AMERICAN UNIVERSITY OF BEIRUT

GENERATION OF RANDOM VARIATES FOR PHDS, MAPS AND BMAPS

by
FARAH NASSER GHIZZAWI

A thesis
submitted in partial fulfillment of the requirements
for the degree of Master of Engineering
to the Department of Industrial Engineering and Management
of the Faculty of Engineering and Architecture
at the American University of Beirut

Beirut, Lebanon
May 2016

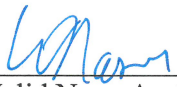
AMERICAN UNIVERSITY OF BEIRUT

GENERATION OF RANDOM VARIATES FOR PHDS, MAPS AND BMAPS

by

FARAH NASSER GHIZZAWI

Approved by:



Dr. Walid Nasr, Assistant Professor — Advisor
Industrial Engineering and Management

Dr. Bacel Maddah, Associate Professor and Chair — Member of Committee
Industrial Engineering and Management

Dr. Hussein Tarhini, Assistant Professor — Member of Committee
Industrial Engineering and Management

Date of thesis defense: April 15, 2016

AMERICAN UNIVERSITY OF BEIRUT

THESIS, DISSERTATION, PROJECT RELEASE FORM

Student Name: __ Ghizzawi_____Farah_____Nasser__
Last First Middle

☑ Master's Thesis ○ Master's Project ○ Doctoral Dissertation

☑ I authorize the American University of Beirut to: (a) reproduce hard or electronic copies of my thesis, dissertation, or project; (b) include such copies in the archives and digital repositories of the University; and (c) make freely available such copies to third parties for research or educational purposes.

☐ I authorize the American University of Beirut, **three years after the date of submitting my thesis, dissertation, or project,** to: (a) reproduce hard or electronic copies of it; (b) include such copies in the archives and digital repositories of the University; and (c) make freely available such copies to third parties for research or educational purposes.

Signature Date

May 3, 2016

ACKNOWLEDGEMENTS

I would like to acknowledge the support of my thesis advisor and the initiator of this research, Dr. Walid Nasr who gave me the opportunity to expand my engineering knowledge and expertise.

AN ABSTRACT OF THE THESIS OF

Farah Nasser Ghizzawi for Master of Engineering
Major: Engineering Management

Title: Generation of Random Variates for PHDs, MAPs and BMAPs

This research proposes the simulation of three arrival processes: *phase-type (PH) process/distribution, Markovian arrival process (MAP)* and *batch Markovian arrival process (BMAP)*. Two simulation models are developed and utilized to randomly generate inter-arrival times, i.e. the time headway between two successive event occurrences.

PHDs, MAPs and BMAPs do not belong to the distributions or stochastic processes that are commonly used in simulation tools, but it is usually straightforward to integrate them into simulation software by drawing on the underlying Markov chains which govern the activity of these processes.

Building stochastic simulation models based on the underlying Markov chain becomes extensive and error-prone for processes with higher orders, lagging in both time and traceability. Therefore, an alternative approach to simulating these processes is proposed, such that only the start and end states of an arrival epoch, rather than its whole transition activity, are utilized to set a cumulative distribution for the inter-arrival time. Since the inter-arrival time understudy is a matrix exponential, then the corresponding cumulative distribution function cannot be inverted and the classical inverse transform method cannot be applied. In this context, discretization of the function is an appropriate alternative whereby a database of inter-arrival times and their corresponding cumulative probabilities is formulated and then randomly sampled to generate inter-event times.

The scope of work comprises of conducting: (1) the simulation of the underlying Markov chain such that arrivals and their corresponding arrival times are recorded and alternatively (2) the discretization of the cumulative distribution function indicated by the start and end states of arrival epochs and random sampling of the latter to produce random inter-arrival times. Approaches (1) and (2) are applied on several examples and compared in terms of accuracy and efficiency.

Results suggest that the approach (2) is capable of performing as accurately and efficiently as its counterpart. Yet, they also show that simulating the underlying Markov chain is generally faster for fully populated MAPs and BMAPs.

CONTENTS

LIST OF ILLUSTRATIONS

LIST OF TABLES

CHAPTER I

INTRODUCTION

This research proposes the simulation of three arrival processes: phase-type (PH) process/distribution, Markovian arrival process (MAP) and batch Markovian arrival process (BMAP). Two simulation models are developed and utilized to randomly generate inter-event times, i.e. the time headway between two successive event occurrences.

PHDs, MAPs and BMAPs do not belong to the distributions or stochastic processes that are commonly used in simulation [10] and are therefore not available as standard components in simulation tools. Yet, it is usually straightforward to integrate them into simulation software by drawing on the underlying Markov chain which governs the activity of the process. The underlying Markov chain can be translated to a simulation model to generate inter-event times. Nonetheless, an alternative approach is proposed to generate random variates for these processes such that the start and end states of an arrival epoch are utilized to set a cumulative distribution for the corresponding inter-arrival time, which is then discretized and sampled to randomly generate inter-arrival times.

Two simulation approaches are applied: (1) the simulation of the underlying Markov chain such that arrivals and their corresponding arrival times are recorded and alternatively (2) the discretization of the cumulative distribution function indicated by

the start and end states of arrival epochs and random sampling of the latter to estimate inter-arrival times. Both simulation approaches are applied on the same set of examples and thusly compared in terms of accuracy (relative error of estimates) and efficiency (running time). Approach (2) is presented as a powerful, traceable and equally accurate simulation technique for arrival processes, yet also as a generic model which can be further deployed and manipulated to flexibly simulate diverse stochastic processes on the non-negative axis. In this context and under Approach (2), variance reduction becomes more attainable rather than utilizing the complete randomness of the transitions in the underlying Markov chains.

The most popular real-life applications of such processes are queuing systems, such that they are heavily utilized in modeling arrival events and/or service completions. Commonly, PHDs have been used to model service times and MAPs are a common means to describe arrival processes in single queues and also in networks of queues. PHDs and MAPs are also used to describe failure and repair times or the duration of availability or unavailability intervals, providing flexible models that can be mapped onto Markov processes and analyzed numerically. In this context, building meaningful and realistic models necessarily requires precise representation of the time to failure, the required repair time of components and systems, arrival patterns, service mechanisms, etc. Further real-life applications of the BMAP are fewer, yet recent research in the telecommunications industry, has focused on studying batch arrivals; i.e. jobs which arrive at a server system simultaneously.

PHDs and MAPs have been introduced by Neuts; however, the matrix representation of MAPs, which is used here and in most other papers on the subject, is due to Lucantoni's later work. Unlike PHDs, the theoretical foundation of MAPs is less advanced, which is not surprising since the future behavior of MAPs may depend on the whole history and not only on the time since the last event occurred as it is the case for PHDs, rendering such a process more complex to manipulate in modeling. Several extensions of MAPs exist to account for different arrival types or batches of arrival which are denoted as MMAPs or BMAPs respectively. Both process types are useful in practice but are even more complex than MAPs. Another generalization is the Rational Arrival Process (RAP) which results from a linear algebraic view without probabilistic interpretation similar to Matrix Exponential distributions. These processes are rarely used, since the theory is not completely developed although some newer results show interesting relations between MAPs and RAPs.

In this research, we will first simulate the processes under study by mimicking the internal transition activity in the underlying Markov chain, such that arrivals are registered by their occurrence times. The simulation of the underlying Markov chain is based on the Stochastic Simulation Algorithm (SSA), yet tailored to adequately capture the specific features of the Markov chains lending their activity to each of the aforementioned arrival processes. We will refer to this approach as Approach (1): Simulation of Underlying Markov Chain.

We further attempt at providing an alternative algorithm to simulate these processes. We try to articulate and model an approximate inversion method, which is

well documented and heavily utilized for all distributions that have a definite and invertible cumulative distribution functions. Yet, this is not the case for these arrival processes, as the cumulative distribution function is either impossible to invert or indefinite, and hence we claim that this second approach is approximate and utilizes a discretization approach to approximate the distributions under study. We will refer to the alternative simulation algorithm by Approach (2): Approximate Inversion Method. Approach (2) first dictates specifying the cumulative distribution function of the inter-arrival time and assessing whether it is invertible or otherwise. We then further utilize this function in creating an inter-arrival time/cumulative probability database which can be later used to randomly generate inter-arrival times randomly. The setup of this database is computationally extensive, but the core execution program is easy and straightforward.

CHAPTER II

GENERAL OVERVIEW OF PHDS, MAPS AND BMAPS

The analysis of man-made systems such as computer systems, communication networks, manufacturing processes, logistics networks, to mention only a few, is commonly done through the utilization of discrete-event models [9] or simulation [10]. One detrimental issue in these models is the adequate modeling of the load which describes the occurrence of events, be it the arrival of customers in queuing networks, failure times in reliability models or packet sizes in simulation of computer models [1]. In simple terms, arrivals, service completions, and/or failure times can be represented by their corresponding inter-event times, which are in turn represented by random variables or stochastic processes generating non-negative numbers [1].

A key and prerequisite stage of developing simulation models is input modeling. Under input modeling, a stochastic model is constructed to capture the key features of an input process from which statistical and real-time measurements are available. Historically, more often than not, it was assumed that inter-event times are independent and identically distributed and accordingly a best-fitting distribution is selected from a given set of distributions to stochastically model the input data in simulation models. However, sometimes the usual set of available distributions is not flexible enough to capture measured behavior and/or the assumption of independence and identicality of events doesn't hold because the events are rather correlated as it is the case for many real-life systems. Therefore, the choice of an appropriate discrete-

events model is sensitive to the type of the latter and its relationship with the mechanics of the man-made system.

In simulation, modeling systems is heavily dependent on the input modeling stage [12, 10] such that the right parameters are determined and fed into the simulation model, guaranteeing accurate analysis and results. In input modeling, software tools are available to perform parameter fitting; i.e. the selection of an adequate distribution to describe a set of observed events [31, 13]. In cases where the assumption of independence and identicality is weak or unjustified, the literature advises the use of "empirical distributions, time series models or multivariate normal or Johnson distributions" [12, 10]. Nonetheless, most simulation software tools do not readily or directly support the aforementioned approaches, and hence such stochastic models have to be manually constructed and integrated into simulation software which tends to be a cumbersome and error-prone exercise.

In this context, Markov processes could be utilized to model such systems, namely the phase-type (PH) process/distribution, the Markovian arrival process (MAP) and the batch Markovian arrival process (BMAP). PHDs are known to be very flexible and allow the approximation of a wide variety of distributions on the positive axis [18]. Nevertheless, the use of PHDs was in the past mainly restricted to a few of its subclasses, mainly Erlang or hyper-exponential distributions [20]. Given the aforementioned, the approximation of a general distribution by a PHD is a complex non-linear optimization problem, which has only recently been developed into

comprehensive computational algorithms, of which only a few are available in today's popular modeling software [1].

The modeling power of Markov processes lies in the fact that they can largely reflect the correlation between inter-event times, especially MAPs and their generalized counterpart BMAPs [21]. For instance, the analysis of single-server queues with MAP input or service completions is developed and based on well-known matrix analytic techniques primarily led by Neuts [104, 22]. Yet, parameter fitting for MAPs is more complex than that of PHDs [23], and hence is not richly discussed in the literature. Only recently have several algorithms for generating MAPs from measured data become available [24, 25]; however, these algorithms are not well-established and documented in the literature. Similar to PHDs, MAPs can be integrated in simulation models but again this approach is not really supported by available simulation tools because the first approaches describing the integration of MAPs in simulation models have been published only recently [11, 27, 26]. PHDs and MAPs are highly flexible and can capture a variety of stochastic behavior, yet this flexibility comes at a high price. This price is effectively the huge effort associated with finding the best fit parameters whereby the resulting model approximates the observed or required behavior closely enough [1]. This fitting complexity has somehow contributed to the scarcity of the literature in the applications of PHDs and MAPs in the past. Yet, this has greatly improved recently as there is a relatively large number of papers, mainly in queuing theory, discussing ways of solving models including PHDs or MAPs and many theoretical papers describing features of PHDs and MAPs are available [1].

Fitting is an important stage in developing numerical and simulative models as they present tools to represent input data. Usually, observations are recorded and measured in a real system and collected to form a trace, in an attempt to estimate (fit) the parameters of a distribution to accurately capture characteristics of the observed data [1]. The matrix representation of the phase type process is highly redundant in general, which makes fitting of PHDs a difficult optimization problem. Many of the existing fitting approaches for PHDs are tailored to specific subclasses for which canonical representations exist, such as the Erlang and hyper-exponential distributions [1]. Generally, fitting algorithms for phase-type distributions can be divided in two classes, depending on the information from the trace they use. The first class of techniques, usually categorized as Expectation Maximization (EM) algorithms, uses the complete trace for parameter estimation, while the second class only uses some derived measures like moments [1].

The complexity inherent in the parameter fitting of MAPs is majorly attributed to the fact that they lack the required canonical representations as in the cases of some phase-type distributions and the obligation to account for longer traces of input data to adequately capture correlation amongst the latter [1]. Although most algorithms for PHDs can be generalized for MAPs, the effort of finding MAP parameters is quite higher and the algorithms tend to be less reliable and stable when applied to MAPs.

The inter-arrival time in PHDs, MAPs and BMAPs is a matrix-exponential random variable. The generation of matrix-exponential random variables hasn’t been comprehensively studied in the literature. Yet, one can simply generate the latter by

utilizing the transition mechanism in the associated Markov chain as we propose in the first simulation approach. Banks investigated three particular simulation algorithms for continuous-time Markov chains to model the progression of Vancomycin-resistant enterococcus (VRE) infection in a hospital unit and the dynamics of HIV during the early stage of infection, in which the target cells are still at very high level while the infected cells are at very low level [2]. Banks utilizes the stochastic simulation algorithm (SSA), as well as explicit and implicit tau-leaping algorithms. The three algorithms were compared in light of the analysis of two stochastically modeled infectious diseases (VRE and HIV) in terms of computational time and degree of precision.

The stochastic simulation algorithm (SSA) is common and largely used and documented in the literature and has been used to simulate the underlying Markov chain of the processes under study via Approach (1). The algorithm is rather easily modeled as it utilizes direct step-wise jumping among states, yet it tends to drag along in time and accuracy when the Markov chain has large number of states, where the tau-leaping algorithms become more efficient [2]. Yet, the purpose of our research is not to investigate the efficiency of this simulation approach; we are merely using it as a reference to which the second approximate approach is compared.

As the name implies, the tau-leaping method proposes conducting leaps from one-time subinterval to another to overcome the extensiveness and impracticality of keeping track of every transition in a Markov chain [2]. So, leaping algorithms jump from one sub-interval to another by approximating how many transitions take place

during a given sub-interval. The value or size of the leap must be chosen such that there is no significant change in the value of the transition rates along the subinterval. So the objective is to simulate many transitions in one leap, speeding by that the computational time and the efficiency. There are two types of tau leaping methods considered in this paper: the explicit and implicit tau leaping methods. Banks' major conclusion was that all three algorithms have comparable computational efficiency for the VRE model due to the low number of species and small number of transitions, yet tau leaping methods are preferred for HIV model due to the larger number of species and higher transition probabilities. However, in the case of PHDs, MAPs and BMAPs, traceability is important such that the nature of transitions among states is more important than the latter's count in determining inter-event times.

In SSA, given the one-step transition probability matrix P, a recursive procedure is presented to simulate the activity/transitions in a Markov chain for maximum number of steps or the maximum sojourn time [3]. First, the steady-state/long-term probabilities are computed and randomly sampled using the inverse transform method to choose the initial state of the chain. Given the current state of the chain, the successive state is randomly predicted by sampling the probability row vector corresponding to the current state using again the inverse transform method. The nature of the simulation stopping criterion is sensitive to nature of the Markov chain. If the chain is a discrete-time Markov chain, then we only care about the number of transitions performed whenever the chain is activated; otherwise, if it is a continuous-time Markov chain (CTMC), then the time spent in the chain is more significant. Accordingly, we either counter the number of transitions occurring in a chain, or accumulate the time

spent in the chain by adding up the sojourn times in the visited states. Yet, the number-of-transitions stopping criterion also works well for a CTMC, because eventually it has a discrete-time embedded Markov chain that is responsible for the instantaneous transitions occurring in the chain.

In a CTMC, whenever the chain hits a state, it spends a certain amount of time, commonly referred to as the holding time or the sojourn time. This quantity is a positive random number and the holding times of states are independent of each other. The transitions among states; however, follows the same mechanism as in a discrete-time Markov chain. However, this can become extensive for higher order Markovian point processes with small transition probabilities.

The Batch Markovian arrival process (BMAP) is a stochastic point process that generalizes the MAP by allowing for correlated arrival batches as opposed to single-unit arrivals, as well as dependent and correlated inter-arrival times [4]. The origins of the BMAP can be traced back to the development of the versatile Markovian point process (VMPP) by Neuts. Neuts' primary objective was to extend the standard Poisson process to account for more complex customer arrival processes in queuing models. The VMPP is characterized by three distinct classes of batch arrivals, each of which is determined by the transition type of an external Markov chain with n transient states and one absorbing state. One type of arrival is from a Markov-modulated Poisson process, which occurs during the sojourn of the external Markov process in any of the n transient states. Another type occurs when the Markov process transitions from transient state i to transient state j, such that $i \neq j$. The third type of arrival occurs when the Markov process

transitions from any transient state to the single absorbing state, and then restarts in state j. This type of transition is called an (i,j) renewal transition, and by virtue of restarting the Markov chain, admits the possibility of a "self-transition" from a transient state to itself. In this context, it is clear that Neuts founded the VMPP on the notion of the phase-type distribution. Neuts played a major role in the advancement of the use of the PHD in queuing theory, culminating ultimately in the development of the VMPP. Lucantoni further extended the original definition of the VMPP to define the Markovian arrival process (MAP) [4]. In queuing systems literature, Ramaswami was the first to incorporate the BMAP as an arrival process to a single-server queue with generally-distributed service times [4].

CHAPTER III

CONTINUOUS-TIME MARKOV CHAINS

It is important to highlight some key aspects of continuous-time Markov chains (CTMCs) before indulging in PHDs, MAPs and BMAPs because the dynamics of these processes can be described and interpreted using the transitional behavior in their underlying continuous-time Markov chains. Continuous-time Markov chains (CTMCs) are a class of stochastic processes characterized by a discrete state space in which the time between transitions is exponential.

Let *S* denote a finite and countable set of states, and $\{X(t)\}_{t\geq 0}^{\infty}$ a stochastic process with state space *S*. *S* is isomorphic to $\mathbb{N}$ (the set of integer numbers), and states are denoted by their numbers.

The stochastic process $\{X(t)\}_{t\geq 0}^{\infty}$ is a continuous-time Markov chain if it can be described by the Markovian property given by (1):

$$P\big(X(t_{k+1}) = x_{k+1} | X(t_k = x_k), \dots, X(t_0 = x_0)\big) \tag{1}$$

$$P\big(X(t_{k+1}) = x_{k+1} | X(t_k = x_k)\big)$$

$$for\ any\ 0 \leq t_0 \leq t_1 \leq \cdots \leq t_k \leq t_{k+1}\ and\ x_l \in S$$

Property (1) indicates that the future state of a stochastic process, specifically a Markov chain, depends only on its current state and not the whole past activity of the process. A process is homogeneous if for all $u \geq 0$, the identity given by (2) is true.

$$P(X(t+u) = j | X(u) = i) = P(X(t) = j | X(0) = i) = P_{ij}(t) \quad (2)$$

Homogeneity in this context implies that the transition probabilities in a Markov chain only depend on the length of the time interval and not on the actual times of occurrence of one state or another. The values of $P_{ij}(t)$ at any time *t* and for all states *i* and *j* define a matrix of transition probabilities $\boldsymbol{P}(\boldsymbol{t})$ such that $\sum_j P_{ij}(t) = 1$.

In CTMCs, the inter-event times, or the time separating the event occurrences are exponentially distributed, yet not necessarily identical. Each state *i* is characterized by an exponential sojourn time denoted by λ_i. Suppose that V_i is the exponential holding time in state *i*, then its distribution is defined in (3):

$$P(V_i \leq t) = 1 - e^{-\lambda_i t}, t \geq 0 \;\&\; \lambda_i > 0 \quad (3)$$

The Markov process evolves as follows: at any time *t*, whereby $X(t) = i$, the process remains in state *i* for an exponential amount of time with a mean $1/\lambda_i$, and then jumps to state *j* with probability $P_{ij} = \lambda_{ij}/\lambda_i$, such that λ_{ij} is the transition rate from state *i* to state *j* and λ_i is the total departure rate from state *i*. Accordingly, the transition dynamics in a CTMC can be described in terms of its infinitesimal generator matrix $\boldsymbol{Q}$. If the state space of the chain has *n* states, then $\boldsymbol{Q}$ will be a $n \times n$ matrix as defined in the expression given by (4).

$$Q(i,j) = \begin{cases} -\lambda_i \; if \; i = j \\ \lambda_{ij} \; if \; i \neq j \end{cases}, where \; \lambda_i = \sum_{j=1}^{n} \lambda_{ij} \;\; for \; \forall \; i, i = 1, \dots, n \tag{4}$$

It is worth nothing that $\{X_l\}_{l \in \mathbb{N}}, X_0 = X(0)$ is the embedded discrete-time Markov chain (DTMC) of the process $\{X(t)\}_{t \geq 0}^{\infty}$. The embedded DTMC is governed by the transition probability matrix $\boldsymbol{P}$ given by (5).

$$P(i,j) = \begin{cases} \dfrac{Q(i,j)}{-Q(i,j)} \; if \; i \neq j \\ 0 \qquad\quad if \; i = j \end{cases}, where \sum_{j} P(i,j) = 1 \; for \; \forall \; i \, , i = 1, \dots, n \tag{5}$$

$$P(i,j) = \begin{cases} 0, if \; i \neq j \\ 1, if \; i = j \end{cases} \; if \; i \; is \; absorbing \; .$$

CHAPTER IV

PHASE TYPE DISTRIBUTION

In this chapter, a brief overview of the definition, notations and subclasses of the phase-type distribution are presented and explained. The phase-type distribution is the distribution of the lifetime of a single-absorbing state Markov chain $\{X(t)\}_{t\geq 0}^{\infty}$; i.e. the time to enter the absorbing state from the set of transient states.

A. Single-Absorbing State Markov Chains and Phase-Type Distributions

Two states *i* and *j* communicate with each other if either is reachable from the other. Suppose that *C* is a subset of the state space *S*, and then if all states in *C* communicate, it is called a communicating set. If there isn't a feasible transition from any state in *C* to any external state, then *C* forms a closed set, in which the Markov process terminates, and hence the concept of absorbing states to which transitions are feasible, yet from which transitions are impossible arises. The counterparts of absorbing states are transient states, to and from which transitions are possible.

If every state in a chain is either transient or absorbing, then the Markov chain is called an absorbing Markov chain. A particular example of absorbing Markov chains is the single-absorbing state Markov chain. These chains lend their behavioral activity to the underlying Markov chains of phase-type distributions.

Suppose we are given a finite single-absorbing state Markov chain with state space *S* defined as the union of the set of transient states S_T and the set of the single absorbing state S_A, $S = S_T \cup S_A = \{1,2,\dots,n\} \cup \{n+1\}$. The infinitesimal generator matrix $\boldsymbol{Q}$ can be formulated as: $\boldsymbol{Q} = \begin{bmatrix} \boldsymbol{D_0} & \boldsymbol{d_1} \\ \boldsymbol{0} & 0 \end{bmatrix}$

Matrix $\boldsymbol{D_0}$ is a $n \times n$ non-singular and invertible matrix describing the transitions among the transient states, such that:

$$D_0(i,j) = \begin{cases} -\lambda_i \; if \; i = j \\ \lambda_{ij} \; if \; i \neq j \end{cases}, for \; \forall \; i \text{ and } j, i, j = 1,2,\dots,n$$

In the long-term, the probability of absorption is 1 and the matrix $(-\boldsymbol{D_0})^{-1}$ is the fundamental matrix of the absorbing CTMC, such that $(-\boldsymbol{D_0})^{-1}(i,j)$ is the expected time spent in state *j* before absorption given that the chain started in state *i*.

Vector $\boldsymbol{d_1}$ is a $n \times 1$ matrix describing the transitions from the transient states to the single absorbing state, such that $\boldsymbol{D_0 1} + \boldsymbol{d_1} = \boldsymbol{0}$, where $\boldsymbol{1}$ and $\boldsymbol{0}$ are n-vectors of ones and zeros respectively

$\boldsymbol{0}$ is an $1 \times n$ zero vector describing the impossibility of transitions from the absorbing state to the transient states and *0* is the rate at which the chain exits the single-absorbing state, implying the termination of the process at state $(n+1)$.

Given the preceding description of single-absorbing state Markov chains, Neuts derived the concept of the phase-type distribution (PHD). In this sense, the phase-type distribution is the distribution of the lifetime of the single-absorbing state Markov chain $\{X(t)\}_{t\geq 0}^{\infty}$; i.e. the time to enter the absorbing state from the set of transient states S_T. For a phase-type process, the transient states are called phases and the order of the process is defined as the number of transient states.

The vector $\boldsymbol{\alpha} = [\alpha_1, \alpha_2, \dots, \alpha_n,]$ describes the state probabilities as to where the chain is to arbitrarily start. Whenever the single absorbing state is hit, the chain restarts itself according to $\boldsymbol{\alpha}$. In most cases, it is assumed that the probability of starting at the absorbing state is zero; however, in some cases it is permissible. If a random variable X has a phase-type distribution, then it has the representation $X \sim (\boldsymbol{\alpha}, \boldsymbol{D_0})$ such that $\boldsymbol{d_1}$ is redundant and can be implicitly derived from $\boldsymbol{D_0}$.

The phase-type process can be described as follows: The chain arbitrarily starts in any state according to $\boldsymbol{\alpha}$; if the starting state is transient, the chain progresses until absorption and the lifetime of the chain is the accumulation of the exponential sojourn times spent in all of the transient states visited prior to absorption [6].

Given a Markov process $\{X(t)\}_{t\geq 0}^{\infty}$ with an infinitesimal generator $\boldsymbol{Q}$, then the probability that the phase of the chain is j at time t, given that the chain initially started at phase i is given by matrix $\boldsymbol{P(t)} = e^{\boldsymbol{Q}t}$, such that $e^{\boldsymbol{Q}t} = \begin{bmatrix} e^{\boldsymbol{D_0}t} & \mathbf{1} - e^{\boldsymbol{D_0}t} \\ \mathbf{0} & 1 \end{bmatrix}$. Therefore, the distribution of the time until absorption is given by:

$$F(t) = P(X(t) = n+1) = \sum_{i=1}^{n+1} P(X(0) = i)\, P(X(t) = n+1 | X(0) = i)$$

$$F(t) = \sum_{i=1}^{n+1} \alpha_i \, \boldsymbol{P}_{i(n+1)}(t)$$

Therefore, the cumulative density function of *X* is as given by (6):

$$F(t) = 1 - \boldsymbol{\alpha} e^{\boldsymbol{D_0} t} \mathbf{1} \tag{6}$$

Knowing that $(-D_0)^{-1}(i,j)$ is the expected time spent in state *j* before absorption given that the chain started in state *i*, then the k^{th} moment of the PHD is given by (7):

$$E[X^k] = k!\, \boldsymbol{\alpha}(-\boldsymbol{D_0})^{-k} \mathbf{1} \tag{7}$$

The mean (8), variance (9) and skewness (10) can be calculated using the first, second and third moments respectively.

$$E[X] = \boldsymbol{\alpha}(-\boldsymbol{D_0})^{-1} \mathbf{1} \tag{8}$$

$$Var(X) = E[X^2] - (E[X])^2 = 2\boldsymbol{\alpha}(-\boldsymbol{D_0})^{-2} \mathbf{1} - (E[X])^2 \tag{9}$$

$$Skew(X) = \frac{E[X^3]}{\sqrt[1.5]{Var(X)}} = \frac{6\boldsymbol{\alpha}(-\boldsymbol{D_0})^{-3} \mathbf{1}}{\sqrt[1.5]{Var(X)}} \tag{10}$$

B. Subclasses of the Phase-Type Distribution

The phase-type process is one example of renewal point processes, whereby only renewal events/arrival epochs of size equal to one are allowed to take place once the single absorbing state is hit. However, point processes are very diverse and heavily deployed in the literature to analytically derive tractable results and/or accurately model challenging qualitative and quantitative features of the arrival processes, service completion epochs, equipment failures and many more [1]. Such processes are usually dense and can be to a varying degree of accuracy used to model different processes on $[0, \infty)$.

In this research paper, we only consider PHDs in continuous time. However, it is can be easily implied that discrete time PHDs are based on DTMCs. Several results can be transferred from the continuous time to the discrete time area, but there are also some specific aspects that need to be considered.

PHDs can be considered as a generalization of the exponential, hyper-exponential, hypo-exponential and Erlang distributions, to mention only a few.

1. Exponential Distribution

The exponential distribution is the simplest case of a PHD, such that it is characterized by one parameter only which is the arrival rate λ. As a PHD, $\boldsymbol{D_0} = -\boldsymbol{\lambda}$ and $\boldsymbol{\alpha} = 1$, such that the underlying Markov chain is dual (Figure 1), having one transient state and one absorbing state. The chain can only start at the transient state and

can only move to the absorbing state at a rate of λ after spending an exponential amount of time with a mean of $(1/\lambda)$ in the transient state.

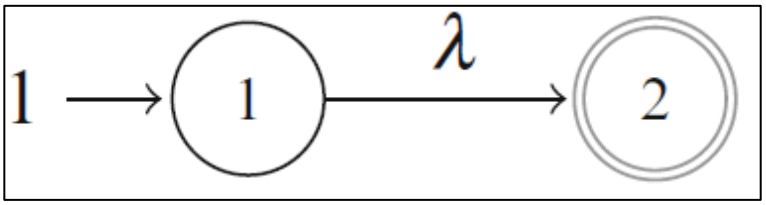


Figure 1 Markov Chain Representation of the Exponential Distribution

2. Erlang Distribution

The Erlang distribution is the distribution of the sum of *n* exponential phases with the same intensity λ. In this context, the underlying Markov chain can be visualized as a set of successive transient states and one final absorbing state (Figure 2). The chain starts at state 1, moves on to state 2 at a rate λ, then to state 3 at the same rate until it eventually reaches the absorbing state $(n+1)$ after spending an exponential amount of time with a mean of $(1/\lambda)$ in each of the preceding transient states.

$$\boldsymbol{D}_0 = \begin{bmatrix} -\lambda & \lambda & \dots & 0 & 0 \\ 0 & -\lambda & \cdots & 0 & 0 \\ \cdots & \cdots & \ddots & \cdots & \cdots \\ 0 & 0 & \cdots & -\lambda & \lambda \\ 0 & 0 & \cdots & 0 & -\lambda \end{bmatrix} \& \ \boldsymbol{\alpha} = [1 \quad 0 \quad 0 \quad \cdots \quad 0]$$

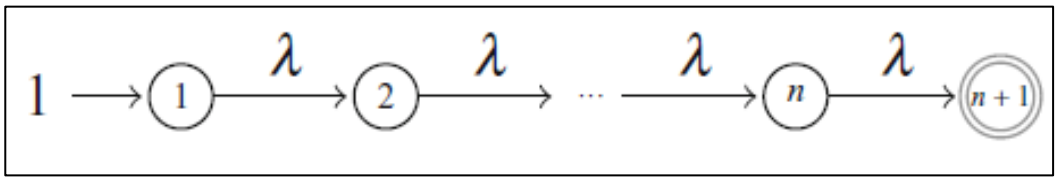


Figure 2 Markov Chain Representation of the Erlang Distribution

3. Hypo-Exponential Distribution

The hypo-exponential distribution is a generalization of the Erlang distribution in the sense that it is the sum of *n* distinct exponential random variables, each with a rate $\lambda_i \; for \; i = 1,2, \dots, n$. Accordingly, the underlying Markov chain is composed of successive states ordered *1* through *n+1*, such that the first *n* states are transient and the last state is absorbing (Figure 3). The chain starts at state *1* where it lingers for an exponential period with parameters λ_1, then moves to state 2, state 3, … state *i*, and eventually moves to the last and single absorbing state where the chain is terminated and restarted again at state 1.

$$\boldsymbol{D}_0 = \begin{bmatrix} -\lambda_1 & \lambda_1 & \dots & 0 & 0 \\ 0 & -\lambda_2 & \cdots & 0 & 0 \\ \cdots & \cdots & \ddots & \cdots & \cdots \\ 0 & 0 & \cdots & -\lambda_{n-1} & \lambda_{n-1} \\ 0 & 0 & \cdots & 0 & -\lambda_n \end{bmatrix} \; \& \; \boldsymbol{\alpha} = [1 \quad 0 \quad 0 \quad \cdots \quad 0]$$

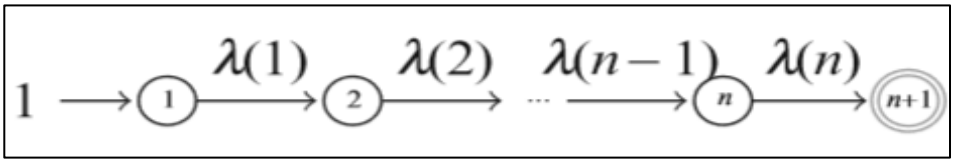


Figure 3 Markov Chain Representation of the Hypo-Exponential Distribution

4. Hyper-Exponential Distribution

The hyper-exponential distribution is a convex mixture of *n* exponential distributions. Suppose *S* is the overall state space of the Markov process $\{X(t)\}_{t\geq 0}^{\infty}$ which lends its activity to the hyper-exponential distribution. Then *S* is made up of n transient states and one absorbing state. The chain can start in any of the transient states, where it spends an exponential amount of time and then moves to the absorbing state at which the chain terminates and starts over from any of the transient states (Figure 4).

$$\boldsymbol{D}_0 = \begin{bmatrix} -\lambda_1 & 0 & \dots & 0 & 0 \\ 0 & -\lambda_2 & \cdots & 0 & 0 \\ \cdots & \cdots & \ddots & \cdots & \cdots \\ 0 & 0 & \cdots & -\lambda_{n-1} & 0 \\ 0 & 0 & \cdots & 0 & -\lambda_n \end{bmatrix}$$

$$\boldsymbol{\alpha} = [\alpha_1 \quad \alpha_2 \quad \dots \quad \alpha_{n-1} \quad \alpha_n]$$

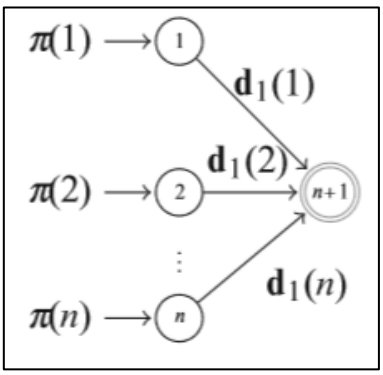


Figure 4 Markov Chain Representation of the Hyper-Exponential Distribution

5. Hyper-Erlang distribution

Another example is the hyper-Erlang distribution (HErD) which is a mixture of *m* mutually independent Erlang distributions (Figure 5).

$$\boldsymbol{D}_0 = \begin{bmatrix} -\lambda_1 & \lambda_1 & \dots & 0 & 0 & \cdots & 0 & \cdots & 0 \\ 0 & -\lambda_1 & \lambda_1 & 0 & 0 & \cdots & 0 & \cdots & 0 \\ \cdots & \cdots & \ddots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ 0 & 0 & \cdots & -\lambda_1 & 0 & \cdots & 0 & \cdots & 0 \\ 0 & 0 & \cdots & 0 & -\lambda_m & \lambda_m & 0 & \cdots & 0 \\ 0 & 0 & 0 & 0 & 0 & -\lambda_m & \lambda_m & \cdots & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & \ddots & \cdots & \cdots \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & 0 & -\lambda_m & \lambda_m \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -\lambda_m \end{bmatrix}$$

$$\boldsymbol{\alpha} = [\alpha_1 \quad 0 \quad \cdots \quad 0 \quad \alpha_2 \quad 0 \quad \cdots \quad 0 \quad \cdots \quad \alpha_m \quad 0 \quad \cdots \quad 0]$$

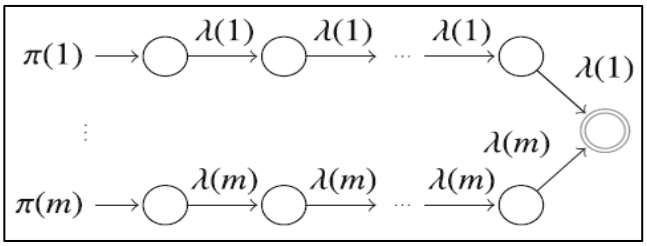


Figure 5 Markov Chain Representation of the Hyper-Erlang Distribution

CHAPTER V

MARKOVIAN ARRIVAL PROCESS

As explained in Chapter IV, after a PH renewal occurs, the underlying Markov chain is immediately restarted based on the initial probability vector $\boldsymbol{\alpha}$ which predicts the starting state of the chain succeeding an event occurrence. However, there are many real-life applications, especially in the telecommunications industry, in which strong correlation exists between subsequent inter-event intervals [7], and so it is only natural that the concept of dependence of subsequent intervals is introduced. Accordingly, the phase distribution becomes dependent on the last phase visited and upon which an event was triggered.

In other words, each state in the underlying Markov chain can behave as an absorbing state and cause an arrival whenever hit by an observed transition. This can be visualized as a CTMC in which two changing variables are registered, the phase and the level of the system. A hidden transition changes the phase of the system as the arrival epoch jumps from one state to another. On the other hand, an observed transition changes not only the phase of the system, but also the level of the system as it causes the occurrence of an arrival epoch. This Markov chain describes the Markovian Arrival Process, shortly referred to as a MAP. This process is heavily used to describe a variety of arrival processes in today's queuing models.

In simple terms, a MAP can be interpreted as an irreducible Markov chain in which transitions could be marked, and marked transitions describe events or arrivals. Formally, a MAP $(\boldsymbol{D}_0, \boldsymbol{D_1})$ is an irreducible Markov chain with a finite state space *S* and an infinitesimal generator matrix $\boldsymbol{Q}$ which can be expressed as: $\boldsymbol{Q} = \begin{bmatrix} \boldsymbol{D_0} & \boldsymbol{D_1} \\ \boldsymbol{0} & \boldsymbol{0} \end{bmatrix}$

Let *n* be the size of state space *S* or the order of the MAP. The process is interpreted as follows: The process randomly starts with a probability α_i in state *i*, resides there an exponentially distributed time with a rate $\lambda_i = \sum_{j \neq i} \boldsymbol{D}_0(i,j) + \boldsymbol{D_1}(i,j)$, and moves to state *j* with a hidden transition with a probability $\boldsymbol{D}_0(i,j)/\lambda_i$ or an observed transition (associated with an arrival event) with a probability $\boldsymbol{D_1}(i,j)/\lambda_i$.

An *n-state* MAP, denoted as MAP_n, can be interpreted as a two-dimensional Markov process $\{J(t), N(t)\}$ defined on the state space $\{(i,j): 1 \leq i,j \leq n\}$. Each state can be visualized as a phase and a level at the same time. If this state is hit by a hidden transition (governed by generator matrix $\boldsymbol{D_0}$), then it causes a change in the phase of the chain $J(t)$ only, if it is; otherwise, hit by an observed/marked transition (governed by generator matrix $\boldsymbol{D_1}$), it causes a change in both the phase of the chain $J(t)$ and the level of the chain $N(t)$.

The aforementioned interpretation implies that $\boldsymbol{D} = \boldsymbol{D_0} + \boldsymbol{D_1}$ is the infinitesimal generator matrix of the underlying Markov process $J(t)$. Knowing that $\boldsymbol{D_0}$ is non-singular and the transition times are finite with probability 1 and the process does

not terminate. The role of the states in this model is to provide the inter-event times distributed as a random sum of non-identical exponential random variables.

Let X_n be the state of the underlying Markov process $J(t)$ at the time of the n^{th} event occurrence, and T_n the time between the events $(n-1)^{th}$ and n^{th}, then $\{X_n, T_n\}_{n=1}^{\infty}$ is a Markov renewal process. In particular $\{X_n\}_{n=1}^{\infty}$ is a Markov chain whose transition probability matrix $\boldsymbol{\beta}$ is given by (11):

$$\boldsymbol{\beta} = (-\boldsymbol{D_0})^{-1}\boldsymbol{D_1} \tag{11}$$

The stationary probability vector $\boldsymbol{\phi}$ of the Markov chain $\{X_n\}_{n=1}^{\infty}$ is obtained by solving (12) and (13):

$$\boldsymbol{\phi\beta} = \boldsymbol{\phi} \tag{12}$$

$$\boldsymbol{\phi 1} = 1 \tag{13}$$

In our research, we will deploy $\boldsymbol{\phi}$ to initiate the Markov chain under Approaches (1) and (2). Under Approach (1), the Markov chain is triggered using $\boldsymbol{\phi}$, and then progressed according to $\boldsymbol{D_0}$ and $\boldsymbol{D_1}$ up until the maximum number of arrival epochs is achieved. Under Approach (2), $\boldsymbol{\phi}$ is used to start up the process, then it is restarted according to the absorption status of preceding event according to $\boldsymbol{\beta}$ such that the correlation and dependence among the inter-event times is not neglected.

The structure of $\boldsymbol{D_0}$ and $\boldsymbol{D_1}$ distinguishes different MAP subclasses. For example, if $\boldsymbol{D_1}$ is a diagonal matrix, then the process is denoted as a Markov Modulated Poisson process (MMPP) because matrix $\boldsymbol{D_1}$ describes up to *n* Poisson processes for an MMPP with *n* states that are selected by a background Markov process defined by $\boldsymbol{D_0}$. A specific case of an MMPP is an Interrupted Poisson Process (IPP) where diagonal elements of the diagonal matrix $\boldsymbol{D_1}$ are either *0* or λ, the rate of the basic Poisson process. On and off times of the Poisson process are given by a PHD.

Several extensions of MAPs exist; foremost the extension to generate different arrival types or batches of arrival which are denoted as MMAPs or BMAPs respectively. Both process types are useful in practice but are even more complex than MAPs. Another generalization is Rational Arrival Processes (RAPs) which result from a linear algebraic view without probabilistic interpretation similar to Matrix Exponential distributions. These processes are rarely used yet, since the theory is not completely developed although some newer results show interesting relations between MAPs and RAPs.

CHAPTER VI

BATCH MARKOVIAN ARRIVAL PROCESS

In today's computer and telecommunication networks for example, it is very common for multiple jobs to be spent to the server simultaneously, and hence jobs arrive in batches [7]. Therefore, MAPs which describe single arrivals can be further generalized to account for arrival batches of different sizes in the Batch Markovian Arrival Process, commonly referred to as the BMAP.

MAPs generate a single type of events, and so this can be extended by allowing *K* different event types resulting in a Marked MAP (MMAP) defined as $(\boldsymbol{D}_0, \boldsymbol{D_1}, \dots, \boldsymbol{D_K})$, where $(\boldsymbol{D}_0, \sum_{k=1}^{K} \boldsymbol{D_k})$ represents a MAP and all matrices $\boldsymbol{D_k}$ are non-negative. If the different events are interpreted as batches of arrivals; i.e. matrix $\boldsymbol{D_k}$ is associated with those arrivals of batch size *k*, then the MAP is extended into what is commonly referred to as batch Markovian Arrival process (BMAP). The aforementioned analysis can be further tailored to describe BMAPs and MMAPs.

Let $J \equiv \{J(t): t \geq 0\}$ be an irreducible, continuous-time Markov chain with state space $S = \{1,2,..,n\}$, where n is a finite and positive integer. Suppose J has just entered state i, such that $i \in S$. The process spends an exponentially distributed amount of time in state i with rate λ_i. The process then transitions to state j, and the transition could be hidden or observed with batch size k. A hidden transition occurs with a

probability $P_0(i,j)$ where $i \neq j$. An observed transition of batch size k occurs with a probability $P_k(i,j)$ for $k \geq 1$ where i could be equal to j. Hence the identity (14) is valid.

$$\sum_{k=1}^{\infty}\sum_{j=1}^{n} \boldsymbol{P_k(i,j)} + \sum_{j=1,i\neq j}^{n} \boldsymbol{P_0(i,j)\ for\ \forall\ i,i \in S} \tag{14}$$

Accordingly, $\boldsymbol{P}_0$ is the transition probability matrix governing the hidden transitions in the external Markov chain, while $\boldsymbol{P}_k$ governs the observed transition of batch size *k*. The transition rates of the aforementioned jumps can be formulated as follows:

$$D_0(i,j) = \begin{cases} -\lambda_i & for\ i = j \\ \lambda_i P_0(i,j) & for\ i \neq j \end{cases} \ \&\ \boldsymbol{D}_k(i,j) = \lambda_i \boldsymbol{P}_k(i,j)\ for\ \forall\ i,j\ \in S, k \geq 1$$

Therefore, $\boldsymbol{D}_0$ contains the transition rates of *J* for which no arrivals occur, while $\boldsymbol{D}_k$ for all $k \geq 1$ contains the rates of the observed transitions of batch size *k*.

It is worth noting that the total transition matrix $\boldsymbol{D}$ ($\boldsymbol{D} = \boldsymbol{D_0} + \sum_{k=1}^{K} \boldsymbol{D_k}$) is given by the following:

$$D(i,j) = \begin{cases} -\lambda_i\ if\ i = j \\ \alpha_{ij} + \sum_{k=1}^{K} \sigma_{ij}^{k}\ if\ i \neq j \end{cases} such\ that\ \alpha_{ij} + \sum_{k=1}^{K} \sigma_{ij}^{k} = \lambda_i$$

Inter-arrival times are defined as the accumulation of the sojourn times in all of the states visited prior to the occurrence of the arrival epochs. Knowing that the phase distribution is sensitive to the state at which an arrival epoch occurs, the very initial state of the underlying Markov chain can be predicted using the stationary or steady-state probabilities defined by $\boldsymbol{\phi}$, such that an arbitrary arrival epoch begins in any state *i* with a probability ϕ_i [6].

The stationary probability row vector can be computed by solving equations (15) and (13).

$$\boldsymbol{\phi}(-\boldsymbol{D_0})^{-1}\sum_{k=1}^{K}\boldsymbol{D_k} = \boldsymbol{\phi} \tag{15}$$

The m^{th} marginal moment of the inter-arrival time *X* can be calculated using (16) for both MAPs and BMAPs:

$$E[X^m] = m!\,\phi(-D_0)^{-m}1, m = 1, 2 \ldots \tag{16}$$

The Lag-1 autocorrelation can be calculated as:

$$\rho_1 = \frac{\boldsymbol{\phi}(-\boldsymbol{D_0})^{-1}\boldsymbol{\beta}(-\boldsymbol{D_0})^{-1}\boldsymbol{e} - E[X^2]}{E[X^2] - (E[X])^2} \tag{17}$$

CHAPTER VII

COMPUTATION OF MATRIX EXPONENTIAL

Mathematical models of many physical, biological and economic processes involve systems of linear, constant coefficient ordinary differential equations (18).

$$\dot{x}(t) = \boldsymbol{A}x(t) \tag{18}$$

Here $\boldsymbol{A}$ is a given, fixed, real or complex $n \times n$ matrix. A solution vector $x(t)$ is sought after to satisfy initial condition such that $x(0) = x_0$. In theory, $\boldsymbol{A}$ is referred to as the state companion matrix and $x(t)$ is the system response. Therefore, the solution is given by (19), where $e^{t\boldsymbol{A}}$ can be formally defined by the convergent power series given by (20).

$$x(t) = e^{t\boldsymbol{A}}x_0 \tag{19}$$

$$e^{t\boldsymbol{A}} = I + t\boldsymbol{A} + \frac{(t\boldsymbol{A})^2}{2!} + \cdots + \frac{(t\boldsymbol{A})^n}{n!} + \cdots = \sum_{n=0}^{\infty} \frac{(t\boldsymbol{A})^n}{n!} \tag{20}$$

Dozens of methods for computing $e^{t\boldsymbol{A}}$ can be obtained from more or less classical results in analysis, approximation theory, and matrix theory. Some of the methods have been proposed as specific algorithms, while others are based on less constructive characterizations. Several particular classes of matrices lead to special

algorithms. For examples, if $\boldsymbol{A}$ is symmetric, then methods based on eigenvalue decompositions are particularly effective. However, for other classes of matrices, eigenvalues might become confluent leading to the loss of accuracy. On the other hand, algorithms which avoid using eigenvalues tend to require considerably more computer time for any particular problem. They may also be adversely affected by round off errors especially in problems where the matrix $\boldsymbol{tA}$ has large elements. One potential shortcoming with almost all algorithms illustrates a sensitive property of $e^{\boldsymbol{tA}}$: As t increases, the elements of $e^{\boldsymbol{tA}}$ may grow before they decay, which might eventually cause significant approximation errors.

The methods used to compute the matrix exponential can be classified into the following categories: series methods, matrix decomposition methods, ordinary differential equation methods, polynomial methods and splitting methods. However, we will focus on series methods and matrix decomposition methods.

A. Series Methods

The common theme of what we call series methods is the direct application to matrices of standard approximation techniques for the scalar function e^t. In these methods, neither the order of the matrix nor its eigenvalues play a direct role in the actual computations.

1. Method 1 Taylor Series

The matrix exponential of tA, defined as e^{tA} can be formally defined by the convergent power series given by (21). Ignoring efficiency, terms of the series can be simply summed until adding another term does not alter the result such that the following is true:

$$If \frac{(t\boldsymbol{A})^m}{m!} \cong 0 \; for \; all \; m \geq k, \; then \; e^{t\boldsymbol{A}} = \sum_{n=0}^{\infty} \frac{(t\boldsymbol{A})^n}{n!} \cong \sum_{n=0}^{k} \frac{(t\boldsymbol{A})^n}{n!} \tag{21}$$

However, concern over where to truncate the series is important if efficiency is being considered, especially when the identity given by (21) becomes cumbersome to compute and lags such that *k* becomes huge. Among several papers concerning the truncation error of Taylor series, the paper by Liou [32] is frequently cited, whereby he suggests some prescribed error tolerance φ (22) to control the choice of *k*.

$$\left\| \sum_{n=0}^{k} \frac{(t\boldsymbol{A})^n}{n!} - e^{t\boldsymbol{A}} \right\| \leq \left(\frac{\|t\boldsymbol{A}\|^{k+1}}{(k+1)!} \right) \left(\frac{1}{1 - \frac{\|t\boldsymbol{A}\|}{(k+2)}} \right) \leq \varphi \tag{22}$$

In other related papers, Everling [33] has sharpened the truncation error bound suggested by Liou in (22). On the other hand, Bickhart [34] has considered relative instead of absolute error. Unfortunately, all these approached ignore the effects of round off errors and so might fail in the actual computation with certain matrices. However, if one wants to compute $e^{t\boldsymbol{A}}$ for several values of *t*, the exercise of determining *k* for every *t* becomes very extensive and slow.

2. *Method 2 Padé Approximation*

The (p, q) Padé approximation of $e^{\boldsymbol{tA}}$ is given by (23).

$$R_{pq}(t\boldsymbol{A}) = \left[D_{pq}(t\boldsymbol{A})\right]^{-1} N_{\boldsymbol{pq}}(t\boldsymbol{A}) \tag{23}$$

Knowing that $D_{pq}(t\boldsymbol{A})$ and $N_{\boldsymbol{pq}}(t\boldsymbol{A})$ are given by (24) and (25) respectively.

$$D_{\boldsymbol{pq}}(t\boldsymbol{A}) = \sum_{j=0}^{p} \frac{(p+q-j)!\,p!}{(p+q)!\,j!\,(p-j)!} (t\boldsymbol{A})^j \tag{24}$$

$$N_{\boldsymbol{pq}}(t\boldsymbol{A}) = \sum_{j=0}^{q} \frac{(p+q-j)!\,p!}{(p+q)!\,j!\,(q-j)!} (-t\boldsymbol{A})^j \tag{25}$$

It is worth noting that the non-singularity of $D_{\boldsymbol{pq}}(t\boldsymbol{A})$ is ensured if p and q are large enough and/or if the eigenvalues of $t\boldsymbol{A}$ are negative. Zakian [35] and Wragg and Davies [36] elaborated more on the various representations of these rational approximations as well as the choice of p and q to obtain the prescribed accuracy. However, similar to the Taylor series expansion, round off errors make Padé approximation unreliable in some cases. For example, for large enough values of q, $D_{\boldsymbol{pq}}(t\boldsymbol{A})$ approaches the series for $e^{-t\boldsymbol{A}/2}$ and $N_{\boldsymbol{pq}}(t\boldsymbol{A})$ approaches the series for $e^{t\boldsymbol{A}/2}$ leading to a cancellation error which in turn prevents accurate determination of $R_{\boldsymbol{pq}}(t\boldsymbol{A})$.

3. Method 3 Scaling and Squaring

As mentioned earlier, the round off complexities and the computational costs of the Taylor series expansion and Padé approximants $D_{pq}(t\boldsymbol{A})$ and $N_{pq}(t\boldsymbol{A})$ increase as $t\|\boldsymbol{A}\|$ increases or as the spread of the eigenvalues of $t\boldsymbol{A}$ increases. However, both difficulties can be considerably controlled by utilizing a fundamental property unique to the exponential function given by (26).

$$e^{t\boldsymbol{A}} = \left(e^{t\boldsymbol{A}t/m}\right)^m \tag{26}$$

One key issue is to choose *m* to be a power of *two* such that $e^{t\boldsymbol{A}t/m}$ can be reliably and efficiently computed, and then to form the matrix $\left(e^{t\boldsymbol{A}t/m}\right)^m$ by repeatedly squaring $e^{t\boldsymbol{A}t/m}$. One common criterion used to choose *m* is to make it the smallest power of two such that $\|A\|/m \leq 1$. With this restriction $e^{t\boldsymbol{A}t/m}$ can be satisfactorily computed by either Taylor series expansion according to *Method 1* or Padé approximants according to *Method 2*.

The squaring and scaling method has been suggested by many authors, including but not limited to, Ward [37], Kammler [38], Kallstrom [39], Scraton [40] and Shah [41, 42].

B. Matrix Decomposition Methods

Methods involving factorization or decompositions of $t\boldsymbol{A}$ are most efficient, but necessarily most accurate, for problems involving large matrices and the repeated

evaluation of $e^{\boldsymbol{tA}}$. They are also especially effective and accurate for symmetric and orthogonal matrices.

All matrix decompositions are based on similarity transformations of the form given by (27), such that $e^{\boldsymbol{tA}}$ can be computed using (28).

$$\boldsymbol{A} = \boldsymbol{SBS}^{-1} \tag{27}$$

$$e^{\boldsymbol{tA}} = \boldsymbol{S}e^{\boldsymbol{tB}}\boldsymbol{S}^{-1} \tag{28}$$

The main idea is to find $\boldsymbol{S}$ such that $e^{\boldsymbol{tB}}$ is easy to compute. The difficulty; however, is that $\boldsymbol{S}$ may be close to singular which means that it conditionality might be large.

1. Method 4 Eigenvalue Decomposition

Under this approach, $\boldsymbol{S}$ is set equal to the matrix whose columns are the eigenvectors of $\boldsymbol{A}$, such that: $\boldsymbol{V} = [v_1| \dots |v_n]$ & $\boldsymbol{A}v_j = \lambda_j v_j$

The *n* equations can then be written as: $\boldsymbol{AV} = \boldsymbol{VD}$. Matrix $\boldsymbol{D}$ is interpreted as: $\boldsymbol{D} = diag(\lambda_1, \dots, \lambda_n)$ and hence the exponential of $\boldsymbol{tD}$ can be easily evaluated as $e^{\boldsymbol{tD}} = diag\left(e^{\lambda_1 t}, \dots, e^{\lambda_n t}\right)$. Since $\boldsymbol{V}$ is non-singular, then $e^{\boldsymbol{tA}}$ can be computed using (29).

$$e^{tA} = \boldsymbol{V} e^{t\boldsymbol{D}} \boldsymbol{V}^{-1} \tag{29}$$

The theoretical difficulty with this method occurs when $\boldsymbol{A}$ does not have a complete set of linearly independent eigenvectors and is thus considered defective, such that no invertible $\boldsymbol{V}$ exists and hence the algorithm fails.

2. Method 5 Jordan Canonical Form

In principle, the problem posed by defective Eigen systems can be solved by resorting to the Jordan canonical form (*JCF*).

If (30) is the JCF of matrix $\boldsymbol{A}$, then the exponential of $t\boldsymbol{A}$ can be calculated using (31).

$$\boldsymbol{A} = \boldsymbol{P}[\boldsymbol{J_1} \oplus \ldots \oplus \boldsymbol{J_k}]\boldsymbol{P^{-1}} \tag{30}$$

$$e^{t\boldsymbol{A}} = \boldsymbol{P}[e^{t\boldsymbol{J}_1} \oplus \ldots \oplus e^{t\boldsymbol{J}_k}]\boldsymbol{P}^{-1} \tag{31}$$

The difficulty is that the JCF cannot be computed using floating point arithmetic. A single rounding error may cause some multiple eigenvalue to become distinct or vice versa altering the entire structure of *J* and *P*. For further discussion of the difficulties of computing the JCF, see the papers by Golub and Wilkinson [43] and Kågstrom and Ruhe [44].

Other decomposition methods include the triangular system of eigenvectors, Schur decomposition and the block diagonal. However, the objective of all decomposition methods is to have $\boldsymbol{B}$ as close as possible to diagonal to facilitate the computation of its matrix exponential and make $\boldsymbol{S}$ well-conditioned such that errors are not magnified.

It is worth mentioning that MATLAB, the computation tool offered by MathWorks to perform various matrix operations uses a combinatorial method of scaling and squaring and Padé approximation to compute the matrix exponential with special attention given to avoiding round off error. MATLAB's demo directory contains three files that employ three different algorithms to compute the matrix exponential. One method uses the built-in function (a combination of scaling and squaring and Padé approximation) developed by MathWorks and recommend by the latter for its generality and adaptability to many matrix classes. The second method uses Taylor series expansion and it is advised for matrices with $\|t\boldsymbol{A}\| \leq 1$. Finally, the third method implements eigenvalue decomposition, emphasizing the accuracy and efficiency of this method for symmetric, orthogonal and other normal matrices. The accuracy deteriorates as the condition number of the eigenvector matrix increases, and the method completely fails when a matrix is defective.

Despite the fact that in our proposed approach, the evaluation of $e^{t\boldsymbol{A}}$ at many values of t is a must, we chose to compute the matrix exponential according to the method employed by MATLAB's built-in function, which is a combination of scaling and squaring and Padé approximation. We have also tried eigenvalue decomposition,

and in terms of computation speed, both methods are comparable, yet eigenvalue decomposition tends to be less accurate for some of the subclasses of the phase-type distribution such as the Erlang distribution and its variations, as well as bursty MAPs for Markovian arrival processes.

CHAPTER VIII

SIMULATION OF MARKOVIAN ARRIVAL PROCESSES

As mentioned earlier, the BMAP is a generalization of the MAP, yet observed transitions are associated with arrival epochs of different sizes rather than single arrivals only. It is also a generalization of the PHD in which strictly independent single arrivals are allowed to occur only. The objective of the simulation of the BMAP, whether under Approach (1) or Approach (2), is to produce random inter-arrival times and their corresponding batch sizes.

Let (t_m, b_m) represent respectively the inter-event time and corresponding batch size of arrival epoch *m*, where $b_m = 1,2, \dots, K$. Accordingly the output of the simulation would be the vector $\{(t_m, b_m)\}_{m=1}^{M}$, such that *M* is the maximum number of arrival epochs simulated.

The BMAP parameters required for both approaches are the following:

- Order of the BMAP *n*; i.e. the number of states in the underlying Markov chain
- Maximum batch size *K*
- Matrix $\boldsymbol{D}_0$; an $n \times n$ matrix holding the rates associated with hidden transitions

- Matrix $\boldsymbol{D}_k \; for \; \forall \; k, k = 1,2, \dots, K; n \times n$ matrices that hold the rates associated with the observed transitions of size *k*

A. Approach (1) Simulation of Underlying Markov Chain

To simulate the activity of the underlying Markov chain, it is necessary to set a relevant stopping criterion because the chain never terminates. Since the main objective behind this approach is to record arrivals and their occurrence times, we will set a maximum number of arrival epochs as a stopping criterion. An alternative stopping criterion would be a system clock; i.e. the maximum time to be spent in the Markov chain; however, the maximum number of arrival epochs is more relevant in the case. Furthermore, we will initiate the chain according to the stationary probability vector $\boldsymbol{\phi}$.

The output of the simulation procedure is a vector of size M containing the arrival times of the recorded arrival epochs and their associated batch sizes. The inter-arrival time of each epoch can then be easily computed

The matrices used in the simulation procedure are the following:

- $D_g = \sum_{k=1}^{K} D_k$, n × n matrix holding the total rates of the observed transitions
- Matrix $\boldsymbol{P}$, such that $P(i,j) = \begin{cases} \frac{D_g(i,j)}{-D_0(i,i)} & if \; i = j \\ \frac{D_{0(i,j)} + D_g(i,j)}{-D_0(i,i)} & if \; i \neq j \end{cases}$

- Matrix $\boldsymbol{P_g}$, such that $P_g(i,j) = \frac{D_g(i,j)}{-D_0(i,i)}$

- Matrix $\boldsymbol{P_k}$, such that $P_k(i,j) = \frac{D_k(i,j)}{-D_0(i,i)}\ for \forall\ k, k = 1,2,\dots,K$

Denote by m the current number of arrival epochs and T the time spent in the Markov chain. We start with m and T set to zero.

The simulation procedure comprises of the following steps:

Step 1: Predict the initial state of the chain: $iInitial$

1.1.Generate a uniform random number: $r\ \in\ [0,1]$

1.2.Set $iInitial = i$ such that $r \leq \sum_{stat=1}^{i} \phi_{state}$.

1.3.Update T. $T = T + h_{iInitial}$, where $h_{iInitial} \sim exp(\lambda_{iInitial})$

1.4.Update the current state of the chain. $iCurrent = iInitial.$

Step 2: Simulate the Markov chain while $m < M$

2.1.Predict the subsequent state of the chain: $iNext$

- Generate a uniform random number: $r\ \in\ [0,1]$
- Set $iNext = i$ such that $r \leq \sum_{state=1}^{i} P(iCurrent, state)$

2.2.Predict the type of transition from $iCurrent$ to $iNext$: hidden or observed

- Generate a uniform random number r such that $r\ \in\ [0,1]$
- Calculate the probability that a transition from $iCurrent$ and $iNext$ is observed: $P_{iCurrent,iNext}^{observed} = \frac{P_g(iCurrent,iNext)}{P(iCurrent,iNext)}$

- The transition is observed if $\leq P^{observed}_{iCurrent,iNext}$; otherwise it is hidden and skip to Step 3.
- Update m, $m = m + 1$
- Set t_m, $t_m = T$
- Predict the size of the arrival epoch:
  - Generate a uniform random number r such that $r \in [0,1]$
  - Sample the following probabilities: $\left\{\frac{P_K(iCurrent,iNext)}{P_g(iCurrent,iNext)}\right\}_{k=1}^{K}$
  - Set batch size b_m: $b_m = k\ if\ r \leq \sum_{size=1}^{k} \frac{P_{size}(iCurrent,iNext)}{P_g(iCurrent,iNext)}$

Step 3: Update the current state of the chain. $iCurrent = iNext$

Step 4: Update T. $T = T + h_{iCurrent}$, where $h_{iCurrent} \sim exp(\lambda_{iCurrent})$.

Step 5: Go back to Step (2) and repeat

B. Approach (2) Approximate Inversion Method

Under this approach, we utilize the expanded state space of the underlying Markov chain, such that every state is represented by $K + 1$ virtual states, whereby K is the maximum batch size associated with the arrival process. Effectively, a state can be both transient and absorbing; transient when hit by a hidden transition and absorbing when hit by an observed transition of size $k, k = 1,2, \dots, K$. It is worth noting that

hidden transitions are governed by $\boldsymbol{D_0}$, while observed transitions of size k are governed by $\boldsymbol{D_k}$.

The distribution of the inter-arrival time in the batch Markovian arrival process is not definite; however, it is well-established that the inter-arrival times are dependent and highly correlated. Hence, the joint probability of the inter-arrival time and the batch size of an arrival event is sensitive to the start and end states of the event.

The objective of this approach is to set up the row vectors given by (32) and (33); i.e. the objective is to discretize the phase-type distribution. The vector given by (32) is a vector of increasing time values, such that successive pairs differ by an increment δ (the evaluation of which will be introduced in Chapter IX). Consequently, (33) is the evaluation of the cumulative distribution function corresponding to the time values, the start and end states of the arrival epoch, as well as its size.

$$t = [t_o \; t_1 \; \dots \; t_{max}] \tag{32}$$

$$t_i = t_{i-1} + \delta \;\; where \; t_0 = 0, \delta = E[T]/\wp\,, \qquad i = 1, 2 \dots max$$

For an arrival epoch of size k and which starts at i and ends at j:

$$F_{ij}^{k}(t) = \left[F_{ij}^{k}(t_0) \;\; F_{ij}^{k}(t_1) \dots \; F_{ij}^{k}(t_{max})\right] \tag{33}$$

It is worth noting that t_{max} is set to be as high as possible to ensure that $F_{ij}^{k}(t)$ converges to 1. Yet, the evaluation of (32) is terminated whenever $F_{ij}^{k}(t_n)$ converges to 1 and accordingly (33)is truncated and the maximum time value is reduced such that $t_{max} = t_n$ where $F_{ij}^{k}(t_n) \rightarrow 1.0$.

The matrices used in the simulation procedure are the following:

- $\boldsymbol{D_s}$, an $n \times nK$ matrix such that $\boldsymbol{D_s} = [\boldsymbol{D_1}|\boldsymbol{D_2}| \dots |\boldsymbol{D_K}]$
- $\boldsymbol{\beta}$, an $n \times nK$ matrix such that $\boldsymbol{\beta} = (-\boldsymbol{D_0})^{-1}\boldsymbol{D_S}$ such that $\beta_{i(j-kn)}$ is the probability that an arrival epoch will get absorbed in state $(j - kn)$ with batch size k given starting in state i.
- $\boldsymbol{Q}$, an $n(K+1) \times n(K+1)$ infinitesimal matrix $\boldsymbol{Q} = \begin{bmatrix} D_0 & D_1 & \dots & D_K \\ 0 & 0 & & 0 \end{bmatrix}$

1. Setup Procedure

Given the start and end states, as well as the batch size of an arrival epoch, (33) can be evaluated as follows:

$$e^{tQ} = \begin{bmatrix} - & R^1(t) & \dots & R^K(t) \\ 0 & 0 & & 0 \end{bmatrix} \tag{34}$$

Such that: $P(X_m(t) = j, J_m = k, t_m \leq t | X_m(0) = i) = R_{ij}^{k}(t)$

It follows that (33) can be evaluated as in (35):

$$F_{ij}^{k}(t) = \left[\frac{R_{ij}^{k}(t_0)}{\beta_{i(j+kn)}} \frac{R_{ij}^{k}(t_1)}{\beta_{i(j+kn)}} \dots \frac{R_{ij}^{k}(t_{max})}{\beta_{i(j+kn)}}\right] for\ \forall\ i,j,k \tag{35}$$

We end up with (n × n × K) vectors of (35) corresponding to the times vector (32), to be utilized in the simulation procedure of Approach (2). The setup procedure is conducted once per example and stored to be reused regardless of the targeted number of variates to be generated via simulation.

2. Simulation Procedure

Again, we denote by *m* the current number of arrival epochs. We start with *m* set to zero. The simulation procedure proceeds as follows:

Step 1: Predict the start state of the chain $iStart$; i.e. the start state of the very first arrival epoch

1.1.Generate a uniform random number r such that $r \in [0,1]$

1.2.Set the initial state of the chain: iInitial = i such that $r \leq \sum_{stat=1}^{i} \phi_{state}$

Step 2: While $m < M$, do the following:

2.1.Predict the absorption status of the arrival epoch m; the end state $iEnd$ and the batch size *k* by sampling the vector $\beta(iStart, i)\ for\ all\ i = 1, .. n \times K.$

State i reflects $iEnd$ and k: $iEnd = i - kn$, where $k = floor\left(\frac{i}{n}\right)$

2.2.Generate the inter-arrival time t_m

- Generate a uniform random number r such that $r \in [0,1]$
- Locate r in $F^k_{iStart,iEnd}(t)$ using the bisection method such that:

  $F^k_{iStart,iEnd}(t_i) \leq \mathrm{r} \leq F^k_{iStart,iEnd}(t_{i+1})$

- Linearly interpolate to estimate t_m:

$$t_m = t_i + \frac{(t_{i+1} - t_i)}{\left(F^k_{iStart,iEnd}(t_{i+1}) - F^k_{iStart,iEnd}(t_i)\right)} \left(r - F^k_{iStart,iEnd}(t_i)\right)$$

2.3.Update m such that: $m = m + 1$

2.4.Update iStart, $iStart = iEnd$

2.5.Go back to Step 2 and repeat

It is worth noting that the MAP is a special case of the BMAP, whereby the maximum batch size allowed is 1, and hence the algorithms built for Approaches (1) and (2) can be reduced by the elimination of some redundant parameters to fit the MAP. Accordingly, under Approach (1), we eliminate the steps which involves predicting the size of the arrival epoch. Under Approach (2), similar adjustments are conducted to eliminate accounting for the size of the arrival epoch, which is in the MAP a deterministic quantity such that *K* is equal to 1.

CHAPTER IX

SIMULATION OF PHASE-TYPE DISTRIBUTION

The phase-type process is a special case of the MAP, yet it is distinguished with an underlying Markov chain that has a single absorbing state that defines arrivals once hit, and a constant phase distribution such that whenever an arrival occurs, the Markov chain restarts according to the initial probability vector. The objective of the simulation of the phase-type process, whether under Approach (1) or Approach (2), is to produce random inter-arrival times. It is to be noted that Approach (1) for the phase-type process was discussed in [5] by Neuts.

Suppose *X* is the inter-arrival time associated with the process. Under Approach (1), the transition activity in the underlying Markov chain is modeledsuch that transitions are monitored up until absorption, and arrival times are registered. On the other hand, under Approach (2), we attempt to mimic the classical inversion method by discretizing the cumulative distribution function of the PH inter-arrival as it is not invertible, and then sampling it to generate inter-arrival times. The parameters required for the execution of either simulation approach are the following:

- Order of the distribution n; i.e. the number of transient states in the underlying Markov chain
- $\boldsymbol{\alpha} = [\alpha_1\ \alpha_2\ \ldots\ \alpha_n\ \alpha_{n+1}]$ such that α_i is the probability of starting at state i

- Matrix $\boldsymbol{D_0}$; the $n \times n$ sub-generator matrix holding the rates associated with hidden transitions among the n transient states

Note that $\boldsymbol{d_1}$, the $n \times 1$ column vector holding the rates associated with the observed transitions is redundant and can be derived from $\boldsymbol{D_0}$. Given $\boldsymbol{D_0}$, we set up $\boldsymbol{d_1}$ as $\boldsymbol{d_1}(i) = |\boldsymbol{D_0}(i,i)| - \sum_{\substack{j=1 \\ i \neq j}}^{n} \boldsymbol{D_0}(i,j) \;\; for \; \forall \; i = 1, 2 \ldots n.$

A. Approach (1) Simulation of Underlying Markov Chain

For each arrival epoch, we simulate the corresponding transition activity until hitting the absorbing state, whereby all transient states visited are recorded and the time until absorption or the inter-arrival time can be calculated by aggregating the holding times over all the transient states visited by the arrival epoch before absorption.

First, designate by $\boldsymbol{D}$ an $n \times (n+1)$ matrix such that:$\boldsymbol{D} = [\boldsymbol{D_0} \; \boldsymbol{d_1}]$. Second, denote by S_m an array list to store the transient states visited by the arrival epoch *m* before absorption. This array is dynamic in the sense that it is sensitive to the transition behavior of the arrival epoch. We also designate by h_m the corresponding inter-arrival time and h_m^i the time spent in transient state *i* by arrival *m*.

For every *m*, the following iterative procedure is conducted to simulate the corresponding Markov chain activity until absorption. Note that *M* is the maximum number of arrivals and *m* is initially set to *zero*.

Step 1: Predict the initial state of the arrival m, $iInitial$

1.1. Generate a uniform random number r such that $r \in [0,1]$

1.2. Sample $\boldsymbol{\alpha}$: $iInitial = i$ such that $r \leq \sum_{state=1}^{i} \alpha_{state}$

1.3. Set $iCurrent = iInitial$

Step 2: Simulate the Markov chain while $iCurrent < n + 1$, where $(n + 1)$ is the single absorbing state

2.1. Update S_m, such that $S_m = S_m \cup \{iCurrent\}$

2.2. Predict $iNext$

- Generate a uniform random number r such that $r \in [0,1]$
- Set $iNext = i$, such that $r \leq \sum_{state=1}^{i} \frac{D(iCurrent,state)}{|D(iCurrent,iCurrent)|}$ $state \neq iCurrent$
- Update $iCurrent$, $iCurrent = iNext$

2.3. Go back to Step (2) and repeat.

Step 3: Compute the inter-arrival time of m, $h_m = \sum_{i \in S_m} h_m^i$

We end up with a vector of size M containing the inter-arrival times of all arrivals $(1, 2, \dots, M)$.

B. Approach (2) Approximate Inversion Method

The cumulative distribution function of the inter-arrival time associated with the phase-type process is given in (6). Yet, it is mathematically impossible to invert (6), and hence the approximate nature of Approach (2).

The objective of this approach is to set up the row vectors given by (32) and (36); i.e. the objective is to discretize the phase-type distribution. The vector given by (32) is a vector of increasing time values, such that successive pairs differ by an increment δ. Consequently, (36) is the evaluation of the cumulative distribution function corresponding to the time values in (32).

$$F(t) = [F(t_0)\ \ F(t_1) \ldots\ F(t_{max})] \tag{36}$$

1. Initialization and Setup Procedure

Knowing that $F_X(t)$ represents the cumulative distribution function of $X \sim PH(\alpha, D_0)$, then $F_X(t)$ is by definition continuous and strictly increasing from zero to 1 for all t, such that $t \in [0, \infty)$. If we designate by c and $c + h$ as non-negative PH random numbers on $[0, \infty)$, then the following is true:

$$F_X(c) = 1 - \alpha e^{cD_0} 1 \ \& F_X(c+h) = 1 - \alpha e^{(c+h)D_0} 1$$

$$F_X(c+h) - F_X(c) = C(1 - e^{hD_0} 1), where\ C = \alpha e^{cD_0}$$

$$\lim_{h \to 0} \frac{F_X(c+h) - F_X(c)}{h} = \lim_{h \to 0} \frac{C(1 - e^{hD_0} 1)}{h} = \lim_{h \to 0} \frac{C(1 - he^{hD_0} 1)}{1} = C(1 - hI_n 1)$$

It is worth noting that $C(\mathbf{1} - h\boldsymbol{I_n}\mathbf{1})$ is a single-element matrix; i.e. a scalar quantity, proving that $lim_{h\to 0}\frac{F_X(c+h)-F_X(c)}{h}$ exists for all t, such that $t \in [0,\infty)$; therefore, $F_X(t)$ is differentiable at every t.

Since $F_X(t)$ is differentiable over its domain of definition $t \in [0,\infty)$, then it can be expanded about any point *a*, such that $a \in [0,\infty)$ using Taylor's theorem as given by (37).

$$F_X(t) = F_X(a) + F_X^{(1)}(a)(t-a) + \frac{F_X^{(2)}(\tau)}{2!}(t-a)^2, a \le \tau \le t \tag{37}$$

Using linear approximation, **Error! Reference source not found.** is reduced and the error term is designated by (38).

$$\varepsilon(t,a) = \left|\frac{F_X^{(2)}(\tau)}{2!}\right|(t-a)^2 = \frac{\varepsilon(\tau)}{2}(t-a)^2 \tag{38}$$

Accordingly, the smaller the difference between *t* and *a*, the error term given by (38) converges to *zero*, implying that linear approximation given by (39) is a good estimate of $F_X(t)$.

$$F_X(t) = F_X(a) + F_X^{(1)}(a)(t-a) \tag{39}$$

However, under Approach (2), $F_X(t)$ is known for all time values given in (32). The CDF values are sampled to estimate the corresponding inter-event time. For every arrival epoch *m*, $F_X(t_m) = r, r \sim U(0,1)$ and the corresponding inter-event time t_m is located in an interval $[t_i, t_{i+1}]$, such linear approximation or alternatively linear interpolation can be utilized to estimate t_m.

The main issue is then to determine increment size δ which ensures that the error term doesn't exceed a certain maximum as will be demonstrated hereafter.

The second derivative $F_X^{(2)}(t)$ is computed for all time values in (32), and $max\{\varepsilon(\tau)\}_{\tau\in[t_0,t_{max}]}$ is utilized to determine the size of δ as follows:

$$F_X^{(2)}(t) = \left[F_X^{(2)}(t_0)\ \ F_X^{(2)}(t_1) \ldots\ F_X^{(2)}(t_{max}) \right] \tag{40}$$

Hence, $max\{\varepsilon(\tau)\}_{\tau\in[t_0,t_{max}]} = max\left\{F_X^{(2)}(t)\right\}_{t\in[t_0,t_{max}]}$

$$max\{\varepsilon(t,a)\}_{\tau\in[t_0,t_{max}]} = \frac{max\{\varepsilon(\tau)\}_{\tau\in[t_0,t_{max}]}}{2}\delta^2 \tag{41}$$

We start with an initial estimate of $\wp_{initial}$ (greater than 1), such that $\delta_{initial} = E[X]/\wp_{initial}$.

The following simple check is done to assess the accuracy of $\wp_{initial}$ and modify according to (42).

$$\delta = \begin{cases} \delta_{initial} & if\ 1/\wp_{initial} \geq max\{\varepsilon(t,a)\}_{\tau\in[t_0,t_{max}]} \\ \sqrt{\dfrac{2}{\wp_{initial} \times max\{\varepsilon(t,a)\}_{\tau\in[t_0,t_{max}]}}} & otherwise \end{cases} \quad (42)$$

After determining the adequate spacing of (32) using (42), we reconfigure vectors (32) and (36) accordingly, setting up by that the database for the simulation procedure.

It is worth noting that t_{max} is set to be as high as possible to ensure that $F(t)$ converges to 1. Yet, the evaluation of (36) is terminated whenever $F(t_n)$ converges to 1 and accordingly (32) is truncated and the maximum time value is reduced such that $t_{max} = t_n\ where\ F(t_n) \rightarrow 1.0$.

2. Simulation Procedure

We denote by m the current number of arrivals and start with m set to zero. For every arrival m, the following is the simulation procedure:

Step 1: Generate a uniform random number r such that $r \in [0,1]$

Step 2: Sample *F(t)*, such that $F(t_i) \leq r \leq F(t_{i+1})$

Step 3: Linearly interpolate using the $\left(t_i, F(t_i)\right)$ & $\left(t_{i+1}, F(t_{i+1})\right)$ such that:

$$h_m = t_i + \frac{(t_{i+1} - t_i)}{\left(F(t_{i+1}) - F^k_{iStart,iEnd}(t_i)\right)} \left(r - F(t_i)\right)$$

The output of the simulation procedure is a vector of size M containing the inter-arrival times of the arrival epochs.

It is worth noting that Approach (2) was inspired by the work done by Brown, Place and Liefvoort on the generation of matrix exponential random varieties [8]. Analogous to what has been presented later in the context of the algorithm of Approach (2), the authors suggest: (i) the generation of a uniform random number "r" on the interval [0, 1], (ii) setting a maximum decay value for which the cumulative distribution function can be computed with confidence, (iii) if "r" is less than the decay value, then the bisection method is used to locate "r" in a vector of increasing CDF values and consequently evaluate the corresponding time value utilizing the decay value and an exponential tail.

CHAPTER X

APPLICATIONS, RESULTS AND DISCUSSION

To compare Approaches (1) and (2), several cases/examples of PHDs, MAPs and BMAPs were run and analyzed using both approaches. This section summarizes the findings of these applications.

A. Randomly Populated Examples

To eliminate bias to either simulation approach, we first introduce the efficiency of Approach (2) versus Approach (1) by applying both on the same set of randomly populated examples. Random PHD, MAP and BMAP examples were generated such that the required parameters were randomly populated.

1. Phase-Type Distribution

Random PHD examples were generated such that the two required parameters α and D_0 were randomly populated. This was mainly done to assess the sensitivity of either approach, yet namely Approach (2), to the order of the phase-type distribution under study.

The execution procedure under Approach (1) is effectively the simulation process as no setup is required prior to initiating the underlying Markov chain.

Alternatively, the execution procedure under Approach (2) can be decomposed into two components: the initialization and setup stage and the simulation process. In the initialization and setup stage, a database of time values and their corresponding cumulative probabilities is created independently from and before the simulation process. The setup procedure can be performed and stored once per numerical example regardless of the targeted number of random PH numbers to be generated.

The output is composed of the average inter-event time (estimate of the first moment), the corresponding 95th confidence interval, estimates of the variance and skewness and the duration of simulation procedure for Approaches (1) and (2), as well as the duration of the setup procedure for Approach (2). Randomly populated PHD examples of orders 1 through 10 were run and replicated ten times each for one million arrivals. Refer to Appendix I for the output of the simulation under Approaches (1) and (2).

The simulation approaches were compared in terms of the accuracy of the approximation of either approach of the mean inter-event time, its variance and the skewness of the cumulative distribution function such that the exercise reflects the accuracy of the estimation of the first, second and third moments of the inter-arrival time respectively. The efficiency of Approach (2) is assessed in light of the duration of the simulation process as compared to that of Approach (1).

Tables 1 and 2 summarize the calculation results of the error of the estimates of the mean inter-arrival times, its variance and the skewness of the corresponding CDF

relative to the true values as derived from equation (6) for orders 1 through 5 and 9 through 10 respectively.

Table 1 Error Analysis of Randomly Populated PHD Examples – Orders 1 to 5

	Percent Relative Error of Estimate				
Order	1	2	3	4	5
Approach (1)					
Mean	0.1498%	-0.1477%	0.1143%	0.1406%	-0.1281%
Variance	0.3124%	-0.5639%	0.1257%	0.1924%	-0.3955%
Skewness	-0.1000%	-0.2417%	-0.1041%	0.0000%	-0.3393%
Approach (2)					
Mean	-0.0963%	-0.1477%	0.1143%	0.0000%	0.0641%
Variance	0.9380%	-0.1738%	0.0153%	-0.2270%	0.0466%
Skewness	-0.2000%	-0.1933%	-0.0520%	-0.7890%	-0.3393%

Table 2 Error Analysis of Randomly Populated PHD Examples – Orders 6 to 10

	Percent Relative Error of Estimate				
Order	6	7	8	9	10
Approach (1)					
Mean	0.0000%	0.0916%	0.0000%	-0.1098%	0.0606%
Variance	-0.4201%	0.0829%	-0.0786%	-0.1968%	0.3115%
Skewness	-0.6623%	0.2510%	-0.1515%	-0.4403%	0.3428%
Approach (2)					
Mean	0.0000%	0.0916%	0.0000%	-0.1098%	-0.1819%
Variance	0.3682%	-0.1095%	0.0429%	-0.1509%	-0.0767%
Skewness	0.6113%	0.5522%	-0.3030%	-0.1468%	0.3918%

For the mean inter-arrival time, the estimate resulting from Approach (2); i.e. the approximate inversion method is as accurate as that resulting from Approach (1) via the simulation of the underlying Markov chain. The absolute error of the average inter-arrival time as a percentage of the mean inter-arrival time ranges from 0% to 0.15% under Approach (1) and to 0.18% under Approach (2). It is worth noting that the true

mean of the inter-arrival time always falls within the 95^{th} confidence interval of the estimated average inter-arrival time for both approaches.

In terms of accurately reflecting the variability of the process, simulation of the underlying Markov chain under Approach (1) is slightly more accurate than the approximate inversion method under Approach (2), similarly Approach (1) can match the skewness of the cumulative distribution function of the inter-arrival time better than its counterpart. However, the difference between the errors of the estimates of Approaches (1) and (2) is slight, whereby the maximum absolute error as a percentage of the true value (variance or skewness) is less than 1% for both approaches.

Figures 6 and 7 illustrate the variation of the duration of the execution procedure (in milliseconds) of Approaches (1) and (2) respectively. The execution procedure of Approach (2) is composed of a setup stage and a simulation procedure, while that of Approach (1) is effectively the simulation process. No prior setup is required to initiate the simulation under Approach (1); however unlike Approach (2) in which the database setup per example can be reused innumerably, the simulation under Approach (1) is always renewed regardless of the number of random numbers to be generated.

For Approach (2), a noticeable increasing trend is registered for the setup time, such that the time required to prepare the simulation database or to discretize the distribution grows with the increase in the order of the PHD; i.e. the number of transient states in the underlying Markov chain. This can be attributed to the growing size of the

involved matrix operations. While the setup time of Approach (2) increases with the order of the PHD, the simulation time varies only slightly, such that no particular trend can be realized for the variation of the simulation time as a function of the order. Generally, the duration of the simulation procedure under Approach (2) fluctuates closely about its average and independently from the order of the PHD.

For Approach (1), the duration of the execution procedure; i.e. the simulation process increases with the order of the underlying Markov chain. Generally, a potential arrival initiates the underlying Markov chain and undergoes one, two, or even hundreds of transient transitions before hitting the single absorbing state and exiting the chain, and then the larger the number of transient states, it is more likely that the longer it will take to simulate the underlying Markov chain especially for a larger targeted number of arrivals.

Similarly, for Approach (2), the execution time increases with the order of the underlying Markov chain as the size of the sub-generator matrix grows and the required matrix operations become more extensive. However, given that the simulation time varies only slightly with the order, the increase in the execution time is mostly due to the increase in the setup time, such that the sensitivity to the order of the underlying Markov chain is shifted to the setup stage as opposed to the case of Approach (1).

The difference in the execution times between Approach (1) and Approach (2) tends to increase for higher-order PHDs, whereby it might be more efficient to utilize

the underlying Markov chain to simulate lower-order PHDs (less than 4) and alternatively deploy Approach (2) for higher-order PHDs.

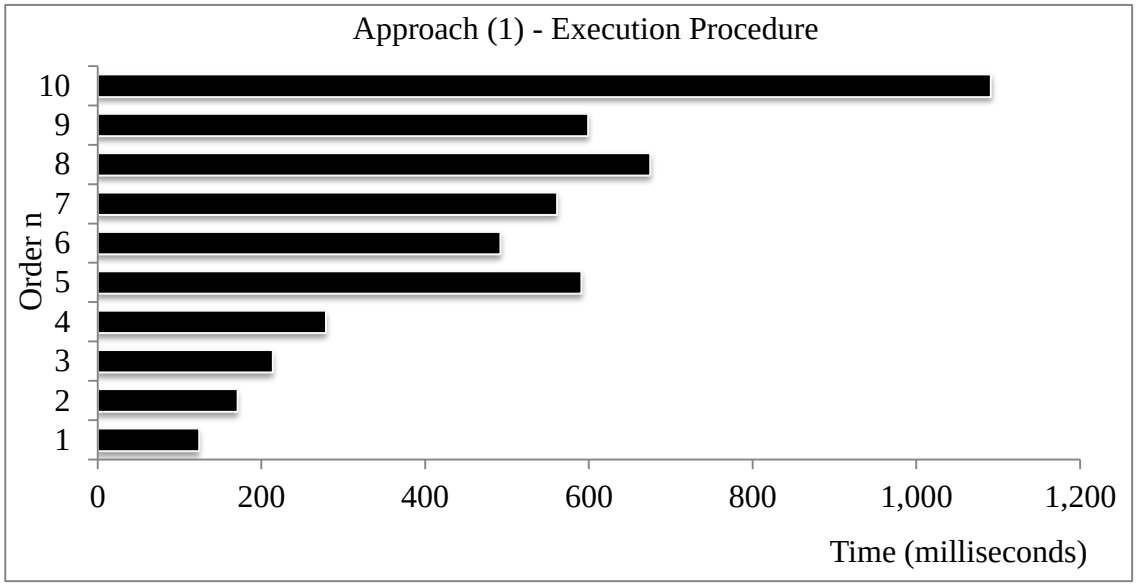


Figure 6 Variation of Execution (Simulation) Time of Approach (1)

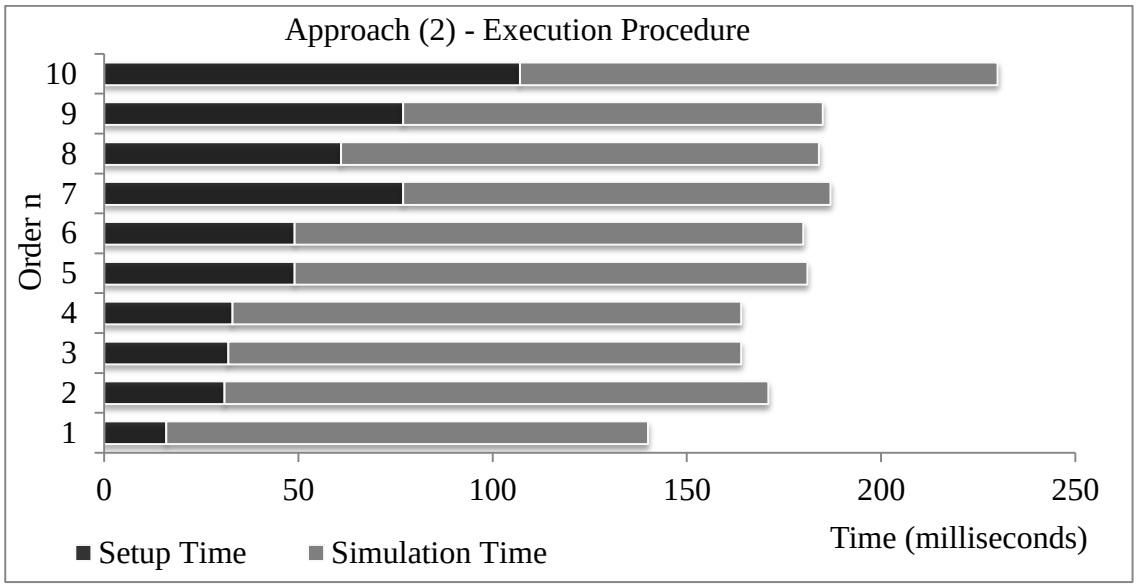


Figure 7 Variation of Execution Time of Approach (2)

Figure 8 shows the variation of the simulation time for both approaches as a function of the order of the PHD. As aforementioned, the duration of the simulation process varies only slightly under Approach (2), as opposed to the increasing simulation time under Approach (1). The simulation process under Approach (2) is fairly independent from the order of the underlying Markov chain, unlike the case for Approach (1).

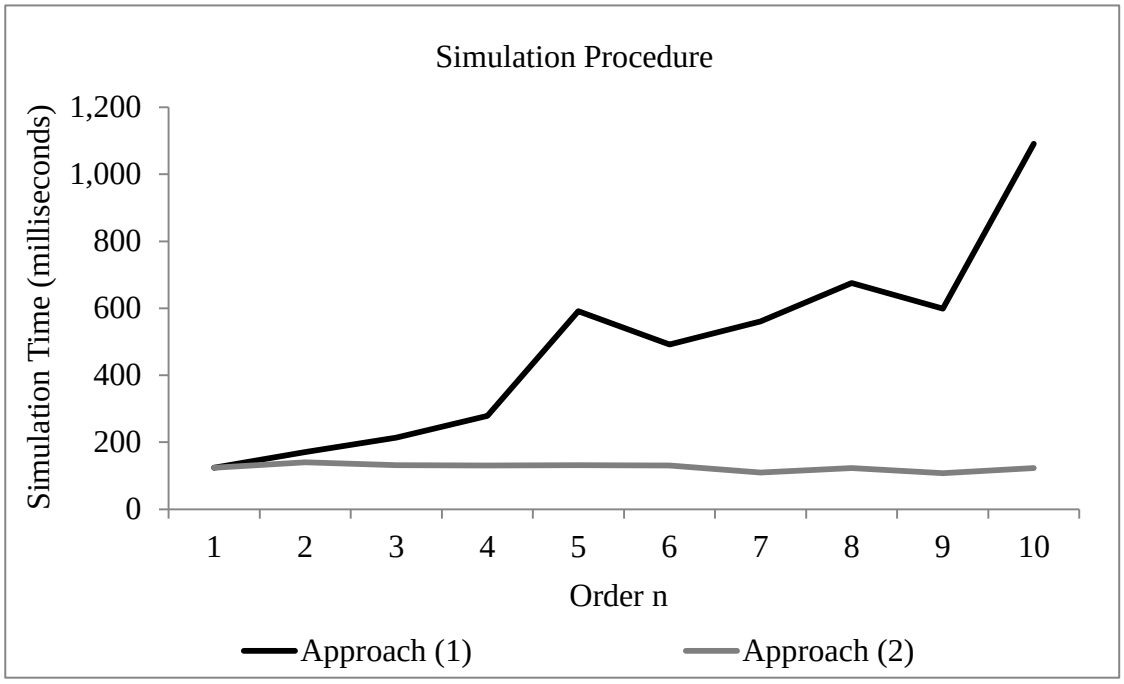


Figure 8 Randomly Populated PHD Examples - Variation of the Simulation Time

2. Markovian Arrival Process

Random MAP examples were utilized to compare Approaches (1) and (2), such that the MAP parameters D_0 and D_1were randomly populated. The output components are similar to that of the randomly populated PHD examples. Refer to Appendix II for more information about the simulation output. Tables 3 and 4 summarize the error analysis of the estimated average inter-arrival time for each order under both approaches in terms of the percent error as compared to the true mean.

Table 3 Error Analysis of Randomly Populated MAP Examples – Orders 2 to 6

Percent Relative Error of Estimate					
Order	2	3	4	5	6
Approach (1)					
Mean	0.0000%	0.0000%	0.0000%	0.0000%	0.0000%
Variance	0.0370%	-0.0708%	0.0000%	0.0000%	0.0000%
Skewness	0.1511%	0.0000%	0.0000%	-0.0499%	0.1474%
Approach (2)					
Mean	0.0000%	0.0000%	0.0000%	0.0000%	0.0000%
Variance	-0.1110%	-0.1416%	0.0000%	0.0000%	0.0000%
Skewness	-0.0504%	-0.0512%	-0.0505%	0.0000%	-0.0491%

Table 4 Error Analysis of Randomly Populated MAP Examples – Orders 7 to 10

Percent Relative Error of Estimate				
Order	7	8	9	10
Approach (1)				
Mean	0.0000%	0.0000%	0.0000%	0.0000%
Variance	0.0000%	0.0000%	0.0000%	0.0000%
Skewness	-0.1001%	0.2011%	-0.0495%	0.1493%
Approach (2)				
Mean	0.0000%	0.0000%	0.0000%	0.0000%
Variance	0.0000%	0.0000%	0.1429%	0.0000%
Skewness	-0.1001%	-0.1006%	0.0445%	0.0498%

Approaches (1) and (2) are equally accurate in estimating the mean inter-arrival times, such that they both reach 100% accuracy for all orders. Although Approach (1) captures the variability of the process and the skewness of the corresponding distribution of the inter-arrival time more accurately than its counterpart, Approach (2) is much more accurate in reflecting the correlation among the generated inter-arrival times; however, the estimation of either approach of the correlation cannot be considered efficient as the absolute error is larger than 5% of the true value. It was noted that for examples with weak correlation, simulation by either approach tends to overestimate the correlation

Figure 9 illustrates the variation of the duration of the setup time (in milliseconds) of Approach (2) per order. As expected, the setup time increases as the order of the underlying process increases.

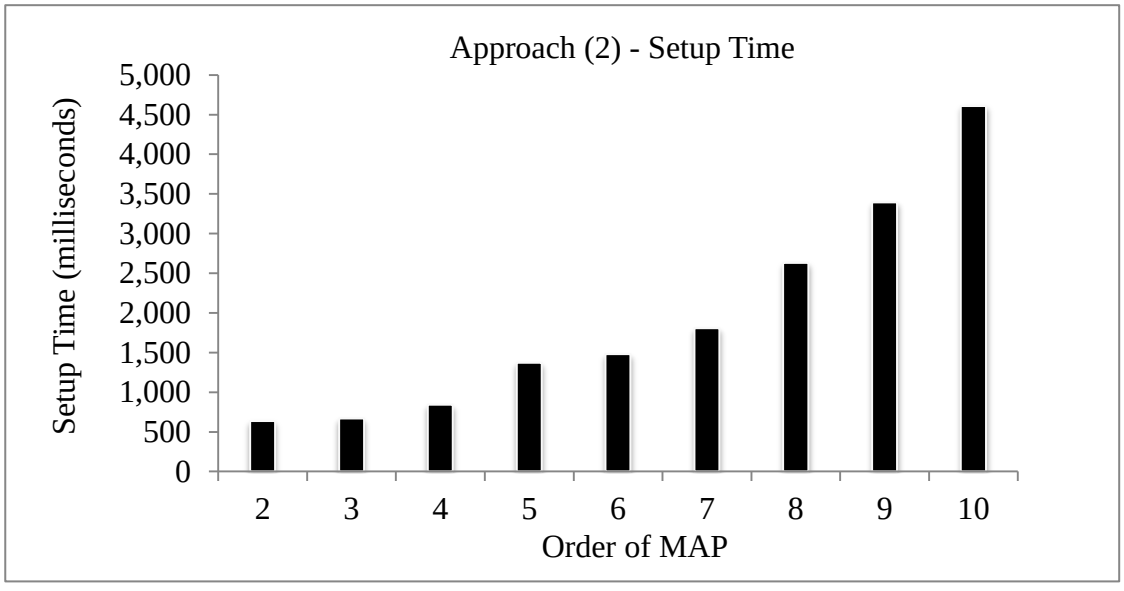


Figure 9 Randomly Populated MAP Examples-Variation of Setup Time of Approach (2)

Figure 10 displays the variation of the duration of the simulation time (in milliseconds) of both approaches per order. The sensitivity of the simulation time of either approach to the order of the underlying process is not clear, such that no specific pattern can be realized from these observations. Generally, Approach (1) performs more efficiently than Approach (2)

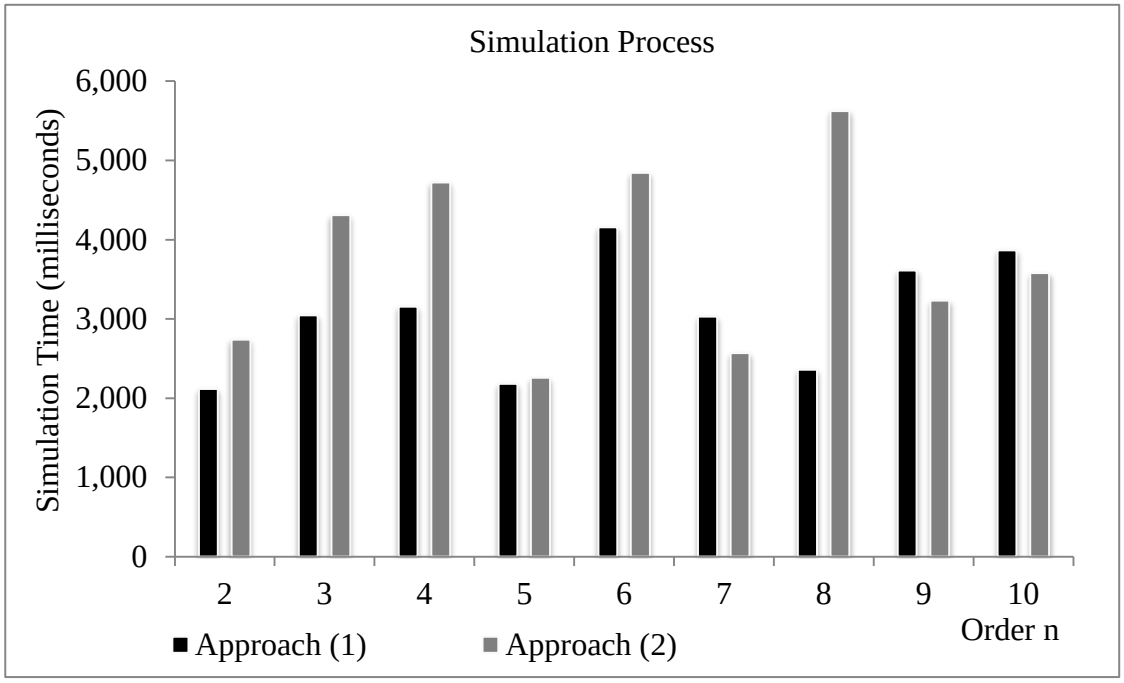


Figure 10 Randomly Populated MAP Examples-Variation of Simulation Times

Simulating a fully populated MAP via utilizing the underlying Markov chain is found to be generally faster because any instantaneous transition between any two states could mark an arrival; the transitional dynamics lend this simulation approach its superior efficiency and accuracy as compared to Approach (2). However, if the probability of arrival; i.e. absorption in one state, given starting in any state were to be

reduced, we presume that Approach (2) would be more appropriate. Instead of fully populating $\boldsymbol{D}_1$, arrival can be restricted to and/or from a specific number of states, which is effectively the case in many real-life applications of MAPs. Within this context, we introduce four numerical examples to highlight the efficiency of Approach (2) relative to Approach (1).

a. MAP (4) and Variants

We consider the example, whose hidden transitions are represented by Figure 11. Four variants of this example are run using Approaches (1) and (2) for ten million arrivals. All four variants share one $\boldsymbol{D}_0$, yet differ by $\boldsymbol{D}_1$. Ten replications were performed per variant and the best estimates are reported.

Figure 11 illustrates the hidden transitions in the four-state Markov chain underlying the MAP process understudy. This transitional activity is expressed by $\boldsymbol{D}_0$.

$$\boldsymbol{D_0} = \begin{bmatrix} -31 & 10 & 6 & 12 \\ 15 & -27 & 7 & 1 \\ 7 & 8 & -23 & 3 \\ 6 & 3 & 9 & -18 \end{bmatrix}$$

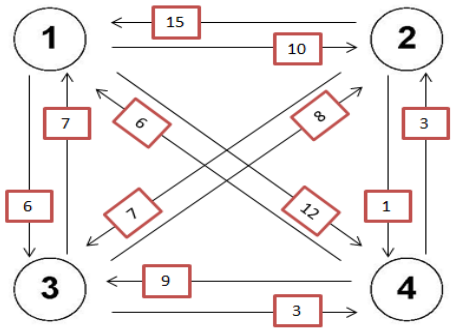


Figure 11 Example One - Hidden Transitions in Underlying Markov Chain

The marked transitions in each of the four variants can be expressed by $\boldsymbol{D_1^1}$, $\boldsymbol{D_1^2}$, $\mathbf{D}_1^3$ and $\mathrm{D}_1^{\mathbf{4}}$ respectively.

$$\mathrm{D}_1^{\mathbf{1}} = \begin{bmatrix} 0 & 3 & 0 & 0 \\ 0 & 0 & 4 & 0 \\ 0 & 0 & 0 & 5 \\ 0 & 0 & 0 & 0 \end{bmatrix} \mathrm{D}_1^2 = \begin{bmatrix} 0 & 0 & 0 & 3 \\ 0 & 0 & 0 & 4 \\ 0 & 0 & 0 & 5 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

$$\mathrm{D}_1^3 = \begin{bmatrix} 0 & 1.5 & 0 & 1.5 \\ 0 & 0 & 2 & 2 \\ 0 & 0 & 2.5 & 2.5 \\ 0 & 0 & 0 & 0 \end{bmatrix} \mathrm{D}_1^4 = \begin{bmatrix} 3 & 0 & 0 & 0 \\ 0 & 4 & 0 & 0 \\ 0 & 0 & 5 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

Table 5 summarizes the performance of both approaches for all four variants in terms of the setup and simulation. Unlike any MAP example whose $\boldsymbol{D}_1$is fully and randomly populated, Approach (2) is much more efficient for MAPs with restrictions on marked transitions, such as reducing their probabilities or rates relative to the total departure rate from the origin state and/or limiting marked transitions to a reduced set of specific states. Not only is the simulation less, but also the overall speed of the execution process under Approach (2) is greater.

Table 5 MAP(4) and Variants : Performance of Approaches (1) and (2)

Variant	1	2	3	4
Approach (1)				
Simulation Time (milliseconds)	9,256	10,304	9,169	8,407
Approach (2)				
Setup Time (milliseconds)	1,767	1,569	1,756	1,818
Simulation Time (milliseconds)	2,109	3,092	2,301	1,817

3. Batch Markovian Arrival Process

Random BMAP examples were generated to compare Approaches (1) and (2), such that the BMAP parameters $\boldsymbol{D}_0$ and $\boldsymbol{D}_k$, for $k = 1,2 \ldots, K$ were randomly populated. The output is composed of the average inter-event time, the corresponding 95th confidence interval, and the duration of simulation procedure for Approaches (1) and (2), as well as the duration of the setup procedure for Approach (2). Refer to Appendix III for more information about the output of the simulation.

BMAPs of orders 2 through 6 were run for one million arrival epochs using both approaches. For each order, batch sizes 2, 3, 4 and 5 were run. Tables 6 through 9 summarize the error analysis of the estimated average inter-arrival time for each batch size under both approaches.

For BMAPs with a maximum batch size of 2 (Table 6), transition matrices $\boldsymbol{D}_0$, $\boldsymbol{D}_1$ and $\boldsymbol{D}_2$ were randomly populated per example. For orders 2 and 3, the absolute error under Approach (2) is higher than that under Approach (1). For orders 5 and 6, both approaches result in equal estimates of the mean inter-arrival times. The accuracy of Approach (2) per one batch size and relative to Approach (1) improves as the order of the underlying process increases.

Table 7 displays the results of the simulation of BMAPs of maximum batch size 3. Generally, for lower-order BMAPs, simulation using Approach (1) is more accurate (orders 2, 3 and 4). However, the accuracy of the output resulting from the simulation under Approach (2) tends to improve as the order of the underlying process

increases to 5 and 6. However, given a specific order, increasing the batch size from 2 to 3 does not have a clear impact on the error.

Table 6 Randomly Populated BMAP Examples - Batch Size 2

Percent Relative Error of Estimate	Order				
Batch Size 2	2	3	4	5	6
Mean	0.4765	0.2699	0.2557	0.1526	0.1392
Approach (1)					
Sample Mean	0.4760	0.2699	0.2552	0.1528	0.1391
Relative Error (%)	0.1049%	0.0000%	0.1955%	-0.1311%	0.0718%
Approach (2)					
Sample Mean	0.4764	0.2700	0.2553	0.1528	0.1391
Relative Error (%)	0.0210%	-0.0371%	0.1564%	-0.1311%	0.0718%

Table 7 Randomly Populated BMAP Examples - Batch Size 3

Percent Relative Error of Estimate	Order				
Batch Size 3	2	3	4	5	6
Mean	0.5471	0.2418	0.1296	0.0987	0.0840
Approach (1)					
Sample Mean	0.5481	0.2416	0.1295	0.0986	0.0840
Relative Error (%)	-0.1828%	0.0827%	0.0772%	0.1013%	0.0000%
Approach (2)					
Sample Mean	0.5473	0.2422	0.1298	0.0986	0.0840
Relative Error (%)	0.0365%	0.1652%	0.1541%	-0.1014%	0.0000%

Tables 8 and 9 display the results of the simulation of BMAPs of maximum batch size 4 and 5 respectively. For batch sizes 4 and 5, Approaches (1) and (2) are highly comparable in terms of accuracy resulting mostly in equal estimates. Additionally, similar to the previous examples, the mean decreases with the increase in the order of the underlying process per batch size, as well as with the increase in the

batch size per order. This can be explained by the presumption that generally the increase in the order of the process and/or the maximum batch size tends to inflate the mean sojourn time in each state, especially if the transition rates are randomly generated with no prior restriction.

Table 8 Randomly Populated BMAP Examples - Batch Size 4

Percent Relative Error of Estimate	Order				
Batch Size 4	2	3	4	5	6
Mean	0.3108	0.1363	0.0987	0.0736	0.0585
Approach (1)					
Sample Mean	0.3106	0.1364	0.0988	0.0736	0.0586
Relative Error (%)	0.0644%	-0.0734%	-0.1013%	0.0000%	-0.1709%
Approach (2)					
Sample Mean	0.3110	0.1362	0.0987	0.0735	0.0584
Relative Error (%)	-0.0644%	0.0734%	0.0000%	0.1359%	0.1709%

Table 9 Randomly Populated BMAP Examples - Batch Size 5

Percent Relative Error of Estimate	Order				
Batch Size 5	2	3	4	5	6
Mean	0.2982	0.1318	0.0779	0.0609	0.0459
Approach (1)					
Sample Mean	0.2983	0.1317	0.0779	0.0609	0.0459
Relative Error (%)	-0.0335%	0.0759%	0.0000%	0.0000%	0.0000%
Approach (2)					
Sample Mean	0.2983	0.1317	0.0780	0.0610	0.0459
Relative Error (%)	-0.0335%	0.0759%	-0.1284%	-0.1642%	0.0000%

Figure 12 illustrates the variation of the duration of the setup time (in milliseconds) of Approach (2) per order and batch size. As expected, the setup time increases as the order and/or the batch size increases.

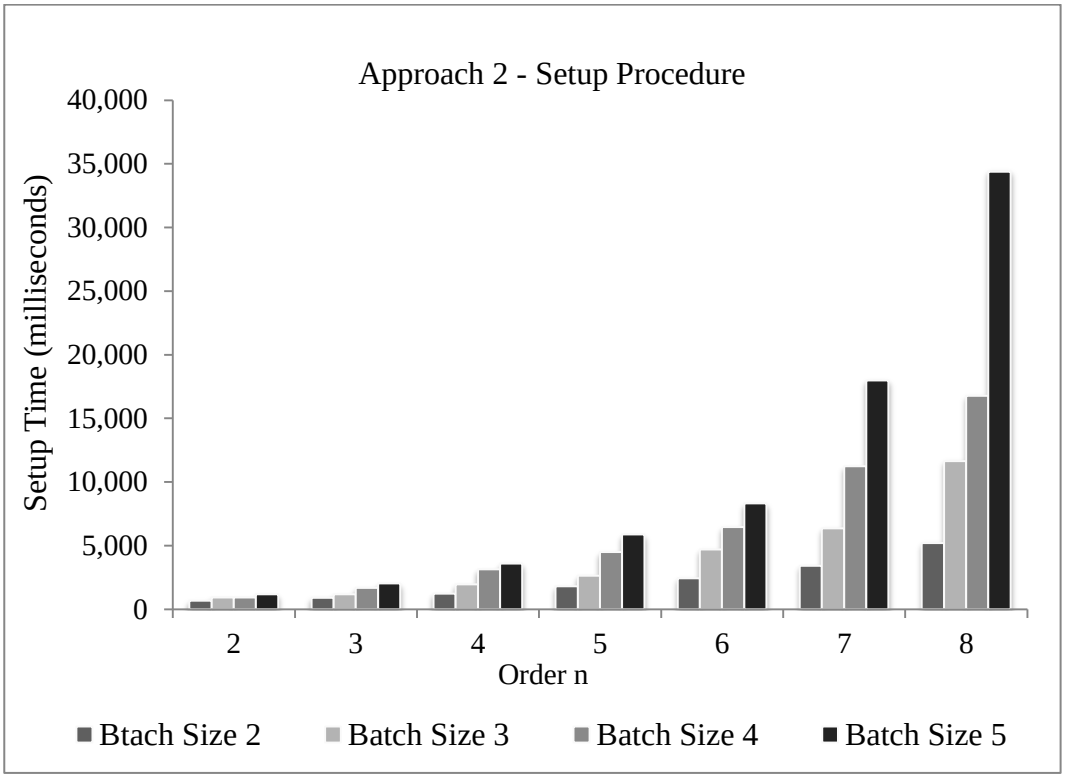


Figure 12 Randomly Populated BMAP Examples - Setup Time of Approach (2)

As for the duration of the simulation time, no specific pattern is observed relative to the variation of the order of the underlying process and/or the maximum batch size associated with marked transitions for both approaches. However, the simulation duration of any example using Approach (1) is faster, and on average, the simulation time under Approach (2) is almost 3 times greater than that under Approach (2) primarily due to the search mechanisms heavily deployed in the algorithm. Yet, similar to MAPs, if the marked transitions were reduced and/or restricted to and from specific states, Approach (2) is most likely to become more efficient than Approach (1).

B. Effect of Variability

A sought after property of any simulation approach is the flexibility to accurately reflect the variability of a process. We chose the balanced two-level mixture of Erlangs, which is a subclass of PHDs also referred to as the hyper-Erlang, to analyze the sensitivity of Approach (2) namely to the change in the variability of the process.

1. Balanced Two-Level Mixture of Erlangs

We consider the mixture of $E(m_1, m_1\lambda_1)$ and $E(m_2, m_2\lambda_2)$. An arrival epoch activates the underlying Markov chain at state 1 with a probability α or state $(m_1 + 1)$ with a probability $(1 - \alpha)$, and then proceeds through the corresponding successive series of transient states until it hits the single absorbing state.

One important characteristic of the balanced two-level mixture of Erlangs is reflected in (43).

$$\lambda_1 = \frac{\lambda_2 \alpha}{1 - \alpha} \tag{43}$$

The mean inter-event time can be expressed by the simplified form given by (44).

$$E[X] = \frac{2(1 - \alpha)}{\lambda_2} \tag{44}$$

Suppose we propose a constant mean equal to μ such that $\mu > 0$. Transition rate λ_2 can then be expressed as a function of μ and α as in (45), and accordingly transition rate λ_1 can be computed using (43).

$$\lambda_2 = \frac{2(1-\alpha)}{\mu} \tag{45}$$

We consider a numerical example depicted in Figure 13.

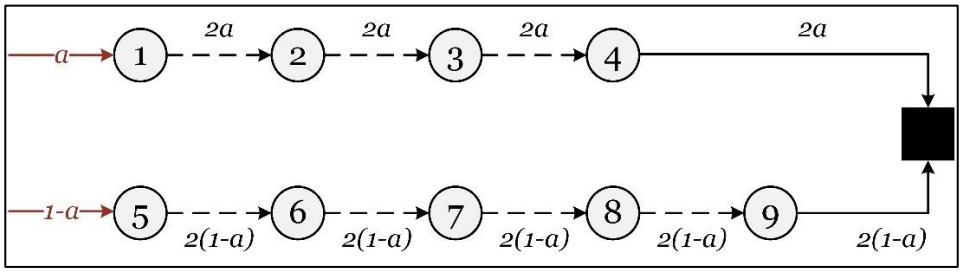


Figure 13 Markov Chain Representation of Balanced Two-Level Mixture of Erlangs

We vary the values of α from 0.1 to 0.9 at a step size of 0.1, and run Approaches (1) and (2) on the corresponding process for ten million arrivals with ten replications per case.

Varying alpha is directly associated with varying the variance of the inter-arrival time as displayed in Figure 14. The coefficient of variation (CV), as the ratio of the standard deviation to the mean, decreases from 1.568 at $\alpha = 0.1$ to 0.474 at $\alpha = 0.5$,

then increases back to 1.568 at $\alpha = 0.9$. The curve depicting the change in the coefficient of variation is almost symmetric about $\alpha = 0.5$, such that generally equidistant values of alpha result in the same variance.

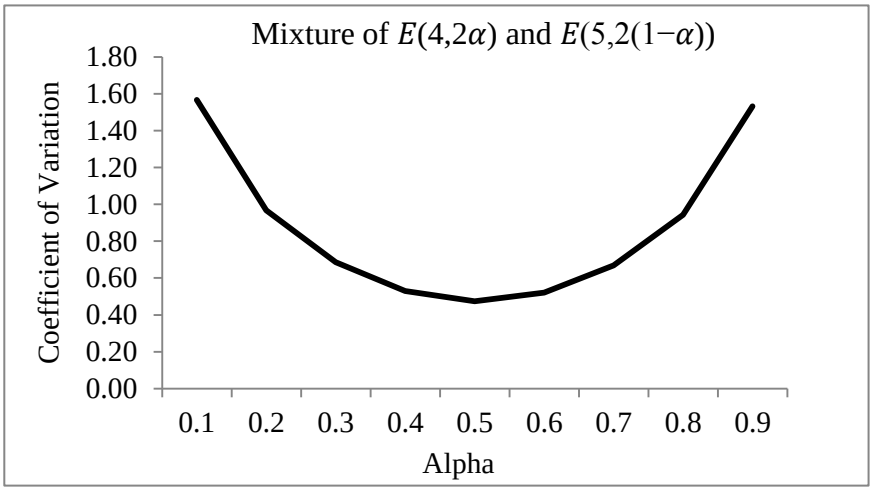


Figure 14 Variation of CV of Balanced Two-Level Mixture of Erlangs with Alpha

Tables 10 and 11 summarizes the error analysis of the estimates of the mean inter-arrival time, its variance and the skewness of the corresponding cumulative distribution function as measures of the accuracy of either approach in estimating the first, second and third moments respectively.

Table 10 Balanced Two-Level Mixture of Erlangs – Alpha [0.1,0.5]

Percent Relative Error of Estimate					
Alpha	0.1	0.2	0.3	0.4	0.5
CV	1.568	0.968	0.686	0.530	0.474
Approach (1)					
Mean	-0.0617%	-0.0200%	-0.0100%	0.0033%	0.0117%
Variance	-0.0136%	-0.1600%	-0.0506%	-0.0889%	0.0000%
Skewness	0.1258%	-0.1911%	-0.1037%	0.0729%	-0.1042%
Approach (2)					
Mean	0.1150%	0.0467%	0.0367%	0.0033%	-0.0100%
Variance	0.2712%	0.1493%	0.1620%	-0.0889%	0.0000%
Skewness	-0.0503%	-0.1147%	0.0000%	-0.1459%	-0.1042%

Table 11 Balanced Two-Level Mixture of Erlangs – Alpha [0.6,0.9]

Percent Relative Error of Estimate				
Alpha	0.6	0.7	0.8	0.9
CV	0.520	0.668	0.944	1.532
Approach (1)				
Mean	-0.0083%	-0.0083%	0.0167%	0.0150%
Variance	0.0615%	-0.0960%	0.0421%	0.0757%
Skewness	-0.0933%	-0.0631%	0.0000%	-0.0814%
Approach (2)				
Mean	0.0117%	-0.0100%	-0.0250%	-0.0633%
Variance	0.0615%	0.1280%	-0.0702%	-0.1373%
Skewness	-0.2799%	0.1263%	0.1297%	0.1086%

For all three quantities, mean inter-arrival time, variance and skewness, Approach (1) generally results in more accurate estimates than Approach (2); however, slightly so. Under Approach (1), the impact of changing the variability of the process on the error of the estimated mean inter-arrival time is not clear. Yet under Approach (2), a relatively clearer pattern can be observed, such that the absolute error decreases as the variance decreases when alpha increases from 0.1 to 0.5, then generally increases back for values of alpha ranging from 0.6 to 0.9. This means that the reduction in the

variability of the process generally resulted in estimates closer to the true value of the mean inter-arrival time under Approach (2), and the opposite is true as well. This is not the case for the variance and skewness, such that under both approaches, no particular trend can be realized for the variation of the absolute error. The impact of variability can be traced by the length of the 95th confidence interval of the estimate of the mean inter-arrival time. The lower the variability of the PHD under study, the smaller is the interval (Figure 15).

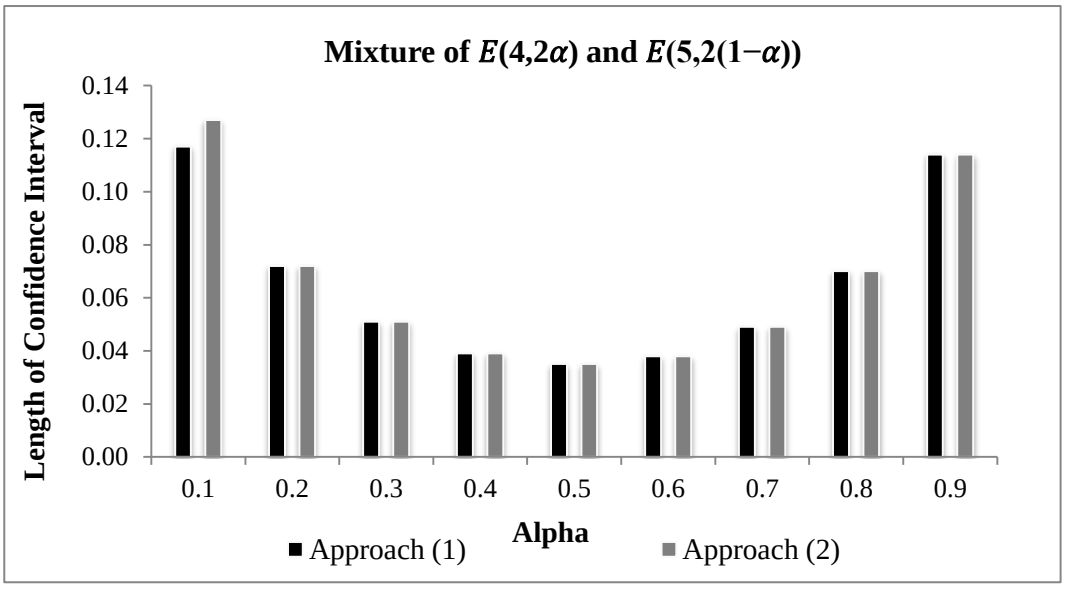


Figure 15 Variation of the Length of 95th Confidence Interval as a Function of Alpha

Another important aspect is the impact of variability on the duration of the execution of the simulation approach. As aforementioned, the execution of Approach (2) can be decomposed into the setup procedure and the simulation process, as opposed to the execution of Approach (1) which is effectively the simulation process itself.

Figure 16 shows the variation of the setup time as a function of alpha under Approach (2). The striking feature of the curve depicting the trend of the variation of the setup time of Approach (2) is its similarity to that depicting the change in CV (coefficient of variation) as a function of alpha. The setup time decreases with the decrease in the variability of the process under study, and vice versa, which can be attributed to the increase in the complexity of the matrix operations associated with processes with higher variability.

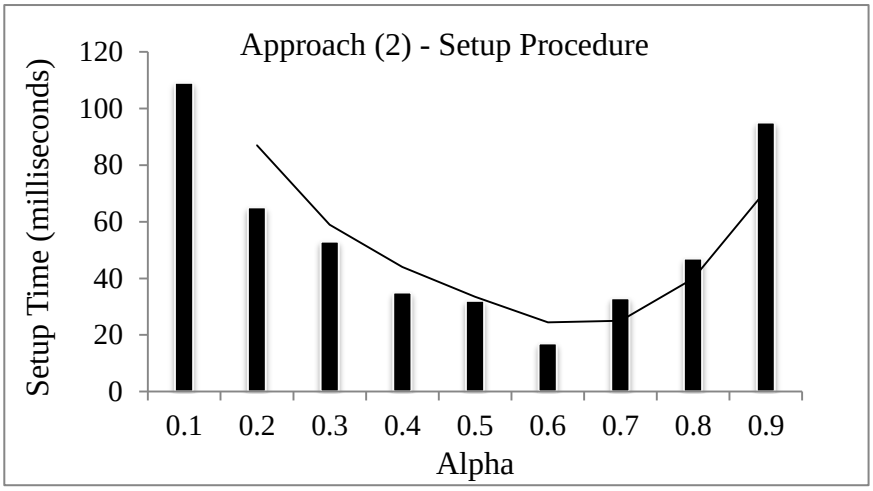


Figure 16 Variation of Setup Time of Approach (2) as a Function of Alpha

As for the simulation process, as in the case of any PHD example, the simulation process under Approach (2) is independent of the features of the underlying process, such as the order and variability, the impact of the variability, similar to that of the order, is shifted to the setup procedure (Figure 17). Therefore, the duration of the simulation process for a specific number of arrivals is not expected to vary largely under

Approach (2), which is exactly the case in this example. Similarly, no specific trend can be traced for the variation of the simulation time under Approach (1), indicating that the variability of the underlying process has little impact on the simulation process, unlike the order of the underlying Markov chain. It is also noticeable that simulation process under Approach (2) is faster and the overall execution procedure, inclusive of the setup is also faster than the simulation/execution procedure under Approach (1).

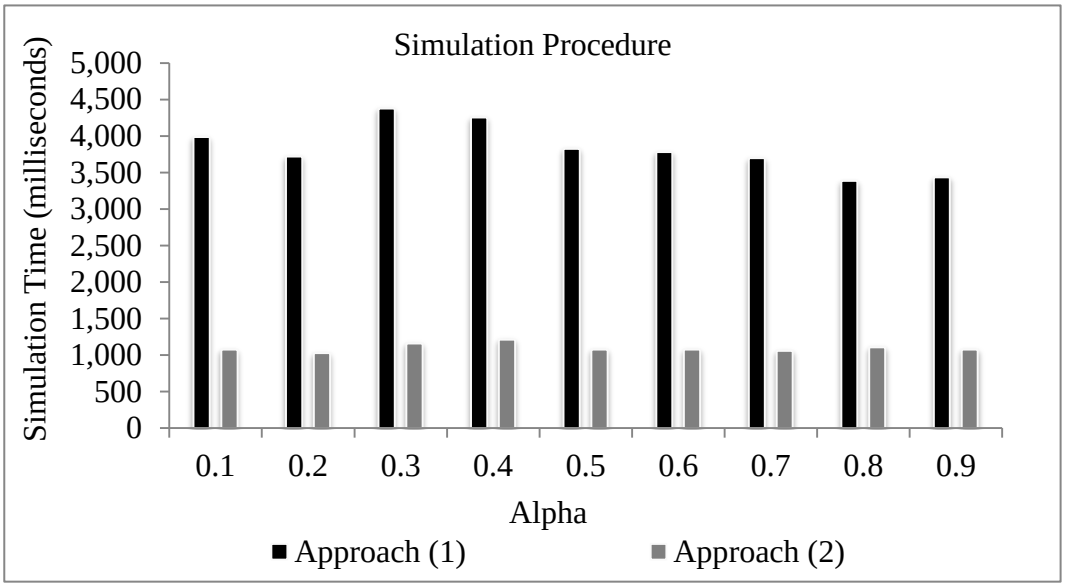


Figure 17 Variation of Simulation Time as a Function of Alpha

2. M/PH/1 Queue Model

We consider an M/PH/1 queue model; such the distribution of inter-arrival times is exponential with a constant arrival rate of 0.5 units per unit of time while that of the service times is phase-type, particularly the cases of the balanced two-level mixture of Erlangs analyzed in the previous section. There is only one server channel, the queue

discipline is FIFO (first-in-first-out) and the capacity of the system is assumed infinite. We are interested in analyzing the accuracy of either approach in estimating the performance measure or the steady-state quantities of the queue model under study while at the same time analyzing the impact of changing the variability of the service time on the estimation of the latter quantities. The performance measures consist of: the average number of units in the system L, the average number of units waiting in queue L_q, the average waiting time or the time spent in the system W and the average delay time or the time spent waiting in the queue before being served W_q.

The exact values of the steady-state quantities were computed according to closed-form equations of the M/G/1 queue model. Given that λ is the arrival rate and μ is the average service rate, then ρ is the capacity utilization ratio. The expected number of units waiting in queue is computed using (46) such that σ_S^2 is the variance of the service time. Using Little's law, the expected delay time can be calculated straightforwardly using (47). The expected waiting time is then the expected delay time plus the expected service time $1/\mu$ as given by (49). The expected number of units in the system can be derived using Little's law or as the summation of the expected number of units in the queue and the capacity utilization ratio.

$$L_q = \frac{\lambda^2 \sigma_S^2 + \rho^2}{2(1-\rho)}, \text{where } \rho = \frac{\lambda}{\mu} \tag{46}$$

$$W_q = \frac{L_q}{\lambda} \tag{47}$$

$$W = W_q + \frac{1}{\mu} \tag{48}$$

$$L = L_q + \rho = \lambda W \tag{49}$$

The average service rate is kept constant at 1; such that the service time is on average 1 unit of time. The variability of the distribution of the service time is varied for values of alpha ranging from 0.1 to 0.9. It is worth noting that the curves which depict the variation of the steady-state quantities as a function of alpha (Figure 18) are similar to that of the CV symmetric about α=0.5, such that the decrease in variability decreases all four performance measures and the opposite is true.

We constructed a simulation model for each case, such that the inter-arrival times for ten million units were first generated as random exponential random variables with an arrival rate of 0.5 units per time unit. The service times for the ten million units admitted into the system were generated first according to Approach (1) by simulating the underlying Markov chain and second according to Approach (2) by applying the proposed approximate inversion method. The same vector of inter-arrival times was used for both scenarios: (1) service times generated using Approach (1) and (2) service times generated using Approach (2).

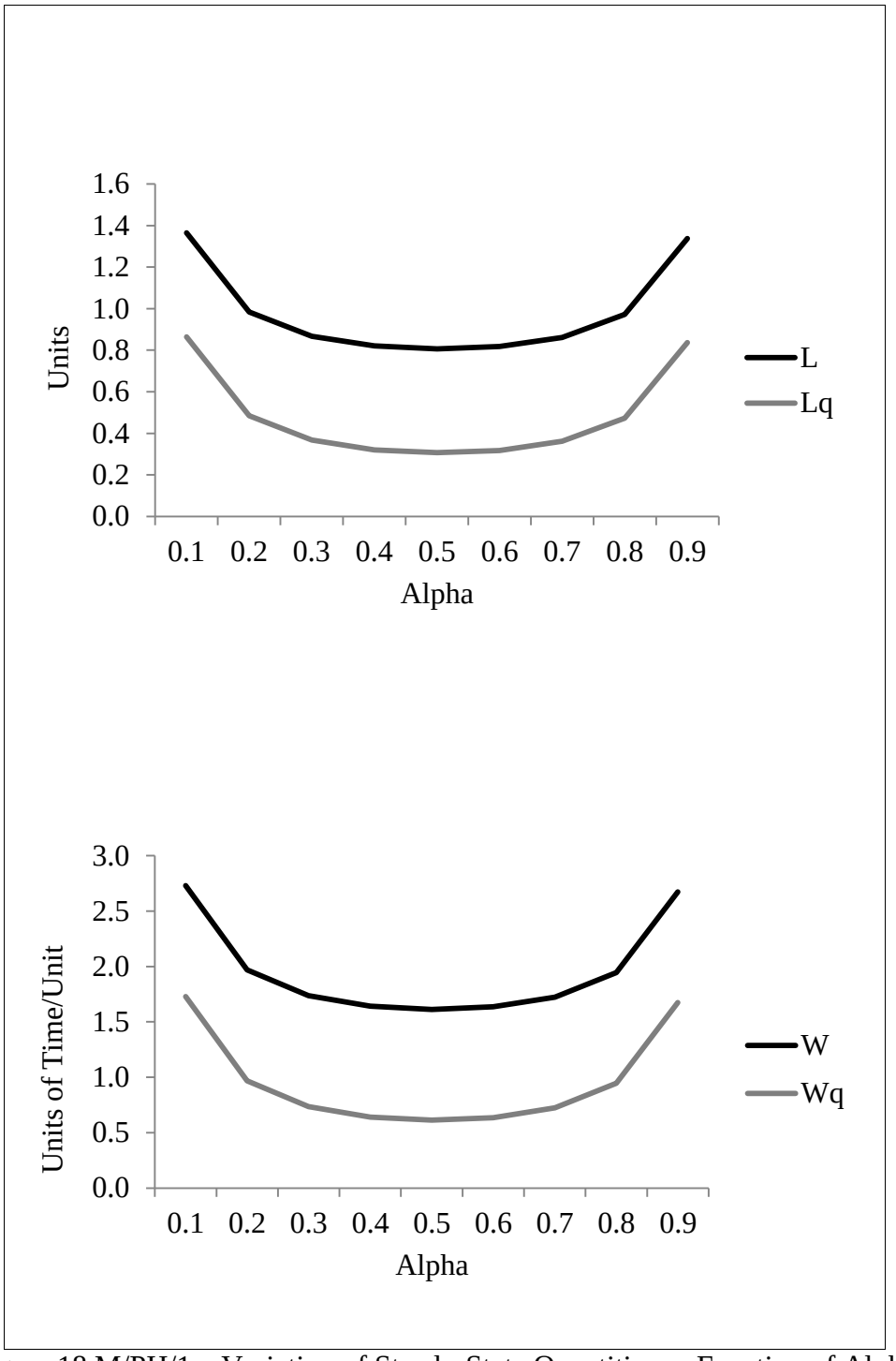


Figure 18 M/PH/1 – Variation of Steady-State Quantities as Function of Alpha

Let *M* be the maximum number of arrivals admitted into the system and served (ten million). If H_m, t_m^{Arr}, S_m and t_m^{Dep} represent the inter-arrival time, the arrival time,

the service time and the departure time of a unit *m*, where $H_1 = t_1^{Arr} = 0$, then the performance measures can be calculated using the following equations.

$$L_q = \frac{\sum_{m=1}^{M}\left(t_m^{Dep} - t_m^{Arr} - S_m\right)}{t_M^{Arr}} \tag{50}$$

$$W_q = \frac{\sum_{m=1}^{M}\left(t_m^{Dep} - t_m^{Arr} - S_m\right)}{M} \tag{51}$$

$$W = \frac{\sum_{m=1}^{M}\left(t_m^{Dep} - t_m^{Arr}\right)}{M} \tag{52}$$

$$L = \frac{\sum_{m=1}^{M}\left(t_m^{Dep} - t_m^{Arr}\right)}{t_M^{Arr}} \tag{53}$$

An error analysis was performed on the simulation estimates of the performance measures and the results are illustrated in Figures 19 and 20. A common variation trend is observed for all four quantities, which is expected since they are dependent. By generating inter-arrival times via the approximate inversion method under Approach (2), more accurate estimates are achieved as compared to the simulation of the Markov chain under Approach (1) when the variability of the underlying process is high (at extreme values of alpha), and the opposite is true. The decrease in the variability of the process generally decreases the absolute error of the estimates given by Approach (1), while increases that of the estimates given by Approach (2). However, generally, Approach (1) yielded better results, although the maximum absolute error under Approach (2) is approximately 0.3%, which is slightly greater than the 0.2% of Approach (1).

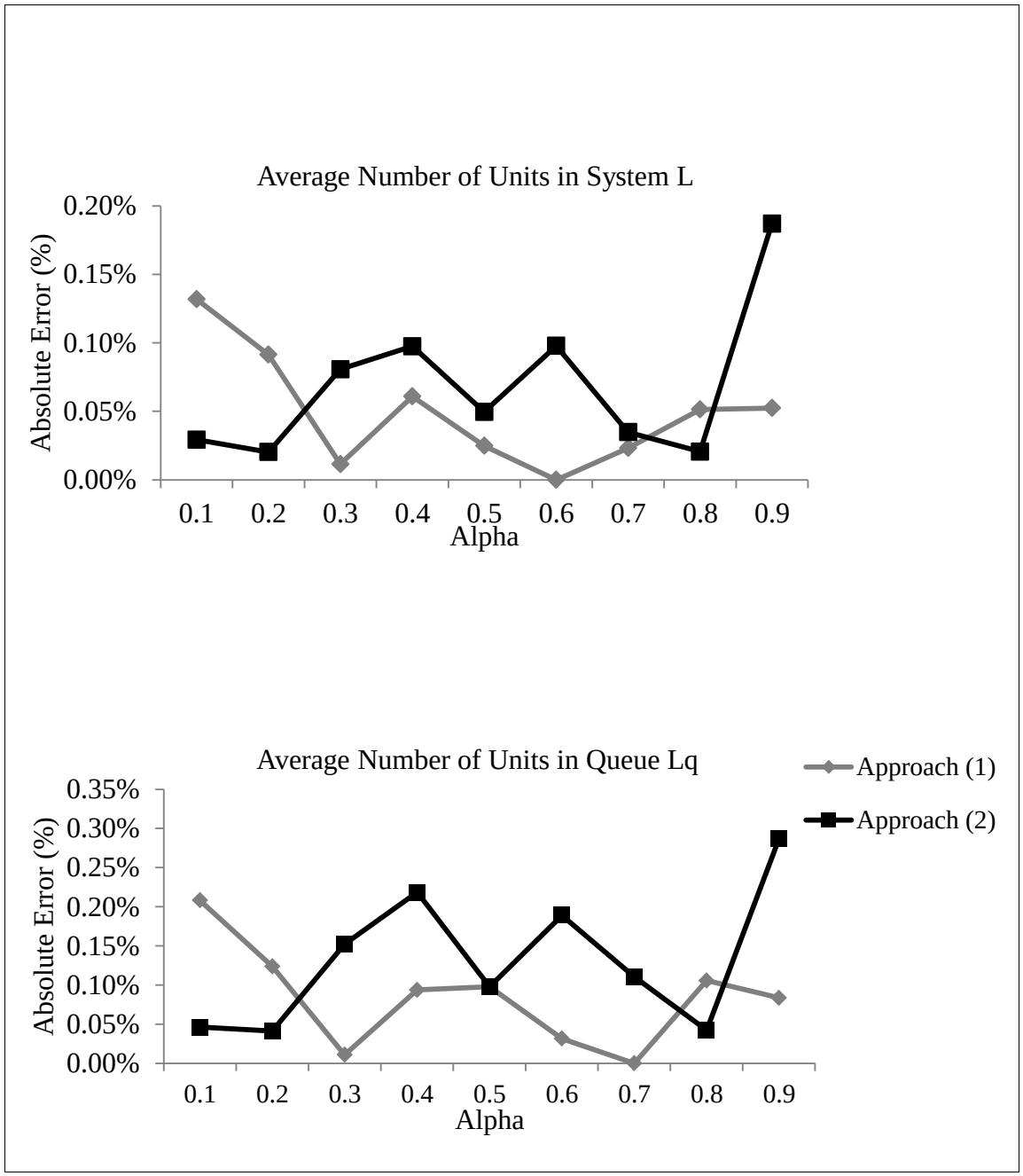


Figure 19 Variation of the Absolute Error of the Estimates of L and Lq

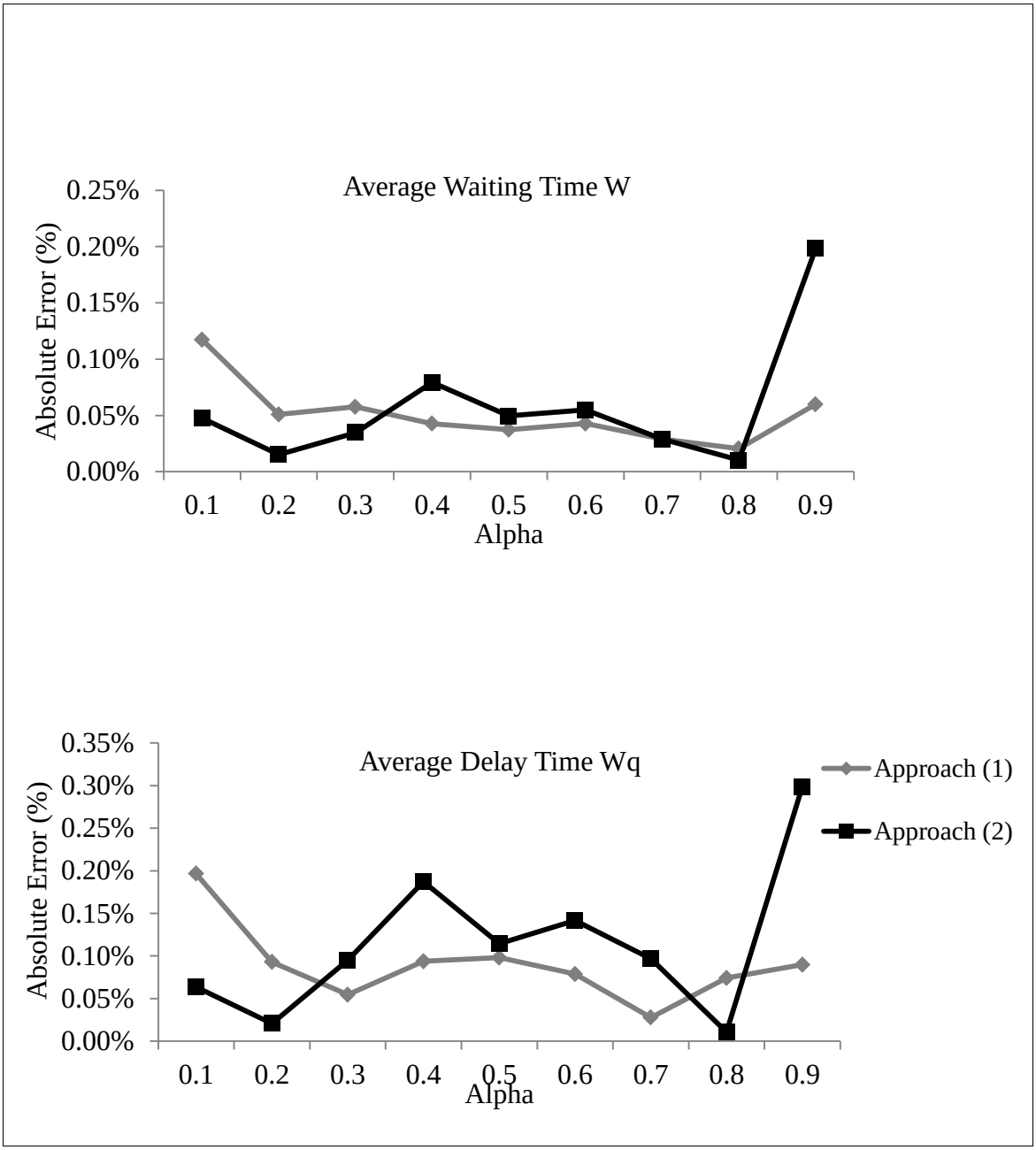


Figure 20 Variation of the Absolute Error of the Estimates of W and Wq

C. Effect of Correlation: MAP (4) Example

To assess the impact of correlation among the inter-arrival times in a MAP, we consider the case whose underlying Markov chain is illustrated in Figure 21. Hidden transitions are represented by solid arrows, while marked transitions associated with arrivals are represented by dashed arrows. Hidden transitions are fixed such that $\boldsymbol{D}_0$ is kept constant, while $\boldsymbol{D}_1$is varied with u. Increasing u is bound to increase the correlation among the inter-arrival times.

$$\boldsymbol{D_0} = \begin{bmatrix} -30 & 30 & 0 & 0 \\ 0 & -30 & 30 & 0 \\ 0 & 0 & -30 & 0 \\ 0 & 0 & 0 & -4 \end{bmatrix}$$

$$\boldsymbol{D_1}(u) = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 30u & 0 & 0 & 30(1-u) \\ 4(1-u) & 0 & 0 & 4u \end{bmatrix}$$

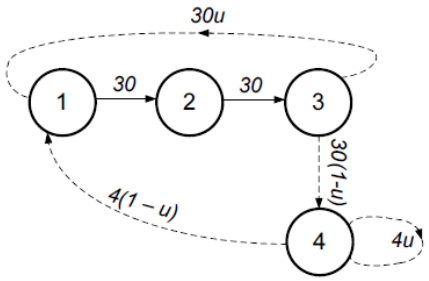


Figure 21 Underlying Markov Chain of MAP (4) Example

This example is formulated such that the mean inter-arrival time and its corresponding variance are kept constant, such that the impact of correlation can be more closely assessed. As the value of u increases from 0.1 to 0.9 (at a step size of 0.1), the correlation, represented in this case with the Lag-1 autocorrelation measure among the inter-event times of the process understudy increases linearly as shown in Figure 22.

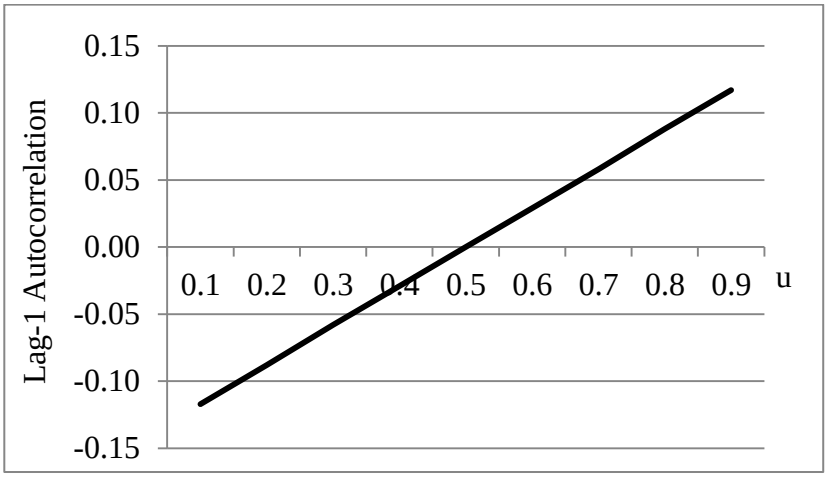


Figure 22 MAP (4) Example-Variation of Correlation

Tables 12 and 13 summarize the relative errors of the estimates of the mean, variance, skewness and correlation. Each variant is simulated for ten million arrivals, such that a stream of ten million inter-arrival times is generated.

For the mean inter-arrival time, the approximate inversion method via Approach (2) results in a slightly more accurate estimate than the simulation of the underlying Markov chain; however, the true mean has always fallen within the 95^{th} confidence interval of its estimate for either approach and the maximum absolute error is less than 1%. For either approach, the absolute error of the estimate of the mean is not sensitive to the degree of correlation among the events in the underlying process, such that no one clear trend can be depicted for the variation of the absolute error of the estimate as the correlation increases.

Generally, both simulation approaches accurately capture the variability of the underlying process and the correlation among subsequent arrival events. However,

simulation via the approximate inversion method results in a better approximation of the skewness of the corresponding distribution, whereby it more accurately reflects the shape of the variation of the cumulative distribution function. Similar to the estimate of the mean inter-arrival times, the absolute error of the estimates of the variance, skewness and correlation is independent of the degree of correlation in the underlying process, whereby increasing the correlation among the events does not affect the capability of either simulation approach in accurately approximating the quantities understudy.

Table 12 MAP (4) Example-Error Analysis u [0.1,0.5]

Relative Percent Error of Estimate					
u	0.1	0.2	0.3	0.4	0.5
Lag-1 Autocorrelation	-0.117	-0.088	-0.058	-0.029	0.000
Approach (1)					
Mean	-0.0095%	-0.0476%	0.0286%	-0.0476%	0.0286%
Variance	-2.5641%	0.0000%	0.0000%	0.0000%	0.0000%
Skewness	-0.1352%	0.1014%	0.1014%	-0.1352%	0.1014%
Lag-1 Autocorrelation	0.0000%	0.0000%	1.7241%	0.0000%	0.0000%
Approach (2)					
Mean	0.0571%	-0.0286%	0.0667%	0.0095%	0.0381%
Variance	0.0000%	0.0000%	0.0000%	0.0000%	0.0000%
Skewness	-0.0338%	0.0000%	0.0338%	0.1014%	0.0676%
Lag-1 Autocorrelation	0.0000%	0.0000%	1.7241%	0.0000%	0.0000%

Table 13 MAP (4) Example-Error Analysis u [0.6,0.9]

Relative Percent Error of Estimate				
u	0.6	0.7	0.8	0.9
Lag-1 Autocorrelation	0.029	0.058	0.088	0.117
Approach (1)				
Mean	-0.0190%	0.0190%	0.1048%	0.0857%
Variance	0.0000%	0.0000%	0.0000%	0.0000%
Skewness	0.1014%	0.0676%	-0.0676%	-0.1352%
Lag-1 Autocorrelation	0.0000%	1.7241%	0.0000%	0.0000%
Approach (2)				
Mean	0.0476%	0.0190%	0.0190%	-0.0286%
Variance	0.0000%	0.0000%	0.0000%	0.0000%
Skewness	0.2028%	0.0000%	-0.1352%	0.2028%
Lag-1 Autocorrelation	0.0000%	0.0000%	0.0000%	-0.8547%

As for the performance of Approach (2), the setup time mostly follows a decreasing trend, albeit a very slightly gradual one, which can be attributed to the presumption that increasing the correlation in the underlying process resulted in less extensive matrix operations. However, unlike the setup time, the simulation time does not seem to maintain a specific trend which can be attributed to varying the correlation. Figure 23 illustrates the variation of the duration of the execution of Approach (2), distinguishing between the durations of the setup and simulation procedures.

On the other hand, Figure 24 illustrates the variation of the execution time of Approach (1) which is effectively the simulation time. Similar to Approach (2), the simulation times generally fluctuates closely about its average such that the impact of increasing the correlation among arrival events is irrelevant to the performance of Approach (1). Nonetheless, despite the almost equal accuracy of the estimates

approximated by both approaches, simulation of the underlying Markov chain via Approach (1) remains almost three times faster than its counterpart.

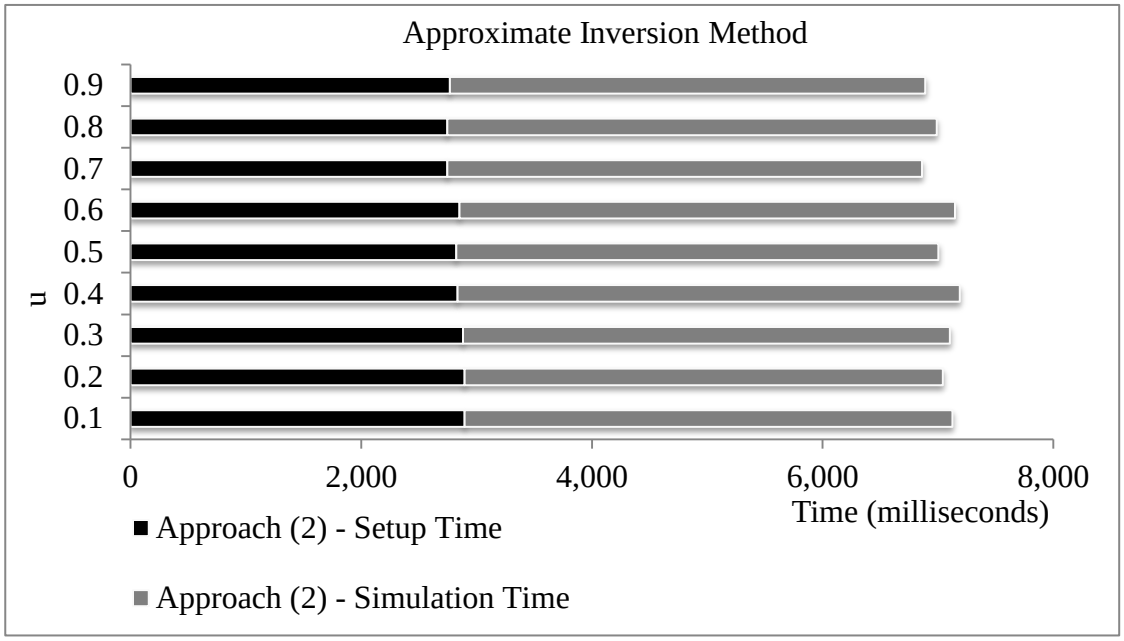


Figure 23 MAP (4) Example-Variation of Execution Time of Approach (2)

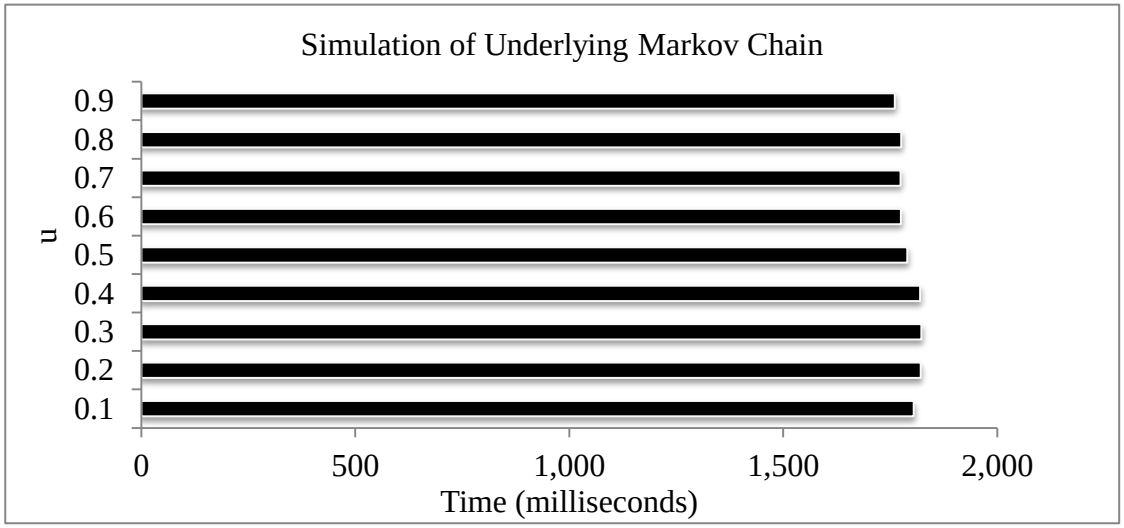


Figure 24 MAP (4) Example-Variation of Simulation Time of Approach (1)

CHAPTER XI

CONCLUSION

In this research, we proposed the simulation of three point processes, the Phase-Type Distribution (PHD), the Markovian Arrival Process (MAP) and the Batch Markovian Arrival Process (BMAP) such that inter-arrival times are randomly generated via an approximate inversion method, referred to as Approach (2). We used the simulation of the underlying Markov chain as a reference approach (Approach 1) to which our proposal can be compared and thusly assessed. The power of Approach (2) lies in its generality, such that given the parameters of any distribution; a database of time-values vs. CDF-values can be compiled and used to perform simulation of corresponding distributions and/or stochastic processes. The approximate nature of the algorithm doesn't render it inaccurate; the relative errors of the estimates of the mean inter-arrival time, its variance and skewness of its distribution are generally acceptable. However, neither approach was capable of producing an accurate degree of correlation among events in the case of MAPs and BMAPs, specifically when simulating randomly and fully populated MAPs and BMAPs.

For PHDs, the cumulative distribution function of the inter-arrival time is invertible, so we suggested discretizing it by formulating a time-CDF value database, whereby we create an equally-spaced time-values vector and a corresponding CDF-values vector. The spacing of the time-values vector was derived based on Taylor series expansion and linear approximation utilizing the fact that the cumulative distribution

function is differentiable about the non-negative axis. It is worth noting that the time-values vector is truncated whenever the CDF converges to 1.0. For every arrival epoch, a uniform random number ranging from 0 to 1 inclusively is generated and located in the CDF-values vector via the binary search technique or the bisection method, and accordingly the binding values are used to linearly interpolate an estimate of the corresponding value of the inter-arrival time. We also developed a PHD-sensitive Stochastic Simulation Algorithm (SSA) to model the transitional behavior in the underlying Markov chains as a reference approach to which Approach (2) can be compared. The simulation process follows the transition dynamics of an arrival epoch until it hits the absorbing state.

To assess the efficiency of Approach (2) in light of the performance of Approach (1), we ran both algorithms on the same set of examples. First, we generated randomly-populated PHD examples and ran them for 1 million arrivals. Approach (2) tends to be faster than its counterpart, namely the simulation speed under Approach (2) is higher than that under Approach (1) especially as the order of the underlying Markov chain increases. The sensitivity of either approach to the order of the PHD is reflected by the increase in the duration of the setup procedure of Approach (2) and the increase in the duration of the simulation process of Approach (1). This implies that under Approach (2), the effect of the order of the PHD or the underlying Markov chain is shifted from the simulation procedure as in Approach (1), to the setup procedure. The setup procedure of Approach (2) can be conducted once per example and used to generate random inter-event times innumerably, as opposed to Approach (1), which has can be reiterated every time a set of inter-event times is to be generated. Additionally,

we applied both approaches to simulate a specific case of a balanced two-level mixture of Erlangs of order 9. The mean is held constant at 1 minute, yet the variance was changed over 9 variants of the same example, and the resulting variation of the CV (coefficient of variation) was recorded. It was deduced that decreasing the variability of the underlying process decreases the duration of the setup procedure of Approach (2) and improves the accuracy of the estimates under both approaches. It was also noted that the curves depicting the variation of the CV, length of the 95^{th} confidence interval for both approaches and the setup time of Approach (2) have similar variation trends. Additionally, an M/PH/1 queuing model was also simulated using both approaches and the estimates of the performance measures or the steady-state queue quantities by Approaches (1) and (2) are comparable with minimal deviation from their true values.

Unlike PHDs, Approach (1) proved to be faster than Approach (2), yet equally accurate, for fully populated MAPs and BMAPs; however, the difference in the durations of the execution procedures of both approaches differs only slightly, and is only strengthened when more than 10 million arrivals are simulated at one time. In general, the probability of arrivals in a MAP or a MAP is high because any state can act as an absorbing state. Therefore, imposing some restrictions on marked transitions matrix by restricting marked transitions to/from specific states or decreasing the probability of those transitions, renders Approach (2) much more efficient as a simulation method as compared to Approach (1). Additionally, we tested the impact of the variation of the degree of correlation among inter-arrival times on the efficiency of simulation, and generally no one pattern or trend can be realized, implying the insensitivity of either approach to the increase in the correlation.

APPENDIX I

RESULTS OF THE SIMULATION OF RANDOMLY POPULATED PHD EXAMPLES

Table 14 Simulation Results of Random PHD Examples – Approach (1)

Number of Arrivals: 1,000,000					Approach (1) Simulation of Markov Chain			
Order	Mean (seconds)	Variance (10^{-3})	Skewness	CV	Average Inter-arrival Time (seconds)	95th Confidence Interval	Sample Variance (10^{-3})	Sample Skewness
1	18.695	97.087	2.000	1.000	18.723	[18.687,18.760]	97.391	1.998
2	0.677	0.140	2.069	1.050	0.676	[0.674,0.677]	0.140	2.064
3	0.875	0.214	1.922	1.004	0.876	[0.874,0.878]	0.214	1.920
4	0.711	0.143	2.028	1.011	0.712	[0.711,0.714]	0.144	2.028
5	1.561	0.798	2.063	1.085	1.559	[1.556,1.563]	0.794	2.056
6	1.072	0.293	1.963	0.959	1.072	[1.070,1.074]	0.292	1.950
7	1.092	0.329	1.992	0.996	1.093	[1.090,1.095]	0.329	1.997
8	1.305	0.439	1.980	0.964	1.305	[1.303,1.308]	0.438	1.977
9	0.911	0.246	2.044	1.034	0.910	[0.909,0.912]	0.246	2.035
10	1.649	0.873	2.042	1.075	1.65	[1.647,1.654]	0.875	2.049

Table 15 Simulation Results of Random PHD Examples – Approach (2)

Number of Arrivals: 1,000,000					Approach (2) Approximate Inversion Method			
Order	Mean (seconds)	Variance (10^{-3})	Skewness	CV	Average Inter-arrival Time (seconds)	95th Confidence Interval	Sample Variance (10^{-3})	Sample Skewness
1	18.695	97.087	2.000	1.000	18.723	[18.687,18.760]	97.391	1.998
2	0.677	0.140	2.069	1.050	0.676	[0.674,0.677]	0.140	2.064
3	0.875	0.214	1.922	1.004	0.876	[0.874,0.878]	0.214	1.920
4	0.711	0.143	2.028	1.011	0.712	[0.711,0.714]	0.144	2.028
5	1.561	0.798	2.063	1.085	1.559	[1.556,1.563]	0.794	2.056
6	1.072	0.293	1.963	0.959	1.072	[1.070,1.074]	0.292	1.950
7	1.092	0.329	1.992	0.996	1.093	[1.090,1.095]	0.329	1.997
8	1.305	0.439	1.980	0.964	1.305	[1.303,1.308]	0.438	1.977
9	0.911	0.246	2.044	1.034	0.910	[0.909,0.912]	0.246	2.035
10	1.649	0.873	2.042	1.075	1.65	[1.647,1.654]	0.875	2.049

APPENDIX II

RESULTS OF THE SIMULATION OF RANDOMLY POPULATED MAP EXAMPLES

Table 16 Randomly Populated MAP Examples-True Values of the Mean, Variance, and Skewness

Order	Mean (seconds)	Variance (10^{-4})	Skewness
2	0.992	2.703	1.986
3	0.725	1.412	1.954
4	0.385	0.410	1.981
5	0.329	0.305	2.004
6	0.235	0.158	2.035
7	0.195	0.105	1.999
8	0.179	0.088	1.989
9	0.157	0.070	2.022
10	0.132	0.049	2.009

Table 17 Randomly Populated MAP Examples – Simulation Results of Approach (1)

Approach (1) Simulation of Markov Chain				
Order	Average Inter-arrival Time (seconds)	95^{th} Confidence Interval	Sample Variance (10^{-4})	Sample Skewness
2	0.992	[0.992,0.993]	2.704	1.989
3	0.725	[0.725,0.726]	1.411	1.954
4	0.385	[0.385,0.385]	0.410	1.981
5	0.329	[0.328,0.329]	0.305	2.003
6	0.235	[0.235,0.235]	0.158	2.038
7	0.195	[0.195,0.195]	0.105	1.997
8	0.179	[0.179,0.179]	0.088	1.993
9	0.157	[0.157,0.158]	0.070	2.021
10	0.132	[0.132,0.132]	0.049	2.012

Table 18 Randomly Populated MAP Examples – Simulation Results of Approach (2)

Approach (2) Approximate Inversion Method				
Order	Average Inter-arrival Time (seconds)	95^{th} Confidence Interval	Sample Variance (10^{-4})	Sample Skewness
2	0.992	[0.992,0.993]	2.700	1.985
3	0.725	[0.725,0.726]	1.410	1.953
4	0.385	[0.385,0.385]	0.410	1.980
5	0.329	[0.329,0.330]	0.305	2.004
6	0.235	[0.235,0.235]	0.158	2.034
7	0.195	[0.194,0.195]	0.105	1.997
8	0.179	[0.179,0.179]	0.088	1.987
9	0.157	[0.157,0.158]	0.070	2.023
10	0.132	[0.132,0.132]	0.049	2.010

APPENDIX III

RESULTS OF THE SIMULATION OF RANDOMLY POPULATED BMAP EXAMPLES

Table 19 Randomly Populated BMAP Examples – Simulation Results of Approach (1)

Number of Arrival Epochs: 1,000,000				Approach (1) Simulation of Markov Chain	
Order	Batch Size	Mean (seconds)	CV	Average Inter-arrival Time (seconds)	95th Confidence Interval
2	2	0.4765	1.0071	0.4760	[0.4751,0.4770]
	3	0.5471	1.0451	0.5481	[0.5470,0.5492]
	4	0.3108	0.9969	0.3106	[0.3100,0.3112]
	5	0.2982	1.0124	0.2983	[0.2977,0.2989]
3	2	0.2699	1.0115	0.2699	[0.2693,0.2704]
	3	0.2418	0.9975	0.2416	[0.2411,0.2421]
	4	0.1363	1.0361	0.1364	[0.1362,0.1367]
	5	0.1318	1.0032	0.1317	[0.1315,0.1320]
4	2	0.2557	0.9987	0.2552	[0.2547,0.2557]
	3	0.1296	1.0098	0.1295	[0.1293,0.1298]
	4	0.0987	0.9975	0.0988	[0.0986,0.0989]
	5	0.0779	1.0149	0.0779	[0.0776,0.0781]
5	2	0.1526	1.0073	0.1528	[0.1525,0.1531]
	3	0.0987	1.0023	0.0986	[0.0985,0.0988]
	4	0.0736	1.0036	0.0736	[0.0735,0.0738]
	5	0.0609	1.0223	0.0609	[0.0607,0.0610]
6	2	0.1392	1.0124	0.1391	[0.1389,0.1394]
	3	0.0840	1.0035	0.0840	[0.0838,0.0842]
	4	0.0585	1.0039	0.0586	[0.0585,0.0587]
	5	0.0459	1.0044	0.0459	[0.0458,0.0460]
7	2	0.1039	1.0155	0.1039	[0.1037,0.1042]
	3	0.0633	1.0138	0.0633	[0.0632,0.0634]
	4	0.0487	1.0112	0.0488	[0.0487,0.0488]
	5	0.0400	1.0045	0.0400	[0.0399,0.0401]
8	2	0.0854	1.0070	0.0854	[0.0853,0.0856]
	3	0.0532	1.0006	0.0531	[0.0530,0.0532]
	4	0.0437	1.0019	0.0437	[0.0436,0.0438]
	5	0.0331	1.0090	0.0331	[0.0331,0.0332]

Table 20 Randomly Populated BMAP Examples – Simulation Results of Approach (2)

Number of Arrival Epochs: 1,000,000				Approach (2) Approximate Inversion Method	
Order	Batch Size	Mean (seconds)	CV	Average Inter-arrival Time (seconds)	95th Confidence Interval
2	2	0.4765	1.0071	0.4764	[0.4755,0.4774]
	3	0.5471	1.0451	0.5473	[0.5462,0.5485]
	4	0.3108	0.9969	0.3110	[0.3104,0.3116]
	5	0.2982	1.0124	0.2983	[0.2977,0.2988]
3	2	0.2699	1.0115	0.2700	[0.2694,0.2705]
	3	0.2418	0.9975	0.2422	[0.2417,0.2427]
	4	0.1363	1.0361	0.1362	[0.1359,0.1365]
	5	0.1318	1.0032	0.1317	[0.1315,0.1320]
4	2	0.2557	0.9987	0.2553	[0.2548,0.2558]
	3	0.1296	1.0098	0.1298	[0.1295,0.1300]
	4	0.0987	0.9975	0.0987	[0.0985,0.0989]
	5	0.0779	1.0149	0.0780	[0.0778,0.0782]
5	2	0.1526	1.0073	0.1528	[0.1525,0.1531]
	3	0.0987	1.0023	0.0986	[0.0984,0.0988]
	4	0.0736	1.0036	0.0735	[0.0734,0.0737]
	5	0.0609	1.0223	0.0610	[0.0608,0.0611]
6	2	0.1392	1.0124	0.1391	[0.1389,0.1394]
	3	0.0840	1.0035	0.0840	[0.0839,0.0842]
	4	0.0585	1.0039	0.0584	[0.0583,0.0586]
	5	0.0459	1.0044	0.0459	[0.0458,0.0460]
7	2	0.1039	1.0155	0.1039	[0.1037,0.1041]
	3	0.0633	1.0138	0.0633	[0.0632,0.0634]
	4	0.0487	1.0112	0.0487	[0.0486,0.0488]
	5	0.0400	1.0045	0.0400	[0.0399,0.0401]
8	2	0.0854	1.0070	0.0854	[0.0852,0.0856]
	3	0.0532	1.0006	0.0532	[0.0531,0.0533]
	4	0.0437	1.0019	0.0437	[0.0436,0.0438]
	5	0.0331	1.0090	0.0331	[0.0330,0.0332]

BIBLIOGRAPHY

1. P. Buchholz, J. Kriege and I. Felko, I "Input Modeling with Phase-Type Distributions and Markov Models, Theory and Applications", Springer Cham Heidelberg New York Dordrecht London (2014) pp. 1-27, pp. 105-114
2. H.T. Banks, A. Broido, B. Canter, K. Gayvert, S. Hu, M. Joyner and K. Link "Simulation Algorithms for Continuous Time Markov Chains", (2011)
3. K. Sigman, "Simulating Markov Chains", Excerpt (2007)
4. J.D. Cordeiro and J.P. Kharoufeh, "Batch Markovian Arrival Processes (BMAP)", University of Pittsburgh, Directorate of Force Management Policy Headquarters, United States Air Force (2010)
5. , M.F. Neuts and M. Pagano "Generating Random Variates from a Distribution of Phase Type", University of Delaware, Winter Simulation Conference Proceedings (1981)
6. B.U. Narayan, "An Introduction to Queuing Theory: Modeling and Analysis in Applications", 2nd Edition (2015), Chapter 3, Markovian Point Processes
7. B.U. Narayan, "An Introduction to Queuing Theory: Modeling and Analysis in Applications", 2nd Edition (2015), Chapter 10, Markovian Arrival Processes
8. E. Brown, J. Place and A.V Liefvoort, "Generating Matrix Exponential Random Variates", University of Missouri-Kansas City, Simulation Councils 70:4(1998), pp. 224-230
9. W.J. Stewart, "Probability, Markov Chains, Queues, and Simulation", Princeton University Press, Princeton (2009)

10. A.M. Law and W.D. Kelton, “Simulation Modeling and Analysis”, 3rd Edition, McGraw Hill, Boston (2000)

11. F. Bause, P. Buchholz, and J. Kriege, “A Comparison of Markovian Arrival Processes and ARMA/ARTA Processes for the Modeling of Correlated Input Processes”, Proceedings of the Winter Simulation Conference (2009)

12. B. Biller and C. Gunes, “Introduction to Simulation Input Modeling”, Proceedings of the Winter Simulation Conference (2010), pp. 49-58

13. A.M. Law and M.G. McComas, “ExpertFit Distribution Fitting Software: How ExperFit Distribution Fitting Software Makes Simulation Models more Valid”, Proceedings of the Winter Simulation Conference, (2003), pp. 169-174

14. W.E. Leland, M.S. Taqqu, W. Willinger and D.V. Wilson, “On the Self-Similar Nature of the Internet Traffic”, IEEE/ACM Transportation Networks (1994), pp. 1-15

15. J. Bilmes, “A Gentle Tutorial on the EM Algorithm and its Application to Parameter Estimation for Gaussian Mixture and Hidden Markov Models”, University of Berkley (1997)

16. V. Paxson and S. Floyd, “Wide Area Traffic: The Failure of Poisson Modeling”, IEEE/ACM Transportation Networks (1995), pp. 1-6

17. M. Rahnamay-Naeini, J.E. Pezoa, G. Azar, N. Ghani, and M.M. Hayat, “Modeling Stochastic Correlated Failures and their Effects on Network Reliability”, Proceedings of the 20th International Conference on Computer Communications and Networks (ICCCN), (2011), pp. 1-6

18. C.A. O’Cinneide, “Phase-Type Distributions: Open Problems and a Few Properties”, (1999), pp. 731-757

19. A. Bobbio and M. Telek, “Parameter Estimation of Phase Type Distributions”, Instituto Elettrotechnico Nazional Galileo Ferraris, (1997)

20. C.H. Sauer and K.M. Candy, “Computer Systems Performance Modeling”, Prentice Hall, Englewood Cliffs, (1981)

21. M.F. Neuts, “A Versatile Markovian Point Processes”, Journal of Applied Probability Vol:16, (1979), pp. 764-779

22. M.F. Neuts, “Matrix-Geometric Solutions in Stochastic Models”, John Hopkins University Press, Baltimore (1981)

23. A. Horváth and M. Telek, “Markovian modeling of real data traffic: Heuristic phase type and MAP fitting of heavy tailed and fractal like samples”, Proceedings of the Performance 2002. Lecture Notes in Computer Science, Vol. 2459 (2002), pp. 405–434.

24. H. Okamura, T. Dohi and K.S. Trivedi, “Markovian arrival process parameter estimation with group data”, IEEE/ACM Transportation Networks, (2009), pp. 1326–1339

25. M. Telek and G.A. Horváth, “A minima lrepresentation of Markov arrival processes and a moments matching method”, (2007)

26. S. Heckmüller and B.E. Wolanger, “Using load transformations for the specification of arrival processes in simulation and analysis”, (2009)

27. C. Brickner, D. Indrawan, D. Williams and S.R. Chakravarthy, “Simulation of a stochastic model for a service system”, Proceedings of the Winter Simulation Conference (2010), pp. 1636–1647

28. A. Thümmler, P. Buchholz and M. Telek, “A novel approach for phase-type fitting with the EM algorithm”, IEE, (2006)

29. A. Riska, V. Diev and E. Smirni, “An EM-based technique for approximating long-tailed data sets with PH distributions”, (2004)

30. Y. Takahashi, “Asymptotic exponentiality of the tail of the waiting-time distribution in a PH/PH/c queue”, Advanced Applied. Probably, (1981)

31. W.E. Leland, R.P. Sadowski and D.A. Sadowski, “Simulation with Arena”, 4th Edition, McGraw Hill, New York (2007)

32. M.L. Liou, “A novel method of evaluating transient response”, Proceedings of IEEE, 54 (1966), pp. 20-23

33. W. Everling, “On the evaluation of matrix exponential by power series”, Proceedings of IEEE, 55 (1967), p.413

34. T.A. Bickart, “Matrix exponential: Approximation by truncated power series”, Proceeding of IEEE, 56 (1968), pp.372-373

35. V. Zakian, “Rational approximants to the matrix exponential”, Electronics Letters, 6 (1970), pp. 814-815

36. A. Wragg and C. Davies, “Computation of the exponential of a matrix II: Practical considerations”, Journal of the Institute of Mathematics and its Applications., 15 (1975), pp. 273-278

37. R.C. Ward, “Numerical computation of the matrix exponential with accuracy estimate”, SIAM, Journal on Numerical Analysis, 14 (1977), pp. 600-610

38. D.W. Kammler, “Numeric Evaluation of the matrix exponential”, University of Southern Illinois, Carbondale, IL, 1976

39. C. Kallstrom, “Computing exp(A) and ∫exp(As) ds”, report 7309, Division of Automatic Control, Lund Institute of Technology, Lund, Sweden, 1973

40. R.E. Scraton, “Comment on rational approximants to the matrix exponential”, Electronic Letters, 7 (1971), pp. 260-261

41. M.M. Shah, “On the evaluation of exp(At)”, Cambridge Report CUED/B-Control TR8, Cambridge, England, (1971)

42. M.M. Shah, “Analysis of round off and truncation errors in the computation of transition matrices”, Cambridge Report CUED/B-Control TR12, Cambridge, England, (1971)

43. G.H. Golub and J.H. Wilkinson, “Ill-conditioned Eigen systems and the computation of the Jordan canonical form”, SIAM Review, 18 (1976), pp. 578-619

44. B. Kågstrom and A. Ruhe, “An algorithm for numerical computation of the Jordan normal form of a complex matrix”, Report UMINF 51.74, Department of Information Processing, University of Umea, Umea, Sweden, 1974, Subsequently published in ACM Transactions on Mathematical Software, 6 (1980), pp. 398-419

45. W. Nasr and M. Taffe, “Fitting the $Ph_t/M_t/s/c$ Time Dependent Departure Process for Use in Tandem Queuing Networks”, INFORMS Journal on Computing, 25(4), (2013), pp. 758-773

46. M.F. Neuts, “The burstiness of point processes”, Stochastic Models, Vol. 9, (1993) pp. 445-466

47. J.R. Artalejo, A. Gomez-Corral and Q. He, “Markovian arrivals in stochastic modeling: a survey and some new results”, SORT 34 (2), (2010), pp. 101-144