ANALYSIS OF COMPUTER COMMUNICATION SYSTEMS BY USE OF DIGITAL SIGNAL PROCESSING METHODS

BY

RAYMOND JOHN CHATAMBUDZA KANYANGARARA

Submitted for the degree of

Doctor of Philosophy

at the

University of Aston in Birmingham

Department of Electrical and Electronic Engineering

September 1982

ANALYSIS OF COMPUTER COMMUNICATION SYSTEMS BY USE OF DIGITAL SIGNAL PROCESSING METHODS

RAYMOND JOHN CHATAMBUDZA KANYANGARARA

A thesis submitted to
The University of Aston in Birmingham
for the degree of
Docotor of Philosophy 1982

Summary

Queueing theory is an effective tool in the analysis of computer communication systems. Many results in queueing analysis have been derived in the form of Laplace and z-transform expressions. Accurate inversion of these transforms is very important in the study of computer systems, but the inversion is very often difficult. In this thesis, methods for solving some of these queueing problems, by use of digital signal processing techniques, are presented.

The z-transform of the queue length distribution for the $M|G^Y|1$ system is derived. Two numerical methods for the inversion of the transform, together with the standard numerical technique for solving transforms with multiple queue-state dependence, are presented.

Bilinear and Poisson transform sequences are presented as useful ways of representing continuous-time functions in numerical computations. These transforms permit digital signal processing techniques to be used in queue analysis. Systematic methods for inverting Laplace transforms using bilinear and Poisson transforms are obtained. The Poisson transform is also used in the analysis of the $M|G|1$ queue.

The inversion of Laplace transforms by use of bilinear and Poisson transforms requires the ability to compute bilinear and Poisson sequences from continuous-time functions and these functions from the sequences. Efficient computational methods are developed.

Bilinear and Poisson transforms are then used in: the inversion of the Laplace transforms of the waiting time distributions for the $M|G|1$ and $G|G|1$ queues, inversion of an irrational Laplace transform expression, and the computation of the busy period distribution for the $M|G|1$ queue. Bounds for the waiting time distribution in the $G|G|1$ queue are obtained using Skinner's method. The distribution for the number served in the $M|G|1$ queue busy period is computed using recurrence relations.

A method is derived to calculate the link delay estimate used in routing decisions in computer communication networks.

Key Words

COMPUTER COMMUNICATION SYSTEMS, QUEUEING ANALYSIS,

DIGITAL SIGNAL PROCESSING METHODS, TRANSFORMS, LINK DELAY ESTIMATE .

ACKNOWLEDGEMENTS

The author wishes to express his appreciation and gratitude to his supervisor, Dr. M. H. Ackroyd, for his valuable assistance and guidance throughout the period spent on this research. The author is grateful for the many helpful comments and discussions he had with Professor J. E. Flood, Mr. D. Bear, and Dr. P. M. Masterman (formerly at RSRE in Great Malvern) who suggested the problem on Link cost evaluation.

The author is also grateful for the financial support from the Association of Commonwealth Universities and the Africa Educational Trust, without which the author would not have been able to undertake this research.

Thanks are also due to colleagues for many helpful discussions. Finally, thanks are due to Mrs. Helen Turner for her patience in the typing of this thesis.

CONTENTS

	Page
List of Symbols	(i)
List of Figures	(v)
List of Graphs	(ix)
List of Tables	(xi)
CHAPTER 1 - INTRODUCTION	1
1.1 Computer Communication Systems	1
1.2 Queueing Analysis	2
1.3 Review of Existing Literature	5
1.4 Outline of Present Research	9
CHAPTER 2 - COMPUTING QUEUE LENGTH DISTRIBUTION FOR BULK SERVICE $M\|G^Y\|1$ SYSTEM	13
2.1 Introduction	13
2.2 z-transform of Queue Length Distribution for $M\|G^Y\|1$ System	15
2.3 Iterative Computational Scheme	20
2.4 Direct Computational Method	24
2.5 Solution by Root Finding and Linear Equations	31
2.6 Solutions for Queues $M\|M^Y\|1$ and $M\|E_2^Y\|1$	33
2.7 Accuracy and Limitations of Computational Methods	39
2.8 Conclusion and Application to Computer Communications	52
CHAPTER 3 - BILINEAR AND POISSON TRANSFORMS IN QUEUE ANALYSIS	55
3.1 Introduction	55

		Page
3.2	Poisson Transform in M|G|1 Queue Analysis	57
3.3	Discrete Representation of Continuous-Time Probability Functions	63
3.4	Bilinear and Poisson Transforms of Functions	67
3.5	Approximate Laplace Tranform Inversion	72
3.6	Laplace Transform Inversion by use of Bilinear and Poisson Transforms	77
3.7	Application to Analysis of Computer Communication Systems	87
CHAPTER 4	- COMPUTING BILINEAR AND POISSON SEQUENCES AND THEIR INVERSES	89
4.1	Bilinear and Poisson Sequences and their Inverses by Analogue Filtering	89
4.2	Digital Computation of Bilinear and Poisson Sequences	96
4.2.1	Impulse-Invariant Digital Filter Designs	97
4.2.2	Filter Designs Based on Numerical Integration Techniques	102
4.2.3	Performance and Comparison of Computational Methods	109
4.3	Inversion of Bilinear and Poisson Transforms	116
4.3.1	Inversion by use of Laguerre Polynomial Expansions	116
4.3.2	Inversion by use of Digital Filters	125
4.3.3	Performance of Computational Methods	129
4.4	Conclusions	143

Page

CHAPTER 5 - COMPUTATION OF VARIOUS PROBABILITY DISTRIBUTIONS IN QUEUE ANALYSIS 145

5.1 Introduction 145

5.2 Skinner's Method for Computing Bounds on Distributions 147

5.3 Distribution with Irrational Laplace Transform 156

5.4 Busy Period Distribution for M|G|1 Queue 159

5.4.1 Bilinear and Poisson Transforms of Busy Period Density 159

5.4.2 Computational Procedure 163

5.4.3 Solution for M|M|1 Queue 168

5.4.4 Accuracy of Computational Procedure 171

5.5 Number Served in the Busy Period of the M|G|1 Queue 177

5.6 Conclusions and Application to Analysis of Computer Communication Systems 181

CHAPTER 6 - LINK COST EVALUATION IN THE PILOT PACKET SWITCHED NETWORK 184

6.1 Introduction 184

6.2 Delay Through Line Unit i 187

6.3 Estimate of the Link Delay 190

6.4 Average of the Estimated Delay 193

6.5 Simulation and Numerical Examples 195

CHAPTER 7 - CONCLUSIONS AND SUGGESTIONS FOR FURTHER RESEARCH 199

7.1 Conclusions 199

7.2 Suggestions for Further Research 202

Page

APPENDICES

Appendix 1	Components of a Digital Filter	204
Appendix 2	Smith's Algorithm for Summation of Polynomial Series and their Derivatives	206
Appendix 3	Listings of Programs	210
	3.1 Listing of Program for Computing Bilinear Sequence of Waiting Time Distribution for G|G|1 Queue	210
	3.2 Listing of Program for Computing M|G|1 Queue Busy Period Distribution	221
REFERENCES		229

LIST OF SYMBOLS USED ≠ AND ABBREVIATIONS USED

Symbol	Definition
$a(t)$	Probability density function of the interarrival time
$a_b(k)$	Bilinear sequence of $a(t)$
$a_p(k)$	Poisson sequence of $a(t)$
$A_b(z)$	z-transform of sequence $a_b(k)$
$A_L(s)$	Laplace transform of $a(t)$
$A_p(z)$	z-transform of sequence $a_p(k)$
$b(t)$	Probability density function of service time
$b_b(k)$	Bilinear sequence of $b(t)$
$b_p(k)$	Poisson sequence of $b(t)$
$B_b(z)$	z-transform of sequence $b_b(k)$
$B_L(s)$	Laplace transform of $b(t)$
$B_p(z)$	z-transform of sequence $b_p(k)$
$c(t)$	Convolution of $a(-t)$ and $b(t)$ $(=a(-t)*b(t))$
$c_b(k)$	Bilinear sequence of $c(t)$
$c_p(k)$	Poisson sequence of $c(t)$
$C_b(z)$	z-transform of sequence $c_b(k)$
$C_L(z)$	Laplace transform of $c(t)$
$c_p(z)$	z-transform of sequence $c_p(k)$
$C(t)$	Cumulative distribution function corresponding to $c(t)$
DFT	Discrete Fourier Transform
$f(k)$	Probability of k customers being served in the busy period
FCFS	First come first served

≠ In cases where a symbol has more than one meaning, the context or a specific statement resolves the ambiguity.

Symbol	Definition
FFT	Fast Fourier transform
$F(z)$	z-transform of $f(k)$
$g(t)$	Probability density function of the busy period
$g_b(k)$	Bilinear transform sequence of $g(t)$
$g_p(k)$	Poisson transform sequence of $g(t)$
$G_b(z)$	z-transform of $g_b(k)$
$G_L(s)$	Laplace transform of $g(t)$
$G_p(z)$	z-transform of $g_p(k)$
$G(t)$	Cumulative distribution function corresponding to $g(t)$
$h_n(t)$	Unit impulse response of an analogue filter
$H_n(s)$	Laplace transform of $h_n(t)$
$H_n(z)$	z-transform obtained by replacing z in $H_n(s)$ by bilinear or Poisson transformations
$Im[X]$	Imaginary part of complex variable X.
L	Block size for the fast Fourier transform algorithm
LCFS	Last come first served
$\ell_n(t)$	Laguerre polynomial function of degree n
$L_n(t)$	Laguerre functions of degree n
m	Service capacity for a queueing system with fixed service capacity
N	Maximum service capacity for a queueing system with a random service capacity
$\tilde{q}_n$	Number of customers in the queue just prior to the nth service instant
$q(k)$	$\lim \tilde{q}_n(k)$ as $n\to\infty$

Symbol	Definition
$q_n(k)$	Probability of having k customers in the system just prior to the nth service period
$Q(z)$	lim $Q_n(z)$ as $n \to \infty$ or z-transform of $q(k)$
$Q_n(z)$	z-transform of $q_n(k)$
$r(t)$	Probability density function of total time spent in system
$Re[X]$	Real part of complex variable X
$R_L(s)$	Laplace transform of $r(t)$
s	Laplace transform variable
$\tilde{s}_n$	Service capacity of server, at service instant n
$s(k)$	lim $s_n(k)$ as $n \to \infty$
$s_n(k)$	Probability of k customers being served at service instant n
$S(z)$	lim $S_n(z)$ as $n \to \infty$ or z-transform of $s(k)$
$S_n(z)$	z-transform of $s_n(k)$
t	Time variable
T	Sampling period
$\tilde{v}_n$	Number of customers arriving during the (n-1)th and nth service instants
$v(k)$	lim $v_n(k)$ as $n \to \infty$
$v_n(k)$	Probability of k customers arriving during the (n-1)th and nth service instants
$V(z)$	lim $V_n(z)$ as $n \to \infty$ or z-transform of $v(k)$
$V_n(z)$	z-transform of $v_n(k)$
$w(t)$	Probability density function of the waiting time

Symbol	Definition
$w_b(k)$	Bilinear sequence of $w(t)$
$w_p(k)$	Poisson sequence of $w(t)$
$W(t)$	Cumulative distribution function corresponding to $w(t)$
$W_b(z)$	z-transform of $w_b(k)$
$W_L(s)$	Laplace transform of $w(t)$
$W_p(z)$	z-transform of $w_p(k)$
z	z-transform variable
α	Bilinear parameter
$\delta(t), \delta(k)$	Delta function
γ	Poisson parameter
λ	Arrival rate
μ	Service rate
ρ	Utilization factor
$\Pi(.)$	Process of sweeping the negative time part of a function up to the origin
$(x)^+$	$\mathrm{Max}(0,x)$
$\binom{n}{k}$	Binomial coefficient $\frac{n!}{(n-k)!k!}$

LIST OF FIGURES

Page

Chapter 3

Fig. 1 Diagram showing how the Poisson transform and its inverse interrelates the various probability density functions for the M|G|1 queue. 58

Fig. 2 Digital filter for generating q(k), the probability of having k customers in the M|G|1 queue. Input to digital filter is v(k), the probability of k arrivals in a service period and output is q(k). 60

Fig. 3(a) The mapping from the s-plane to the z-plane corresponding to the bilinear transformation. 70

Fig. 3(b) The mapping from the s-plane to the z-plane corresponding to the Poisson transformation. 70

Chapter 4

Fig. 1 All-pass network for conversion of a continuous-time function x(t) to its bilinear representation $x_b(n)(=\tilde{x}_n(0))$. 92

Fig. 2 The all-pass analogue chain for recovering a continuous-time function x(t) from its bilinear representation $x_b(n)$. 92

Page

Fig. 3 Analogue filter chain for conversion of a continous-time function x(t) to its Poisson elements $x_p(n)$ $(=\tilde{\tilde{x}}_n(0))$. 94

Fig. 4 Analogue filter chain for recovering a continuous-time function x(t) from its Laguerre series coefficients $x_\ell(n)$. 94

Fig. 5 Impulse invariant digital filter design for generating Poisson elements $x_p(n)$ from periodic samples of a function x(t). 99

Fig. 6 Impulse invariant digital filter design for generating bilinear elements $x_b(n)$ from periodic samples of x(t). 103

Fig. 7 Digital filter design, based on the rectangular integration rule, for generating the bilinear sequence $x_b(n)$. The input to digital filter are the periodic samples of the function x(t), in reverse order. $x_b(n)$ is obtained at the output of the (n+1)th stage when the last sample x(o) is processed. 107

Fig. 8 Digital filter design, based on the rectangular integration rule, for generating the Poisson elements $x_p(n)$. The input to digital filter are periodic samples of the function x(t), in reverse order. $x_p(n)$ is obtained at the output of the (n+1)th stage when the last sample x(o) is processed. 108

Page

Fig. 9 Digital filter design, based on the trapezoidal integration rule, for generating the bilinear sequence $x_p(n)$, of a function $x(t)$. The input to digital filter are the periodic samples of $x(t)$ time reversed. $x_b(n)$ is obtained at the output of the $(n+1)$th stage when the last sample $x(o)$ is processed. 110

Fig. 10 Digital filter design, based on the trapezoidal integration rule, for generating the Poisson sequence $x_p(n)$. The input to digital filter are periodic samples of the function $x(t)$, time-reversed. $x_p(n)$ is obtained at the output of the $(n+1)$th stage when the last sample $x(o)$ is processed. 111

Fig. 11 Digital filter for generating Laguerre series coefficients $x_\ell(n)$ from Poisson elements $x_p(k)$. 118

Fig. 12 Digital filter for generating the bilinear sequence $x_b(k)$ from the Poisson sequence $x_p(n)$. 123

Fig. 13 Digital filter design, based on the rectangular integration rule, for generating $x(t)$ from its bilinear representation $x_b(n)$. 127

Fig. 14 Digital filter design, based on the rectangular integration rule, for generating 128

Page

the function x(t) from Laguerre series coefficients $x_\ell(n)$. 128

Fig. 15 Digital filter design, based on the trapezoidal integration rule, for generating the function x(t) from its bilinear representation $x_b(n)$. 130

Fig. 16 Digital filter design, based on the trapezoidal integration rule, for generating x(t) from Laguerre series coefficients $x_\ell(n)$. 131

Appendix 1

Fig. 1 The components of a digital filter. 205

LIST OF GRAPHS

Page

Chapter 2

Graph 1 Plot of queue length probability density function for the $M|E_2^Y|1$ queue with various service capacity distributions for $\rho=0.1$. 50

Chapter 4

Graph 1 Plot of service time function E_4 computed by the three approaches, for service rate $\mu=\frac{10}{9}$ and $\alpha=\gamma=\lambda=1$. 138

Graph 2 Plot of service time function E_4 computed by the three approaches, for service rate $\mu=10$ and $\alpha=\gamma=5$. 139

Graph 3 Plot of service time function E_4 computed from the three approaches, for service rate $\mu=10$ and $\alpha=\gamma=20$. 140

Chapter 5

Graph 1 Plot of the busy period function for the $M|M|1$ queue for $\mu=10$ and $\lambda=1$. Curve (a) is the function computed with $\alpha=1.0$. Curve (b) is the exact solution from Eqn (25). Curve (c) is the function computed with $\alpha=9.999$. 173

Graph 2 Plot of busy period function for the $M|M|1$ queue. Curve (a) is a plot of computed results from Eqn (25). Curve (b) is a plot 175

Page

of results from numerical inversion of Laplace transform of busy period density. 175

Graph 3 Comparison of busy period function for the $M|M|1$ and $M|E_4|1$ queues with $\lambda=1$ and $\mu=10$. 176

LIST OF TABLES

Page

Chapter 2

Table 1 Root of denominator of Q(z) for $M|M^Y|1$ queue with a constant service capacity m. 40

Table 2 Roots of Denominator of Q(z) for $M|E_2^Y|1$ queue with a constant service capacity m. 41

Table 3 Comparison of results from iterative method and root-finding method for the queue $M|M^Y|1$ with a constant service capacity m. 42

Table 4 Comparison of results from iterative method and root-finding method for $M|E_2^Y|1$ queue with a constant service capacity m. 43

Table 5 Comparison of results from direct computational method and root-method for $M|M^Y|1$ queue with a constant service capacity m. 46

Table 6 Comparison of results from direct computational method and root-method for $M|E_2^Y|1$ queue with a constant service capacity m. 47

Table 7 A comparison of results from the direct computational method and the root-finding method for the $M|E_2^Y|1$ queue with different service capacity distributions. 49

Chapter 3 | Page

Table 1 Comparison of approximate and exact waiting time distributions for the M|M|1 queue with $\lambda=1$, $\rho=0.1$, and different values of sampling period T. 75

Chapter 5

Table 1 Upper and lower bounds on the waiting time distribution function for the M|M|1 queue with $\lambda=1$ and $\mu=1.5$. 154

Table 2 Upper and lower bounds on the waiting time distribution function for the E_2|M|1 queue. 155

Chapter 6

Table 1 Simulation to determine the delay which will be experienced by a packet which arrives to find 5 packets in the Link Controller queue and $q_1(1)=1$, $q_2(1)=3$, $q_3(1)=5$, $w_1(1)=12$, $w_2(1)=8$ and $w_3(1)=4$. After queueing for 5 time units in the Link Controller queue, the packet is allocated to Line Unit 3 and it will spend 31 time units in this queue. 197

Table 2 Simulation to determine the delay which will be experienced by a packet which arrives to find 9 packets in the Link Controller queue and $q_1(1)=2$, $q_2(1)=3$, $q_3(1)=5$, $w_1(1)=5$, $w_2(1)=2$ 198

Page

and $w_3(1)=1$. After queueing for 9 time units in the Link Controller queue, the packet is allocated to Line Unit 1 and will spend 32 time units in this queue. 198

CHAPTER 1

INTRODUCTION

1.1 COMPUTER COMMUNICATION SYSTEMS

In computer communication systems, users are connected to computers by communication links. Computers can also be connected to other computers, which can be at geographically distributed locations, through communication links. Computer communication systems range from the various forms of teleprocessing used in the data processing industry, the time-sharing systems between collections of terminals and central computers, to the large scale computer-to-computer communication networks. Many large scale networks have been developed and are now in operation. Examples of these are the ARPANET, the trans-Canada telephone system, the multipurpose data network set up by the British Post Office. Other systems are in operation and they include airline reservation systems, medical data systems, banking networks, military networks, educational networks and information service systems.

There are several important issues in the analysis and design of computer communication systems. These issues include performance evaluation, routing and flow control, link capacity assignment, topological design, buffer size allocation. For a cost-effective design, one must have systematic methods for predicting quantitative relations between the system input parameters, system performance measures and system workloads.

1.2 QUEUEING ANALYSIS

The class of systems that generally lends itself to queueing analysis is one in which customers compete for access to a resource which can perform work at some finite rate, i.e., a finite-capacity resource. Resources that customers compete for include terminals, communication lines connecting these terminals to the computer system, storage capacity in the main memory and in the secondary memory devices and processing capacity of both the central processing unit and the input/output processor that controls the flow of data between the main memory and secondary devices. Customers can be for instance, terminal users, data packets, speech and other computers. Queueing theory is an effective tool for studying the throughput, response and other measures of performance for computer systems. There are other disciplines of applied mathematics, like graph theory, mathematical programming, optimization techniques and reliability theory, which must augment the queueing analysis in order to cope with the overall analysis and design issues.

Three different stages can be identified in the analysis and modelling of computer communication systems(1). At the first stage, a suitable queueing model for the system is developed. The second stage involves the identification and investigation of the basic process underlying the whole model. It is at this stage that quantitative relations between system input parameters, system workloads and measures of system performance are derived. The third stage involves obtaining numerical results from the derived quantitative relations. Approximations can be introduced at any one of these stages. If an exact or an approximate queueing model, or process or numerical solution cannot be obtained, one resorts to simulation of the system. This can be very costly, and it is therefore preferable to obtain functional relations from suitable queueing models wherever possible.

There are several performance measures for a queueing system. These include server utilization or utilization factor, throughput, the time a customer has to wait in a queue, the total time a customer must spend in the system or end-to-end delay, the duration of the busy and idle periods, the number of customers in the queue, the number of customers served in the busy period. Percentile performance specifications are also important in the design of a computer communication system. These percentile specifications are obtained from the detailed distribution functions of the system variables. For example, a performance specification for a computer communication network can be that the end-to-end delay for a data packet must be 3 seconds or less, 99% of the time.

In order to specify the type of a given queueing system, Kendall's shorthand notation $A^X|B^Y|m|N$ is used. A and B describe the interarrival time distribution and the service time distribution, respectively. X and Y represent the distributions of the sizes of the arriving groups and the service batches, respectively. m represents the number of parallel servers of the servicing system. N is the maximum number of customers that can wait in the queue.

The following is a list of distributions used or referred to in the subsequent sections and chapters.

M Exponential distribution.

D Deterministic variable, i.e., interarrival times/ service times are constant values.

E_k k-stage Erlangian distribution.

H_k k-stage hyperexponential distribution.

G General distribution.

1.3 REVIEW OF EXISTING LITERATURE

There is a large literature on queueing theory and queueing models for computer communication systems[2-38]. Many useful results for performance measures, and parameters of interest have been obtained as averages, variances, moments, Laplace and z-transforms. The numerical evaluation of the derived expressions is not always easy. Some work has been done in the development of numerical methods for the solution of these queueing problems[29,39-69]. However, this forms only a very small proportion of the total literature on queueing theory. In the present age of computers, there has been a surprising concentration on deriving analytical results rather than numerical solutions.

The solution of queueing problems can involve the approximation of the actual queueing models by simpler models for which solutions are available. Results for the simple M|M|1 queue have been used extensively in the transit delay analysis for computer communication networks. Kleinrock[3] has used these results in predicting the delay of messages flowing through a network. Sometimes the basic process underlying the mathematical model can be so complex that a direct analysis for the model is not worthwhile. The system model can be simplified. Alternatively, a simpler process whose analysis is known can be identified and used. Fluid approximation[3,38], a first-order approximation, and diffusion approximation[3,39,40], a second-order approximation, have been used in process approximation.

Numerical approximation is also useful in the solution of queueing problems. Bear[41] obtained an approximate method for calculating the waiting time distribution in an M|G|1 queue. The method computes the approximate distribution from the first two moments using standard delay curves for the M|M|1 and M|D|1 queues. A class of approximations for the waiting time distribution in the G|G|1 queue are presented in Reference (42). The distribution is obtained as exponential approximations. Cosmetatos[43] considers the multiserver system G|M|m. Approximate formulas are developed for evaluating the steady-state average queueing time for the system with hyperexponential interarrival and exponential service times. Data to represent the different distribution functions are often obtained by measurements or statistical observations. For numerical calculations, it may be necessary to approximate the measured data by some distribution function. Bux and Herzog[44] presented an algorithm for determining a phase-type distribution function which fits the actual measured data with a prescribed accuracy.

In certain situations, it is easier to obtain numerical results for discrete queueing systems than for continuous-time systems. When discrete solutions are used to approximate continuous solutions, the continuous-time distributions are replaced by discrete distributions obtained by periodic sampling. Ackroyd[45,46] has used this technique to compute the waiting time distribution function for the G|G|1 queue and the busy period distribution for the M|G|1 queue. Sampled discrete distributions have also been used in the exact formulations of the continuous solution. Molloy[47] and Wong[48] computed the waiting time distribution for the continuous G|G|1 queue, by replacing the continuous-time distributions with their sampled versions in the exact formulation of the continuous system.

Bounds on the mean, the moments and distribution functions are sometimes derived. This approach is more useful than having approximate results whose accuracy is not known. Raychaudhuri and Rappaport[16] derived the z-transforms of the queue length distribution for the sampled, multiserver G|D|m queues. From the transforms, they derived upper and lower bounds on the average waiting time. Skinner[49] developed an algorithm for computing, numerically, upper and lower bounds on the waiting time distribution function for the M|G|1 queue. Skinner's method was shown to be applicable to many other problems[50,51].

Bagchi and Templeton[52] presented an algorithm for computing, numerically, the time-dependent probabilities for discrete variables in a sizeable class of bulk queues and queues with non-stationary input and service conditions. Holman, Grassmann and Chaudhry[29,30] derived the z-transform of the number of elements in the system, for the bulk arrival, multiserver system $E_k^X|M|c$ and they also presented a numerical method for the inversion of the derived transform.

A common problem in the analysis of queueing systems is the inversion of the Laplace transform. Some numerical methods for inverting this transform have been developed[53-59]. Some of the methods have been used in the analysis of computer communication systems. The busy period distribution for an $M|G|1|N$ type queue occurring in a time-sharing computer system was computed in reference 60 using some of these methods. An IBM disk storage facility was modelled as an $M|G|1$ queue[61], and its response time was computed by the numerical method in reference 58.

Digital signal processing techniques have recently been applied in the solution of queueing problems[45-47,62-68]. The fast Fourier transform (FFT) algorithm was proposed as a means of inverting the z-transforms of probability density functions in queueing analysis by Caver[67]. The FFT algorithm and the complex cepstrum were used in the computation of the waiting time distribution for the $G|G|1$ queue[45,47]. The various distributions for the $M|G|1$ queue were computed in references 46, 62-66. The solution for the $M|M|1$ transient-state occupancy probability is given in reference 2 as a series summation of the modified Bessel functions. Stern[69] obtained a simple approximation to this transient solution. However, Ackroyd[68] used the inverse discrete Fourier transform (DFT) to compute the transient-state occupancy probabilities.

1.4 OUTLINE OF PRESENT RESEARCH

Digital signal processing techniques have been used in the processing of radar, speech, brain and communication signals. In queueing theory, the signals are probability density and distribution functions rather than waveforms. The task of numerically evaluating the results derived in queueing analysis is often difficult. Therefore the numerical solution of queueing problems is important in the analysis of computer communication systems. This research involves the development of systematic methods for solving the queueing problems by use of digital signal processing techniques.

In Chapter 2, the z-transform of the queue length probability density function for the bulk service $M|G^Y|1$ queue is derived. The inversion of the derived transform is difficult because each queue state depends on several other queue states. In sections 2.3 and 2.4, two methods for the inversion of the transform by use of digital signal processing techniques are presented. The standard numerical solution for transforms with multiple queue-state dependence is outlined in section 2.5. Solutions for some queueing systems are obtained in section 2.6. To test and assess the accuracy of the computational methods, the methods were tried for these queues and details of the comparison are given in section 2.7. The chapter concludes in section 2.8 with a discussion of bulk service and multiserver queues in computer communications.

Chapter 3 deals with the application of the bilinear and the Poisson transforms in queue analysis. The Poisson transform interrelates the various distributions of the M|G|1 queue. These relationships, together with the inversion of the Pollaczek-Khinchin transform equation for the number of customers in the system , are discussed in section 3.2. The theory of representing continuous-time functions by digital sequences is discussed in section 3.3. The bilinear and Poisson sequences are then presented in section 3.4 as useful discrete function representations suitable for numerical calculations. An approximate Laplace transform inversion procedure by use of the bilinear transformation is outlined in section 3.5 and this method is used in the computation of the waiting time distribution for the M|G|1 queue. A systematic procedure for the computation of the inverse Laplace transform by use of the bilinear and Poisson transforms and digital signalprocessing techniques is presented in section 3.6. The procedure is used in the inversion of the Laplace transform of the waiting time distribution for the G|G|1 queue.

In Chapter 4, several methods for computing the bilinear and Poisson sequences and their inverses are presented. The sequences are generated from digital filters derived from analogue filters by use of the impulse invariance and numerical integration design procedures. The first inversion technique involves the use of Laguerre polynomial series expansions and the second technique involves the use of digital filters obtained by the numerical integration design procedures.

Some probability distribution functions are computed in Chapter 5. Skinner's method [49] is outlined and used in the derivation of bounds on the waiting time distribution for the $G|G|1$ queue. The bilinear transform is used in the inversion of a transcendental Laplace transform. The bilinear and Poisson transforms of the busy period probability density function for the continuous $M|G|1$ queue are derived. A procedure for the computation of the busy period distribution function from these transforms is outlined and used in obtaining numerical solutions for the $M|M|1$ and $M|E_4|1$ queues. To test the computational method, the exact solution for the $M|M|1$ queue is rearranged into a set of recurrence relations, and results from the two methods are compared. A recurrence procedure for computing the distribution of the number served in the busy period for the $M|G|1$ queue is derived. The chapter concludes with a discussion of the application of the computational methods in the analysis of computer communication systems.

In Chapter 6, an algorithm for computing an estimate of the link cost, for a computer communication network, is developed. In making routing decisions, a node must have an estimate of the delay a packet can experience along all possible routes to a particular node. The information about the delay a packet can experience along a given link between two nodes is stored in the link cost table. The algorithm was developed specifically for the Pilot Packet-Switched Network[70].

Finally, the conclusions are to be found in Chapter 7, together with suggestions for further work.

CHAPTER 2

COMPUTING QUEUE LENGTH DISTRIBUTION FOR BULK SERVICE $M|G^Y|1$ SYSTEM

2.1 INTRODUCTION

In the next section of this chapter, the z-transform of the queue length probability density function for the $M|G^Y|1$ queue is derived. This queue has an exponential interarrival time probability density function, a general service time distribution, and a single server capable of serving a random number of customers. The inversion of the derived transform is difficult because each queue state depends on some of the other queue states.

A common numerical procedure for inverting transforms with multiple-state dependence involves the use of a root-finding routine and a routine for solving a system of linear equations. In the absence of an exact solution to this problem, it is desirable to experiment with several numerical techniques. In sections 2.3 and 2.4, two methods for inverting the derived z-transform are presented. These methods use digital signal processing techniques. The first method is a frequency domain iterative scheme. The second is a direct computational method which uses the complex cepstrum. The common procedure for inverting such transforms is briefly described in section 2.5. In section 2.6, the solutions for the queues $M|M^Y|1$ and $M|E_2^Y|1$ with constant, binomial and geometric service capacities are obtained by use of a root-finding procedure.

The solutions for these queues are used to assess the performance of the two computational methods presented, in section 2.7. The computational methods are also applicable in the inversion of transforms for some multiserver queues[16,20]. The chapter concludes in section 2.8 with a discussion of bulk service and multiserver systems in computer communications.

2.2 Z-TRANSFORM OF QUEUE LENGTH DISTRIBUTION FOR $M|G^Y|1$ SYSTEM

The $M|G^Y|1$ queue with a constant service capacity, m, is considered in references 2 and 10. The system operates as follows: service is instantaneous, but is only available at service instants. The interval between successive service instants is independently and identically distributed with probability density function b(t). If there are $\tilde{q}_n$ customers waiting in the queue just before service instant n, $\min(\tilde{q}_n, m)$ customers are served at service instant n, where m is the service capacity.

Consider the same system, but with the service capacity of the server at any service instant a random variable. This system corresponds to the $M|G^Y|1$ queue[52,91]. Let $\tilde{s}_n$ be the service capacity of the system at service instant n. At service instant n, $\min(\tilde{s}_n, \tilde{q}_n)$ customers are served.

Let $\tilde{v}_n$ be the number of customers arriving during the time interval between the (n-1)th and nth service instants. The basic equation of the system, giving the number of customers in the queue, just prior to the (n+1)th service instant is[52]

$$\tilde{q}_{n+1} = (\tilde{q}_n - \tilde{s}_n)^+ + \tilde{v}_{n+1} \qquad (1)$$

where the notation $(i)^+$ denotes max (0,i). The following discrete probability density functions and their z-transforms are defined:

$q_n(k) = pr(\tilde{q}_n=k)$ with z-transform $Q_n(z)$,

$s_n(k) = pr(\tilde{s}_n=k)$ with z-transform $S_n(z)$, and

$v_n(k) = pr(\tilde{v}_n=k)$ with z-transform $V_n(z)$.

To derive the z-transform of $q_{n+1}(k)$ from Eqn (1), the following facts are used:

(a) The z-transform of $(\tilde{x}_n)^+$ is $\Pi(X_n(z))$[52], where $\Pi(.)$ denotes the operation of "sweeping values on the negative axis to the origin", and where $X_n(z) = \Sigma pr(\tilde{x}_n=k)z^{-k}$.

(b) The z-transform of the probability density function of a variable $-\tilde{x}_n$ is $X_n(\frac{1}{z})$.

(c) The variables $\tilde{v}_n$ and $\tilde{s}_n$ are independent of $\tilde{q}_n$, and of each other.

Using these facts, and the z-transforms of the probability density functions of the variables $\tilde{q}_n$, $\tilde{s}_n$ and $\tilde{v}_n$, $Q_{n+1}(z)$ the z-transform of $q_{n+1}(k)$, is

$$Q_{n+1}(z) = \Pi[Q_n(z)\ S_n(\tfrac{1}{z})]\ V_{n+1}(z) \tag{2}$$

Expanding this equation and performing the $\Pi(.)$ operation, Eqn. (2) becomes

$$Q_{n+1}(z) = \Big[Q_n(z) S_n(\tfrac{1}{z}) - \sum_{k=1}^{N} s_n(k) \sum_{j=0}^{k-1} q_n(j) z^{-j+k} +$$

$$\sum_{k=1}^{N} s_n(k) \sum_{j=0}^{k-1} q_n(j)\Big] V_{n+1}(z) \qquad (3)$$

where N is the maximum service capacity of the system. Expanding the right side of Eqn. (3) in ascending powers of z^{-1}, and matching them to similar powers of z^{-1} on the left side, $q_{n+1}(k)$ is found to be

$$q_{n+1}(k) = \sum_{i=0}^{k} v_{n+1}(i) \sum_{m=0}^{N} s_n(m) q_n(m+k-i) +$$

$$v_{n+1}(k) \sum_{i=1}^{N} s_n(i) \sum_{m=0}^{i-1} q_n(m), \quad k=0,1,2,... \qquad (4)$$

In the steady-state, the subscripts n and n+1 can be dropped in Eqns.(3) and (4). Solving Eqn. (3), the z-transform of the steady-state queue length probability density function for the $M|G^Y|1$ queue is

$$Q(z) = \frac{V(z) \sum_{k=1}^{N} s(k) \sum_{i=0}^{k-1} q(i)(1-z^{-i+k})}{1 - V(z)\, S(\frac{1}{z})} \qquad (5)$$

The average queue length, $\bar{q}$, is obtained from the condition $dQ(z)/dz^{-1}|_{z^{-1}=1}$, with the help of l'Hôpital's rule. This is given by the equation

$$\bar{q} = \frac{\sum_{n=1}^{N} s(n) \sum_{i=0}^{n-1} (i-n)(i-n-1)q(i)}{2(\bar{v}+\bar{s})} - \frac{\left[\bar{\bar{v}}+\bar{\bar{s}}-2(\bar{v})^2\right]}{2(\bar{v}+\bar{s})^2} \times \sum_{n=1}^{N} s(n) \sum_{i=0}^{n-1} (i-n)\, q(i) \tag{6}$$

where

$$\bar{v} = \left.\frac{dV(z)}{dz^{-1}}\right|_{z^{-1}=1} \; ; \qquad \bar{\bar{v}} = \left.\frac{d^2V(z)}{dz^{-2}}\right|_{z^{-1}=1}$$

$$\bar{s} = \left.\frac{-dS(z)}{dz^{-1}}\right|_{z^{-1}=1}$$

$$\bar{\bar{s}} = \left.\frac{d^2S(z)}{dz^{-2}} + 2\,\frac{dS(z)}{dz^{-1}}\right|_{z^{-1}=1}$$

Equations similar to (4) and (5) were obtained in reference 20, for a multiserver system with general interarrival time, constant service time, several servers, but with the number of servers available at any service instant being a random variable.

To obtain the detailed queue length distribution, Eqn. (5) must be inverted. Unfortunately, the explicit solution of this equation is difficult because of the multiple dependence of the queue states. To obtain the density $q(k)$ requires a knowledge of the values of several other elements. Even to compute the average queue length from Eqn (6) requires a knowledge of the values of the first N elements. As N becomes large, the problem is further complicated. Other reasons that would make the inversion of this transform, Eqn. (5), complicated are discussed below.

If V(z) and S(z) are irrational functions the task of analytically inverting this transform becomes more difficult. For Poisson arrivals, it is sometimes sufficient to specify the service time distribution without specifying v(k), the probability of k arrivals in a service period. For this arrival process, one has to generate the density v(k) from the service time density b(t). v(k) is the Poisson sequence of the service time density for this arrival process[2]. The problem of generating this sequence from a continuous-time function is discussed in Chapter 4.

The common numerical method for inverting z-transforms (like Eqn. 5) with multiple-state dependence is by solving for the roots of the denominator and then solving a system of linear equations generated by use of these roots. In the absence of an exact solution, experimenting with several numerical methods provides a mutual check on the agreement of the computed results as noted in references 1, 55 and 60. Afterall, the performance of a method depends on the nature of the function which is being computed.

2.3 ITERATIVE COMPUTATIONAL SCHEME

Time-domain iterative methods for solving bulk systems have been considered in reference 52. Equation (4) gives a direct recurrence relation for the transient queue length distribution and this can be used to compute the steady-state solution for this distribution. If L elements of this distribution are computed per iteration, then at least $NL^2/2$ multiplications per iterations are required. Most of these multiplications are contributed by the first term which involves convolutions. Convolutions in the time-domain correspond to multiplications, in the frequency-domain. Therefore the frequency-domain iterative procedure can reduce the number of multiplications, considerably, for certain values of N and L.

Equation (4) can be transformed into a frequency-domain recurrence relation by obtaining its discrete Fourier transform (DFT). If x(k) is a finite sequence, its DFT is, by definition[72-74]

$$X(m) = \sum_{k=0}^{L-1} x(k)\exp\left(-j\frac{2\pi km}{L}\right), \quad m=0,1,\ldots,L-1 \tag{7}$$

The inverse DFT is

$$x(k) = \frac{1}{L}\sum_{m=0}^{L-1} X(m)\exp\left(j\frac{2\pi mk}{L}\right), \quad k=0,1,\ldots,L-1 \tag{8}$$

where π is used in its usual context of $\pi=3.141\ldots$.

Efficient algorithms for computing X(m) from x(k) and vice-versa have been developed. The most commonly used are known as the fast Fourier transform (FFT) algorithms[72-74].

Let the DFT's of the probability density functions $q_n(k)$, $s_n(k)$ and $v_n(k)$ be, respectively, $Q_n^*(\ell)$, $S_n^*(\ell)$ and $V_n^*(\ell)$. Using Eqns. (7) and (4), the DFT of $q_{n+1}(k)$ can be obtained. This would be a laborious process. However, this DFT can be obtained from the z-transform of $q_{n+1}(k)$ which is given in Eqn. (3). To transform the z-transform to a DFT, z is replaced by $\exp(j\frac{2\pi\ell}{L})$[67,72-74]. Using this transformation, the frequency-domain recurrence relation for the DFT of the transient queue length probability density function is

$$Q_{n+1}^*(\ell) = \{Q_n^*(\ell)S_n^*(-\ell) + \sum_{k=1}^{N} s_n(k) \sum_{i=0}^{k-1} q_n(i) \times \left[1-\exp(-j\frac{2\pi\ell(i-k)}{L}\right]\} V_{n+1}^*(\ell), \quad \ell=0,1,\ldots,L-1 \tag{9}$$

When the steady-state solution is to be obtained from Eqn. (9),the steady-state probability density functions s(k) and v(k) are usually known,and the subscripts n on these functions in Eqn. (9) can be dropped.

The $M|G^Y|1$ queue with a constant service capacity was considered. The probability density function of the service capacity for this queue is $s(k)=\delta(k-m)$ where m is the service capacity of the system and $\delta(n)$ is the delta function. The frequency-domain recurrence relation Eqn. (9), for this system reduces to

$$Q^*_{n+1}(\ell) = \{Q^*_n(\ell)W_L^{-m\ell} + \sum_{i=0}^{m-1} q_n(i)\left[1-W_L^{(i-m)\ell}\right]\}V^*(\ell),$$

$$\ell=0,1,\ldots,L-1$$

$$n=0,1,\ldots \qquad (10)$$

where $W_L = \exp(-j\frac{2\pi}{L})$. To start the iteration by use of Eqn. (10), an idle queue, i.e., $q_o(k) = \delta(k)$ is used. The procedure for computing the steady-state queue length probability density function by use of the frequency-domain recurrence relation Eqn (10) is summarised in the steps below:

<u>Step 0</u>

0.1) Compute $V^*(\ell)$, the DFT of $v(k)$, using Eqn. (7).

0.2) Compute $Q_o^*(\ell)$, the DFT of $q_o(k)$. For $q_o(k)=\delta(k)$, $Q_o^*(\ell)=1$, for $\ell=0,1,\ldots,L-1$.

<u>Step n</u>, for n=1,2,3,...

n.1) Compute the inverse DFT of $Q_n^*(\ell)$, to obtain the elements $q_n(0)$, $q_n(1)$, ..., $q_n(m-1)$. This should be done using a non-in-place FFT algorithm so that the DFT $Q_n^*(\ell)$ is not lost.

n.2) Compute the DFT of $q_{n+1}(k)$, using Eqn. (10).

n.3) Obtain $q_{n+1}(0)$ by computing the inverse DFT at zero time only, i.e.

$$q_{n+1}(0) = \frac{1}{L} \sum_{\ell=0}^{L-1} Q^*_{n+1}(\ell).$$

n.4) Repeat from Step n.1) until $|q_{n+1}(0)-q_n(0)|<T$, where T is a small number to test for convergence.

n.5) Compute the inverse DFT of $Q^*_{n+1}(\ell)$, to obtain $q_{n+1}(k)$.

Care must be exercised in choosing the FFT block size L. The choice must be such that aliasing does not cause much error. Thus, one can begin the iterations with a block size of L. If the block size is not adequate at the end of the iterations, it is doubled and the iterations performed again. The process is repeated until a suitable block size is obtained.

In reference 62, Ackroyd computed the queue length distribution for the $M|G|1$ queue, iteratively. This computational procedure just outlined is an extension of his method to the case of the bulk service $M|G^Y|1$ queue with constant service capacity m, For m=1, the computational procedure outlined reduces to that in reference 62. The procedure can be extended to the $M|G^Y|1$ queue with a general service capacity distribution, by use of Eqn (9).

2.4 DIRECT COMPUTATIONAL METHOD

The direct method involves rearranging Eqn. (5) into a product of a maximum phase function (function with all its poles and zeros outside the unit circle) and a minimum phase function (all poles and zeros inside the unit circle). Digital signal processing methods[72-74] can then be used to extract the queue length probability distribution from the product. Root-finding methods for polynomials can also be used to obtain this distribution[75]. However, there may be problems with high-order polynomials.

The denominator of Eqn (5) can be shown to have N zeros outside or on the unit circle in the z-plane by use of Rouché's theorem. Because the function Q(z) converges inside the unit circle, these N zeros must cancel with the N zeros of the numerator which are outside or on the unit circle. The N zeros of the numerator of Eqn. (5) result from the polynomial function

$$H(z) = z^{-N} \sum_{k=1}^{N} s(k) \sum_{i=0}^{k-1} q(i)(1-z^{-i+k}) \tag{11}$$

This is a polynomial of order N and it has N zeros, of which (N-1) are outside the unit circle and one is at the point z=1. The function $H(z)/(z^{-1}-1)$ is thus a maximum phase function as all its zeros lie outside the unit circle.

The denominator of Eqn. (5) consists of a product of

H(z) and a function R(z) which has all its zeros inside the unit circle. Using Eqn (5) and (11), the function $(z^{-1}-1)\times Q(z)/H(z)$ is then

$$(z^{-1}-1)\,\frac{Q(z)}{H(z)} = \frac{(z^{-1}-1)}{H(z)\ R(z)} = \frac{(z^{-1}-1)V(z)}{z^{-N}\left[1-V(z)S(\frac{1}{z})\right]} \tag{12}$$

If V(z) has some zeros* outside the unit circle, then the function $(z^{-1}-1)Q(z)/H(z)V(z)$ is formed instead. If V(z) has no zeros outside the unit circle, then the function $(z^{-1}-1)Q(z)/H(z)$ is a product of a minimum phase function Q(z) and a maximum phase function $H(z)/(z^{-1}-1)$. As the z-transforms V(z) and S(z) are known, the right side of Eqn. (12) can be obtained. All the zeros of the denominator of the right side of Eqn. (12), inside the unit circle, are the zeros of the denominator of Q(z). Therefore Q(z) can be obtained by root-finding procedures for polynomials[(75)] or by the complex cepstrum[(72-74)].

A method for extracting a minimum phase function from a mixed phase function by use of the complex cepstrum is described in Reference 45. The complex cepstrum of a sequence x(k), is a sequence $\hat{x}(k)$, whose z-transform is the complex logarithm of the z-transform of x(k). It can be

---

* The z-transform of a discrete probability density function can have zeros inside or outside the unit circle in the z-plane. However, because it must converge inside the unit circle, all its poles must lie inside this circle.

shown that, if $x(k)$ is a minimum phase sequence, then $\hat{x}(k) = 0$, for $k<0$. If $\hat{x}(k)$ is a maximum phase function, then $\hat{x}(k)=0$, for $k>0$. Thus, for a mixed phase function, the complex cepstrum of the function corresponding to the minimum phase part will be zero for $k<0$, and that corresponding to the maximum phase part will be zero for $k>0$. To obtain the sequence $\hat{x}(k)$, the DFT $X^*(m)$ of $x(k)$ is computed. The complex logarithm of $X^*(m)$ is then obtained. The inverse DFT of the result gives the cepstrum sequence $\hat{x}(k)$.

The procedure for computing the probability density function of the queue length, from Eqn(12), by use of the complex cepstrum, is described below. The sequence corresponding to the right side of Eqn. (12) can be computed by digital filtering or by the use of the FFT algorithm. However, digital filtering is simulated in software as recursive relations. These relations involve convolutions resulting in large numbers of real multiplications and additions or subtractions. The numbers of multiplications and additions or subtractions are reduced by using the FFT algorithm.

To use the FFT algorithm to obtain the sequence whose z-transform corresponds to the right side of Eqn (12), z is replaced by $\exp(j\frac{2\pi m}{L})$ in this equation. The DFT of the z-transform function $P(z) = \frac{(z^{-1}-1)Q(z)}{H(z)}$ is

$$P\left[\exp\left(j\frac{2\pi m}{L}\right)\right]=\begin{cases}\dfrac{-1}{S'(1)+V'(1)}\,, & m=0\\[2ex] \gamma_m + j\theta_m\,, & m=1,2,\ldots,\frac{L}{2}-1\\[2ex] \dfrac{2V(-1)}{S(-1)V(-1)-1}\,, & m=\frac{L}{2}\end{cases} \tag{13}$$

where

$$\gamma_m = \frac{d_m f_m + c_m g_m}{f_m^2 + g_m^2}$$

$$\theta_m = \frac{c_m f_m - d_m g_m}{f_m^2 + g_m^2}$$

$$f_m = \alpha_m \sigma_m - \beta_m \phi_m - 1$$

$$g_m = \alpha_m \phi_m + \beta_m \sigma_m$$

$$d_m = \sigma_m\left[1-\cos\left(\frac{2\pi m}{L}\right)\right] - \phi_m \sin\left(\frac{2\pi m}{L}\right)$$

$$c_m = \sigma_m \sin\left(\frac{2\pi m}{L}\right) + \phi_m\left[1-\cos\left(\frac{2\pi m}{L}\right)\right]$$

$$\alpha_m = \mathrm{Re}\left\{S\left[\exp\left(-j\frac{2\pi m}{L}\right)\right]\right\}$$

$$\beta_m = \mathrm{Im}\left\{S\left[\exp\left(-j\frac{2\pi m}{L}\right)\right]\right\}$$

$$\sigma_m = \mathrm{Re}\left\{V\left[\exp\left(j\frac{2\pi m}{L}\right)\right]\right\}$$

$$\phi_m = \mathrm{Im}\{V[\exp(j\frac{2\pi m}{L})]\}$$

$$S'(1) = \left.\frac{dS(\frac{1}{z})}{dz^{-1}}\right|_{z^{-1}=1} = -\left.\frac{dS(z)}{dz^{-1}}\right|_{z^{-1}=1}$$

and

$$V'(1) = \left.\frac{dV(z)}{dz^{-1}}\right|_{z^{-1}=1}$$

The sequence whose z-transform is $S(\frac{1}{z})$ will be on the negative-time axis. Therefore the array used to store this sequence is divided into two halves. The lower half holds the positive-time part which will be zero,and the top half holds the negative-time part. The DFT of this sequence can then be computed. Alternatively, the DFT of the sequence with z-transform $S(z)$ is computed. The imaginary parts of the result are then negated,to obtain the DFT of the sequence with z-transform $S(\frac{1}{z})$. Because the sequence corresponding to the right side of Eqn (12) has both positive-and negative-time parts, the FFT block size should be twice the normal size. The choice of the block size must be such that the magnitude of the last element of the computed queue length density is negligible.

The procedure for computing the queue length probability distribution function for the $M|G^Y|1$ queue, from Eqn. (12), by use of the DFT Eqn (13) is summarised in the steps below.

1. From the probability density functions of service capacity $s(k)$ and number of arrivals in a service period $v(k)$, compute $S'(1)$, $S(-1)$, $V'(1)$, $V(-1)$.

2. Obtain the DFT's of sequences $s(k)$ and $v(k)$.

3. Compute the DFT of the sequence whose z-transform is the right side of Eqn (12) by use of Eqn. (13) and results from steps 1 and 2.

4. Obtain the complex logarithm of this DFT.

5. Compute the inverse DFT, to obtain the complex cepstrum of the sequence whose z-transform corresponds to the right side of Eqn (12).

6. Set the negative time part of the cepstrum to zero.

7. Compute the DFT of this modified cepstrum.

8. Obtain the exponential of this DFT.

9. Compute the inverse DFT and normalise, to get a total probability of unity, and hence $q(k)$, the queue length probability density function.

10. Obtain the cumulative distribution of the queue length by passing the density $q(k)$ through a digital filter with transfer function $\dfrac{1}{(1-z^{-1})}$.

The solutions for some queues were obtained by a root-finding procedure and details are given in section 2.6. A comparison of numerical results obtained from the cepstrum method and results obtained from these solutions is provided in section 2.7.

2.5 SOLUTIONS BY ROOT FINDING AND LINEAR EQUATIONS

The z-transform of the probability density function of the queue length Eqn. (5) consists of N unknowns. These unknowns can be computed by an auxiliary system of linear equations. The coefficients of the auxiliary system depend on the roots, outside the unit circle in the z-plane, of the denominator of Eqn (5). Thus to invert this equation, using a root-finding procedure and a system of linear equations, one proceeds as follows. The roots of the denominator outside or on the unit circle are computed. By substituting these roots into the numerator polynomial and equating to zero, a system of linear equations is obtained. These equations are then solved, to obtain the N unknowns, from which the remaining q(k)'s are obtained.

The roots of the denominator of Eqn (5) are obtained from the equation

$$z^{-N}V(z)\ S(\tfrac{1}{z}) - z^{-N} = 0 \tag{14}$$

Let the roots, outside or on the unit circle, be located at $z=\sigma_n$, for $n=1,2,\ldots,N$. One of these roots, σ_1 is at $z=1$. The system of linear equations is then obtained by substituting these roots into the numerator of Eqn (5) to obtain

$$\sum_{k=0}^{N-1} q(k) \sum_{i=k+1}^{N} s(i)(1-\sigma_n^{i-k}) = 0, \quad \text{for } n=2,3,\ldots,N$$

and

$$\sum_{k=0}^{N-1} q(k) \sum_{i=k+1}^{N} (i-k)s(i) = V'(1) - S'(1)$$

where

$$S'(1) = \left.\frac{dS(z)}{dz^{-1}}\right|_{z^{-1}=1}$$

The root at z=1 does not produce a linear equation. The Nth equation is obtained from the condition Q(1)=1, with the help of l'Hôpital's rule. The N unknowns, i.e., q(i), i=0,1,...N-1, are then computed by numerical methods for solving simultaneous linear equations. Most computer subroutine libraries have programs for solving linear equations[75]. The remaining q(i)'s are then obtained from Eqn. (5) by use of digital filtering or by use of the FFT algorithm.

2.6 SOLUTIONS FOR QUEUES $M|M^Y|1$ AND $M|E_2^Y|1$

Three methods for computing the probabiliy density function of the queue length for the $M|G^Y|1$ system have been outlined. To assess the performance of these numerical methods, specific service time and service capacity probability density functions have to be assumed in order to obtain the queue length distribution.

Fortunately, some functions V(z) and S(z) yield denominators, for Eqn (5), which are finite polynomials of orders greater than N. If this denominator(see Eqn (14)) is a polynomial of order say N_1,where $N_1>N$, then this denominator will have N_1-N zeros inside the unit circle of the z-plane. With appropriate normalisation, these zeros give Q(z). The N zeros of the denominator polynomial outside or on the unit circle then cancel with all the N zeros of the numerator polynomial Eqn(11). Therefore when this happens, there is no need of solving the system of linear equations. The problem is easily solved by computing the (N_1-N) zeros of the denominator which are inside the unit circle.

For the $M|G^Y|1$ system, the arrival process is Poisson[2]. Two service time distributions and three service capacity distributions were considered. The service time determines the number of arrivals in a service period. The Poisson sequence,v(k),of the service time density is the probability density function of the number of arrivals in a service period[2]. To obtain the Poisson transform of the service time probability density function b(t), the Laplace transform of b(t) is obtained. When s is replaced by $\lambda(1-z^{-1})$ in $B_L(s)$, the z-transform V(z), of the distribution of the number of arrivals v(k), is obtained. The Poisson transform can also be computed from b(t) by use of digital filters[66,78]. These methods are described in full in Chapter 4. For the exponential service time density $b(t)=\mu e^{-\mu t}$, the z-transform of the distribution of the number of arrivals in a service period is $V(z)=\frac{(1-\alpha)}{(1-\alpha z^{-1})}$, where $\alpha=\frac{\lambda}{(\mu+\lambda)}$ and λ and μ are the average arrival and service rates,respectively. For the Erlangian (E_2) service time density $b(t) = 2\mu(2\mu t)e^{-2\mu t}$, $V(z) = \left[\frac{\alpha}{(1+\alpha-z^{-1})}\right]^2$, where $\alpha=2\mu$.

The three service capacity probability density functions which were considered were: constant, binomial and geometric service capacities. The z-transform of the constant service capacity density is $S(z) = z^{-m}$, where m is the service capacity. For stability, the average service capacity per service period must exceed the average number of customers served per service period, that is $m > \frac{\lambda}{\mu}$. With the binomial service capacity, the probability of the service capacity being k is derived as follows. Let the probability that service is available to one customer at the beginning of a service period be p, and let the probability that it is not available be (1-p). Provided these are independent events, then the probability that service is available to k customers, at the beginning of a service period, is binomial[5, 20] and this is given by

$$s(k) = \binom{N}{k} p^k (1-p)^{N-k}, \qquad k=0,1,\ldots,N$$

where N is the maximum service capacity. The z-transform of s(k) is thus

$$S(z) = (1-p)^N \left[1+ \frac{p}{1-p} z^{-1}\right]^N \tag{16}$$

The utilization factor for the $M|G^Y|1$ queue with binomial service capacity distribution and a service rate μ is

$$\rho = \frac{\lambda}{Np\mu} = \frac{\lambda}{\mu\bar{s}}$$

where Np is the average service capacity $\bar{s}$. For the geometric service capacity distribution, the probability that the service capacity is k is $(1-\gamma)\gamma^k$ and its z-transform is $\frac{(1-\gamma)}{(1-\gamma z^{-1})}$.

These arrival and service capacity distributions result in denominator polynomials of finite degree for Eqn (12). Each denominator polynomial will have, say k, zeros inside the unit circle. Q(z), after normalization, is of the form

$$Q(z)=\frac{\prod_{i=1}^{k}(1-\sigma_i)}{\prod_{i=1}^{k}(1-\sigma_i z^{-1})} \tag{17}$$

where σ_i, i=1, ..., k are the zeros inside the unit circle The denominator of Eqn. (12), for the solution of the $M|M^Y|1$ queue with constant service capacity m, is

$$1+z^{-1}+z^{-2}+\ \dots\ +z^{-m+2}+z^{-m+1}-\frac{\alpha}{1-\alpha}\ z^{-m} \tag{18}$$

where $\alpha = \frac{\lambda}{(\lambda+\mu)}$. Only one zero of this polynomial is inside the unit circle. Using Eqn. (17), q(k), the queue length density is given by $(1-\sigma)\sigma^k$ where σ is the only zero inside the unit circle. The solution for the $M|E_2^Y|1$ queue with constant service capacity m results in the denominator polynomial

$$1+z^{-1}+z^{-2}+\dots+z^{-m+1}-\frac{1+2\alpha}{\alpha^2}\ z^{-m}-\frac{z^{-m-1}}{\alpha^2} \tag{19}$$

where $\alpha = 2\mu$. Two of the zeros of this polynomial lie inside the unit circle. If these roots are located at σ_1 and σ_2, then by solving Eqn (17), q(k) is given by

$$q(k)=\frac{(1-\sigma_1)(1-\sigma_2)}{(\sigma_1-\sigma_2)}\left[\sigma_1^{k+1}-\sigma_2^{k+1}\right],\quad k=0,1,\dots \tag{20}$$

The denominator polynomial for the solution of the $M|E_2^Y|1$ queue with a binomial service capacity is

$$-\frac{z^{-N-2}}{\alpha^2}+\frac{2(1+\alpha)}{\alpha^2}z^{-N-1}-\frac{1}{\alpha^2}[(1+\alpha)^2-(1-p)^N\alpha^2]z^{-N}+$$

$$(1-p)^N\sum_{n=1}^{N}\varphi_n\left(\frac{p}{1-p}\right)^n z^{-N+n} \qquad (21)$$

where $\alpha=2\mu$, and the coefficients φ_n are obtained recursively from

$$\varphi_{n+1}=\frac{N-n}{n+1}\varphi_n\ ,\quad n=0,1,\ldots,N$$

$$\varphi_o=1$$

This polynomial of order N+2 has a total of N+2 zeros in the entire z-plane. Two of the zeros lie inside the unit circle and these are the zeros of the denominator of Q(z). If these two zeros are located at $z=\sigma_1$ and σ_2, then q(k), the queue length density is obtained from Eqn. (20). For the $M|E_2^Y|1$ queue with a geometric service capacity density, $S(z)=\frac{(1-\gamma)}{(1-\gamma z^{-1})}$ and $V(z)=\left[\frac{\alpha}{(1+\alpha-z^{-1})}\right]^2$. The two zeros of the denominator of Eqn. (12), which lie inside the unit circle, are located at

$$z=\sigma_1,\sigma_2=\frac{1+2\alpha+\gamma\pm(1-\gamma)\sqrt{1+\frac{4\alpha(1+\alpha)}{1-\gamma}}}{2\gamma(1+\alpha)^2}$$

q(k), the queue length probability density function, is obtained from Eqn. (20). For stability, $\gamma \geq \frac{1}{(1+\mu)}$ for $\lambda=1$.

The foregoing results enable the probability density function of the queue length to be computed, to provide a comparison with the iterative computational scheme and the cepstrum method. A comparison of these methods is made in the next section.

2.7 ACCURACY AND LIMITATIONS OF COMPUTATIONAL METHODS

To access the frequency-domain iterative method, the queue length probability density functions for the $M|M^Y|1$ and $M|E_2^Y|1$ queues with constant service capacities were computed. For comparison, the queue length densities were also computed from the roots of Q(z). The roots of the denominator polynomial of Eqn. (12) were computed using the NAG library subroutine CØ2AEF Mark 5[75] on the Harris 500 computer. Tables 1 and 2 show the location of the zeros for constant service capacities m=2, 15, 30 and 50 and utilization factors $\rho \simeq 0.1$ and 0.9. Tables 3 and 4 compare the number of digits in agreement for the q(k) computed by the iterative method and that computed from the roots of the denominator of Eqn. (12). The accuracy of the roots computed by the subroutine CØ2AEF is good for well-behaved polynomials. From experience, the denominator polynomials for these examples are well behaved. Therefore the zeros, and hence the queue length density, were computed to good accuracy.

Tables 3 and 4 show how the number of digits in agreement for the computed results depended on the utilization factor, the FFT block size L, the number of iterations performed and the service capacity m. The convergence criterion for the iterations was $|q_{n+1}(o) - q_n(o)| \leqslant 10^{-7}$.

From these tables, the following conclusions were drawn about the iterative method. For small utilization

TABLE 1

Root of Denominator of Q(z) for $M|M^Y|1$

Queue with a Constant Service Capacity m

Utilization Factor $\rho = \frac{1}{\mu}$	Service Capacity m	α	Reciprocal of Root σ
$\simeq 0.9$	2	0.642857	1.0732127318
	15	0.9310345	1.0134686809
	30	0.964286	1.0069362908
	50	0.978261	1.0042125217
$\simeq 0.1$	2	0.1666667	5.8541007402
	15	0.6	1.6663523217
	30	0.75	1.333273726
	50	0.833334	1.1999770422

For $\rho \simeq 0.9$, $\alpha = \frac{9m}{10+9m}$

For $\rho \simeq 0.1$, $\alpha = \frac{m}{10+m}$

TABLE 2

Roots of Denominator of Q(z) for $M|E_2^Y|1$

Queue with a Constant Service Capacity m

Utilization Factor $\rho = \frac{1}{\mu}$	Service Capacity m	α	Reciprocal of the root σ_1	Reciprocal of the root σ_2
$\simeq$ 0.9	2	1.11111111	2.547302398	1.11111111111
	15	0.14814814815	1.1886735059	1.0254512433
	30	0.0740740747	1.0934648024	1.0134687156
	50	0.04444444	1.0558633534	1.0082744608
$\simeq$ 0.1	2	10	11.844288771	9.9999999994
	15	1.3333333	2.3356342662	2.3309978535
	30	0.666666666	1.6669797298	1.6663518350
	50	0.4	1.4000907889	1.3999089289

For $\rho \simeq 0.9$, $\alpha = \frac{20}{9m}$

$\rho \simeq 0.1$, $\alpha = \frac{20}{m}$

TABLE 3

Comparison of Results from Iterative Method and Root-Finding Method for the Queue $M|M^Y|1$ With a Constant Service Capacity m

Utilization Factor ρ	Service Capacity m	FFT Block Size L	Number of Iterations N	Number of Digits in Agreement
$\simeq 0.1$	2	64	7	7
		128	7	7
	15	64	4	9
		128	4	9
	30	64	3	8
		128	3	9
		256	3	9
	50	128	3	9
		256	3	9
$\simeq 0.9$	2	1024	1033	4
	15	1024	556	4
	30	1024	408	3
	50	1024	230	3
		2048	393	3

TABLE 4

Comparison of Results from Iterative Method and Root-Finding Method for $M|E_2^Y|1$ Queue With a Constant Service Capacity m

Utilization Factor ρ	Service Capacity m	FFT Block Size L	Number of Iterations N	Number of Digits in Agreement
≃0.1	2	64	6	8
		128	6	8
	15	64	3	10
		128	3	10
		256	3	10
	30	64	3	7
		256	3	7
		512	3	7
	50	64	2	8
		128	2	9
		256	2	9
≃0.9	2	512	733	4
		1024	733	4
	15	512	415	4
		1024	417	4
		2048	417	5
		4096	417	4
	30	1024	312	4
		4096	313	4
	50	4096	248	5

factors, small block sizes are adequate for a given service capacity, and the number of iterations performed before convergence are relatively fewer than for high utilization factors. This follows from the fact that relatively small queues are formed and these are cleared quickly by the server, and hence steady state is rapidly achieved. For systems with large service capacities under a constant utilization factor, fewer iterations need to be performed, compared to systems with small service capacities. A server with a large service capacity clears a queue much faster than one with a small service capacity. Systems with high utilization factors require large FFT block sizes as large queues are formed. Furthermore, more iterations need to be performed to satisfy a given convergence criterion, compared to systems with small utilization factors. Provided the block size is sufficiently large to minimise errors due to folding, greater accuracy can be achieved by performing more iterations. For example, the queue length density for the $M|E_2^Y|1$ queue, with a constant service capacity m=2, was computed for a utilization factor ρ=0.9, using a block size of 2048. About 2500 iterations resulted in an agreement of the computed results of about 6 digits.

To summarise, the agreement of the computed results was at least 7 digits for queues with small utilization factors. A block size of 64 was adequate and in all cases, the number of iterations that were performed to achieve this accuracy were less than 10. For queues with utilization factor $\rho=0.9$, large block sizes (e.g. 1024, 2048, 4096) were required. The numbers of iterations resulting in accuracies of about 4 digits were up to several hundreds for all the service capacities m considered.

The direct method was used in the computation of the queue length distribution for the $M|M^Y|1$ and $M|E_2^Y|1$ queues with constant service capacities. Tables 5 and 6 show how the accuracy of the computed results depend on the utilization factor ρ, the service capacity m and the FFT block size L. As observed for the iterative method, for small utilization factors, small block sizes are adequate. The accuracy is higher at small utilization factors and small service capacities.

A comparison of Tables 3-6 shows that in general, for small utilization factors, the frequency iterative method requires smaller block sizes than the direct method. As mentioned before, the direct method requires half the block size for the positive-time samples and the other half for the negative-time samples, whereas the iterative method just requires the block size for the positive time samples. The iterative method is computationally expensive because, for high queue utilizations, several hundreds of iterations are performed to achieve the same accuracy as that obtained by use of the direct method.

TABLE 5

Comparison of Results from Direct Computational Method and Root-Method for $M|M^Y|1$ Queue with a Constant Service Capacity m

Utilization Factor ρ	Service Capacity m	Block Size L	Number of Digits in Agreement
$\simeq 0.1$	2	256	9
		512	8
		1024	8
	15	512	4
		1024	6
		2048	8
	30	1024	4
		2048	6
	50	4096	7
$\simeq 0.9$	2	256	5
		512	8
		1024	8
	15	512	3
		1024	5
	30	2048	5
	50	2048	4
		4096	7

TABLE 6

Comparison of Results from Direct Computational Method and Root-Method for $M|E_2^Y|1$ Queue With a Constant Service Capacity m

Utilization Factor ρ	Service Capacity m	Block Size L	Number of Digits in Agreement
$\simeq 0.9$	2	256	6
		512	6
		2048	5
	15	512	5
		1024	5
		2048	4
	30	512	3
		1024	4
		2048	4
		4096	3
	50	1024	5
		2048	4
		4096	3
$\simeq 0.1$	2	64	5
		128	8
		256	8
	15	1024	6
		2048	6
	30	2048	6
		4096	5
	50	2048	4
		4096	5
		8192	5

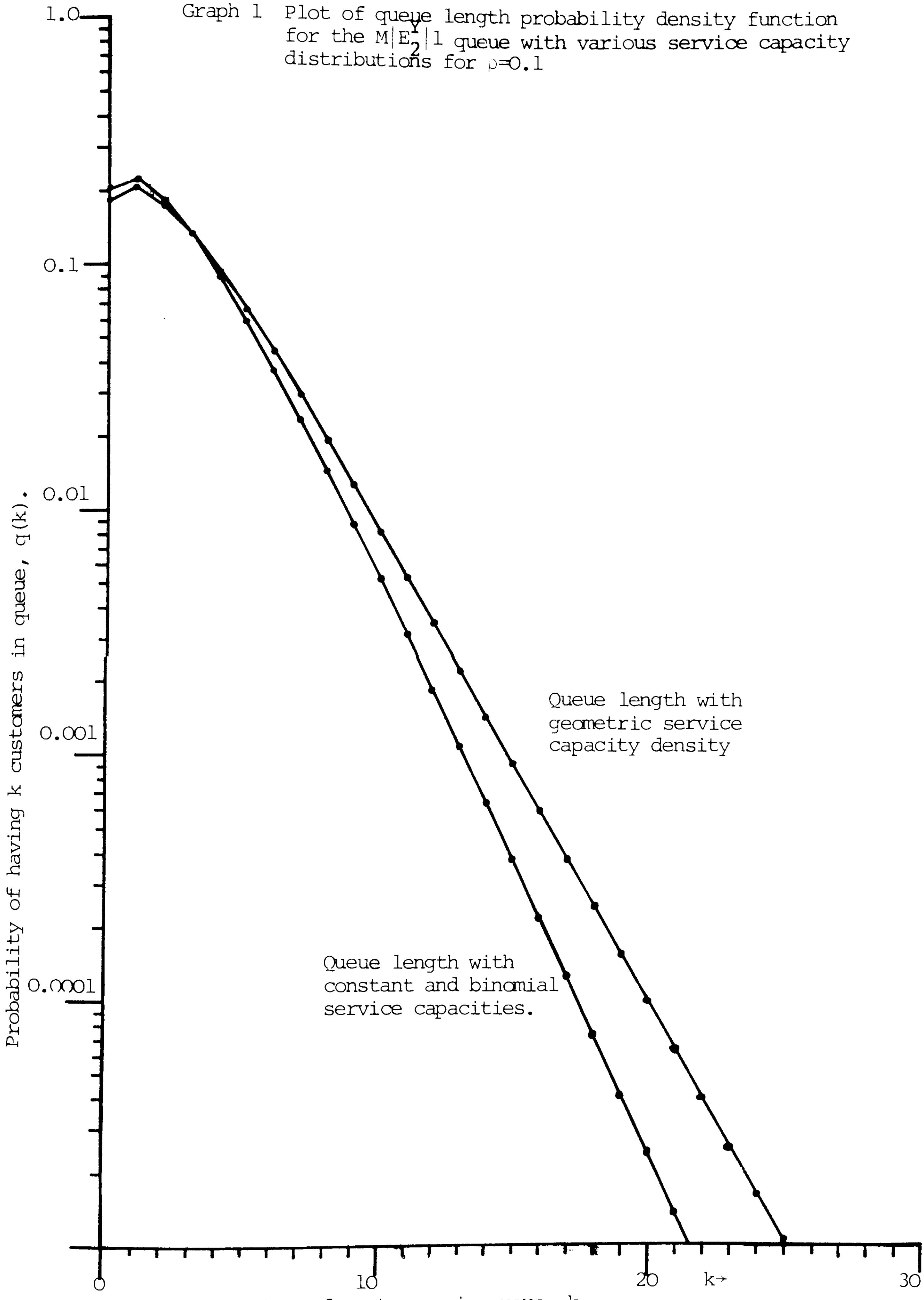
The queue length probability density functions for the $M|E_2^Y|1$ queue under different service capacity densities were computed by use of the direct method. Using the computed results and Eqn. (6), the mean queue length was also computed. The computed results were compared with the q(k) obtained by use of the zeros of the denominator of Eqn (12). The mean service capacity for the different service capacity densities was set at $\bar{s}=24$. Table 7 shows how the accuracy of the computed results vary with the utilization factor ρ, the service capacity density and the FFT block size L. The accuracy of the computed mean queue length Eqn (6) is also shown. For the $M|E_2^Y|1$ queue with constant, binomial and geometric service capacity densities, the computed queue lengths were accurate to at least 6 digits and the computed means were accurate to about 5 digits. Graph 1 shows a plot of the queue length densities for these systems for a utilization factor $\rho=0.1$.

One point to note is that the standard numerical method for solving Eqn (5) by use of root-finding and solution of linear equations in section 2.5 will not necessarily produce more accurate results than the two methods presented. Although the root-finding procedure can produce results whose accuracy is comparable to that of the computer used, for well behaved polynomials, this procedure can present problems for high-order polynomials which may be less-well behaved. These problems can reduce the precision of the computed roots. There are several

TABLE 7

A Comparison of Results from the Direct Computational Method and the Root-Finding Method for the $M|E_2^Y|1$ queue with Different Service Capacity Distributions

Utilization Factor ρ	Service Capacity Density	FFT Block Size L	Accuracy of Computed q(k)	Accuracy of Computed Mean $\bar{q}$
0.1	Constant m=24	2048	6	4
	Binomial p=0.5 N=48	2048	7	5
	Geometric γ=0.96	2048	7	5
0.9	Constant m=24	4096	7	5
	Binomial p=0.5 N=48	4096	7	4
	Geometric γ=0.96	4096	8	5



Graph 1 Plot of queue length probability density function for the $M|E_2^Y|1$ queue with various service capacity distributions for $\rho=0.1$

problems associated with the solution of linear equations. If the magnitudes of the coefficients of the linear equations vary over a wide range, there is loss of precision in computing the solution to these equations. Secondly, the computation time associated with the solution of these simultaneous linear equations using say the subroutine FØ4ADF[75] is roughly proportional to N^3 where N is the maximum service capacity.

2.8 CONCLUSION AND APPLICATION TO COMPUTER COMMUNICATIONS

The transient- and steady-state z-transforms of the queue length probability density function for the $M|G^Y|1$ queue were derived. Two numerical procedures for the inversion of the derived transforms were presented. These methods made use of digital signal processing techniques. The standard numerical methods for inverting transforms with multiple-state dependencies was outlined. For the queues $M|M^Y|1$ and $M|E_2^Y|1$ under constant, binomial and geometric service capacities, the standard numerical solution reduced to just a root-finding procedure. The queue length probability density functions for these systems were computed using the three procedures. The agreement of the computed results was good for all the systems considered, for utilization factors $\rho=0.1$ and 0.9. The two methods which use digital signal processing techniques can also be used in the analysis of multiserver systems[10,16,18,20].

Bulk service and multiserver queues are systems where customers (data messages, speech sources, batch jobs, etc.) are served in bulk by a server or by several servers. These types of queues arise in many aspects of computer communication systems. They arise in the asynchronous multiplexing of low-speed buffered data terminals in time-sharing computer communication systems[15,16], in store-and-forward computer networks with integrated digital voice-data systems [18-20], and in stored-program-control (SPC) telecommunication systems. In a time-sharing computer system, slow terminals input data into a buffer. The data messages are then asynchronously multiplexed and transmitted to a central processing unit (CPU) at a faster speed. If each time frame for the multiplexer has m time slots, and there are $\tilde{q}$ data messages in the buffer, at the beginning of the frame, min $(m,\tilde{q})$ data messages are transmitted to the CPU during the frame. For a large number of terminals, the arrival process to the buffer will be close to Poisson [15]. This system then corresponds to the bulk service $M|D^Y|1$ queue with a constant service capacity m, or the multiserver $M|D|m$ queue.

In integrated digital voice-data systems and stored-program-control systems, the transmission facility is shared by real-time speech sources (or calls) and non-real time data messages which are buffered. The unbuffered speech sources have priority over the data messages to prevent degradation of the speech quality. The speech is synchronously multiplexed onto the channel while the data is asynchronously injected into the idle time slots which arise during speech silences. The idea of integrating speech and data stems from the fact that speech is only active for about forty percent of the conversation time [76]. Transmitting the speech alone under-utilizes the channel capacity by about sixty percent. Thus by synchronously multiplexing the speech sources and asynchronously injecting data messages into the idle time slots, the channel capacity utilization is greatly increased. This scheme is an important part of store-and-forward computer communication and telecommunication networks[18-21]. In this scheme, the service capacity (number of data messages injected into the idle slots) of the server (the transmission facility) is not fixed, but depends on the number of idle slots. The buffering process of the data messages is thus modelled as a bulk service or multiserver queue.

CHAPTER 3

BILINEAR AND POISSON TRANSFORMS IN QUEUE ANALYSIS

3.1 INTRODUCTION

Frequently, probability distribution functions used in performance evaluation and design of computer communication systems are derived in the form of Laplace transform expressions[2,3,23,36,60]. Unfortunately, the inversion of these transforms is often difficult even for simple systems. One thus resorts to the numerical inversion of these transforms. A problem encountered in doing this is the handling of continuous-time functions. For the purpose of numerical computations, only a finite number of samples can be used, whereas the continuous-time function consists of an uncountably infinite number of samples. Periodic samples of the function can be used for bandlimited functions. However, most probability functions in queue analysis are not bandlimited and in some cases the Laplace transforms of the functions can not be replaced by the z-transforms of their periodic samples. Furthermore, if periodic samples are used, a question to answer is how does one take into account the missing data between the samples? Some numerical methods on the inversion of the Laplace transform[53-59] have been developed.

The Poisson transform[77-80] is useful in the analysis of the M|G|1 queue[63,65,66]. Section 3.2 briefly discusses the uses of the Poisson transform in the M|G|1 queue analysis. The Pollaczek-Khinchin transform equation for the number of customers in the queue is inverted by use of digital filtering. Section 3.3 reviews the theory of representing continuous-time functions, for the purpose of digital filtering, by discrete sequences. The bilinear[81] and Poisson transforms are presented in section 3.4 as suitable discrete function representations. An approximate Laplace transform inversion method is outlined in section 3.5. The method makes use of the bilinear transformation. This method is applied to the inversion of the Laplace transform of the waiting time distribution for the M|G|1 queue.

In section 3.6, a method for the inversion of the Laplace transform by use of the bilinear and Poisson transforms is presented. In the method, analogue filtering is replaced by digital filtering. The method is then used in the computation of the waiting time distribution, for the G|G|1 queue, from its Laplace transform. The chapter concludes with a discussion of the application of these methods to the analysis of computer communication systems.

3.2 POISSON TRANSFORM IN M|G|1 QUEUE ANALYSIS

The M|G|1 system is a queue with an exponential interarrival time probability density function, general service time and a single server. The Poisson transform and its inverse have an application in the analysis of this queue (63,65,66). This transform and its inverse, inter-relates the various distribution functions for this queue. Fig. 1 illustrates the inter-relations between these distributions. The following probability density functions can be derived from the service time (2): the number of arrivals in a service period $v(k)$, the number in the queue $q(k)$, the waiting time $w(t)$, the response time (the total time spent in the system) $r(t)$, the number served in the busy period $f(k)$, and the busy period $g(t)$, as shown in Fig. 1.

The analysis of the M|G|1 queue requires the ability to :

(a) compute the Poisson transform of a function (Chapter 4),

(b) invert the Pollaczek-Khinchin transform equation (2),

(c) compute the inverse Poisson transform (Chapter 4).

The queue length distribution function is obtained by inverting the Pollaczek-Khinchin transform equation.

The Pollaczek-Khinchin transform equation for the

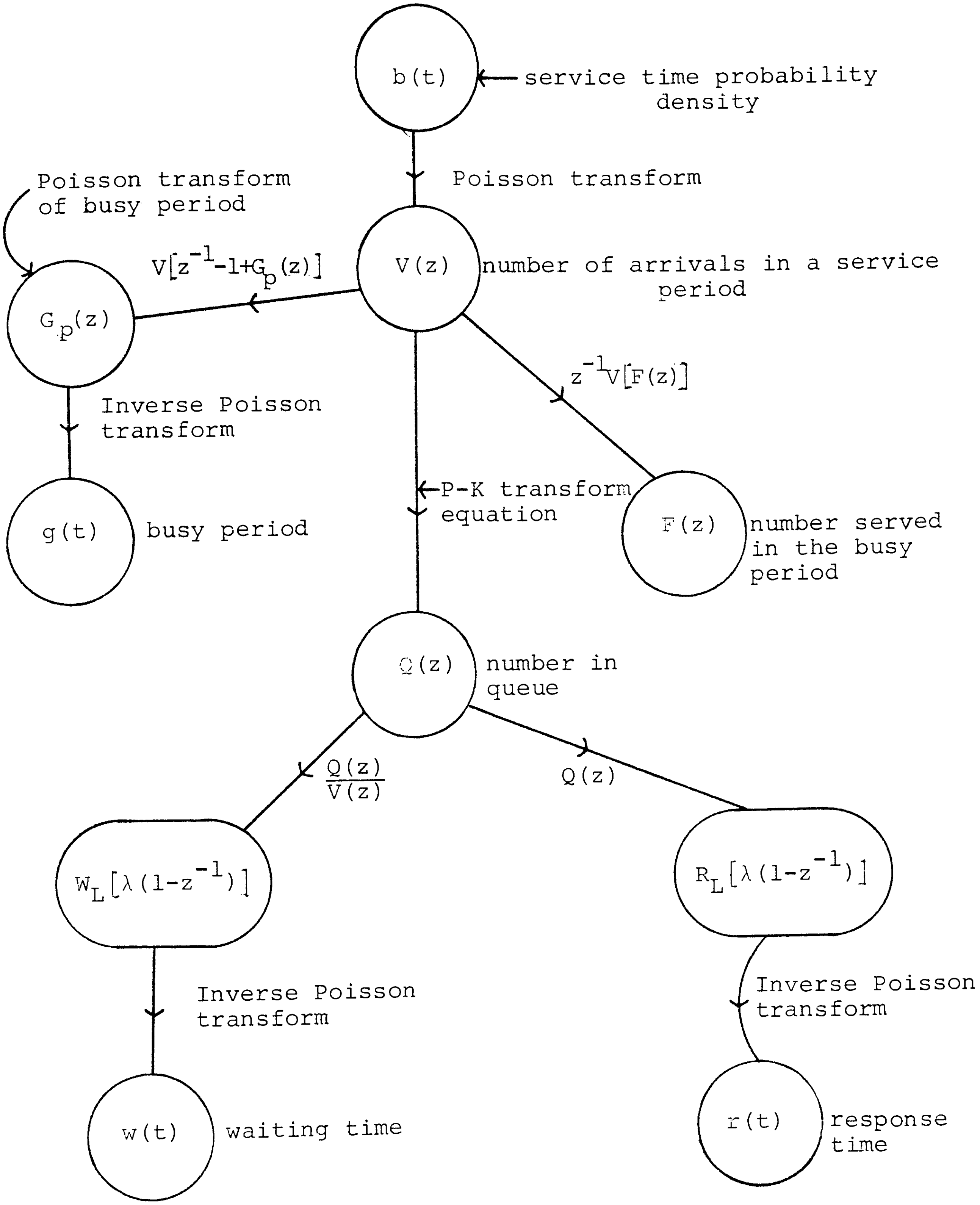


Fig.1 Diagram showing how the Poisson transform and its inverse interrelates the various probability density functions for the M|G|1 queue

number in the $M|G|1$ queue is given by[2]

$$Q(z) = \frac{(1-\rho)(1-z^{-1})\ V(z)}{V(z) - z^{-1}} \tag{1}$$

where $V(z)$ is the z-transform of the sequence $v(k)$ which is the Poisson sequence of the service time density, $b(t)$. ρ is the utilization factor. Eqn (1) can be inverted to obtain $q(k)$, the probability of having k customers in the queue, by digital filtering. The basic components of a digital filter are described in Appendix 1. The transfer function of a digital filter which can be used to invert Eqn (1) is

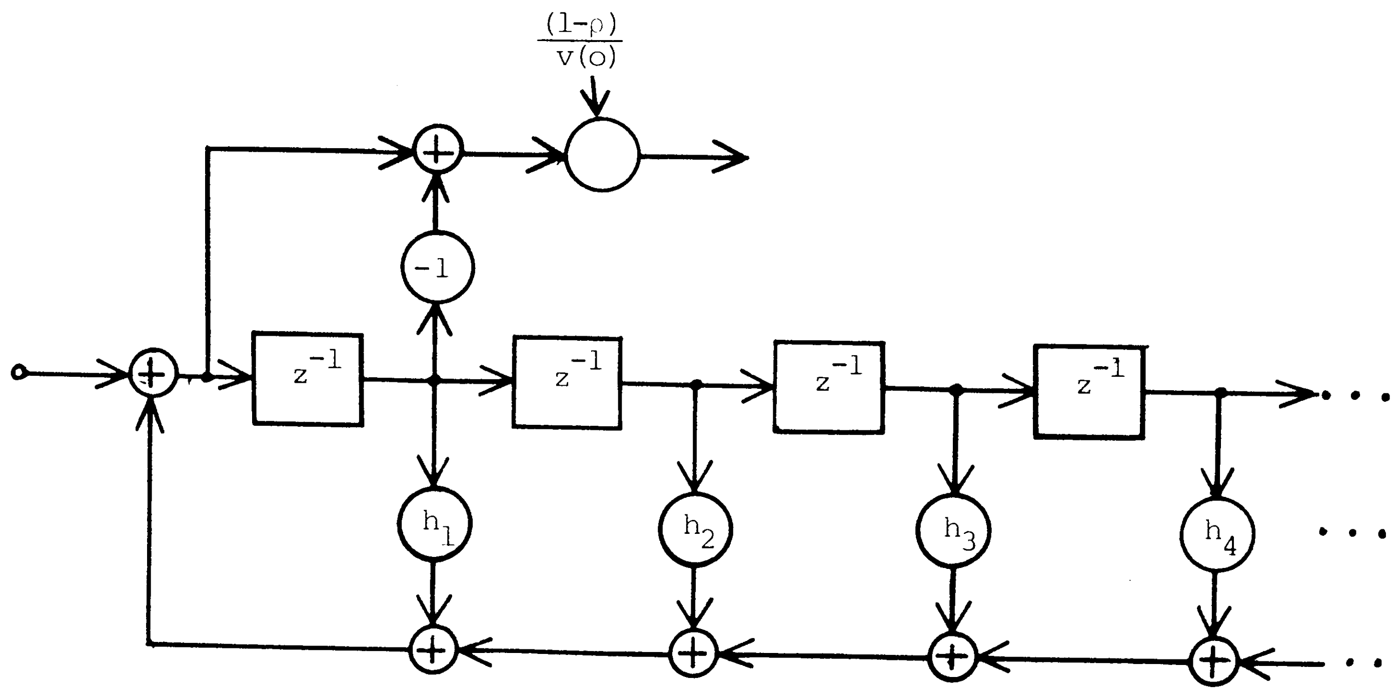
$$H(z) = \left(\frac{1-\rho}{v(o)}\right) \frac{(1-z^{-1})}{1+h_1 z^{-1}+h_2 z^{-2}+h_3 z^{-3}+\ldots} \tag{2}$$

where

$$h_k = \begin{cases} -\left(\frac{v(1)-1}{v(o)}\right) , & \text{for } k = 1 \\ -\frac{v(k)}{v(o)} , & \text{for } k > 1 \end{cases}$$

such that $Q(z) = H(z)V(z)$. By passing the sequence $v(k)$ through a digital filter with transfer function $H(z)$ Eqn (2), the sequence $q(k)$, the probability of k customers in the queue, is obtained at the output of this digital filter. The recursive digital filter with transfer function $H(z)$ is shown in Fig. 2. Alternatively, Eqn (1)

Fig. 2 Digital filter for generating q(k), the probability of having k customers in the M|G|1 queue. Input to digital filter is v(k), the probability of k arrivals in a service period, and output is q(k).



can be inverted by use of the FFT algorithm or recursive equations as in reference 64.

Using the digital filter method, and the sequences v(k) given in reference 2, the queue length probability density functions for the queues $M|M|1$, $M|E_2|1$ and $M|H_2|1$ (with service time density $b(t)=\frac{1}{4}e^{-t}+\frac{3}{2}e^{-2t}$) were computed for utilization factors $\rho=0.1$ and $\rho=0.9$ and $\lambda=1$. For the $M|M|1$ queue, the exact queue length density is $(1-\rho)\rho^k$, for $k=0,1,2,\ldots$. For the $M|E_2|1$ queue, q(k) is given by

$$q(k) = \frac{4\mu(\mu-\lambda)}{\sqrt{\lambda+8\mu}} (\sigma_1^{k+1} - \sigma_2^{k+1}), \quad k=0,1,2,\ldots$$

where σ_1 and σ_2 are obtained from the equations

$$\sigma_1 = \frac{\lambda+4\mu+\sqrt{\lambda+8\mu}}{8\mu^2} \quad \text{and} \quad \sigma_2 = \frac{\lambda+4\mu-\sqrt{\lambda+8\mu}}{8\mu^2}$$

For the $M|H_2|1$ queue considered, with $\lambda=1$, and $\rho=\frac{5}{8}$, the sequence q(k), is

$$q(k) = \frac{3}{32}\left(\frac{2}{5}\right)^k + \frac{9}{32}\left(\frac{2}{3}\right)^k, \qquad k=0,1,2,\ldots$$

On the ICL 1904 computer, the accuracy of the computed results compared to the exact solutions, using double precision arithmetic was at least 17 digits for the queues considered. The double precision arithmetic on the

ICL 1904 computer has an accuracy of about 20 digits[82].

In this present section, the sequences v(k) which are the Poisson sequences of the service time densities b(t), were assumed to be known. In fact as shown in Fig. 1, the Poisson transform and its inverse interrelates the various probability distributions, in the analysis of the M|G|1 queue. Equations giving the Poisson sequence $x_p(n)$ in terms of the inverse Poisson transform x(t), and vice-versa are given in section 3.4. Methods for computing the Poisson sequence $x_p(n)$ from x(t) and vice-versa are presented in the next chapter. These methods will enable the sequence v(k) to be obtained from b(t). In addition, the waiting time density w(t) and the response time density r(t) can then be obtained from the queue length density q(k).

3.3 DISCRETE REPRESENTATION OF CONTINUOUS-TIME PROBABILITY FUNCTIONS

A continuous-time function $x(t)$, can be represented as a series expansion of the form

$$x(t) = \sum_{n=0}^{\infty} x_n h_n(t) \quad , \qquad t \geq 0 \tag{3}$$

where x_n are the coefficients of the series expansion. $h_n(t)$, $n=0,1,\ldots$ are a set of functions which can be polynomials which are used in approximation theory(83), complex exponentials or sine and cosine functions as in the case of the Fourier transform(72).

The coefficients x_n are unique for each function $x(t)$ and the functions $h_n(t)$ chosen. It is thus possible to use the coefficients x_n in place of $x(t)$,in numerical calculations,by choosing certain functions $h_n(t)$. If these coefficients x_n become insignificant in magnitude for large n, then a finite number of these elements can be used to represent the function $x(t)$.

To use digital signal processing methods and these discrete sequences in the solution of queueing problems, it is desirable to select the functions $h_n(t)$ which enable certain conditions to be satisfied(81). The first condition is that these functions must permit a linear time-invariant continuous-time system to be mapped to a linear shift-invariant discrete system. A second condition is that the mapping from the continuous function to the discrete function must preserve convolution. This means that if a function x(t) is obtained by convolving y(t) with w(t), and if x_n, y_n and w_n are the discrete representations of these functions, respectively, then x_n must be obtained by the discrete convolution of y_n and w_n.

Oppenheim and Johnson[81] have shown that the second condition corresponds to the mapping of the Laplace transform of the continous-time function to the z-transform of its discrete representation by a substitution of variables. If $X_L(s)$ and $H_n(s)$ are the Laplace transforms of x(t) and $h_n(t)$, respectively, then using Eqn. (3), $X_L(s)$ is

$$X_L(s) = \sum_{n=o}^{\infty} x_n \; H_n(s) \tag{4}$$

By definition, the z-transform $X_d(z)$ of the discrete sequence x_n is

$$X_d(z) = \sum_{n=0}^{\infty} x_n z^{-n} \tag{5}$$

The requirement that the mapping of the Laplace transform $X_L(s)$, to the z-transform $X_d(z)$, be achieved by a substitution of variables, implies that $H_n(s)$ must be of the form $H_n(s) = [H_1(s)]^n$ such that

$$z = [H_1(s)]^{-1} \triangleq M(s) \tag{6}$$

A third condition is that a stable system in the s-plane must map into a stable system in the z-plane. In other words, the transformation Eqn (6) must be such that the left half of the s-plane maps into the interior of the unit circle in the z-plane.

For bandlimited functions, periodic sampling meets all these conditions. The discrete representation of the function $x(t)$ is a sequence x_n, consisting of equally spaced samples of the function, such that $x_n = Tx(nT)$, where T is the sampling period. Periodic sampling can then be viewed as an expansion of the continuous-time function $x(t)$ in the form of Eqn (3) with the functions $\{h_n(t), n=0,1,...\}$ given by

$$h_n(t) = \frac{\frac{1}{T}\sin\left[\frac{\pi}{T}(t-nT)\right]}{\frac{\pi}{T}(t-nT)} = \frac{1}{T}\operatorname{sinc}\left[\frac{1}{T}(t-nT)\right], \quad n=0,1,.. \qquad (7)$$

The Laplace transforms of these functions converge only on the $j\omega$-axis. On this axis, their transforms are[81,83]

$$H_n(j\omega) = \begin{cases} \left[e^{j\omega T}\right]^{-n} & , \quad |\omega T| \leqslant \pi \\ 0 & , \quad |\omega T| > \pi \end{cases} \qquad n=0,1,... \qquad (8)$$

The advantage of a discrete representation based on periodic sampling is that the coefficients x_n, in the expansion Eqn (3), are easily obtained. The major disadvantage is that it requires the function x(t) to be bandlimited. Unfortunately, most probability density functions in queue analysis are not bandlimited. Another disadvantage is that the underlying processes by which the Laplace transforms of probability density functions are derived, do not always permit the Laplace transforms of these functions to be replaced by the z-transforms of their sampled versions. It is thus desirable to obtain other discrete function representations which are suitable for numerical calculations in queue analysis.

3.4 BILINEAR AND POISSON TRANSFORMS OF FUNCTIONS

Two discrete function representations which meet the conditions required for digital filtering are the bilinear transformation[81] and the Poisson transform[77-80]. The bilinear transform corresponds to a mapping from the z-plane to the s-plane by the transformation

$$z = \left[\frac{\alpha+s}{\alpha-s}\right] \tag{9}$$

where α is a real parameter. This parameter will be referred to as the bilinear parameter. This transformation maps the $j\omega$-axis in the s-plane onto the unit circle in the z-plane and the left half of the s-plane into the interior of the unit circle as shown in Fig. 3(a).

The bilinear sequence of a function $x(t)$ which is zero for $t<0$ is defined as[81]

$$x_b(n) = \begin{cases} \int_0^\infty x(t)h_n(t)dt, & n=1,2,3,\ldots \\ -\sum_{m=1}^{\infty} (-1)^m x_b(m), & n=0 \end{cases} \tag{10}$$

The functions $h_n(t)$ are given by the equation

$$h_n(t) = \frac{t}{n} \phi_n(t) \tag{11}$$

where

$$\phi_n(t) = \begin{cases} \delta(t), & n=0,\ t\geq 0 \\ 2\alpha(-1)^n e^{-\alpha t} L_n^{(1)}(2\alpha t) + (-1)^n \delta(t), & n\geq 1,\ t\geq 0 \end{cases}$$

$$L_n^{(1)}(y) = \frac{dL_n(y)}{dy}, \qquad n=0,1,2,\ldots \qquad (12)$$

$$L_n(y) = \sum_{k=0}^{n} \binom{n}{k} \frac{(-y)^k}{k!}, \qquad n=0,1,2,\ldots \qquad (13)$$

$L_n(y)$ are the Laguerre polynomial functions, and $L_n^{(1)}(y)$ are their first derivatives. $x(t)$, the inverse bilinear transform of $x_b(n)$, is obtained from the expansion Eqn. (3), with $x_n = x_b(n)$ and the functions $h_n(t)$ given by Eqn (11). The inverse bilinear transform of the sequence $x_b(n)$ is[81]

$$x(t) = \begin{cases} 2\alpha \sum_{n=1}^{\infty} (-1)^n n\ x_b(n), & t=0 \\ 2\alpha e^{-\alpha t} \sum_{n=1}^{\infty} (-1)^n x_b(n) L_n^{(1)}(2\alpha t), & t>0 \end{cases} \qquad (14)$$

where the functions $L_n^{(1)}(y)$ are given in Eqn. (12).

A second discrete function representation meeting the conditions necessary for digital filtering is the

Poisson transform. The function M(s) from Eqn.(6) resulting from this transformation is $z=\frac{\gamma}{(\gamma-s)}$ [81], where γ is a real parameter which will be referred to as the Poisson parameter. It is straightforward to verify that the $j\omega$-axis in the s-plane maps to a circle in the z-plane with radius $\frac{1}{2}$ and centre at $z=\frac{1}{2}$. This circle passes through the points z=0 and z=1 and thus it is enclosed within the unit circle. The left half of the s-plane maps into the interior of this smaller circle as shown in Fig. 3(b). However, a function with poles and zeros in the right-half of the s-plane can be mapped into the interior of the unit circle but outside the smaller circle. This does not make the Poisson transform invalid for digital simulation.

The Poisson sequence of a function x(t), which is zero for t<0 is given by[77-80]

$$x_p(n) = \int_0^\infty \frac{(\gamma t)^n}{n!} e^{-\gamma t} x(t)\, dt, \qquad n=0,1,2,... \tag{15}$$

x(t), the inverse Poisson transform, is obtained from the Poisson sequence $x_p(n)$ through the equation[80]

$$x(t) = \sum_{n=0}^{\infty} x_\ell(n)\ell_n(\gamma t), \qquad t\geq 0 \tag{16}$$

where

$$x_\ell(n) = \sqrt{2\gamma}\ (-1)^n \sum_{k=0}^{n} \binom{n}{k} (-2)^k x_p(k), \qquad n=0,1,2,...$$

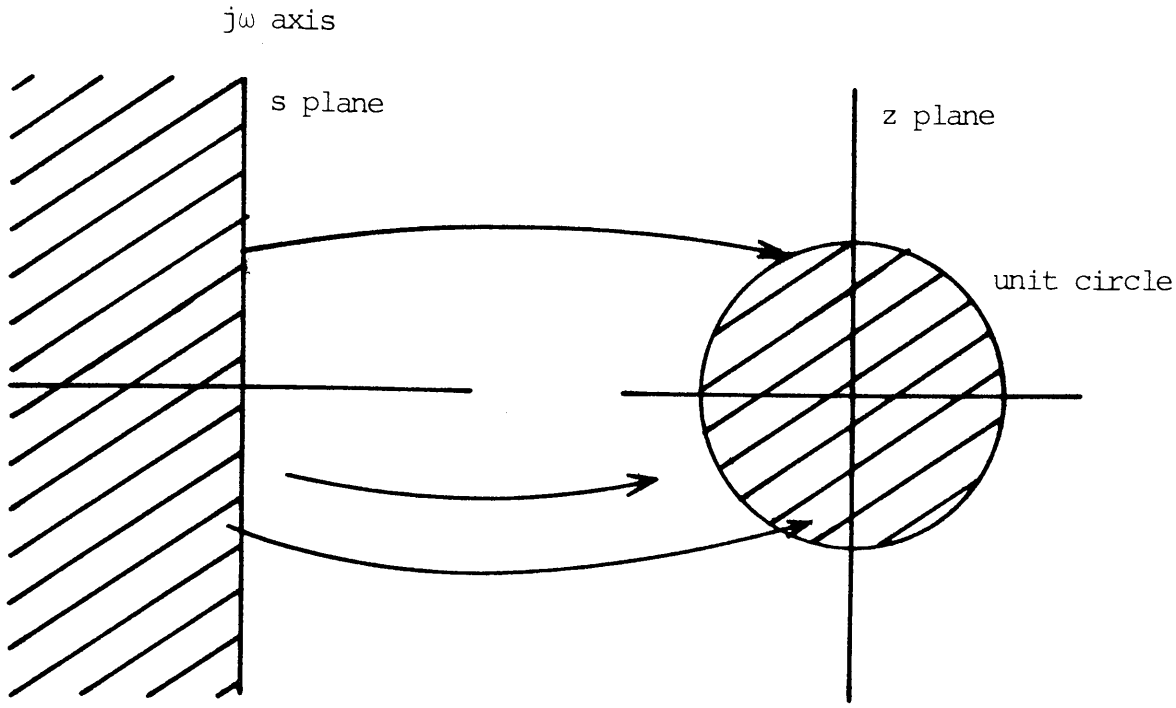


Fig. 3(a) The mapping from the s plane to the z plane corresponding to the bilinear transformation.

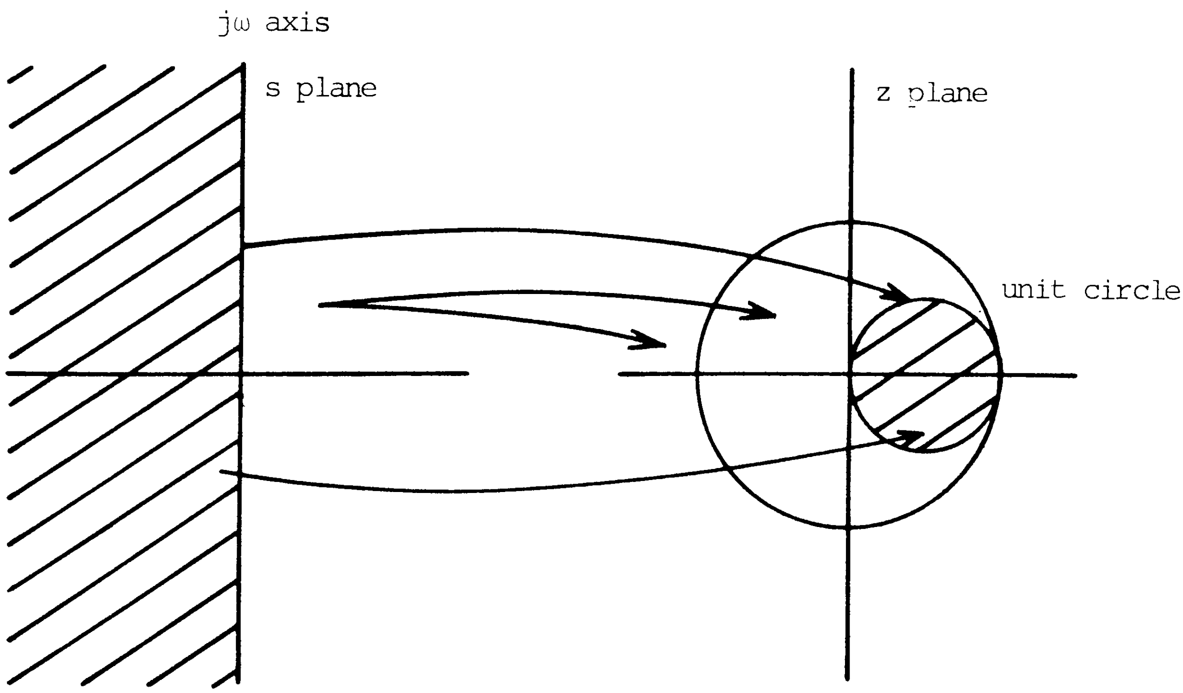


Fig. 3(b) The mapping from the s plane to the z plane corresponding to the Poisson transformation.

$$\ell_n(\gamma t) = \sqrt{2\gamma}\ (-1)^n L_n(2\gamma t)e^{-\gamma t}, \quad n=0,1,2,...$$

The Laguerre polynomial functions $L_n(y)$ are given by Eqn (13). The inverse Poisson transform is obtained by use of the Laguerre transform sequence $x_\ell(n)$, because the Poisson kernel functions $\frac{(\gamma t)^n e^{-\gamma t}}{n!}$ do not form orthonormal functions which can be used directly in a series expansion. However, the Laguerre functions $\ell_n(y)$ are orthonormal. The inverse Poisson transform can also be obtained by use of the bilinear transform and this is discussed in the next chapter.

To obtain a discrete representation of a continuous-time function $x(t)$, based on the bilinear transformation, one uses Eqn (10). The elements $x_b(n)$ can be used in place of the function $x(t)$ in numerical calculations. The probability function $x(t)$ can be recovered from the bilinear sequence $x_b(n)$ by use of Eqn. (14). To obtain a discrete representation based on the Poisson transformation, Eqn (15) is used. To recover the function $x(t)$ from the sequence $x_p(n)$, Eqn. (16) is used.

The sequence $x_b(n)$ can also be used to approximate the function $x(t)$ in the inversion of the Laplace transform as shown in the next section.

3.5 APPROXIMATE LAPLACE TRANSFORM INVERSION

The approximate Laplace transform inversion method approximates the continuous-time function by sampled data using the classical numerical integration formulas. The relationship between the Laplace transform of a continuous-time function and the z-transform of its periodic samples is given by $z=e^{sT}$, where T is the sampling period. s, the Laplace variable, is then $\frac{(\ell nz)}{T}$. This function can be approximated by a power series[72,84]

$$s = \frac{1}{T}\ln z = \frac{2}{T}\left[\left(\frac{1-z^{-1}}{1+z^{-1}}\right) + \frac{1}{3}\left(\frac{1-z^{-1}}{1+z^{-1}}\right)^3 + \ldots\right]$$

for $z>0$. Truncating this series after the first term results in the bilinear transformation

$$s = \frac{2}{T}\,\frac{(1-z^{-1})}{(1+z^{-1})} \qquad (17)$$

This corresponds to the numerical integration rule, the trapezoidal rule.

To show how this transformation can be used to compute probability distribution functions in queue analysis, the waiting time distribution function for the $M|G|1$ queue was used as an example. The Laplace transform of the waiting time density for this queue is given by[2]

$$W_L(s) = \frac{s(1-\rho)}{s-\lambda+\lambda\ B_L(s)} \qquad (18)$$

where $B_L(s)$ is the Laplace transform of the service time probability density function, ρ and λ are the utilization factor and arrival rate, respectively. The following procedure can be used to obtain the approximate waiting time distribution for this system:

(a) Divide Eqn (18) by s, to obtain the Laplace transform of the waiting time cumulative distribution function.

(b) Substitute the transformation Eqn (17) into the result, to obtain the z-transform of the approximate waiting time distribution.

(c) Invert the resulting z-transform, to obtain the approximate waiting time cumulative distibution.

If $B_L(s)$ is not a simple function, then numerical methods like the FFT algorithm or digital filtering may have to be used to obtain the inverse in (c).

For the $M|M|1$ queue, $B_L(s) = \frac{\mu}{(s+\mu)}$, and the analytic solution of the approximate waiting time cumulative distribution function can be obtained easily. This is given by the equation

$$W_a(t) = 1-\left(\frac{1+\sigma}{2\sigma}\right)\left(\frac{\sigma-\beta}{1-\beta}\right)\sigma^k, \quad t=kT,\ k=0,1,2,... \qquad (19)$$

where

$$\sigma = \frac{2-T(\mu-\lambda)}{2+T(\mu-\lambda)} \quad \text{and} \quad \beta = \left(\frac{2-\mu T}{2+\mu T}\right)$$

For this system, the exact solution of the cumulative distribution of the waiting time is given by

$$W(t) = 1- \frac{\lambda}{\mu}\exp\left[-(\mu-\lambda)t\right], \qquad t\geq 0 \tag{20}$$

Table 1 shows computed results from the approximate solution in Eqn (19) and the exact solution in Eqn (20) for values of T=0.1, 0.01, 0.001,and ρ=0.1. As can be expected, as T is decreased, the approximate solution approaches the exact solution. For very small values of T, there can be a loss of significance if the approximate solution is computed directly from Eqn (19). To avoid this, the equation can be rearranged, or extended precision arithmetic can be used.

If the z-transform,obtained by substituting Eqn (17) into(18), is inverted numerically, then the computed approximate solution will not be identical to the approximate solution due to computational errors. In turn, the approximate solution will not be identical to the exact solution,as revealed in Table 1. As T is decreased, the accuracy improves. However, if the approximate solution is computed,numerically,by either the FFT algorithm or digital filtering or recurrence equations, a reduction in T would result in more samples being needed in computing the high-order terms of the distribution function.

Some of the advantages of this approximate inversion

TABLE 1

Comparison of approximate and exact waiting time distributions for the M|M|1 queue with $\lambda=1$, $\rho=0.1$, and different values of sampling period T

t	$W_a(t)$ for T=0.1	$W_a(t)$ for T=0.01	$W_a(t)$ for T=0.001	Exact Solution W(t)
0.0	0.8746082	0.8997971	0.8999980	0.9000000
0.1	0.9524376	0.9592853	0.9593425	0.9593430
0.2	0.9819590	0.9834567	0.9834700	0.9834701
0.3	0.9931569	0.9932781	0.9932794	0.9932794
0.4	0.9974043	0.9972687	0.9972676	0.9972676
0.5	0.9990154	0.9988902	0.9988891	0.9988891
0.6	0.9996265	0.9995491	0.9995483	0.9995483
0.7	0.9998583	0.9998168	0.9998164	0.9998164
0.8	0.9999463	0.9999255	0.9999253	0.9999253
0.9	0.9999796	0.9999698	0.9999696	0.9999696

method are

(1) the method is easy to implement because the mapping from the s-plane to the z-plane is achieved by a substitution of variables, and

(2) the inversion of the resulting z-transform can be done efficiently by use of digital signal processing methods.

3.6 LAPLACE TRANSFORM INVERSION BY USE OF BILINEAR AND POISSON TRANSFORMS

A method for the inversion of the Laplace transform by use of the bilinear and Poisson transforms is presented. Suppose a continuous-time probability function $x(t)$ is given as a function of other probability functions $p_1(t)$, $p_2(t)$,..., and parameters σ_1, σ_2, ... $x(t)$ is then given by

$$x(t) = f[p_1(t),\ p_2(t),\ \ldots,\ \sigma_1,\ \sigma_2, \ldots] \tag{21}$$

where $f[.]$ is the functional operator relating the functions $p_1(t)$, $p_2(t)$,... and parameters σ_1, σ_2,... to $x(t)$. The Laplace transform of $x(t)$, from Eqn (21), will be of the form

$$X_L(s) = F[P_1(s),\ P_2(s), \ldots,\ \sigma_1,\ \sigma_2, \ldots] \tag{22}$$

where $P_1(s)$, $P_2(s)$,..., are the Laplace transforms of the probability functions $p_1(t)$, $p_2(t)$,..., respectively. $F[.]$ is the Laplace domain equivalent of the functional operator $f[.]$.

In queue analysis, the probability functions $p_1(t)$, $p_2(t)$,... and parameters σ_1, σ_2, ..., would in general be known or they could be determined. The more complex is the functional operator $F[.]$ and the functions $P_n(s)$, the more difficult it is to invert the Laplace transform $X_L(s)$. Thus numerical inversion techniques for the Laplace transforms of probability functions are of great interest in queue analysis.

To invert $X_L(s)$ by use of digital signal processing methods, it is desirable to be able to compute the bilinear and Poisson sequences from the continuous-time functions or from their Laplace transforms. It is also desirable to be able to compute the continuous-time functions from their bilinear and Poisson sequences. Methods for computing these sequences from the continuous-time functions and vice-versa are discussed in the next chapter. With the knowledge that these computational methods have been developed, it is sufficient to illustrate how these transforms can be used to compute the bilinear and Poisson sequences of probability density functions from Laplace transform expressions.

To obtain the bilinear transform from the Laplace transform, s is replaced by $\frac{\alpha(1-z^{-1})}{(1+z^{-1})}$, using the transformation Eqn (9). To obtain the Poisson transform from the Laplace transform, s is replaced by $\gamma(1-z^{-1})$. The inverse z-transforms of the resulting functions are the bilinear and Poisson sequences, respectively. The following procedure is used to invert the Laplace transform $X_L(s)$ Eqn (22) by use of the bilinear and Poisson transforms:

(a) Obtain $P_n[\frac{\alpha(1-z^{-1})}{(1+z^{-1})}]$ and $P_n[\gamma(1-z^{-1})]$, for $n=1,2,3,\ldots$. This can be done in two ways. If $P_n(s)$ is available as an expression, then replace s by $\frac{\alpha(1-z^{-1})}{(1+z^{-1})}$ and $\gamma(1-z^{-1})$, respectively. If $p_n(t)$ is available as an expression and $P_n(s)$ cannot be obtained analytically, then the bilinear and Poisson sequences of the functions $p_n(t)$ can be obtained by the digital filter methods in the next chapter. The z-transforms of the resulting sequences are then the required transforms.

(b) Substitute these transforms into the right side of Eqn (22), to obtain expressions for $X_b(z)$ and $X_p(z)$, the z-transforms of the bilinear and Poisson sequences of the function $x(t)$.

(c) Obtain the inverse z-transforms of $X_b(z)$ and $X_p(z)$ from the expressions, to get the bilinear and Poisson sequences $x_b(n)$ and $x_p(n)$, respectively. This can be done by use of either the FFT algorithm or digital filtering.

(d) Obtain the inverse bilinear and Poisson transforms of the sequences $x_b(n)$ and $x_p(n)$, to get the required function $x(t)$. Computational methods for obtaining $x(t)$ from either $x_b(n)$ or $x_p(n)$ are given in the next chapter.

As an example, the bilinear transform was used in the computation of the waiting time distribution for the G|G|1 queue. The Laplace transform of the probability density function of the waiting time for this queue is given by[2]

$$W_L(s) = \frac{K}{\left[\dfrac{A_L(-s)B_L(s)-1}{s}\right]_+} \tag{23}$$

where the notation $[\cdot]_+$ denotes the process of taking the minimum phase part (a function with all its zeros and poles in the left half of the s-plane) of the function inside the brackets. $A_L(s)$ and $B_L(s)$ are the Laplace transforms of the probability density functions of the interarrival time and the service time, respectively. K is a normalising factor, obtained from the condition $W_L(0)=1$.

The function Eqn (23) involves the process of separating functions which converge in opposite halves of the s-plane. To use the bilinear transform to invert Eqn (23), the transformation $s \to \dfrac{\alpha(1-z^{-1})}{(1+z^{-1})}$ is used. The inverse z-transform of the result is the bilinear sequence of the probability density function of the waiting time for the G|G|1 queue.

The z-transform of the bilinear sequence of the waiting time density is then

$$\frac{W_b(z)}{H_b(z)} = \frac{K\alpha(1-z^{-1})}{[A_b^-(z)B_b(z)-1][1+z^{-1}]} \tag{24}$$

where $W_b(z)$, $A_b^-(z)$, and $B_b(z)$ are the bilinear transforms of probability density functions $w(t)$, $a(-t)$ and $b(t)$, respectively. The function $H_b(z)$ has all its zeros and poles outside the unit circle. The zeros and poles of this function must cancel with all the zeros and poles of the function $(A_b^-(z)B_b(z)-1)$ which are outside the unit circle. $W_b(z)$ is the minimum phase part (all zeros and poles inside the unit circle) of the right side of Eqn (24). As can be expected, the analytic inversion of Eqn (24) is difficult except for very simple cases. Numerical techniques for inverting this equation are therefore desirable.

To compute the bilinear sequence of the waiting time density from Eqn (24), the following procedure is used

(a) Compute the bilinear sequences of the probability density functions of interarrival time and service time, $a(-t)$ and $b(t)$. These can be obtained by use of digital filter methods in the next chapter, or by replacing s by $\frac{\alpha(1-z^{-1})}{(1+z^{-1})}$ in the Laplace transforms of these functions. Let the bilinear sequences of these probability density functions be, respectively, $a_b^-(n)$ and $b_b(n)$.

(b) Convolve the sequences $a_b^-(n)$ and $b_b(n)$, to form the z-transform $A_b^-(z)B_b(z)$.

(c) Remove the zero at z=-1 from the right side of Eqn (24). The zero at z=1 in the numerator will cancel with that of the denominator.

(d) Compute the DFT of the transform $\frac{\alpha(1-z^{-1})}{(A_b^-(z)B_b(z)-1)}$ by replacing z by $\exp(j\frac{2\pi m}{L})$, for m=0,1,2,..., L-1.

(e) Compute the complex natural logarithm of this DFT.

(f) Compute the inverse DFT,to obtain the complex cepstrum corresponding to the sequence with z-transform $\frac{\alpha(1-z^{-1})}{(A_b^-(z)B_b(z)-1)}$.

(g) Set the negative time part of cepstrum to zero, to remove the contribution to the cepstrum due to the maximum phase function $H_b(z)$.

(h) Compute the DFT of the modified cepstrum.

(i) Compute the complex exponential of this DFT.

(j) Compute the inverse DFT. This gives a sequence whose z-transform is, to within a scale factor, $W_b(z)$.

(k) Normalise to obtain $w_b(k)$, with $\Sigma w_b(k)=1$

The resulting sequence is the bilinear sequence of the waiting time density. The waiting time density can then be computed from the sequence $w_b(k)$ by the methods in the next chapter. The normalisation in (k) is necessary because the bilinear transform is derived from the Laplace transform. The Laplace transform of a probability density function, evaluated at s=0 is unity. This corresponds to the evaluation of the bilinear transform at z=1, and evaluating a z-transform at z=1 corresponds to a summation of the sequence elements.

Starting with the bilinear sequences of the probability density functions a(-t) and b(t), the bilinear sequences of the probability density functions of the waiting time for the queues $M|M|1$ and $M|H_2|1$ were computed by use of the procedure outlined in steps (a) to (k). For both these queues, the interarrival time density is exponential with λ as the average arrival rate. The bilinear sequence of the function a(-t) is $a_b^-(n) = \delta(n+1)$, $n=0,\pm1,\pm2,\ldots$, where $\delta(n)$ is the delta function.

The service time density for the $M|M|1$ queue is exponential with μ as the average service rate. The bilinear sequence of this function is

$$b_b(k) = \begin{cases} \frac{\mu}{\alpha+\mu} , & k=0 \\ \frac{2\alpha\mu}{(\alpha+\mu)^2} \left(\frac{\alpha-\mu}{\alpha+\mu}\right)^{k-1} , & k=1,2,3,\ldots \end{cases} \tag{25}$$

The service time density for the $M|H_2|1$ queue considered was $b(t) = \frac{1}{4}e^{-t} + \frac{3}{2}e^{-2t}$, and the corresponding bilinear sequence for this density is

$$b_b(k)=\begin{cases}\dfrac{1}{4(\alpha+1)}+\dfrac{3}{2(\alpha+2)}, & k=0\\[2ex] \dfrac{\alpha}{2(\alpha+1)^2}\left(\dfrac{\alpha-1}{\alpha+1}\right)^k+\dfrac{3\alpha}{(\alpha+2)^2}\left(\dfrac{\alpha-2}{\alpha+2}\right)^k, & k=1,2,3,\ldots\end{cases} \tag{26}$$

The probability density function of the waiting time for the $M|M|1$ system is

$$w(t) = (1-\rho)\,\delta(t)+\lambda(1-\rho)\exp\left[-\mu(1-\rho)t\right], \quad t\geq 0$$

where ρ is the utilization factor. The bilinear sequence corresponding to this function is given by

$$w_b(k)=\begin{cases}\dfrac{(\mu-\lambda)(\alpha+\mu)}{\mu(\alpha+\mu-\lambda)}, & k=0\\[2ex] 2\alpha\left(\dfrac{\lambda}{\mu}\right)\dfrac{(\mu-\lambda)}{(\alpha+\mu-\lambda)^2}\left(\dfrac{\alpha-\mu+\lambda}{\alpha+\mu-\lambda}\right)^k, & k=1,2,3,\ldots\end{cases} \tag{27}$$

The probability density function of the waiting time and its bilinear sequence for the $M|H_2|1$ system considered are given, respectively, by

$$w(t) = \frac{3}{8}\delta(t)+\frac{3}{32}\exp\left(-\frac{3}{2}t\right)+\frac{9}{32}\exp\left(-\frac{t}{2}\right), \qquad t\geq 0$$

and

$$w_b(k)=\begin{cases} \frac{3}{8}+\frac{3}{16(2\alpha+3)}+\frac{9}{16(2\alpha+1)}, & k=0 \\ \frac{3\alpha}{4(2\alpha+3)^2}\left(\frac{2\alpha-3}{2\alpha+3}\right)^k+\frac{9\alpha}{4(2\alpha+1)^2}\left(\frac{2\alpha-1}{2\alpha+1}\right)^k, & k=1,2,3,\ldots \end{cases} \quad (28)$$

The bilinear sequences of the probability density function of the waiting time for these queues, computed from the procedure in steps (a) to (k), were compared with the results from the analytic expressions for $w_b(k)$. The computations were performed for $\lambda=1$. The accuracy of the computed results for the $M|M|1$ queue was up to 9 digits for the utilization factor $\rho=0.1$, using the Double Precision * 6 arithmetic≠ which has an accuracy of 11 digits (71), and DFT block sizes of L=128 and 256. For a utilization factor $\rho=0.9$, the accuracy of the computed results was up to 7 digits when block sizes of L=128 and 256 were used. For the $M|H_2|1$ queue, the accuracy was about 9 digits for L=128. It is worth noting that the convergence of the bilinear sequences of the service time and waiting time probability density functions can be improved for some of these functions by a proper selection of the bilinear parameter α. This can be verified by looking at the equations for the exact bilinear sequences in Eqns (25)-(28). An improved convergence results in fewer bilinear transform elements with significant magnitude and these few elements will adequately represent the continuous functions b(t)

≠ On the Harris 500 computer.

and w(t) along the real time line $0 \leq t < \infty$. The use of fewer elements will in turn result in a reduction in the computation time.

The Poisson transformation can map zeros and poles, which are in the right half of the s-plane, into the interior of the unit circle of the z-plane, but outside the circle with centre at $z = \frac{1}{2}$ and radius $\frac{1}{2}$, as shown in Fig. 3(b). Therefore the Poisson sequence of the waiting time density, obtained by use of Eqn (23) can be incorrect. The use of the bilinear transform does not result in this because the $j\omega$-axis in the s-plane is mapped onto the unit circle in the z-plane, and a stable function in the s-plane results in a stable function in the s-plane and vice-versa.

A computer program for computing the bilinear sequence of the probability density function of the waiting time for the $G|G|1$ queue, using the procedure outlined, is given in Appendix 3.

3.7 APPLICATION TO ANALYSIS OF COMPUTER COMMUNICATION SYSTEMS

In section 3.2, a diagram showing how the Poisson transform interrelates the various distributions for the M|G|1 queue was given. The Pollaczek-Khinchin transform equation for the number of customers in the queue was inverted by use of digital filtering. The bilinear and Poisson transforms were presented as suitable discrete function representations for the purpose of numerical calculations in queue analysis. A unified approach to the inversion of the Laplace transform by use of the bilinear and Poisson transforms was presented. Examples showing how these transforms can be used in computing the waiting time density for the M|G|1 and G|G|1 queues were given.

As continuous probability distribution functions used in the analysis of computer communication systems are often derived as Laplace transforms, the usefulness of these transform results depends on whether they can be inverted. Accurate inversion is very important in studying computer systems especially those with dedicated real-time applications. Frequently, performance criteria are specified in the 10^{-5} probability range[42], e.g. the probability of delay greater than τ seconds shall be less than 10^{-5}.

These methods will therefore find applications in the computation of some probability distributions like the busy period and the waiting time for computer communication systems with priority[3], the busy period for time-sharing computer systems[60], the waiting time for systems in which customers require a random number of servers[23]. In fact the methods will have applications wherever there is a need for the inversion of the Laplace transform.

CHAPTER 4

COMPUTING BILINEAR AND POISSON SEQUENCES AND THEIR INVERSES

4.1 BILINEAR AND POISSON SEQUENCES AND THEIR INVERSES BY ANALOGUE FILTERING

Bilinear and Poisson transforms can be used in the inversion of Laplace transforms of probability functions. The sequences generated by these transformations are suitable discrete representations of continuous-time probability functions. It was shown in the last chapter that these sequences can be used in place of the continuous-time functions in numerical calculations. The Poisson transform is also useful in the analysis of the M|G|1 queue. In this chapter, methods for computing the bilinear and Poisson sequences, from continuous-time functions, are presented. Methods for computing the continuous-time functions from these sequences are also given.

The digital computational methods presented are derived from analogue filters. The transfer functions for these analogue filters are given in this section. From these transfer functions and the corresponding unit impulse responses, digital filter designs for generating the bilinear and Poisson sequences are derived in sections 4.2.1 and 4.2.2. Examples of functions computed by use of these digital filters are given in section 4.2.3. In the same section, problems encountered are discussed and possible sources of computational errors are discussed. In section 4.3, methods for computing continuous-time functions, from their corresponding bilinear and Poisson sequences, are presented. Examples of functions computed by use of these methods, problems encountered, and possible sources of errors are also given.

The nth element of the bilinear sequence is given by

$$x_b(n) = \int_0^\infty x(t)h_n(t)dt, \quad n=0,1,2,... \tag{1}$$

where the functions $h_n(t)$ are[81]

$$h_n(t)=\begin{cases} \exp(-\alpha t), & n=0 \\ 2\alpha\frac{t}{n}(-1)^n \exp(-\alpha t)L_n^{(1)}(2\alpha t)+(-1)^n \frac{t}{n}\delta(t), & n=1,2,3,.. \end{cases} \tag{2}$$

$L_n^{(1)}(y)$ are the first derivative of the Laguerre polynomial function

$$L_n(y) = \sum_{k=0}^{n} \binom{n}{k} \frac{(-y)^k}{k!}, \quad n=0,1,2,... \tag{3}$$

The nth element of the bilinear sequence can be obtained as a convolution of the function x(t), time-reversed, and a system's impulse response function $h_n(t)$.

The Laplace transforms of the functions $h_n(t)$ in Eqn (2) are[81]

$$H_n(s)=\begin{cases} \frac{1}{\alpha+s}, & n=0 \\ \frac{2\alpha}{(\alpha+s)^2}\left(\frac{\alpha-s}{\alpha+s}\right)^{n-1}, & n=1,2,3,... \end{cases} \tag{4}$$

$H_n(s)$ is the transfer function of a chain of analogue all-pass filters shown in Fig. 1. When a function $x(t)$, is time-reversed, and passed through this filter chain then, at $t=0$, $x_b(n)$ the nth element of the bilinear sequence, is obtained at the output of the (n+1)th stage.

The continuous-time function $x(t)$ can be recovered from the bilinear sequence $x_b(n)$ by use of an analogue filter. $x(t)$ is obtained from the bilinear sequence $x_b(n)$ using the series expansion[81]

$$x(t) = \sum_{n=0}^{\infty} x_b(n) h_n(t) \left(\frac{n}{t}\right) \tag{5}$$

where the functions $h_n(t)$ are given in Eqn. (2).

The Laplace transform of $x(t)$ from Eqn (5) is[81]

$$X_L(s) = \sum_{n=0}^{\infty} x_b(n) \left(\frac{\alpha-s}{\alpha+s}\right)^n \tag{6}$$

The continuous-time function $x(t)$ can be obtained from its bilinear representation $x_b(n)$, as the unit impulse response of the all-pass analogue filter chain shown in Fig. 2.

The nth element of the Poisson sequence is given by[77]

$$x_p(n) = \int_0^{\infty} x(t) \frac{(\gamma t)^n}{n!} \exp(-\gamma t) dt, \quad n=0,1,2,\ldots \tag{7}$$

The Laplace transform of the Poisson kernel functions

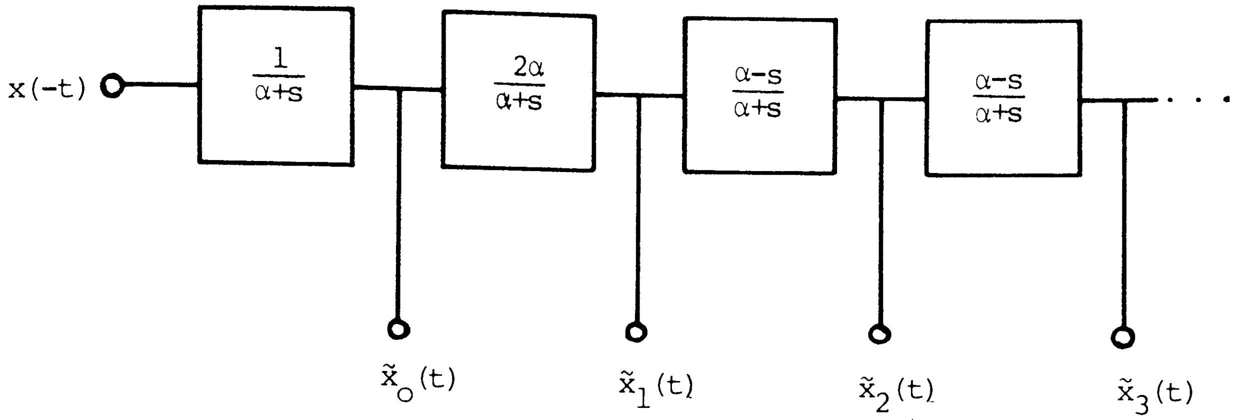


Fig. 1 All-pass network for conversion of a continuous-time function $x(t)$ to its bilinear representation $x_b(n)\,(=\tilde{x}_n(0))$

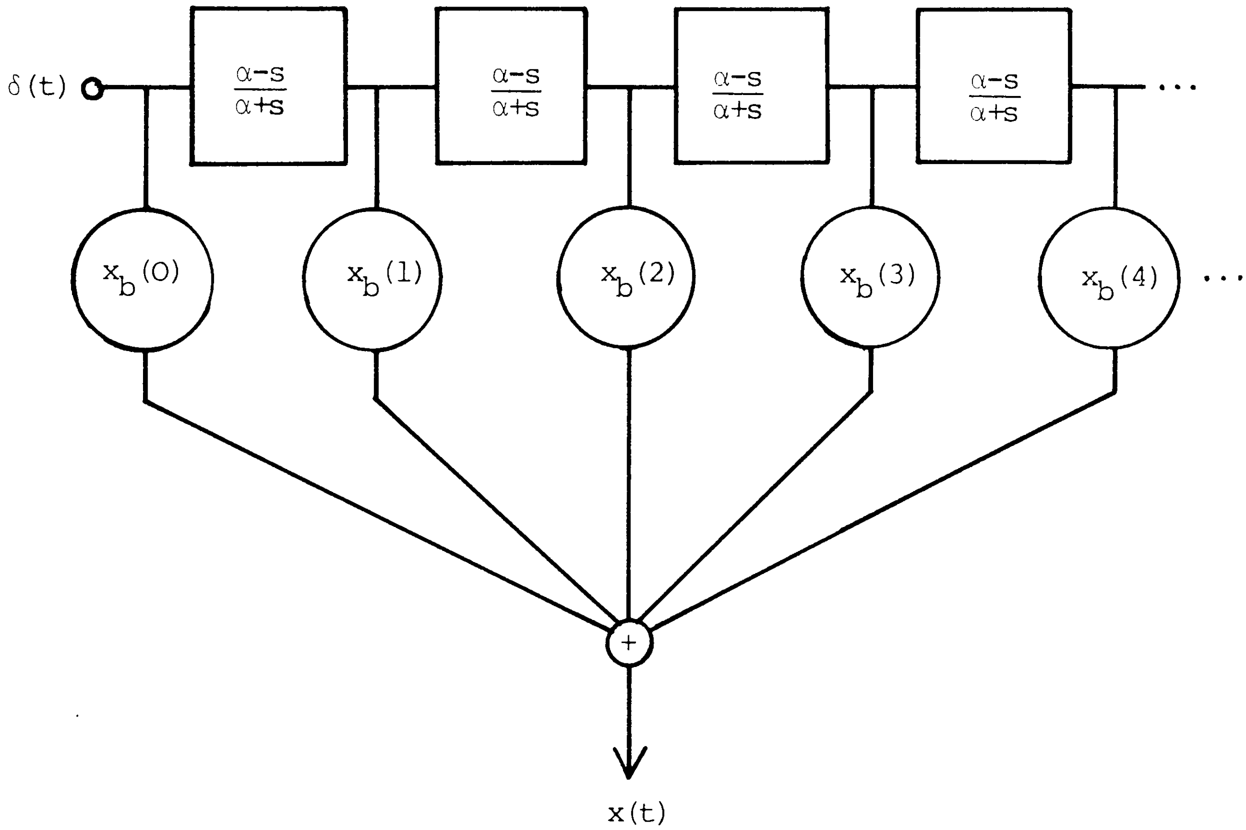


Fig. 2 The all-pass analogue filter chain for recovering a continuous-time function $x(t)$ from its bilinear representation $x_b(n)$.

$\frac{(\gamma t)^n \exp(-\gamma t)}{n!}$ is

$$H_n(s) = \left(\frac{\gamma}{\gamma+s}\right)^{n+1}, \qquad n=0,1,2,\ldots \tag{8}$$

The analogue low-pass filter chain corresponding to this transfer function is shown in Fig. 3. When a function $x(t)$ is time-reversed and passed through this filter chain, then at $t=0$, $x_p(n)$ is obtained at the output of the $(n+1)$th stage.

The function $x(t)$ is obtained from the Poisson sequence $x_p(n)$ by use of the Laguerre elements $x_\ell(n)$, from the series expansion[80]

$$x(t) = \sum_{n=0}^{\infty} x_\ell(n)\ell_n(\gamma t) \tag{9}$$

The Laguerre elements $x_\ell(n)$ are related to the Poisson elements $x_p(n)$ by

$$x_\ell(n) = \sqrt{2\gamma}\ (-1)^n \sum_{k=0}^{n} \binom{n}{k} (-2)^k x_p(k), \quad n=0,1,2,\ldots \tag{10}$$

The nth order Laguerre function $\ell_n(\gamma t)$ is related to the Laguerre polynomial function $L_n(2\gamma t)$, of order n, by the equation[80]

$$\ell_n(\gamma t) = \sqrt{2\gamma}(-1)^n L_n(2\gamma t)\exp(-\gamma t), \quad n=0,1,2,\ldots \tag{11}$$

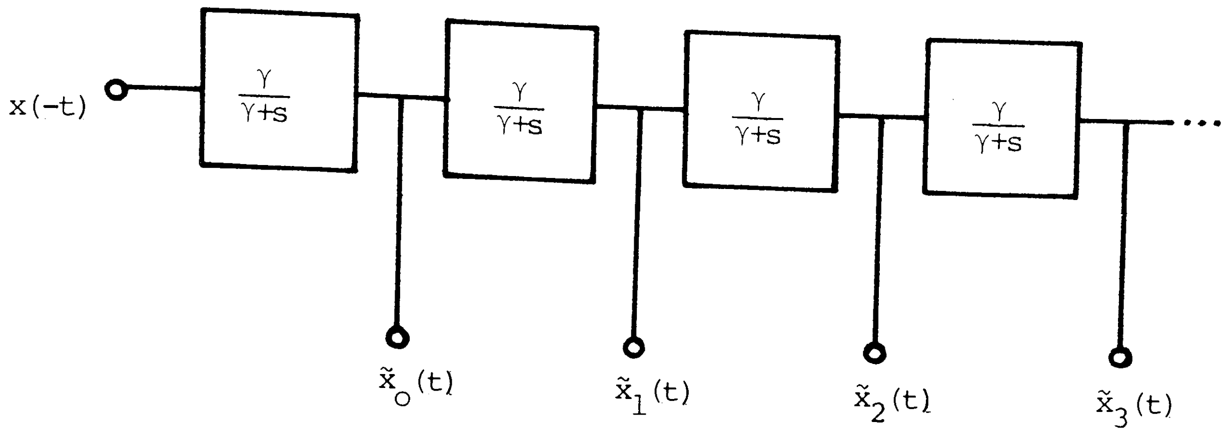


Fig. 3 Analogue filter chain for conversion of a continuous-time function $x(t)$ to its Poisson elements $x_p(n)$ $(=\tilde{x}_n(0))$.

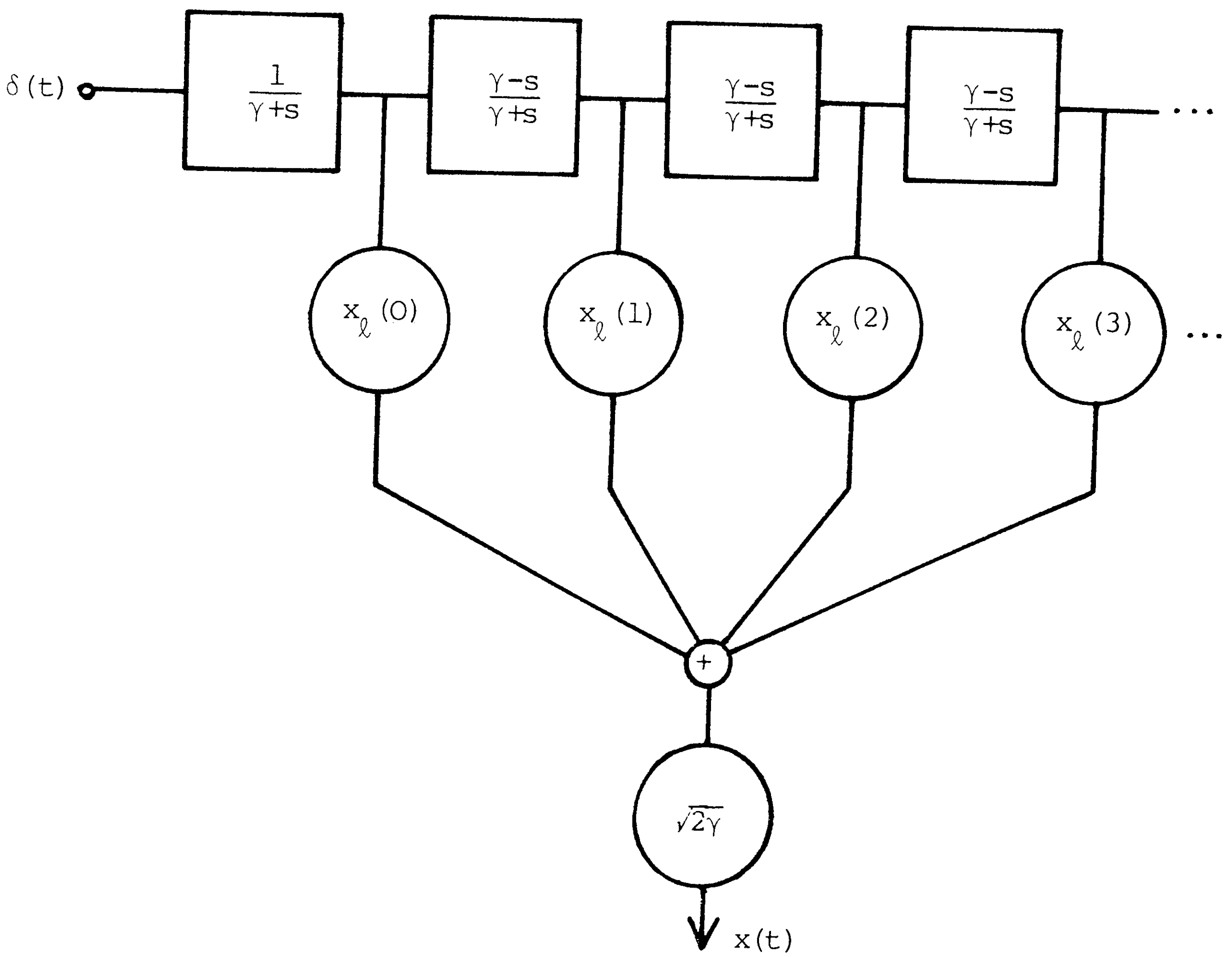


Fig. 4 Analogue filter chain for recovering a continuous-time function $x(t)$ from its Laguerre series coefficients $x_\ell(n)$.

The Laguerre polynomial functions $L_n(2\gamma t)$ are given by Eqn. (3). It can be shown that the Laplace transform of x(t), by use of Eqn (3) and Eqn (11) in Eqn (19), is given by

$$X_L(s) = \sqrt{2\gamma} \sum_{n=0}^{\infty} x_\ell(n) \left(\frac{\gamma-s}{\gamma+s}\right)^n \left(\frac{1}{\gamma+s}\right) \tag{12}$$

The analogue filter chain for recovering x(t) from $x_\ell(n)$ is shown in Fig. 4.

4.2 DIGITAL COMPUTATION OF BILINEAR AND POISSON SEQUENCES

Because of the availability of computers and efficient algorithms in digital signal processing, it is advantageous to use digital simulation to replace analogue filtering. The analogue system impulse response $h_n(t)$, must be approximated by a causal-discrete-time impulse response $h_n(k)$. The analogue input function $x(t)$ must also be transformed into a digital signal, and this is conveniently done by periodic sampling. In transforming the analogue system, with a transfer function $H_n(s)$, to a discrete system, one must obtain $h_n(k)$ or $H_n(z)$ from the analogue filter design. In such transformations, it is required that the essential properties of the analogue frequency response be preserved in the frequency response of the resulting digital system[72-74]. These essential properties are:

(1) the imaginary axis in the s-plane must map onto the unit circle of the z-plane.

(2) a stable analogue filter must be transformed to a stable digital filter.

The second property means that the transformation must map the left half of the s-plane into the interior of the unit circle in the z-plane.

Three procedures for transforming an analogue filter design to a digital filter design are described in references 72 to 74. The procedures were used to obtain digital filter transfer functions $H_n(z)$ from the unit impulse response $h_n(t)$ and the transfer function $H_n(s)$ of the analogue filters described in section 4.1.

4.2.1 IMPULSE-INVARIANT DIGITAL FILTER DESIGNS

In the design of an impulse invariant digital filter, the impulse response $h_n(t)$ of the analogue filter is obtained. This is periodically sampled to obtain $h_n(kT)$, where T is the sampling period and k is an integer. The z-transform of the resulting samples is then taken as the transfer function of the digital filter. The relationship between z, the z-transform variable, and s the Laplace transform variable, is $z=e^{sT}$. To accurately represent the analogue filter by the digital filter, the sampling rate must be greater than the Nyquist rate to avoid aliasing effects. Put in another way, to avoid aliasing, the analogue transfer function $H_n(s)$ must be bandlimited. From Eqns(4) and (8), it can be seen that the transfer functions are not bandlimited. It is thus desirable to use a very small sampling period, T, to minimise the effects of aliasing.

Bolgiano and Piovoso[78] derived the transfer function of a digital filter for generating the Poisson sequence, using the Poisson kernel functions $\frac{(\gamma t)^n \exp(-\gamma t)}{n!}$ and the impulse-invariant digital filter design procedure. The transfer function of this digital filter is[78]

$$H_n(z) = \frac{T^{n+1}}{n!} \sum_{m=0}^{n} \phi(n,m) \left[\frac{1}{1-\exp(-\gamma T)z^{-1}}\right]^{m+1}, \quad n=0,1,2,.. \tag{13}$$

where T is the sampling period. The coefficients $\phi(n,m)$ are generated recursively by the formula

$$\phi(n+1,m) = \begin{cases} 0, & m>n+1 \text{ and } n,m<0 \\ m\phi(n,m-1)-(m+1)\phi(n,m), & 1\leq m\leq n+1 \text{ and } m=0 \end{cases} \tag{14}$$

where $\phi(0,0) = 1$. The digital filter with transfer function $H_n(z)$ is shown in Fig. 5. The filter is realised as a weighted sum of the outputs of the first (n+1) stages in a chain of filters with identical transfer function $\frac{1}{[1-\exp(-\gamma T)z^{-1}]}$. To obtain the Poisson sequence of x(t), the time-reversed, periodically sampled function x(kT), is passed through the digital filter. When the last sample x(o) is processed, the output of the first (n+1) stages weighted appropriately and summed, give the nth element of the Poisson sequence.

The transfer function of the impulse-invariant digital filter for generating the bilinear sequence is derived as follows:

(a) Differentiate $L_n(y)$ in Eqn (3) to get $L_n^{(1)}(y)$.

(b) Replace $L_n^{(1)}(y)$ in Eqn (2) by the resulting expression.

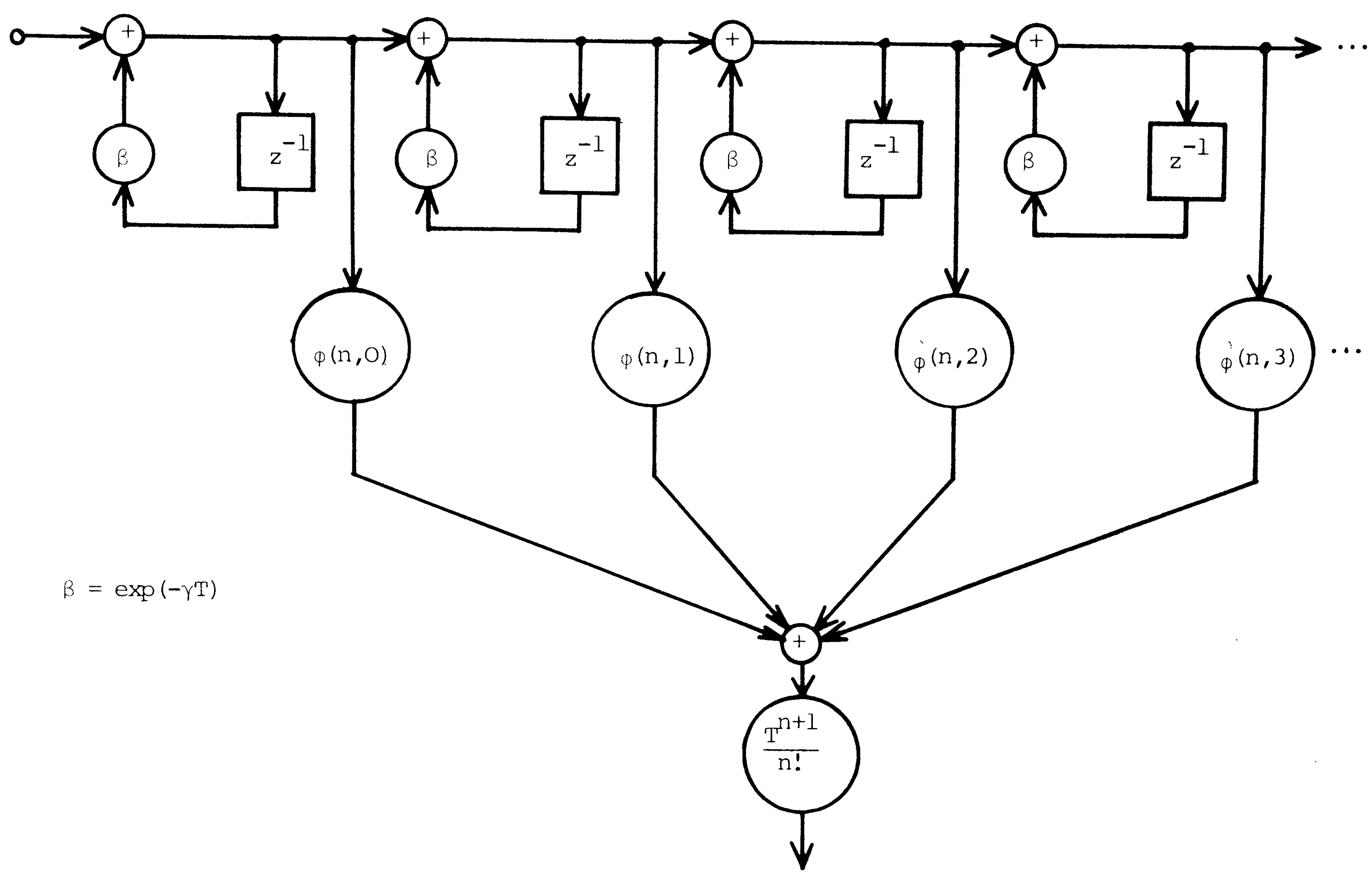


Fig. 5 Impulse invariant digital filter design for generating Poisson elements $x_p(n)$ from periodic samples of a function $x(t)$.

(c) Obtain periodic samples of $h_n(t)$ in Eqn (2) by replacing t by kT, to get $h_n(kT)$.

(d) Obtain the z-transform of $h_n(kT)$ to get $H_n(z)$.

The resulting z-transform $H_n(z)$, is

$$H_n(z) = \begin{cases} \sum_{k=0}^{\infty} \exp(-\alpha kT) z^{-k}, & n=0 \\ \frac{(-1)^n}{n} \sum_{j=1}^{n} \binom{n}{j} \frac{(-2\alpha T)^{j-1}}{(j-1)!} \; T \sum_{k=0}^{\infty} k^j \exp(-\alpha kT) z^{-k}, & n=1,2,3,.. \end{cases} \tag{15}$$

To simplify the transfer function $H_n(z)$ for $n>0$, the one-sided z-transform mentioned in reference (78) is used. The z-transform is

$$Z\{\frac{(k+1)^{[m]}}{m!} \exp(-\alpha kT)\} = \left[\frac{1}{1-\exp(-\alpha T) z^{-1}}\right]^{m+1} \tag{16}$$

where $(k+1)^{[m]}$ denotes the mth ascending factorial power of k+1 defined as $(k+1)(k+2)\ldots(k+m)$. k^j can be represented as a factorial polynomial[78]

$$k^j = \sum_{m=0}^{j} \phi(j,m) \frac{(k+1)^{[m]}}{m!} \tag{17}$$

Substituting Eqn (17) and then Eqn (16) into Eqn (15), the expression for the transfer function $H_n(z)$ simplifies to

$$H_n(z) = \begin{cases} \dfrac{1}{1-\exp(-\alpha T)z^{-1}}, & n=0 \\ \sum_{j=1}^{n} \theta(n,j) \sum_{m=0}^{j} \phi(j,m) \left[\dfrac{1}{1-\exp(-\alpha T)z^{-1}}\right]^{m+1}, & n=1,2,3,\ldots \end{cases} \tag{18}$$

where coefficients $\theta(n,j)$ are obtained from the equation

$$\theta(n,j) = T\binom{n}{j} \frac{(-1)^n}{n} \frac{(-2\alpha T)^j}{(j-1)!}, \quad 1 \leqslant j \leqslant n \tag{19}$$

The coefficients $\phi(j,m)$ are generated recursively from Eqn (14).

To simplify the realisation of the digital filter with transfer function $H_n(z)$ in Eqn (18), weighting coefficients $\Gamma(n,m)$ are obtained from the coefficients $\theta(n,j)$ and $\phi(n,m)$ using the equation

$$\Gamma(n,m) = \begin{cases} \sum_{k=1}^{n} \theta(n,k)\phi(k+1,1), & m=1 \text{ and } n \geqslant 1 \\ \sum_{k=m-1}^{n} \theta(n,k)\phi(k+1,m), & m=2,3,\ldots, n \text{ and } n \geqslant m \\ 0, & m>n \geqslant 1 \text{ and } m,n \leqslant 0 \end{cases} \tag{20}$$

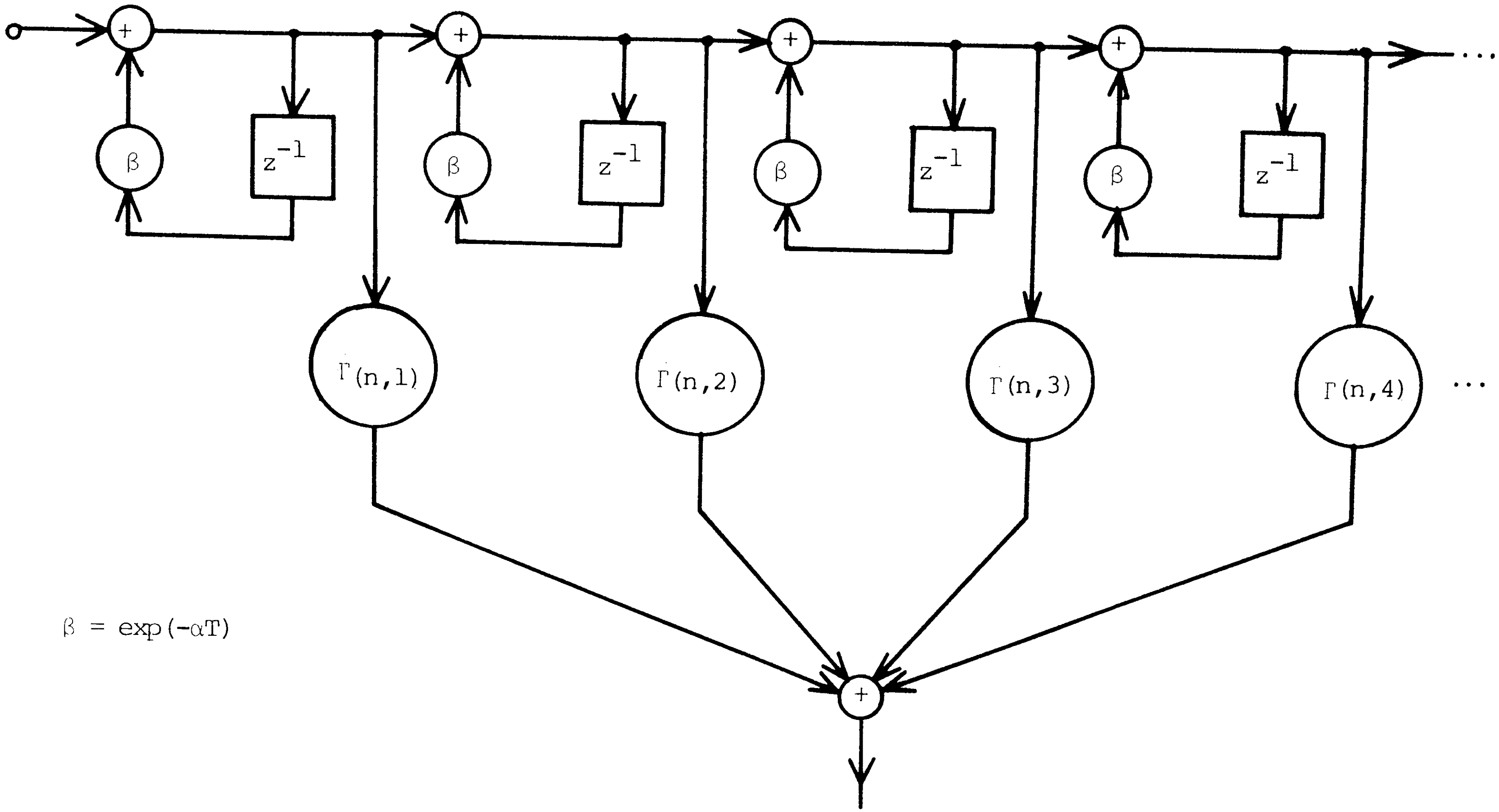
The transfer function $H_n(z)$ in Eqn (18) is obtained as a function of the coefficients $\Gamma(n,m)$ and it is then given by

$$H_n(z) = \begin{cases} \dfrac{1}{1-\exp(-\alpha T)z^{-1}}, & n=0 \\ \sum\limits_{m=1}^{n} \Gamma(n,m)\left[\dfrac{1}{1-\exp(-\alpha T)z^{-1}}\right]^m, & n=1,2,3,\ldots \end{cases} \tag{21}$$

The digital filter with transfer function Eqn (21), for $n>0$ is shown in Fig. 6. The filter is realised as a weighted sum of the outputs of the first $(n+1)$ stages in a chain of filters with identical transfer function $\frac{1}{[1-\exp(-\alpha T)z^{-1}]}$. To obtain the bilinear sequence $x_b(n)$, the periodic samples of $x(t)$, time-reversed, are passed through the digital filter. When the last sample $x(0)$ is processed, the outputs of the first $(n+1)$ stages, weighted appropriately and summed, give the nth element $x_b(n)$. The first elements $x_b(0)$, is obtained as the unweighted output of the first stage of the digital filter.

4.2.2 FILTER DESIGNS BASED ON NUMERICAL INTEGRATION TECHNIQUES

Numerical integration of differential equations is used in the design of digital filters (72-74). In the numerical integration methods, the analogue transfer function $H_n(s)$, is rearranged into a form involving terms of the form $\frac{1}{s^m}$. $H_n(s)$ need not be factored, so it is not necessary to find the roots of the numerator and denominator polynomials. Since $\frac{1}{s^m}$ implies an mth order integration in the time



$\beta = \exp(-\alpha T)$

Fig. 6 Impulse invariant digital filter design for generating bilinear elements $x_b(n)$ from periodic samples of $x(t)$.

domain, the integration operator in the z-domain is then substituted for the $\frac{1}{s^m}$ term in the analogue transfer function $H_n(s)$. The digital filter transfer function is the $H_n(z)$ that results from such a transformation. The two types of numerical integration techniques considered were the rectangular and trapezoidal integration rules.

In rectangular integration, it is assumed that the integrated function is piecewise constant, and corresponds to the recursive relation[72]

$$g_{n+1} = g_n + Tf_n \tag{22}$$

where f is the function which is being integrated and g is the value of the integral. The forward difference $g_{n+1}-g_n$ is related to Eqn (22) by the relation

$$\frac{dg}{dt} \approx \frac{(g_{n+1}-g_n)}{T} = f_n$$

Therefore Eqn (22) corresponds to the pulse transfer function

$$I(z) = \frac{T}{(1-z^{-1})} \leftrightarrow \frac{1}{s} \tag{23}$$

This corresponds to replacing s by $\frac{(1-z^{-1})}{T}$ in the expression for the analogue transfer function $H_n(s)$.

The higher order numerical integration formulas can also be used to approximate the integration process $\frac{1}{s}$.

Trapezoidal integration, which assumes the integrated function is piecewise linear, corresponds to the recursive relation[72]

$$g_n = g_{n-1} + \frac{T}{2}(f_{n-1}+f_n)$$

where g_n is the value of the integral and f_n is the integrated function. This corresponds to the trapezoidal integrating operator

$$I(z) = \frac{T(1+z^{-1})}{2(1-z^{-1})} \leftrightarrow \frac{1}{s} \tag{24}$$

By replacing s by $\frac{2(1-z^{-1})}{T(1+z^{-1})}$ in the expression for the analogue transfer function $H_n(s)$, the digital filter design based on the trapezoidal integration rule is obtained.

Using the rectangular integration rule and substituting $\frac{(1-z^{-1})}{T}$ for s in Eqn (4), the transfer function of the digital filter for generating the bilinear sequence is obtained as

$$H_n(z)=\begin{cases} \dfrac{\beta T}{1-\beta z^{-1}}, & n=0 \\[2ex] 2\alpha\left[\dfrac{\beta T}{1-\beta z^{-1}}\right]^2\left[\dfrac{\sigma+\beta z^{-1}}{1-\beta z^{-1}}\right]^{n-1}, & n=1,2,3,\ldots \end{cases} \tag{25}$$

where $\beta = \frac{1}{1+\alpha T}$

$\sigma = \alpha\beta T$

Using the rectangular integration rule and Eqn (8), the transfer function of the digital filter for generating the Poisson sequence is obtained as

$$H_n(z) = \left[\frac{\sigma}{1-\beta z^{-1}}\right]^{n+1}, \quad n=0,1,2,... \tag{26}$$

where

$$\beta = \frac{1}{1+\gamma T}$$

$$\sigma = \gamma\beta T$$

The digital filter structures corresponding to transfer functions Eqns(25) and (26) are shown in Figs. 7 and 8. By passing a periodically sampled version of the function x(t), time-reversed, through the digital filters, the bilinear and Poisson elements, $x_b(n)$ and $x_p(n)$, are obtained at the output of the (n+1)th stage of the filter chains when the last sample, x(o), is processed.

The digital filter designs based on the trapezoidal integration rule are obtained by using Eqn (24) in Eqns (4) and (8). Replacing s by $\frac{2(1-z^{-1})}{T(1+z^{-1})}$ in Eqn (4), the transfer function of the digital filter for generating the bilinear sequence is

$$H_n(z) = \begin{cases} \dfrac{\sigma(1+z^{-1})}{(1-z^{-1})}, & n=0 \\[2ex] 2\alpha\left[\dfrac{\sigma(1+z^{-1})}{(1-\beta z^{-1})}\right]^2 \left[\dfrac{-\beta+z^{-1}}{1-\beta z^{-1}}\right]^{n-1}, & n=1,2,3,... \end{cases} \tag{27}$$

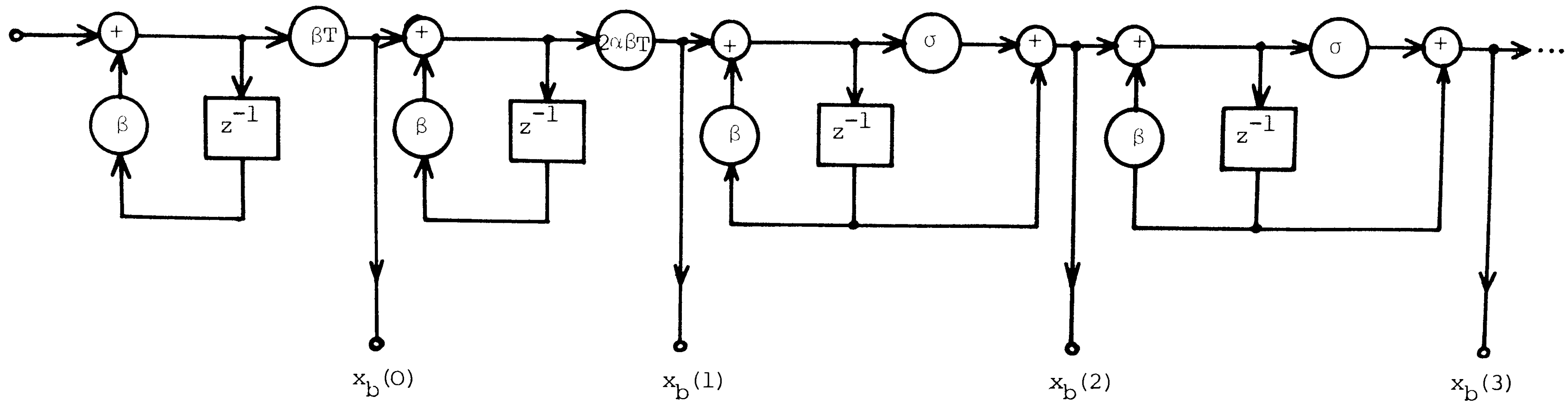


Fig. 7 Digital filter design, based on the rectangular integration rule, for generating the bilinear sequence $x_b(n)$. The input to the digital filter are the periodic samples of the function $x(t)$, in reverse order. $x_b(n)$ is obtained at the output of the $(n+1)$th stage when the last sample $x(o)$ is processed.

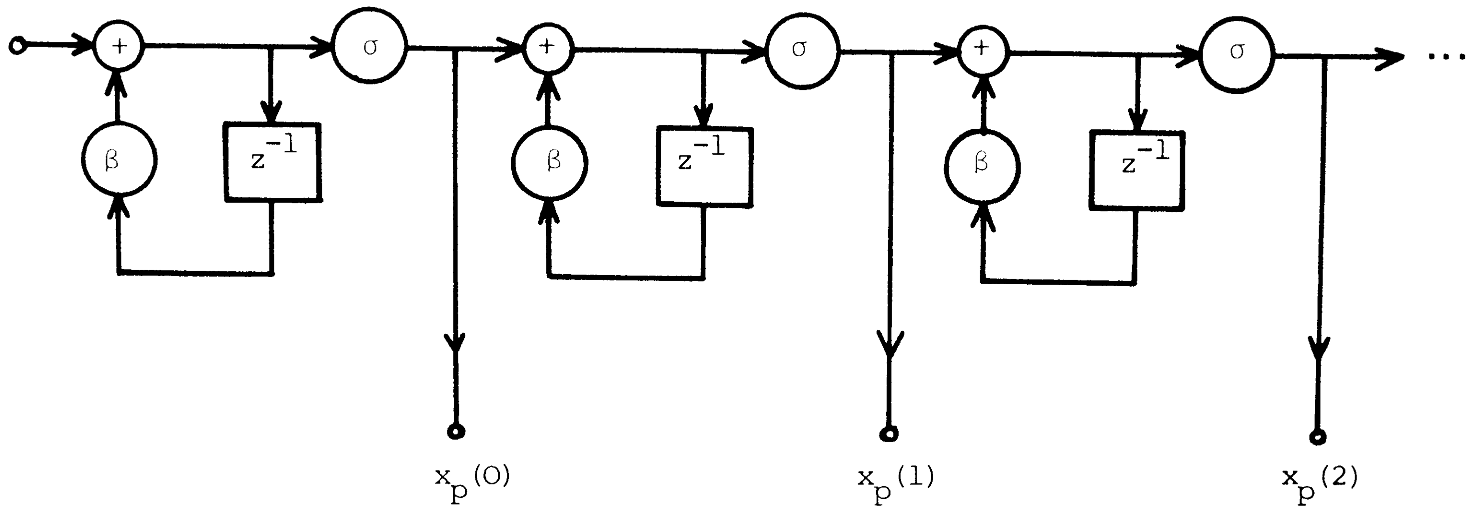


Fig. 8 Digital filter design, based on the rectangular integration rule, for generating the Poisson elements $x_p(n)$. The input to digital filter are periodic samples of the function $x(t)$, in reverse order. $x_p(n)$ is obtained at the output of the $(n+1)$th stage when the last sample $x(o)$, is processed.

where $\beta = \frac{2-\alpha T}{2+\alpha T}$

$\sigma = \frac{T}{2+\alpha T}$

The transfer function of the digital filter design, based on the trapezoidal integration rule, for generating the Poisson sequence is

$$H_n(z) = \left[\frac{\sigma(1+z^{-1})}{1-\beta z^{-1}}\right]^{n+1}, \qquad n=0,1,2,\ldots \tag{28}$$

where $\beta = \frac{2-\gamma T}{2+\gamma T}$

$\sigma = \frac{\gamma T}{2+\gamma T}$

The digital filters with transfer functions in Eqns (27) and (28) are shown in Figs. 9 and 10. When a sampled version of the function $x(t)$ is passed, time-reversed, through the digital filters, the bilinear and Poisson elements, $x_b(n)$ and $x_p(n)$, are obtained at the outputs of the $(n+1)$th stages, when the last sample $x(o)$ is processed.

4.2.3 PERFORMANCE AND COMPARISON OF COMPUTATIONAL METHODS

The digital filter designs for computing the bilinear and Poisson sequences were simulated in Fortran 77 on the

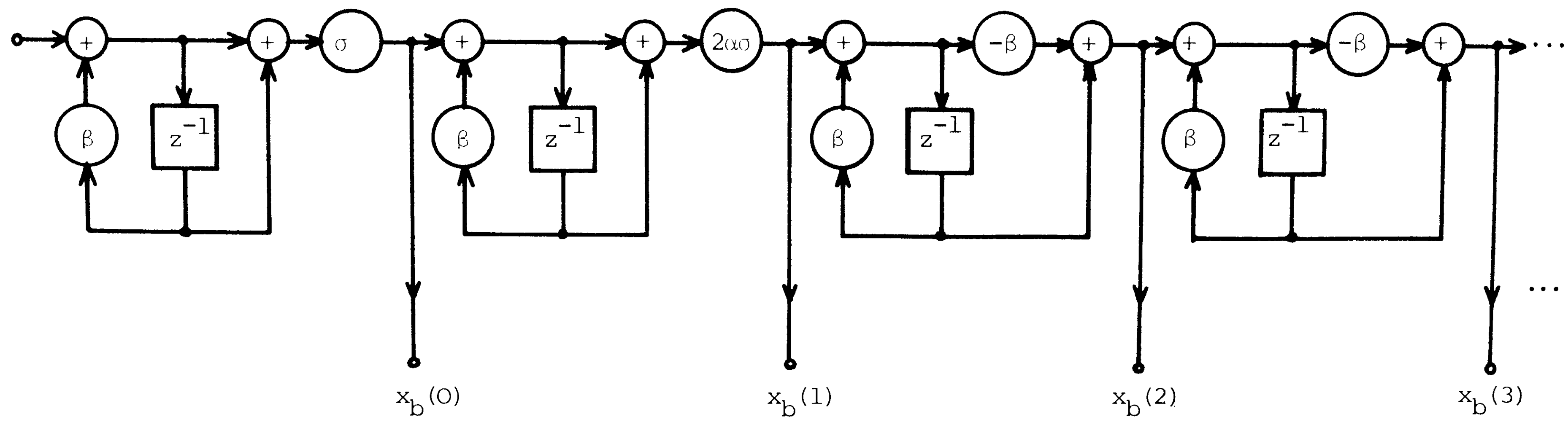


Fig. 9 Digital filter design, based on the trapezoidal integration rule, for generating the bilinear sequence $x_b(n)$, of a function $x(t)$. The input to digital filter are the periodic samples of $x(t)$ time reversed. $x_b(n)$ is obtained at the output of the $(n+1)$th stage when the last sample $x(o)$ is processed.

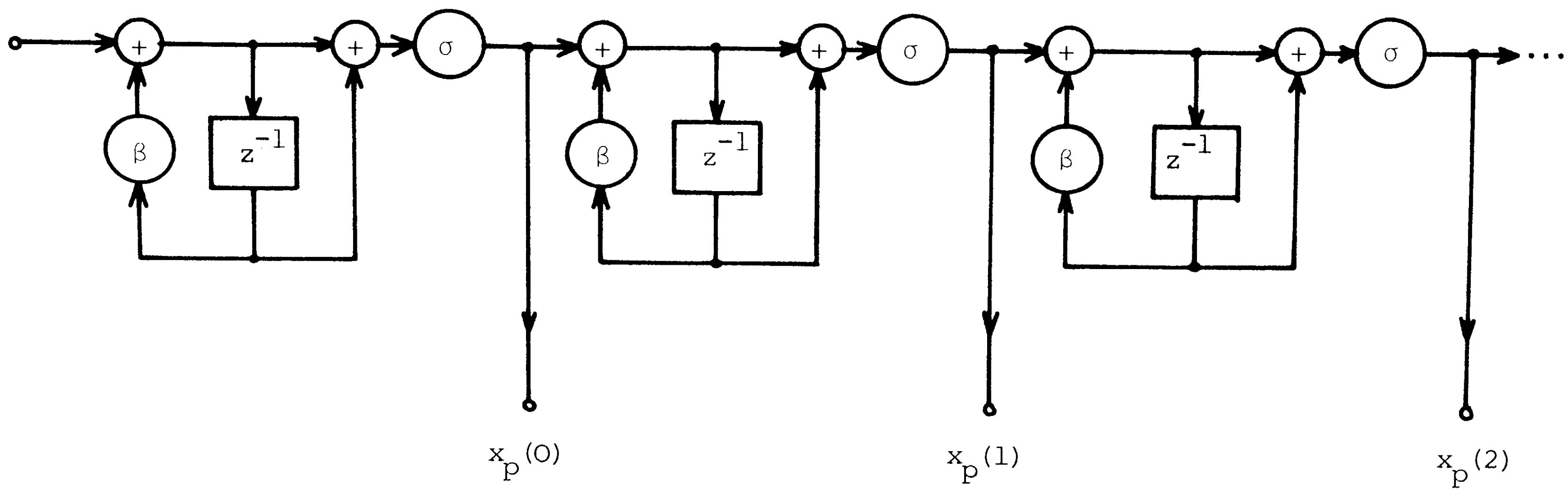


Fig. 10 Digital filter design, based on the trapezoidal integration rule, for generating the Poisson sequence $x_p(n)$. The input to the digital filter are periodic samples of the function $x(t)$, time reversed. $x_p(n)$ is obtained at the output of the (n+1)th stage when the last sample $x(o)$ is processed.

Harris 500 computer. There were no computational problems encountered in using the digital filter designs based on the numerical integration rules. However, there were some problems encountered in using the impulse invariant digital filter designs.

The impulse invariant digital filter design can result in overflow and underflow problems. The contents (see Figs. 5 and 6) of delay unit n, at clock instant i, are a sum of the contents of delay unit n-1, at instant i, and the contents of delay unit n, at instant i-1. The contents of the delay units thus accumulate. For large n, this can result in overflow, especially if many samples of the function x(t) are processed.

In using the impulse invariant digital filter designs to compute the bilinear and Poisson sequences, the coefficients $\phi(n,m)$ in Eqn (14) must be evaluated. The coefficients become large in magnitude for large n and m. The evaluation of these coefficients for large n and m can result in overflow problems.

The weighting coefficients $\Gamma(n,m)$ in Eqn (20) are used in computing the bilinear sequence. The evaluation of the coefficients $\Gamma(n,m)$, involves the multiplication of the coefficients $\phi(k,m)$ and $\theta(n,k)$, and the summation of the products. As already mentioned, the coefficients $\phi(k,m)$ become large in magnitude for large k and m. The coefficients $\theta(n,k)$ in Eqn (19) tend to zero for large n and k and small T.

The product of the coefficients $\phi(k+1,m)$ and $\theta(n,k)$ can be significant in magnitude and its contribution to the summation in Eqn (20) will then be significant. Thus, if the evaluation of the coefficients $\theta(n,k)$ suffers from underflow problems, then the significant contributions to the summation are lost, resulting in errors in the values of the coefficients $\Gamma(n,m)$. On the other hand if the evaluation of the coefficients $\phi(k,m)$ results in overflow problems, the coefficients $\Gamma(n,m)$ may not be obtained.

The bilinear and Poisson sequences of some service time densities for the $M|G|1$ queue were computed using the digital filter methods. The service time density functions considered were the exponential $b(t)=\mu e^{-\mu t}$, the hyperexponential $b(t)=0.25e^{-t}+1.5e^{-2t}$ and the Erlangian (E_2) $b(t)=4\mu^2 te^{-2\mu t}$. The computed results were compared with results from exact solutions.

The Double Precision *6 arithmetic on the Harris 500 computer gives an accuracy of 11 significant digits (71). The accuracy of the computed results was about 6 digits, for all the service time densities considered.
For comparison, the bilinear and Poisson sequences of the exponential service time density with rate $\mu = \frac{10}{9}$ were computed by use of all the digital filter designs. 10^5 time-samples and a sampling period of T=0.000162 were used. The sequences computed using the impulse invariant designs were accurate to 6 digits. The

digital filter designs based on the rectangular integration rule produced a similar accuracy. The filter designs based on the trapezoidal integration rule produced an accuracy of 8 digits. In general, the sequences computed by use of the filter designs, based on the trapezoidal integration rule, were more accurate than the sequences computed by use of the other two designs.

The use of the Double Precision *12 arithmetic, which has an accuracy of 20 digits[71] on the Harris 500 computer, only improved the accuracy by one or two digits. The value of the average service rate μ did not affect the accuracy of the computed results. The number of periodic samples used did affect the accuracy. Too few samples results in poor accuracy, and too many (say more than 10^5) does not improve the accuracy. Truncation errors also affect the computed results. The last sample of the function x(t) must be insignificant in magnitude, to minimise the truncation errors.

The filter designs based on the numerical integration rules do not require the computation of the coefficients $\phi(n,m)$ and $\theta(n,m)$ in Eqns (14) and (19), as required by the impulse invariant filter designs. These coefficients have large and small magnitudes, respectively, for large m and n. This can result in overflow and underflow problems.

Because all the digital filter designs developed were approximations to analogue filters, there were inherent errors in the computed results, due to the approximations.

4.3 INVERSION OF BILINEAR AND POISSON TRANSFORMS

The bilinear and Poisson transforms are related to the Laplace transform by substitution of variables. As a result, their inversions are closely related. The bilinear and Poisson transforms of a function $x(t)$ are obtained from the Laplace transform $X_L(s)$ by replacing the variable s by $\frac{\alpha(1-z^{-1})}{(1+z^{-1})}$ and $\gamma(1-z^{-1})$, respectively. The inverse z-transform of the resulting expressions then give the bilinear and Poisson sequences, $x_b(n)$ and $x_p(n)$, respectively. It is thus apparent that by obtaining $x(t)$ from the elements $x_b(n)$ and $x_p(n)$, the Laplace transform is inverted. Methods for computing the inverses of the bilinear and Poisson transforms are discussed in this section.

4.3.1 INVERSION BY USE OF LAGUERRE POLYNOMIAL EXPANSIONS

In the inversion of the transforms, the function $x(t)$ must be obtained from the bilinear and Poisson elements, $x_b(n)$ and $x_p(n)$, or from their respective z-transforms. The method for computing the inverse Poisson transform by use of Laguerre polynomial expansions[63] is described first. To compute $x(t)$ from the Poisson sequence $x_p(n)$, Eqns (10), (3), (11) and (9) are used. The following procedure is used to compute the function $x(t)$ from $x_p(n)$:

(a) Compute the Laguerre series coefficients $x_\ell(n)$, from the Poisson elements $x_p(n)$, by use of Eqn (10).

(b) Compute the Laguerre polynomial functions $L_n(2\gamma t)$, using Eqn (3).

(c) Compute the Laguerre functions $\ell_n(\gamma t)$ by use of Eqn. (11).

(d) Then compute the function $x(t)$ from Eqn. (9).

The computation of the Laguerre series coefficients $x_\ell(n)$ and the Laguerre polynomial functions $L_n(y)$ from Eqns (10) and (3), respectively, involves the evaluation of factorials. For large n, this will result in overflow. More efficient methods for computing $x_\ell(n)$ and $L_n(2\gamma t)$, which avoid the evaluation of factorials, are given in reference 63.

The operation of computing the Laguerre series coefficients $x_\ell(n)$ from the Poisson elements $x_p(n)$ is carried out efficiently by a multistage process[63]. This is equivalent to the use of a chain of first-order digital filters as shown in Fig. 11. The first N Laguerre series coefficients are generated by applying the first N Poisson elements in reverse order, at the input to the first stage of the filter chain. As the last element, $x_p(o)$, is applied at the input, the Laguerre series

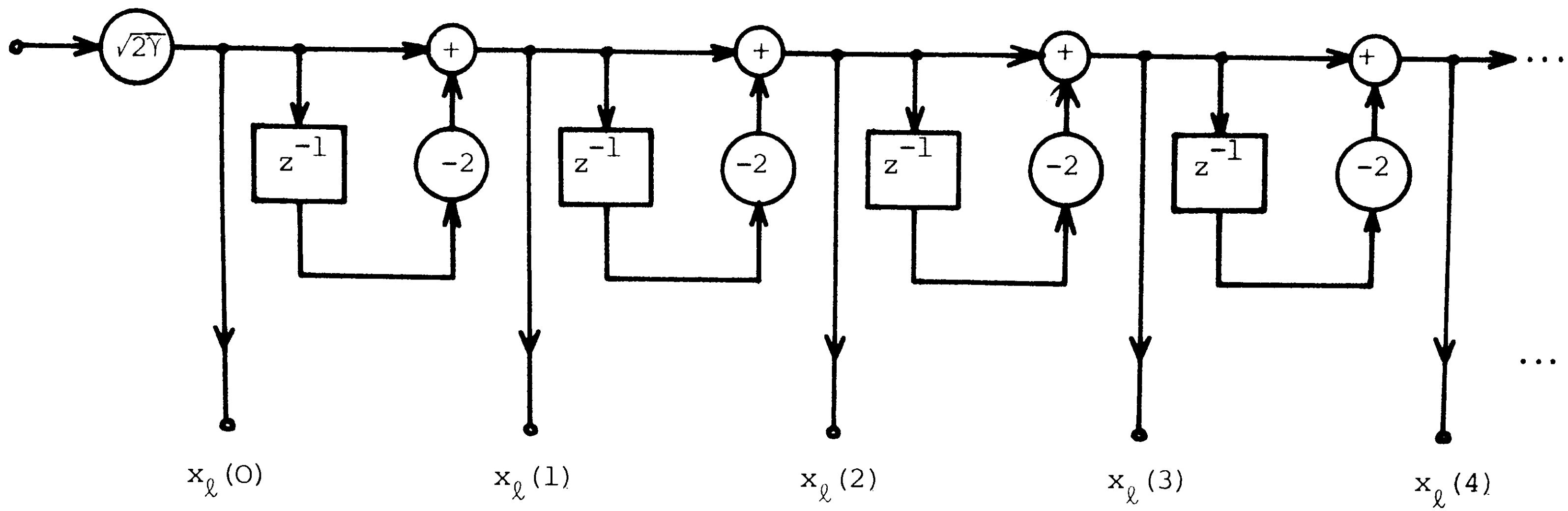


Fig. 11 Digital filter for generating Laguerre series coefficients $x_\ell(n)$ from Poisson elements $x_p(k)$.

coefficients $x_\ell(n)$, $n=0,1,\ldots,N-1$, appear at the tapping output numbers $n=1,2,\ldots N$. In this method, only multiplications by 2 are performed. A direct computation of the sequence $x_\ell(n)$, using Eqn. (10), would involve the evaluation of several factorials, for each element computed.

An efficient method for computing the Laguerre polynomial functions in Eqn (3) is now described. The method is based on the recurrence relation for the Laguerre polynomial series, given by

$$L_o(y) = 1$$

$$L_1(y) = 1-y$$

$$mL_m(y) = (2m-1-y)L_{m-1}(y)-(m-1)L_{m-2}(y), \quad m=2,3,\ldots \tag{29}$$

In computing the function $x(t)$, using Eqn (9), this method combines the process of forming the partial sums of the series, with the recursive computation of the polynomial values of the polynomial function $L_n(y)$. Smith[85] describes an algorithm for the summation of polynomial function series and their derivatives and details of his algorithm are given in Appendix 2. To use the algorithm to evaluate the function $x(t)$, Eqn (11) is substituted into Eqn (9), to obtain

$$x(t) = \sqrt{2\gamma}\; e^{-\gamma t} \sum_{n=0}^{N} (-1)^n x_\ell(n) L_n(2\gamma t) \tag{30}$$

Thus Eqn (30) is a polynomial function series truncated to N terms. Using this algorithm and Eqn (30), the set of quantities $S_N(t)$, $S_{N-1}(t), \ldots, S_o(t)$ for the Laguerre polynomial series summation are given by[63] (Appendix 2)

$$S_k(t) = \begin{cases} 0, & \text{for } k>N \\ (-1)^k x_\ell(k) + \dfrac{2k+1-2\gamma t}{k+1} S_{k+1}(t) - \dfrac{k+1}{k+2} S_{k+2}(t), & \\ & \text{for } 0 \leqslant k \leqslant N \end{cases} \tag{31}$$

The partial sum for computing the Laguerre polynomial series summation is

$$S_o(t) = \sum_{n=0}^{N} (-1)^n x_\ell(n) L_n(2\gamma t) \tag{32}$$

$S_o(t)$ can be computed efficiently by evaluating the set of quantities $S_N(t)$, $S_{N-1}(t)$, ..., giving, as the truncated Laguerre function series

$$x(t) \simeq \sqrt{2\gamma}\, e^{-\gamma t} S_o(t) \tag{33}$$

This uses approximately N multiplications fewer than a direct computation of x(t) from Eqn (30) using the recurrence relation Eqn (29).

The process for computing the function x(t) from the corresponding Poisson elements $x_p(n)$, using Smith's algorithm,is

(i) Compute the Laguerre series coefficients $x_\ell(n)$ from $x_p(n)$ using the digital filter shown in Fig. 11.

(ii) Compute the set of quantities $S_N(t)$, $S_{N-1}(t)$,... $S_o(t)$, using Eqn (31).

(iii) Compute $x(t)$ using Eqn (33) and the $S_o(t)$ obtained from Eqn (31).

The function $x(t)$ can also be computed from the Poisson sequence $x_p(n)$ by transforming $x_p(n)$ to the corresponding bilinear sequence $x_b(n)$ as follows. The mapping from the Poisson sequence to the bilinear sequence is achieved by use of the transformation

$$\gamma(1-z_p^{-1}) \rightarrow \frac{\alpha(1-z_b^{-1})}{(1+z_b^{-1})} \tag{34}$$

where z_p is the z-transform variable for the Poisson sequence and z_b is the z-transform variable for the bilinear sequence. Solving for z_p^{-1}, the following transformation is obtained

$$z_p^{-1} = \frac{1}{\gamma}\left[\frac{(\gamma-\alpha)+(\gamma+\alpha)z_b^{-1}}{1+z_b^{-1}}\right] \tag{35}$$

Let $X_b(z)$ and $X_p(z)$ be the z-transforms of the bilinear and Poisson sequences,respectively. Using the transformation Eqn. (35), $X_b(z)$ is obtained from $X_p(z)$ by use of the equation

$$X_b(z) = X_p\left[\frac{\gamma(1+z^{-1})}{(\gamma-\alpha)+(\gamma+\alpha)z^{-1}}\right]$$

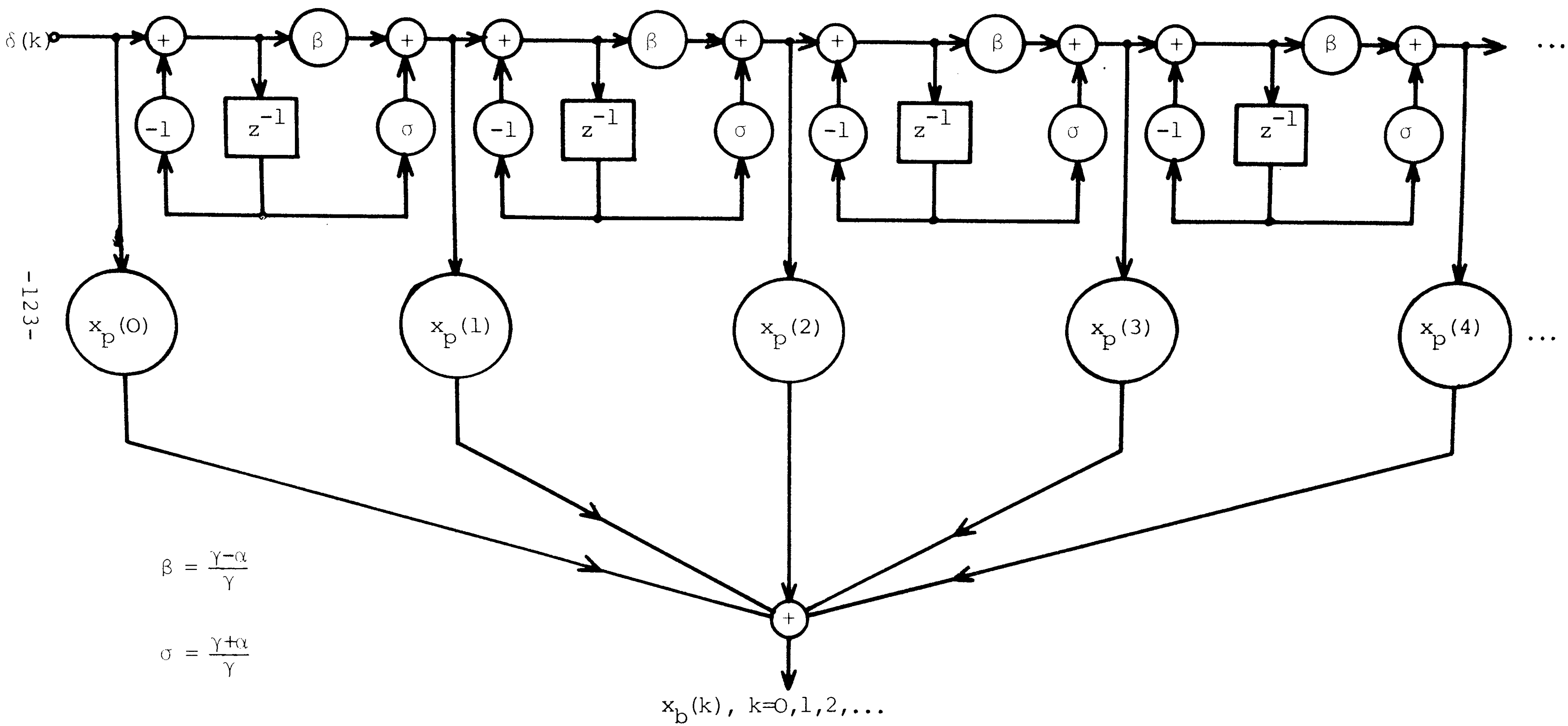
$$= \sum_{k=0}^{\infty} x_p(k)\left[\frac{(\gamma-\alpha)+(\gamma+\alpha)z^{-1}}{\gamma(1+z^{-1})}\right]^k \qquad (36)$$

$x_b(k)$, the bilinear sequence can be obtained from the Poisson sequence $x_p(k)$ as the unit-impulse response of a digital filter with transfer function $X_b(z)$ Eqn (36). The digital filter is shown in Fig. 12. The function $x(t)$ can then be computed from the sequence $x_b(n)$ by use of the method which is now described.

To compute the function $x(t)$ from its bilinear sequence $x_b(n)$, Eqn (2) is substituted into Eqn (5), to obtain

$$x(t) = \begin{cases} 2\alpha \sum_{n=1}^{\infty} (-1)^n n x_b(n), & t=0 \\ 2\alpha e^{-\alpha t} \sum_{n=1}^{\infty} (-1)^n x_b(n) L_n^{(1)}(2\alpha t), & t>0 \end{cases} \qquad (37)$$

Eqn (37) involves the summation of the first derivatives of the Laguerre polynomial series. Smith's algorithm[85] can also be used to sum the derivatives of polynomial series. The set of quantities $S_N(t)$, $S_{N-1}(t)$,..., $S_o(t)$, $S_N'(t)$, $S_{N-1}'(t)$,..., $S_1'(t)$ are obtained by use of the



$$\beta = \frac{\gamma-\alpha}{\gamma}$$

$$\sigma = \frac{\gamma+\alpha}{\gamma}$$

Fig. 12 Digital filter for generating the bilinear sequence $x_b(k)$, from the Poisson sequence $x_p(n)$.

algorithm. They are defined by the equations (Appendix 2)

$$S_k(t) = \begin{cases} 0, & \text{for } k>N \\ (-1)^k x_b(k) + \dfrac{2k+1-2\alpha t}{k+1} S_{k+1}(t) - \dfrac{k+1}{k+2} S_{k+2}(t), & \\ & \text{for } 0\leq k\leq N \end{cases} \tag{38}$$

and

$$S_k'(t) = \begin{cases} 0, & \text{for } k>N \\ -\dfrac{1}{k}S_k(t) + \dfrac{2k-1-2\alpha t}{k} S_{k+1}'(t) - \dfrac{k}{k+1} S_{k+2}'(t), & \\ & \text{for } 1\leq k\leq N \end{cases} \tag{39}$$

$S_1'(t)$ can be computed efficiently by evaluating the set of quantities $S_N(t)$, $S_{N-1}(t),\ldots S_0(t)$, $S_N'(t)$, $S_{n-1}'(t),\ldots,$ $S_1'(t)$, giving as the truncated polynomial series

$$x(t) = \begin{cases} 2\alpha \sum\limits_{n=1}^{N} (-1)^n n x_b(n), & \text{for } t=0 \\ 2\alpha e^{-\alpha t} S_1'(t), & \text{for } t>0 \end{cases} \tag{40}$$

The method uses approximately 2N multiplications less than a direct evaluation of Eqn (37) using the derivative of the recurrence relation Eqn (29).

4.3.2 INVERSION BY USE OF DIGITAL FILTERS

The function x(t) can be obtained as the unit impulse response of a chain of analogue filters shown in Figs 3 and 4. As digital filters can be used to approximate analogue filters, they can also be used in the inversion of the bilinear and Poisson transforms. In this section, digital filters for generating the function x(t), from the bilinear and Laguerre elements, $x_b(n)$ and $x_\ell(n)$, are obtained. These digital filter designs are based on the numerical integration rules.

x(t), the inverse bilinear transform of the sequence $x_b(n)$, is obtained as the unit impulse response of an analogue filter with transfer function $X_L(s)$, Eqn (6). A digital filter design, based on the rectangular integration rule, to approximate an analogue filter is obtained by use of the transformation $s = \frac{(1-z^{-1})}{T}$. Using this transformation in Eqn (6), the transfer function of a digital filter for generating the function x(t) from $x_b(n)$ is

$$X_L(z) = \sum_{n=0}^{\infty} x_b(n) \left[\frac{-\beta+\sigma z^{-1}}{1-\sigma z^{-1}}\right]^n \qquad (41)$$

where $\beta = \frac{1-\alpha T}{1+\alpha T}$

$$\sigma = \frac{1}{1+\alpha T}$$

The transfer function of a digital filter design, based on the rectangular integration rule, for generating the inverse

Poisson transform from Eqn (12) is

$$X_L(z) = \sqrt{2\gamma} \sum_{n=0}^{\infty} x_\ell(n) \left[\frac{T\sigma}{1-\sigma z^{-1}}\right]\left[\frac{-\beta+\sigma z^{-1}}{1-\sigma z^{-1}}\right]^n \tag{42}$$

where $\beta = \dfrac{1-\gamma T}{1+\gamma T}$

$\sigma = \dfrac{1}{1+\gamma T}$

The digital filter structures corresponding to the transfer functions Eqn (41) and Eqn (42) are shown in Figs 13 and 14, respectively. x(t) is thus approximated by the unit impulse response of these digital filters at t=0,T,2T,... .

For the digital filter designs based on the trapezoidal integration rule, s is replaced by $\dfrac{2(1-z^{-1})}{T(1+z^{-1})}$ in Eqns (6) and (12). The transfer function of the digital filter for generating the inverse bilinear transform is

$$X_L(z) = \sum_{n=0}^{\infty} x_b(n) \left[\frac{-\beta+z^{-1}}{1-\beta z^{-1}}\right]^n \tag{43}$$

where $\beta = \dfrac{2-\alpha T}{2+\alpha T}$

The transfer function of the digital filter design, based on the trapezoidal integration rule, for generating the inverse Poisson transform, is

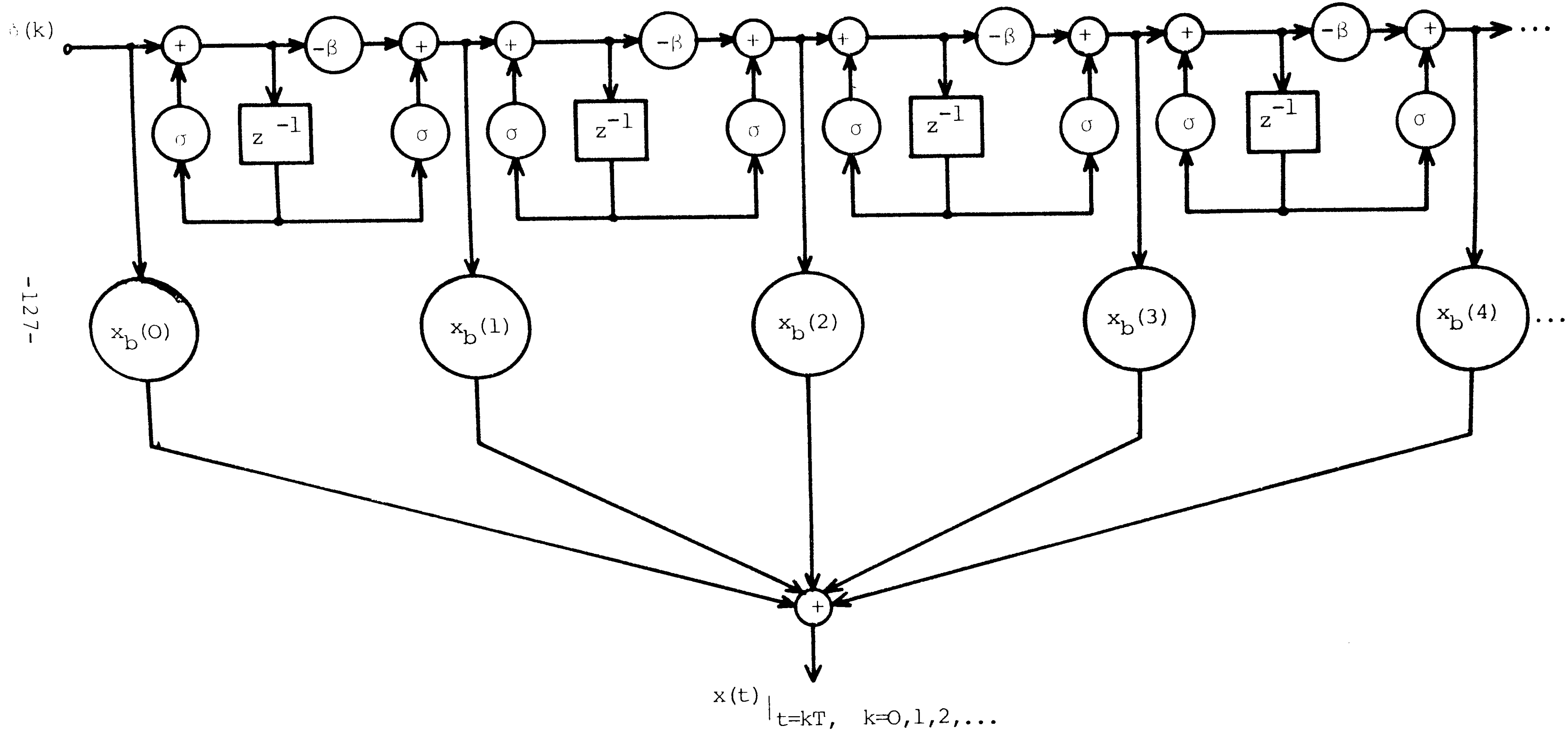


Fig. 13 Digital filter design, based on the rectangular integration rule, for generating $x(t)$ from its bilinear representation $x_b(n)$.

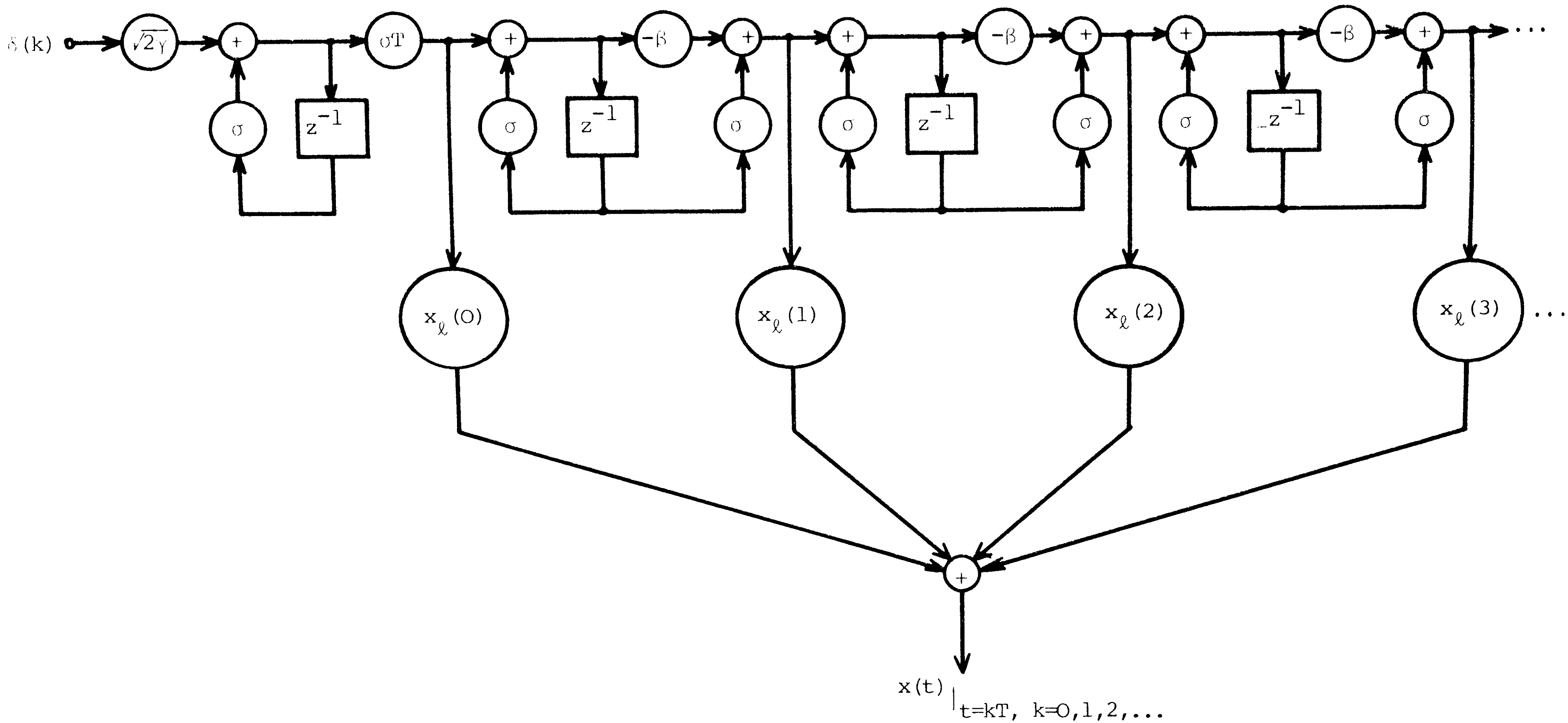


Fig. 14 Digital filter design, based on the rectangular integration rule, for generating the function $x(t)$ from Laguerre series coefficients $x_\ell(n)$.

$$X_L(z) = \sqrt{2\gamma} \sum_{n=0}^{\infty} x_\ell(n) \left[\frac{\sigma(1+z^{-1})}{(1-\beta z^{-1})}\right] \left[\frac{-\beta+z^{-1}}{1-\beta z^{-1}}\right]^n \tag{44}$$

where

$$\beta = \frac{2-\gamma T}{2+\gamma T}$$

$$\sigma = \frac{T}{2+\gamma T}$$

The digital filters corresponding to these transfer functions are shown in Figs. 15 and 16. The unit impulse responses of these digital filters at clock instants $k=0,1,2,...$ approximate the function $x(t)$, at $t=0,T,2T,...$.

4.3.3 PERFORMANCE OF COMPUTATIONAL METHODS

In computing the inverses of the bilinear and Poisson transforms by use of the Laguerre polynomial functions, there are possibilities of overflow and underflow problems. If the function $x(t)$ approaches zero very slowly as t increases, then there can be overflow or underflow problems. $x(t)$, from Eqns (30) and (37), consists of summations of polynomial series weighted by factors $\exp(-\gamma t)$ and $\exp(-\alpha t)$, respectively. For large values of γt and αt, the exponential terms can suffer underflow. The summations of the polynomial series can on the other hand result in overflow problems.

In computing the inverse Poisson transform, the Poisson sequence must first be transformed to Laguerre series coefficients or to the bilinear sequence. The gains of the

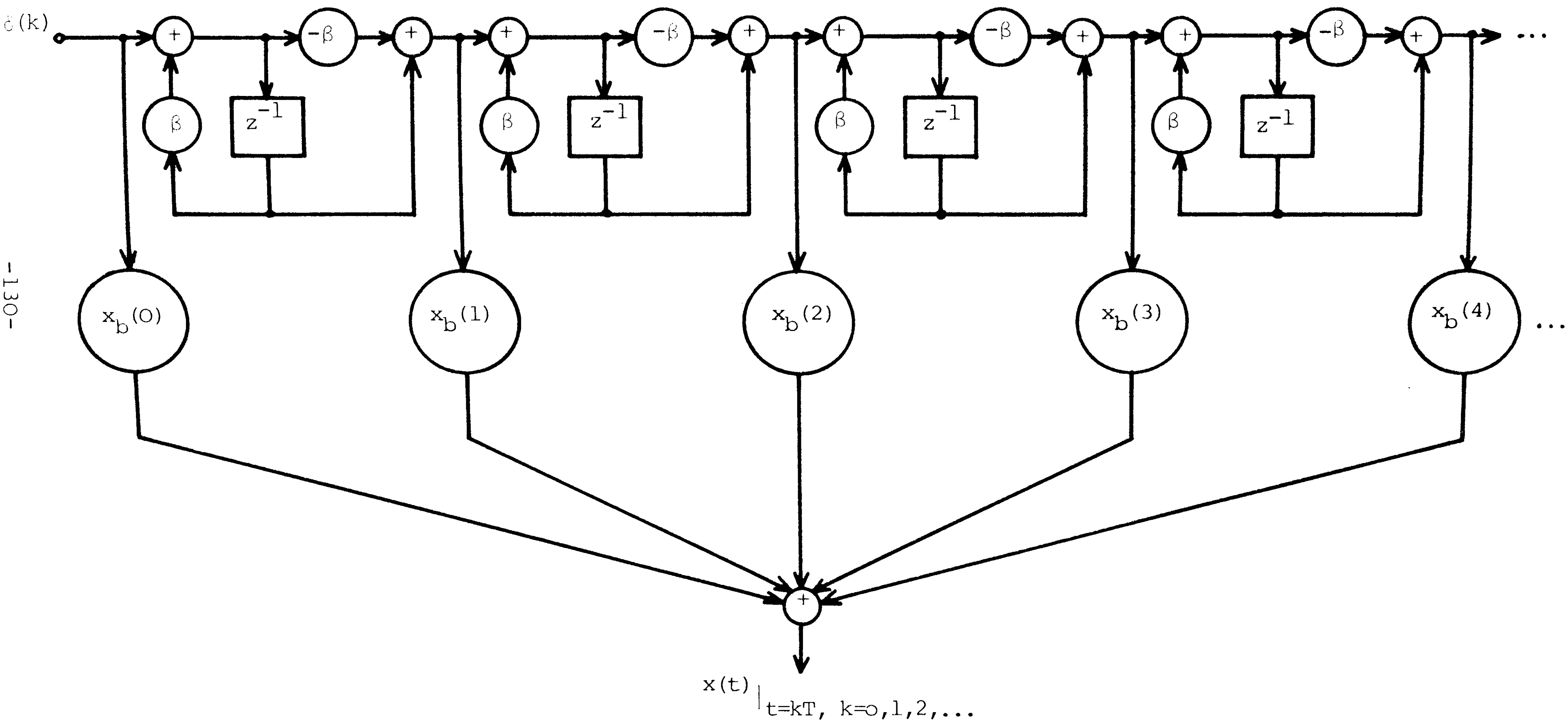


Fig. 15 Digital filter design, based on the trapezoidal integration rule, for generating the function $x(t)$ from its bilinear representation $x_b(n)$.

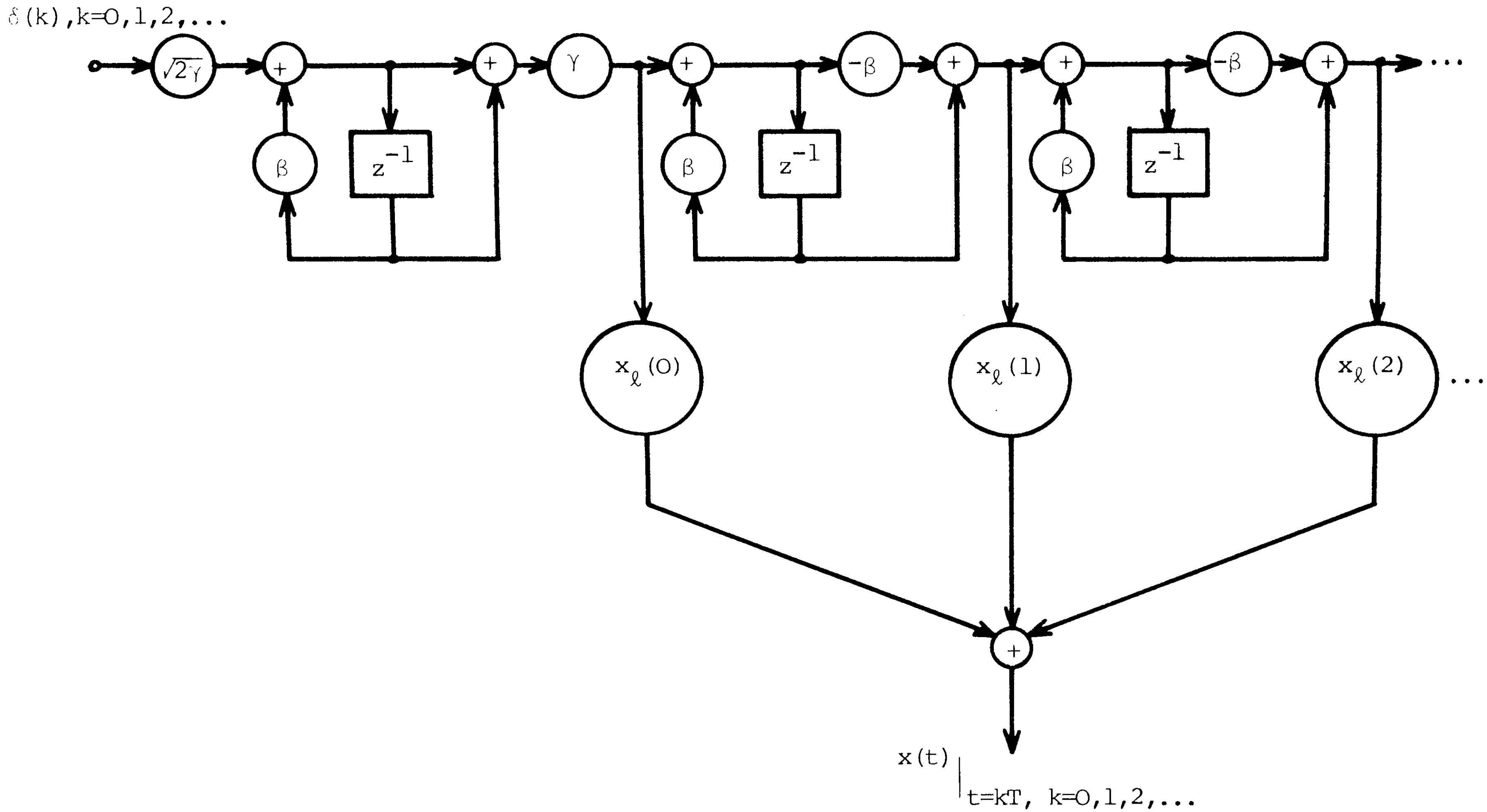


Fig. 16 Digital filter design, based on the trapezoidal integration rule, for generating $x(t)$ from Laguerre series coefficients $x_\ell(n)$.

digital filters for generating the Laguerre series coefficients and the bilinear elements from the Poisson elements, are high at the Nyquist frequency[63]. Thus the computed values of the high order Laguerre series coefficients and bilinear elements are determined primarily by the high frequency components of the Poisson sequence $x_p(n)$. The high frequency components can be dominated by errors due to their representation. These errors become significant in computing the high order Laguerre series and bilinear elements. It is therefore necessary to check the computed high order elements for any increase in magnitude due to the accumulation of the errors. This can be difficult to detect because the exact elements may increase in magnitude as n is increased, even though they may tend to zero for very large n.

The digital filter methods for generating the function x(t) from its bilinear and Poisson elements are approximations of analogue filters through transformations. Therefore there are inherent errors due to these approximations, and these errors are unavoidable.

To assess the accuracy of the inversion method which uses Laguerre series expansions , the Erlangian (E_4) service time density was used. The service time density is[2,8]

$$b(t) = \frac{4\mu(4\mu t)^3}{3!} \exp(-4\mu t), \qquad t \geq 0 \tag{45}$$

The bilinear and Poisson sequences of b(t) are obtained from the Laplace transform $B_L(s) = [\frac{4\mu}{(s+4\mu)}]^4$. The bilinear sequence is given by

$$b_b(k) = \begin{cases} (\frac{4\mu}{\alpha+4\mu})^4, & k=0 \\ 4(1+\beta)(\frac{4\mu}{\alpha+4\mu})^4, & k=1 \\ 2(3+5\beta)(1+\beta)(\frac{4\mu}{\alpha+4\mu})^4, & k=2 \\ 4(1+5\beta+5\beta^2)(1+\beta)(\frac{4\mu}{\alpha+4\mu})^4, & k=3 \\ (\frac{4\mu}{\alpha-4\mu})^4[\binom{k+3}{3}(1+\beta)^4-4\binom{k+2}{2}(1+\beta)^3+6\binom{k+1}{1}(1+\beta)^2- & \\ \qquad 4\binom{k}{0}(1+\beta)]\beta^k, & k=4,5,6,.. \end{cases} \tag{46}$$

where $\beta = \frac{(\alpha-4\mu)}{(\alpha+4\mu)}$. The elements $b_b(1)$, $b_b(2)$ and $b_b(3)$ can all be obtained from the last line of Eqn (46). However, for a value of α close to that of 4μ, there will be a loss of significance in the computed elements if the last line is used. It is therefore necessary to rearrange the equation into the form given in Eqn (46). The Poisson sequence corresponding to the service time density b(t) is

$$b_p(k) = (\frac{4\mu}{\gamma+4\mu})^4\binom{k+3}{3}(\frac{\gamma}{\gamma+4\mu})^k, \quad k=0,1,2,... \tag{47}$$

For practical purposes, the probability distribution function, $B(t) = \int_0^t b(\tau)d\tau$, or the complementary distribution function, 1-B(t), are usually of greater interest than the probability density function. However, the function B(t) approaches unity for large t, therefore the magnitudes of the high order bilinear and Poisson elements approach a non-zero constant. This will result in slow convergence of the series in Eqns (30) and (37), especially for large values of t. Instead of computing the probability distribution function, B(t), faster convergence of the series is obtained by computing the complementary function, 1-B(t), whose bilinear and Poisson elements approach zero for large k.

The complementary function d(t) = 1-B(t) can be obtained from Eqn (45) and it is given by

$$d(t) = \frac{\exp(-4\mu t)}{3}\ [32(\mu t)^3+24(\mu t)^2+12\mu t+3], \quad t \geq 0 \qquad (48)$$

The Laplace transform of d(t) is $D_L(s) = \frac{[1-B_L(s)]}{s}$ where $B_L(s)$ is the Laplace tranform of the service time density, b(t). Replacing s by $\frac{\alpha(1-z^{-1})}{(1+z^{-1})}$ and $\gamma(1-z^{-1})$ in $D_L(s)$, and inverting the resulting z-transforms, the bilinear and Poisson sequences of d(t), i.e., $d_b(k)$ and $d_p(k)$, are obtained.

To compute the function d(t) from the bilinear and Poisson sequences, $b_b(k)$ and $b_p(k)$, in Eqns (46) and (47), the following process is performed.

(i) Obtained the bilinear and Poisson sequences of d(t), from $b_b(k)$ and $b_p(k)$, respectively. The bilinear sequence $d_b(k)$ is obtained by passing the sequence $\{(1-b_b(o)), -b_b(1), -b_b(2), \ldots\}$ through a digital filter with a transfer function $\frac{(1+z^{-1})}{\alpha(1-z^{-1})}$. The Poisson sequence $d_p(k)$ is obtained by passing the sequence $\{(1-b_p(o)), -b_p(1), -b_p(2), \ldots\}$ through a digital filter with a transfer function $\frac{1}{\gamma(1-z^{-1})}$.

(ii) Using the methods outlined in section 4.3, the function $d(t)$ is computed from its bilinear and Poisson sequences, $d_b(k)$ and $d_p(k)$.

It is sometimes necessary to select suitable values for the bilinear and Poisson parameters α and γ, respectively, in computing functions from their bilinear and Poisson sequences. For some functions, these parameters can be used to improve the convergence of the bilinear and Laguerre elements and hence the series Eqns (30) and (37). The choice of the Poisson parameter γ is important in the computation of the inverse Poisson transform. The choice must be such that the Poisson sequence transforms to rapidly converging Laguerre series or bilinear elements. As a result, the computed high-order Laguerre or bilinear elements will become insignificant in magnitude before they become grossly affected by computational errors.

To select a value for the parameters α and γ, the analytic expressions for the bilinear and Laguerre sequences or their z-transforms are examined. As an example, the convergence of the bilinear sequence in Eqn (46) can be improved by choosing a value for α which is as close to the value of 4μ as possible. Alternatively, if expressions for the z-transforms of the sequences are available, the pole positions of these transforms can be determined. If these pole positions are functions of the parameter α or γ, then these parameters can be used to push the pole positions inwards towards the origin, in the z-plane. Otherwise, one experiments with several values for these parameters.

The function $d(t)$ was computed from the bilinear and Poisson sequences $b_b(k)$ and $b_p(k)$ using three approaches:

(i) It was computed directly from its bilinear elements $d_b(k)$ which had been obtained from $b_b(k)$.

(ii) $d(t)$ was computed from the Laguerre series coefficients $d_\ell(k)$. The coefficients $d_\ell(k)$ had been obtained from the Poisson elements $d_p(k)$, which in turn, were obtained from the elements $b_p(k)$.

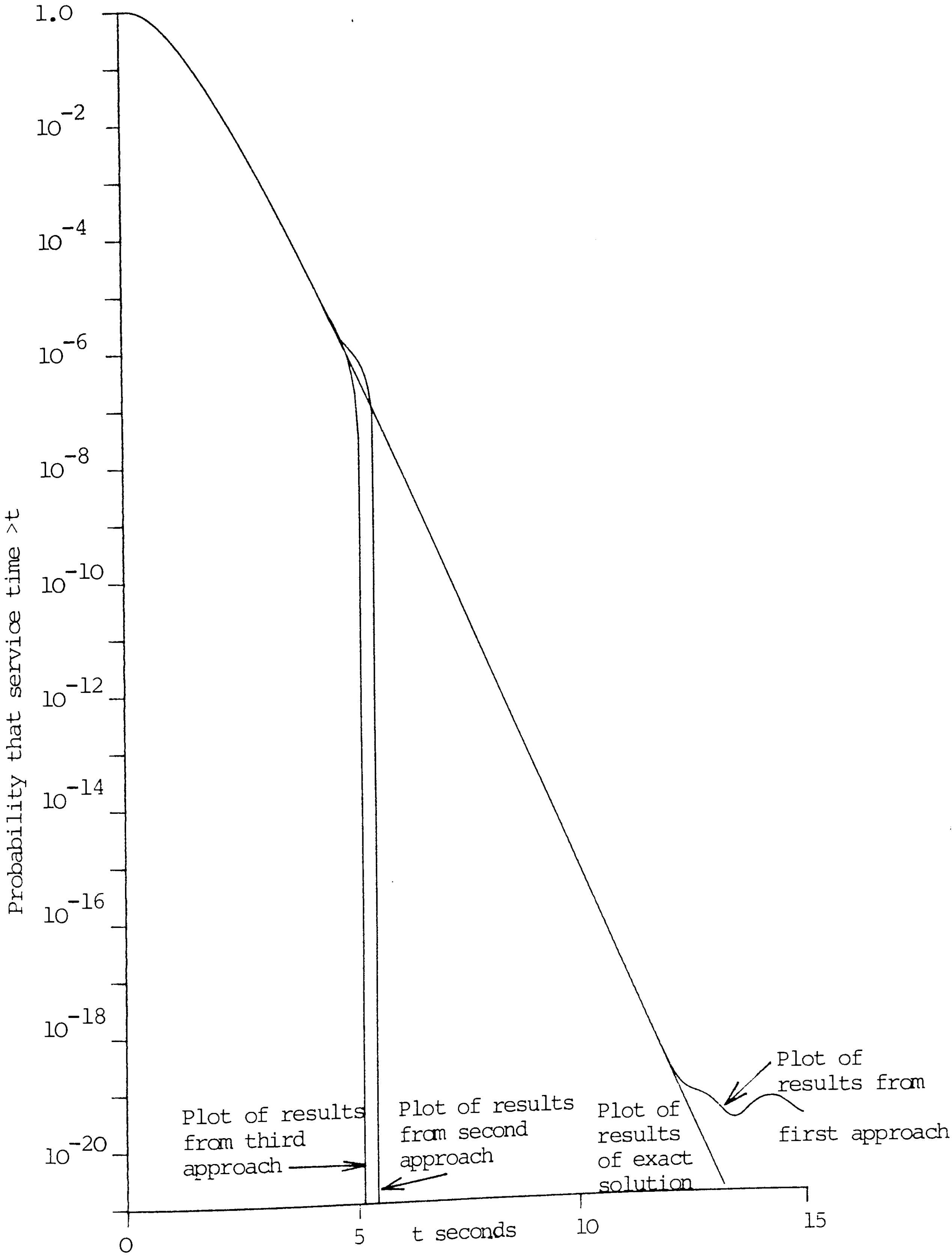
(iii) $d(t)$ was also computed from the bilinear elements $d_b(k)$ which had been obtained from the Poisson elements $d_p(k)$. The Poisson elements $d_p(k)$ had in turn, been obtained from the Poisson elements $b_p(k)$.

For a service rate $\mu = \frac{10}{9}$, $\alpha=\gamma=1$, and using Double Precision *12 which has an accuracy of 20 digits on the Harris 500 computer, the computed results were accurate to at least 17, 6 and 5 digits, respectively, for the three approaches. Graph 1 is a plot of the computed results from the three approaches. For $\alpha=\gamma=4.5$, the computed results were accurate up to at least 18, 16 and 15 digits, respectively. A plot of the results from these three approaches is indistinguishable from the exact solution Eqn (48), in Graph 1.

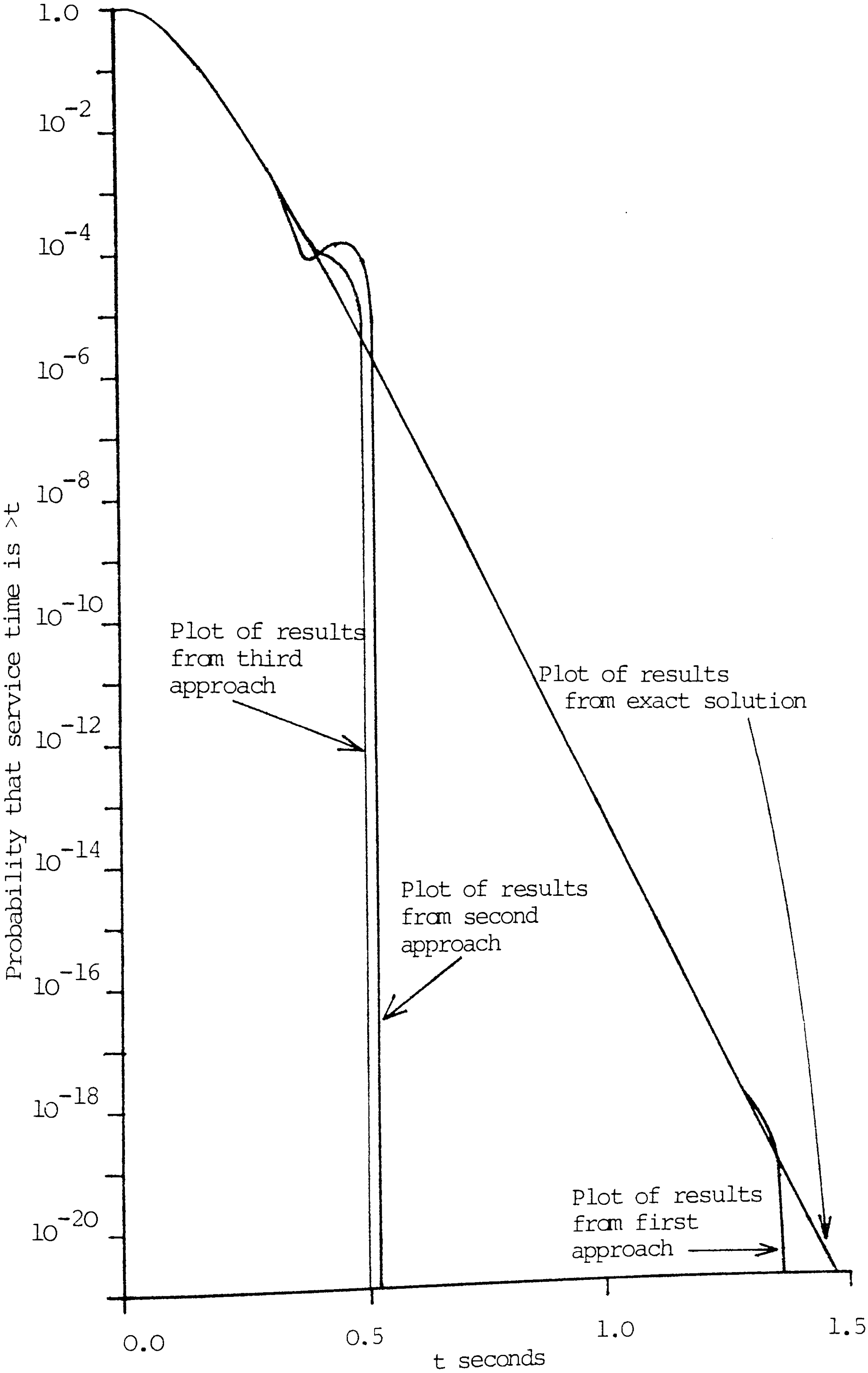
For a service rate $\mu=10$ and $\alpha=\gamma=5$, the computed results were accurate to 15, 2 and 2 digits, respectively, for the three approaches as shown in Graph 2. For the same service rate and $\alpha=\gamma=20$, the accuracies were to about 17, 11 and 8 digits, respectively. Graph 3 is a plot of the results from the three approaches, together with the exact solution. The plot of the results from all three approaches for $\alpha=\gamma=39.5$ (note $4\mu=40$) is indistinguishable from the exact solution, in Graph 3.

As the values of α and γ approach the value of 4μ, very few bilinear elements and Laguerre series coefficients were used in the calculations. In addition, the accuracy of the computed results improved as the values of α and γ approached the value of 4μ.

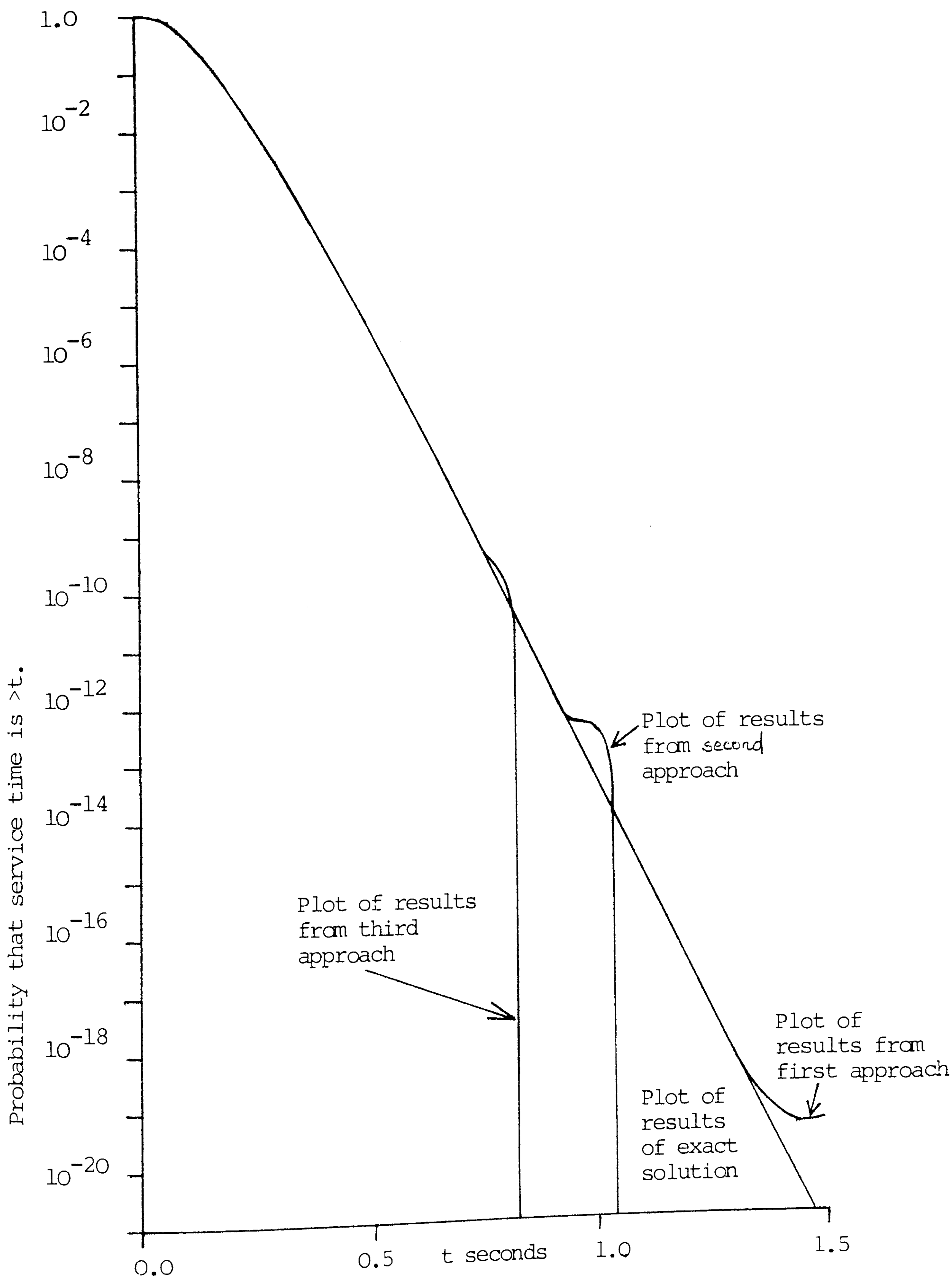
The digital filter designs based on the rectangular integration rule produced results with poor accuracy. The digital filter designs based on the trapezoidal integration rule were used in the computation of the function



Graph 1. Plot of service time function E_4 computed by the three approaches, for service rate $\mu = \frac{10}{9}$ and $\alpha=\gamma=\lambda=1$.



Graph 2. Plot of service time function E_4 computed by the three approaches for service rate $\mu=10$ and $\alpha=\gamma=5$.



Graph 3. Plot of service time function E_4 computed from the three approaches, for service rate $\mu=10$ and $\alpha=\gamma=20$.

$x(t) = \mu\exp(-\mu t)$, from its bilinear and Poisson sequences. For T=0.001, the results computed from the bilinear sequence were accurate to about 5 digits for a service rate $\mu = \frac{10}{9}$, using Double Precision *6 on the Harris 500 computer. The results computed from the Poisson sequence via Laguerre series coefficients were accurate to about 3 digits. As T approaches zero, the accuracy of the computed results improves. For T=0.00001, the accuracy of the function computed from the bilinear sequence was about 10 digits using Double Precision *12 arithmetic.

To obtain the number of multiplications performed by use of the digital filters per computed sample, let the required sample be at $t=T_1$. T_1 must be an integer multiple of the sampling period T. From Figs. 15 and 16, it can be seen that for each value of $t=kT$, where $k=0,1,\ldots,\frac{T_1}{T}$, approximately 3N multiplications are required, where N is the number of elements used. Therefore to compute the sample at $t=T_1$ requires approximately $\frac{3NT_1}{T}$ multiplications. To minimise the approximation error, T must be reduced, thus increasing the ratio $\frac{T_1}{T}$. The method which uses the Laguerre polynomial functions requires about 6N and 3N multiplications to evaluate the function from the bilinear and Poisson elements, respectively, at $t=T_1$. The method gives better accuracy and also offers some computational savings over the digital filter method. However, the digital filter methods will not suffer from overflow and underflow problems which the Laguerre series expansion methods can suffer from.

4.4 CONCLUSIONS

Several digital filter methods for computing the bilinear and Poisson sequences have been presented. The methods were derived by means of the impulse invariant and the numerical integration design procedures. The digital filter designs based on the trapezoidal integration rule produced results with better accuracy than the results obtained by the other design procedures. Its implementation was simple compared to the impulse invariant design which required the computation of the coefficients $\theta(n,m)$ and $\varphi(n,m)$ which, in turn, required the evaluation of factorials.

Two methods for the inversion of the bilinear and Poisson transforms were presented. It was found to be easier to invert the bilinear transform than the Poisson transform. By a careful choice of the bilinear and Poisson parameters α and γ, the method based on the Laguerre polynomial expansions was capable of achieving very high accuracy. The accuracy obtained by use of the digital filters was reasonable, but the accuracy of the first method was higher.

As most continuous-time probability functions in queue analysis are derived in the form of Laplace transform expressions, the computational methods presented are therefore useful in the inversion of these expressions.

The Poisson transform has applications in the analysis of the $M|G|1$ queue as well. The methods for computing the Poisson sequences and their inverses are thus useful in computing the various distribution functions for this queueing system.

CHAPTER 5

COMPUTATION OF VARIOUS PROBABILITY DISTRIBUTIONS IN QUEUE ANALYSIS

5.1 INTRODUCTION

Some of the problems faced by a computer communication system analyst involve performance evaluation. Queueing theory provides the basic mathematical tools for solving these problems. However, the results obtained by the application of queueing theory must further be solved by numerical methods to obtain the actual data required in the system specifications and performance evaluation. A problem in the solution of queueing problems is that of inverting the Laplace and z-transforms. This has been dealt with in the previous chapters. A systematic approach for solving queueing problems by use of digital signal processing methods has been presented. However, specific problems will require certain techniques.

In section 5.2, Skinner's method(49) is used to obtain upper and lower bounds on the waiting time distribution for the $G|G|1$ queue. Bounds for the queues $M|M|1$ and $E_2|M|1$ are derived. In section 5.3, a function $x(t)$ is computed, from an irrational Laplace transform expression, by use of the bilinear transform and digital signal processing techniques.

The bilinear and Poisson transforms of the probability density function of the busy period of the $M|G|1$ queue are derived in section 5.4. These transforms are obtained as functions of the bilinear and Poisson transforms of the service time density and the transforms themselves. A procedure for computing the busy period distribution, from these transforms and digital signal processing techniques, is presented. The procedure is used in the computation of the busy period distribution for the $M|M|1$ and $M|E_4|1$ queues. The exact solution for the $M|M|1$ queue presents some computational problems. However, this exact solution is rearranged into a set of recurrence relations which minimise the computational problems.

In section 5.5, the discrete probability density function for the number of customers served in the busy period, for the $M|G|1$ queue, is computed recursively. Section 5.6 discusses the application of some of the computational methods to the analysis of computer communication systems.

5.2 SKINNER'S METHOD FOR COMPUTING BOUNDS ON DISTRIBUTIONS

Some cumulative distribution functions are obtained from the convolution of probability density functions[2,23]. Skinner's method[49] provides a means of computing, numerically, upper and lower bounds on such cumulative distribution functions. By use of this method, approximate numerical results, whose accuracy is known precisely, can be computed. Skinner used his method to compute the upper and lower bounds on the waiting time distribution in the M|G|1 queue. However, in references 50 and 51, it was shown that the method can be applied to the computation of the numerical solutions of other problems as well.

Skinner's method can be used to evaluate numerically, terms of the form[49,51]

$$\int_{-\infty}^{t} x^{(n)}(\tau)\, d\tau , \qquad n=0,1,2,\ldots \tag{1}$$

that is, the integral of an n-fold self convolution of a probability density function. The method then enables bounds on terms like Eqn (1) to be obtained by the computation of discrete convolutions. The lower and upper bounds on Eqn (1) are obtained as follows. The cumulative distribution function corresponding to the density $x(t)$ is $X(t) = \int_{-\infty}^{t} x(\tau)d\tau$. Lower and upper bounds on $X(t)$ are obtained by sampling it and forming piecewise constant functions. The lower bound function $\underset{\sim}{X}(t)$ takes the value of the sample $X(kT)$ in the time-interval $kT \leq t < (k+1)T$. The upper bound function $\tilde{X}(t)$ takes the value of the sample $X(kT)$ in the time interval $(k-1)T \leq t < kT$. This is the same as $\underset{\sim}{X}(t)$ except for a left shift of T. $\underset{\sim}{x}_k$, the discrete probability function corresponding to $\underset{\sim}{X}(t)$ is given by the forward difference

$$\underset{\sim}{x}_k = X(kT) - X[(k-1)T], \qquad k=\ldots,-1,0,1,\ldots$$

$$= \int_{(k-1)T}^{kT} x(t)\,dt \tag{2}$$

The discrete probability distribution corresponding to $\tilde{X}(t)$ is identical to $\underset{\sim}{x}_k$ except for a left shift of one place, i.e.,

$$\tilde{x}_k = \underset{\sim}{x}_{k+1} \tag{3}$$

The expression for the lower and the upper bounds on the cumulative distribution corresponding to the probability density function which is an n-fold self convolution of $x(t)$ is[51]

$$\sum_{m=-\infty}^{k} \underset{\sim}{x}_m^{(n)} \leq \int_{-\infty}^{t} x^{(n)}(\tau)\,d\tau \leq \sum_{m=-\infty}^{k} \tilde{x}_m^{(n)} \tag{4}$$

where $\underset{\sim}{x}_k^{(n)}$ and $\tilde{x}_k^{(n)}$ are the n-fold discrete self convolutions of the sequences $\underset{\sim}{x}_k$ and $\tilde{x}_k$, respectively. Because of

Eqn (3), $\tilde{x}_k^{(n)}$ is the same as $\underset{\sim}{x}_k^{(n)}$ except for a left shift of n places [51], i.e.,

$$\tilde{x}_k^{(n)} = \underset{\sim}{x}_{k+n}^{(n)} \tag{5}$$

Therefore in Eqn (4), the upper bound sequence $\tilde{x}_m^{(n)}$, can be replaced by the lower bound sequence $\underset{\sim}{x}_m^{(n)}$, but with the top limit of the summation as k+n instead of k.

If a function is given in a form similar to Eqn (1), and there is a need to evaluate it numerically, then Skinner's method is applied as follows:

(i) Obtain the cumulative distribution corresponding to the density x(t), analytically, and periodically sample it.

(ii) Difference the samples, to obtain the discrete probability distribution $\underset{\sim}{x}_m$ Eqn (2).

(iii) Compute the n-fold discrete self convolution of $\underset{\sim}{x}_m$, either directly or by use of the FFT algorithm.

(iv) Sum the first k elements of the convolution to obtain the lower bound for values of t in the range $kT \leq t < (k+1)T$.

(v) Sum the first k+n elements, to obtain the upper bound for values of t in the range $kT \leq t < (k+1)T$.

This method can be used to obtain the bounds on the waiting time distribution for the G|G|1 queue. The waiting time distribution function for this queue can be expressed as a convolution[2] and it is thus suitable for the application of Skinner's method.

The solution of the continuous-time G|G|1 queue is obtained by approximating it with that of the discrete-time G|G|1 queue[45], from

$$W_d(z) = \frac{1}{(1-z^{-1})\left[\frac{C_d(z)-1}{1-z^{-1}}\right]_+} \tag{6}$$

where $[.]_+$ denotes the process of taking the minimum phase function. $W_d(z)$ is the z-transform of the discrete waiting time distribution function. $C_d(z) = A_d(\frac{1}{z})B_d(z)$, where $A_d(z)$ and $B_d(z)$ are the z-transforms of the discrete densities of the interarrival time and the service time, respectively. The function $\frac{C_d(z)}{(1-z^{-1})}$ corresponds to a discrete cumulative distribution function and it is similar to Eqn (1). Skinner's method will therefore give bounds on the approximation to the continuous-time queue solution. There are several numerical methods which can be used to solve Eqn (6)[45,47].

Expressions for the bounds on the cumulative distribution of the waiting time for the G|G|1 queue are obtained as follows:

(i) Obtain an expression for c(t). c(t) is the convolution of the densities of the interarrival time and the service time, i.e. $c(t)=a(-t)*b(t)$.

(ii) Obtain the expression for the cumulative distribution C(t), corresponding to c(t).

(iii) Periodically sample the result to obtain $\underset{\sim}{C}(z)$, the lower bound on the cumulative distribution corresponding to c(t).

(iv) Obtain the upper bound function $\tilde{C}(z)$ from $\underset{\sim}{C}(z)$ by use of the equation $\tilde{C}(z) = z\underset{\sim}{C}(z)$.

(v) Solve Eqn (6), to obtain the expressions for the z-transforms of the lower and upper bounds, by replacing $\dfrac{C_d(z)}{(1-z^{-1})}$ by $\underset{\sim}{C}(z)$ and $\tilde{C}(z)$, respectively.

(vi) Invert the resulting z-transform expressions, to obtain the expressions for the lower and upper bounds on the cumulative distribution function of the waiting time.

For the M|M|1 queue, the probability density functions for the interarrival time and service time are respectively, $a(t) = \lambda e^{-\lambda t}$, and $b(t) = \mu e^{-\mu t}$. Using steps (i)-(vi), the expressions for $\underset{\sim}{W}(t)$ and $\tilde{W}(t)$, the lower and upper bounds on the waiting time distribution are, for $t \geq 0$

$$\underset{\sim}{W}(t) = 1-\left(\frac{\sigma_L-\beta}{1-\beta}\right)\sigma_L^k , \quad kT \leq t < (k+1)T \tag{7}$$

and

$$\tilde{W}(t) = 1-\left(\frac{\sigma_U-\beta}{1-\beta}\right)\sigma_U^k \quad , \quad kT\leq T<(k+1)T \tag{8}$$

respectively, where:

$$\beta = \exp(-\mu T)$$

$$\sigma_L = \frac{\lambda+\mu\exp[-(\lambda+\mu)T]}{(\lambda+\mu)\exp(-\lambda T)}$$

$$\sigma_U = \frac{(\mu+\lambda)\exp(-\mu T)}{\mu+\lambda\exp[-(\lambda+\mu)T]}$$

The exact solution for the waiting time distribution for the $M|M|1$ queue is given by

$$W(t) = 1-\frac{\lambda}{\mu}\exp[-(\mu-\lambda)t] \quad , \qquad t\geq 0$$

The $E_2|M|1$ queue in which the two arrival stages have different death rates which are linear multiples of the service rate[2] was also considered. The service time density is the same as for the $M|M|1$ queue. The interarrival time density is given by[2]

$$a(t) = 2\mu\exp(-\mu t)[1-\exp(-\mu t)] , \qquad t\geq 0$$

with rate $\lambda = \frac{2\mu}{3}$. For this system, the upper and lower bounds on the waiting time distribution are similar to

Eqns (7) and (8), but σ_U and σ_L are given by the equations

$$\sigma_U = \frac{3\beta}{(1+2\beta^3)+(1-\beta)\left[(1+\beta+4\beta^2)(1+\beta+\beta^2)\right]^{\frac{1}{2}}}$$

$$\sigma_L = \frac{(1+3\beta^2-\beta^3)}{(2+\beta^3)\beta+(1-\beta)\beta\left[(4+\beta+\beta^2)(1+\beta+\beta^2)\right]^{\frac{1}{2}}}$$

where $\beta = \exp(-\mu T)$. The exact solution for the waiting time distribution for this $E_2|M|1$ queue is[2]

$$W(t) = 1-(2-\sqrt{2})\exp\left[-\mu(\sqrt{2}-1)t\right], \quad t \geq 0.$$

It is worth pointing out that, in using Eqns (7) and (8) directly, to compute the bounds, there will be a loss of significance in the computed bounds as T tends to zero. To avoid this Eqns (7) and (8) can be rearranged or alternatively, extended precision arithmetic can be used.

Table 1 displays the upper and lower bounds for the cumulative distribution of the waiting time for the $M|M|1$ queue, together with the exact solution. The results were computed with $\lambda=1$, $\mu=1.5$, for three values of the sampling period $T=0.1$, $T=0.01$, and $T=0.001$. Table 2 displays the results for the $E_2|M|1$ queue for the same values of λ, μ, and T. The bounds approach the exact solution as the magnitude of T tends to zero.

TABLE 1 Upper and Lower Bounds on the Waiting Time Distribution Function for the M|M|1 Queue with $\lambda=1$ and $\mu=1.5$

T = 0.1

t	$\tilde{W}(t)$	$W(t)$	$\underset{\sim}{W}(t)$
0	0.400200	0.333333	0.297984
1	0.662012	0.595646	0.540551
2	0.809543	0.754747	0.699304
3	0.892677	0.851247	0.803204
4	0.939524	0.909776	0.871203
5	0.965921	0.945277	0.915706
6	0.980797	0.966809	0.944832
7	0.989179	0.979868	0.963894
8	0.993902	0.987790	0.976370
9	0.996564	0.992594	0.984535
10	0.998064	0.995508	0.989878

T=0.01

t	$\tilde{W}(t)$	$W(t)$	$\underset{\sim}{W}(t)$
0	0.340005	0.333333	0.329980
1	0.602679	0.595646	0.590548
2	0.760810	0.754747	0.749782
3	0.856006	0.851247	0.847091
4	0.913315	0.909776	0.906557
5	0.947815	0.945277	0.942896
6	0.968584	0.966809	0.965104
7	0.981088	0.979868	0.978675
8	0.988615	0.987790	0.986968
9	0.993146	0.992594	0.992036
10	0.995874	0.995508	0.995133

T=0.001

t	$\tilde{W}(t)$	$W(t)$	$\underset{\sim}{W}(t)$
0	0.334000	0.333333	0.333000
1	0.596353	0.595646	0.595140
2	0.755359	0.754747	0.754256
3	0.851729	0.851247	0.850837
4	0.910137	0.909776	0.909460
5	0.945536	0.945277	0.945044
6	0.966991	0.966809	0.966642
7	0.979994	0.979868	0.979752
8	0.987875	0.987790	0.987710
9	0.992651	0.992594	0.992540
10	0.995546	0.995508	0.995472

TABLE 2

Upper and Lower Bounds on the Waiting Time

Distribution Function for the $E_2|M|1$ Queue

T=0.1

t	$\tilde{W}(t)$	$W(t)$	$\underset{\sim}{W}(t)$
0	0.483802	0.414214	0.376520
1	0.743068	0.685295	0.636201
2	0.872115	0.830930	0.787725
3	0.936346	0.909170	0.876138
4	0.968317	0.951203	0.927727
5	0.984230	0.973784	0.957829
6	0.992151	0.985916	0.975393
7	0.996093	0.992434	0.985642
8	0.998055	0.995935	0.991622
9	0.999032	0.997816	0.995112
10	0.999518	0.998827	0.997148

T=0.01

t	$\tilde{W}(t)$	$W(t)$	$\underset{\sim}{W}(t)$
0	0.421323	0.414214	0.410703
1	0.691568	0.685295	0.680871
2	0.835608	0.830930	0.827178
3	0.912380	0.909170	0.906410
4	0.953299	0.951203	0.949317
5	0.975109	0.973784	0.972553
6	0.986733	0.985916	0.985136
7	0.992929	0.992434	0.991951
8	0.996231	0.995935	0.995641
9	0.997991	0.997816	0.997639
10	0.998929	0.998827	0.998722

T=0.001

t	$\tilde{W}(t)$	$W(t)$	$\underset{\sim}{W}(t)$
0	0.414926	0.414214	0.413865
1	0.685928	0.685295	0.684857
2	0.831404	0.830930	0.830560
3	0.909496	0.909170	0.908898
4	0.951417	0.951203	0.951018
5	0.973920	0.973784	0.973664
6	0.986000	0.985916	0.985840
7	0.992485	0.992434	0.992387
8	0.995966	0.995935	0.995907
9	0.997837	0.998816	0.997799
10	0.998837	0.998827	0.998817

$\lambda=1.0$ and $\mu=1.5$

5.3 DISTRIBUTION WITH IRRATIONAL LAPLACE TRANSFORM

Occasionally, one is faced with the task of inverting transcendental Laplace transform expressions. If a probability function is specified in this form, one can be interested in computing either the inverse Laplace transform or the bilinear and Poisson sequences, for the purpose of numerical calculations. This is therefore a problem of computing the bilinear or Poisson sequence from the Laplace transform expression. If the actual function x(t) is required, it can be computed from the resulting bilinear or Poisson sequence.

To illustrate how these transcendental functions can be inverted, the Laplace transform

$$X_L(s) = \exp(\beta\sqrt{\theta})\ \exp(-\beta\sqrt{s+\theta}) \tag{9}$$

was used as an example. The inversion of the Poisson transform poses some problems because it must be transformed to other sequences like the Laguerre series coefficients or the bilinear sequence , therefore this transform was not used in this example. To obtain the bilinear transform, s is replaced by $\frac{\alpha(1-z^{-1})}{(1+z^{-1})}$. The bilinear transform of x(t) from Eqn (9) is then

$$X_b(z)=X_L\left[\frac{\alpha(1-z^{-1})}{1+z^{-1}}\right] = \exp(\beta\sqrt{\theta})\exp\left[-\beta\sqrt{\frac{\theta+\alpha+(\theta-\alpha)z^{-1}}{1+z^{-1}}}\right] \tag{10}$$

To invert $X_b(z)$, in order to get the bilinear sequence $x_b(n)$, is not easy. One therefore resorts to numerical methods.

The FFT algorithm can be used to invert this equation by use of the transformation $z = \exp(j\frac{2\pi m}{L})$. Using this transformation and De Moivre's theorem[86] in Eqn (10), the DFT of the bilinear sequence is obtained as

$$X_b\left[\exp(j\frac{2\pi m}{L})\right] = \exp(\beta\sqrt{\theta})\, r_m(\varphi_m + j\sigma_m), \quad m=0,1,\ldots,L-1 \tag{11}$$

where:

$$r_m = \exp\{-c_m \cos\left[\tfrac{1}{2}\tan^{-1}(\tfrac{w_m}{\theta})\right]\}$$

$$\varphi_m = \cos\{c_m \sin\left[\tfrac{1}{2}\tan^{-1}(\tfrac{w_m}{\theta})\right]\}$$

$$\sigma_m = -\sin\{c_m \sin\left[\tfrac{1}{2}\tan^{-1}(\tfrac{w_m}{\theta})\right]\}$$

$$c_m = \beta\left[\theta^2 + w_m^2\right]^{\frac{1}{4}}$$

$$w_m = \frac{\alpha \sin(\frac{2\pi m}{L})}{1+\cos(\frac{2\pi m}{L})}$$

The inverse DFT of Eqn (11) gives the bilinear sequence $x_b(n)$. In the absence of an exact expression for this sequence, it is difficult to assess the accuracy of the sequence computed by this method.

However, it is possible to obtain an expression for the function x(t) from Eqn (9), by use of entries in tables given in reference 87 and the properties of the Laplace transform. The probability density function x(t), is given by

$$x(t) = \beta\frac{\exp(\beta\sqrt{\theta})}{2\sqrt{\pi t^3}} \exp\left[-\left(\theta t + \frac{\beta^2}{4t}\right)\right], \quad t \geq 0 \tag{12}$$

The method which makes use of the Laguerre polynomial functions in chapter 4, for inverting the bilinear transform, was used. The function x(t) was computed from the sequence $x_b(n)$ obtained as the inverse DFT of Eqn (11). Double Precision *6 arithmetic, which has an accuracy of 10 digits, was used. The computed function x(t) was compared with results from a direct evaluation of Eqn (12). For $\theta=1$ and $\theta=15$ and FFT block sizes of 2048 and 4096, the accuracy of the computed function was at least 6 digits.

This example demonstrates how to invert transcendental Laplace transform functions, which may arise in connection with more complicated queueing systems. Chapter 4 dealt with the problem of computing the bilinear and Poisson sequences from the time functions. This example also shows how these sequences can be computed from Laplace transforms.

5.4 BUSY PERIOD DISTRIBUTION FOR M|G|1 QUEUE

This section deals with the computation of the busy period distribution for the M|G|1 queue by use of the bilinear and Poisson transforms. The section is divided into four parts, the first part describes the derivation of the bilinear and Poisson transforms of the busy period density. The second part outlines the computational procedure. The third and last parts discuss the solution for the M|M|1 queue and the performance of the computational method, respectively.

5.4.1 BILINEAR AND POISSON TRANSFORMS OF BUSY PERIOD DENSITY

A busy period of a queueing system commences when a customer arrives to find the system empty and the server free to serve him immediately. It ends when the server completes the service of a customer, leaving the system empty again. The Laplace transform of the probability density function of the busy period for the bulk arrival $M^X|G|1$ queue is given in reference 88, as

$$G_L(s) = B_L\{s-\lambda+\lambda X_L[\log G_L(s)]\} \tag{13}$$

where $B_L(s)$ is the Laplace transform of the service time density. For this queue, arrivals are Poisson and they are in groups of random size. $X_L(s)$ is the Laplace transform of the discrete distribution of the group size.

When the group size is unity, $X_L(s) = e^s$ and $G_L(s)$ reduces to[2,3]

$$G_L(s) = B_L[s-\lambda+\lambda G_L(s)] \tag{14}$$

which is the equation for the Laplace transform of the busy period density for the M|G|1 queue. The solution of this equation is known for only a few cases in particular, for the M|M|1 and M|D|1 queues. The numerical solution of this equation is therefore desirable.

It is suggested in reference 2 that Eqn (14) can be solved numerically through the iterative equation

$$G_{L,n+1}(s) = B_L[s+\lambda-\lambda G_{L,n}(s)] \tag{15}$$

The limit of this iterative scheme as $n\to\infty$ converges to $G_L(s)$ for a stable queue. Eqn (15) is in a form suitable for the application of the bilinear and the Poisson transform methods in Chapters 3 and 4. By use of these transforms, the iterations can be performed on the bilinear and Poisson sequences of $g_{n+1}(t)$, the busy period density, instead of the Laplace transform $G_{L,n+1}(s)$.

To use the bilinear transform to solve Eqn (15), numerically, s is replaced by $\frac{\alpha(1-z^{-1})}{(1+z^{-1})}$ to obtain the bilinear transform of the busy period density, at the (n+1)th iteration. The bilinear transform of the busy period density from Eqn (15) is then

$$G_{b,n+1}(z) \triangleq G_{L,n+1}\left[\frac{\alpha(1-z^{-1})}{(1+z^{-1})}\right]=B_L\left[\frac{\alpha(1-z^{-1})}{(1+z^{-1})}+\lambda-\lambda G_{b,n}(z)\right] \tag{16}$$

If the Laplace transform of the service time density can be obtained, then Eqn (16) is solved to get $G_{b,n+1}(z)$. If instead, the expression for the service time density b(t), or its samples are available, then the bilinear sequence of b(t) can be used to obtain $G_{b,n+1}(z)$. The bilinear sequence of the service time density can be obtained from the function or its samples, by use of the digital filter methods in chapter 4. For this situation, it is necessary to express the bilinear transform $G_{b,n+1}(z)$ as a function of the bilinear transform of the service time density, i.e., as a function of $B_b(z)$.

To do this, one proceeds as follows. The bilinear transform of the service time probability density function is by definition $B_b(z) = B_L\left[\frac{\alpha(1-z^{-1})}{1+z^{-1}}\right]$. To find $G_{b,n+1}(z)$ in terms of $B_b(z)$, Eqn (16) must be expressed in the form

$$G_{b,n+1}(z)=B_L\left[\frac{\alpha\left[1-(F_n(z))^{-1}\right]}{\left[1+(F_n(z))^{-1}\right]}\right]=B_b\left[F_n(z)\right] \tag{17}$$

Using Eqns(17) and (16), the terms within the brackets of $B_L[\cdot]$ in these equations are equated. Solving the resulting equation, $(F_n(z))^{-1}$ is

$$(F_n(z))^{-1} = \frac{-[\lambda+(\lambda-2\alpha)z^{-1}]+\lambda(1+z^{-1})G_{b,n}(z)}{(\lambda+2\alpha)+\lambda z^{-1}-\lambda(1+z^{-1})G_{b,n}(z)} \tag{18}$$

The bilinear transform of the busy period probability density function, at the (n+1)th iteration, is then given by

$$\begin{aligned} G_{b,n+1}(z) &= B_b[F_n(z)] \\ &= \sum_{k=0}^{\infty} b_b(k)[F_n(z)]^{-k} \end{aligned} \tag{19}$$

where $b_b(k)$ is the bilinear sequence of the service time density. Because the elements of the bilinear sequence $b_b(k)$ tend toward zero for large k, the summation in Eqn (19) can be truncated after say, M terms. The iterations are then performed on the bilinear sequence of the busy period density, instead of its Laplace transform, by use of Eqns (19) and (18).

Replacing s by $\gamma(1-z^{-1})$ in Eqn (15) and proceeding in a similar manner, the Poisson transform of the busy period density is obtained from the Poisson transform of the service time density. The Poisson transform of the busy period density at the (n+1)th iteration is therefore

$$G_{p,n+1}(z) = B_p[F_n(z)] = \sum_{k=0}^{\infty} b_p(k)[F_n(z)]^{-k} \tag{20}$$

where

$$[F_n(z)]^{-1} = -\frac{\gamma}{\lambda} + \frac{\gamma}{\lambda} z^{-1} + G_{p,n}(z)$$

The sequence $b_p(k)$ is the Poisson sequence of the service time density $b(t)$. For the $M|G|1$ queue, $b_p(k)$ corresponds to the probability of k customers arriving in a service period when $\gamma=\lambda$. By use of Eqn (20), the Poisson transform of the busy period density can be computed, iteratively, from the Poisson sequence of the service time probability density.

5.4.2 COMPUTATIONAL PROCEDURE

The busy period distribution is computed iteratively by use of Eqns (19) and (20). Because of the uniqueness of the bilinear and Poisson sequences of functions, when these sequences converge and the inverse bilinear and Poisson transforms obtained, the result is the busy period density.

For practical purposes, the cumulative distribution function is more useful than the density function. It is easier to compute the survivor function (complementary cumulative distribution) of the busy period. This is given by $h(t) = 1-G(t)$ where $G(t)$ is the cumulative distribution function corresponding to the busy period density $g(t)$. The Laplace transform $H_L(s)$, of $h(t)$ is

$H_L(s) = \frac{(1-G_L(s))}{s}$. The bilinear and the Poisson sequences of the survivor function h(t) are obtained from the steady-state bilinear and Poisson sequences of the busy period density as follows: Let $g_b(k)$ and $g_p(k)$ be the steady-state bilinear and Poisson sequences of the busy period density g(t), and $h_b(k)$ and $h_p(k)$ those of the function h(t), respectively. The sequences $h_b(k)$ and $h_p(k)$ can be obtained by passing the sequences $\{1-g_b(o), -g_b(1), -g_b(2), \ldots\}$ and $\{1-g_p(o), -g_p(1), -g_p(2), \ldots\}$ through digital filters with transfer functions $\frac{(1+z^{-1})}{\alpha(1-z^{-1})}$ and $\frac{1}{\gamma(1-z^{-1})}$, respectively.

To compute the bilinear and Poisson sequences $g_{b,n+1}(k)$ and $g_{p,n+1}(k)$ using Eqns (19) and (20), the FFT algorithm is used. Using Eqns (18) and (19) and then replacing z by $\exp(j\frac{2\pi m}{L})$, the DFT of the bilinear sequence $g_{b,n+1}(k)$ is, after using De Moivre's theorem

$$G_{b,n+1}\left[\exp(j\frac{2\pi m}{L})\right] = \sum_{k=0}^{\infty} b_b(k) r_{n,m}^k \exp(jk\theta_{n,m})$$

$$= \varphi_{n+1,m} + j\sigma_{n+1,m}, \quad m=0,1,\ldots,L-1 \tag{21}$$

where

$$r_{n,m} = \left[\frac{c_{n,m}^2 + d_{n,m}^2}{e_{n,m}^2 + f_{n,m}^2}\right]^{\frac{1}{2}}$$

$$\theta_{n,m} = \tan^{-1}\left[\frac{d_{n,m}}{c_{n,m}}\right] - \tan^{-1}\left[\frac{f_{n,m}}{e_{n,m}}\right]$$

$$f_{n,m} = -(1-\phi_{n,m})\sin(\frac{2\pi m}{L})-\lambda\sigma_{n,m}[1+\cos(\frac{2\pi m}{L})]$$

$$e_{n,m} = \lambda(1-\phi_{n,m})\cos(\frac{2\pi m}{L})+(2\alpha+\lambda-\lambda\phi_{n,m})-\lambda\sigma_{n,m}\sin(\frac{2\pi m}{L})$$

$$d_{n,m} = -(2\alpha+\lambda\phi_{n,m}-\lambda)\sin(\frac{2\pi m}{L})+\lambda\sigma_{n,m}[1+\cos(\frac{2\pi m}{L})]$$

$$c_{n,m} = (2\alpha+\lambda\phi_{n,m}-\lambda)\cos(\frac{2\pi m}{L})+\lambda[\phi_{n,m}-1+\sigma_{n,m}\sin(\frac{2\pi m}{L})]$$

$$\phi_{n,m} = \mathrm{Re}\ \{G_{b,n}\ [\exp(j\frac{2\pi m}{L})]\}$$

$$\sigma_{n,m} = \mathrm{Im}\ \{G_{b,n}[\exp(j\frac{2\pi m}{L})]\}$$

The inverse DFT of Eqn (21) gives the bilinear sequence of the busy period probability density function at iteration (n+1).

Replacing z by $\exp(j\frac{2\pi m}{L})$ in Eqn (20), the DFT of the Poisson sequence of the busy period probability density function at iteration n+1 is obtained. The DFT is

$$G_{p,n+1}[\exp(j\frac{2\pi m}{L})]= \sum_{k=o}^{\infty} b_p(k)r_{n,m}^k \exp[jk\theta_{n,m}]$$

$$= \phi_{n+1,m} + j\sigma_{n+1,m}\ , \qquad m=0,1,\ldots,L-1 \tag{22}$$

where

$$r_{n,m}=\{\phi_{n,m}^2 + \sigma_{n,m}^2 - 2\frac{\gamma}{\lambda}[\phi_{n,m}-(\phi_{n,m}-\frac{\gamma}{\lambda})\cos(\frac{2\pi m}{L}) + \sigma_{n,m}\sin(\frac{2\pi m}{L})-\frac{\gamma}{\lambda}]\}^{\frac{1}{2}}$$

$$\theta_{n,m} = \tan^{-1}\left[\frac{\sigma_{n,m} - \frac{\gamma}{\lambda}\sin(\frac{2\pi m}{L})}{\varphi_{n,m} - \frac{\gamma}{\lambda} + \frac{\gamma}{\lambda}\cos(\frac{2\pi m}{L})}\right]$$

$$\varphi_{n,m} = \mathrm{Re}\{G_{p,n}[\exp(j\frac{2\pi m}{L})]\}$$

$$\sigma_{n,m} = \mathrm{Im}\{G_{p,n}[\exp(j\frac{2\pi m}{L})]\}$$

The inverse DFT of Eqn (22) gives the Poisson sequence of the busy period probability density function at the (n+1)th iteration.

From experience with the computational procedure, it is necessary to make a correction for the phase $\theta_{n,m}$ in Eqns(21) and (22). In computing this phase, when its sign becomes reversed, it is necessary to add the value π to the computed value. This is because the phase is computed modulo 2π.

An idle system, that is, a system whose busy period is given by $g_o(t) = \delta(t)$, can be used as the initial state. The cumulative distribution of the busy period for the M|G|1 queue is computed from the service time probability density function by the following process.

(i) Compute the bilinear or Poisson sequence of the service time density b(t). This is done by replacing s by $\frac{\alpha(1-z^{-1})}{(1+z^{-1})}$ or $\gamma(1-z^{-1})$ in the expression for $B_L(s)$. The inverse z-transform of the result gives the required sequence. These sequences can also be computed from b(t) by use of the methods in chapter 4.

(ii) Compute the DFT of the bilinear or Poisson sequence of the busy period density of an idle system, i.e., the DFT of $g_{b,o}(k)=g_{p,o}(k)=\delta(k)$, this is $\phi_{o,m}+j\sigma_{o,m}=1$, $m=0,1,\ldots,L-1$.

(iii) Using Eqn (21) or Eqn (22), compute $G_{b,n+1}[\exp(j\frac{2\pi m}{L})]$ or $G_{p,n+1}[\exp(j\frac{2\pi m}{L})]$, the bilinear or Poisson transform of $g_{n+1}(t)$.

(iv) Compute the inverse DFT,to obtain the bilinear or Poisson sequence,$g_{b,n+1}(k)$ or $g_{p,n+1}(k)$.

(v) Repeat (iii) - (v) until the sequence $g_{b,n+1}(k)$ or $g_{p,n+1}(k)$ converges to $g_b(k)$ or $g_p(k)$.

(vi) Compute the bilinear or Poisson sequence of h(t), the survivor function of the busy period, from the sequence $g_b(k)$ or $g_p(k)$. This is conveniently done by passing the sequences $\{1-g_b(o),-g_b(1),-g_b(2),\ldots\}$ and $\{1-g_p(0),-g_p(1),-g_p(2),\ldots\}$ through digital filters with transfer functions $\frac{(1+z^{-1})}{\alpha(1-z^{-1})}$ and $\frac{1}{\gamma(1-z^{-1})}$,respectively.

(vii) Compute the survivor function h(t) from its bilinear or Poisson elements by use of the methods in chapter 4.

(viii) Obtain the cumulative distribution of the busy period, i.e., $G(t)= 1-h(t)$.

There is no reason why a similar procedure cannot be obtained to solve Eqn (13), the Laplace transform of the busy period density for the bulk arrival $M^X|G|1$ queue.

An alternative method to this procedure is outlined in reference 46. In this method, the solution for the discrete $M|G|1$ queue is used to approximate the solution for the continuous $M|G|1$ queue. The busy period distribution is computed by use of the FFT algorithm. The method computes the approximate results as upper and lower bounds, which are more useful than having approximate results of unknown accuracy.

5.4.3 SOLUTION FOR M|M|1 QUEUE

Solutions for the $M|M|1$ and $M|D|1$ queues are available[2]. For the $M|D|1$ queue, the service times and busy periods are discrete functions. The busy period distribution functions for systems with discrete service times are discussed in references 46, 88 and 89.

The busy period probability density function for the $M|M|1$ queue is given by[2]

$$g(t)= \frac{1}{t\sqrt{\rho}} \exp[-(\lambda+\mu)t] \sum_{m=0}^{\infty} \frac{[t\sqrt{\lambda\mu}]^{2m+1}}{(m+1)(m!)^2}, \quad t\geq 0 \tag{23}$$

The summation corresponds to the modified Bessel function $I_1[2t\sqrt{\lambda\mu}]$. The summation is obtained by use of information in reference 90, p 12. The cumulative distribution of the busy period is then

$$G(t) = \frac{1}{\sqrt{\rho}} \sum_{m=0}^{\infty} \left[\left(\frac{\sqrt{\lambda\mu}}{\lambda+\mu}\right)^{2m+1} \frac{(2m)!}{(m+1)(m!)^2} \right] \left[1-\exp(-(\lambda+\mu)t) \times \sum_{k=0}^{2m} \frac{[t(\lambda+\mu)]^k}{k!} \right], \quad t \geq 0 \qquad (24)$$

It is worth noting that the computation of the exact solution for the busy period distribution Eqn (24) will present computational problems, because of the evaluation of factorials. To avoid the evaluation of the large number of factorials, Eqn (24) can be rearranged to

$$G(t) = \frac{1}{\sqrt{\rho}} \sum_{m=0}^{\infty} [\alpha_m - \beta_m(t)], \quad t \geq 0 \qquad (25)$$

where α_m and $\beta_m(t)$ are obtained from the following set of recursive relations:

$$\alpha_{m+1} = \frac{(2m+1)(2m+2)}{(m+1)(m+2)} \left(\frac{\sqrt{\lambda\mu}}{\lambda+\mu}\right)^2 \alpha_m, \qquad m=0,1,2,\ldots \qquad (26)$$

where $\alpha_0 = \frac{\sqrt{\lambda\mu}}{\lambda+\mu}$

$$\beta_{m+1}(t) = \frac{(2m+1)(2m+2)}{(m+1)(m+2)}\left(\frac{\sqrt{\lambda\mu}}{\lambda+\mu}\right)^2 \beta_m(t)+\gamma_{m+1}(t)\alpha_{m+1}, \quad m=0,1,2,\ldots \tag{27}$$

where $\beta_o(t) = \alpha_o\gamma_o(t)$

and

$$\gamma_{m+1}(t) = \frac{[t(\lambda+\mu)]^2}{(2m+1)2m+2)}\left[\frac{2m+2+t(\lambda+\mu)}{2m+t(\lambda+\mu)}\right]\gamma_m(t), \quad m=0,1,2,\ldots \tag{28}$$

where $\gamma_o(t) = \exp[-(\lambda+\mu)t]$

If G(t) is computed from Eqn (24) or (25), the term $\exp[-(\lambda+\mu)\ t]$ can cause underflow problems which can result in errors in the computed G(t). Underflow problems are most likely to occur for large values of $(\lambda+\mu)t$. The evaluation of factorials in Eqn (24) can result in overflow problems.

The G(t) computed from Eqn (25) partly depends on the factor $\gamma_{m+1}(t)$ in Eqn (28). This factor is a product of the term

$$\frac{[t(\lambda+\mu)]^{2m}}{(2m+1)!(2m+2)!}\quad \exp[-(\lambda+\mu)t] \tag{29}$$

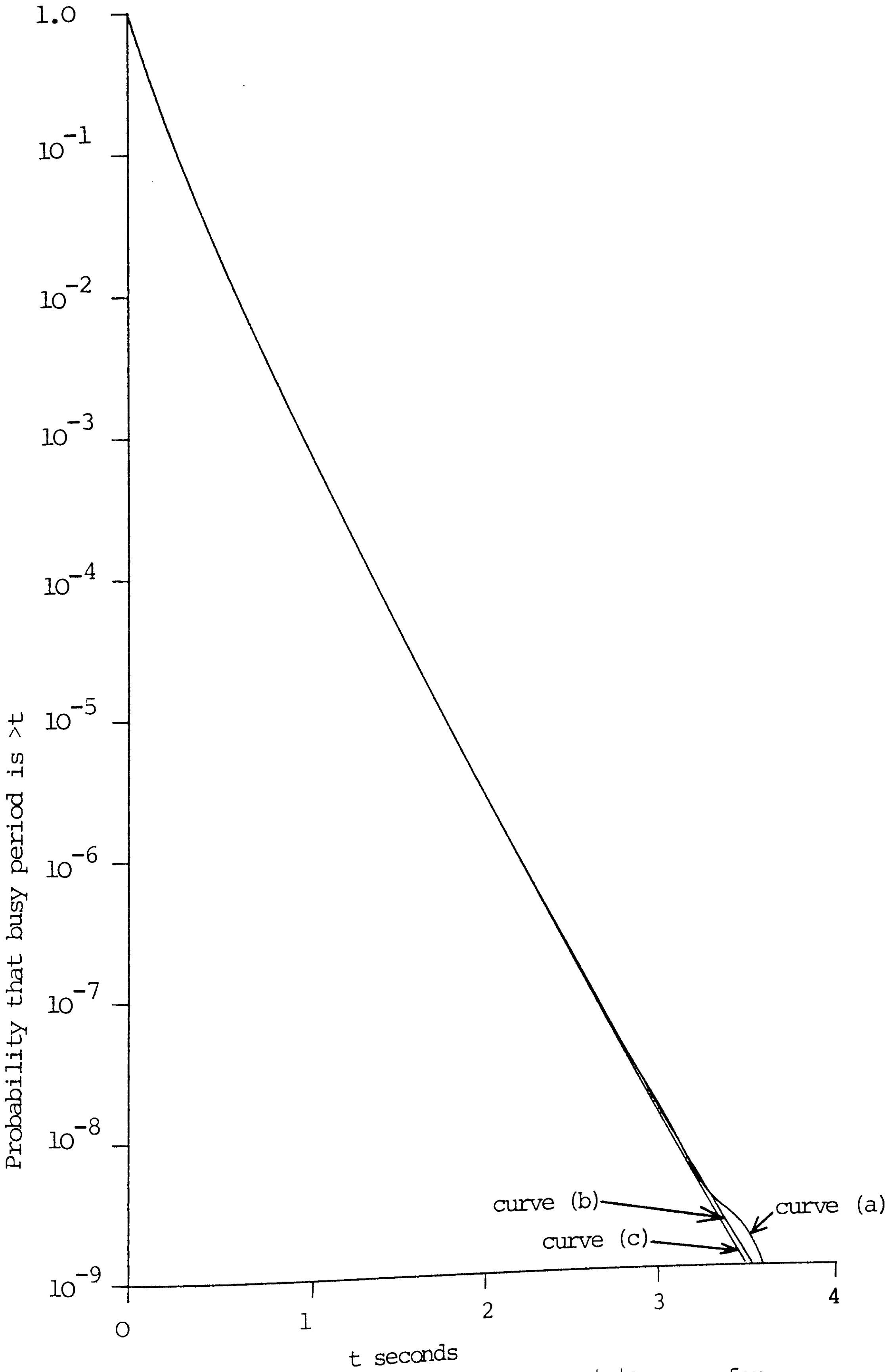
and another term. The factor $\gamma_{m+1}(t)$ can have a significant value for large values of t and m. But for large values of $(\lambda+\mu)t$, the term $\exp[-(\lambda+\mu)t]$ can result in underflow thus zeroing the term $\gamma_{m+1}(t)$. This results in an error in the value of the computed factor $\beta_{m+1}(t)$ in Eqn (27) and hence in G(t) in Eqn (25). Underflow problems can be avoided by computing instead, the natural logarithms of $\gamma_o(t)$ (which is $[-(\lambda+\mu)t]$) and the successive $\gamma_m(t)$'s, through Eqn (28). The factor $\gamma_{m+1}(t)$ is then obtained by taking the exponential of the computed logarithm.

5.4.4 ACCURACY OF COMPUTATIONAL PROCEDURE

The busy period lasts for very long periods for high utilizations. This then means that the busy period probability density function can be significant for very large values of t. It was mentioned in chapter 4 that the inversion of the bilinear and Poisson transforms by use of the Laguerre polynomial series can suffer from underflow and overflow problems if the function which is being computed tends to zero very slowly as t is increased. These problems can therefore be expected in computing the busy period for $\lambda=1$ and $\rho\to 1$. The chances of these problems occurring can be minimised by a suitable choice of the bilinear and Poisson parameters, α and γ, respectively. This is at the expense of, possibly, using more bilinear and Laguerre series elements in the series expansions.

The busy period distribution for the M|M|1 queue was computed by the computational procedure outlined and the inversion method in Chapter 4,section 4.3.1, for $\lambda=1$. There were no computational problems encountered in computing the distribution $g(t)$ by use of this procedure or by use of Eqn(25), for the utilization factor $\rho=0.1$. For $\mu=10$ and $\alpha=9.999$, about 8 bilinear elements of the service time density resulted in about 30 significant bilinear elements of the busy period density, using Double Precision *6 arithmetic with an accuracy of 10 digits. The number of iterations performed, before the convergence criterion $|g_{b,n+1}(o) - g_{b,n}(o)| \leq 10^{-9}$ was satisfied, was about 7. The computed results for the function $G(t)$ obtained by the computational procedure and from Eqn (25) agreed to about 8 digits. For the same queue, with same values of λ and μ, and the same convergence criterion, and $\alpha=1$, about 170 bilinear elements of the service time density were used. The FFT block size for the bilinear elements of the busy period density was 256. 9 iterations were performed and the accuracy achieved was about 8 digits.

Graph 1 is a plot of the computed busy period function for both values of α. Also plotted in the same graph are the results from the exact solution Eqn (25). For utilization factor $\rho=0.1$, the value of the bilinear parameter $\alpha=9.999$ results in much fewer significant bilinear elements of both the service time and the busy period densities. Fewer iterations are also performed. As



Graph 1 Plot of the busy period function for the M|M|1 queue for μ=10 and λ=1. Curve (a) is the function computed with α=1.0. Curve (b) is the exact solution from Eqn (25). Curve (c) is the function computed with α=9.999.

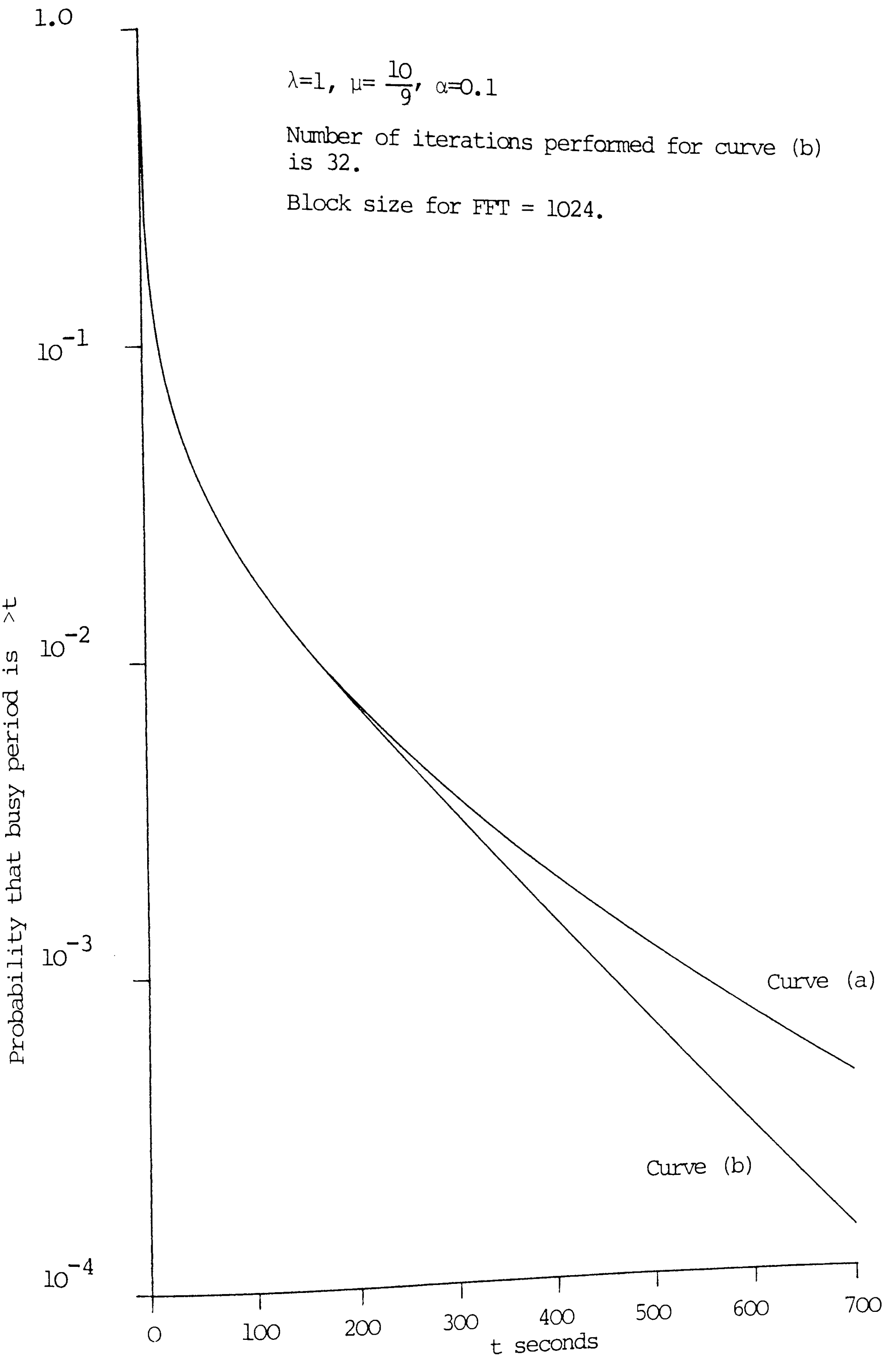
mentioned before, there is a need for experimenting with several values of the bilinear parameter α.

Graph 2 is a plot of the complementary function corresponding to the busy period distribution for the $M|M|1$ queue with $\lambda=1$ and $\mu=\frac{10}{9}$. For these values of λ and μ, the busy period lasts for several hundreds of seconds of time. The value of α, the bilinear parameter, used was 0.1. The accuracy of the computed results compared to an evaluation of the solution Eqn (25) was at least 4 digits. In Graph 3, the computed busy period functions for the $M|M|1$ and $M|E_4|1$ queues are plotted together for $\lambda=1$ and $\mu=10$.

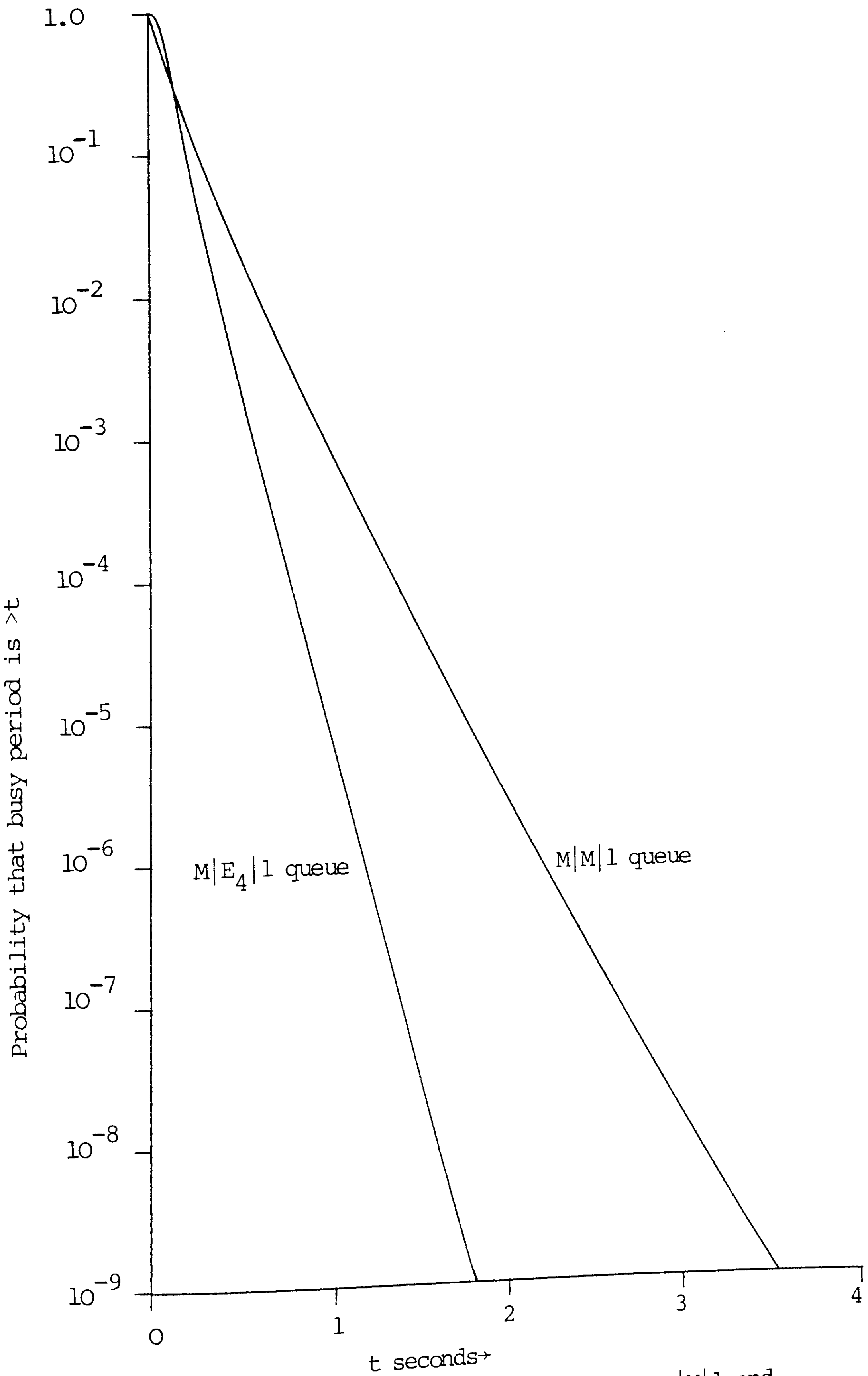
The Poisson elements of the busy period density converged after a few iterations. The accuracy of the busy period distribution computed from this sequence was poor. The poor accuracy was obtained because of the difficulties in the inversion of the Poisson transform.

The bilinear and Poisson elements can be used to compute the moments of the busy period distribution. The moments are obtained by evaluating derivatives of the Laplace transforms at s=0. This corresponds to the evaluation of the derivatives of the bilinear and Poisson transforms at z=1. The moments are obtained by a summation of the bilinear and Poisson elements weighted, appropriately.

A listing of the program for computing the busy period distribution for the $M|G|1$ queue is in Appendix 3.2.



Graph 2. Plot of busy period function for the M|M|1 queue. Curve (a) is a plot of computed results from Eqn (25). Curve (b) is a plot of results from numerical inversion of Laplace transform of the busy period density.



Graph 3 Comparison of busy period function for the $M|M|1$ and $M|E_4|1$ queues with $\lambda=1$ and $\mu=10$.

5.5 NUMBER SERVED IN THE BUSY PERIOD OF THE M|G|1 QUEUE

The distribution of the number served in the busy period is also important in performance evaluation of computer communication systems. Let f(k) and F(z) be the probability density function of the number served in the busy period and its z-transform, respectively. F(z) is given by[2]

$$F(z) = z^{-1}V[F(z)] = z^{-1} B_L[\lambda(1-F(z))] \tag{30}$$

where V(z) is the z-transform of the sequence v(k) which is the probability of k arrivals in a service period. This sequence v(k), corresponds to the Poisson sequence of the service time probability density function.

To compute the discrete probability density f(k), the following equation is used

$$F(z) = z^{-1}v(o) + z^{-1} \sum_{k=1}^{\infty} v(k)R_K(z) \tag{31}$$

where

$$R_k(z) = [F(z)]^k = \sum_{n=1}^{\infty} r(k,n)z^{-n}$$

After expanding both sides of Eqn (31) and matching same powers of z^{-1}, it can be shown that f(1)=v(o). Furthermore, it can be shown that f(k), for k>1 depends on f(1), f(2),..., f(k-1) and the sequence v(k). The probability of k arrivals in a service period, $v(k)$, is known. Thus $f(k)$, the probability of k customers being served in the busy period, can be computed from the sequences $r(n,m)$ and $v(k)$ by use of the equation

$$f(n) = \begin{cases} v(o), & n = 1 \\ \sum_{i=1}^{n} v(i)\, r(i,n-1), & n \geq 2 \end{cases} \tag{32}$$

The elements $r(m,n)$ are computed recursively from the equation

$$r(m,n) = \begin{cases} 0, & n<m;\ n,m \leq 0 \\ f(n), & m=1 \text{ and } n=1,2,\ldots \\ \sum_{i=1}^{n-1} r(1,i)\, r(m-1,n-1), & n \geq m > 1 \end{cases} \tag{33}$$

The process of computing the elements $r(m,n)$ must be combined with that of computing $f(n)$. This is because the value of $r(i,n-1)$ in Eqn (32) which must be evaluated from Eqn (33) depends on the values of the elements $f(1)$, $f(2),\ldots,$ $f(n-1)$.

Exact solutions of the sequence $f(n)$ for the M|M|1 and M|D|1 queues are available(2). For the M|M|1 queue, the probability of having n customers served in the busy period is given by

$$f(n) = \frac{(2n-2)!}{n[(n-1)!]^2} \rho^{n-1}(1+\rho)^{1-2n}, \quad n=1,2,... \tag{34}$$

For the M|D|1 queue f(n) is obtained from

$$f(n) = \frac{(n\rho)^{n-1}}{n!} \exp(-n\rho), \quad n=1,2,3,... \tag{35}$$

The evaluation of these two expressions Eqns(34) and (35) directly can result in overflow and underflow problems. To avoid these problems, these expressions must be rearranged. For the M|M|1 queue, f(n) Eqn (34) is computed recursively from the equation

$$f(n) = \begin{cases} \frac{1}{1+\rho}, & n=1 \\ \\ \frac{2\rho(2n-3)}{(1+\rho)^2 n} f(n-1), & n=2,3,4,... \end{cases} \tag{36}$$

For the M|D|1 queue, f(n) Eqn (35) is computed recursively from

$$f(n) = \begin{cases} \exp(-\rho), & n=1 \\ \\ (\frac{n}{n-1})^{n-2} \rho \ \exp(-\rho) f(n-1), & n=2,3,... \end{cases} \tag{37}$$

Using the computational procedure Eqns (32) and (33), f(n) was computed for the M|M|1 and M|D|1 queues on the

ICL 1904 computer, using double precision arithmetic with a significance of 20 digits[82]. The computed results were compared with the exact solutions from Eqns (36) and (37), for $\lambda=1$ and utilization factors in the range 0.1 to 0.9. The results from the computational procedure were accurate up to 17 digits.

5.6 CONCLUSIONS AND APPLICATION TO ANALYSIS OF COMPUTER COMMUNICATION SYSTEMS

Skinner's method[49] was used to obtain bounds on the waiting time distribution for the $G|G|1$ queue. Bounds for the $M|M|1$ and $E_2|M|1$ queues were derived. The bounds approached the exact solution as T approached zero. This method will find applications wherever there is a need of evaluating integrals of an n-fold self convolution of a probability density function. Another example where Skinner's method can be used, is the system considered in reference 23.

A transcendental Laplace transform was inverted by use of the bilinear transform and digital signal processing techniques. Such an equation can be for instance, the Laplace transform of the busy period initiated by a customer whose service time is x sec., which is being approximated by a diffusion process (see reference 3, p. 105). It can also be the Laplace transform of the service time density which is to be used to compute other probability functions.

The bilinear and Poisson transforms of the busy period density were obtained. An iterative procedure for computing the busy period distribution by use of the bilinear and

Poisson transforms was outlined. This distribution, for the $M|M|1$ and $M|E_4|1$ queues, was computed by use of the procedure. The accuracy of the computed results, compared to the exact solution, was good especially for low utilization factors. The busy period is important in the study of computer communication systems with priority(3,18-21,60). In time-sharing computer systems(60), the interruption of users to perform maintenance and system functions can create problems to the users, e.g., the loss of files and the incomplete execution of programs. Therefore information concerning the occurrence and the length of the busy and idle periods can be used to schedule system maintenance to minimise these interferences.

Stored-program-control telephone systems(6,22) and integrated digital voice-data systems in store-and-forward networks(18-21) operate on a priority basis. The real-time messages (e.g., speech, or terminal users) have priority over the non-real-time messages (e.g., batch jobs or data messages). The busy period distribution then provides useful performance measures like link utilization, and some percentile performance specifications. The latter cannot be obtained without the knowledge of the detailed busy period distribution.

The solution of the busy period distribution enables the waiting time distribution for the first-come-first-served (FCFS) $M|G|1$ queue(2) and the last-come-first-served (LCFS)

M|G|1 queue[3,p.119] to be computed. This computational method will also enable the waiting time distribution for the head-of-the-line priority (HDL) queue[3, pp.119-123] to be computed.

The distribution for the number served in the busy period was computed by use of a recurrence relation. For the cases tried, the accuracy of the computed results was only a few digits poorer than the inherent accuracy of the computer used. This is sufficient for most practical purposes.

CHAPTER 6

LINK COST EVALUATION IN THE PILOT PACKET SWITCHED NETWORKS

6.1 INTRODUCTION

An estimate of the delay a packet would experience if it were assigned to a particular link by a node of the Pilot Packet Switched Network (PPSN) is obtained. A link is the set of lines which join a particular pair of nodes. Before a node can assign packets to links, it must have information about the delay the packets will experience through each link. This information is stored in the Cost Matrix of each node and it is updated at regular intervals. The node assigns packets to minimum-delay links.

Fig. 1 illustrates the processes performed after a packet has been assigned to a link. After being allocated for transmission on a particular link, a packet for output from a PPSN node passes to the Link Controller process for the selected link. The Link Controller selects one of the lines of the link for transmission of the packet and passes the packet to the appropriate Line Handler process. The Link Controller selects the line having minimum delay. The Line Handler process in turn, passes the packet to the HDLC (High-Level Data Link Controller) Line Unit responsible for the chosen line. At each stage, the packet must queue for service, if there are packets ahead of it.

Each node calculates an estimate of the delay through each link at regular intervals. An interval of 5s is used in the PPSN.

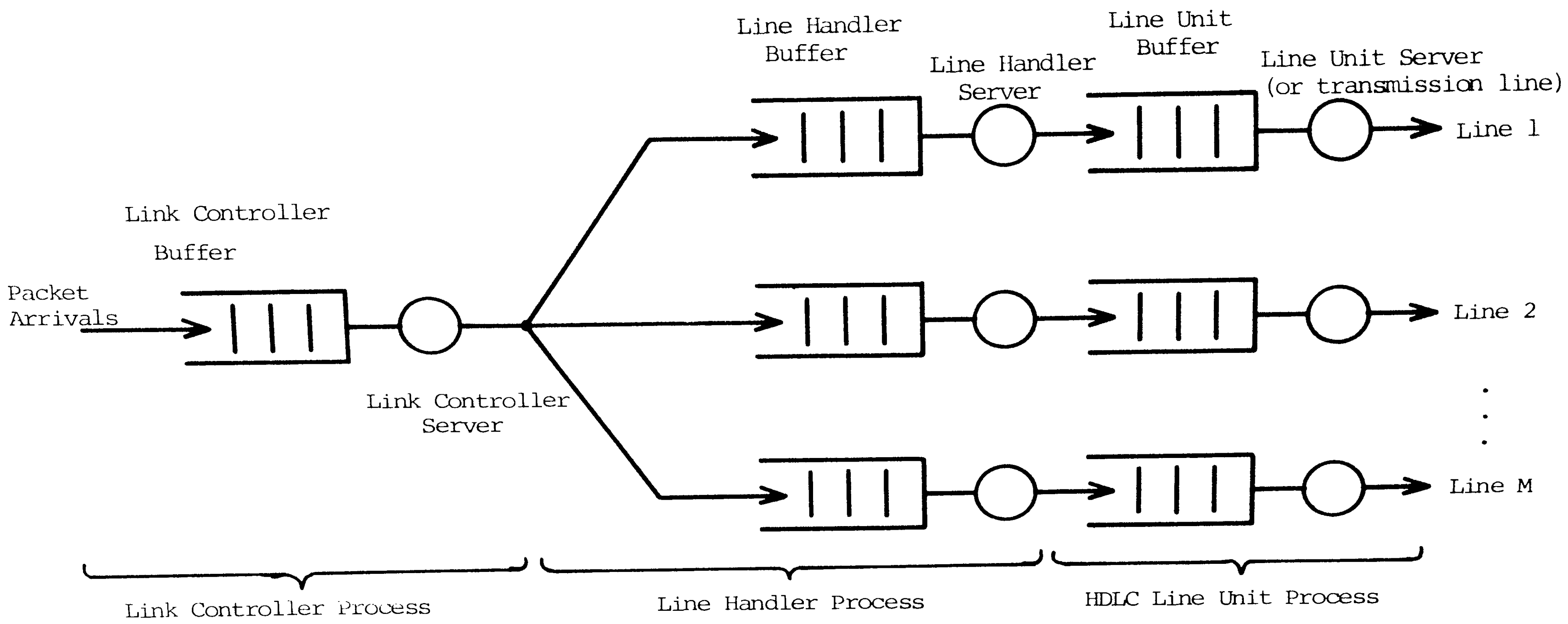


Fig. 1 Queueing for Transmission at a Pilot Packet Switched Network Node

One method of estimating this delay has been described in reference 70. In this method, only the delay through the Line Units is used to evaluate the link delay. Ten samples of the Line Unit's queue lengths are taken during the 5s interval and an average queue length obtained for each Line Unit. These averages, together with the line speeds, are used in a formula to produce an estimate of the link delay. The link delay estimates obtained are then used in routing decisions.

Only the Line Unit queue lengths are taken into account in this evaluation; the Link Controller and Line Handler queues are ignored. This results in the estimate of the link delay being less accurate than could be obtained by using both the Link Controller and the Line Unit queue lengths in forming the estimate. This chapter presents a method for estimating the link delay, which takes into account both the Link Controller and Line Unit queue lengths.

6.2 DELAY THROUGH LINE UNIT i

The operation of the Link Controller is assumed to be as follows. At time instant n, it removes a packet (provided there is at least one) from its buffer and allocates this packet to the Line Unit offering minimum delay (calculated from the Line Unit queue length and line speed). If two or more Line Units offer the minimum delay, an allocation is made randomly, to one of these Line Units. Other assumptions used are:

(i) The Link Controller service time is very much smaller than that of the Line Units, thus it is ignored in the derivation of an estimate of the link delay.

(ii) All packets are of equal length.

(iii) The Line Handler queue and the time to execute the Line Handler process can be neglected.

Let $d_i(n)$ be the delay a packet to be allocated to the Line Units, at instant n, would experience through Line Unit i. The Link Controller allocates a packet to the Line Unit having minimum delay $d_i(n)$. The delay is given by

$$d_i(n) = \frac{s_o}{s_i} \max(o, q_i(n)-1) + w_i(n) + \frac{s_o}{s_i} \quad \text{(time units)} \tag{1}$$

where $q_i(n)$ = number of packets in the buffer of Line Unit i, plus the one in transmission, just prior to time instant n.

$w_i(n)$ = number of time units, at instant n, required for the completion of the transmission of the packet being transmitted by line i.

s_i = speed of line i in bits per second.

s_o = base speed in bits per second.

A time unit is the reciprocal of the base speed.

Eqn (1) can be understood as follows. A packet allocated to Line Unit i at instant n, will find $\max(0, q_i(n)-1)$ packets queueing ahead of it and one in transmission, if any. The packets queued up will require a service time of $(\frac{s_o}{s_i})\max[0, q_i(n)-1]$ time units. The packet in transmission will still have $w_i(n)$ time units to complete transmission. The allocated packet will wait for $(\frac{s_o}{s_i})\max[0, q_i(n)-1]+w_i(n)$ time units before service. Its transmission (or service) time is $(\frac{s_o}{s_i})$, thus the total delay through Line Unit i is given by Eqn (1). By simplification of Eqn (1), the delay is obtained as

$$d_i(n) = \frac{s_o}{s_i}\max[1, q_i(n)]+w_i(n) \quad \text{(time units)} \qquad (2)$$

Suppose a packet arrives at the Link Controller at instant n and finds $\ell(n)$ packets in the Link Controller buffer.

As long as the Link Controller buffer contains packets, one packet is allocated to a Line Unit at each instant. It is assumed that the Line Unit buffers have unlimited capacity, so that this is possible. The packet arriving at time instant n will then be allocated to a Line Unit at instant $n+\ell(n)$. The delay it will experience through Line Unit i is thus given by

$$d_i(n+\ell(n)) = \frac{s_o}{s_i}\{\max[1,q_i(n)] + \sum_{m=0}^{\ell(n)} a_i(n+m)\} + w_i(n) - \ell(n), \quad \ell(n)=0,1,2,\ldots \tag{3}$$

where $a_i(n)$ is the number (0 or 1) of packets allocated to Line Unit i just after instant n.

Eqn (3) gives the delay through Line Unit i. To include the effect of the Link Controller queue, an expression is needed for c(n), the delay experienced by a packet which arrives at the Link Controller at time instant n.

6.3 ESTIMATE OF THE LINK DELAY

$\ell(n)$ is the number of packets in the Link Controller buffer at instant n. $F_j(n)$, the delay a packet arriving at instant n experiences through the Link Controller and through Line Unit j is then given by

$$F_j(n) = \ell(n) + d_j(n+\ell(n)) \quad \text{(time units)} \tag{4}$$

This is because one packet is allocated to a Line Unit at each instant, as long as the Link Controller buffer contains packets. The packet arriving at instant n thus waits $\ell(n)$ time units in the Link Controller buffer plus $d_j(n+\ell(n))$ time units in the buffer of Line Unit j.

It is assumed that the Link Controller allocates packets to the Line Units by comparing the delays $d_i(n)$, and it allocates a packet to the Line Unit having minimum delay. This selection strategy will approximately equalise the delays through the Line Units. Thus c(n), the delay experienced by a packet which arrives at the Link Controller at time n, is

$$c(n) \simeq F_j(n), \quad \text{for all the } j\text{'s} \tag{5}$$

Using Eqns(5), (4) and (3), the delay c(n) becomes

$$c(n) \simeq \frac{s_o}{s_j}\{\max(1,q_j(n)) + \sum_{m=0}^{\ell(n)} a_j(n+m)\} + w_j(n) \tag{6}$$

In Eqn (6), $\Sigma a_j(n+m)$ is the total number of packets allocated to Line Unit j during the interval n to $n+\ell(n)$. To a close approximation, this can be replaced by $\ell(n)\rho_j(n)$, the product of the total number of packets in the Link Controller at time n and the proportion of packets sent via Link Unit j. Thus Eqn (6) becomes

$$c(n) \simeq \frac{s_o}{s_j}\{\max(1,q_j(n))+\ell(n)\rho_j(n)\}+ w_j(n) \qquad (7)$$

$\rho_j(n)$ in this expression can be eliminated by the following steps:

(i) Equate the right hand side of Eqn (7) to the same expression, but with j replaced by i (see Eqn (5)).

(ii) Manipulate to get an equation with $\rho_i(n)$ on one side.

(iii) Sum for all i's and equate the result to 1 ($\rho_i(n)$'s are proportions).

(iv) Rearrange to obtain $\rho_j(n)$.

(v) Substitute the expression for $\rho_j(n)$ into Eqn (7).

The resulting expression is

$$\rho_j(n) \simeq \frac{1+\dfrac{1}{\ell(n)}\sum_{i=1}^{M}\max(1,q_i(n)) - \dfrac{w_j(n)}{\ell(n)}\sum_{i=1}^{M}\dfrac{s_i}{s_o} + \dfrac{1}{\ell(n)}\sum_{i=1}^{M}\dfrac{s_i w_i(n)}{s_o}}{\sum_{i=1}^{M}\dfrac{s_i}{s_j}} - \frac{\max(1,q_j(n))}{\ell(n)}$$

where M is the total number of Line Units.

Substitution of this expression into Eqn(7) gives, after simplification,

$$c(n) \simeq \frac{\ell(n) + \sum_{i=1}^{M} \max(1, q_i(n)) + \sum_{i=1}^{M} \frac{s_i w_i(n)}{s_o}}{\sum_{i=1}^{M} \frac{s_i}{s_o}} \qquad (8)$$

For a packet arriving at the Link Controller at instant n, and finding $\ell(n)$ packets already in the Link Controller queue, Eqn (8) provides an estimate of the delay it would experience.

6.4 AVERAGE OF THE ESTIMATED DELAY

Let N be the number of time units in each sampling interval. The delay experienced by a packet arriving at the Link Controller at the mth instant in the jth sampling interval is given by $c[(j-1)N+m]$. This can be averaged over the sampling interval by summing for m=0,1,...,N-1 and dividing the result by N. The average estimated delay is then given by the following approximation

$$C(jN) \simeq \frac{1}{\sum_{i=1}^{M} \frac{s_i}{s_o}} \left[L(jN) + \sum_{i=1}^{M} Q_i(jN) + \sum_{i=1}^{M} \frac{s_i W_i(jN)}{s_o} \right] \qquad (9)$$

where $L(jN)$ is the average Link Controller queue length during the jth sampling interval taken over N samples.

$Q_i(jN)$ is the average queue length for Line Unit i and it is assumed that

$$Q_i(jN) \simeq \frac{1}{N} \sum_{m=0}^{N-1} \max\{1, q_i[(j-1)N+m]\}$$

$W_i(jN)$ is the average of the number of time units left for completing the transmission of the packet in service in Line Unit i.

The PPSN node does not have information about $w_i(n)$. In this case it can be assumed that the packet in transmission, at each time instant, still has the maximum number of time units, which is $(\frac{s_o}{s_i})$, before completing transmission. Replacing $W_i(jn)$ by $(\frac{s_o}{s_i})$ in Eqn (9), the estimated delay $C(jN)$ is then given by

$$C(jN) \simeq \frac{1}{\sum_{i=0}^{M} \frac{s_i}{s_o}} \left[L(jN) + \sum_{i=1}^{M} Q_i(jN) + M\right] \tag{10}$$

Thus the average of the delay estimate can be obtained by taking N samples and averages of

(i) Link Controller queue length, and

(ii) Line Unit queue lengths,

and using these together with line speeds in Eqn (10).

The link delay estimate which ignores the Link Controller queue is given by[70]

$$C(jN) = \frac{s_o M}{\sum_{i=1}^{M} \left[\frac{s_i}{1+Q_i(jN)}\right]} \tag{11}$$

6.5 SIMULATION AND NUMERICAL EXAMPLES

Table 1 shows a simulation of the Link Controller/Line Units process, to determine the delay $F_j(n)$, Eqn (4). For this simulation, line speed ratios $\frac{s_o}{s_1} = 12$, $\frac{s_o}{s_2} = 8$ and $\frac{s_o}{s_3} = 4$ and initial conditions $q_1(1) = 1$, $q_2(1) = 3$, $q_3(1) = 5$, $\ell(1) = 5$ are assumed. The initial values of the variables $w_i(1)$ are assumed to be equal to the values of the respective line speed ratios. $d_i(n)$ indicates the delay obtained by use of Eqn (2). The table shows that a packet arriving at instant n=1, to find $\ell(1)=5$ and the above initial conditions, will experience a total delay of 36 time units. The delay obtained by use of the derived equation (8) is 37.1 time units. Using Eqn (11) which ignores the Link Controller queue, the delay is 26.2 time units.

Table 2 is a simulation for line speed ratios $\frac{s_o}{s_1} = 9$, $\frac{s_o}{s_2} = 7$, and $\frac{s_o}{s_3} = 5$ and initial values for $q_i(1)$ and $w_i(1)$ being $q_1(1)=2$, $q_2(1)=3$, $q_3(1)=5$, $w_1(1)=5$, $w_2(1)=2$ and $w_3(1)=1$. The table shows that a packet arriving at instant n=1, and finding the above initial conditions and $\ell(1)=9$ packets in the Link Controller queue, will experience a total delay of 41 time units. By use of Eqn (8), the total delay is 44.1 time units. Using the less optimistic Eqn (10), the delay is 48.5 time units. The delay obtained by use of the equation which ignores the Link Controller queue is 28.3 time units.

From the examples given, Eqn (10) does provide a reasonable estimate of the delay compared to the delay $F_j(n)$, while Eqn (11) gives an underestimate of the delay. This algorithm for the estimation of the Link Cost was incorporated into the PPSN, with minor modifications to take into account the limited buffer capacities of the Line Units.

TABLE 1

Simulation to determine the delay which will be experienced by a packet which arrives to find 5 packets in the Link Controller queue and $q_1(1)=1$, $q_2(1)=3$, $q_3(1)=5$, $w_1(1)=12$, $w_2(1)=8$ and $w_3(1)=4$. After queueing for 5 time units in the Link Controller queue, the packet is allocated to Line Unit 3 and it will spend 31 time units in this queue.

n	$\frac{s_o}{s_1} = 12$				$\frac{s_o}{s_2} = 8$				$\frac{s_o}{s_3} = 4$			
	$q_1(n)$	$a_1(n)$	$w_1(n)$	$d_1(n)$	$q_2(n)$	$a_2(n)$	$w_2(n)$	$d_2(n)$	$q_3(n)$	$a_3(n)$	$w_3(n)$	$d_3(n)$
1	1	1	12	24	3	0	8	32	5	0	4	24
2	2	0	11	35	3	0	7	31	5	1	3	23
3	2	0	10	34	3	0	6	30	6	1	2	26
4	2	0	9	33	3	0	5	29	7	1	1	29
5	2	0	8	32	3	1	4	28	7	0	4	32
6	2	0	7	31	4	0	3	35	7	1	3	31

TABLE 2

Simulation to determine the delay which will be experienced by a packet which arrives to find 9 packets in the Link Controller queue and $q_1(1)=2$, $q_2(1)=3$, $q_3(1)=5$, $w_1(1)=5$, $w_2(1)=2$ and $w_3(1)=1$. After queueing for 9 time units in the Link Controller queue, the packet is allocated to Line Unit 1 and will spend 32 time units in this queue.

n	$\frac{s_o}{s_1} = 9$				$\frac{s_o}{s_2} = 7$				$\frac{s_o}{s_3} = 5$			
	$q_1(n)$	$a_1(n)$	$w_1(n)$	$d_1(n)$	$q_2(n)$	$a_2(n)$	$w_2(n)$	$d_2(n)$	$q_3(n)$	$a_3(n)$	$w_3(n)$	$d_3(n)$
1	2	1	5	23	3	0	2	23	5	0	1	26
2	3	0	4	31	3	1	1	22	4	0	5	25
3	3	0	3	30	3	0	7	28	4	1	4	24
4	3	0	2	29	3	1	6	27	5	0	3	28
5	3	0	1	28	4	0	5	33	5	1	2	27
6	2	1	9	27	4	0	4	32	6	0	1	31
7	3	0	8	35	4	0	3	31	5	1	5	30
8	3	0	7	34	4	1	2	30	6	0	4	34
9	3	0	6	33	5	0	1	36	6	1	3	33
10	3	1	5	32	4	0	7	35	7	0	2	37

CHAPTER 7

CONCLUSIONS AND SUGGESTIONS FOR FURTHER RESEARCH

7.1 CONCLUSIONS

Digital signal processing methods were used in the solution of some queueing problems which often arise in the analysis of computer communication systems.

The z-transform of the queue length distribution for the bulk service $M|G^Y|1$ queueing system was obtained. The exact inversion of the transform, like many other transforms, was found to be difficult because of the multiple dependence of the queue-states. Some methods for the inversion of this transform by use of digital signal processing techniques were presented.

Bilinear and Poisson transforms were used in the solution of several queueing problems. The Poisson transform interrelates the various distributions for the $M|G|1$ queue. This transform was used in the analysis of this queue. The Pollaczek-Khinchin transform equation for the number in this queue was inverted by use of a recursive digital filter.

The bilinear and Poisson transform sequences were presented as useful discrete representations of continuous-time probability functions. The transforms enabled digital signal processing techniques to be applied in queue analysis. A method in which the Laplace transform was approximately inverted by use of the bilinear transformation was presented. The method was used in computing the waiting time distribution for the $M|G|1$ queue. A systematic procedure for the inversion of the Laplace transform by use of the bilinear and Poisson transforms was outlined. The procedure was used in the computation of the bilinear sequence of the waiting time distribution for the $G|G|1$ queue.

The inversion of Laplace transforms by use of the bilinear and Poisson transforms requires the ability to compute the bilinear and Poisson sequences from continuous-time probability functions and these functions from the sequences. Efficient methods for computing the sequences from the continuous-time functions and vice-versa were developed.

The bilinear transform was used in the inversion of a transcendental Laplace transform expression. Expressions for the bilinear and Poisson transforms of the busy period distribution for the $M|G|1$ queue were derived. From these transforms, a procedure for the computation of the busy period distribution was obtained.

Skinner's method was used to obtain upper and lower bounds on the waiting time distribution for the $G|G|1$ queue. The distribution for the number served in the $M|G|1$ queue busy period was computed using a recurrence relation.

Solutions for some queues, which were easy to derive or were available, were used to test the computational methods for utilization factors $\rho=0.1$ and $\rho=0.9$. In practice, communication systems are designed to operate within these values. Accurate inversion of transforms is important in the study of computer communication systems in which the design criteria are often in the 10^{-5} probability range. The accuracy obtained by many of the computational methods was sufficient for most practical purposes.

Another problem analysed, though not by use of digital signal processing methods, was that of estimating the delay along a link between two nodes. This estimate is useful in routing decisions in a computer communication network. An equation for the link delay estimate was derived. The delay obtained by simulation, was compared with the delay obtained from the derived equation. The comparison showed that the equation gave a reasonable estimate of the link delay.

7.2 SUGGESTIONS FOR FURTHER RESEARCH

There are several areas in which further research is necessary. Probability functions used for this work were known expressions and the samples obtained from these functions retained the full accuracy of the computer. In practice, these functions can be obtained by statistical observations and data measurements. Samples obtained from the observations and measurements will often have errors whose magnitudes are greater than the magnitude of the inherent error of the computer. An area for further research is the study of the effects of these errors on the accuracy of the results computed by use of the methods presented.

In this research, systems whose exact solutions were either available or easy to obtain were analysed. Applying the computational methods to the study of computer systems, whose exact solutions cannot be obtained easily, is another area for further research. A few examples are: computer systems with priority [3], frequency-division multiple-access (FDMA) and time-division multiple-access (TDMA) systems [34], and discrete-time queues [31,32].

This research should be only a guideline to a systematic approach to the solution of queueing problems by use of digital signal processing techniques. There are many problems that the computational methods will not solve directly, e.g. systems considered in references 19, 29, 30 and 33. Further research is therefore necessary in the analysis of such systems by use of digital signal processing methods.

APPENDIX 1

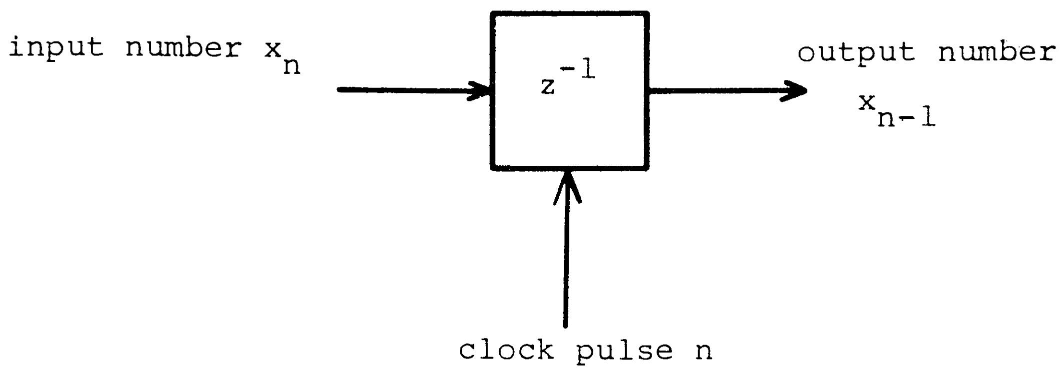
Components of a Digital Filter

A digital filter processes a set of input numbers to obtain another set of numbers at its output. A linear time invariant digital filter consists of three basic components. The first component is a delay unit as shown in Fig. 1(a). When a pulse clocks the delay unit, the real number stored in this delay unit is pushed out and a new number at the input is stored. The second component is an adder/subtractor as shown in Fig. 1(b). This component operates instantly by adding/subtracting the numbers at the inputs. The sum/difference is obtained at the output. The third component is a multiplier as shown in Fig. 1(c). For a linear digital filter these coefficients are fixed numbers. The input number is multiplied by the coefficient of the multiplier and the result is obtained instantly at the output.

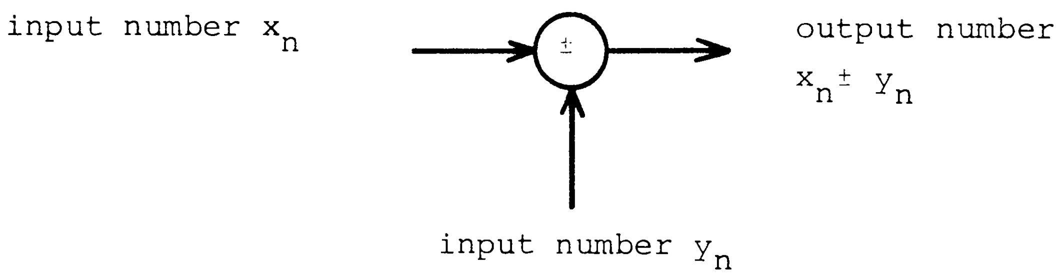
By an interconnection of these components, linear difference equations can be solved. There are several forms of realisations of difference equations[73,74]. The methods of realization can be divided into two classes, recursive and non-recursive. For a recursive realization, the current output sample is a function of past outputs as well as present and past input samples. For a non-recursive realization, the current output sample is a function of only the past and present input samples. Further information on digital filtering can be obtained from references 72-74.

Fig. 1 The components of a digital filter

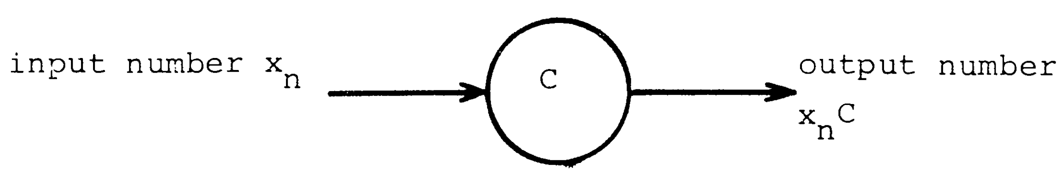
(a) The delay unit



(b) The adder/substractor



(c) The multiplier



APPENDIX 2

Smith's Algorithm for Summation of Polynomial Series and their Derivatives

In this appendix, Smith's algorithm(85) is outlined and used to obtain the set of quantities $S_N(t)$, $S_{N-1}(t)$, $\ldots, S_o(t)$, $S_N^1(t), \ldots, S_1^1(t)$ which are in Chapter 4.

Summing the Polynomial Series

Any set of polynomials which are orthogonal over an interval of the real line or orthogonal over a discrete set of real numbers can be shown to satisfy a three-term recurrence relation.

$$p_o(t) = \beta_o$$

$$p_1(t) = (\beta_1 t - \sigma_1) p_o(t)$$

$$p_n(t) = (\beta_n t - \sigma_n) p_{n-1}(t) - \phi_n p_{n-2}(t), \quad n=2,3,4,\ldots \tag{1}$$

where $p_n(t)$ is a polynomial of degree n and σ_n, ϕ_n and β_n are constant coefficients which are usually simple and known. The recurrence relation Eqn(1) can be used to sum a polynomial series

$$x(t) = \sum_{n=0}^{N} x_n \, p_n(t) \tag{2}$$

A set of quantities S_m is defined by the equation

$$S_m = 0 \quad , \; m>N$$

$$S_m = x_m+(\beta_{m+1}t - \sigma_{m+1})S_{m+1}-\phi_{m+1}S_{m+2}, \qquad 0\leq m\leq N \tag{3}$$

By substituting the recurrence relation Eqn (1) into Eqn (2) and using Eqn (2), it can be shown that the series is given simply by

$$x(t) = \beta_o S_o \tag{4}$$

S_o is calculated by successively evaluating the quantities S_N, S_N-1,....

The recurrence relation for the Laguerre polynomial series is

$$L_o(t) = 1$$

$$L_1(t) = 1-t$$

$$nL_n(t) = (2n-1-t)L_{n-1}(t)-(n-1)L_{n-2}(t) \tag{5}$$

Comparing Eqn (5) and Eqn (1), the coefficients β_n, σ_n and ϕ_n, for the Laguerre polynomial functions are

$$\beta_n = \begin{cases} 1 & , \; n=0 \\ -\frac{1}{n} & , \; n=1,2,... \end{cases} \tag{6}$$

$$\sigma_n = -(\frac{2n-1}{n}), \quad n>0 \tag{7}$$

$$\varphi_n = \frac{n-1}{n} \quad , \quad n>0 \tag{8}$$

Substituting these coefficients into Eqn (3), the set of quantities $S_m(t)$ for the Laguerre polynomial functions are obtained, as given in Chapter 4.

Summing the First Derivative of Polynomial Series

The first derivative of the polynomial series Eqn (2) is

$$x^1(t) = \sum_{n=1}^{N} x_n p_n^1(t) \tag{9}$$

By taking the derivative of Eqn (1), a new recurrence relation involving the derivatives of the polynomials $p_n^1(t)$ is obtained

$$p_1^1(t) = \beta_1 p_o(t)$$

$$p_n^1(t) = (\beta_n\ t - \sigma_n)p_{n-1}^1(t) - \varphi_n p_{n-2}^1(t) + \beta_n p_{n-1}(t),$$

$$n=2,3,... \tag{10}$$

Substitution of Eqn (10) into Eqn (9) and using the set of quantitues S_m in Eqn (3) results in

$$x^1(t) = \sum_{n=1}^{N} S_n \beta_n p_{n-1}(t) \tag{11}$$

The right-hand side of this equation is similar to Eqn (2) and can be treated as a polynomial series. Another set of quantities S_m^1 are defined by

$$S_m^1 = 0 \,, \quad m>N$$

$$S_m^1 = \beta_m S_m + (\beta_m t - \sigma_m) S_{m+1}^1 - \phi_{m+1} S_{m+2}^1 \,, \qquad 0 \leq m \leq N \tag{12}$$

giving the series Eqn (11) as

$$x'(t) = \beta_o S_1^1 \tag{13}$$

The derivative of the series at any value of t can be obtained by calculating successively the set of quantities S_N, $S_{N-1}, \ldots,$ S_o, S_N^1, $S_{N-1}^1(t), \ldots, S_1^1$.

Using Eqns (6)-(8) in Eqn (12), the set of quantities $S_N(t)$, $S_{N-1}(t), \ldots,$ $S_o(t)$, $S_{N-1}^1(t), \ldots,$ $S_1^1(t)$ for the Laguerre polynomial functions are obtained, as given in Chapter 4.

APPENDIX 3

Listing of Programs

3.1 Listing of Program for Computing Bilinear Sequence of Waiting Time Distribution for $G|G|1$ Queue.

```
*************************************************************************
*************************************************************************
***********
** Pdn     7    2499-RK                    TERMINAL:  72                 6 SEP 82 10:4
***********
*************************************************************************
*************************************************************************


C
C Program computes the bilinear transform of the waiting time p.d.f.
C for the G/G/1 queue by use of the complex cepsrum. This method
C uses the bilinear transforms of the p.d.f.'s of service and
C interarrival times in place of the periodic samples. This permits
C digital filtering to be done more efficiently.
C
C
      DOUBLE PRECISION A(2048),B(2048),ASUM,UFACT,MU
      DOUBLE PRECISION LAMDA,AI,AR
      REAL CHECK
      DATA NOUT/15/
      MU=8.0D0/5.0D0
      LAMDA=1.0D0
      UFACT=LAMDA/MU
      M=8
      N=2**M
      NHALF=N/2
      WRITE(NOUT,10)
   10 FORMAT(//,10X,'The bilinear transform of the waiting time p.d.f fo
     *r the G/G/1 queue'/,10X,'computed by use of the bilinear transform
     *s of the service time and'/,10X,'the interarrival time p.d.f.'s. T
     *he cepstrum method is used to extr-'/,10X,'act the bilinear transf
     *orm of the waiting time p.d.f..')
C
C Generate the bilinear transform of the service time p.d.f and store
C this in array B
C
      CALL SERPDF(B,N)
C
C Generate the bilinear transform of the interarrival time p.d.f.
C
      CALL ARRPDF(A,N,LAMDA)
C
C Convolve the service time p.d.f. with the interarrival time p.d.f.
C time-reversed i.e. obtain b(k)*a(-k). This is done in the frequency
C domain by taking the DFT's of a(-k) and b(k) and multiplying
C them element by element. The inverse DFT then gives c(k) which is
C stored in array A.
C
      CALL REFFT(B,M-1,NHALF)
      CALL REFFT(A,M-1,NHALF)
      A(1)=A(1)*B(1)
      A(2)=A(2)*B(2)
      DO 30 KQ=2,NHALF
      AR=A(2*KQ-1)
      AI=A(2*KQ)
      A(2*KQ-1)=AR*B(2*KQ-1)-AI*B(2*KQ)
      A(2*KQ)=AI*B(2*KQ-1)+AR*B(2*KQ)
   30 CONTINUE
      CALL RFTINV(A,M-1,NHALF)
```

```
      WRITE(NOUT,32)
   32 FORMAT(//,10X,'The sequence c(k) obtained by convolution'//)
      CALL OUTPUT(A,N)
C
C Compute sequence from analytic solution
C
      CALL RESOLN(A,N)
      WRITE(NOUT,33)
   33 FORMAT(//,10X,'c(k) from analytic solution'//)
      CALL OUTPUT(A,N)
C
C Subroutine LHSIDE generates the sequence whose z-transform is
C H-{a[1-z]/[1+z]}/W{a[1-z]/[1+z]}.
C
      CALL LHSIDE(A,N)
C
C Get complex cepstrum of a(k)
C
      CALL REFFT(A,M-1,NHALF)
      CALL LOGARR(A,N,CHECK)
      WRITE(NOUT,35)CHECK
   35 FORMAT(//,8X,'Value of CHECK',F20.6/)
      CALL RFTINV(A,M-1,NHALF)
C
C Zero negative time part of cepstrum.
C
      CALL NULNEG(A,N)
C
C Get minimum phase part.
C
      CALL REFFT(A,M-1,NHALF)
C
C Negate because inverse sequence is required.
C
      DO 70 I=1,N
   70 A(I)=-A(I)
C
C Obtain the exponential of complex sequence.
C
      CALL EXPARR(A,N)
C
C Normalise sequence since for p.d.f. zero freq coeff=1
C
      ASUM=A(1)
      DO 90 I=1,N
   90 A(I)=A(I)/ASUM
C
C Now obtain the inverse DFT to get the required p.d.f.
C
      CALL RFTINV(A,M-1,NHALF)
      WRITE(NOUT,100)
  100 FORMAT(//,8X,'Computed bilinear trans of waiting time pdf.'//)
      WRITE(NOUT,104)MU
  104 FORMAT(//,8X,'MU-the average service rate       ',F25.13/)
      WRITE(NOUT,105)LAMDA
  105 FORMAT(//,8X,'Lamda the average arrival rate    ',F25.13/)
      WRITE(NOUT,106)N
  106 FORMAT(//,8X,'Block size used for this computation',I7/)
      CALL OUTPUT(A,N)
C
```

```
C
C Now compute the bilinear transform of waiting time p.d.f. from
C analytic solution.
C
      CALL RWAIT(A,N)
      WRITE(NOUT,120)
  120 FORMAT(//,8X,'Bilinear trans of waiting time from analytic soln'/
      CALL OUTPUT(A,N)
      STOP
      END
C
C
C
      SUBROUTINE SERPDF(F,M)
      DOUBLE PRECISION F(M)
      MHALF=M/2
      F(1)=5.0D0/8.0D0
      F(2)=11.0D0/24.0D0
      DO 10 K=3,MHALF
      F(K)=(-1.0D0/3.0D0)**(K-1)
      F(K)=-F(K)
   10 CONTINUE
      MHP1=MHALF+1
      DO 20 KL=MHP1,M
   20 F(KL)=0.0D0
      RETURN
      END
C
C
C
      SUBROUTINE ARRPDF(A,N,LAMDA)
      DOUBLE PRECISION A(N),LAMDA
      DO 10 KQ=1,N
   10 A(KQ)=0.0D0
      A(1)=LAMDA/2.0D0
      A(N)=LAMDA/2.0D0
      RETURN
      END
C
C
C
      SUBROUTINE RESOLN(B,N)
      DOUBLE PRECISION B(N)
      NHALF=N/2
      DO 10 KP=3,NHALF
   10 B(KP)=(-1.0D0/3.0D0)**KP
      NHP1=NHALF+1
      DO 20 KT=NHP1,N
   20 B(KT)=0.0D0
      B(1)=13.0D0/24.0D0
      B(2)=25.0D0/144.0D0
      B(N)=5.0D0/16.0D0
      RETURN
      END
C
C
C
      SUBROUTINE RWAIT(A,N)
      DOUBLE PRECISION A(N)
      DO 10 K=1,N
```

```
      A(K)=0.75D0*((1.0D0/3.0D0)**(K-1))-0.15D0*((-0.2D0)**(K-1))
   10 CONTINUE
      RETURN
      END
C
C
C
      SUBROUTINE OUTPUT(A,N)
      DOUBLE PRECISION A(N)
      M=N
      IF(M.GT.256)M=256
      WRITE(15,10)(A(I),I=1,M)
   10 FORMAT(1H ,5F25.15)
       WRITE(15,20)A(N)
   20 FORMAT(1H0,8X,'Last element of the sequence is ',F25.15)
      RETURN
      END
C
C
C
      SUBROUTINE LHSIDE(V,M)
      DOUBLE PRECISION V,SUM
      DIMENSION V(M)
C
C SUBROUTINE GENERATES THE LHS OF H-[1-z]/W[1-z]
C
C Note that the zero at z=-1 is not introduced
C Zero at z=1 is removed the sequence is processed through a digital
C filter with transfer function 1/(1-z).
C
      MM1=M-1
      MHALF=M/2
      MHP1=MHALF+1
      MHM1=MHALF-1
      V(1)=V(1)-1.0D0
C
C DEAL WITH NEGATIVE TIME SAMPLES FIRST
C
      SUM=0.0D0
      DO 80 I=MHP1,M
      SUM=SUM+V(I)
   80 V(I)=SUM
C
C CONTINUE WITH POSITIVE TIME SAMPLES
C
      DO 90 I=1,MHALF
      SUM=SUM+V(I)
   90 V(I)=SUM
      RETURN
      END
```

```
       SUBROUTINE LOGARR(A,N,CHECK)
      DOUBLE PRECISION A(N),PI,PHZ,PHZPRE,PJ,PTREND,DMAGSQ
      REAL CHECK
C
C COMPUTE LOG OF DOUBLE PRECISION COMPLEX ARRAY
C
      CHECK=0.0
      IF(A(1).LE.1.0D-10)GO TO 100
      A(1)=DLOG(A(1))
      A(2)=DABS(A(2))
      IF(A(2).LE.1.0D-10)GO TO 100
      A(2)=DLOG(A(2))
      NHALF=N/2
      DO 10 I=2,NHALF
      DMAGSQ=A(2*I-1)**2+A(2*I)**2
      IF(DMAGSQ.LE.1.0D-10)GO TO 100
      PHZ=DATAN2(A(2*I),A(2*I-1))
      A(2*I-1)=0.5D0*DLOG(DMAGSQ)
   10 A(2*I)=PHZ
C
C MAKE PHASE CONTINUOUS
C
      PI=3.14159265358979320D0
      PJ=0.0D0
      PHZPRE=0.0D0
      DO 50 I=2,NHALF
      PHZ=A(2*I)
      IF(DABS(PHZ-PHZPRE+PJ).LT.PI)GO TO 45
C
C DEAL WITH PHASE JUMPS EXCEEDING PI
C
      IF((PHZ-PHZPRE+PJ).GE.PI)GO TO 40
C
```

```
C DEAL WITH NEGATIVE JUMP



      PJ=PJ+2.0D0*PI
      GO TO 45
C
C DEAL WITH POSITIVE JUMP
C
   40 PJ=PJ-2.0D0*PI
   45 CONTINUE
      A(2*I)=PHZ+PJ
   50 PHZPRE=A(2*I)
C
C REMOVE LINEAR TREND
C
      PHTOT=A(N)
      PIMUL=PHTOT/3.1415926535
C
C ROUND TO NEAREST INTEGER MULTIPLE OF PI
C
      NPI=IFIX(ABS(PIMUL)+0.5)
      NPI=ISIGN(NPI,IFIX(PHTOT))
      DO 70 I=2,NHALF
      PTREND=DBLE(FLOAT(NPI))
      PTREND=PTREND*DBLE(FLOAT(I-1))*PI/DBLE(FLOAT(NHALF))
   70 A(2*I)=A(2*I)-PTREND
      IF(CHECK.EQ.0.0)GO TO 110
  100 CHECK=1.0
  110 RETURN
      END


      SUBROUTINE EXPARR(A,N)
      DOUBLE PRECISION A(N),DEXMAG,DRE,DIM
C
C TAKES EXPONENTIAL OF DOUBLE PRECISION COMPLEX ARRAY
C
      NHALF=N/2
      A(1)=DEXP(A(1))
      A(2)=DEXP(A(2))
      DO 10 I=2,NHALF
      DRE=A(2*I-1)
      DIM=A(2*I)
      DEXMAG=DEXP(A(2*I-1))
      A(2*I-1)=DEXMAG*DCOS(DIM)
   10 A(2*I)=DEXMAG*DSIN(DIM)
      RETURN
      END
      SUBROUTINE NULNEG(A,N)
      DOUBLE PRECISION A(N)
C
C SETS NEGATIVE TIME SAMPLES (IN TOP END OF ARRAY) TO ZERO
C
      NHP2=N/2+2
      DO 10 I=NHP2,N
   10 A(I)=0.0D0
      A(NHP2-1)=A(NHP2-1)*0.5D0
      RETURN
      END
```

```
      SUBROUTINE REFFT(A,R,N)
      DOUBLE PRECISION A,A1,A2,A3,A4,A5,A6,A7,A8,WR,WI,C,S,X
      DOUBLE PRECISION ARG,XN
      DIMENSION A(2,N)
      INTEGER R
      CALL RFT1(A,R,N)
      LIM1=N/2
      LIM2=LIM1-1
      A1=A(2,1)*0.5D0
      A2=A(2,N)*0.5D0
      A3=A(1,1)*0.5D0
      A4=A(1,N)*0.5D0
      XN=N
      ARG=3.14159265358979320D0/XN
      WR=DCOS(ARG)
      WI=-DSIN(ARG)
      C=1.0D0
      S=0.0D0
      DO 1 I=2,LIM1
      A5=A(1,I)*0.5D0
      A6=A(2,I)*0.5D0
      NMI=N-I+1
      A7=A(1,NMI)*0.5D0
      A8=A(2,NMI)*0.5D0
      X=C
      C=X*WR-S*WI
      S=S*WR+X*WI
      A(1,I)=A5+A7*C+A8*S
      A(2,I)=A6-A7*S+A8*C
      A(1,NMI)=A5-A7*C-A8*S
      A(2,NMI)=-A6-A7*S+A8*C
    1 CONTINUE
      DO 2 I=1,LIM2
      NMI=N-I+1
      NMMI=N-I
      A(1,NMI)=A(1,NMMI)
      A(2,NMI)=A(2,NMMI)
    2 CONTINUE
      A(1,LIM1+1)=A1
      A(2,LIM1+1)=A2
      A(1,1)=A3+A4
      A(2,1)=A3-A4
      DO 100 I=1,N
      A(1,I)=A(1,I)*2.0D0
  100 A(2,I)=A(2,I)*2.0D0
      RETURN
      END


      SUBROUTINE RFT1(A,R,N)
      DOUBLE PRECISION A,A1,A2,A3,A4,A5,A6
      DIMENSION A(2,N)
      INTEGER R
      N2=2*N
      CALL FFT(A,R,N2,1)
      LIM1=N/2
      LIM2=LIM1+1
      LIM3=N-1
```

```
      A1=A(2,1)
      A(2,1)=A(1,LIM2)
      A2=A(2,LIM2)
      DO 1 I=2,LIM1
      NMI=N-I+2
      A3=A(1,I)
      A4=A(2,I)
      A5=A(1,NMI)
      A6=A(2,NMI)
      A(1,I)=(A5+A3)*0.5D0
      A(2,I)=(A4-A6)*0.5D0
      A(1,NMI)=(A4+A6)*0.5D0
      A(2,NMI)=(A5-A3)*0.5D0
    1 CONTINUE
      DO 2 I=LIM2,LIM3
      A(1,I)=A(1,I+1)
      A(2,I)=A(2,I+1)
    2 CONTINUE
      A(1,N)=A1
      A(2,N)=A2
      RETURN
      END


      SUBROUTINE RFTINV(A,R,N)
      DOUBLE PRECISION A,A1,A2,A3,A4,A5,A6,A7,A8,C,S,WR,WI
      DOUBLE PRECISION ARG,X
      DIMENSION A(2,N)
      INTEGER R
      LIM1=N/2
      LIM2=LIM1+1
      LIM3=N-1
      A1=A(1,LIM2)*2.0D0
      A2=-A(2,LIM2)*2.0D0
      A3=A(1,1)
      A4=A(2,1)
      DO 1 I=LIM2,LIM3
      A(1,I)=A(1,I+1)
      A(2,I)=A(2,I+1)
    1 CONTINUE
      ARG=3.1415926535897932D0/FLOAT(N)
      WR=DCOS(ARG)
      WI=-DSIN(ARG)
      C=1.0D0
      S=0.0D0
      DO 2 I=2,LIM1
      NMI=N-I+1
      A5=A(1,I)
      A6=A(2,I)
      A7=A(1,NMI)
      A8=A(2,NMI)
      A(1,I)=A5+A7
      A(2,I)=A6-A8
      X=C
      C=X*WR-S*WI
      S=S*WR+X*WI
      A(1,NMI)=(A5-A7)*C-(A6+A8)*S
      A(2,NMI)=(A5-A7)*S+(A6+A8)*C
    2 CONTINUE
```

```
      A(1,1)=A3+A4
      A(1,N)=A3-A4
      A(2,1)=A1
      A(2,N)=-A2
      CALL RFT2(A,R,N)
      DO 100 I=1,N
      A(1,I)=A(1,I)*0.5D0
  100 A(2,I)=A(2,I)*0.5D0
      RETURN
      END



      SUBROUTINE RFT2(A,R,N)
      DOUBLE PRECISION A,A1,A2,A3,A5,A6,A7,A8
      DIMENSION A(2,N)
      INTEGER R
      LIM1=N/2
      A1=A(1,N)
      A2=A(2,N)
      DO 1 I=1,LIM1
      NMI=N-I+1
       A(1,NMI)=A(1,NMI-1)
      A(2,NMI)=A(2,NMI-1)
    1 CONTINUE
      A3=A(2,1)
      A(2,1)=A1
      A(2,LIM1+1)=A2
      A(1,LIM1+1)=A3
      DO 2 I=2,LIM1
      NMI=N-I+2
      A5=A(1,I)-A(2,NMI)
      A6=A(2,I)-A(1,NMI)
      A7=A(1,I)+A(2,NMI)
      A8=A(2,I)+A(1,NMI)
      A(1,I)=A5
      A(2,NMI)=-A6
      A(1,NMI)=A7
      A(2,I)=A8
    2 CONTINUE
      N2=2*N
      CALL FFT(A,R,N2,-1)
      RETURN
      END



      SUBROUTINE FFT(A,M,N2,MXD)
      DOUBLE PRECISION A,X,XN,XLE1,TR,TI,PI,UR,UI,WR,WI,AMXD
      DIMENSION A(N2)
      INTEGER MXD
C
C IF MXD IS 1 THE DFT IS OBTAINED, IF -1 THE INVERSE DFT IS OBTAINED
C
      N=2**M
      XN=N
      NV2=N/2
      NM1=N-1
      J=1
      DO 7 I=1,NM1
```

```
      IF(I.GE.J)GO TO 5
      TR=A(2*J-1)
      TI=A(2*J)
      A(2*J-1)=A(2*I-1)
      A(2*J)=A(2*I)
      A(2*I-1)=TR
      A(2*I)=TI
    5 K=NV2
    6 IF(K.GE.J)GO TO 7
      J=J-K
      K=K/2
      GO TO 6
    7 J=J+K
      PI=3.14159265358979320D0
      DO 20 L=1,M
      LE=2**L
      LE1=LE/2
      XLE1=LE1
      UR=1.0D0
      UI=0.0D0
      AMXD=MXD
      WR=DCOS(PI/XLE1)
      WI=AMXD*DSIN(PI/XLE1)
      DO 20 J=1,LE1
      DO 10 I=J,N,LE
      IP=I+LE1
      TR=A(2*IP-1)*UR-A(2*IP)*UI
      TI=A(2*IP)*UR+A(2*IP-1)*UI
      A(2*IP-1)=A(2*I-1)-TR
      A(2*IP)=A(2*I)-TI
      A(2*I-1)=A(2*I-1)+TR
   10 A(2*I)=A(2*I)+TI
      X=UR
      UR=X*WR-UI*WI
   20 UI=UI*WR+X*WI
      IF(MXD.EQ.1)GO TO 35
      XN=1.0D0/FLOAT(N)
      DO 30 I=1,N2
   30 A(I)=A(I)*XN
   35 RETURN
      END
```

3.2 Listing of Program for Computing M|G|1 Queue Busy Period Distribution

For subroutines REFFT and RFTINV , see Appendix 3.1.

The following subroutines were called from the Harris 500 Computer Subroutine Library GINOGRAF[92].

OPEN

AXIPOS

AXISCA

AXIDRA

GRAPOL

DEVEND

```
****************************************************************************
****************************************************************************
***********
** Pdn     7    2499-RK                    TERMINAL: 72                3 SEP 82 11:2
***********
****************************************************************************
****************************************************************************


C
C
C    *****************************************************************
C    *                                                               *
C    *                                                               *
C    * PROGRAM COMPUTES BUSY PERIOD PROBABILITY DENSITY FUNCTION FOR *
C    * M/G/1 QUEUE BY USE OF AN ITERATIVE METHOD. THIS METHOD BEGINS *
C    * WITH BILINEAR TRANSFORM OF SERVICE TIME PDF AND THE BILINEAR  *
C    * TRANSFORM OF A DELTA FUNCTION AT T=0 (I.E. AN IDLE SYSTEM ).  *
C    * FROM THIS THE BILINEAR TRANSFORM OF BUSY PERIOD PDF FOR N=1,2, *
C    * 3,..., IS COMPUTED. THIS IS REPEATED UNTIL THE BILINEAR TRAN- *
C    * SFORM OF BUSY PERIOD PDF CONVERGES. THE INVERSE BILINEAR      *
C    * TRANSFORM IS THEN COMPUTED BY USE OF SMITH'S ALGORITHM.       *
C    *                                                               *
C    *                                                               *
C    *****************************************************************
C
C
      DOUBLE PRECISION B(400),GT(200),GR(200),G(2048),GTEMP(2048),
     *ER(200),MU,T,A,TIME,L,CHECK,TVAL,AKP,UT,DIF,SUM
      REAL*6 XTR(200),TR(200)*6,S*6,S1*6
      DATA NOUT/16/
      S=1.0E-9
      S1=4.0D0
      NINTV=4
      NT=199
      T=DBLE(S1)/DFLOAT(NT-1)
      L=1.0D0
      MU=10.0D0
      MB=170
      UT=L/MU
      A=1.0D0
      M=8
      N=2**M
      MBP=112
      NHALF=N/2
      WRITE(NOUT,5)
    5 FORMAT(//,5X,'THE BUSY PERIOD PROBABILITY DENSITY FUNCTION FOR THE
     * M/G/1 QUEUEING SYSTEM COMPUTED FROM THE BILINEAR TRANSFORM OF THE
     *'/,5X,'SERVICE TIME PDF'/120(1H-))
C
C COMPUTE BILINEAR TRANSFORM FROM LAPLACE TRANSFORM EXPRESSION
C OF SERVICE TIME P.D.F.
C
      CALL BTSERV(B,A,MU,MB)
      WRITE(NOUT,10)
   10 FORMAT(/,10X,'BILINEAR TRANSFORM OF SERVICE TIME P.D.F.'/)
      CALL OUTPUT(B,MB)
C
C GENERATE INITIAL BILINEAR TRANSFORM OF BUSY PERIOD P.D.F. AND OBTAIN
```

```
C
      CALL IBPPDF(G,N)
      CALL REFFT(G,M-1,NHALF)
C
C SCALE AND DRAW AXES FOR GRAPH
C
      GTEMP(1)=0.0D0
      CALL OPEN
      CALL AXIPOS(1,50.0,50.0,120.0,1)
      CALL AXIPOS(1,50.0,50.0,200.0,2)
      CALL AXISCA(3,NINTV,0.0,S1,1)
      CALL AXISCA(4,10,S,1.0,2)
      CALL AXIDRA(1,1,1)
      CALL AXIDRA(-1,-1,2)
C
C NOW DO THE ITERATIONS TO COMPUTE BILINEAR TRANSFORM OF BUSY PERIOD
C
      NITS=0
   49 NITS=NITS+1
      CHECK=GTEMP(1)
C
C COMPUTE D.F.T. OF BILINEAR TRANSFORM OF BUSY PERIOD P.D.F.
C
      CALL DFTPDF(G,B,A,L,N,MB)
C
C MOVE CONTENTS OF G TO GTEMP TO AVOID TRANSFORMING BACK
C
      CALL MOVE(GTEMP,G,N)
C
C COMPUTE INVERSE D.F.T. TO OBTAIN BILINEAR TRANSFORM OF BUSY PERIOD
C
      CALL RFTINV(GTEMP,M-1,NHALF)
C
C CHECK FOR CONVERGENCE OF BILINEAR TRANSFORM SEQUENCE OF BUSY PERIOD
C PDF
C
      DIF=CHECK-GTEMP(1)
      IF(DABS(DIF).GE.1.0D-8)GO TO 49
C
C COMPUTE BILINEAR TRANSFORM OF BUSY PERIOD CUMMULATIVE DISTRIBUTION
C FUNCTION BY PROCESSING SEQUENCE BY A DIGITAL FILTER WITH TRANSFER
C FUNCTION (1+1/Z)/(A(1-1/Z))
C
   50 CALL DIGFIL(GTEMP,N,A)
      WRITE(NOUT,55)
   55 FORMAT(//,10X,'BILINEAR TRANSFORM OF BUSY PERIOD DISTRN'/)
      CALL OUTPUT(GTEMP,N)
C
C COMPUTE INVERSE BILINEAR TRANSFORM USING SMITH'S ALGORITHM
C
      DO 60 KGB=2,NT
      TR(KGB)=T*DFLOAT(KGB-1)
      TIME=T*(KGB-1)
      CALL SUMDLA(GTEMP,N,A,TIME,TVAL)
   60 GT(KGB)=2.0D0*A*DEXP(-A*TIME)*TVAL
      TR(1)=0.0
      SUM=0.0D0
      DO 70 KP=1,N
      AKP=KP
      SUM=SUM+((-1.0D0)**KP)*AKP*GTEMP(KP)
```

```
   70 CONTINUE
      GT(1)=2.0D0*A*SUM
      WRITE(NOUT,100)GTEMP(1),GTEMP(N)
  100 FORMAT(//,10X,'1ST & NTH ELEMENT OF BILINEAR TRANSFORM OF BUSY PER
     *IOD PROBABILITY DENSITY FUNCTION'/2F25.15/)
      WRITE(NOUT,102)A
  102 FORMAT(/,10X,'BILINEAR PARAMETER ALPHA=',F10.4)
      WRITE(NOUT,103)MB
  103 FORMAT(//,10X,'NUMBER OF BILINEAR ELEMENTS USED ',I7)
      WRITE(NOUT,107)T
  107 FORMAT(//,10X,'TIME INTERVAL BETWEEN COMPUTED SAMPLES',F15.6/)
      WRITE(NOUT,110)N
  110 FORMAT(/,10X,'BLOCK SIZE OF DFT FOR THIS ITERATION',I7)
      WRITE(NOUT,120)NITS
  120 FORMAT(/,10X,'NUMBER OF ITERATIONS PERFORMED',I7)
      WRITE(NOUT,130)UT,MU
  130 FORMAT(/,10X,'UTILISATION FACTOR AND AVERAGE SERVICE TIME',2F15.6)
      CALL COMPLF(GT,NT)
      CALL DATAGR(XTR,GT,NT,JX,S)
      CALL GRAPOL(TR,XTR,JX)
C
C COMPUTE P.D.F. FROM ANALYTIC SOLUTION
C
      CALL BUSYFN(GR,NT,L,MU,T)
C
C COMPUTE ERROR BETWEEN ITERATIVE METHOD AND ANALYTIC SOLUTION
C
      DO 135 KJ=1,NT
  135 ER(KJ)=GR(KJ)-GT(KJ)
      WRITE(NOUT,140)
  140 FORMAT(//,5X,'COMPUTED BUSY PERIOD',18X,'BUSY PERIOD FROM SOLN',
     *18X,'ERROR '/100(1H-))
      DO 160 KGB=1,NT
      WRITE(NOUT,150)GT(KGB),GR(KGB),ER(KGB)
  150 FORMAT(F25.15,10X,F25.15,10X,F25.15)
  160 CONTINUE
C
C GENERATE DATA FOR GRAPH; POSITION, SCALE AND DRAW AXES; AND DRAW GRAPH
C
      CALL COMPLF(GR,NT)
      CALL DATAGR(XTR,GR,NT,JX,S)
      CALL GRAPOL(TR,XTR,JX)
      CALL DEVEND
      STOP
      END



      SUBROUTINE REPDF(G,N,L,MU,T)
      DOUBLE PRECISION G(N),L,MU,UT,PROD,SUM,A1,AM,AKGB,AMP1
      DOUBLE PRECISION A(100),AK,T,AI,AMM,AMMP1
      DOUBLE PRECISION SUM1,SUM2
      DOUBLE PRECISION AKXX
      MAX=40
      UT=L/MU
      A1=DSQRT(L*MU)
      A(1)=A1/(L+MU)
      DO 5 M=1,MAX
      AMM=2*(M-1)+1
```

```
      AMMP1=AMM+1.ODO
      AM=M
      AMP1=AM+1.ODO
    5 A(M+1)=AMM*AMMP1*((A1/(L+MU))**2.ODO)*A(M)/(AM*AMP1)
      DO 40 I=2,N
      AI=(I-1)*T*(L+MU)
      SUM2=(1.ODO-DEXP(-AI))*A(1)/DSQRT(UT)
      SUM=0.ODO
      DO 30 M=1,MAX
      MT2=M*2
      SUM1=1.ODO
      DO 20 K=1,MT2
      PROD=1.ODO
      KP1=K+1
      DO 10 KXX=2,KP1
      AKXX=KXX-1
   10 PROD=AI*PROD/AKXX
      SUM1=SUM1+PROD
   20 CONTINUE
      SUM=(1.ODO-DEXP(-AI)*SUM1)*A(M+1)+SUM
   30 CONTINUE
      SUM=SUM/DSQRT(UT)
      G(I)=SUM+SUM2
   40 CONTINUE
      G(1)=0.ODO
      RETURN
      END



      SUBROUTINE DATAGR(XTR,XT,NT,JX,S)
      DOUBLE PRECISION XT(NT),S1
      REAL*6 XTR(NT),S*6
      INTEGER JX
      S1=S
      JX=0
   20 JX=JX+1
      IF(JX.GT.NT)GO TO 30
      IF(XT(JX).LE.0.ODO.OR.XT(JX).LE.S1)GO TO 30
      XTR(JX)=XT(JX)
      GO TO 20
   30 IF(XT(JX).LE.0.ODO)XTR(JX)=S
      IF(XT(JX).GT.0.ODO.AND.XT(JX).LT.S1)JX=JX-1
      IF(JX.GT.NT)JX=NT
      RETURN
      END



      SUBROUTINE BUSYFN(G,N,L,M,T)
      DOUBLE PRECISION G(N),L,M,S1,S2,AMP1,AM,BMP1,BM,T,CON
      DOUBLE PRECISION*12 GMP1,GM*12,X*12
      DOUBLE PRECISION*12 GML,GMPL*12,XG1*12,XG2*12
      NTERM=100
      DO 30 K=2,N
      AM=DSQRT(L*M)/(L+M)
      CON=AM**2.ODO
      X=T*DFLOAT(K-1)*(L+M)
      GM=DEXP(-X)
      BM=GM*AM
```

```
      GML=-X
      S1=BM
      S2=AM
      DO 20 MXX=1,NTERM
      MX=MXX-1
      AMP1=AM*DFLOAT((2*MX+1)*(2*MX+2))*CON/DFLOAT((MX+1)*(MX+2))
      XG1=DFLOAT(2*MX+1)*DFLOAT(2*MX+2)
      XG2=(DFLOAT(2*MX+2)+X)/(DFLOAT(2*MX)+X)
      GMPL=2.0D0*DLOG(X)-DLOG(XG1)+GML+DLOG(XG2)
      IF(GMPL.LT.60.0D0)THEN
      GMP1=DEXP(GMPL)
      ELSE
      WRITE(16,221)
221   FORMAT(//,10X,'UNREASONABLE RESULTS!!!!!!'///)
      STOP
      ENDIF
      BMP1=BM*DFLOAT((2*MX+1)*(2*MX+2))*CON/DFLOAT((MX+1)*(MX+2))+
     %GMP1*AMP1
      BM=BMP1
      GML=GMPL
      AM=AMP1
      S1=S1+BMP1
 20   S2=S2+AMP1
 30   G(K)=(S2-S1)/DSQRT(L/M)
      G(1)=0.0D0
      WRITE(16,40)AMP1,BMP1
 40   FORMAT(//,10X,'AMP1 AND BMP1',2E25.10)
      RETURN
      END


      SUBROUTINE OUTPUT(A,N)
      DOUBLE PRECISION A(N)
      M=N
      IF(M.GT.200)M=200
      WRITE(16,10)(A(I),I=1,M)
 10   FORMAT(5F25.15)
      WRITE(16,20)A(N)
 20   FORMAT(//,10X,'LAST ELEMENT A(N)',F25.15/)
      RETURN
      END


      SUBROUTINE MOVE(A,B,N)
      DOUBLE PRECISION A(N),B(N)
      DO 10 K=1,N
 10   A(K)=B(K)
      RETURN
      END


      SUBROUTINE DIGFIL(G,N,A)
      DOUBLE PRECISION G(N),SUM,A
      SUM=0.0D0
      G(1)=1.0D0-G(1)
      DO 5 K=2,N
 5    G(K)=-G(K)
```

```
      DO 10 K=1,N
      SUM=SUM+G(K)
   10 G(K)=SUM/A
      DO 20 K=2,N
   20 G(N-K+2)=G(N-K+2)+G(N-K+1)
      RETURN
      END


      SUBROUTINE COMPLF(G,N)
      DOUBLE PRECISION G(N)
      DO 20 K=1,N
   20 G(K)=1.0D0-G(K)
      RETURN
      END


      SUBROUTINE BTSERV(B,AB,MU,MB)
      DOUBLE PRECISION B(MB),AB,MU,A1
      A1=MU/(AB+MU)
      DO 10 K=2,MB
   10 B(K)=A1*2.0D0*(((AB-MU)/(AB+MU))**(K-2))*AB/(AB+MU)
      B(1)=A1
      RETURN
      END


      SUBROUTINE IBPPDF(G,N)
      DOUBLE PRECISION G(N),AN
      DO 10 K=1,N
   10 G(K)=0.0D0
      AN=0.5D0
      G(1)=1.0D0
      RETURN
      END


      SUBROUTINE DFTPDF(G,B,A,L,N,MM)
      DOUBLE PRECISION G(N),B(MM),A,T,S,GM,BT,CNM,DNM,ENM,FNM,PI,ARG,A1,
     *A2,A3,A4,AM,AN,AK,SUMR,SUMI,L,RNK,WR,WI
      AN=N
      PI=3.14159265358979320D0
      NH=N/2
      DO 60 M=2,NH
      AM=M-1
      ARG=2.0D0*PI*AM/AN
      A1=DCOS(ARG)
       A2=DSIN(ARG)
      T=G(2*M-1)
      S=G(2*M)
      CNM=(2.0D0*A+L*T-L)*A1+L*(T-1.0D0-S*A2)
      DNM=(2.0D0*A+L*T-L)*A2+L*S*(1.0D0+A1)
      ENM=L*(1.0D0-T)*A1+(2.0D0*A+L-L*T)+L*S*A2
      FNM=L*(1.0D0-T)*A2-L*S*(1.0D0+A1)
      A3=DABS(ENM)**2.0D0+DABS(FNM)**2.0D0
      A4=DABS(CNM)**2.0D0+DABS(DNM)**2.0D0
      RNK=DSQRT(A4/A3)
```

```
      GM=DATAN((DNM*ENM-CNM*FNM)/(CNM*ENM+DNM*FNM))
      IF(GM.LT.0.0D0)GM=PI+GM
      SUMR=0.0D0
      SUMI=0.0D0
      DO 50 K=1,MM
      AK=K-1
      BT=RNK**AK
      WR=DCOS(AK*GM)
      WI=DSIN(AK*GM)
      SUMR=SUMR+B(K)*WR*BT
      SUMI=SUMI+B(K)*WI*BT
   50 CONTINUE
      G(2*M-1)=SUMR
      G(2*M)=SUMI
   60 CONTINUE
      A1=0.0D0
      A2=0.0D0
      DO 70 KJ=1,MM
      A1=A1+B(KJ)
   70 A2=A2+((-1.0D0)**(KJ-1))*B(KJ)
      G(1)=A1
      G(2)=A2
      RETURN
      END


      SUBROUTINE SUMDLA(A,N,LAMBDA,T,PDF)
C
C SUBROUTINE USES SMITH'S ALGORITHM TO SUM THE DIFFERENTIATED
C LAGUERRE POLYNOMIAL SERIES FOR THE INVERSE BILINEAR TRANSFORM
C
      DOUBLE PRECISION A(N),PDF,LAMBDA,T,SK,SKP1,SKP2,SDK,SDKP1,
     *SDKP2,AK,AKP1,AKP2,A1
      SK=0.0D0
      SKP1=0.0D0
      SKP2=0.0D0
      SDK=0.0D0
      SDKP1=0.0D0
      SDKP2=0.0D0
      NM1=N-1
      DO 20 K=1,NM1
      AK=N-K
      AKP1=AK+1.0D0
      AKP2=AK+2.0D0
      SKP2=SKP1
      SKP1=SK
      A1=2.0D0*AK+1.0D0-2.0D0*LAMBDA*T
      SK=((-1.0D0)**(N-K))*A(N-K+1)+A1*SKP1/AKP1-AKP1*SKP2/AKP2
      SDKP2=SDKP1
      SDKP1=SDK
      A1=2.0D0*AK-1.0D0-2.0D0*LAMBDA*T
      SDK=-SK/AK+A1*SDKP1/AK-AK*SDKP2/AKP1
   20 CONTINUE
      PDF=SDK
      RETURN
      END
EOF..
```

REFERENCES

1. Bhat, U. N., and Shalaby, M., "Approximation techniques in the solution of queueing problems", Naval Research Logistics Quarterly, Volume 26, 1979, pp. 311-326.

2. Kleinrock, L., Queueing Systems, Volume 1:Theory, Wiley, New York, 1975.

3. Kleinrock, L., Queueing Systems, Volume 2:Computer Applications, Wiley, New York, 1975.

4. Kleinrock, L., Communications Nets: Stochastic Message Flow and Delay, McGraw-Hill, U.S.A., 1964.

5. Schwartz, M., Computer-Communication Network Design and Analysis, Prentice-Hall, Englewood Cliffs. 1977.

6. Bear, D., Principles of Telecommunication-Traffic Engineering, Peter Peregrinus, Stevenage, 1976.

7. Cox, D. R., and Smith, W. L., Queues, Methuen, London, 1961.

8. Kobayashi, H., Modelling and Analysis: An Introduction to System Performance Evaluation Methodology, Addison-Wesley, Philippines, 1978.

9. Crommelin, C. D., "Delay probability formulae", The Post Office Electrical Engineers' Journal, Volume 26, 1934, pp. 266-274.

10. Georganas, N. D., "Buffer behaviour with Poisson arrivals and bulk geometric service", IEEE Transactions on Communications, Volume, COM-24, Number 8, August 1976, pp. 938-940.

11. Konheim, A. G., and Meister, B., "Service in a loop system", Journal of the Association for Computing Machinery, Volume 19, Number 1, January 1972, pp. 92-108.

12. Hsu, J., "Buffer behaviour with Poisson arrival and geometric output processes", IEEE Transactions on Communications, Volume COM-22, Number 12, December 1974, pp. 1940-1941.

13. Janakiraman, N., Pagurek, B., and Neilson, J. E., "Multiplexing low-speed buffered data terminals", IEEE Transactions on Communications, Volume COM-28, Number 10, October 1980, pp. 1838-1843.

14. Chu, W. W., "Buffer behaviour for Poisson arrivals and multiple synchronous constant outputs", IEEE Transactions on Computers, Volume C-19, Number 6, June 1970, pp. 530-534.

15. Chu, W. W., "Buffer behaviour for batch Poisson arrivals and single constant output", IEEE Transactions on Communication Technology, Volume COM-18, Number 5, October 1970, pp. 613-618.

16. Raychaudhuri, D., and Rappaport, S. S., "Sampled multiserver queues with general arrivals and deterministic service time", IEE Proceedings, Volume 127, Part E, Number 3, May 1980, pp. 88-92.

17. Tzafestas, S. G., "Buffer length for Erlang arrivals and multiple synchronous regular removals", Electronics Letters, Volume 7, Number 8, April 1971, pp. 176-178.

18. Fischer, M. J., and Harris, T. C., "A model for evaluating the performance of an integrated circuit- and packet-switched multiplex structure", IEEE Transactions on Communications, Volume COM-24, Number 2, February 1976, pp. 195-202.

19. Maglaris, B., and Schwartz, M., "Performance evaluation of a variable frame multiplexer for integrated switched networks", IEEE Transactions on Communications, Volume COM-29, Number 6, June 1981, pp. 800-807.

20. Kekre, H. B., and Khalid, M., "Buffer analysis with multiservers having binomial distribution", IEE Proceedings, Volume 128, Part E, Number 2, March 1981, pp. 67-73.

21. Gruber, J. G., "Delay related issues in integrated voice and data networks", IEEE Transactions on Communications, Volume COM-29, Number 6, June 1981, pp. 786-799.

22. Jung, M. M., "Busy period distribution in an SPC processor having a clock-pulse operated gate", Philips Telecommunication Review, Volume 38, Number 2, April 1980, pp. 84-89.

23. Green, L., "A queueing system in which customers require a random number of servers", Operations Research, Volume 28, Number 6, November-December 1980, pp. 1335-1346.

24. Kleinrock, L., "Performance models and measurements of the ARPA computer network", in Computer Communication Networks, Grimsdale, R. L., and Juo, F. F. (Editors), Noordhoff, Leyden, 1975, pp. 63-87.

25. Tobagi, F. A., Gerla, M., Peebles, R. W., and Manning, E. G., "Modelling and measurement techniques in packet communication networks", Proceedings of the IEEE, Volume 66, Number 11, November 1978, pp. 1423-1447.

26. Wong, J. W., "Queueing network modelling of computer communication networks", Association for Computing Machinery Computing Surveys, Volume 10, Number 3, September 1978, pp. 343-351.

27. Chu, W. W., and Konheim, A. G., "On the analysis and modelling of a class of computer communication systems", IEEE Transactions on Communications, Volume COM-20, Number 3, June 1972, pp. 645-660.

28. Kobayashi, H., and Konheim, A. G., "Queueing models for computer communications system analysis", IEEE Transactions on Communications, Volume COM-25, Number 1, January 1977, pp. 2-29.

29. Holman, D. F., Grassmann, W. K., and Chaudhry, M. L., "Some results of the queueing system $E_K^X|M|c$ ", Naval Research Logistics Quarterly, Volume 27, 1980, pp. 217-222.

30. Holman, D. F., Grassman, W. K., and Chaudhry, M. L., "Some results of the queueing system $E_K^X|M|c$", full version of (29), available from authors.

31. Calo, S. B., "Delay analysis of a two-queue, nonuniform message channel", IBM Journal of Research and Development, Volume 25, Number 6, November 1981, pp. 915-929.

32. Chan, W. C., Maa, D. Y., and Chung, W. K., "Discrete-time queueing system with instantaneous defection and service interruption", Proceedings of the IEE, Volume 122, Number 12, December 1975, pp. 1372-1375.

33. Morrison, J. A., "Two discrete-time queues in tandem", IEEE Transactions on Communications, Volume COM-27, Number 3, March 1979, pp. 563-573.

34. Rubin, I., "Message delays in FDMA and TDMA communication channels", IEEE Transactions on Communications, Volume COM-27, Number 5, May 1979, pp. 769-777.

35. Eckberg, Jr., A. E , "The single server queue with periodic arrival process and deterministic service times", IEEE Transactions on Communications, Volume COM-27, Number 3, March 1979, pp. 556-562.

36. Choo, Q. H., and Conolly, B., "Waiting time analysis for a tandem queue with correlated service", European Journal of Operations Research, Volume 4, Number 5, May 1980, pp. 337-345.

37. Swartz, G. B., "Polling in a loop system", Journal of the Association for Computing Machinery, Volume 27, Number 1, January 1980, pp. 42-59.

38. Newell, G. F., Applications of Queueing Theory, Chapman and Hall, London, 1971.

39. Kobayashi, H., "Application of the diffusion approximation to queueing networks I: Equilibrium queue distributions", Journal of the Association for Computing Machinery, Volume 21, Number 2, April 1974, pp. 316-328.

40. Kobayashi, H., "Application of the diffusion approximation to queueing networks II:Nonequilibrium distributions and applications to computer modelling", Journal of the Association for Computing Machinery, Volume 21, Number 3, July 1974, pp. 459-469.

41. Bear, D., "An approximate method for calculating delay distributions for queueing systems with Poisson input and arbitrary holding time distributions", Proceedings of the 8th International Teletraffic Congress, Melbourne, 1976.

42. Fredericks, A. A., "A class of approximations for the waiting time distribution in a GI|G|1 queueing system", Bell System Technical Journal, Volume 61, Number 3, March 1982, pp. 295-325.

43. Cosmetatos, G. P., and Godsave, S. A., "Approximations in the multi-server queue with hyper-exponential inter-arrival times and exponential service times", Journal of the Operational Research Society, Volume 31, Number 1, January 1980, pp. 57-62.

44. Bux, W., and Herzog, U., "The phase concept:approximation of measured data and performance analysis", Proceedings of the International Symposium on Computer Performance, Modelling, Measurement, and Evaluation; Yorktown Heights; North-Holland; New York; 1977, pp. 23-38.

45. Ackroyd, M. H., "Computing the waiting time distribution for the G|G|1 queue by signal processing methods", IEEE Transactions on Communications, Volume COM-28, Number 1, January 1980, pp. 52-58.

46. Ackroyd, M. H., "Numerical computation of the busy period distribution in M|G|1 queues", unpublished note, April 1982.

47. Molloy, M. K., "An application of the discrete Fourier transform to the FCFS GI|G|1 queue", Modelling and Simulation Proceedings of the 11th Annual Pittsburgh Conference, Instrument Society of America, Volume 11, Part 2, May 1980, pp. 755-759.

48. Wong, D. K., "A discrete approximation for G|G|1 queue ", M.Sc. Thesis, Computer Science Department, School of Engineering and Applied Sciences, University of California in Los Angeles, December 1974.

49. Skinner, C. E., "A numerical inversion of the Pollaczek-Khinchine formula", Presented at the 33rd National Meeting of the Operations Research Society of America, San Francisco, May 1968.

50. Greenberg, I., "Distribution-free analysis of M|G|1 and G|M|1 queues", Operations Research, Volume 21, Number 2, March-April 1973, pp. 629-235.

51. Ackroyd, M. H., and Kanyangarara, R., "Skinner's method for computing bounds on distributions and the numerical solution of continuous-time queueing problems", IEEE Transactions on Communications, Volume COM-30, Number 7, July, 1982, pp. 1746-1749.

52. Bagchi, T. P., and Templeton, J. G. C., Numerical Methods in Markov Chains and Bulk Queues, Lecture Notes in Economics and Mathematical Systems, Springer-Verlug, Berlin and New York, 1972.

53. Gaver, Jr., D. P., "Observing stochastic processes, and approximate transform inversion", Operations Research, Volume 14, Number 3, May-June 1966, pp. 444-459.

54. Piessens, R., and Branders, M., "Numerical inversion of the Laplace transform using generalised Laguerre polynomials", Proceedings of the IEE, Volume 118, Number 10, October 1971, pp. 1517-1522.

55. Stehfest, H., "Algorithm 368: Numerical inversion of the Laplace transforms", Communications of the Association for Computing Machinery, Volume 1, Number 1, January 1970, pp. 47-49.

56. Chiu, R. F., Chen, C. F., and Huang, C. J., "A new method for the inverse Laplace transformation via the fast Fourier transform", 22nd Annual Southwestern IEEE Conference Record, Dallas, Texas, April 1970, pp. 201-203.

57. Nototný, B., "An approximate Laplace transform inversion using exponential series representation", International Journal for Numerical Methods in Engineering, Volume 15, Number 2, 1980, pp. 291-295.

58. Dubner, H., and Abate, J.,"Numerical inversion of the Laplace transforms by relating them to the finite Fourier cosine transform", Journal of the Association for Computing Machinery, Volume 15, Number 1, January 1968, pp. 115-123.

59. Weeks, W. T.,"Numerical inversion of Laplace transforms using Laguerre functions", Journal of the Association for Computing Machinery, Volume 13, Number 3, July 1966, pp. 419-426.

60. Nance, R. E.,Bhat, U. N., and Claybrook, B. G., "Busy period analysis of a time-sharing system: transform inversion", Journal of the Association for Computing Machinery, Volume 19, Number 3, July 1972, pp. 453-463.

61. Abate, J., Dubner, H., and Weinberg, S. B., "Queueing analysis of the IBM 2314 disk storage facility", Journal of the Association for Computing Machinery, Volume 15, Number 4, October 1968, pp. 577-589.

62. Ackroyd, M. H., "Iterative computation of the M|G|1 queue length distribution via the discrete Fourier transform", IEEE Transactions on Communications, Volume COM-28, Number 11, November 1980, pp. 1929-1932.

63. Ackroyd, M. H., "M|G|1 queue analysis via the inverse Poisson transform", IEE Proceedings, Volume 129, Part E, Number 3, May 1982, pp. 119-122.

64. Ackroyd, M. H., "Numerical inversion of the Pollaczek-Khinchine transform equation for the number in the M|G|1 system", Unpublished Note, March 1982.

65. Ackroyd, M. H., and Kanyangarara, R., "The Poisson transform in signal processing and queue analysis", International Conference on Digital Processing of Signals in Communications, Loughborough, IERE Conference Proceedings, April 1981, pp. 227-236.

66. Kanyangarara, R., "Digital signal processing methods applied to the M|G|1 queue", M.Sc. project, University of Aston in Birmingham, Department of Electrical and Electronic Engineering, December 1979.

67. Cavers, J. K., "On the fast Fourier transform inversion of probability generating functions", Journal of the Institute of Mathematics and its Applications, Volume 22, Number 3, November 1978, pp. 275-282.

68. Ackroyd, M. H., "M|M|1 transient state occupancy probabilities via the discrete Fourier transform", IEEE Transactions on Communications, Volume COM-30, Number 3, March 1982, pp. 557-559.

69. Stern, T. E., "Approximations of queue dynamics and their application to adaptive routing in computer communication networks", IEEE Transactions on Communications, Volume COM-27, Number 9, September 1979, pp. 1331-1335.

70. Royal Signals and Radar Establishment (Malvern), "Pilot Packet Switched Network:Network control and routing", Document D3, January 1980, Unpublished report.

71. Harris Fortran Reference Manual, Harris Corporation, Computer Systems Division, 1978.

72. Childers, D. , and Durling, A., Digital Filtering and Signal Processing, West, USA, 1975.

73. Rabiner, L. R., and Gold, B., Theory and Application of Digital Signal Processing, Prentice-Hall, Englewood Cliffs, 1975.

74. Oppenheim, A. V., and Schafer, R. W., Digital Signal Processing, Prentice-Hall, Englewood Cliffs, 1975.

75. Numerical Algorithms Group, Fortran Library Manual Mark 8, Volume 1 and 2, 1981.

76. Brady, P. T., "A statistical analysis of on-off pattern in 16 conversations", Bell System Technical Journal, Volume 47, Number 1, January 1968, pp. 73-91.

77. Bolgiano, Jr., L. P., and Piovoso, M. J., "Poisson transform signal analysis", IEEE Transactions on Information Theory, Volume IT-14, Number 4, July 1968, pp. 600-601.

78. Bolgiano, Jr., L. P., and Quick, L. T., "Obtaining digital signals for Poisson filtering from periodic samples", Proceedings of the IEEE, Volume 63, Number 9, September 1975, pp. 1366-1367.

79. Fairman, F. W., "Periodic sampling and the Poisson transform", Electronics Letters, Volume 5, Number 22, October 1969, pp. 569-570.

80. Bolgiano, Jr.,L. P., and Piovoso, M. J., "Relationship of Poisson transform to Laguerre expansions", IEEE Transactions on Computers, Volume C-18, Number 12, December 1969, pp. 1128-1131.

81. Oppenheim, A. V., and Johnson, D. H., "Discrete representation of signals", Proceedings of the IEEE, Volume 60, Number 6, June 1972, pp. 681-691.

82. The University of Aston Computer Centre, User's Handbook: Fortran, February 1976.

83. Freeman, J., Discrete-Time Systems: An Introduction to the Theory, John Wiley and Sons, New York, 1965.

84. Karwoski, R. J., Introduction to the z-transform and its Derivation; LSI Publication TP6A-8/81; TRW LSI Products, Box 2472, La Jolla, CA 92038; August 1981.

85. Smith, F. J., "An algorithm for summing orthogonal polynomial series and their derivatives with applications to curve-fitting and interpolation", Mathematics of Computation, Volume 19, 1965, pp. 33-36.

86. Gow, M. M., A Course in Pure Mathematics, English Universities Press, London, 1960.

87. Spiegel, M. R., Schaum's Outline of Theory and Problems of Laplace Transforms, McGraw-Hill, USA, 1965.

88. Heimann, D., and Neuts, M. F., "The single server queue in discrete time - numerical analysis IV", Naval Research Logistics Quarterly, Volume 20, Number 4, December 1973, pp. 753-766.

89. Yum, T., "Measuring the utilisation of a synchronous data link: an application of busy-period analysis", Bell System Technical Journal, Volume 59, Number 5, May-June 1980, pp. 731-744.

90. Department of Electrical Engineering Science, Mathematical Data for Electronics, University of Essex, 3rd Edition, October 1978.

91. Prabhu, N. U., Queues and Inventories: A Study of their Basic Stochastic Processes, John Wiley and Sons, New York, 1965.

92. Computer Aided Design Centre, GINOGRAF User Manual, Issue 1, CAD Centre, Cambridge, 1976