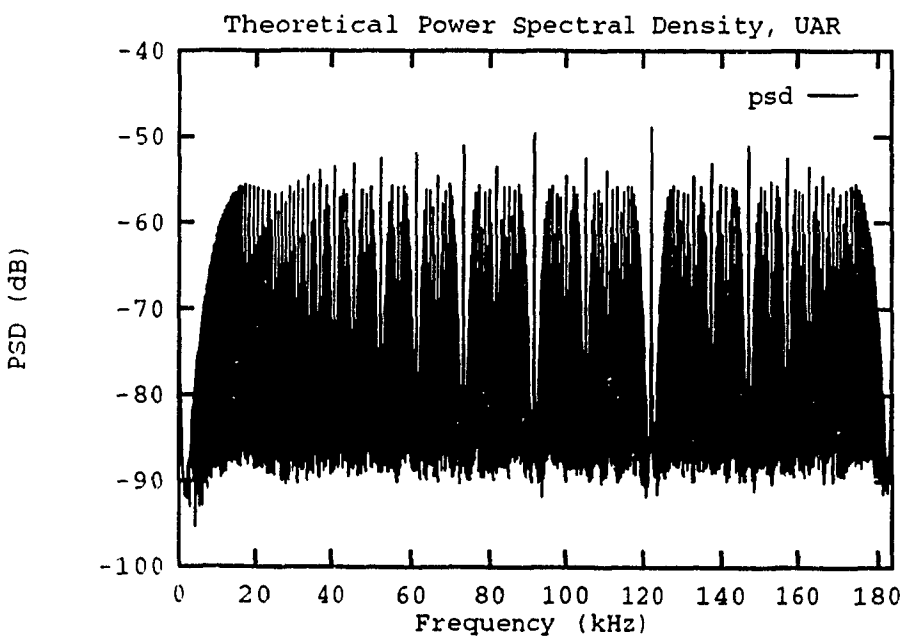


Figure 3.17: Power spectral density of the UAR model.

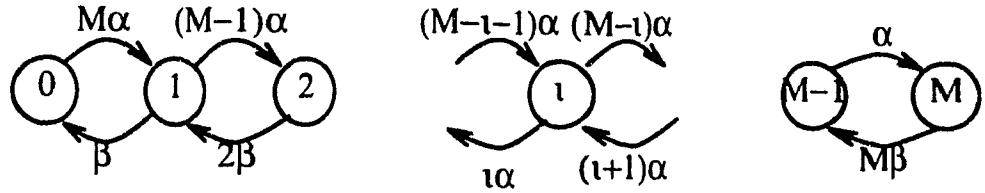


Figure 3.18: Discrete-State, Continuous-Time Markov Chain

of A bpp. Consequently, in state i, the aggregate bit rate is $i \cdot A$ bpp and the maximum aggregate bit rate is $M \cdot A$ bpp. Figure 3.18 is a representation of this Markov chain.

Like the voice model, the process is embedded at the cell interval. If the cell duration is small enough, the Markov process can be decomposed in a sum of M independent identical minisources. A minisource is a binary Markov process like voice. When it is in the active state, it generates information at a constant bit rate of A bpp. The transition rate from idle to active is α, and from active to idle β. For the modeling equivalence to hold, the cell interval $\mathcal{T}$ has to be small enough such that at most, one minisource may change of state between two embedded points. If the cell duration is not small enough, more than one transition may occur during a cell period, and the previous Markov chain cannot be decomposed into M independent minisources. To get the minisource parameters, the theoretical mean and autocovariance function have been fitted with the experimental measures of Section 3.2.2. Let κ denote the ratio of the number of minisources (M) to the number of multiplexed sources (N). For a given $\kappa = \frac{M}{N}$, α, β and A follow [11].

$$\beta = \frac{3.9}{1 + 5.04458/\kappa}, \tag{3.48}$$

$$\alpha = 3.9 - \beta, \tag{3.49}$$

$$A = 0.1 + 0.52/\kappa \text{ bits/pixel}. \tag{3.50}$$

With a maximum bit rate of 1.42 bpp for a single source, it is easy to show that $\kappa \geq 9$ spans all the bit rate intervals. For reliable results, the authors in [11] suggested to choose $\kappa \geq 10$, and that $\kappa = 20$ provides good results in performance analysis of

κ	10	20	100
α	1.3077	0.7856	0.1260
β	2.5923	3.1144	3.7127
A (bit/pixel)	0.152	0.126	0.1052
A (Mbps)	1.14	0.945	0.789

Table 3.4: Values of α, β and A for differents κ.

packet multiplexers. Table 3.4 gives the minisource parameters for different values of κ.

Like the AR(1) model for the frame-to-frame average bit rate, the intraframe bit rate behavior is unknown for the minisource model. Although this model has the same autocorrelation function, it is not interesting for power spectral estimation because its spectral characteristics depend on the ratio of the number of minisources to the number of video sources κ. In fact, as concluded previously for the BMP model, the shape and magnitude of the power spectral density depend on the bit rate A and the transition rates α and β which themselves depend on the ratio κ. Consequently, the spectrum is function of an arbitrary choice of the parameter κ. However, the minisource model may still be useful for an estimator that uses a counting process (see Section 4.4.3). The minisource model is useful for performance analysis of ATM networks that use traffic flow.

3.3 Data Modeling

Data sources may be modeled several ways. In this study, the data sources are modeled either as Bernoulli or Poisson processes. They may be viewed as background noise to the more correlated sources. The next section presents the case of the Bernoulli source and Section 3.3.2 the Poisson source. For both processes, the cell autocorrelation function and the power spectral density are given.

3.3.1 Bernoulli Model

A Bernoulli source is a source that generates packets with a probability p for every cell, and independently cell-to-cell. Let $X(i)$ denote this process.

$$X(i) = \begin{cases} 0 & \text{with probability } 1-p \\ 1 & \text{with probability } p \end{cases} \quad \text{for } i = 1, 2, 3 \ldots \tag{3.51}$$

The cell autocorrelation function is given by

$$\begin{aligned} R(t_1, t_2) &= E\left[X(t_1)\,X(t_2)\right], && (3.52) \\ &= \Pr\{X(t_1) = 1, X(t_2) = 1\}, && (3.53) \\ &= \Pr\{X(t_2) = 1 \mid X(t_1) = 1\} \cdot \Pr\{X(t_1) = 1\}, && (3.54) \end{aligned}$$

where the probability $\Pr\{X(t_1) = 1\} = p$. The conditional probability is given by

$$\Pr\{X(t_2) = 1 \mid X(t_1) = 1\} = \begin{cases} 1 & \text{if } t_1 = t_2 \\ p & \text{if } t_1 \neq t_2. \end{cases} \tag{3.55}$$

The autocorrelation function can thus be rewritten as

$$R(\tau) = p^2 + p\,(1-p)\,\delta(\tau). \tag{3.56}$$

The power spectral density is given by the Fourier transform of Equation (3.56) using the Fourier transform pairs

$$\delta(t) \Leftrightarrow 1 \text{ and } K \Leftrightarrow K\,\delta(f). \tag{3.57}$$

Hence

$$P_c(f) = p^2\,\delta(f) + p(1-p). \tag{3.58}$$

For the discrete-time version of the process, the power spectral density equals $P_c(f)$ plus all its displacements [20, p.290] leading to

$$P(f) = \frac{p^2}{T}\sum_{n=-\infty}^{\infty}\delta(f-\frac{n}{T}) + p(1-p). \tag{3.59}$$

3.3.2 Poisson Model

A Poisson source is characterized by an exponential packet interarrival time. Let $X(t)$ denote an impulse process where the time between two consecutive impulses is exponentially distributed with parameter λ.

$$X(t) = \sum_{n=-\infty}^{\infty}\delta(t-t_i), \tag{3.60}$$

where t_i is a point process with $z_i = t_i - t_{i-1}$ exponentially distributed with parameter λ. The mean of this process is λ and the autocorrelation function is given by

$$R(\tau) = \lambda^2 + \lambda\,\delta(\tau). \tag{3.61}$$

The proof of Equation (3.61) is easy and is given in [20, pp.239–240]. The power spectral density is obtained by using the Fourier transform pairs of Equation (3.57). Hence,

$$P_c(f) = \lambda^2\,\delta(f) + \lambda. \tag{3.62}$$

As for the Bernoulli source, the discrete-time version is obtained by adding $P_c(f)$ and all its displacements.

$$P(f) = \frac{\lambda^2}{T}\sum_{n=-\infty}^{\infty}\delta(f-\frac{n}{T}) + \lambda. \tag{3.63}$$

3.4 Source Characteristics

This section is an attempt to summarize the major characteristics of each type of source described in the chapter.

For the voice source, an exponential autocorrelation function was obtained. The discrete-time autocorrelation function has null values for all lags which are not an integer multiple of the packets interarrival I. Consequently, the spectrum is periodically repeated with period $1/IT$ hertz. The period is function only of the cell size, the link speed, and the source constant bit rate in active state. Finally, the shape and magnitude of the spectrum depend on the transition rates λ and μ.

For the video source two main models were derived: the former in which all the packets generated for a video frame are sent in a burst at regular interval, the latter which sends all its packets at a uniform rate during an entire video frame duration. The periodic autocorrelation function of the bursty model is weighted by an exponential function of which parameter depends only on the autoregressive parameter a. Maxima in the former autocorrelation function appear every integer multiple of the video frame duration, i.e. every one thirtieth of a second, and may be viewed like a packetization delay, leading to relatively high power for thirty hertz harmonics. Further, the constant throughput of the source during a burst leads, like for the voice source model, to nulls in the autocorrelation function, consequently, in the spectral domain, to a repetition of the spectrum. For the uniform model, the correlation function is time limited leading and the power spectral density non-repetitive. Further, the value of the autocorrelation function at lag zero is unity and smaller than 0.2 elsewhere. This leads to a spectrum which looks very similar to noise.

Concerning the data sources, both have white noise spectral density which is characterized by a flat spectrum plus a DC component characterized by an impulse at frequency zero. Due to their flat spectrum, these sources are considered as noise for the purpose of the estimation of the traffic composition. However, if the number

of voice and video sources can be estimated, their power density spectrum retrieved from the total power spectrum, then the noise spectrum and the number of data sources can be estimated.

At an internal point in the network, the traffic stream will be a random combination of all these types of sources. Since each kind of source is very different in term of periodicity, the spectrum will have to be estimated using a common basis, the cell, resulting in a spectrum where multiple periodicities will appear.

Given these characteristics, we can now extend our study to spectral estimation. The next chapter is an introduction to digital signal processing focusing on power spectral estimation.

Chapter 4

Digital Signal Processing and Spectral Analysis

This chapter is an introduction to digital signal processing. First in the Section 4.1 spectral density definition and basics are provided. Immediately following in Section 4.2 are the traditional methods such as periodogram and correlogram spectral estimators. In the present study, to efficiently use traditional estimators, data decimation must be applied; this is the topic of the Section 4.3. The modern approach of spectral analysis is presented in the Section 4.4. Then some filtering strategies which can be used are presented. Finally, the Section 4.6 is a discussion on the methods described in this chapter.

4.1 Spectral Density Definition

This section briefly summarizes the basic definitions associated with spectral analysis. First the case of a deterministic signal is considered, then the results are extended to the case of random signals.

4.1.1 Deterministic Signals

Let $x(n)$ denote the value of the signal at the time $n \cdot \Delta t$. The sequence $x(n)$ is actually the sampled version of the continuous-time function $x(t)$. In general, $x(n)$ would be considered as complex values. In the present case, only real sequences will be considered.

The discrete-time Fourier transform $X(f)$ of a sequence $x(n)$ is defined as

$$X(f) = \Delta t \sum_{n=-\infty}^{\infty} x(n) \cdot \exp\left(-j2\pi \cdot f \cdot n \cdot \Delta t\right), \tag{4.1}$$

for $|f| \leq \frac{1}{2\Delta t}$. The energy spectral density $S(f)$ is

$$S(f) = |X(f)|^2 . \tag{4.2}$$

A sufficient condition for Equations (4.1) and (4.2) to exist is that the sequence is of finite energy or

$$\mathcal{E} = \sum_{n=-\infty}^{\infty} |x(n)|^2 < \infty. \tag{4.3}$$

Notice that the discrete-time Fourier transform is an approximation of the usual continuous-time Fourier transform and not the usual discrete Fourier transform. The differences are in the weighting factor Δt and in the range of summation. In the literature, the sampling interval Δt is often normalized and set equal to one. In Equation (4.1), the factor Δt ensures the conservation of the integrated area, yielding the correct energy value, as Δt leads to zero.

If the data available is of finite length, say N points from 0 to $N-1$, the familiar discrete Fourier transform can be developed. Let $X(m)$ denote the value of the Fourier transform at the frequency $f_m = m\Delta f$ for m ranging from 0 to $N-1$, and where $\Delta f = \frac{1}{N\Delta t}$. Hence,

$$X(f_m) = \Delta t \sum_{n=0}^{N-1} x(n) \cdot \exp(-j2\pi \cdot n \cdot \Delta t \cdot m\Delta f), \tag{4.4}$$

$$= \Delta t \sum_{n=0}^{N-1} x(n) \cdot \exp(-j2\pi mn/N). \tag{4.5}$$

The inverse discrete Fourier transform is given by

$$x(n) = \Delta f \sum_{m=0}^{N-1} X(f_m) \cdot \exp(+j2\pi mn/N). \tag{4.6}$$

Both Equations (4.5) and (4.6) yield to a cyclic representation of the function with period N, even though the original sequence was not periodic. Similarly to the case of infinite data length, a discrete energy spectral density can be defined as

$$S(f_m) = |X(f_m)|^2. \tag{4.7}$$

It is interesting to notice that when both discrete and continuous energy spectral density are evaluated at the same frequency, say $f_m = \frac{m}{N\Delta t}$, they do not yield the same value. In fact, when we take only a finite length data interval, we are multiplying the original infinite data sequence with a rectangular window. The multiplication of the data with this window yields in the spectral domain to a convolution between the true Fourier transform of the data with a sinc function. Thus, the discrete version of $X(f)$ yields to a distorted energy spectral density $S(f)$.

4.1.2 Random signals

When analyzing stochastic processes, a different point of view must be taken. Rather than being energy signals, random signals are usually power signal (infinite energy). The starting point is the Wiener-Khinchin theorem that relates the autocorrelation function of a stationary process to the power spectral density. Throughout, it will be assumed that the random processes are stationary such that the autocorrelation function depends only on the time separation τ between the two sampled random variables.

Let $R_{xx}(\tau)$ denote the continuous-time autocorrelation function of the real-valued random process $x(t)$.

$$R_{xx}(\tau) = E\left[x(t) \cdot x(t+\tau)\right]. \tag{4.8}$$

The power spectral density is obtained by taking the Fourier transform

$$P(f) = \int_{-\infty}^{\infty} R_{xx}(\tau) \cdot \exp\left(-j2\pi f\tau\right) d\tau. \tag{4.9}$$

A practical problem is that one does not usually measure the statistical ensemble properties of a process which leads to the statistic properties but only with a single time function. Consequently ergodicity must be assumed. The ergodicity of a process means that the ensemble properties are equal to the time average properties of any single sample function. The ergodicity assumption enables one to substitute the time autocorrelation function to the statistical autocorrelation function and similarly with the means. It can be proven that if the averaging is made over an infinite time interval, the statistical and time properties equal. Therefore, the autocorrelation function can be computed as

$$R_{xx}(\tau) = \lim_{T\to\infty} \frac{1}{2T} \int_{-T}^{T} x(t)x(t+\tau)dt, \tag{4.10}$$

and the power spectral density as

$$P(f) = \lim_{T\to\infty} E\left[\frac{1}{2T}\left|\int_{-T}^{T} x(t)\exp\left(-j2\pi ft\right)dt\right|^2\right]. \tag{4.11}$$

The two pairs of equations ((4.8), (4.9)) and ((4.10), (4.11)) are the basis of the traditional approaches in spectral estimation. However, in both cases, the expectation operator which is an ensemble average, is replaced by a time average. The first pair yields to the correlogram method while the second yields to the periodogram power spectral estimator.

4.2 Traditional Estimators

The traditional spectral estimation techniques are based on Fourier analysis. Two major approaches exist:

1. An indirect approach that needs first to estimate the autocorrelation function coefficients from the data before applying the Fourier transform to obtain the power spectral density estimator.

2. A direct approach that directly applies the Fourier transform to the data sequence and then squares the magnitude of the output to obtain the power spectral density estimator.

The first approach is called the correlogram or Blackman-Tukey spectral estimator since it was popularized by them in 1958. The second approach is older than the correlogram and called the periodogram because it was originally proposed to find hidden periodicities. The periodogram was misunderstood and computationally very demanding such that it was left aside until 1965 with the introduction by Cooley and Tukey of the fast Fourier transform (FFT) algorithm to efficiently evaluate the discrete Fourier transform of Equation (4.5). The periodogram is currently the most popular spectral estimator.

The next section is about the estimation of the autocorrelation function, its coefficients will be needed for the correlogram estimator which is presented in Section 4.2.2. The periodogram estimator is presented in the Section 4.2.3.

For further details on classical spectral estimation, the reader is referred to the excellent text of Jenkins and Watts [25]. For good summaries, refer to Kay [26], Marple [27] or Shanmugan and Breipohl [28].

4.2.1 Autocorrelation Function Estimation

The autocorrelation function for an ergodic random signal was defined by Equation (4.10). Assuming a finite number N of data samples, an estimate for the autocorrelation function is easily obtained by discretizing Equation (4.10). Let $\hat{R}(k)$ denote such estimator.

$$\hat{R}(k) = \frac{1}{N - |k|} \sum_{n=0}^{N-|k|-1} x(n) \cdot x(n + |k|). \tag{4.12}$$

According to the symmetric property of the autocorrelation function of real processes, the negative coefficients are obtained from the positive ones $\hat{R}(-k) = \hat{R}(k)$.

This estimator has the following properties:

1. It is unbiased since $E[\hat{R}(k)] = R(k)$.

2. If the process is Gaussian and real, then for $N \gg k$ the variance is approximately equal to

$$V\left[\hat{R}(k)\right] \simeq \frac{N}{(N - |k|)^2} \sum_{n=-\infty}^{\infty} \left(R^2(n) + R(n+k) + R(n-k)\right). \tag{4.13}$$

It is interesting to note that the variance increases for the high-order coefficients. This is caused by the fewer number of lag products averaged [26]. One way to reduce this problem is to weight the coefficients by the use of a *lag window*. The lag window will be introduced in Section 4.2.2 with the correlogram spectral estimator.

An alternative for the autocorrelation function estimate suggested by Blackman and Tukey [29] is

$$\hat{R}'(k) = \frac{1}{N} \sum_{n=0}^{N-|k|-1} x(x) \cdot x(n + |k|). \tag{4.14}$$

Unlike the previous autocorrelation estimator defined by Equation (4.12), this estimate is biased and differs to previous one by a weighting factor. The weighting factor can be viewed as a triangular lag window (also known as the Bartlett window) weighting the unbiased autocorrelation function coefficients.

$$\hat{R}'(k) = \frac{N - |k|}{N} \cdot \hat{R}(k), \tag{4.15}$$

$$= w(k) \cdot \hat{R}(k), \tag{4.16}$$

where

$$w(k) = \begin{cases} 1 - \frac{|k|}{N} & \text{for } |k| < N, \\ 0 & \text{for } |k| > N. \end{cases} \tag{4.17}$$

Like the previous estimator, the bias and variance can be evaluated. The bias is

$$E\left[\hat{R}'(k)\right] = w(k)R(k), \tag{4.18}$$

and as for the unbiased estimator, if N is much greater than k, the variance is approximately equal to

$$V\left[\hat{R}'(k)\right] = (w(k))^2 \cdot V\left[\hat{R}(k)\right], \tag{4.19}$$

$$\simeq \frac{1}{N} \sum_{n=-\infty}^{\infty} \left(R^2(n) + R(n+k) + R(n-k)\right). \tag{4.20}$$

For typical applications, the biased estimator tends to have less mean square error than the unbiased one. Further, the unbiased estimator can yield invalid autocorrelation function coefficients in certain cases, i.e. the autocorrelation matrix is not positive definite, and a solution cannot be obtained. Unlike the unbiased estimator, the biased estimator always yields a valid autocorrelation function. For all the reasons enumerated, the biased estimator is often the preferred one [30].

4.2.2 The Correlogram Spectral Estimator

The Wiener-Khinchin theorem leads naturally to the correlogram, also called the Blackman-Tukey spectral estimator. The correlogram is the discrete Fourier transform of the windowed autocorrelation function coefficient estimates. Hence,

$$\hat{P}_{BT}(f) = \mathcal{F}\left\{w(k) \cdot \hat{R}(k)\right\}, \tag{4.21}$$

where $w(k)$ is a lag window, $\hat{R}(k)$ the kth autocorrelation function coefficient estimate and $\mathcal{F}\{\cdot\}$ the Fourier transform operator.

Taking into account the limited number of autocorrelation function coefficients, Equation (4.21) can be rewritten as

$$\hat{P}_{BT}(f) = \Delta t \sum_{k=-M}^{M} \hat{R}(k) \cdot w(k) \cdot \exp\left(-j2\pi f k \Delta t\right), \tag{4.22}$$

for $|f| \leq \frac{1}{2\Delta t}$, and where $\hat{R}(k)$ is the unbiased estimator of Equation (4.12).

Typically, the number of autocorrelation function values computed is much less than the number of data samples N. Blackman and Tukey [29, p.11] suggested a maximum number of coefficients M, one tenth of the number of data available, $M \simeq \frac{N}{10}$. Practically, from N data points, $2M + 1$ autocorrelation function values are estimated. Then the autocorrelation function coefficients are windowed with an appropriate normalized lag window. From that, the discrete Fourier transform is taken (using FFT algorithm for efficiency) with *zero padding*.

As mentioned earlier, the correlogram is a power spectral *density* estimator; to obtain an estimate of the peak power at a given frequency, the value $\hat{P}_{BT}(m)$ obtained from the FFT must be rescaled by a factor Δf .

Lag windows

Very often, for practical reasons, we are using only a part of the true autocorrelation function which is estimated with a finite data set. This can be viewed as multiplying the true autocorrelation function with a rectangular window, or convolving the true

power spectral density with a sinc function. This leads us to estimate an average-over-frequency of the true power spectrum, and not the true spectral density itself. This phenomenon is termed as leakage and is mainly due to the sidelobes of the sinc function. To obtain a good spectral estimator of the power at a given frequency, one must use a window that concentrates the energy near the estimated frequency. Lag windows are subject to certain conditions:

1. $\omega(0) = 1$ to conserve the power.

2. $\omega(k) = 0$ for $k > M$ because $\hat{R}(k)$ is not defined for lags larger than M.

3. It is an even function.

The problem with choosing the shape of the lag window is that we would like to concentrate all the energy in the mainlobe (of the Fourier transform of the window) while keeping the sidelobes as small as possible. However, by reducing the magnitude of the sidelobes, we are increasing the width of the mainlobe, thus reducing the frequency resolution. A lot of work has been done on spectral window; but so far the windowing issue is a matter of cut and try. Good compromises are Hanning (squared cosine) and Hamming (raised cosine) windows.

Discussion about windowing can be found in several texts, for example [26, 27, 29, 31]. Harris [31] is an exhaustive compilation of such windows. Briefly, a good window selection can reduce the sidelobe leakage, but always at the price of a decrease in resolution.

4.2.3 The Periodogram Spectral Estimator

The periodogram estimator relies on the definition of the power spectral density of ergodic process. The discrete form of Equation (4.11) is

$$P(f) = \lim_{M\to\infty} E\left[\frac{1}{(2M+1)\Delta t}\left|\Delta t \sum_{n=-M}^{M} x(n)\cdot\exp(-j2\pi f n\Delta t)\right|^2\right], \qquad (4.23)$$

for $|f| \leq \frac{1}{2\Delta t}$. Neglecting the expectation operator and limiting the summation to the available data, the periodogram estimator is defined as

$$\hat{P}_{PER}(f) = \frac{1}{N\Delta t}\left|\Delta t \sum_{n=0}^{N-1} x(n) \cdot \exp\left(-j2\pi f n \Delta t\right)\right|^2, \tag{4.24}$$

again for $|f| \leq \frac{1}{2\Delta t}$. Using FFT to evaluate Equation (4.24) permits efficient computation for the N equidistant frequencies $f_m = m\Delta f$ with $\Delta f = \frac{1}{N\Delta t}$ and for m ranging from 0 to $N-1$. Recall that the discrete Fourier transform is cyclic and that our processes are real; hence, for the frequencies with m ranging from $\frac{N}{2}$ to N, we will have redundancy.

Using the discrete Fourier transform coefficients defined by Equation (4.5), we can write

$$\hat{P}_{PER}(f_m) = \frac{1}{N\Delta t}\left|X(f_m)\right|^2. \tag{4.25}$$

Again, to get the peak power at the frequency $f_m = m\Delta f$ we must multiply $\hat{P}_{PER}(f_m)$ by the factor Δf . Consequently, based on a rectangular integration, the total power of the process is

$$\mathcal{P} = \sum_{n=0}^{N-1} \hat{P}_{PER}(f_m) \cdot \Delta f. \tag{4.26}$$

Recall that use of FFT with zero padding is computationally efficient and has rendered the periodogram the most popular power spectral estimator.

Smoothed Periodogram

The lack of the expectation operator in Equation (4.24) for the periodogram yields to an inconsistent estimator, which is defined as an estimator for which the variance does not approach zero as the number of samples increase. To circumvent this instability problem, smoothing can be performed. Several methods have been proposed in the literature; the one used in this study is the Bartlett periodogram. Bartlett's approach is to average K periodograms produced by dividing the N data samples

into K disjointed data subsets of $L = \frac{N}{K}$ samples. Thus

$$\hat{P}_{PER_B}(f) = \frac{1}{K} \sum_{m=0}^{K-1} \hat{P}_{PER}^{(m)}(f), \tag{4.27}$$

where $\hat{P}_{PER}^{(m)}(f)$ is the periodogram produced from the mth data subset.

$$\hat{P}_{PER}^{(m)}(f) = \frac{1}{L\Delta t} \left| \Delta t \sum_{n=m\,L}^{(m+1)L-1} x(n) \cdot \exp\left(-j2\pi f n \Delta t\right) \right|^2 . \tag{4.28}$$

The main advantage of the smoothed periodogram is the reduction of the variance of the original periodogram by a factor proportional to the number of averaged periodograms. For further improvement of the this spectral estimator, *overlapping* of the data segments with data windowing can be done. This improved smoothed periodogram is often called the Welch periodogram. The overlapping of data segment has the advantage to increase the number of data segments, thus reducing the bias and the variance; again, at the price of resolution.

4.2.4 Summary on the Traditional Methods

Classical methods for power spectral estimation are characterized by the use of the FFT algorithm. Consequently, they are computationally very efficient. They have among others the advantage of proportionally estimating the power of sinusoids and estimating any kind of process. In contrast to modern spectral estimators which use a priori information, classical spectral estimators are model-free, which means that they do not use a priori information, and they properly estimate sinusoids.

On the other hand, the main disadvantage is due to the inherent windowing caused by finite data sequence causing the leakage and introducing distortion in the spectrum. Further, the resolution is limited by the length of the data sequence.

4.3 Data Decimation

Data decimation or downsampling is the process of reducing the sampling rate of a sequence by defining a new sequence [32]. In this project, downsampling may be useful to estimate the low-frequency part of the power spectrum. In fact, if we are using the cell interval as the sampling interval, we may oversample the process. For example, as it was shown in Section 3.1, the fundamental period of the power spectral density of the PCM voice source, at the cell level, is bandlimited at about 91 Hz. Sampling at the cell interval means to sample at a rate of about 366000 samples per second or 366 kHz. Obviously, considering Nyquist's sampling theorem, this is an oversampling of about 2000 times.

The advantage of the data decimation procedure is to conserve in low-frequency the same information as the broadband spectrum. Due to the inherent form of the buffering which smears the output process in time, a lot of information would be lost if the original sampling of the data stream was done at the final sampling rate instead of using the foregoing data decimation procedure.

In the following, for practical purposes, downsampling will be done by an integer factor. In theory decimation can be done by any factor; however for a noninteger factor, linear interpolation must previously be achieved.

Downsampling must be done carefully to avoid aliasing. Aliasing occurs when the minimum sampling rate (or Nyquist rate) is not respected. For a bandlimited process $X(f)$ where $X(f) = 0$ for $|f| > f_c$, then the sampling rate must be at least twice the cutoff frequency f_c to avoid aliasing. However, sampling can be done at a slower rate if we are willing to reduce the bandwidth of the process before sampling it. Thus, to avoid any aliasing when downsampling, prefiltering is performed by a low-pass filter with cutoff frequency at least half of the new sampling rate.

An ideal system for downsampling by a factor D would be:

1. An ideal low-pass filter with unity gain and cutoff frequency half of the new

sampling rate followed by

2. A sampling device that picks a filtered sample every D original samples.

This system is called a decimator. In practice an ideal low-pass filter cannot be implemented, but a high-order digital filter can easily be realized. To achieve a good decimation, one should have at least 50 dB loss at the new sampling rate. Furthermore, to avoid any phase distortion, reverse (or backward) filtering can be done.

4.4 Modern Approach

Modern approaches are often termed as model-based approaches in contrast with the model-free approaches described in Section 4.2. They are model-based due to their use of a priori information.

The modern approach uses a parametric description of the process. For example, only a few parameters obtained from the autocorrelation function coefficients with explicit relationships would describe completely the process. Parametric modeling in signal analysis is issued from the analysis of time series in economics [22]. Intuitively, it should be possible to obtain a better estimate of a process by the use of a priori information. The steps of the model-based approach are [26, 28]:

1. Select a parametric model.

2. Estimate the parameters of the model using the available data.

3. Substitute the estimated parameters in the relationships to obtain the estimate of the autocorrelation function and power spectral density.

The model must be chosen with care. Any inaccuracy in the model will lead to a systematic error in the spectral estimator. Hence, the performance of the estimator is intimately related to the identification techniques used to select the model.

Throughout, it will be assumed that the models to estimate are autoregressive. The Section 4.4.1 provides the theoretical relationships between the autocorrelation coefficients and the AR parameters and gives the maximum likelihood estimates of the parameters of an AR(1) process. For further details on modern spectral analysis, the reader is referred to the excellent texts of Kay [26] and Marple [27]. For an excellent summary, the reader can consult a tutorial paper that has appeared in the IEEE proceeding, co-authored by Kay and Marple [30].

4.4.1 Parametric Modeling

The linear difference equation of a discrete-time ARMA process defined by Equation (3.17) and recalled hereafter is the starting point of this section.

Let $x(n)$ denote a discrete-time ARMA(p,q)

$$x(n) = \sum_{i=1}^{p} a_i x(n-i) + \sum_{j=0}^{q} b_j \omega(n-j), \tag{4.29}$$

where $a_p \neq 0$, $b_q \neq 0$ and $\omega(n)$ is a zero mean white Gaussian noise sequence. The system transfer function between the driving noise $\omega(n)$ and the observed output $x(n)$ is

$$H(z) = \frac{B(z)}{A(z)}, \tag{4.30}$$

where $A(z)$ and $B(z)$ are the z-transforms of the AR and MA branch, respectively.

$$B(z) = \sum_{j=0}^{q} b_j z^{-j} \qquad \text{(MA part)}, \tag{4.31}$$

$$A(z) = 1 - \sum_{i=1}^{p} a_i z^{-i} \qquad \text{(AR part)}. \tag{4.32}$$

The power spectral density $P_{ARMA}(f)$ is given by

$$P_{ARMA}(f) = \sigma_\omega^2 \Delta t \cdot \left| \frac{B(z)}{A(z)} \right|^2 \Bigg|_{z=\exp(j2\pi f \Delta t)}, \tag{4.33}$$

for $|f| < \frac{1}{2\Delta t}$, where Δt is the sampling interval and σ_ω^2 the variance of the white noise driving process. Again the scaling factor Δt is there to ensure the correct value of the process when the power spectral density is integrated.

The autocorrelation function can be obtained by taking the inverse Fourier transform of the power spectral density. The explicit relationships of the AR parameters to the autocorrelation function sets place to a set of linear equations called the Yule-Walker equations. The matrix form of this set of equations, including an equation for the variance, is

$$\underbrace{\begin{bmatrix} r_{xx}(0) & r_{xx}(-1) & \dots & r_{xx}(-p) \\ r_{xx}(1) & r_{xx}(0) & \dots & r_{xx}(-(p-1)) \\ \vdots & \vdots & \ddots & \vdots \\ r_{xx}(p) & r_{xx}(p-1) & \dots & r_{xx}(0) \end{bmatrix}}_{\mathbf{R_{XX}}} \underbrace{\begin{bmatrix} 1 \\ -a_1 \\ \vdots \\ -a_p \end{bmatrix}}_{A} - \underbrace{\begin{bmatrix} \sigma_\omega^2 \\ 0 \\ \vdots \\ 0 \end{bmatrix}}_{s}. \tag{4.34}$$

From the positive definite property of the matrix $\mathbf{R_{XX}}$, which is also Toeplitz, the system may be efficiently solved using Levinson's algorithm [26]. The MA parameters, $b_0, b_1, \dots, b_q$ can be obtained by using an extended Levinson algorithm. The reader is referred to [26, p.114] for more details about the relationships between the ARMA parameters and the autocorrelation function coefficients.

From the Yule-Walker equation, it follows that for an AR(1) process which is an ARMA(1,0), that

$$r_{xx}(k) = a_1 \cdot r_{xx}(k-1), \tag{4.35}$$

$$r_{xx}(0) = \frac{\sigma_\omega^2}{1-a_1^2}. \tag{4.36}$$

Equations (4.35) and (4.36) lead to the relation between the AR parameters and the autocorrelation coefficients

$$r_{xx}(k) = \frac{\sigma_\omega^2}{1-a_1^2} \cdot (a_1)^{|k|}. \tag{4.37}$$

Recall that $r_{xx}(k) = r_{xx}(-k)$ for a real process. The power spectral density $P_{AR(1)}(f)$ of the process is related to AR parameters by

$$P_{AR(1)}(f) = \frac{\Delta t \sigma_\omega^2}{1 - 2a_1 \cos(2\pi f \Delta t) + a_1^2}, \qquad \text{for } |f| < \frac{1}{2\Delta t}. \tag{4.38}$$

Maximum Likelihood approach to the AR(1) Parameters Estimation

The probability density function of $\omega(n)$ is Gaussian. Thus the AR(1) parameters a_1 and σ_ω^2 can easily be estimated by the maximum likelihood approach [28] yielding in

$$\hat{a}_1 = \frac{\sum_{j=2}^{N} x(j) \cdot x(j-1)}{\sum_{j=2}^{N} x^2(j)} = \frac{\hat{r}_{xx}(1)}{\hat{r}_{xx}(0)}. \tag{4.39}$$

An unbiased estimator for the variance is

$$\hat{\sigma}_\omega^2 = \frac{1}{N-2} \sum_{j=2}^{N} \left[x(n) - \hat{a}_1 x(n-1)\right]^2. \tag{4.40}$$

Using the estimates given by both Equations (4.39) and (4.40) in Equation (4.38) one obtains an estimator for the power spectral density. Similarly the autocorrelation coefficients can be estimated using Equation (4.37). It is important to note that the power spectral density and the autocorrelation function estimates will be biased although the AR estimates are unbiased. Finally, the quality of the estimators will be improved by reducing the variance of the AR parameters.

4.4.2 Sum of M AR(1) Processes

The superposition of M AR(1) processes can be viewed as a system where M parallel branches excited by a common white noise process would be added; each branch having its own deterministic transfer function. To simulate the random number of sources of each type, each branch would be weighted by a unknown coefficient, c_i, that would be equal to the number of sources of its type.

The global transfer function is given by

$$H(z) = \sum_{i=1}^{M} c_i \cdot H_i(z), \tag{4.41}$$

where $H_i(z)$ is the transfer function of the ith type given by

$$H_i(z) = \frac{b_i}{1 - a_i z^{-1}}. \tag{4.42}$$

The parameters a_i and b_i are assumed to be known. The unknowns here are the weighting factors c_i's. The Equation (4.41) can be transformed to obtain a rational transfer function which is actually an ARMA(M, $M-1$) process. Given the ARMA parameters, the weighting coefficients can be computed easily.

Now, if we assume M types of AR(1) sources with unknown AR parameters, the same solution for the c_i's can be used. The estimation of these parameters is computationally intensive and is difficult to carry out in real-time. In Appendix C an example for $M = 2$ is carried out.

4.4.3 The Issue of a Counting Process

Throughout this study it was assumed that all the off-hook sources were independent and uncorrelated such that the autocorrelation functions and power spectral densities simply add in magnitude. In fact, as mentioned earlier in Section 2.4 on statistical multiplexing, when two packets arrive simultaneously (within the same cell interval) to the multiplexer one is sent while the other colliding packet is kept in a buffer to be sent during the next cell. This delay phenomenon is called jitter and reduces the reliability of the estimators.

Instead of looking to the cell process, if one divides the time axis into a set of disjoint time intervals (or windows), and then counts the number of packets in each interval, a counting process is formed. If the window size is not too large and if the original process is autoregressive, it can be assumed that the counting process is also autoregressive. To be reliable, the counting procedure must be applied to a traffic stream composed of many sources.

The counting process has the advantage of reducing the number of data samples and enables us to use the modern spectral estimators. As an example with the voice source of Section 3.1, if the window size is chosen to be 1 millisecond, and the voice source sends a packet every 11 milliseconds, the voice can now be modeled as an AR(11) process with the ten first AR coefficients null and the eleventh nonnull.

This simply means that there is correlation between packets in windows separated by 11 milliseconds. A similar approach can be used for video sources. However, for the video sources, this model may lead to a seasonal autoregressive, integrated, moving-average process where a season would be 30 cycles. This model would be equivalent to splitting every video frame into subframes and finding the statistical properties between the subframes. The issue of finding these statistical properties is an intensive research area, and will not be examined here. However, in [11], it was stated that the aggregate rate of several video sources would lead to an AR(1) process. Thus windowing at an interval less than the video frame duration means to sample the autocorrelation function at a faster rate than it was done in Section 3.2.2 and in [11].

The selection of a good window size is a matter of trial and error and practical implementation.

4.4.4 Summary on the Modern Methods

Modern methods for spectral estimation are characterized by the use of a priori information. In the modern approach, or model-based approach, one tries to fit the data to a preselected model. The AR(1) process is characterized by an exponential autocorrelation function. The superposition of M AR(1) by an ARMA(M, $M-1$) and is computationally very demanding to solve. The issue of a counting process shows that although the AR sources are not necessarily autoregressive at the cell level, the aggregate rate of many of them lead to an AR(1) process which can be estimated. Finally, the main disadvantage of these approaches is the computation load involved in finding the solution.

4.5 Filtering Approaches to Spectral Estimation

If one only needs a particular frequency component and not the entire spectrum, digital filters can be implemented to extract the needed information. For example, if a certain process has harmonics every 100 Hz, a comb filter can be implemented to retrieve all the harmonics at multiples of 100 Hz.

The filtering approaches described in the next sections have a common characteristic: the design is done by the placement of poles and zeros in the z-plane. Although several types of such filters can be designed, only notch filters and comb filters are described.

4.5.1 Notch Filter

A notch filter is a filter that contains a deep notch in its frequency response. Ideally notch filter would have a perfect null in its frequency response at frequency f_o. Notch filters are useful when specific frequency component must be eliminated.

To create a null, it suffices to introduce a pair of conjugate zeros on the unit circle at an angle of $2\pi f_o$. This results in a transfer function for the filter

$$\begin{aligned} H(z) &= (1 - \exp(-j2\pi f_o)z^{-1}) \cdot (1 - \exp(+j2\pi f_o)z^{-1}), && (4.43) \\ &= 1 - 2\cos(2\pi f_o)z^{-1} + z^{-2}. && (4.44) \end{aligned}$$

An improvement of this filter can be achieved by reducing the width of the notches. We may introduce poles at the same frequency with a weighting coefficient r. The transfer function for this resulting filter is

$$H(z) = \frac{1 - 2\cos(2\pi f_o)z^{-1} + z^{-2}}{1 - 2r\cos(2\pi f_o)z^{-1} + r^2 z^{-2}}. \qquad (4.45)$$

When the coefficient r is large (near to one), the width of the notches is reduced at the price of an increase in the transient effects. In practice, good results are achieved for $r \simeq 0.95$. The major problem with this filtering approach is that it is another trial and error method.

4.5.2 Comb Filter

A comb filter may be viewed as a combination of notch filters where nulls occur periodically. Comb filters are useful in rejection of harmonics.

The system transfer function of a filter which has spectral nulls periodically is

$$H(z) = \frac{1}{M+1} \cdot \frac{1 - z^{-L(M+1)}}{1 - z^{-L}}, \tag{4.46}$$

and the frequency response is

$$H(f) = \frac{1}{M+1} \cdot \frac{\sin(2\pi f L(M+1)/2)}{\sin(\pi f L)}. \tag{4.47}$$

Such filter has spectral nulls at normalized frequency $f = \frac{k}{L(M+1)}$ except for $k = 0, L, 2L, \ldots, ML$, and has peaks at frequency $f = \frac{k}{(M+1)}$ for $k = 0, 1, \ldots, M$.

4.5.3 Summary on Filtering

The two filters described above are in fact rejecting filters. To extract the wanted component, we need to subtract the output of the filter from the input. These filters are cheap and very easy to implement in real-time. Such filters could be designed for each type of sources, and according to the power at the output, the number of off-hook sources estimated. To facilitate the realization, to reduce the transient behavior, and to improve the stability of such filters, data decimation could be previously achieved.

4.6 The Issue of Spectral Estimation

In this chapter, several approaches for spectral estimation have been studied. Three different types of approaches can be outlined. One using the FFT algorithm with data decimation, one using a parametric description of the process with a counting process, and finally, one using digital filters to extract particular characterizing frequencies.

The first approach uses data decimation to reduce the sampling rate without damaging the spectrum in low-frequency. The periodogram and the correlogram are in this group. When using only a small part of the correlation coefficients which may be available, the correlogram leads to a spectral estimator with less resolution than the periodogram. Further, since the correlogram is a computationally demanding estimator, there is even less interest in it. However, the correlogram will still be very important to validate the simulator because that the theoretical power spectral densities of Section 3.2.2 where computed like the correlogram estimator. The second technique is the periodogram. When using the FFT algorithm to evaluate its DFT, it becomes a computationally very efficient method. To circumvent the instability problem caused by the lack of the expectation operator of the estimator, the Bartlett's approach (the averaging of several shorter periodogram) may be used. As mentioned earlier, the periodogram is at the moment the preferred approach for real-time spectral analysis.

The model-based approaches were also examined. They usually yield better results than the first ones, but at the price of intensive computation which make them unattractive in real-time signal processing. However, some of these methods may be used when the process to estimate is well understood and defined. For example, an AR(p) process at the cell level, and not an modulating AR(1) at the burst level like for the case of video. If one is ready to pay the price of waiting for its estimation results, a counting process can be formed and the parameters of a seasonal ARIMA estimated.

Finally, the third approach is simply to use digital filters to pick some particular frequencies characterizing the sources. In practice, this approach may yield good results, but it has the disadvantage of being a trial and error method for the filter design. For system implementation, this may be the fastest and most reliable method because of its simplicity.

Some of these approaches were tried by simulation. The next chapter, shows their accuracy in estimating the traffic composition.

Chapter 5

Simulation Results

Throughout the project of estimating the number of off-hook sources in an ATM network, a simulator was written to verify if simulated spectra match theoretical spectra, to observe if discrimination was possible between various type of sources, and, if so, to determine what should be the sampling parameters to produce an accurate estimation of the traffic composition.

The first point is verified in Section 5.1 with the estimation of autocorrelation function and the correlogram applied to a simulated data stream. Although the correlogram is not the most interesting power spectral estimator, it is useful to compare the power spectrum obtained from the simulated traffic stream with the theoretical results calculated in Chapter 3. Section 5.1 also shows that discrimination is possible between various type of sources, but that the estimation of the number of sources of each type is not as straightforward as expected.

Concerning the estimation of the traffic composition, particular emphasis was given to the easiest approaches to realize in practice. Such methods are the periodogram and the notch filtering approach, both preceded by a data decimation procedure. Section 5.2 shows the periodogram results. The results obtained with the filtering approach are given in Section 5.3.

The results obtained with the parametric approach, using a counting process, are shown in Section 5.4. Finally, Section 5.5 is a discussion on the different approaches tried. This chapter concerns only the simulation results. The simulation setup is given in Appendix D.

5.1 Autocorrelation Function and Correlogram

The autocorrelation function of the sources studied in Chapter 3 has been estimated using the unbiased estimator defined by Equation (4.12). Five million data samples were used to estimate each autocorrelation value. A correlogram was then computed using $M = 262144$ (2^{18}) autocorrelation values.

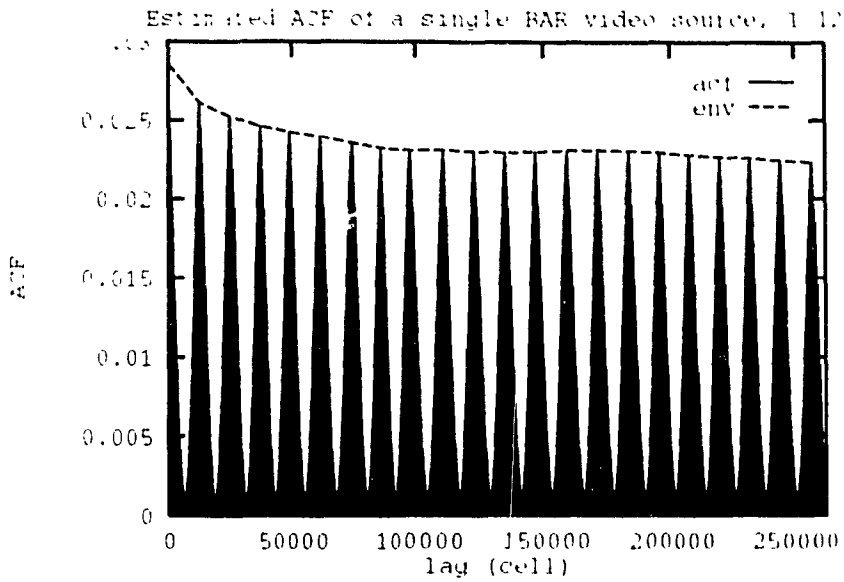


Figure 5.1: Estimated ACF of a single BAR video source.

BAR Model

The estimated autocorrelation function obtained for a single BAR video source is presented in Figure 5.1. Comparing with the theoretical autocorrelation function plot in Figure 3.10, one can see that the shape is very similar. However, there is a constant 10% error at the lattice point (1/30 second multiple). This error is presumed to be caused by the disparity between the theoretical and simulated noise processes. To obtain the burst size transition matrix of the BAR model, a truncated Gaussian distribution with higher mean and variance was used. Although this inaccuracy, when normalizing ($R(0) = 1$), both theoretical and simulated autocorrelation functions match very well. Furthermore, the exponential envelope can be fitted and the AR coefficient ($a = 0.88$) obtained.

The correlogram was computed using the previous autocorrelation estimates. Comparing with the theoretical results for the power spectral density, both spectrum shapes are very similar. However, as in the case of the autocorrelation function, comparisons of the numerical values of the magnitude show systematic error.

UAR Model

The same computations were carried out for the UAR model. The autocorrelation function obtained is plotted in Figure 5.2. A correlation value of 0.0323 at lag zero that represents the load imposed by the source to the network, is computed. Again the shape of the both, simulated and theoretical autocorrelation functions are very similar, but an error of about 10 % is present. Furthermore, for lag values greater than the video frame period there is still correlation (not zero as assumed), but small compared to the one obtained within a frame period. Although set equal to zero in Chapter 3, the correlation values obtained for lag higher than the video frame duration were expected to be equal to the square of the load imposed to the network, or, in other words, to equal square the mean value of the process.

As for the BAR model, the spectrum has a strong DC component from which the load can be evaluated. However, unlike the BAR model, a highly variable power spectral density appears elsewhere. Finally, a source similar to the UAR model is very difficult to observe in the frequency domain, especially if it is mixed with other types of sources.

Voice Sources

For the voice sources, several simulated traffic streams were used. The results of the multiplexing of 100 sources are shown. The estimated autocorrelation function is given in Figure 5.4. As expected, there is high, exponential, correlation between packets spaced by 2017 cells. Although the autocorrelation is exponential, it does not match the one calculated in Section 3.1. The expected exponential coefficient was $(1-p-q)^l \simeq 0.9801$. The one fitted to the simulated traffic is 0.9853 and is shown in Figure 5.5. A rough fitting of the estimated autocorrelation function is

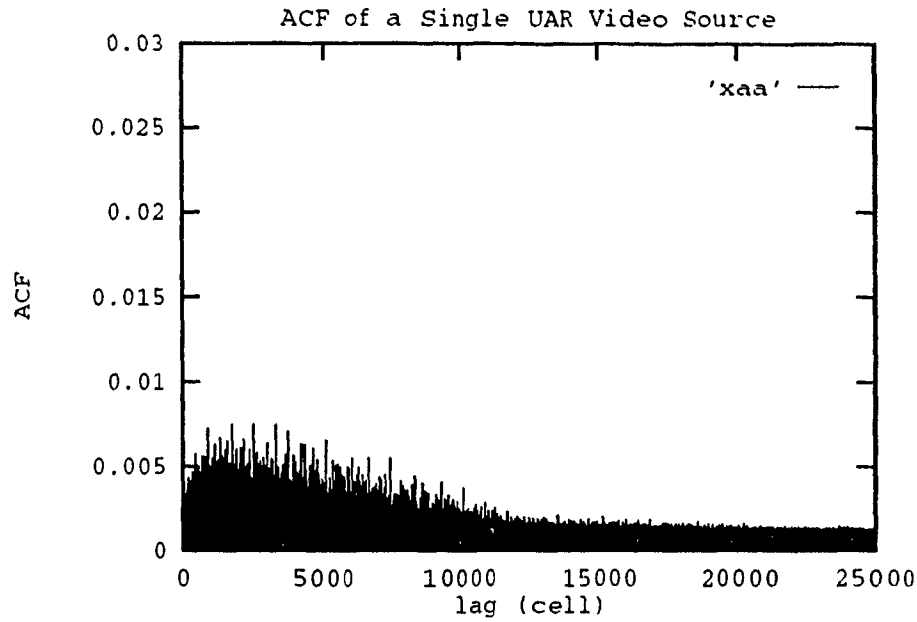


Figure 5.2: Estimated ACF of a single UAR video source.

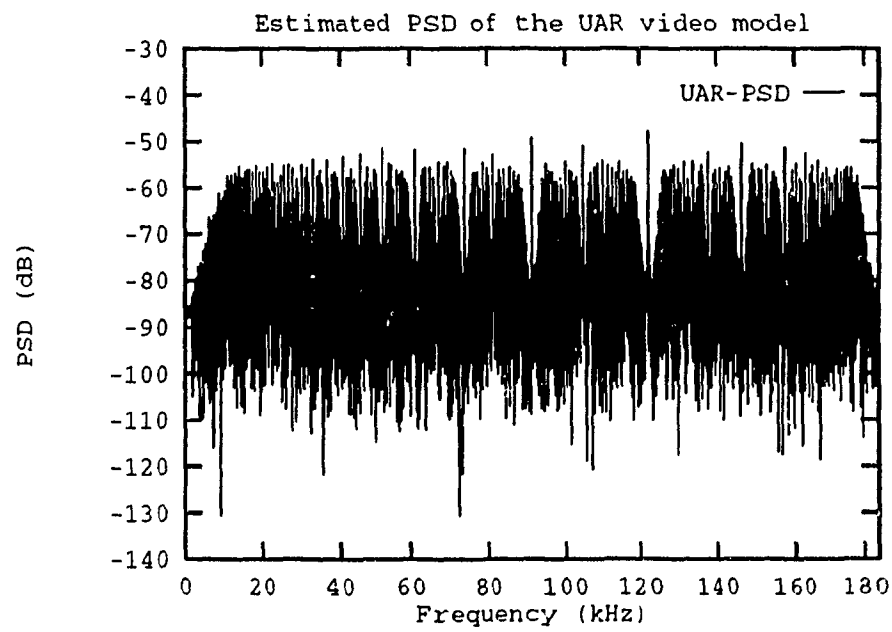


Figure 5.3: Correlogram of the UAR video source.

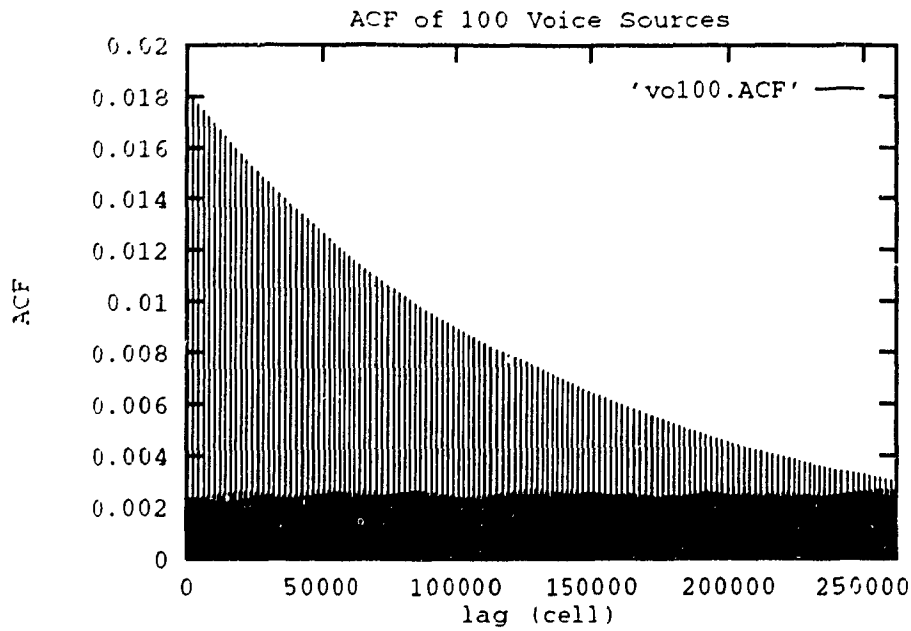


Figure 5.4: Estimated ACF of 100 multiplexed voice sources.

given by

$$\hat{R}(k) = \begin{cases} \mu^2 = 0.0003 & \text{if } k \neq n \cdot I \\ \mu^2 + 0.18 \cdot 0.9853^{k/I} & \text{if } k = n \cdot I \end{cases} \quad \text{for } n = 0, \pm 1, \pm 2 \ldots \tag{5.1}$$

Finally, Figure 5.6 shows the power spectral density obtained with the correlogram. Multiples of 181.84432 Hz were expected to have the same magnitude and equal $3.032\mu W/Hz$ or -55.18 dB. Averaging the value of the ten first peaks and comparing to the expected value, a relative error of 18.6% is obtained on the estimation of the number of voice sources.

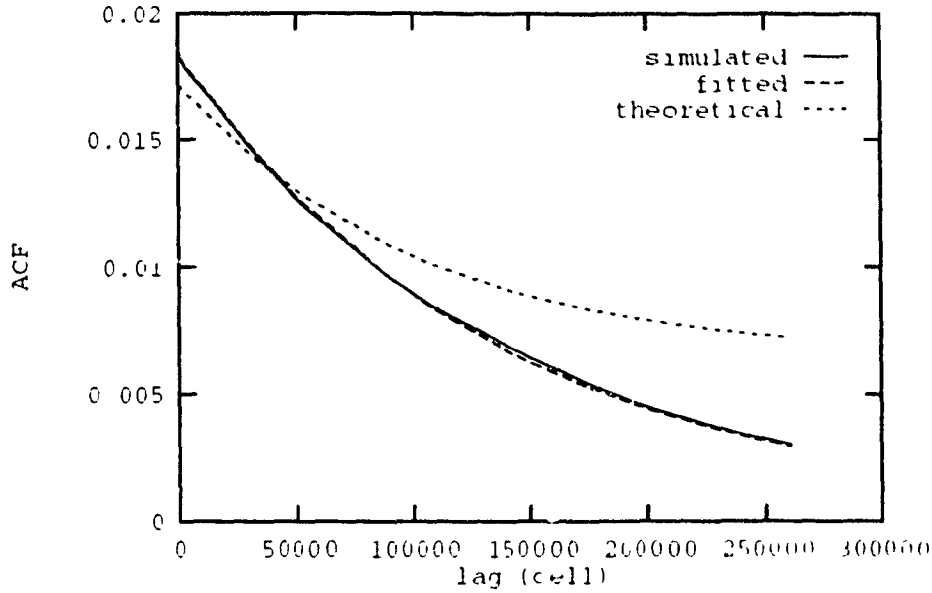


Figure 5.5: Fitting of the voice simulated traffic.

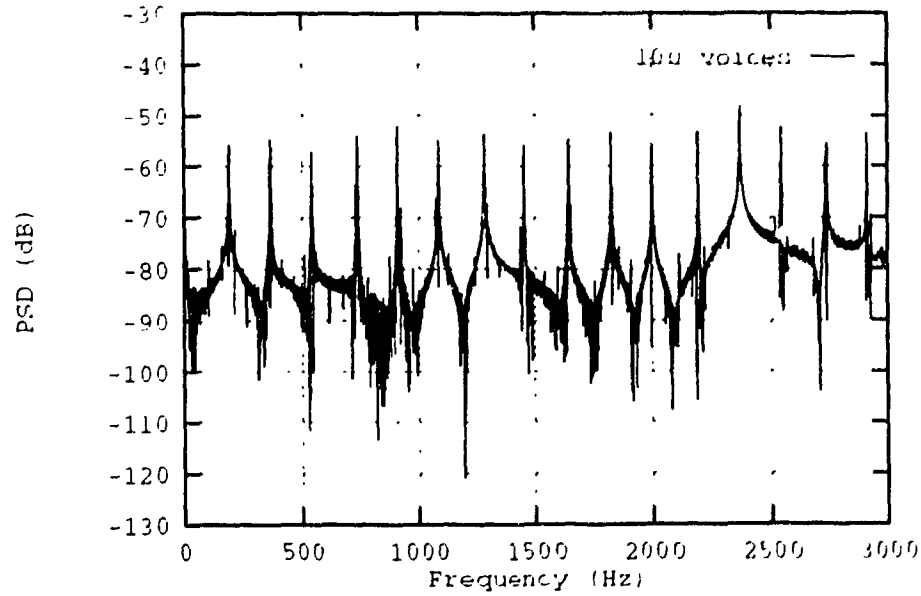


Figure 5.6: Correlogram of 100 multiplexed voice sources.

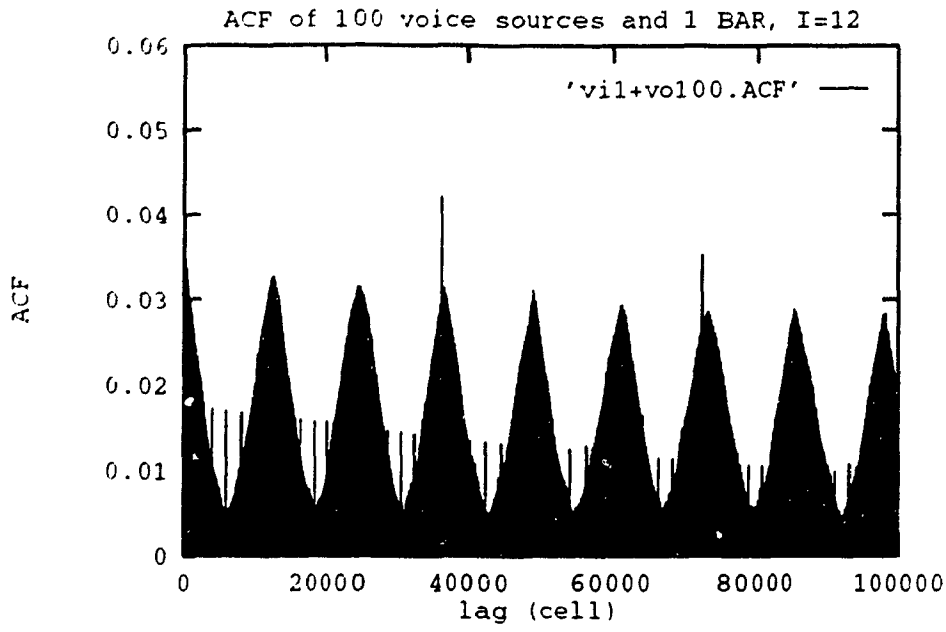


Figure 5.7: Estimated ACF of 100 voice sources and 1 BAR video source.

Mixed Traffic

A data stream composed of 100 voice sources and a single video source was simulated for this example. The autocorrelation function was computed using 1 million data samples. It yields the expected results, i.e. both autocorrelation functions, of the voice and the video, are added. As it can be seen in Figure 5.7, it is easy to discriminate between the two autocorrelation functions. The voice part has peaks every 2017 cells and the BAR video every 12226 cells.

The correlogram was computed using 65536 autocorrelation values padded with the same number of zeros. Figure 5.8 shows that both spectrum add. The magnitude of the voice peaks is about -55 dB, which is the value obtained when simulating only the voice traffic. The magnitude of the 30 Hz harmonics correspond to the one of the simulated BAR video source.

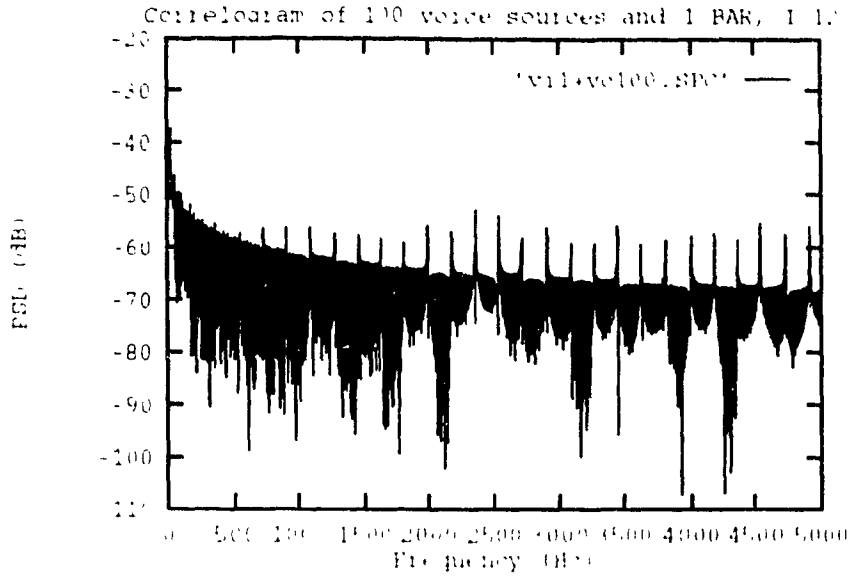


Figure 5.8: Correlogram of 100 voice sources and 1 BAR video source.

Discussion

The estimation of the autocorrelation function and the use of the correlogram to estimate the power spectral density has shown that the multiplexer is similar to an adder. High traffic load have not been extensively studied with this method due to the computation involved and the limited resources.

The positive conclusion which can be drawn from the results of simulation is that different classes of sources can be discriminated on the basis of their spectra. This may be of value in a policing situation where certain classes of sources have prohibited access to the network. However, as we shall see in later sections, spectra of independent sources do not add in a straightforward fashion.

Finally, the correlogram is not a method which is of practical interest. To get good resolution and details, a huge number of data samples are needed, resulting in heavy computational needs. However it is an interesting method when fitting a model to a process with unknown autocorrelation function.

5.2 Periodogram with Data Decimation

In this section, power spectra obtained with the periodogram preceded by a data decimation procedure are shown. The traffic streams of the previous section are used for a single class of traffic. In the following, only the results obtained for the power spectral densities estimation are shown. First the results for a single class of traffic are given. The BAR video and the voice source are used for this purpose. At the end of the section, mixed traffic streams of voice and video are simulated in order to test the estimation techniques. To obtain an accurate estimate of the spectrum in "low-frequency" involving a resonnable amount of computation, downsampling by a factor of 30 is carried out. Thus the original sampling rate of 366780 Hz is reduced to 12226 Hz. A unity gain, 8th order Butterworth digital filter with cutoff frequency (-3dB) of $0.8 \times 12226/2 = 4890.4$ Hz is used. In the sequel, if not otherwise specified, there is no data windowing done and periodograms of 1024 data samples, are smoothed. The acquisition of 1024 samples represents about 1/12 second and needs less than 3 milliseconds to be processed. This is roughly the time parameters of a practical system for monitoring telephone traffic.

BAR Video Sources

First, periodograms using 1024, 2048,..., 16384 data samples are given. In all cases, zero padding is done to get a 16384 point FFT. Further, in order to compare the estimates, the same number of data samples was used for each of them; thus 32, 16,..., 2 periodograms are averaged according to the number of data samples used for a single periodogram. Figures 5.9 to 5.13 show these estimates. The regular pattern which is characteristic of the BAR video model is obvious in all plots However the magnitudes varies according to the number of data samples used for each periodogram. This variation is caused by the intrinsic smoothing performed when computing the periodogram estimator. However, the power contained in a bin

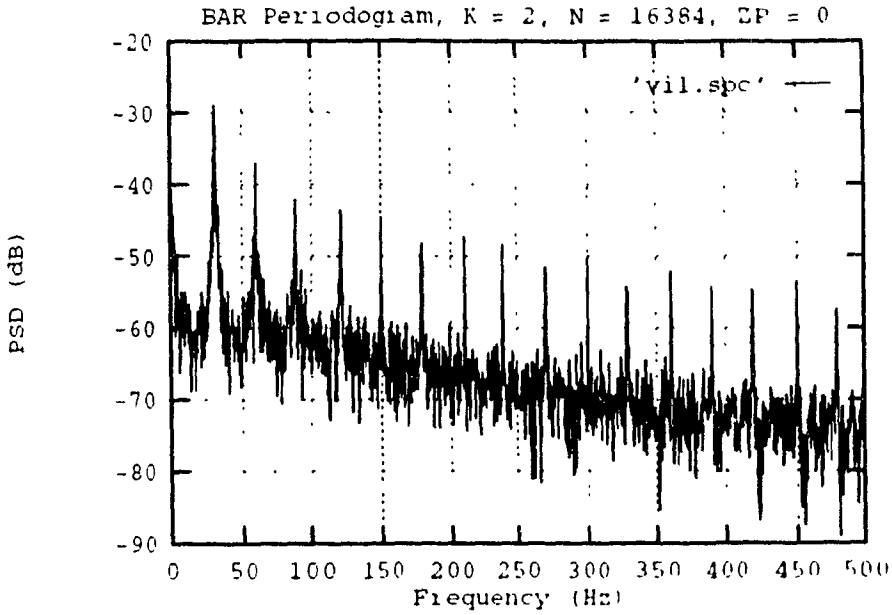


Figure 5.9: Periodogram of a single BAR video, K = 2, M = 16384.

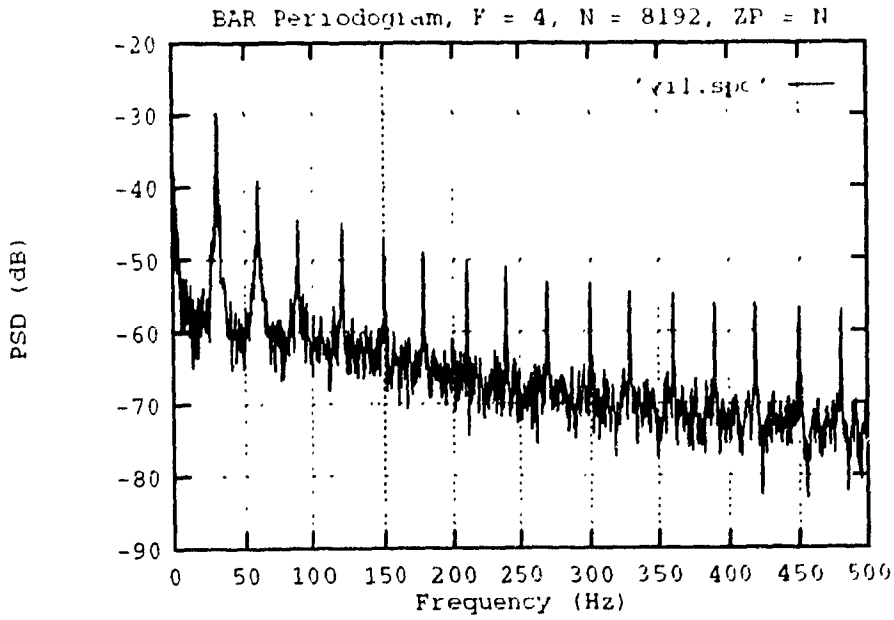


Figure 5.10: Periodogram of a single BAR video, K = 4, M = 8192.

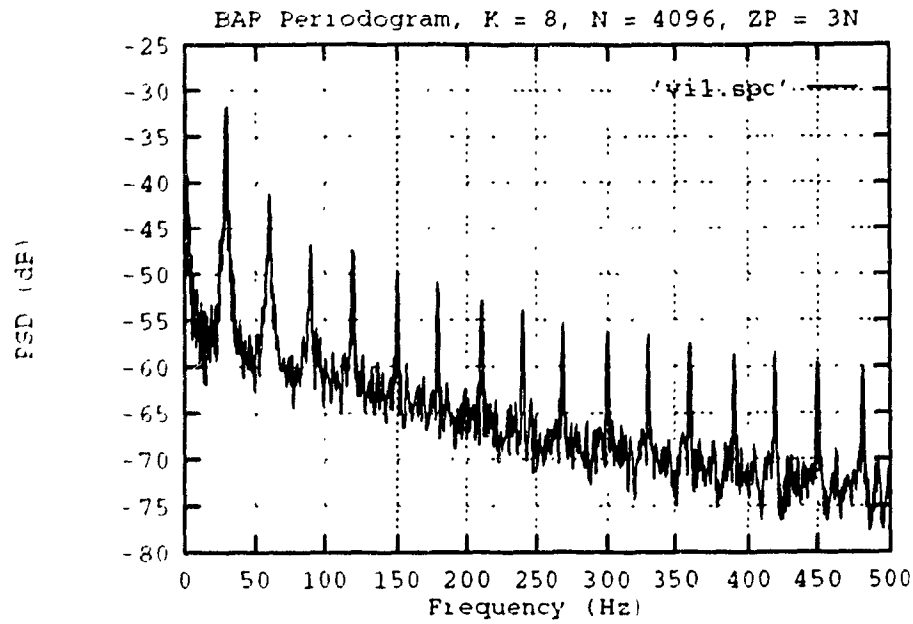


Figure 5.11: Periodogram of a single BAR video, K = 8, M = 4096.

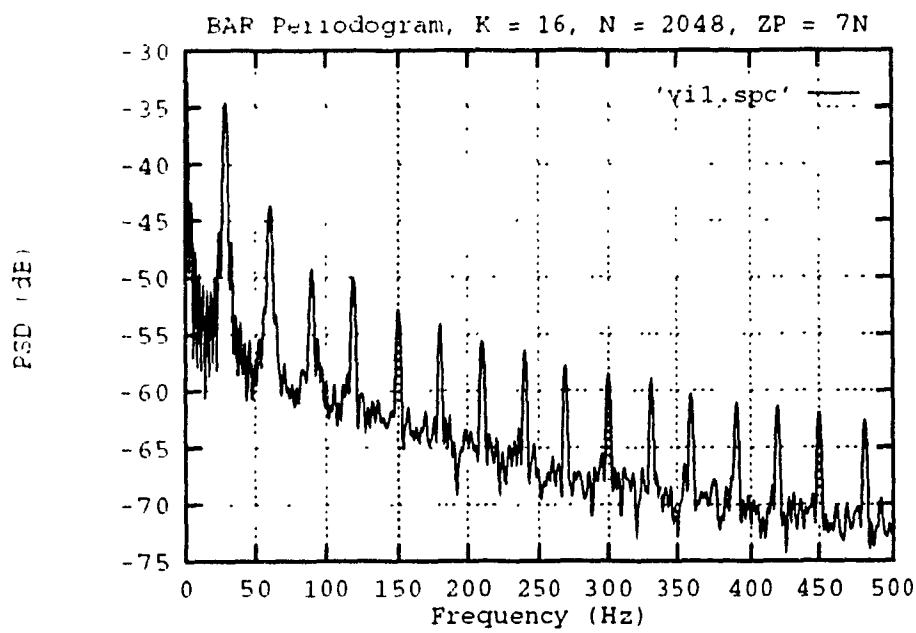


Figure 5.12: Periodogram of a single BAR video, K = 16, M = 2048.

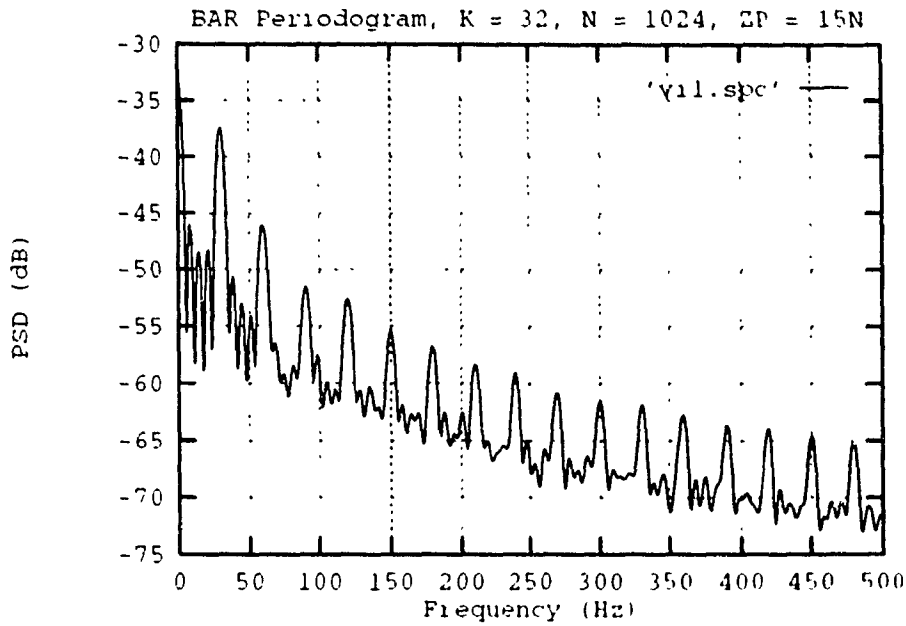


Figure 5.13: Periodogram of a single BAR video, K = 32, M = 1024.

should be equivalent. Furthermore, the periodograms are biased because that the sampling rate does not have a common factor with the factors of 30 Hz.

The second set of plots (Figures 5.14 to 5.17) show spectrum of a single source obtained by smoothing K periodograms of 1024 samples, padded with 15×1024 zeros. It is interesting to observe the increase of the 90 Hz component as the number of smoothed periodograms goes from 2 to 16.

Finally, Figure 5.18 is the periodogram of 10 BAR video sources. Table 5.1 provides some numerical values of the 30 Hz harmonics for 1, 10 and 20 sources. It can be concluded from this table that the way to estimate the number of sources is not straightforward. The index N in the case of 10 and 20 sources, shows the ratio between the power of the aggregate sources (10 or 20) to the single source, for the given frequency. A scan through the values of N let us conclude that the spectrum do not add linearly.

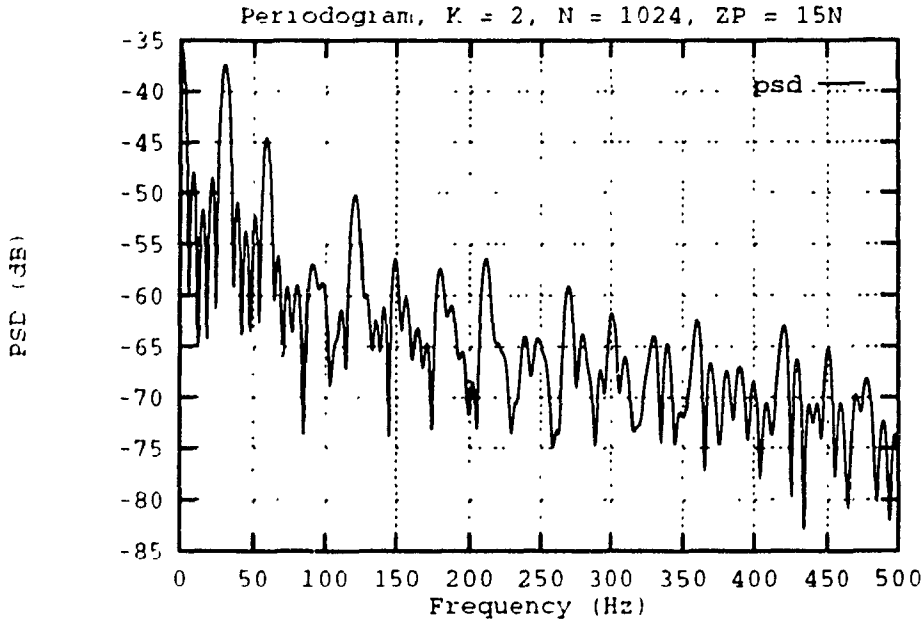


Figure 5.14: Periodogram of a single BAR video, K = 2, M = 1024.

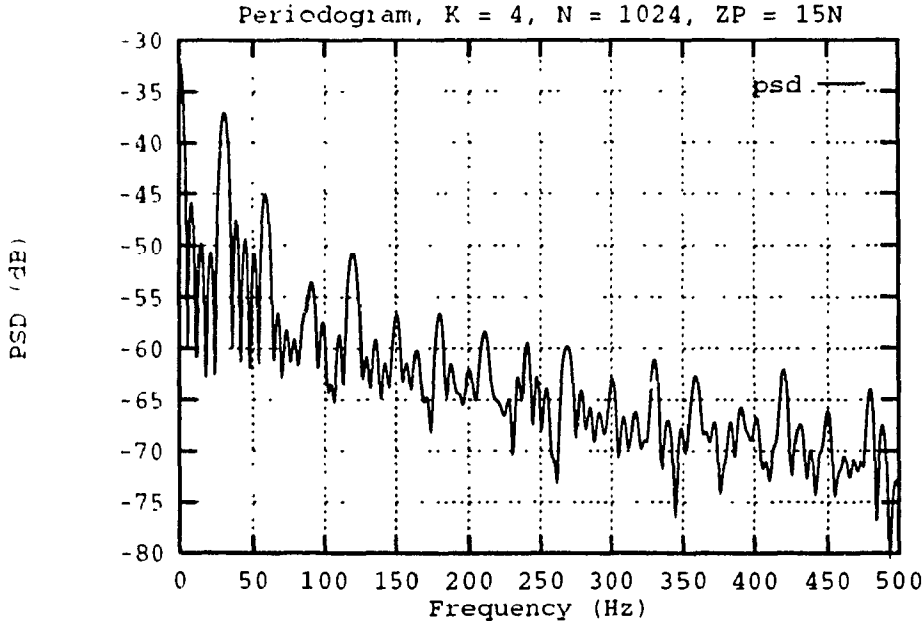


Figure 5.15: Periodogram of a single BAR video, K = 4, M = 1024.

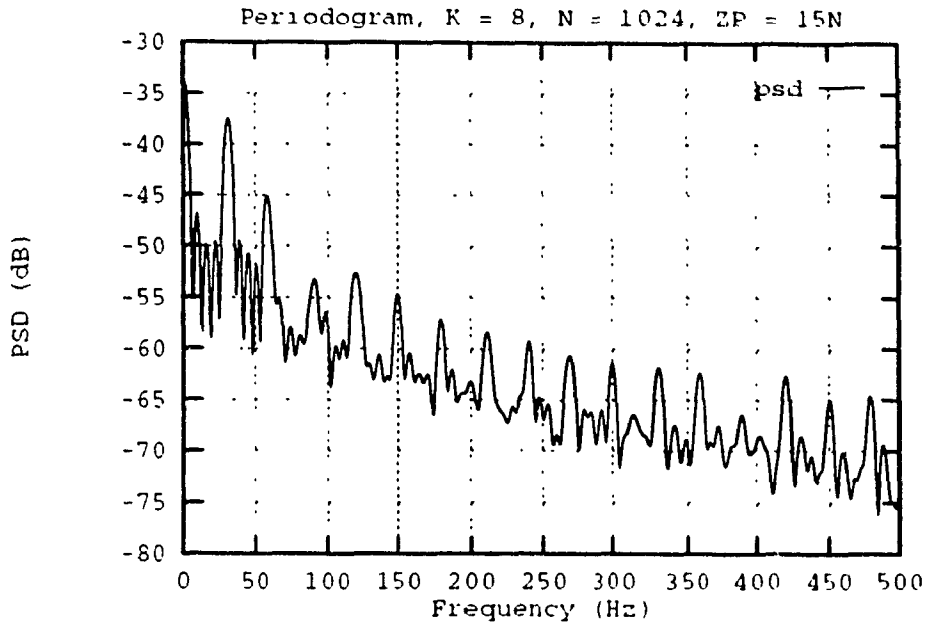


Figure 5.16: Periodogram of a single BAR video, K = 8, M = 1024.

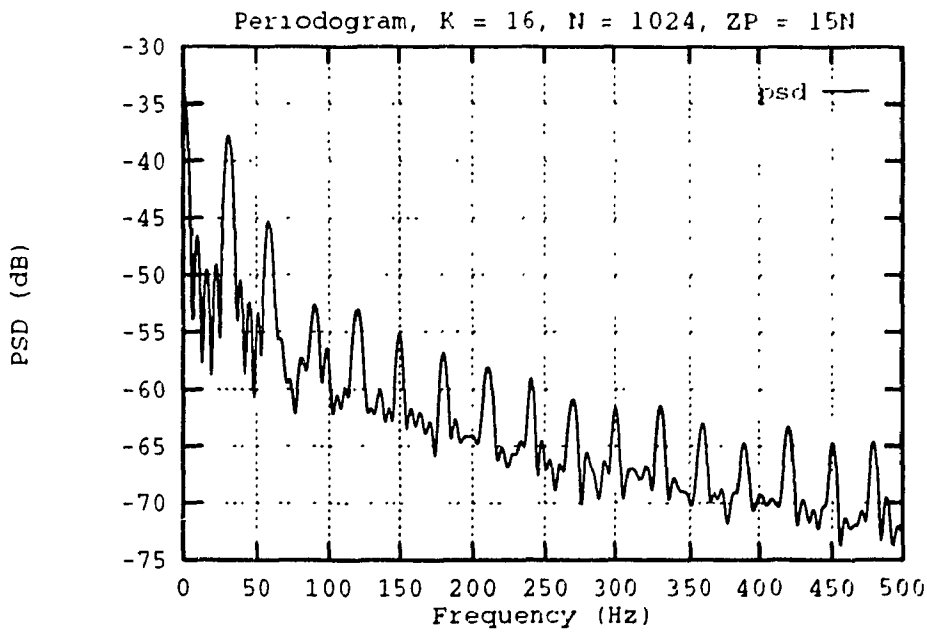


Figure 5.17: Periodogram of a single BAR video, K = 16, M = 1024.

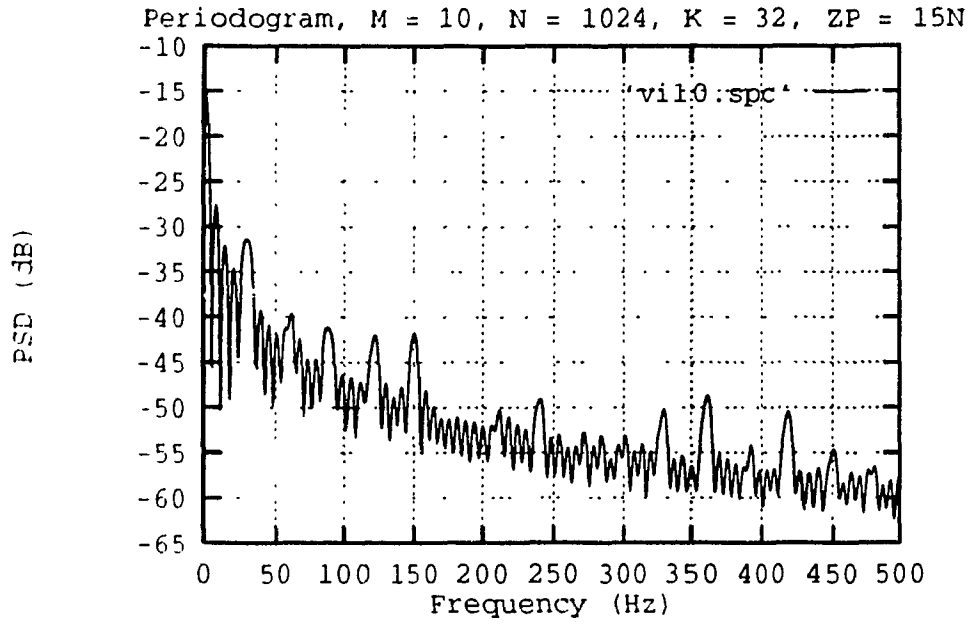


Figure 5.18: Periodogram of a 10 BAR video sources, K = 32, M = 1024.

f		Periodo 1		Periodo - 10			Periodo - 20		
Hz	k	μW/Hz	dB	μW/Hz	dB	N	μW/Hz	dB	N
30	80	179.8	-37.45	717.9	-31.44	4	1817.6	-27.41	10.1
60	161	24.03	-46.19	83.79	-40.77	3.5	242.6	-36.15	10.1
90	241	6.906	-51.61	73.61	-41.33	10.7	95.70	-40.19	13.9
120	322	5.457	-52.63	58.37	-42.34	10.7	143.2	-38.44	26.2
150	402	2.854	-55.45	64.82	-41.88	22.7	125.0	-39.03	43.8
180	482	2.070	-56.84	3.500	-54.56	1.7	15.37	-48.13	7.43
210	563	1.453	-58.38	7.069	-51.51	4.9	23.98	-46.20	16.5
240	643	1.231	-59.10	12.02	-49.20	9.8	24.77	-46.06	20.1
270	724	0.816	-60.88	3.507	-54.55	4.3	23.20	-46.35	28.4
300	804	0.687	-61.32	3.606	-54.43	5.2	31.77	-44.98	46.2

Table 5.1: Comparisons between periodograms for 1, 10 and 20 BAR video sources.

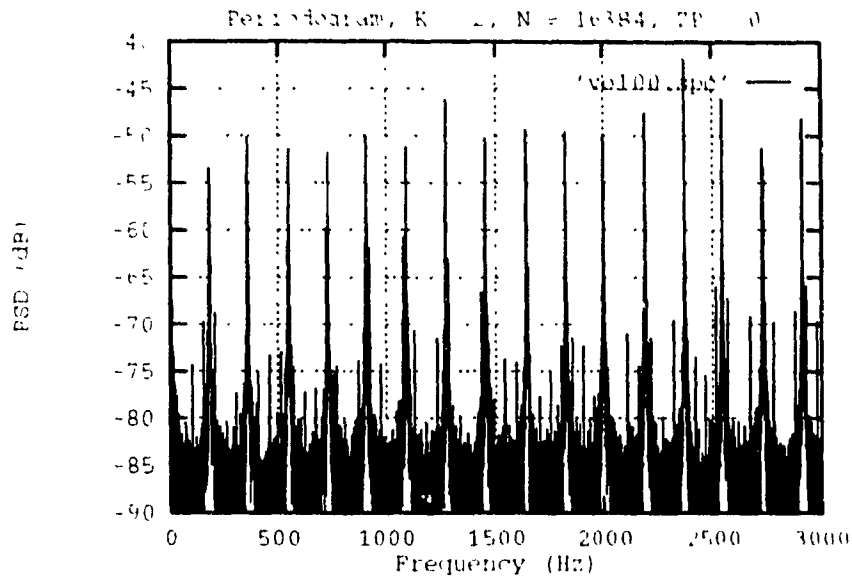


Figure 5.19: Periodogram of 100 voice sources, K = 2, M = 16384.

Voice Sources

This section is similar to the previous one; it gives the results for 100 multiplexed voice sources. There were 32, 8 and 2 periodograms of 1024, 4096 and 16384 data samples averaged to produce Figures 5.19 to 5.21. Table 5.2 provides some numerical values of the periodogram for the multiples of 181.84 Hz.

Mixed Traffic

Now, two mixed traffic streams are used. The former is composed of 1 video and 100 voice sources, the latter of 5 video and 200 voice sources. For the first case, a Bartlett's spectral estimator averaging 32 periodograms of 1024 data samples is plot in Figure 5.22. As it can be seen, both voice and video spectrum add such a way that it is evident that one can detect the presence of each class of sources. However, as it was seen previously, the estimation of the number of source of both type cannot be provided easily. Table 5.3 gives some numerical values for the first traffic stream and Table 5.4 for the one formed of 5 video and 200 voice sources.

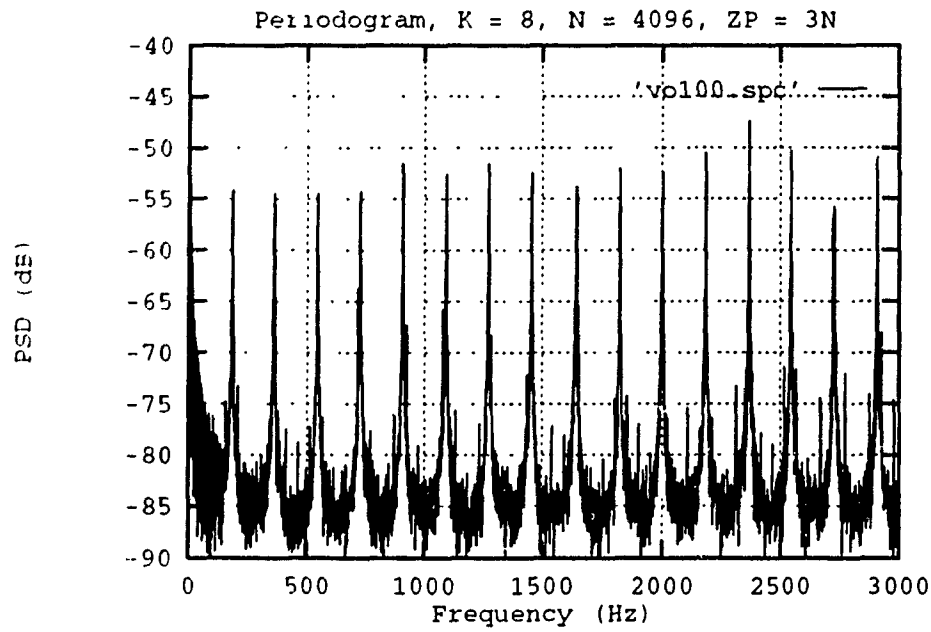


Figure 5.20: Periodogram of 100 voice sources, K = 8, M = 4096.

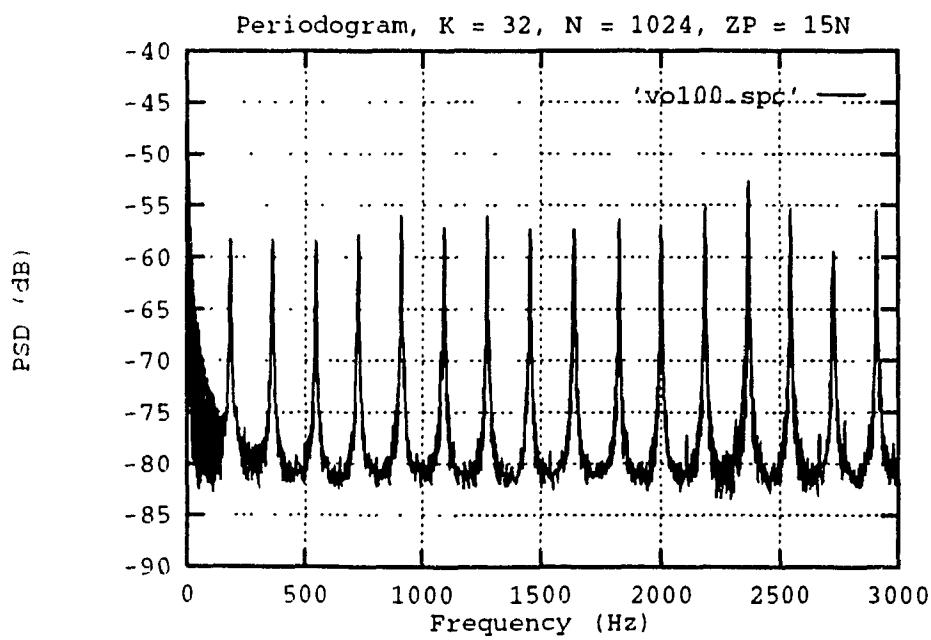


Figure 5.21: Periodogram of 100 voice sources, K = 32, M = 1024.

f		Periodogram		
Hz	k	μW/Hz	dB	N
181.70	487	1.452	−58.38	95.8
363.78	975	1.394	−58.56	92.0
545.48	1462	1.339	−58.73	88.3
727.19	1949	1.594	−57.98	105.2
909.26	2437	2.441	−56.12	161.0
1091.0	2924	1.890	−57.24	125.7
1272.7	3411	2.479	−56.06	163.5
1454.4	3898	1.826	−57.39	120.5
average		1.802		118.9

Table 5.2: Periodogram of 100 voice sources.

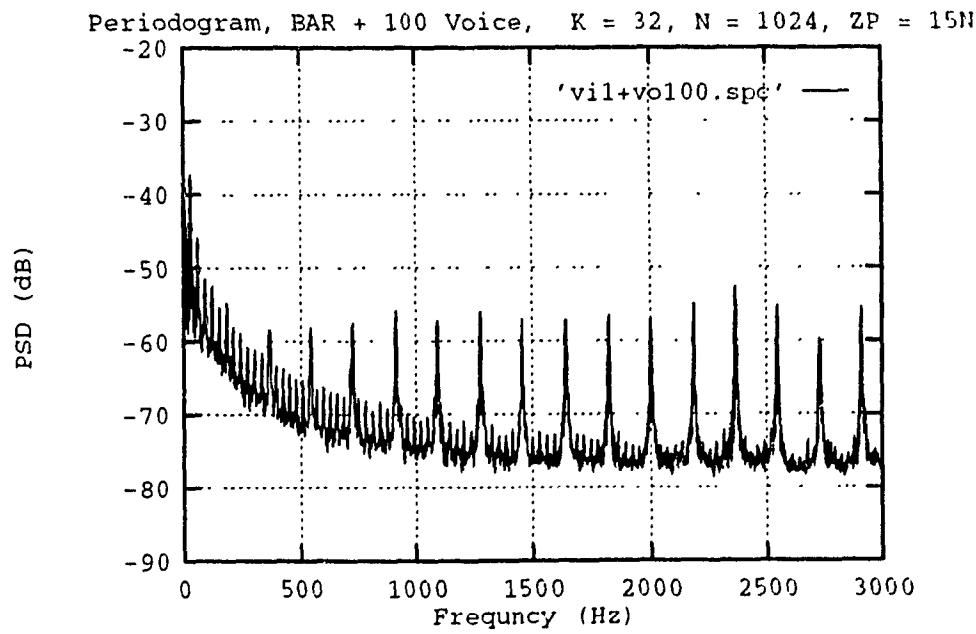


Figure 5.22: Periodogram of 1 video and 100 voice sources, K = 32, M = 1024.

Video				Voice			
f		PSD		f		PSD	
Hz	k	μW/Hz	dB	Hz	k	μW/Hz	dB
30	80	179.6	-37.46	181.70	487	3.092	-55.10
60	161	24.00	-46.20	363.78	975	1.398	58.54
90	241	6.927	-51.60	545.48	1462	1.432	58.44
120	322	5.513	-52.59	727.19	1949	1.596	-57.97
150	402	2.895	-55.38	909.26	2437	2.509	56.01
180	482	3.005	-55.22	1091.0	2924	1.836	-57.36
210	563	1.472	-58.32	1272.7	3411	2.476	-56.06
240	643	1.224	-59.12	1454.4	3898	1.926	57.16
270	724	0.828	-60.82				
300	804	0.699	-61.56				

Table 5.3: Periodogram of 1 video and 100 voice sources.

Video				Voice			
f		PSD		f		PSD	
Hz	k	μW/Hz	dB	Hz	k	μW/Hz	dB
30	80	2576.	-25.89	181.70	487	11.58	49.36
60	161	44.79	-43.49	363.78	975	4.066	53.91
90	241	36.96	-44.32	545.48	1462	4.594	53.38
120	322	41.46	-43.82	727.19	1949	3.157	55.01
150	402	22.82	-46.42	909.26	2437	2.644	55.78
180	482	10.02	-49.99	1091.0	2924	8.190	50.87
210	563	4.845	-53.15	1272.7	3411	3.963	54.02
240	643	4.299	-53.67	1454.4	3898	4.598	53.38
270	724	2.543	-55.95				
300	804	1.531	-58.15				

Table 5.4: Periodogram of 5 video and 200 voice sources.

Discussion

Due to its simplicity, the periodogram is very attractive to implement in hardware. A 1024 point FFT can be performed in less than 3 milliseconds on almost any DSP chip and 30 time faster—less than 100 μsec, with dedicated FFT chips.

As in the case of the correlogram, the periodogram can detect the presence of different classes of sources; however, the results of simulation show that quantitative estimates of the number of sources of each type that are off-hook are not evident from spectral measurements. The spectra of independent sources do not add in a straightforward manner. The phase correlation of different sources over short intervals give rise to nonergodicity with the result that spectra do not add in a simple fashion. By taking measurements at long intervals, the average number of sources of each class operating can be estimated. Although unreported, this was verified with Monte-Carlo simulations.

5.3 Filtering Approaches

For the filtering approach, a notch filter preceded by a downsampler is used. The downsampling is carried out exactly as in the periodogram approach. In the following example, a traffic stream composed of a single BAR video source and 100 voice sources is used. The constraint coefficient r is set equal to 0.999. This is not ideal, but works satisfactorily. The effect of a coefficient near one is to increase the transient duration. In fact, we are not dealing with a notch filter but rather with a selective filter where the output is equal to the input minus the output of the notch filter. Figure 5.23 shows the output of such a filter when selecting the 30 Hz component, and Figure 5.24 is the spectrum of this signal. In the following figures, the same is done for two characteristic tones of the voice model, the 545.53 and 1818.44 Hz components. It was expected for both to have the same envelope and then, just by monitoring these frequencies we would have obtained an estimate for the number of voice sources. To our great disappointment, it is not what has been obtained as it can be seen from Figures 5.25 and 5.27. It is our feeling that this difference is caused by the phase between the processes.

A comb filter may give good results for the voice sources by selecting all the components of 181.84 Hz. Then the output could be fed to a power detector and, the total power divided by the number of tones used to produce an estimate of the number of voice source. However, to use a comb filter for the voice, the sampling must be performed at a multiple of 181.84 Hz. Such sampling rate was very difficult to realize by simulation; for that reason, this method has not been tried.

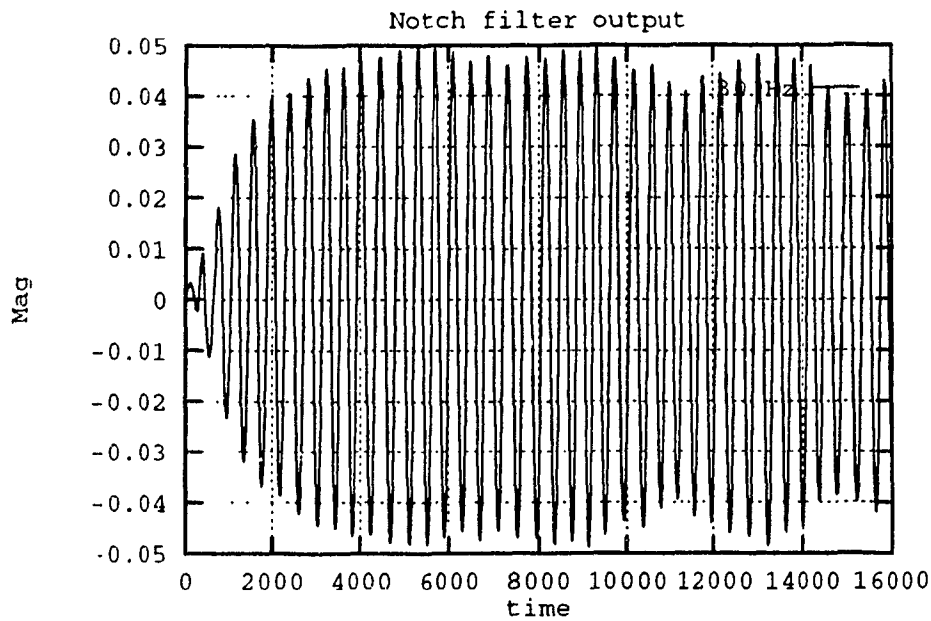


Figure 5.23: Output of the selective filter, 30 Hz component.

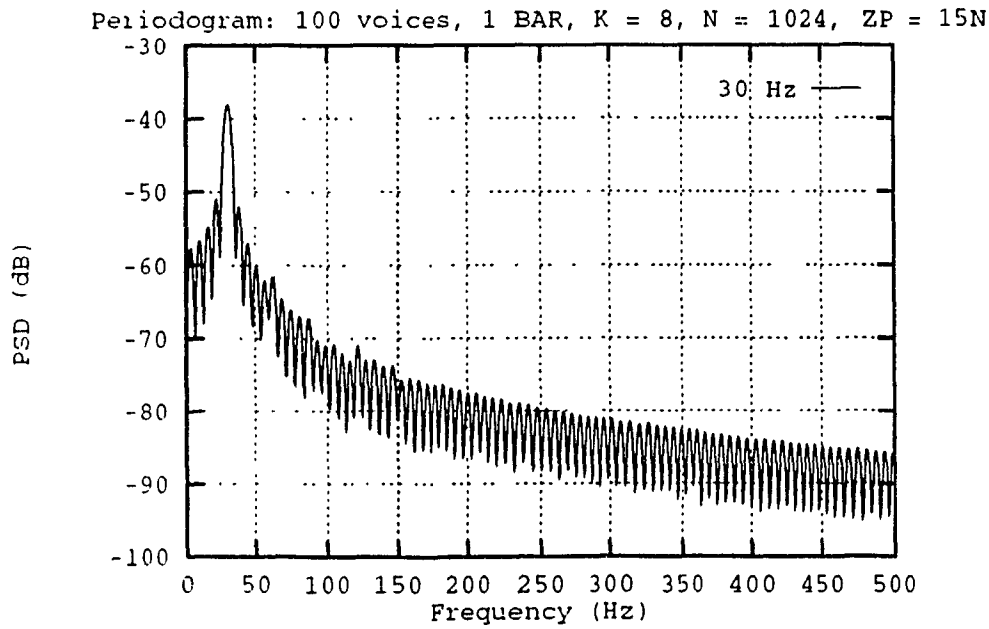


Figure 5.24: Power spectrum around the 30 Hz component.

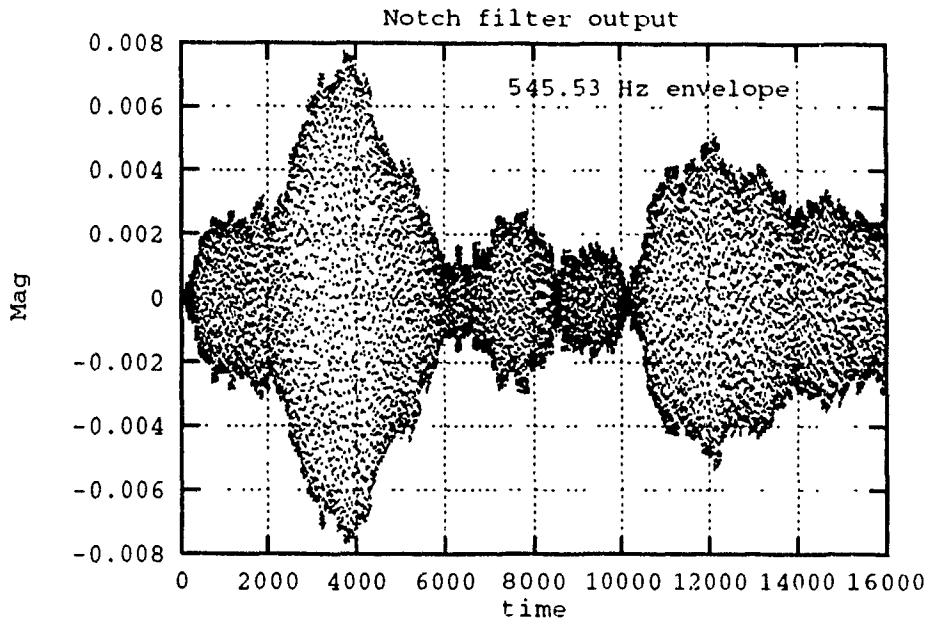


Figure 5.25: Output of the selective filter, envelope of the 545 Hz component.

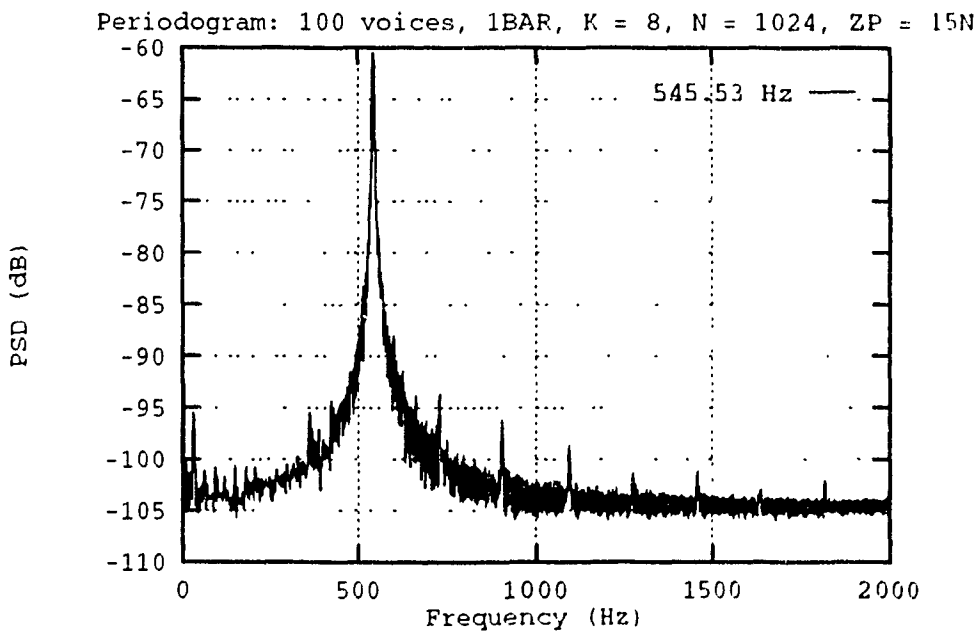


Figure 5.26: Spectrum around the 545 Hz component.

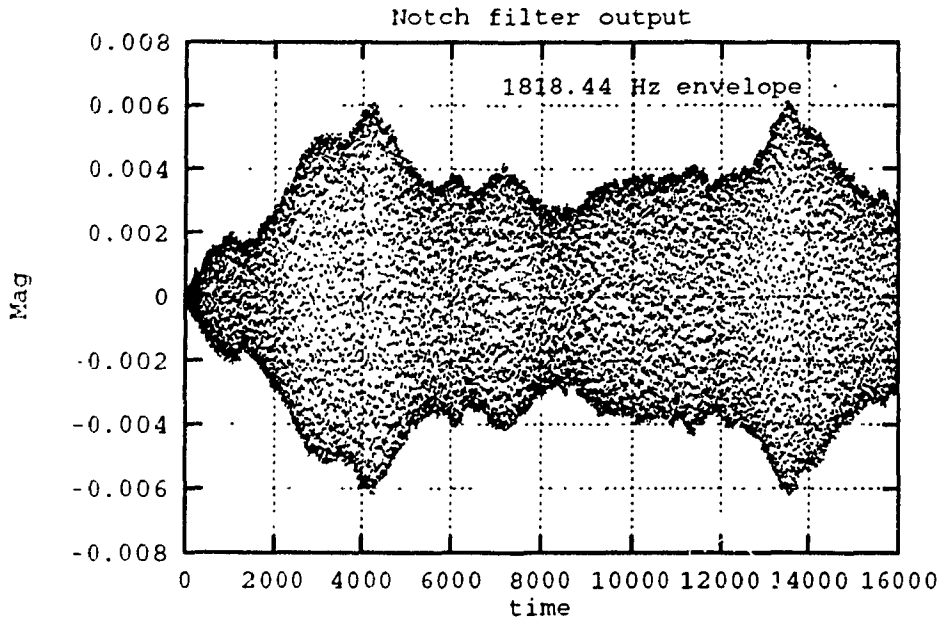


Figure 5.27: Output of the selective filter, envelope of the 1818 Hz component.

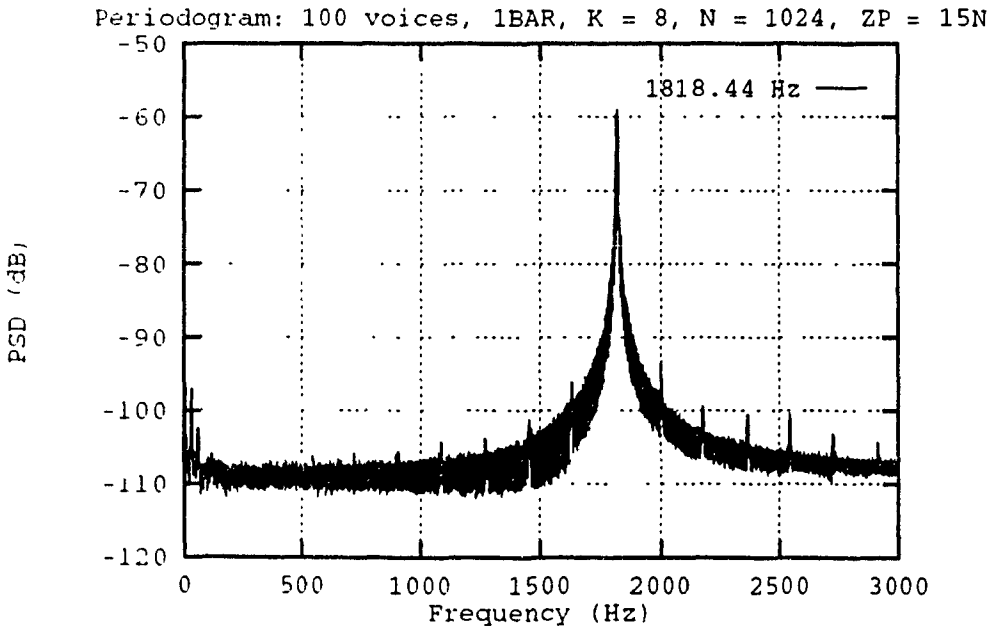


Figure 5.28: Spectrum around the 1818 Hz component.

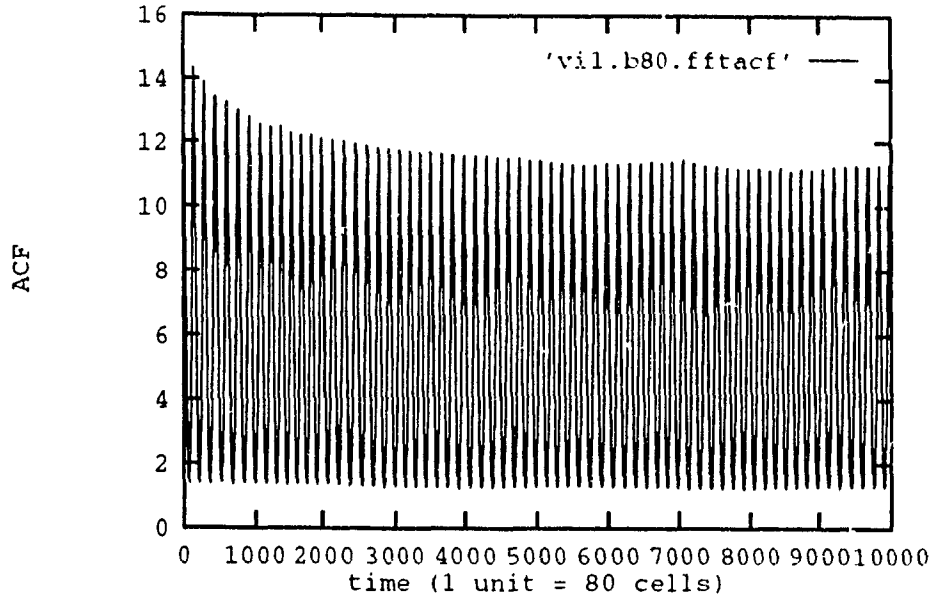


Figure 5.29: Autocorrelation function of 1 BAR video source with bin of 80 cells.

5.4 Counting Process and Modern Approach

This section presents some results obtained for the modern approach with the counting process. Concerning the counting process, a window of any size can be used. As a first cut, bins of 80 (218.1μsec) cells were used. Later, a bin size of 367 cells (1.0006 msec) was tried.

Figures 5.29 to 5.32 show the autocorrelation function obtained for the BAR video model, with 1 and 10 sources, using bins of 80 and 367 cells. The autocorrelation function obtained is very similar to the one of Figure 5.1. An exponential envelope with the AR coefficient (a = 0.88) is easily fitted for the plot of a single source. For 10 sources, the fitting still possible, but is more difficult due to the mean which is increasing with the square of the number of sources while the autocovariance increases linearly. Figure 5.33 is the autocorrelation obtained for a mixed traffic of 1 video and 100 voice sources. It is very similar to Figure 5.30 except that the notches do not go as deep.

This counting method has the advantage of reducing the processing to obtain an estimate of the autocorrelation function, but has the disadvantage to smooth the traffic stream such that the voice correlation disappear; it is hidden by the video correlation. Although the counting process reduces the capability of estimating the voice traffic, it can be used to estimate the video traffic. Using only the lattice points (1/30 second multiples) Prony's method—which tries to fit a weighted sum of exponential, could be used to discriminate between several types of video.

Finally, the bin size chosen influences the choice of the parametric model. For example, if there is a single class of voice source that sends a packet every 5.5 milliseconds, and we are using a bin size of 0.5 milliseconds, the sources can be modeled as an AR(11). This has been studied experimentally and seems to work. However, when multiplexing voice and video, the video acts as noise for the voice, and the parameter estimation is difficult to perform, if not impossible. The problem is that the video load is much higher than the voice load, thus rendering the voice load neglectable.

5.5 Discussion

In the course of this work, a simulator was written to compare the theoretical results, computed numerically, with the estimated ones, obtained by simulation. The simulator was also written to evaluate how well the methods described in Chapter 4 did with respect to accuracy in estimating the traffic composition.

In this chapter, only the results that may be used as a basis for the design of a system that would estimate the traffic composition were presented. There was no, strictly speaking, performance analysis on the estimators themselves, but more a general overview of the methods that may be implemented, with a basic evaluation of their performance by means of their accuracy in estimating the traffic composition using only the magnitude of some characteristic tones.

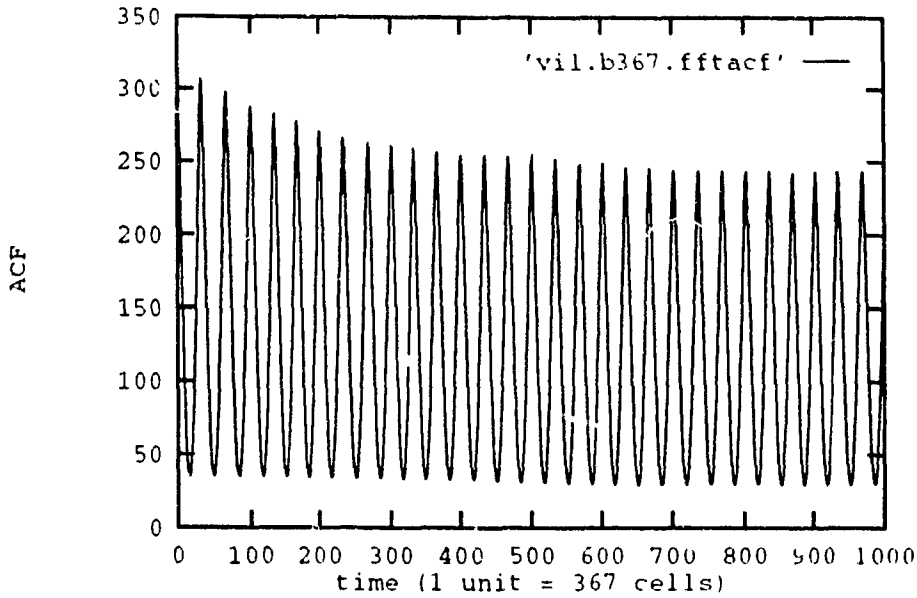


Figure 5.30: Autocorrelation function of 1 BAR video source with bin of 367 cells.

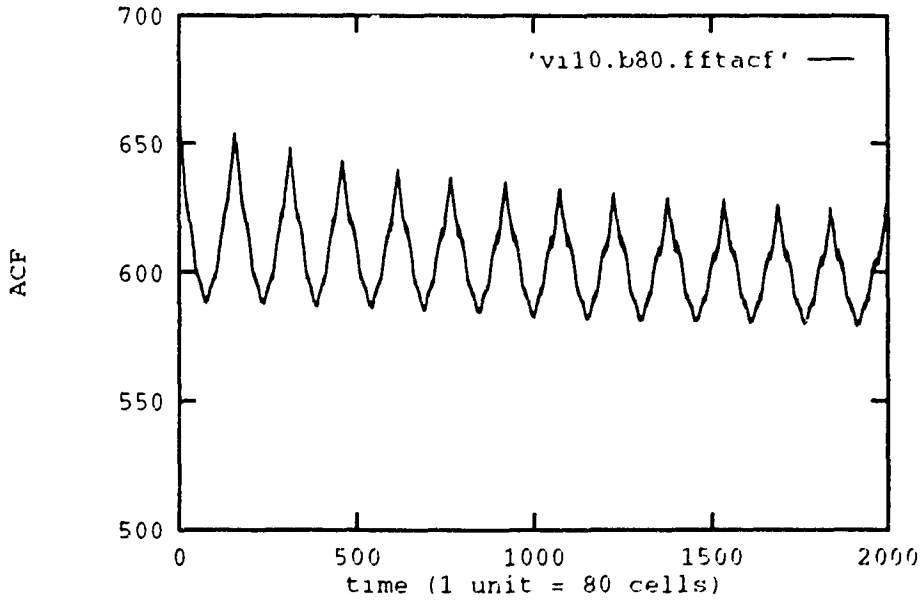


Figure 5.31: Autocorrelation function of 10 BAR video sources with bin of 80 cells.

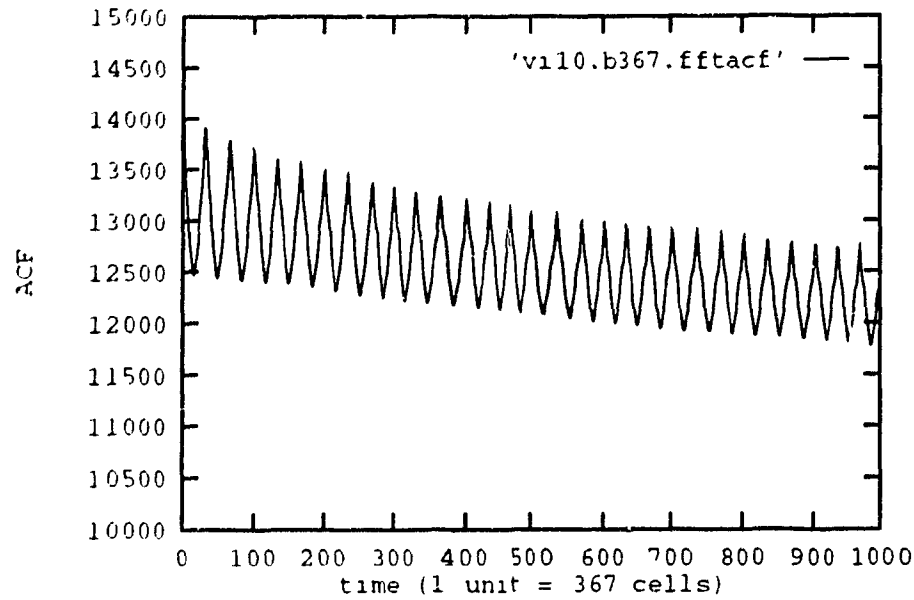


Figure 5.32: Autocorrelation function of 10 BAR video sources with bin of 367 cells.

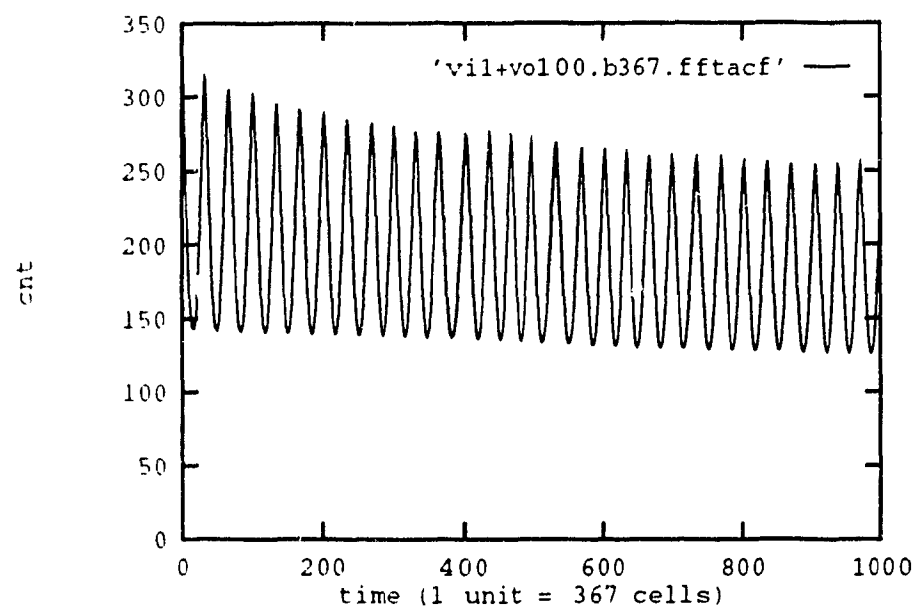


Figure 5.33: Autocorrelation function of 1 video and 100 voice sources with bin of 367 cells.

The section on the autocorrelation function estimation and the correlogram verified the accuracy of the simulator with respect to the numerically computed theoretical power spectral density and autocorrelation function. It was shown that the simulator produces traffic stream similar to the expected ones but not perfectly since there was a constant 10% error in the magnitude of the autocorrelation function.

The section on the periodogram approach showed that real-time traffic estimation may be performed using 1024 point FFT processors. While different classes of sources may be discriminated, more work is required in order that accurate estimates of the number of sources of each type can be obtained. The same conclusions hold for the filtering approaches.

The modern techniques were not as successful as desired. First it is difficult to get estimates in real-time, and the models are not as simple as AR(p). Due to the nature of the process, modern spectral estimation has no meaning at the cell level. To overcome this fact, windowing was performed. The windowing procedure works as in the leaky bucket: it counts the packets for a specified time interval, and then tries to fit a pre-selected model. The counting processes observed were not first-order autoregressive as expected from [11]. A study on a good window size may be carried out to get simple AR(p) models.

The results reported in this chapter are not all that we have obtained. A lot more traffic simulations, spectrum and autocorrelation function computations have been made and support the general conclusion. Spectral estimation may be performed to estimate the nature of a random traffic stream, but further work has to be done in order to characterize the different type of sources that an ATM network is expected to handle.

The next chapter concludes this thesis.

Chapter 6

Conclusion

This thesis concerns first step of a new approach to congestion control for ATM networks which uses spectral analysis. One of the main advantages of this new approach, is that it can be applied anywhere in the network. All that it needs is the ATM cell stream, from which a binary process is formed according to the state of the cell, full or empty. The thesis investigates different techniques to estimate the traffic composition by means of spectral analysis.

After the introduction, Chapter 2 presented some ATM network features such as the variable bit rate source that it is expected to handle, the ATM cell format and the statistical multiplexing.

In Chapter 3 several models for packetized voice, video and data were developed. The voice model is a Markovian binary process which takes its origin in the characterization of telephone speech. For video, two extremes cases have been studied. The first model (BAR) sends all its information, as a burst, at the beginning of every video frame. The second model (UAR) sends all the information of a video frame, uniformly, over an entire video frame interval. Data models were Poisson and Bernoulli processes. It is clear that the voice, video and data models presented are only five of the many kinds of sources that the ATM network may have to support. An area of further study may be to adapt the previous models—and, or imagine new ones, to characterize new types of sources that will have to be supported.

The fourth chapter was on the techniques that can be applied to estimate the traffic composition. Classical estimators—the periodogram and correlogram, were briefly described. Some modern techniques were also presented. At the end, a filtering strategy involving notch filters was described.

Chapter 5 presented the results on the methods that were tried by computer simulation. Emphasis was made on the easiest to implement in hardware. Such methods are the periodogram and the notch filter, both preceded by a downsampler. In terms of estimating the traffic load of a network, these approaches are relatively new. There was limited success in observing the spectra of different classes of

sources; however, further work both theoretical and practical is required in order to obtain accurate estimates of the number of sources of each class that are off-hook.

As expected, each class of traffic have a different signature spectrum characterized by basic harmonic frequencies and envelope. However, due to the short observation periods and the phase correlation between the processes, it was shown that the number of sources of each type cannot be quantified easily from spectral measurements.

Finally, a new approach to congestion control involving spectral analysis have been presented. The ultimate goal was not to evaluate this or that estimation approach with a Monte-Carlo simulation, but to get a good idea of which method could be used as a the basis to a system that would estimate the traffic composition in ATM networks. The performance analysis of the presented methods and the design of the decision rules may be a topic of further research.

Bibliography

[1] CCITT VIIIth Plenary Assembly, "Recommendation I.121 for B-ISDN," 1981.

[2] S. Wolf, C. A. Dvorak, R. F. Kubichek, C. R. South, R. A. Schphorst, and S. D. Voran, "How will we rate telecommunications system performance?," *IEEE Communications Mag.*, vol. 29, pp. 23-29, Oct. 1991.

[3] J. F. Hayes, "An approach to congestion control in ATM networks," tech. rep., Concordia University, Montréal, Québec, Canada, Sept. 1990.

[4] IEEE Journal on Selected Areas in Communications, "Teletraffic Analysis of ATM Systems." Volume 9, Number 3, Apr. 1991.

[5] IEEE Journal on Selected Areas in Communications, "Congestion Control in High Speed Packet Switched Networks." Volume 9, Number 7, Sept. 1991.

[6] IEEE Communications Magazine, "B-ISDN: High Performance Transport." Volume 29, Number 9, Sept. 1991.

[7] IEEE Communications Magazine, "Special issue: Congestion Control in High Speed Networks." Volume 29, Number 10, Oct. 1991.

[8] R. M. Rodriguez-Dagnino, M. R. K. Khansari, and A. Leon-Garcia, "Prediction of bit rate sequences of encoded video signals," *IEEE J. Select. Areas Commun.*, vol. SAC-9, pp. 305-314, Apr. 1991.

[9] CCITT Study Group XVIII, "Recommendation I.361 for B-ISDN," Nov. 1990.

[10] B. Voeten, "private communication," May 1987.

[11] B. Maglaris, D. Anastassiou, P. Sen, G. Karlsson, and J. D. Robbins, "Performance models of statistical multiplexing in packet voice communications," *IEEE Trans. Commun.*, vol. COM-36, pp. 834–844, July 1988.

[12] M. Nomura, T. Fujii, and N. Ohta, "Basic characteristics of variable rate video coding in ATM environment," *IEEE J. Select. Areas Commun.*, vol. SAC-7, pp. 752–760, June 1989.

[13] W. Verbiest, L. Pinnoo, and B. Voeten, "The impact of the ATM concept on video coding," *IEEE J. Select. Areas Commun.*, vol. SAC-6, pp. 1623–1632, Dec. 1988.

[14] O. Gihr and P. Tran-Gia, "A layered description of ATM cell traffic streams and correlation analysis," *Proc. IEEE Infocomm.*, vol. 1, pp. 2D.4.1–2D.4.8, April 7–11 1991.

[15] E. P. Rathgeb, "Modeling and performance comparison of policing mechanisms for ATM networks," *IEEE J. Select. Areas Commun.*, vol. SAC-9, pp. 325–334, Apr. 1991.

[16] R. Gruenfelder, J. P. Cosmas, S. Manthorpe, and A. Odinam-Okafor, "Characterization of video codecs as autoregressive moving average processes and related queueing system performance," *IEEE J. Select. Areas Commun.*, vol. SAC-9, pp. 284–293, Apr. 1991.

[17] P. T. Brady, "A model for generating on-off speech patterns in two-way conversation," *Bell Syst. Tech. J.*, vol. 48, pp. 2445–2472, 1969.

[18] Y. Yatsuzuka, "Highly sensitive speech detector and high-speed voiceband data discriminator in DSI-ADPCM systems," *IEEE Trans. Commun.*, vol. COM-30, pp. 739–750, Apr. 1982.

[19] K. Sriram and W. Whitt, "Characterizing superposition arrival processes in packet multiplexers for voice and data," *IEEE J. Select. Areas Commun.*, vol. SAC-4, pp. 833–846, Sept. 1986.

[20] A. Papoulis, *Probability, Random Variables, and Stochastic Processes.* New York: McGraw-Hill, second ed., 1984.

[21] H. Yamada and S. Sumita, "A traffic measurment method and its application for cell loss probability estimation in ATM networks," *IEEE J. Select. Areas Commun.*, vol. SAC-9, pp. 315–324, Apr. 1991.

[22] G. E. P. Box and G. M. Jenkins, *Time Series Analysis: Forecasting and Control.* San Francisco: Holden-Day, 1970.

[23] M. B. Priestley, *Spectral Analysis and Time Series*, vol. 1. New York: Academic Press, first ed., 1981.

[24] J. M. Wozencraft and I. M. Jacobs, *Principles of Communication Engineering.* New York: John Wiley & Sons, 1965.

[25] G. M. Jenkins and D. G. Watts, *Spectral Analysis and its Applications.* San Francisco: Holden-Day, 1968.

[26] S. M. Kay, *Modern Spectral Estimation: Theory and Application.* Englewood Cliffs, NJ: Prentice-Hall, 1988.

[27] S. L. Marple Jr., *Modern Spectral Analysis with Applications.* Englewood Cliffs, NJ: Prentice-Hall, 1987.

[28] K. S. Shanmugan and A. Breipohl, *Random Signals: Detection, Estimation and Data Analysis.* New York: John Wiley & Sons, 1988.

[29] R. B. Blackman and J. W. Tukey, *The Measurment of Power Spectra from the Point of View of Communication Engineering.* New York: Dover, 1958.

[30] S. M. Kay and S. L. Marple Jr., "Spectrum analysis—a modern perspective," *Proc. IEEE*, vol. 69, pp. 1380-1419, Nov. 1981.

[31] F. J. Harris, "On the use of windows for harmonic analysis with the discrete Fourier transform," *Proc. IEEE*, vol. 66, pp. 51–83, Jan. 1978.

[32] A. V. Oppenheim and R. W. Schafer, *Discrete-Time Signal Processing.* Englewood Cliffs, NJ: Prentice-Hall, first ed., 1989.

[33] D. R. Cox and H. D. Miller, *The Theory of Stochastic Processes.* Methuen, 1965.

[34] W. H. Press, B. P. Flannery, S. A. Teukolsky, and W. T. Vetterling, *Numerical Recipes in C.* Cambridge: Cambridge University Press, 1988.

[35] C. D. McGillem and G. R. Cooper, *Continuous and Discrete Signal and System Analysis.* New York: Holr, Rinehart and Winston, second ed., 1984.

Appendix A

Z-Transform Approach for the Computation of the Autocorrelation Function and the Power Spectral Density of a Binary Markov Process

There exist several ways to compute the burst and cell autocorrelation functions $r(k)$, $R(k)$ and the power spectral density $P(f)$ of a binary Markov process. One of these is the transient behavior matrix approach as used in the Section 3.1.1, another is the z-transform technique (or generating function) that follows.

A.1 z-transform Technique

Let recall the random point process defined in the Section 3.1.1 for the voice source at the burst level.

$$X(i) = \begin{cases} 0 & \text{if the source is idle} \\ 1 & \text{if the source is active} \end{cases} \quad \text{for } i = 1, 2, 3 \ldots \tag{A.1}$$

The autocorrelation function $r(k)$ is given by

$$\begin{aligned} r(k) &= E[X(i) \cdot X(i+k)], \\ &= \sum_{j=0}^{1} \sum_{l=0}^{1} j \cdot l \cdot \Pr\{X(i) = j, X(i+k) = l\}, \\ &= \Pr\{X(i) = 1, X(i+k) = 1\}, \\ &= \Pr\{X(i+k) = 1 \mid X(i) = 1\} \cdot \Pr\{X(i) = 1\}, \\ &= p_{11}^{k} \cdot p_1. \end{aligned} \tag{A.2}$$

where p_{11}^{k} is the conditional probability that $X(i)$ is in active state given that it was in active state k steps previously and p_1 is the steady-state probability to be in active state. Let define the probability generating function

$$P(z) = \sum_{k=0}^{\infty} z^k \cdot p_{11}^{k}. \tag{A.3}$$

The equation for $P(z)$ may be written in term of $F(z)$ the probability generating function of first passage time from active state to itself [33]

$$P(z) = \frac{F(z)}{1 - F(z)}. \tag{A.4}$$

Let define the first time passage transition probability f_{11}^k as

$$f_{11}^k = \Pr\{X(i+k) = 1 \text{ for the first time after } X(i) = 1\} \tag{A.5}$$

such that

$$f_{11}^k = \begin{cases} 0 & \text{for } k = 0 \\ 1-q & \text{for } k = 1 \\ pq(1-p)^{k-2} & \text{for } k = 2, 3, 4, \ldots \end{cases} \tag{A.6}$$

The first time passage probability generating function becomes

$$F(z) = \sum_0^\infty z^k \cdot f_{11}^k, \tag{A.7}$$

$$= \frac{z(1-q) + z^2(1-p-q)}{1 - z(1-p)}. \tag{A.8}$$

Substituting Equation (A.8) into Equation (A.4) we obtain

$$P(z) = \frac{z(1-q) - z^2(1-p-q)}{1 - z(2-p-q) + z^2(1-p-q)},$$

$$= \frac{p}{p+q} \cdot \frac{1}{1-z} + \frac{q}{p+q} \cdot \frac{1}{1 - z(1-p-q)}. \tag{A.9}$$

Taking the inverse z-transform

$$p_{11}^k = \left(\frac{p}{p+q}\right) \cdot \left[1 + \frac{q}{p} \cdot (1-p-q)^k\right], \text{ for } k = 0, 1, 2, \ldots \tag{A.10}$$

Substituting Equation (A.10) in Equation (A.2), and taking the limit for p_{11}^k, we obtain the steady-state probability p_1.

$$p_1 = \lim_{k \to \infty} p_{11}^k = \frac{p}{p+q}. \tag{A.11}$$

Using the symmetry property of autocorrelation function of real process, the burst autocorrelation function becomes

$$r(k) = \left(\frac{p}{p+q}\right)^2 \cdot \left[1 + \frac{q}{p}(1-p-q)^{|k|}\right], \text{ for } k = 0, \pm 1, \pm 2, \ldots \tag{A.12}$$

With the transient behavior matrix approach used in Section 3.1.1, one finds the same results.

A.2 Power Spectral Density

The power spectral density of a point process is by definition [20]

$$\tilde{P}(\omega) = \sum_{m=-\infty}^{\infty} R(m) \cdot \exp(-j\omega \cdot m \cdot T), \tag{A.13}$$

where T is an arbitrary constant that depends on the sampling frequency. In our case, T is the cell duration $\mathcal{T}$. Furthermore, instead of using the pulsation ω, we write the Fourier transform $\tilde{P}(\omega)$ in term of the frequency, f. Hence

$$P(f) = \sum_{m=-\infty}^{\infty} R(m) \cdot \exp(-j2\pi \cdot m \cdot \mathcal{T} \cdot f). \tag{A.14}$$

Substituting the autocorrelation coefficient and taking into account the cell interarrival I the power spectral density can be developed as follows:

$$\begin{aligned}
P(f) &= \sum_{m=-\infty}^{\infty} R(m) \cdot \exp(-j2\pi \cdot m \cdot \mathcal{T} \cdot f), \\
&= \sum_{n=-\infty}^{\infty} \frac{r(nI)}{I} \cdot \exp(-j2\pi \cdot nI \cdot \mathcal{T} \cdot f), \\
&= \frac{1}{I} \cdot \left(\frac{p}{p+q}\right)^2 \cdot \sum_{n=-\infty}^{\infty} \left[1 + \frac{q}{p}(1-p-q)^{|n|I}\right] \cdot \exp(-j2\pi \cdot n \cdot I\mathcal{T} \cdot f), \\
&= \left(\frac{p}{p+q}\right)^2 \cdot \frac{1}{I^2\mathcal{T}} \cdot \sum_{n=-\infty}^{\infty} \delta(f - \frac{n}{I\mathcal{T}}) + \\
&\quad \left(\frac{p}{p+q}\right)^2 \cdot \left\{\frac{q}{Ip} \sum_{n=-\infty}^{\infty} (1-p-q)^{|n|I} \cdot \exp(-j2\pi \cdot n \cdot I\mathcal{T} \cdot f)\right\}, \\
&= \left(\frac{p}{p+q}\right)^2 \cdot \frac{1}{I^2\mathcal{T}} \cdot \sum_{n=-\infty}^{\infty} \delta(f - \frac{n}{I\mathcal{T}}) + \\
&\quad \left(\frac{p}{p+q}\right)^2 \cdot \left\{\frac{q}{Ip} \cdot \frac{1-\gamma^2}{1+\gamma^2 - 2\gamma\cos(2\pi I\mathcal{T}f)}\right\},
\end{aligned} \tag{A.15}$$

where $\gamma = (1-p-q)^I$.

Now we can examine the superposition of several independent and uncorrelated sources. Let define the aggregate process of M sources as X_M.

$$X_M(i) = \sum_{j=1}^{M} X_j(i). \tag{A.16}$$

where $X_j(i)$'s are uncorrelated, i.i.d. voice source cell processes. The cell autocorrelation function is given by

$$\begin{aligned} R_M(k) &= E\left[X_M(i) \cdot X_M(i+k)\right], \\ &= E\left[\sum_{j=1}^{M} X_j(i) \cdot \sum_{l=1}^{M} X_l(i+k)\right], \\ &= \sum_{j=1}^{M}\sum_{l=1}^{M} E\left[X_j(i) \cdot X_l(i+k)\right]. \end{aligned} \tag{A.17}$$

$$\text{If } j = l \qquad E\left[X_j(i) \cdot X_l(i+k)\right] = R(k). \tag{A.18}$$

$$\text{If } j \neq l \qquad E\left[X_j(i) \cdot X_l(i+k)\right] = \left(E\left[X_j(i)\right]\right)^2. \tag{A.19}$$

Therefore,

$$R_M(k) = M \cdot R(k) + M(M-1) \cdot \left(E[X_j(i)]\right)^2. \tag{A.20}$$

Taking the Fourier transform of Equation (A.20) we obtain the power spectral density of the superposition of M voice sources.

$$P_M(f) = M \cdot P(f) + M(M-1) \cdot \frac{\left(E[X_j(i)]\right)^2}{T} \sum_{n=-\infty}^{\infty} \delta(f - \frac{n}{T}). \tag{A.21}$$

Appendix B

Conditional Cross-Correlation Function Analysis

This appendix contains an analysis of the conditional cross-correlation function $r(\tau \mid i, j, k)$ of the Section 3.2.3. Let recall that $r(\tau \mid i, j, k)$ denote the conditional cross-correlation function between bursts of length i at time t and length j at time $t + kT_f + \tau$. The Figure B.1 gives an illustration of the problem.

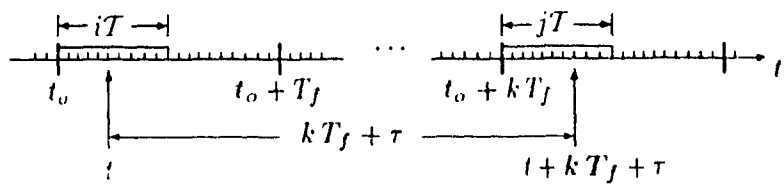


Figure B.1: Illustration of the problem.

Let t denote a uniform random variable in the close interval $[t_o, t_o + T_f]$. Since the process can only take the value zero or one, the conditional cross-correlation function can be written as

$$
\begin{aligned}
r(\tau \mid i, j, k) &= E\left[X(t) \cdot X(t + kT_f + \tau) \mid i, j, k\right], \qquad \text{for } |\tau| \leq T_f, && \text{(B.1)} \\
&= \Pr\left\{X(t) = 1, X(t + kT_f + \tau) = 1 \mid i, j, k\right\}, && \text{(B.2)} \\
&= \Pr\left\{X(t + kT_f + \tau) = 1 \mid X(t) = 1, i, j, k\right\} \\
&\qquad \times \Pr\left\{X(t) = 1 \mid i, j, k\right\}. && \text{(B.3)}
\end{aligned}
$$

Assuming that t is uniformly distributed in the interval $I_{t_o} = [t_o, t_o + T_f]$, then

$$
\begin{aligned}
\Pr\left\{X(t) = 1 \mid i, j, k\right\} &= \Pr\left\{t \in I_{t_o} \mid i\right\}, && \text{(B.4)} \\
&= \frac{iT}{T_f}. && \text{(B.5)}
\end{aligned}
$$

The conditional probability $\Pr\left\{X(t + kT_f + \tau) = 1 \mid X(t) = 1, i, j, k\right\}$ must be studied in three parts: i equal to, less than and greater than j.

B.1 Evaluation of $r(\tau \mid i, j, k \text{ and } i = j)$

First let define the interval $I_{t_o}^{(k)} = [t_o + kT_f, t_o + (k+1)T_f]$, and rewrite the conditional probability as

$$\begin{aligned} p(\tau \mid i = j) &= \Pr\{X(t + kT_f + \tau) = 1 \mid X(t) = 1, i, j, k, i = j\}, & \text{(B.6)} \\ &= \Pr\{t + kT_f + \tau \in I_{t_o}^{(k)} \mid t \in I_{t_o}, i, j, k, i = j\}. & \text{(B.7)} \end{aligned}$$

The only random variable in Equation (B.7) is t. The variable τ is deterministic and is bounded such that $|\tau| \leq T_f$. Equation (B.7) can be rewritten as

$$\begin{aligned} p(\tau \mid i = j) &= \Pr\{t + \tau \in I_{t_o} \mid t \in I_{t_o}, i, j, k, i = j\}, & \text{(B.8)} \\ &= \Pr\{-\tau < t < \tau \mid 0 < t < i\mathcal{T}\}, & \text{(B.9)} \\ &= 1 - \frac{|\tau|}{i\mathcal{T}}, \qquad \text{for } |\tau| \leq i\mathcal{T}. & \text{(B.10)} \end{aligned}$$

Therefore, the conditional cross-correlation function is obtained and

$$\begin{aligned} r(\tau \mid i, j, k \text{ and } i = j) &= p(\tau \mid i = j) \cdot \frac{i\mathcal{T}}{T_f}, & \text{(B.11)} \\ &= \left(1 - \frac{|\tau|}{i\mathcal{T}}\right) \cdot \frac{i\mathcal{T}}{T_f}, & \text{(B.12)} \\ &= \frac{i\mathcal{T} - |\tau|}{T_f}, \qquad \text{for } |\tau| \leq i\mathcal{T}. & \text{(B.13)} \end{aligned}$$

This is a triangle centered at zero, as in Figure B.2, with height $\frac{i\mathcal{T}}{T_f}$ and width $2i\mathcal{T}$.

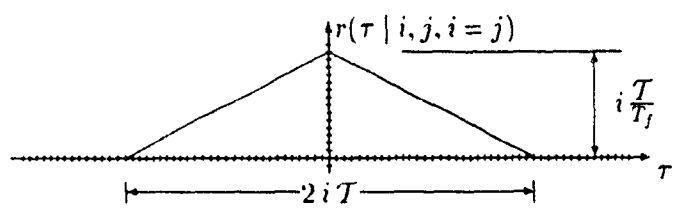


Figure B.2: Conditional cross-correlation function $r(\tau \mid i, j, k, i = j)$.

B.2 Evaluation of $r(\tau \mid i, j, k \text{ and } i < j)$

Rewriting Equation (B.8) with i less than j instead of equal to j, one obtains for τ greater than zero

$$\begin{aligned} p(\tau \mid i < j, \tau > 0) &= \Pr\{t + \tau \in I_{t_o} \mid t \in I_{t_o}, i, j, k, i < j, \tau > 0\}, && \text{(B.14)} \\ &= \Pr\{t < jT - \tau \mid 0 < t < iT, i < j\}, && \text{(B.15)} \\ &= \begin{cases} 1 & \text{for } 0 < \tau \leq (j-i)T \\ \frac{jT-\tau}{iT} & \text{for } (j-i)T < \tau \leq jT. \end{cases} && \text{(B.16)} \end{aligned}$$

For τ less than zero,

$$\begin{aligned} p(\tau \mid i < j, \tau < 0) &= \Pr\{t + \tau \in I_{t_o} \mid t \in I_{t_o}, i, j, k, i < j, \tau < 0\}, && \text{(B.17)} \\ &= \Pr\{-\tau < t < jT - \tau \mid 0 < t < iT, \tau < 0\}, && \text{(B.18)} \\ &= \Pr\{-\tau < t \mid 0 < t < iT, \tau < 0\}, && \text{(B.19)} \\ &= 1 - \Pr\{t < -\tau \mid 0 < t < iT, \tau < 0\}, && \text{(B.20)} \\ &= 1 - \frac{-\tau}{iT}, \qquad \text{for } -iT < \tau < 0. && \text{(B.21)} \end{aligned}$$

Therefore, the conditional cross-correlation function is obtained and

$$\begin{aligned} r(\tau \mid i, j, k \text{ and } i < j) &= p(\tau \mid i < j) \cdot \frac{iT}{T_f}, && \text{(B.22)} \\ &= \begin{cases} \frac{iT - |\tau|}{T_f} & \text{for } -iT < \tau \leq 0 \\ \frac{iT}{T_f} & \text{for } 0 < \tau \leq (j-i)T \\ \frac{jT-\tau}{T_f} & \text{for } (j-i)T < \tau \leq jT. \end{cases} && \text{(B.23)} \end{aligned}$$

B.3 Evaluation of $r(\tau \mid i, j, k \text{ and } i > j)$

The same computation can be done for i greater than j. The following results are obtained

$$r(\tau \mid i, j, k \text{ and } i > j) = \begin{cases} \frac{iT - |\tau|}{T_f} & \text{for } -iT < \tau \leq (j-i)T \\ \frac{jT}{T_f} & \text{for } (j-i)T < \tau \leq 0 \\ \frac{jT-\tau}{T_f} & \text{for } 0 < \tau \leq jT. \end{cases} \qquad \text{(B.24)}$$

All the preceding results can be generalized as follow: the conditional cross-correlation function is always a trapezoid of base width $(i+j)\mathcal{T}$, top width $|i-j|\mathcal{T}$ and height $\min(i,j)\cdot\frac{\mathcal{T}}{T_f}$. The lower left corner is at $\tau = -i\mathcal{T}$ and the lower right corner at $\tau = j\mathcal{T}$. The trapezoid is centered at $(j-i)/2$. Figure B.3 is an illustration of the correlation function for any i and j. Figure B.4 is for the case when i is less than j.

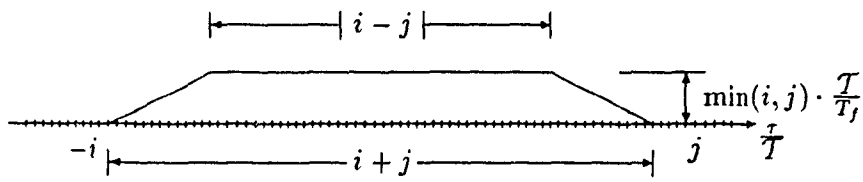


Figure B.3: Conditional cross-correlation function $r(\tau \mid i,j,k)$.

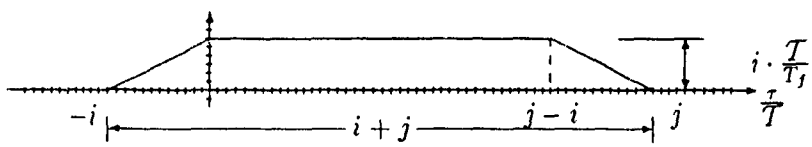


Figure B.4: Conditional cross-correlation function $r(\tau \mid i,j,k,i<j)$.

Appendix C

Example of the Estimation of M Superposed AR(1) Processes

This appendix concerns an example of the estimation of the superposition of M AR(1) processes. For this purpose, an example with M set equal to 2 will be used. The results can easily be extended to higher value of M.

A block diagram to represent the process at a system level would be as follows:

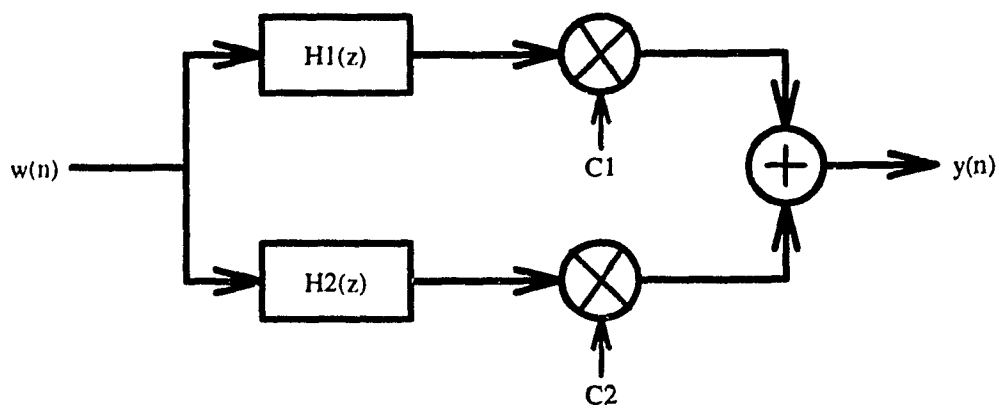


Figure C.1: Block diagram of for $M = 2$

where $w(n)$ is a zero-mean, white Gaussian noise sequence, c_1 and c_2 denote the number of sources of each type, and $H_1(z)$ and $H_2(z)$ are the AR(1) transfer functions. The output of the system is given by

$$y(n) = c_1 \cdot [a_1\, y_1(n-1) + b_1\, \omega(n)] + c_2 \cdot [a_2\, y_2(n-1) + b_2\, \omega(n)]\,. \tag{C.1}$$

To find what type of process is generated by this system, the transfer function is evaluated. Let $H(z)$ denote the system transfer function.

$$H(z) = [c_1\, H_1(z) + c_2\, H_2(z)]\,, \tag{C.2}$$

where

$$H_i(z) = \frac{b_i}{1 - a_i\, z^{-1}}. \tag{C.3}$$

Therefore,

$$H(z) \;=\; \frac{c_1\, b_1}{1 - a_1\, z^{-1}} + \frac{c_2\, b_2}{1 - a_2\, z^{-1}}.$$

$$= \frac{c_1\, b_1 + c_2\, b_2 - (c_1\, b_1\, a_2 + c_2\, b_2\, a_1)\, z^{-1}}{1 - (a_1 + a_2)\, z^{-1} + (a_1\, a_2)\, z^{-2}}. \tag{C.4}$$

This is a rational transfer function. If the input to the system is a white noise process—this is actually the case, then the output process is an autoregressive moving-average process [20, p.293]. The process defined by Equation (C.4) is an ARMA(2,1), and its transfer function can be rewritten as

$$H(z) = \sigma \cdot \frac{1 + \beta_1\, z^{-1}}{1 + \alpha_1\, z^{-1} + \alpha_2\, z^{-2}}, \tag{C.5}$$

where

$$\sigma = c_1\, b_1 + c_2\, b_2, \tag{C.6}$$

$$\beta_1 = -\frac{c_1\, b_1\, a_2 + c_2\, b_2\, a_1}{\sigma}, \tag{C.7}$$

$$\alpha_1 = -(a_1 + a_2), \tag{C.8}$$

$$\alpha_2 = a_1\, a_2. \tag{C.9}$$

Let assume now that we have the ARMA parameters estimates $\hat{\alpha}_1, \hat{\alpha}_2, \hat{\beta}_1$ and $\hat{\sigma}$. If the AR parameters a_1 and a_2 are unknown, then they can be estimated from Equations (C.8) and (C.9). The estimates of the a_i's are denoted by $\hat{a}_i$ and

$$\hat{a}_2 = \frac{-\hat{\alpha}_1 \pm \sqrt{\hat{\alpha}_1^2 - 4\hat{\alpha}_2}}{2}, \tag{C.10}$$

$$\hat{a}_1 = \frac{\hat{\alpha}_2}{\hat{a}_2}. \tag{C.11}$$

The AR parameters are real such that we must have $\hat{\alpha}_1^2 > 4\hat{\alpha}_2$. Further, it is assumed that the parameters are known such that the above relations can be used to adapt the estimation.

Assuming that we have the ARMA parameters (a_1, a_2, b_1, b_2), that we have the estimates $\hat{\beta}_1$ and $\hat{\sigma}$, we easily obtain the relations for $\hat{c}_1$ and $\hat{c}_2$.

$$\hat{c}_2 = \frac{\hat{\sigma}\,(\hat{\beta}_1 + a_2)}{b_2\,(a_2 - a_1)}, \tag{C.12}$$

$$\hat{c}_1 = \frac{\hat{\sigma} - b_2\, \hat{c}_2}{b_1}, \tag{C.13}$$

$$= \frac{\hat{\sigma}\,(\hat{\beta}_1 + a_1)}{b_1\,(a_1 - a_2)}. \tag{C.14}$$

Appendix D

Simulation Setup

This appendix presents the ATM link simulation setup and the DSP analytical tools. First the Section D.1 explains how the sources are simulated. In Section D.2, it is explained how the statistical multiplexing is carried out. The Section D.3 is about the experimental data obtained for video. Finally, the analytical tools that were used are presented in the last section. All the programs are in C language, and a total of about 8500 lines or 300 Kbytes were written.

D.1 Sources Simulation

The next two sections give the pseudo-code for the simulation of the voice and video sources. These algorithms were programmed in C language. The total size of the traffic stream simulator is around 3500 lines, and the numerical calculator (for the BAR and UAR video model) is about 2000 lines.

D.1.1 Voice sources

To simulate the voice sources, the binary Markov process of the Section 3.1 is used. Throughout, it is assumed that when the process is in the active state, it generates information at a uniform bit rate of B bits/sec.

The simulation of every sources is done as follow:

1. Set the initial state randomly ($\Pr\{Active\} = \frac{p}{p+q}, \Pr\{Idle\} = \frac{q}{p+q}$),
   - if active is selected, the fill randomly part of the current packet.
2. Generate a uniform random number δ between 0 and 1.
3. Compare δ with the threshold according to the current state p (idle) or q (active),
   - if δ is smaller than the threshold, the change of state.
4. If in active state, fill part of the current packet,

- if the packet is full, send it and start filling a new one.

5. Return to step 2 or stop if the simulation is finished.

In spite of the simplicity of this algorithm, some details must be mentioned. In step 1, the random filling of the initial packet ensures that the sources are not synchronized. In step 4, if the current packet is not full when a transition from active to idle occurs, it is not sent immediately; it waits until a certain time-out ends. For example, a PCM voice source at 64 kbps generates 0.17507 bits every cell. Hence, it takes about 2017 cells to fill a packet. To simplify the simulation, a packet is sent every 2017 cells and when a transition from active to idle occurs, the time-out is set equal to the remaining time to fill the current packet.

D.1.2 Video sources

In the Section 3.2, three models for video sources were presented. This section explains how each of them is simulated. Due to their similarity, the pseudo-code for the BAR model and UAR model is given simultaneously. Then the simulation of the minisource model is discussed.

Autoregressive Models

The BAR and UAR model relies on the same equation for the average bit rate of a frame. The only difference between these two models is how the information is sent to the network. For the BAR model, all the packets are sent as a burst, at a constant packet rate. For the UAR model, the packets are sent uniformly during the entire video frame duration.

The following pseudo-code has been implemented to simulate every source of both models.

1. Select randomly the initial bit rate ($\lambda(0) \sim \mathcal{N}(\mu, \sigma^2)$).

2. Set randomly the time elapsed in the current video frame, and fill randomly part of the current packet.

3. Send the packets according to the model (BAR or UAR) chosen for the remaining time of the current video frame.

4. Compute a Gaussian random number $w(n) \sim \mathcal{N}(\mu_{br}, \sigma_{br})$.

5. Compute the new bit rate: $\lambda(n) = a\lambda(n-1) + \omega(n)$.

6. Return to step 3 or stop if the simulation is finished.

As in the case of voice sources, some details must be mentioned. The first two steps guarantee that the video sources are not synchronized. In step 3, the packets are sent according to the model chosen; bursty or uniform. For the bursty procedure, all the packets are sent at a predetermined constant packet rate for the entire duration of the video frame. For the uniform procedure, the packets are sent a constant packet rate obtained by dividing the number of packets to send by the number of cells in the video frame. For example, if 200 packets have to be sent in an interval of 10000 cells, in the uniform procedure one packet will be sent every $10000/200 = 50$ cells. In the bursty procedure, assuming that the packet rate is one packet every 5 cells, the packets will be sent one every 5 cells on a period of $5 \times 200 = 1000$ cells, letting the remaining $10000 - 1000 = 9000$ cells empties.

Minisource Model

The minisources model can be simulated with three variations. First, lets reintroduce the original model. In [11] it is stated that *"the aggregate rate out of M voice sources (read as minisources) corresponds exactly to our quantizing aggregate bit rate of N video sources into M levels"*. The actual way the model is proposed is the following: M minisources generate bits in a common buffer where packetization is

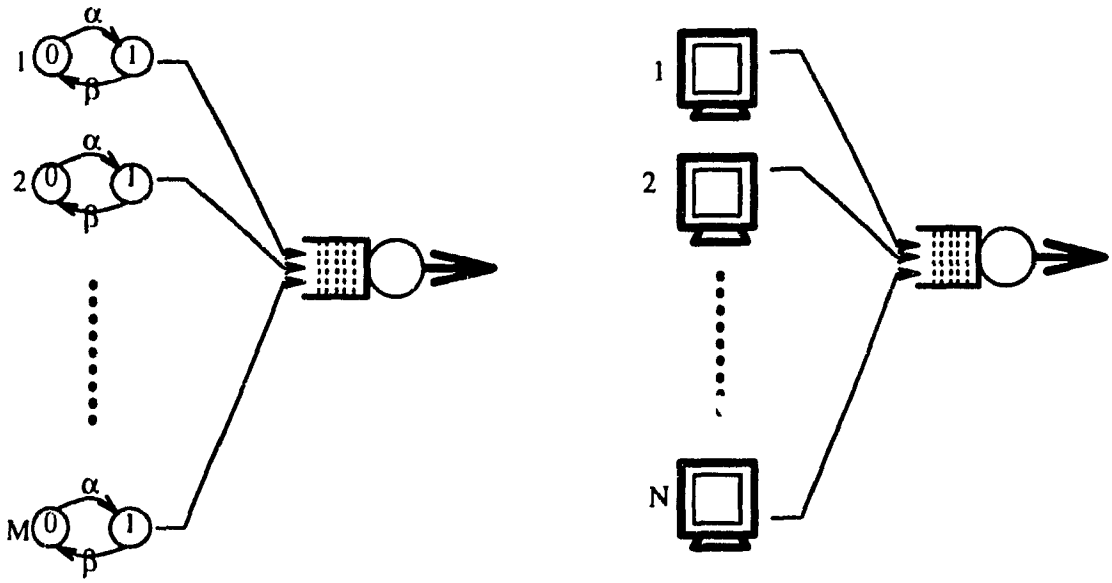


Figure D.1: Minisource model.

provided. It is the aggregate rate of the M minisources that is equal to aggregate rate of the N video sources. Figure D.1 illustrates the model.

The aggregate rate of the minisources can be simulated as defined or with slight variations. The following threes models have been implemented:

1. All the M minisources generate bits in a *common buffer*. The packetized output rate is the aggregate bit rate of the N video sources.

2. Each of the N video sources has its *own* packetizing buffer. Every source is modeled by an integer number $\kappa = \frac{M}{N}$ of minisources that fill its source buffer.

3. All the M minisources are independent, having their own packetizing buffer. Each minisource sends its own packets on the output line.

For all the models enumerated above, the minisources are simulated like the voice source with parameters $p = \alpha T$, $q = \beta T$. The parameters α, β and the bit rate A are given by Equations (3.48) to (3.50) according to the ratio κ chosen. The three models give very similar results for a large number of multiplexed sources, and depend on the choice of the ratio κ.

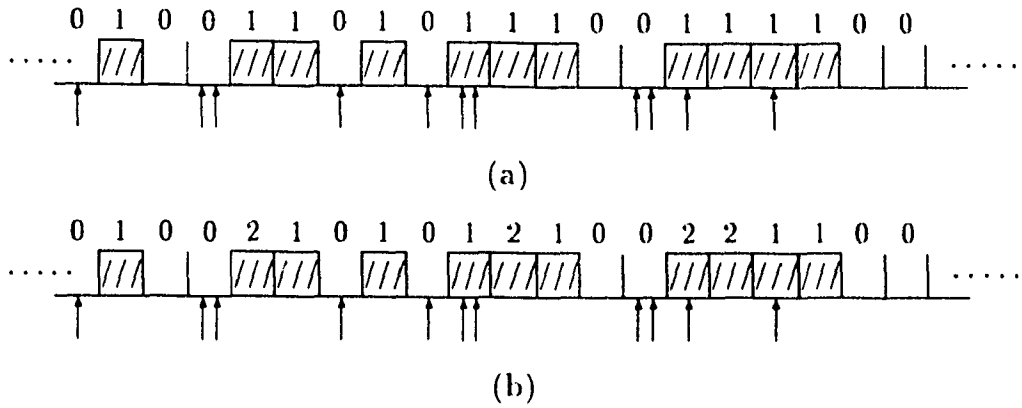


Figure D.2: Multiplexed traffic stream and output file

D.2 Statistical Multiplexing

The software written simulates a high-speed multiplexer with a large number of input lines, a single output line and buffers to resolve short-term contentions. To estimate the traffic composition, the only information available is the state of each cell: full or empty. Cell priority or not, random selection of colliding packet or not, the information obtained from the traffic stream remains the same, the presence or absence of a packet in a cell. At the beginning of every cell, the multiplexer scans all the input lines, stores the packets a contention buffer and sends one packet (if any in the buffer) to the output line. To observe the size of the contention buffer, instead of writing 0 and 1 in the output file, the actual number of packets in the buffer (before sending anyone to the output line) is written. Figure D.2(a) illustrates the multiplexing of arriving packets. Arrivals to the multiplexer are denoted by up-arrows and output packets by hachured boxes. The Figure D.2(b) shows what is actually written in an output file. The output file was generated this way to enable the estimation of the size of the contention buffer needed to achieve a certain packet loss probability for a given load.

D.3 Experimental Video Data

The experimental data for video was obtained from the University of Toronto [8]. The video source was either a video camera or a VCR. The NTSC analog signal was digitized at a rate of 30 frames per second with a resolution of 486×720 pixels with 8 bits per pixel. The original video sequence has a duration of 46 seconds with changes of scene.

To code the original sequence, four algorithms were used: interframe DPCM with run length coding, subband coding run length coding, multidimensional subband coding run length coding and discrete-cosine transform. For the four algorithms, the number of bits to send to the network was computed.

To generate the packet traffic stream, both the UAR and BAR models can be used. The only modification in the pseudo-code of Section D.1.2 is that the bit rate has no more to be computed, but is obtained from the real data.

D.4 Signal Processing Programs

The signal processing programs are written in C language. The Fourier transform program comes from the book *Numerical Recipes in C* [34], but has been modified to correct the error for the sign of the exponent in the definition of the Fourier transform in [34, p.398]. The total size of these tools is about 3000 lines.

D.4.1 Data Decimation

For the data decimation procedure, the digital filter coefficients were computed with MATLAB, using the functions *butter* and *cheby1*. Eighth order Butterworth, and type 1 Chebychev low-pass filters were used. The cutoff frequency was selected equal to $0.8 \times \frac{366780}{2 \cdot D}$, were D denotes the decimating factor. For stability reasons, D was kept smaller or equal to 30. When using Chebychev filter, the ripple was kept

smaller than 0.01 dB. Finally, the Butterworth filter was preferred to the Chebychev for its maximally flat response characteristic.

D.4.2 Periodogram

The periodograms were computed using N data points and $ZP \times N$ zeros padding. To get the good value of the power, normalizing was done as follows.

First, the FFT of the zero padded data sequence was computed.

$$X(f_m) = \sum_{i=1}^{(ZP+1)\cdot N} x(i) \cdot \exp(-j2\pi im/((ZP+1)\cdot N)), \tag{D.1}$$

where $f_m = \frac{m}{(ZP+1)\cdot N\Delta t}$. The periodogram estimates for the bin around frequency f_m was obtained by

$$\hat{P}(f_m) = \frac{\Delta t}{N} |X(f_m)|^2 . \tag{D.2}$$

When using a lag window, instead of dividing by N, division by $\sum_{i=1}^{N} \omega_i^2$ is done. Further, the windowing is applied to the data before zero padding.

D.4.3 Autocorrelation Function Estimates

For the autocorrelation function estimation, the unbiased estimator was used with a very long sequence. Further, since the correlogram and autocorrelation function estimates were used only to validate the numerically computed, theoretical autocorrelation function and power spectral density, the same number, N, of lag products were averaged, i.e. N data samples were utilized to get $R(0)$, $N+1$ for $R(1)$,..., and $N+k$ for $R(k)$.

The autocorrelation function has also been computed using the convolution property which is equivalent in the frequency domain to a multiplication. Autocorrelation function is then efficiently computed using the FFT algorithm.

D.4.4 Correlogram

To compute the correlogram, the FFT routine was also used taking into account its circular property. Given the M points, noncausal, discrete-time autocorrelation function, $\hat{R}(m)$, generates the causal autocorrelation sequence, $\hat{R}^{+}(m)$, of $N = 2^{n}$ data points [35, p.180]. Then obtain the N-points radix-2 FFT.

$$\hat{R}^{+}(m) = \begin{cases} \hat{R}(m) & \text{for } m = 0, 1, \ldots, M \\ 0 & \text{for } m = M+1, \ldots, N-M-1 \\ \hat{R}(m-N) & \text{for } m = N-M, \ldots, N-1. \end{cases} \tag{D.3}$$

The FFT results is multiply by Δt (the time period between two samples) to scale for good power spectral density level, as discussed in Section 4.2.2 and [27, pp.42-43].

D.4.5 AR(1) Parameters Estimation

The AR(1) parameters estimation is carried out using the maximum likelihood estimator exactly as described in Section 4.4.1. The estimation is done on the output of the counting process procedure, not directly to the traffic data stream.

D.4.6 Filtering Approaches

The filtering approaches are done simply by filtering the traffic stream with the appropriate filter. Then, either the power spectral density is estimated, or the power of the filtered signal is extimated.