Department of Computer & Information Science

Technical Reports (CIS)

University of Pennsylvania *Year* 1991

Performance Evaluation via Perturbation Analysis

Tarek M. Sobh
University of Pennsylvania

Performance Evaluation via Perturbation Analysis

MS-CIS-91-38
GRASP LAB 263

Tarek M. Sobh

Department of Computer and Information Science
School of Engineering and Applied Science
University of Pennsylvania
Philadelphia, PA 19104-6389

May 1991

Performance Evaluation via Perturbation Analysis

Tarek M. Sobh

GRASP Laboratory

Department of Computer and Information Science

University of Pennsylvania, Philadelphia, PA 19104

Abstract

In this paper we present an overview for the development of a theory for analyzing and predicting the behaviour of discrete event dynamic systems (DEDS). DEDS are dynamic systems in which state transitions are caused by internal, discrete events in the system. DEDS are attracting considerable interests, current applications are found in manufacturing systems, communications and air traffic systems, future applications will include robotics, computer vision and artificial intelligence. We will discuss the perturbation analysis technique (PA) for evaluating the performance of DEDS.

Keywords : Communication Networks, Control Theory, Dynamic Systems, Discrete Event Systems, Perturbation Analysis, Performance Evaluation, Queueing Networks.

1 Introduction

In this paper, we describe a recently developed framework for analyzing and evaluating the performance of discrete event dynamic systems (DEDS) called perturbation analysis (PA) [1,2,8]. The approach used in this framework is a quantitative approach that focuses on the performance measures of DEDS. There are other state space approaches that concentrate on the qualitative aspects of DEDS [6,7,9,10], however, we shall concern ourselves only with the PA technique as it is more suitable for analyzing communication networks.

Discrete event dynamic systems (DEDS) are dynamic systems (typically asynchronous) in which state transitions are triggered by the occurrence of discrete events in the system. Many existing dynamic system have a DEDS structure, manufacturing systems and communication systems are just two of them. The PA approach to analyzing DEDS is different from the analysis techniques for the state space approach, the existence of a consistent and pre-defined automata-like model of the system under consideration is not necessary to perform PA. For example, if we consider a serial production line with M stations with a queue space of size K_i for each station. Then the total number of states for such a system would be $(\prod_{i=1}^{M}(K_i+1))(2^M)$, which can amount to billions for relatively small values of K_i and M. It is quite clear that modeling such systems as finite state machines is inefficient, if not impossible. It should also be mentioned that the finite state machines approach is more suitable for answering qualitative rather than quantitative questions.

Perturbation analysis (PA) is a technique that calculates the sensitivity of performance measures of DEDS with respect to system parameters by analyzing its sample path. The object of PA is to obtain the perturbed performance from a nominal experiment or sample path without doing a perturbed experiment. To avoid doing more than one experiment or simulate a perturbed experiment is the goal of PA.

2 Infinitesimal Perturbation Analysis (IPA)

To present the idea behind IPA, we shall first introduce a simple system (see Figure 1). It consists of a buffer, call it A, where messages arrive and are placed in a FIFO queue, and is connected via a link to another buffer, call it B, where the messages are received.

Consider the following definitions:

$$\theta = \text{link service time (s/bit)}$$

$$H = \text{header length (bits)}$$

L_i = length of message i (bits)

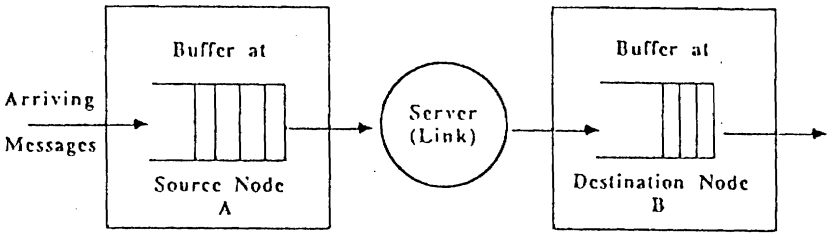


Fig. 1. Link in a communication network.

We define the "service time" to be the time it takes to transmit a message i from A to B assuming the message does not wait in the queue before it gets sent. We denote this by

$$\begin{aligned} X_i &= (H + L_i)\theta \\ &= \gamma + L_i\theta. \end{aligned} \tag{1}$$

Let us also define the "system time", t_i, to be the time since a message i arrives at A till it is completely received by B. Finally let us call our performance measure $T(\theta, \gamma)$. This can be approximated by using the mean system time, $\hat{T}(\theta, \gamma, N)$, where

$$\hat{T}(\theta, \gamma, N) = \left(\frac{1}{N}\right) \sum_{i=1}^{N} t_i. \tag{2}$$

Note that as $N \to \infty$, $\hat{T}(\theta, \gamma, N)$ converges to $T(\theta, \gamma)$.

For sensitivity estimates, we use $dT/d\theta$ and $dT/d\gamma$. A good estimate for $dT/d\theta$ is

$$\hat{F} = [\hat{T}(\theta + \Delta\theta, \gamma, N) - \hat{T}(\theta, \gamma, N)]/\Delta\theta. \tag{3}$$

Similarly a good estimate for $dT/d\gamma$ is

$$\hat{G} = [\hat{T}(\theta, \gamma + \Delta\gamma, N) - \hat{T}(\theta, \gamma, N)]/\Delta\gamma. \tag{4}$$

As can be seen, to obtain the estimates above one needs one more experiment at $\theta + \Delta\theta$ and another at$\gamma + \Delta\gamma$.

The problem here is to choose a value for $\Delta\theta$ (and similarly $\Delta\gamma$). For, if we choose to large a value we will not get a good estimate of the gradient. On the other hand, if we

choose $\Delta\theta$ to be too small, we may amplify the noise interference present in $\hat{T}(\theta, \gamma + \Delta\gamma, N)$ and $\hat{T}(\theta, \gamma, N)$. In this paper, however, we will not concern ourselves with this experimental problem.

2.1 An Unperturbed Experiment

Figure 2 displays the time evolution for a sequence of messages, that arrive and depart the buffer of A, within a certain period of time. Where A_i is the time between the arrival of M_{i-1} and M_i (with the exception that A_1 is from the start of the experiment). We define a busy period (BP) to be the time when the system is busy processing messages.

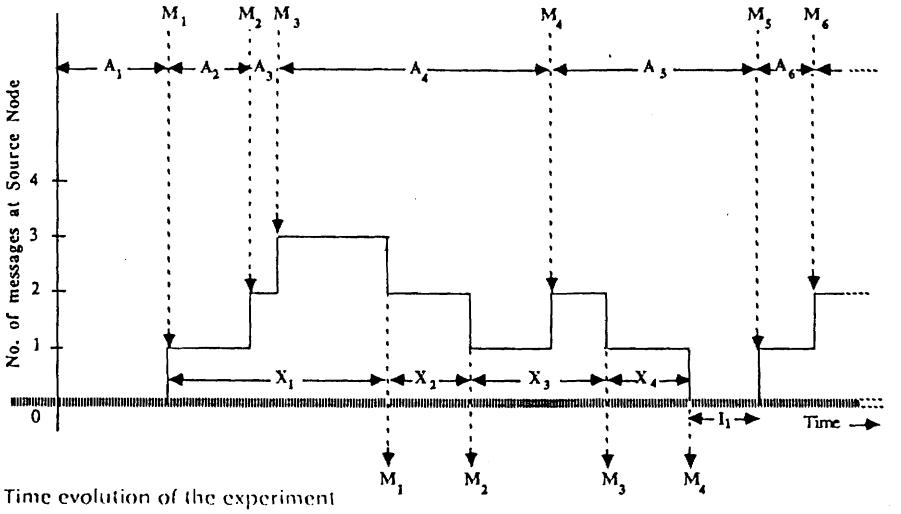


Fig. 2. Time evolution of the experiment

In our example, we start off with the buffer empty, and have to wait a time of length A_1 for the first message to arrive, and another X_1 for the message to be completely transmitted (hence total time is $A_1 + X_1$). However, during this time M_2, followed by M_3 arrive at the queue and have to wait for M_1 to get fully transmitted. In the case of M_2 the arrival time is $A_1 + A_2$ and the departure time is $A_1 + X_1 + X_2$. More generally, M_i has an arrival time of $t_0 + \sum_{j=2}^{j} A_j$ and a departure time of $t_0 + \sum_{j=1}^{i} X_j$, where $t_0 = A_1$. Hence we can define the system time to be

$$(t_0 + \sum_{j=1}^{i} X_j) - (t_0 + \sum_{j=2}^{i} A_i) = \sum_{j=1}^{i} X_j - \sum_{j=2}^{i} A_j. \tag{5}$$

where the sum is zero for the case when $i = 1$. Note that this sum only holds up until the time of the complete departure of the fourth message (i.e. after the first busy period). Therefore, we can rewrite the system time (as would apply to our specific example) in the following way :

$$\sum_{i=1}^{4}\sum_{j=1}^{i} X_j - \sum_{i=1}^{4}\sum_{j=2}^{i} A_j. \tag{6}$$

or more generally, we can define it for the m^{th} busy period as follows :

$$\sum_{i=1}^{n_m}\sum_{j=1}^{i} X_{k_m+j} - \sum_{i=1}^{n_m}\sum_{j=2}^{i} A_{k_m+j}. \tag{7}$$

Hence the average system time of a message can be written as

$$\hat{T}(\theta,\gamma,N) = \left(\frac{1}{N}\right)\sum_{m=1}^{M}\sum_{i=1}^{n_m}\left(\sum_{j=1}^{i} X_{k_m+j} - \sum_{j=2}^{i} A_{k_m+j}\right). \tag{8}$$

2.2 Performing the IPA

We now consider the experiment at hand with the link service time set at $\theta + \Delta\theta$ (the *perturbed experiment*). In this case we will have an increase in the transmission time

$$\begin{aligned} \Delta X_i &= (H + L_i)\Delta\theta \\ &= (\Delta\theta/\theta)X_i \end{aligned} \tag{9}$$

This means that M_1 will take ΔX_1 longer to get fully transmitted, hence M_2 will take $\Delta X_1 + \Delta X_2$, and so on. Hence in the first busy period we have an increase in the system time

$$\begin{aligned} \Delta t_i &= \textstyle\sum_{j=1}^{i} \Delta X_j \\ &= (\Delta\theta/\theta)\textstyle\sum_{j=1}^{i} \Delta X_j \end{aligned} \tag{10}$$

However, when we move to the next busy period we must take into consideration two possibilities. Has the effect of $\Delta\theta$ caused M_4 to get completely transmitted after M_5 arrives? If this is not the case (see Figure 3) then the next busy period can be represented using

equation (10). On the other hand, if this is the case then, returning to our example, we can see from Figure 4 that

$$\Delta t_5 = \Delta S_1 + \Delta X_5. \tag{11}$$

where ΔS_1 is the time where the first busy period has overlapped with the second. Hence, it follows that

$$\Delta t_6 = \Delta S_1 + \Delta X_5 + \Delta X_6. \tag{12}$$

in other words

$$\Delta t_i = \Delta S_1 + (\Delta\theta/\theta)\sum_{j=1}^{i} X_j. \tag{13}$$

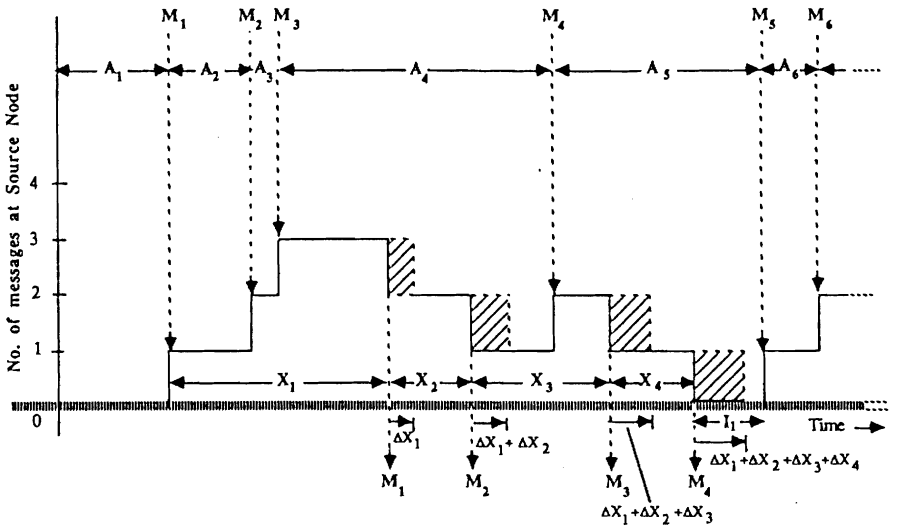


Fig. 3. Perturbations in the sample path for case i).

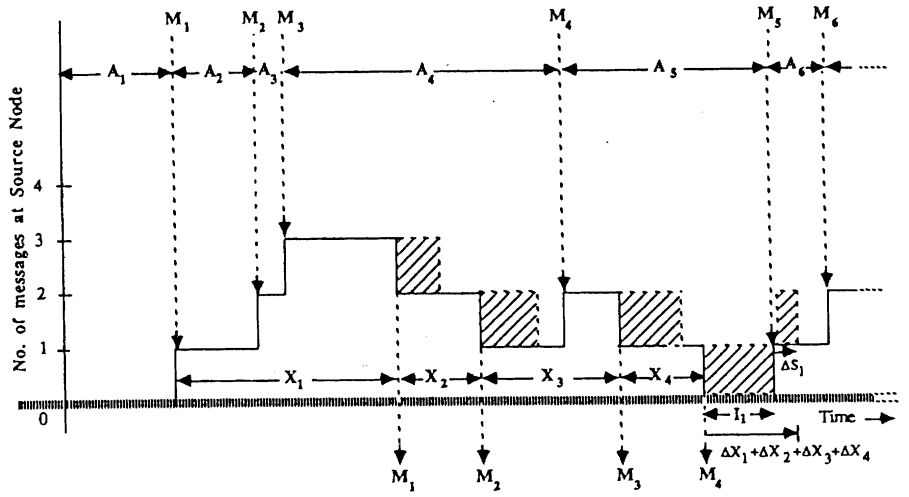


Fig. 4. Perturbations in the sample path for case ii).

We can generalize the equations further so as to represent the m^{th} busy period (let $\Delta S_{m-1}(\Delta\theta)$ be the amount BP_{m-1} overlaps with the arrival of M_{k_m+1}).

$$\Delta t_{k_m+i} = \Delta S_{m-1}(\Delta\theta) + (\Delta\theta/\theta)\sum_{j=1}^{i} X_{k_m+j}, \qquad i \leq n_m. \tag{14}$$

Note that $\Delta S_{m-1}(\Delta\theta)$ includes all effects of the previous busy periods. We are now ready to define the average system time after performing the perturbation:

$$\hat{T}(\theta,\gamma,N) = \left(\frac{1}{N}\right)\sum_{m=1}^{M}\sum_{i=1}^{n_m}[\Delta S_{m-1}(\Delta\theta) + (\Delta\theta/\theta)\sum_{j=1}^{i} X_{k_m+j}]. \tag{15}$$

We are now ready to define the sensitivity of $\hat{T}$ with respect to θ:

$$dT/d\theta = \lim_{\Delta\theta\to 0}\lim_{N\to\infty} \Delta\hat{T}(\theta,\gamma,N)/\Delta\theta. \tag{16}$$

Now we assume that as the number of messages increases the summations of busy periods' overlaps becomes negligible. In other words:

$$\lim_{\Delta\theta\to 0}\lim_{N\to\infty}\left(\frac{1}{N}\right)\sum_{m=1}^{M}\sum_{i=1}^{n_m} \Delta S_{m-1}(\Delta\theta) = 0. \tag{17}$$

Note that we will provide a reason for this assumption later on.

Hence, the correct measure of $dT/d\theta$ is reduced to

$$dT/d\theta = \lim_{N\to\infty}\left(\frac{1}{N}\right)\sum_{m=1}^{M} H_m/\theta. \tag{18}$$

where

$$H_m = \sum_{m=1}^{M}\sum_{i=1}^{n_m} X_{k_m+j}. \tag{19}$$

So finally the gradient estimate can be defined to be

$$\hat{g}_\theta(N) = (\sum_{m=1}^{M} H_m)/(N\theta). \tag{20}$$

We now try to estimate $dT/d\gamma$. We note that the equation

$$X_i = \gamma + L_i\theta. \tag{21}$$

tells us that γ is independent of L_i and θ.Therefore

$$\Delta X_i = \Delta\gamma.. \tag{22}$$

It follows that

$$\begin{aligned} \Delta t_i &= \textstyle\sum_{j=1}^{i} \Delta\gamma \\ &= \Delta\gamma \textstyle\sum_{j=1}^{i} 1 \end{aligned} \tag{23}$$

Hence our estimator is trivially

$$\hat{g}_\gamma(N) = (\sum_{m=1}^{M} \sum_{i=1}^{n_m} \sum_{m=1}^{M} 1)/N. \tag{24}$$

Hence we can implement the following algorithm to calculate both $dT/d\gamma$ and $dT/d\theta$ at the same time.

1. Initialize: Set J, $XSUM$, $JSUM$, $HSUM$, $GSUM = 0$;

   Set $THETA = \theta$;

2. Update: At departure of next message (with service time observed to be XJ);

   1.1) $J + 1$

   1.2a) $XSUM = XSUM + XJ$

   1.2b) $JSUM = JSUM + 1$

   1.3a) $HSUM = HSUM + XSUM$

   1.3b) $GSUM = GSUM + JSUM$

   1.4) If link is now idle then $XSUM = 0$ and $XSUM = 0$

3. Test: If $J = N$ then go to OUTPUT else goto UPDATE ;

4. Output: $dT/d\theta \approx HSUM/(N * THETA)$;

   $dT/d\gamma \approx GSUM/N$;

It was shown that under the assumptions of small perturbation values and in the near-absence of "dramatic" changes in the system's behavior due to the perturbation (i.e. assuming very little overlap between the busy periods, or, in other words, the system has

the property that $\lim_{\Delta\theta\to 0}\lim_{N\to\infty}\left(\frac{1}{N}\right)\sum_{m=1}^{M}\sum_{i=1}^{n_m}\Delta S_{m-1}(\Delta\theta)=0$) that an experimental estimate, which converges to the true value of $dT/d\theta$ as $N\to\infty$, can be easily computed while the nominal (unperturbed) experiment is evolving . It should be noted that this gradient estimate is an infinitismal PA (IPA) estimate, and for "sufficiently small" $\Delta\theta$ the IPA estimate will be equal to the finite difference estimator. In other words we say

$$\hat{g}_\theta(N)=\Delta\hat{T}(\theta,\gamma,N)/\Delta\theta, \qquad \Delta\theta\leq\epsilon. \tag{25}$$

where ϵ is very small.

However, one should notice that the correct definition of the gradient involves letting $N\to\infty$ first and *then* $\Delta\theta\to 0$ for convergence to $dT/d\theta$, but as can be noticed in (25), the order in which we take limits is reversed, for we let $\Delta\theta\to 0$ then let $N\to\infty$. *In order to be able to switch the limits we must make the assumption that the system satisfies :*

$$\lim_{N\to\infty}\lim_{\Delta\theta\to 0}\frac{\Delta\hat{T}(N;\Delta\theta)}{\Delta\theta}=\lim_{\Delta\theta\to 0}\lim_{N\to\infty}\frac{\Delta\hat{T}(N;\Delta\theta)}{\Delta\theta} \tag{26}$$

For it is this assumption that make it feasible to do the estimation for very small $\Delta\theta$ and *then* find the estimator for large N (hence changing the order of taking limits).

Then it follows that

$$\lim_{N\to\infty}\hat{g}_\theta(N)=dT/d\theta. \tag{27}$$

For the class of systems where (26) holds, hence, we can make excellent use of the PA experiment.

3 IPA for a GI/G/1 System

We now consider the PA experiment when applied to a GI/G/1 queue. We start by defining two sets of i.i.d(independent and identically distributed) random variables. First we have the set of r.v.'s

$$\{A_1,A_2,....\}. \tag{28}$$

this represents the sequence of interval times during a given experiment, and

$$\{X_1, X_2, ...\}. \tag{29}$$

represents a sequence of service times. Next, we assume that X_1 is dependent on θ. Finally, we make an assumption that the system is stable, that is $E(X_i) < E(A_i)$. We are interested in the mean service time $T(\theta)$. This - as mentioned earlier - is close to the value of $\hat{T}(\theta, N)$ for large N,or

$$\lim_{N\to\infty} \hat{T}(\theta, N) = T(\theta). \tag{30}$$

To estimate $dT/d\theta$, we first make the assumption that the r.v.'s $X_i(\theta)$ are uniformly differentiable. We make use of this assumption and of (9) and rewrite the equation (14) as

$$\Delta t_{k_m+i} = \Delta S_{m-1}(\Delta\theta) + \sum_{j=1}^{i} \Delta X_{k_m+j}. \tag{31}$$

Also, we have

$$dX_i/d\theta = \lim_{\Delta\theta\to 0} \Delta X_i/\Delta\theta. \tag{32}$$

Hence, as before we try and estimate the sensitivity. We have

$$\begin{aligned} dT/d\theta &= \lim_{N\to\infty} \lim_{\Delta\theta\to 0} \left(\frac{1}{N}\right) \sum_{m=1}^{M} \sum_{m=1}^{M} \sum_{i=1}^{n_m} \Delta X_{k_m+j}/\Delta\theta \\ &= \lim_{N\to\infty} \left(\frac{1}{N}\right) \sum_{m=1}^{M} \sum_{m=1}^{M} \sum_{i=1}^{n_m} dX_{k_m+j}/d\theta \end{aligned} . \tag{33}$$

Thus our IPA estimator is finally

$$\hat{g}(N) = \left(\frac{1}{N}\right) \sum_{m=1}^{M} \sum_{i=1}^{n_m} \sum_{j=1}^{i} dX_{k_m+j}/d\theta. \tag{34}$$

3.1 Sensitivity Analysis for Random Parameters

Earlier in our development, we stated that X_i is dependent on θ. We now need to elaborate more on this matter in order to display some features of the PA experiment. X_i can be dependent on θ in one of two cases. In the first case

$$X_i = (H + L_i)\theta. \tag{35}$$

Therefore

$$\begin{aligned} dX_i/d\theta &= (H + L_i) \\ &= X_i/\theta. \end{aligned} \tag{36}$$

However, there are other systems where

$$X_i = H + L_i + \theta. \tag{37}$$

Then, trivially

$$dX_i/d\theta = 1. \tag{38}$$

What can be observed from the two results above is that $\Delta\theta$ *does not appear on the RHS. This is the whole idea behind the IPA, for it means that we can find the estimate without having to repeat the experiment at* $\Delta\theta$*! Furthermore, in the former result, we need not even concern ourselves with the distribution of the r.v.* X_i*. In the latter, case we don't even need to know* θ*.*

We can now safely make the assumption that $dX_i/d\theta$ can be expressed as $\psi(X_i, \theta)$.

The following is an algorithm for estimating $dX_i/d\theta$:

1. Initialize: Set J, $XSUM$, $HSUM = 0$;
2. Update: At departure of next message (with service time observed to be XJ);
   - 1.1) $J + 1$;
   - 1.2) $XSUM = XSUM + PSI(XJ, THETA)$;
   - 1.3) $HSUM = HSUM + XSUM$;
   - 1.4) If link is now idle then $XSUM = 0$;
3. Test: If $J = N$ then go to OUTPUT else goto UPDATE ;
4. Output: $dT/d\theta \approx HSUM/N$;

3.2 Consistency of IPA

We now want to insure that the assumption that

$$\lim_{N\to\infty} \hat{g}_\theta(N) = dT/d\theta. \tag{39}$$

is solid. But, assuming for the moment that the above assumption is true, we can also make the following inference :

$$\lim_{N\to\infty} E(\hat{g}_\theta(N)) = dT/d\theta. \tag{40}$$

We can prove this fact for an M/M/1 (due to the simplicity of the proof). This system is described by an exponentially distributed arrival times, with rate λ and mean $1/\lambda$, and by an exponentially distributed service times with mean θ. Finally the traffic intensity is defined by $\rho = \lambda\theta$. We are also given

$$\begin{aligned} T(\theta) &= \theta/(1-\rho) \\ E(B) &= \theta/(1-\rho) \\ E(B)^2 &= 2\theta^2/(1-\rho)^3 \end{aligned} \tag{41}$$

where B is a r.v. for the time length of an arbitrary busy period. Differentiating T, we get

$$dT/d\theta = 1/(1-\rho)^2. \tag{42}$$

Also since we can see that θ is a scale parameter of X_i, we have

$$dX_i/d\theta = X_i/\theta. \tag{43}$$

Since we are assuming that the estimate is consistent we can say

$$\begin{aligned} g &= \textstyle\sum_{i=1}^{j} dX_i/d\theta \\ &= (1/\theta)\textstyle\sum_{i=1}^{j} X_i \end{aligned} \tag{44}$$

Looking at $\sum_{i=1}^{j} X_i$ we can see that it is the time from the start of a busy period till the departure of the j^{th} message in this busy period. This summation can be rewritten as the

time from the start of the busy period to the time of the arrival of message j (denoted by z_j), plus the system time of the message. Or,

$$g = (z_j + t_j)/\theta. \tag{45}$$

Now working with the expected value of g (to simplify our proof) we get

$$E(g) = (E(z_j) + E(t_j))/\theta \tag{46}$$

Analyzing the above equation we see that the expected system time was defined by us earlier to be $T(\theta)$. On the other hand, $E(z_j)$ is the expected time for the message to arrive. Hence, one of the following two cases may be the situation. Either the server is idle (denote that by I), or the system is busy (denote that by b). In other words

$$E(z_j) = (E(z_j|I)p_I + E(z_j|b)p_b. \tag{47}$$

But when the system is idle there is no busy period, therefore z_j is zero. Therefore

$$E(z_j) = E(z_j|b)p_b. \tag{48}$$

where p_b is the utilization of the server ρ, and $E(z_j|b)$ is the average time of a busy period seen by a random arrival into the BP (which has been found to be $E(B)^2/2E(B)$). Thus

$$E(z_j) = \rho E(B)^2/2E(B). \tag{49}$$

going back to $E(g)$, we now have

$$E(g) = (\rho E(B)^2/2E(B) + T(\theta))/\theta. \tag{50}$$

Substituting the values the we are given in (41) we get

$$\begin{aligned} E(g) &= (\rho\theta/(1-\rho)^2 + \theta/(1-\rho))/\theta \\ &= 1/(1-\rho)^2. \end{aligned} \tag{51}$$

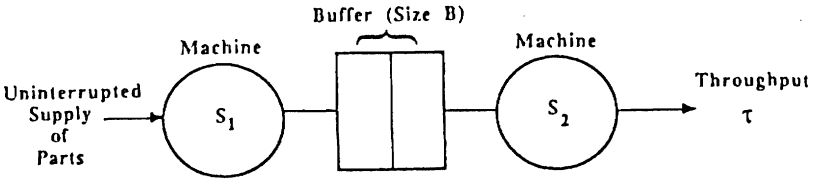
thus proving the assumption made in (39).

4 IPA for General Networks

In the previous section, the main ideas of infinitismal perturbation analysis were illustrated using a single server queue model of a communication link. To make use of IPA in realistic situations, we have to look at IPA for more general systems. We are going to address the problem of finding IPA algorithms for the case of a simple production line with just two machines and then for a general network of servers.

4.1 IPA for a Simple Production Line

IPA can be performed for a simple production line consisting of two servers (machines) and a buffer in between as shown in the figure. The production line can be thought of as a system consisting of two computers and one buffer.



A simple production line.

Server 1 (S_1) is a machine whose cycle time depends on a parameter θ_1. We can assume that S_1 has an uninterrupted supply of parts to work on. After S_1 finishes its work cycle on a part, it places the part in the buffer. The second machine S_2 picks one part from the buffer, works on it for a cycle time (which depends on a parameter θ_2) and then releases it to a finished goods area. The size of the buffer is B. If the buffer is full when S_1 completes a part then the part stays at S_1, which is then unable to work on another part and is said to be *blocked*. S_1 remains blocked until S_2 finishes its current cycle, releases its part, and

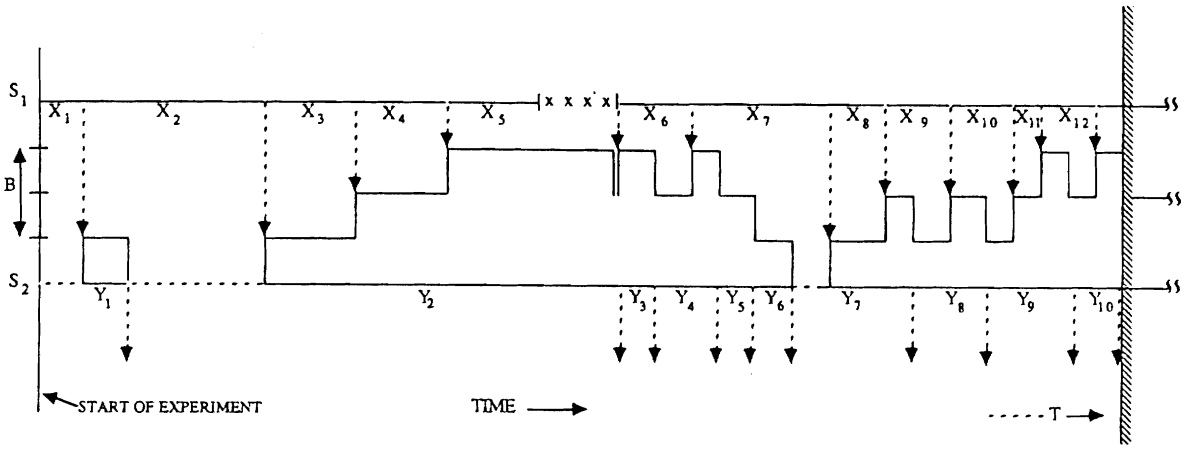
takes the next part from the buffer, thereby releasing a buffer space. We shall assume that all transfers take place in a negligible amount of time, and that the finished goods area is never blocked. The performance measure we shall consider of interest for this system is its steady state throughput (number of parts produced per unit time) which we shall denote $\tau(\theta_1, \theta_2)$. We can define an experiment on this system, starting with no parts in S_1, S_2, or the buffer, and ending when the Nth part is completed by S_2. If T is the length of time for this experiment, then the experimental estimate of the throughput is

$$\hat{\tau}(\theta_1, \theta_2, N) = N/T \tag{52}$$

Under some conditions, this estimate will satisfy

$$\lim_{N \to \infty} \hat{\tau}(\theta_1, \theta_2, N) = \tau(\theta_1, \theta_2) \tag{53}$$

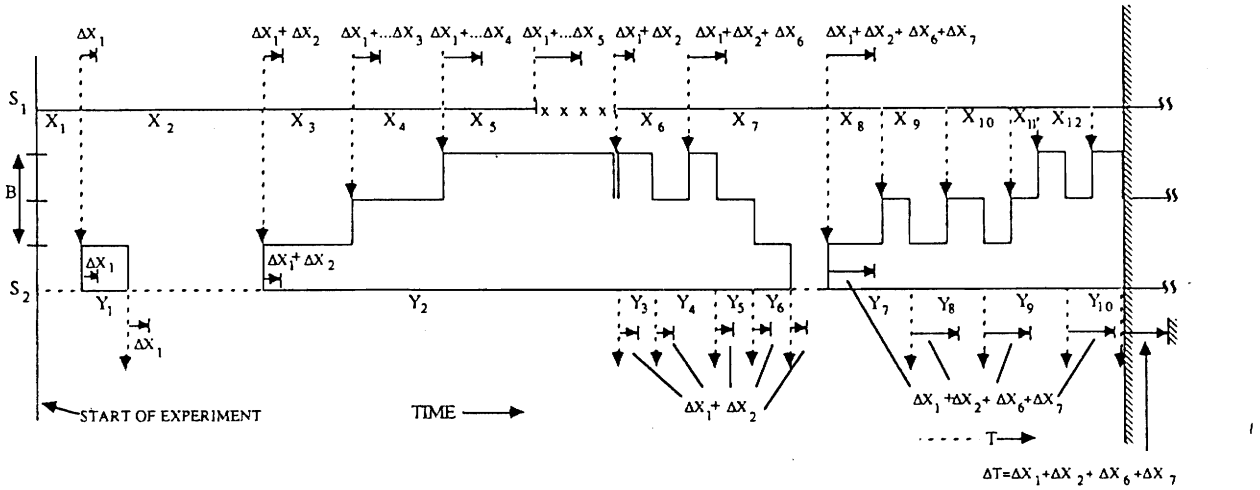
which is desired for a good experimental estimate.



Nominal sample path for the production line.

A typical sample path is shown in the figure with $N = 10$, X_i and Y_i denotes the cycle time for S_1 and S_2 for the ith part. The vertical axis represents the number of parts at S_2 and at the buffer. The size of the buffer B is 2 for this example, part i is denoted by P_i and dashed lines implies that S_2 is idle, crosses implies that S_1 is blocked. Our

goal then is to develop an IPA algorithm to estimate $d\tau/d\theta$ for this system. Introducing a perturbation $\Delta\theta$ in this system, the perturbed sample path is shown in the figure . Where $\Delta X_i(X_i(\theta_1 + \Delta\theta_1) - X_i(\theta_1))$ denotes the change in cycle times at S_1 due to a change $\Delta\theta_1$ in the parameter θ_1. It should be clear that there is an implicit assumption for the perturbed path shown in the figure, namely that the perturbations are small enough so that the order of events does not change, such assumption is standard in IPA.



Perturbations in the sample path for the production line.

With the above assumption, stating the IPA algorithm becomes particularly simple. Letting AC_1 and AC_2 be accumulators associated with S_1 and S_2, AC_j is the perturbation at S_j for the last part that left S_j, and the arrows ($\rightarrow$ |) shows the values of the accumulators. Then we can develop three rules, the first is that whenever a part P_i has been served at S_1 the first accumulator is incremented by ΔX_i, the second is that if P_i finds S_2 idle, then AC_2 gets the value of AC_1 and finally if P_k unblocks S_1 by departing from S_2 then AC_1 gets the value of AC_2. We can then proceed to write the algorithm for calculating the gradient of θ_1.

At the end of the experiment, $\Delta T = AC_2$, and as shown above AC_2 is the sum of some of the ΔX_i values, say for $i \in l$. Under the assumption that the random values $X_i(\theta)$ have the property that $dX_i/d\theta$ can be expressed as $\psi(X_i, \theta)$, we can say that

$$\frac{dT}{d\theta_1} = \lim_{\Delta\theta_1 \to 0} \frac{AC_2}{\Delta\theta_1} = \sum_{i \in l} \frac{dX_i}{d\theta_1} = \sum_{i \in l} \psi(X_i, \theta_1) \tag{54}$$

and since N is fixed by definition of the experiment, then

$$\frac{d\hat{\tau}}{d\theta_1} = -(N/T^2)\frac{dT}{d\theta_1} = -(N/T^2)\sum_{i \in l} \psi(X_i, \theta_1) \tag{55}$$

Which implies that if we accumulate $\psi(X_i, \theta)$ instead of ΔX_i, in the first rule above, and call the accumulators A_1 and A_2, then after the experiment is performed, the value $-(N/T^2)A_2$ will be the IPA estimate of $d\hat{\tau}/d\theta_1$. The algorithm is then developed as follows :

1. Initialize: Set A_1, $A_2 = 0$;

   Set $THETA1 = \theta_1$;

2. Update: Whenever a part (say P_i) completes service, check these conditions :

   1) If P_i completed service at S_i then

      $A_1 \leftarrow A_1 + PSI(X_i, THETA1)$;

   2) If P_i leaves S_1 and terminates an idle

      period of S_2 then $A_2 \leftarrow A_1$;

   3) If P_i leaves S_2 and terminates a blocked

      period of S_1 then $A_1 \leftarrow A_2$;

3. Test: If S_2 has completed N parts go to OUTPUT

   else goto UPDATE ;

4. Output: Let T be the total time since the start of the experiment;

   The IPA estimate of $d\tau/d\theta$ is $-(N/T^2)A_2$.

4.2 IPA for General Networks with Finite Buffers

Considering a general network with finite buffers, having a single server at each station, we can generalize the algorithm described above easily to allow for more than two servers.

It should be noted that the only times when perturbations propagate from one server to another are when idle or blocked intervals are terminated by a customer moving from one server to another. Thus the propagation rules 2 and 3 in the above algorithm can be modified by allowing for any servers S_i and S_k instead of S_1 and S_2 and naming the associated accumulators A_i and A_k and thus replacing $A_2 \leftarrow A_1$ by $A_k \leftarrow A_i$. In general network it is possible to have a situation of "chain" blocking, where, for example, S_k is blocked by S_i, and then in turn the buffer at S_K gets full and it ends up blocking S_j. In this case we just need to implement the propagation for each unblocked server in turn, but there is no change in the rule. A further generalization would be to change the first condition statement in the 2-server algorithm to allow the use of the accumulators associated with different servers. It is also possible to state the algorithm in such a way so that it can compute all K gradients at the same time as follows : (A_{ij} is the accumulator at S_i for gradient with respect to θ_j)

1. Initialize: Set A_{ij}, $i = 1, ..., K$; $j = 1, ..., K$;

   Set $THETA_i = \theta_i$, $i = 1, ..., K$;

2. Update: Whenever a customer (say C) completes service, check these conditions :

   1) If C completed service at S_i then

   $A_{ii} \leftarrow A_{ii} + PSI(i, X, THETA_i)$;

   2) If C leaves S_1 and terminates an idle

   period of S_m then $A_{mj} \leftarrow A_{ij}$,

   for $j = 1, ..., K$; (If there is a chain of blocking

   then continue this procedure through the chain)

3. Test: If S_{end} has completed N parts go to OUTPUT

   else goto UPDATE ;

4. Output: Let T be the total time since the start of the experiment;

   The IPA estimates of the K gradients $d\tau/d\theta_j$

   $(j = 1, ..., K)$ are $-(N/T^2)A_{endj}$ $(j = 1, ..., K)$.

5 Extensions of IPA

In some cases, the IPA technique discussed above will fail to work. One instance might be due to the assumption that small changes in the system parameter θ will not cause coalescing of busy periods in a GI/G/1 queue because of small $\Delta\theta$. Suppose that the performance measure of interest is the average number of messages sent between idle periods of a communication link. If we model the link as a single server queue, this performance measure is the average number of customers served in a busy period (BP). Denoting this average by $\beta(\theta)$, then a simple experimental estimate for $\beta(\theta)$ would be to observe M BPs and then let

$$\hat{\beta}(\theta, M) = \left(\frac{1}{M}\right) \sum_{m=1}^{M} n_m \tag{56}$$

where n_m is the number of customers served in BP_m. Considering the arguments presented in the IPA, we can see that IPA is based entirely on the assumption that no BPs will coalesce. If we make $\Delta\theta$ small enough so that no BPs coalesce, then each n_m value will remain the same, so that there will be no change in the estimate of the performance measure. Thus, the IPA estimate of sensitivity will be zero ! It is clear that this is wrong and thus IPA failed in this example. IPA ignores the effects of some events in the system, when the probability of occurrence of these events, multiplied by the effect of the events on the performance is significant, IPA fails. This motivates some extensions which enable gradient estimation for a wider class of systems.

5.1 Smoothed Perturbation Analysis (SPA)

Motivated by the failure of IPA to work for the simple case above, the idea of using conditional probabilities was introduced to develop an extension for the IPA. A conditioning variable can be used to decompose the gradient estimate expectation expression. The fact that more information is used in developing the conditional probability counts for the "smoother"

kind of performance measure estimate curve that is obtainable by using this method. For example, we can ask the question, for a given $\Delta\theta$, what is the *expected* change in the value of n_i, based on the observed BP_i.

5.2 Extended Perturbation Analysis (EPA)

For systems that can be represented by markov chains, a new approach that may overcome the potential inconsistency of IPA can be applied. The idea behind the extended perturbation analysis is the fact that the perturbed and unperturbed systems should be statistically evolving similarly once they enter a common state x, due to their markovian property. This method works by choosing a finite $\Delta\theta$ and predicting, from the nominal path, where the perturbed path would have branched to a different state, say y, while the nominal path continues in, say, state x. Up to this point, an IPA-like estimator is used to compute the effects of perturbation, but at this point, the computation is "frozen". The algorithm then waits for the system to enter state y during the nominal path, then EPA restarts. When an event order change occurs, the state sequences of the nominal path (NP) and the perturbed path (PP) may or may not start to differ depending on whether some discontinuous change is involved (e.g., a job originally going to server A may now go to server B). As shown in the figure below, if ω_1 and ω_2 are two state sequences of a Markov DEDS and the state sequence jumps on from S_s on ω_1 to S_p on ω_2 instead of S_h on ω_1, subsequent perturbations involving state changes may cause further deviations so that a perturbed path could be made up from segments of state sequences from ω_1, ω_2,...,ω_j,...

We can see right way that EPA cannot be as efficient as IPA, since it may remain "inactive" for significant sections of the nominal experiment. However, there are two factors that make its performance better than one might expect. The first is that in most applications we do the gradient estimation with respect to a number of parameters simultaneously, it will probably turn out that several of the gradient computations are "active", on average, during

the observations and the savings is still better compared to multiple experimentation. The second is that from a practical point of view, one can often aggregate the states of the system to fewer subsets, and use the aggregate state to decide whether to activate or deactivate the EPA calculation. Not only does this keep the computation active for longer segments of the experiment, but it also enables EPA to be applied to non-Markovian systems.

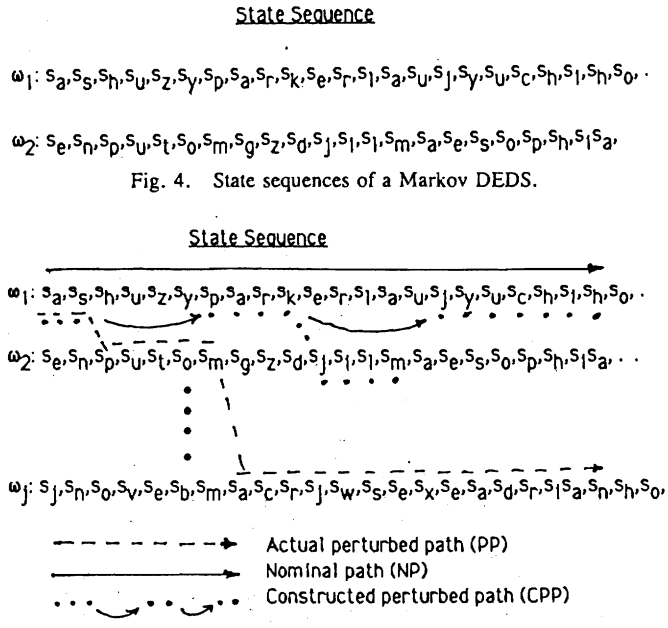


Fig. 4. State sequences of a Markov DEDS.

5.3 Other Perturbation Techniques

Another Perturbation technique is finite perturbation analysis (FPA), this technique was introduced to overcome the IPA assumption that events do not change order. However, FPA considers changes in order of events to a pre-specified limit, for example, it may consider only "first order" changes, that is, changes in the order of adjacent events, and ignores any effects of changes in order beyond adjacent events. The way it works then is to introduce perturbations and propagate them while observing the nominal path, but limiting its calculations by only extrapolating to predict the effect of such changes in order. Originally FPA was heuristic and experimental in nature, however, recent research has been performed to provide more theoretical foundations for it.

Other techniques to make IPA work include changing the system parameter under consideration to transform problems into "easier" versions, or to versions that have already been solved. Using a different representation for the system sometime helps in performing IPA.

6 Research Issues and Future Work

Many problems regarding discrete event dynamic systems in general, and perturbation analysis as an evaluation technique remains open. For example, performing PA for a discrete parameter θ is one such interesting problem. In practical systems, many parameters (such as buffer sizes, or number of servers at a station) are discrete in nature. It should be noticed that IPA, by its nature can be applied only to continuous parameters. Understanding and expanding the domain of IPA needs to be addressed, in fact, to "automate" the process of generating algorithms to calculate the sensitivity of a performance measure remains an open problem. To be able to construct a preprocessing stage, where its inputs are the system specification and the performance measure and parameters of interest, and the output as an IPA algorithm to be run while the nominal experiment is performed, is one challenging problem for researchers. More work still remains to be done on developing efficiency and accuracy measures for the PA output. Trying to get the "maximum" amount of information from a sample path is another long-term goal.

References

[1] Xi-Ren Cao, "The Predictability of Discrete Event Systems", *Proceedings of the* 27^{th} *Conference on Decision and Control*, December 1988.

[2] Y. Ho, "Performance Evaluation and Perturbation Analysis of Discrete Event Dynamic Systems", *IEEE Transactions on Automatic Control*, July 1987.

[3] J. E. Hopcroft and J. D. Ullman, *Introduction to Automata Theory, Languages and Computation*, Addison-Wesley, 1979.

[4] Zvi Kohavi, *Switching and Finite Automata Theory*, McGraw-Hill, 1979.

[5] H. R. Lewis and C. H. Papadimitriou, *Elements of the Theory of Computation*, Prentice-Hall, 1981.

[6] Yong Li and W. M. Wonham, "Controllability and Observability in the State-Feedback Control of Discrete-Event Systems", *Proc. 27^{th} Conf. on Decision and Control*. 1988.

[7] C. M. Özveren, *Analysis and Control of Discrete Event Dynamic Systems : A State Space Approach*, Ph.D. Thesis, Massachusetts Institute of Technology, August 1989.

[8] Rajan Suri, "Perturbation Analysis : The State of the Art and Research Issues Explained via the GI/G/1 Queue", *Proc. of the IEEE*, January 1989.

[9] P. J. Ramadge and W. M. Wonham, "Modular Feedback Logic for Discrete Event Systems", *SIAM Journal of Control and Optimization*, September 1987.

[10] P. J. Ramadge and W. M. Wonham, "Supervisory Control of a Class of Discrete Event Processes", *SIAM Journal of Control and Optimization*, January 1987.

[11] G. E. Révész, *Introduction to Formal Languages*, McGraw-Hill, 1985.