Proceedings of the 1992 Winter Simulation Conference
ed. J. J. Swain, D. Goldsman, R. C. Crain, and J. R. Wilson

PERTURBATION ANALYSIS : CONCEPTS AND ALGORITHMS

YU-CHI HO

Division of Applied Sciences
Harvard University
Cambridge, MA 02138 USA

ABSTRACT.

The subject of Perturbation Analysis (PA) is over ten years old. A substantial literature has been accumulated. It is no longer possible to cover every aspect of this subject and for one person to know everything. This paper nevertheless attempts to present a self-contained tutorial on the state of the art as of the summer of 1992. No inference should be drawn on topics and papers not covered here.

1.INTRODUCTION.

The main tenet of Perturbation Analysis (PA) is that a great deal of information is contained in the sample paths of a Discrete Event Dynamic System (DEDS) beyond the usual statistics collected such as the means and variances of various output variables. Instead of looking at discrete event simulation simply as a special case of statistical analysis of experiments, namely a black box with input parameters and final output results, we can take advantage of our knowledge about the dynamics of the DEDS and squeeze out additional useful information, such as performance gradient and sensitivities from a single experiment. In a broader sense, PA has to do with the problem of doing simulation more efficiently than before; or doing more performance evaluation for the same computing budget. It has also been the thesis of this author that DEDS share many conceptual commonalities with continuous variable dynamic systems (CVDS) governed by differential equations. Many of the successes in the optimization and control of differential equation based dynamic systems can be transplanted to the DEDS domain. In fact, the analog to PA in CVDS is simply the familiar idea of linearization and variational differential equations.

This tutorial is composed of six sections. After introducing notations and terminology, we discuss the basics of Infinitesimal PA and its application to single run gradient estimation in section 2. Section 3 is concerned with the extension of IPA via the general notion of "smoothing". In particular we outline some new results in this area . Section 4 introduces the notions of cut-&-paste and finite PA. Here we view simulation simply as a way of assembling (mapping) a set of random numbers and parameter values. Performance evaluation under different parameters simply means different mappings of the same set of random samples. Such a viewpoint naturally leads to the discussion of the new standard clock approach to simulation and the efficiency that can be achieved using a massively parallel SIMD computer. This is discussed in Section 5. Sections 6 and 7 conclude with some new (1991-92) developments in stochastic optimization and a listing of resources for further study.

1.1 Notations:

Let us introduce some notations first. X is a discrete (possibly infinite) set of **states,** $x \in X$, Γ another discrete finite set of **events**, $\alpha \in \Gamma$, and $\Gamma(x)$, a subset of Γ for each state x representing the **set of feasible (or enabled) events** that can occur in the state x. There are **lifetimes or clock readings** associated with each feasible event as specified below. The clock readings tick down until one of them reaches zero. The associated event is called the **triggering event** in that state. Given the current state and the triggering event, a **state transition function**, $x_{next}=f(x_{now}$, triggering event) , instantaneously takes the DEDS to the next state. (Note: randomness in the state transition can be easily incorporated. We omitted it for simplicity without loss of generality.) The cycle then repeats successively generating a **trace** of the (state, event) pair of sequences. Initially, with a starting state, x_1, we endow every enabled event "α" in x_1 with a lifetime $c_\alpha(1)$, $\alpha \in \Gamma(x_1)$. The smallest $c_\alpha(1)$ determines the **triggering event**, α^*. We then advance time to the triggering instant $t^* \equiv \tau_1 = \tau_0 + c_{\alpha^*}(1)$. For successive states x and for every new "α" enabled in these x we endow it with a lifetime $c_\alpha(n)$ if it is the nth occurrence of event type α; for every old "α" left over from the previous state, we use the remaining lifetime or clock reading as its new lifetime in the new state. (We have adopted the so-called non-interrupt version of the model. If interrupts occur, the model can be easily modified). This way every event enabled (or scheduled) sooner or later becomes a triggering event as its clock reading ticks down and occurs at time $\tau_n(\alpha)$. The totality of the lifetimes $c_\alpha(n)$ for all n and all α defines a two dimensional **clock mechanism** which is completely independent from the state transition function and can be constructed beforehand. The state-event sequence or trace together with the time of occurrence of each and every (triggering) event , τ_n, constitutes a **trajectory** of the DEDS. To help

fix ideas, consider a simple queue-server with random arrival and service times. X is simply the set {0,1,2,3, . . . } representing the possible number of customers waiting in the queue and being served; Γ ={ arrival, departure}; $\Gamma(x=0)$={arrival}, $\Gamma(x\neq 0)$={ arrival, departure}. The clock mechanism consists of two streams of events with random inter-arrival and inter-service times which are generated using given distributional information. The state transition functions are very simple

$$x_{next} = \begin{cases} n+1 & \text{if } x_{now}=n \text{ and } \alpha^*=\text{arrival} \\ n-1 & \text{if } x_{now}\neq 0 \text{ and } \alpha^*=\text{departure} \end{cases}$$

The process starts with say, the arrival of a job which transitions the state to $n\neq 0$. The subsequent colloquial description of the operations of such a FCFS queue-server facility then easily determines the successive interarrival and service time samples to be drawn from the clock mechanism of two event streams. The trace or the trajectory can be determined. Mathematically, we often denote a trajectory simply as (θ, ξ) where θ are the parameters characterizing the state transition function and/or the clock mechanism and ξ all the random occurrences in the DEDS. In the present example, θ obviously is the parameters characterizing the arrival and service distributions and ξ the lifetimes in the clock mechanisms, i.e., the various samples of inter-arrival and service times. Given a trajectory (θ, ξ) we can evaluate its **sample performance** $L(\theta, \xi)$; and its **average performance** $J(\theta) \equiv E[L(\theta, \xi)]$ via a statistical experiment, i.e., a discrete event simulation. In perturbation analysis, we are often interested in not only $J(\theta)$ but also $J(\theta+\Delta\theta)$. We use the adjectives **nominal** and **perturbed** to qualify $J(\theta)$ and $J(\theta+\Delta\theta)$ as well as the trajectories (θ, ξ) and $(\theta+\Delta\theta, \xi)$.

The above description can be considered as a mathematical specification of a discrete event simulation experiment or formally as the Generalized Semi-Markov Process (GSMP) characterization of DEDS. For more mathematical details see Glasserman (1990) pp 27-31, Ho and Cao (1991 § 3.3)

2. WHAT IS INFINITESIMAL PERTURBATION ANALYSIS (IPA)?

2.1 A Short history and the Basic Idea

PA was a problem-driven innovation. In 1977, the author was presented with an interesting consulting problem [Ho, Eyler, and Chien 1979]. (Note M. Bello in a 1977 M.I.T. Masters degree thesis also had a version of the idea of perturbation analysis as applied to an M/D/1 queue, see Bello (1977) and also Woodside (1984)). The FIAT Motor company in Torino, Italy had installed a production monitor system on one of their automobile engine production lines which could be visualized as a simple serial queueing network with finite queue (buffer) capacity between servers (machines). The automatic line monitoring system recorded service initiations, completions, idlings and blockings of various machining stations as well as the movement of the engine parts among them, in short, a complete operating history of the DEDS. A tremendous amount of production information was being generated. The following questions were asked: "Besides the standard statistical information such as downtime, throughput, and utilization that were being generated by the monitoring system from the collected information, could this information be used further for control purposes? In particular, we (FIAT) were interested in whether or not the buffer spaces between machines are optimally distributed for maximal throughput given a limited budget for buffer spaces." The attempt to answer this question (Ho and Cao 1991 p.20) led to the following three generalizable ideas:

(i) A parameter change (e.g., increasing the size of the buffer space by 1) can **generate** perturbations in the timing of events in the sample path of a DEDS.

(ii) Perturbations in the timing of one event (e.g., termination of a service period of a machine) can be **propagated** to another event (e.g., via the termination of an idling period at a downstream machine).

(iii) Since all performance measures of a DEDS depend on the timing of events on its sample path, perturbations in the timing of events will induce perturbations in the sample performance measure, L.

Steps (i) - (iii) suggest a method to calculate efficiently the perturbed performance $L(\theta+\Delta\theta, \xi)$ or the derivative $dL/d\theta$ of a sample path from $x(t; \theta, \xi)$ alone since all three steps only require information directly observable on $x(t; \theta, \xi)$.

2.2 The Interchangeability Issue and the Monotonicity Condition

IPA in a narrow sense is thus a technique for the efficient computation of the n-dimensional gradient vector of performance measure, $J(\theta)$, of a discrete event dynamic system with respect to its parameters (θ) using only one statistical experiment of the system. This is opposed to the traditional method of making n additional experiments and taking differences to approximate the gradient vector, i.e.,

$$\frac{dJ(\theta)}{d\theta} \approx \frac{E[L(\theta+\Delta\theta,\xi)] - E[L(\theta,\xi)]}{\Delta\theta} = \frac{\frac{1}{N}\sum_{i=1}^{N} E[L(\theta+\Delta\theta,\xi_i)] - \frac{1}{N}\sum_{i=1}^{N} L(\theta,\xi_i)}{\Delta\theta}$$

which is a numerically difficult task since we are dividing the difference of two nearly equal random quantities by a small number and are caught by the twin evils of noise and nonlinearity. Instead, IPA proposes to calculate directly the sample derivative $dL(\theta,\xi)/d\theta$ using information on the nominal trajectory (θ,ξ) alone. The basic idea is this: If the perturbations introduced into the trajectory (θ,ξ) are sufficiently small, then we can assume that the event string or sequence of the perturbed trajectory $(\theta+d\theta,\xi)$ remains unchanged from the nominal, i.e., the two trajectories are **deterministically similar** in the order of their event sequences. In this case, the derivative $dL(\theta,\xi)/d\theta$ can be calculated easily. Essentially, once **generated,** perturbations in the timing of events are **propagated** via the same event scheduler (critical timing path) of the nominal simulation. The computational steps are extremely simple and require minimal modification of the simulation code. However, averaging over the sample $dL(\theta,\xi)/d\theta$ we get

$$\frac{1}{N}\sum_{i=1}^{N}\frac{dL(\theta,\xi_i)}{d\theta} \approx E\left[\frac{dL(\theta,\xi)}{d\theta}\right] = ? = \frac{dE[L(\theta,\xi)]}{d\theta} \qquad (1)$$

which raises the $64,000 question above since we are interested only in the right side of equation (1) but PA calcuates the lefthand side. In nontechnical terms, this question translates to "How can you squeeze out information about a trajectory / sample-path operating under one value of the system parameter, θ, from that of another operating under a different value, $\theta'=\theta+\Delta\theta$? Don't the two trajectories behave entirely dissimilarly sooner or later?" The intuitive characterization of the condition under which the above question can be answered in the affirmative (or IPA will give unbiased estimate to the derivative of the expected performance) is as follows:

> While the nominal and the perturbed trajectories $(\theta+\Delta\theta,\xi)$ and (θ,ξ) must sooner or later differ no matter how small we make $\Delta\theta$, or equivalently, the ensemble of trajectories $(\theta+\Delta\theta,\xi)$ and (θ,ξ) will differ on some member of the ensemble if the ensemble is large enough regardless of the size of $\Delta\theta$ if we have a finite time problem, the frequency of occurrence of such difference is of order $\Delta\theta$. Furthermore, if the trajectory difference is also small and of order $\Delta\theta$, then the average net effect of the difference in the nominal and the perturbed performance will be of order $\Delta\theta^2$ which is neglegible for the first derivative calculation. In such cases, Eq.(1) will hold [Cao 1985].

In other words, if the deviations caused by $\Delta\theta$ between the nominal and the perturbed trajectory are small and only temporary, then IPA will give an unbiased estimate for the quantity $\partial J/\partial\theta$. Glasserman (1991 ch.3), Glasserman and Yao (1991) gave a precise characterization of this as the **Commuting (Monotonicity) condition** which requires:

(i) an event once scheduled is never terminated prematurely (the non-interrupt condition)

(ii) if the nominal and the perturbed event strings differ only in order and not in the total number of each event types, then the system from which these event strings are generated must be in the same state (or states that have the same enabled event list) . This is also known as permutability condition

Condition (i) prevents finite discontinuities from occurring frequently and condition (ii) insures that event order permutation creates only temporary divergence between the two trajectories. A simple example for which the C (M) condition holds is a G/G/1 queue under service and arrival time perturbation. Fig.1 illustrates the situation where due to perturbation a departure and arrival event change order.

The satisfaction of condition C (M) can be seen by inspection. (For precision, we note there are technical differences associated with the C vs. the M condition, but these should not concern the tutorial nature of the current discussion)

The classes of DEDS over which a simple IPA algorithm is known to apply is fairly well understood by now. These include,

A. Simple queueing networks: A simple network consists entirely of FCFS, infinite buffer, single-server nodes and a single class of customers with a state independent Markovian routing mechanism. A GI/G/1 queue and networks with general service and arrival distributions are examples of simple queueing networks.

B. Simple networks with multi-class customers in which every node that is visited by more than one class of customers is fed by only a single source. In such a network, a customer cannot change the order of arriving at a server with any other customer of a different class.

C. Networks with blocking in which every node with a finite buffer is fed by a single source. In such a network, no server can directly block two or more servers simultaneously. A cyclic queue with finite buffers is an example of such network.

D. Some networks with state-dependent routing mechanism described as follows: For any (arrival or departure) event α, let $\{X_1(\alpha), ..., X_N(\alpha)\}$ be the partition of the states such that x and x' are in the same subset $X_i(\alpha)$ if and only if the transition probabilities for the customer moving upon the occurrence of α are the same in x and x'. Then the interchangeability holds if that event α is the only event that can trigger a state transition between two states that belong to two different subsets $X_i(\alpha)$ and $X_j(\alpha)$, $i \neq j$.

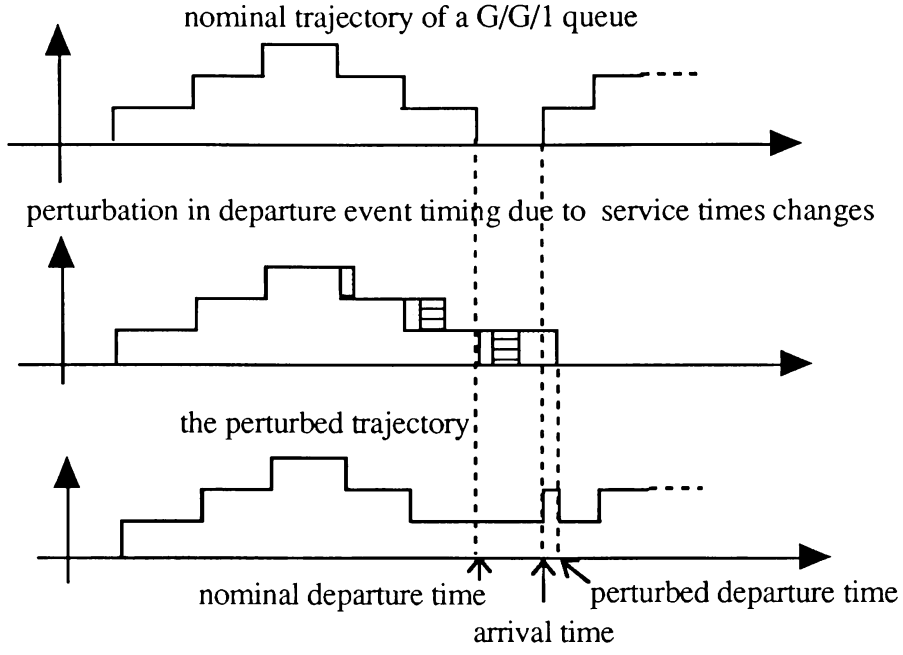

Figure 1 Illustration of the Commuting (Monotonicity) condition

The main virtue of simple IPA is its extreme computational simplicity . Only minimal changes in a simulation program need to be made to enable it (see Ho and Cao (1991 Appendix E and Chapter 3). Multidimensionality of θ adds very little to the computational burden. Numerical stability is also a virtue since no divisions by $\Delta\theta$ are involved. Under certain assumptions one can also prove that it is the minimum variance estimate since it uses common random numbers. Experimentally, it demonstrates excellent variance properties.

Lastly, two additional issues that we have not disccussed in this tutorial due to space limitations are consistency of IPA estimates (see e.g., Glasserman and Hu and Strickland (1991) ,Wardi and Hu (1991)) and realtime nonintrusive application of IPA to real systems (e.g., see Cassandra and Abidi and Towsley 1990), and Ho and Cao (1991 pp285-286))

3. EXTENSION OF IPA.

Of course it is easy to make up examples for which simple IPA rules will lead to biased estimates of the performance gradient, i.e., failure of Eq.(1). Primary examples of such parameter sensitivity problems are routing probabilities and multi-class queues where **finite** outcome or event order perturbations can result from **infinitesimal** parameter changes. Other examples of discontinuities are performance measures that are inherently discrete, such as the number of customers served in a busy period. In all such cases, direct computation of $dL(\theta,\xi)/d\theta$ often yields the value of zero. But for any ξ, there is a value of $\Delta\theta$ that will cause a finite discontinuity in L. And for any $\Delta\theta$, there exist ξ w.p.1 that will produce a finite discontinuity in L. The average of these discontinuous $L(\theta,\xi)$ summed over ξ nevertheless defines a smooth $J(\theta)$ that has a well behaved nonzero slope [see Ho-Cao 1991 p.80]. Such problems are said to be non IPA-applicable since we'll get biased estimates of the term $dE[L]/d\theta$ using the simple IPA algorithms. Extensions of IPA to overcome this difficulty proceed in two major directions. First and most popular, we have what might be called the **probability based extensions** which is based on the idea of "**smoothing**":

> **Infrequent occurrences of finite perturbations are statistically equivalent to frequent occurrences of infinitesimal perturbations. (S)**

A specific example of this idea which we shall use in many guises is the fact that a stream of Poisson events with rate λ can be modified to represent another Poisson stream with rate $\lambda-\Delta\lambda$ in the following two equivalent ways:

(i) stretch the time axis of the Poisson stream by the factor $\lambda/\lambda-\Delta\lambda$ (this corresponds to perturbing every event timing by an infinitesimal amount)

(ii) delete each event w.p. $\Delta\lambda/\lambda$ (this corresponds to occasionally perturbing the inter-arrival times by a finite amount).

The simplest example of such a device is the routing probability parameter sensitivity. Consider the case of routing customers to one of two possible servers with different mean service times s_1 or s_2 according to the proportion d:1-d as follows:

$$s = \begin{cases} -s_1 \ln(v) & 0 \le u \le d, \\ -s_2 \ln(v) & d < u \le 1, \end{cases} \qquad (2)$$

where u and v are both uniform random variables over [0, 1), and "d" plays the role of routing probability. Notice that under the condition that $0 \le u \le d$ (resp. $d < u \le 1$), u/d (resp. (1-u)/(1-d)) is also a uniform random variable over [0, 1). So instead of using (2) to determine service time we can use

$$s = \begin{cases} -s_1 \ln\left(\frac{u}{d}\right) & 0 \le u \le d, \\ -s_2 \ln\left(\frac{1-u}{1-d}\right) & d < u \le 1. \end{cases} \tag{3}$$

It can be observed that if we generate a service time from Eq. (2), a finite discontinuous change can occur in s when "d" is infinitesimally perturbed for "u" nearly equal to "d". On the other hand if we use eq.(3) instead, the service time of *every* customer changes when "d" is changed. The important advantage of using (3) is that the service time changes of customers who switch between servers because of the change of "d" are no longer finite. These changes are now infinitesimal since switching takes place only at values of "u" (not "v") near "d", i.e., when s is near ln(1) = 0. It is clear that the service time changes are now of the order Δd. Thus, when switching takes place, the discontinuities will be of the same order. On the other hand, the probability that a customer in the nominal path is to be switched is also in the order of Δd. Therefore, the expected change in performance, caused by switched customers, is of order $(\Delta d)^2$ and can be ignored in calculating the first derivative. This is equivalent to saying that event order changes are ignorable and IPA will give us unbiased estimates for the sensitivities with respect to routing probability "d." Another version of this (S)-extension is to convert the routing probability perturbation into an equivalent arrival rate perturbation to the two servers (Ho and Cao 1991 ch.5). It is worth emphasizing that the basic notion of (S), *converting finite but infrequent perturbations into frequent but infinitesimal perturbations or vice versa*, is present in both of the above cases.

More generally, the so-called smoothed PA or SPA (Gong and Ho 1987, Glasserman and Gong 1989) approaches the issue of Eq. (1), the interchangeability question, through a slightly different form of smoothing. We first decompose the expectation in (1) into two parts, a conditional expectation and another expectation over the conditioning variables, i.e.,

$$\begin{aligned} dE[L(\theta,\xi)]/d\theta &= d\, E_Z E_{/Z}[L(\theta,\xi)]/d\theta \\ &= E_Z \{ d[E_{/Z}[L(\theta,\xi)]]/d\theta \} \end{aligned} \tag{4}$$

We can expect $E_{/Z}[L(\theta,\xi)] \equiv L(\theta,z)$ to be smoother than $L(\theta,\xi)$ and hence may make the interchange between E_Z and $d/d\theta$ possible. Note that even though $L(\theta,\omega)$ may be discontinuous, sufficient amount of averaging can make $E[L(\theta,\xi)]$ differentiable. In fact, the usual brute force way of computing sensitivity via

$$dL/d\theta = \lim_{n\to\infty,\ \Delta\theta\to 0} \frac{1}{n} \{ \sum_{i=1}^{n} L(\theta+\Delta\theta,\xi_i) - \sum_{i=1}^{n} L(\theta,\xi_i) \} \tag{5}$$

is a simple statement of the above fact. Now the trick with Eq.(4) is to average just enough to avoid discontinuities but not to require the duplication of another experiment as in the brute force case. Between the extremes of differentiating the expectation and taking expectation of the differentiation, a whole spectrum of partial expectation and smoothing possibilities exist . In fact, Glasserman and Gong (1989) showed explicitly that for a large class of problems with inherent discontinuities in performance SPA can yield unbiased derivative estimates by converting the differentiation with respect to L to differentiation with respect to the probability of the occurrence of such discontinuities via appropriate conditioning. The most recent reference is Fu and Hu (1992) which also leads to the next issue.

Of course, SPA begs the question of what to use for the conditioning variable "z"? Both the earlier and recent developments (Bremaud andVasquez 1991, Bremaud and Gong 1991, Shi 1992, Dai and Ho 1992) suggest a natural decomposition and conditioning of the difference $L(\theta+\Delta\theta,\xi)-L(\theta,\xi)$ on the fact that each event in the nominal path has a probability of being deleted due to perturbation (recall again the example of the equivalent ways of generating Poisson stream with rate $\lambda-\Delta\lambda$ mentioned above and the (S) smoothing idea), i.e.,

$$\begin{aligned} \frac{dJ}{d\theta} &= \lim_{\Delta\theta\to 0} \frac{1}{\Delta\theta} E\,[\, L(\theta+\Delta\theta,\xi) - L(\theta,\xi)\,] \\ &= \sum_i \lim_{\Delta\theta\to 0} \frac{1}{\Delta\theta} E[(L_{-i}-L)|i] \end{aligned} \tag{6}$$

where L_{-i} is the sample performance with the ith event deleted as a result of $\Delta\theta$. Now we evaluate the conditional expectation of L_{-i} -L as

$$= \sum_i \lim_{\Delta\theta\to 0} \frac{1}{\Delta\theta}[(L_{-i}-L)\frac{dP_{-i}}{d\theta}\Delta\theta] = \sum_i [(L_{-i}-L)\frac{dP_{-i}}{d\theta}] \tag{7}$$

where $dP_{-i}/d\theta\Delta\theta$ is the probability that the ith event will be deleted due to $\Delta\theta$. Note, once again we avoid the problem of dividing through by $\Delta\theta$ by being able to differentiate the probability P_{-i} . Also P_{-i} is usually directly related to the given elementary random variable distributions. Differentiating it does not present numerical difficulties as in the likelihood ratio (LR) method where one is differentiating the distribution of the stochastic process x(t). In fact,we can view (7) as a kind of improved LR method where we avoid the variance problem of differentiating the sample path distribution and the term L_{-i} as a kind of control variate to further minimize the variance [Shi 1992]. Of course, (7) merely represents a form of intelligent brute force calculation of

(1). The term L_{-i} - L can be easy or complex to compute. Regeneration cycles certainly help (in the sense of reducing the summation over i). A Markov assumption here or there helps. Also alternatives to L_{-i}-L, such as L_{+i} -L and L_{+i} -L_{-i} terms (With the obvious notational interpretations) to provide computational flexibility in the calculation of (5) and (7) have been given by Shi (1992). Other authors, Dai and Ho (1992), and Bremaud and Gong (1991) showed other alternatives, such as instead of deleting an event, perturbing its consequence, the next state. Otherwise [6] and [7] remain the same. Fu (1990) also give another application of SPA to inventory systems.

The other extension to IPA is **Calculus based**. It starts by considering the replications in (1) as decomposed along event sequences, $\sigma=e_1, e_2, \ldots, e_i, \ldots$ where e_i is the ith event of the sample path x(t), i.e.,

$$J(\theta) = \sum_{\sigma} E[L(x(t;\theta,\xi)) \mid \sigma]$$

and

$$\frac{dJ(\theta)}{d\theta} = \sum_{\sigma} \frac{dE[L(x(t;\theta,\xi)) \mid \sigma]}{d\theta} \tag{8}$$

where the conditional expectation is taken over all sample paths that are deterministically similar to the event sequence , σ. The boundary of the integration (conditional expectation) is a hyper cube, $R(\sigma)$, in the multi-dimensional event sequence space with edges defined by the upper and lower limit on the timing of the events before they change order with their neighboring events. Thus

$$\frac{dE[L(x(t;\theta,\xi)) \mid \sigma]}{d\theta} = \frac{d}{d\theta}\int_{R(\sigma)} L(x(t;\theta,\xi))dF_{x/\sigma} \tag{9}$$

Now by elementary calculus,

$$\frac{d}{d\theta}\int_{R(\sigma)} L(x(t;\theta,\xi)dF_{x/\sigma} = \int_{R(\sigma)} \frac{dL(x(t;\theta,\xi)}{d\theta}dF_{x/\sigma} + \frac{dR}{d\theta} \tag{10}$$

The first term on the r.h.s. of (10) is simply the usual IPA term and can be calculated by simple IPA rules. The interchange of integration with differentiation is valid by definition of deterministic similarity. The second term is the correction term that must be added. Gaivoronski-Sreenivas-Shi (1992) have many explicit examples showing the validity of Eq.(10) and the solution of otherwise non IPA-applicable problems. $dR/d\theta$ can be thought of as a bias correction term which may be easy or difficult to calculate. However, Eq.(10) offers valuable insight, e.g., consider the Lindley equation for system time of a GI/G/1 queue. We have,

$$T_{n+1} = \begin{cases} T_n + S_{n+1} - A_{n+1} & \text{if } T_n - A_{n+1} \geq 0 \\ S_{n+1} & \text{otherwise} \end{cases}$$

and

$$\frac{dT_{n+1}}{d\theta} =? = \begin{cases} \frac{dT_n}{d\theta} + \frac{dS_{n+1}}{d\theta} & \text{if } T_n - A_{n+1} \geq 0 \\ \frac{dS_{n+1}}{d\theta} & \text{otherwise} \end{cases} \tag{11}$$

But to prove (11) we must essentially prove that perturbation in the boundary condition $T_n - A_{n+1} \geq 0$ will not affect adversely (i.e., to first order) the validity of (11). This fact is, of course, well known in terms of the by now familiar illustration of Fig.1 in §2.

4. FINITE PERTURBATION ANALYSIS

More generally, PA is a mind set concerned with the **EFFICIENT** exploration of the performance response surface $J(\theta)$ via multiple experiments at different θ's. In particular, we submit

(i) A sample path of a DEDS (real or simulated) inherently contains information about the system far beyond the usual summary statistics, such as time or ensemble averages of variables of interest. If this information is collected in time and processed appropriately, it can yield gradient and other useful performance data. For example, by analyzing a long sample path of a G/G/1 queue at one value of its service rate, one can deduce its performance at all other values of the service rate (Gong & Hu 1991).

(ii) If the model of a DEDS did not change except for some parameter values, then the separate generation of sample paths in traditional simulation for "what if" studies entails a great deal of duplicated effort that can be and should be leveraged to improve computational efficiency.

In other words, Finite PA takes the viewpoint that simulation is a mapping of (θ, the system parameters, and ξ, a sequence of u.i.i.d. random numbers $\in [0,1)$) to L. $L(\theta+\Delta\theta,\xi)$ is merely a slightly different mapping. Particularly in view of the GSMP formulation presented in §1, we note that the clock mechanism which is ξ need not be duplicated. Generating a different trajectory $(\theta+\Delta\theta,\xi)$ means picking out and assembling different pieces of the clock mechanism according to the state transition rules in the structural part of the GSMP. One immediate consequence is the cut-&-paste idea of Ho & Li (1988), Ho & Li & Vakili (1988) under Markov clock assumption and the related augmented Markov chain approach of Cassandras & Strickland (1988).

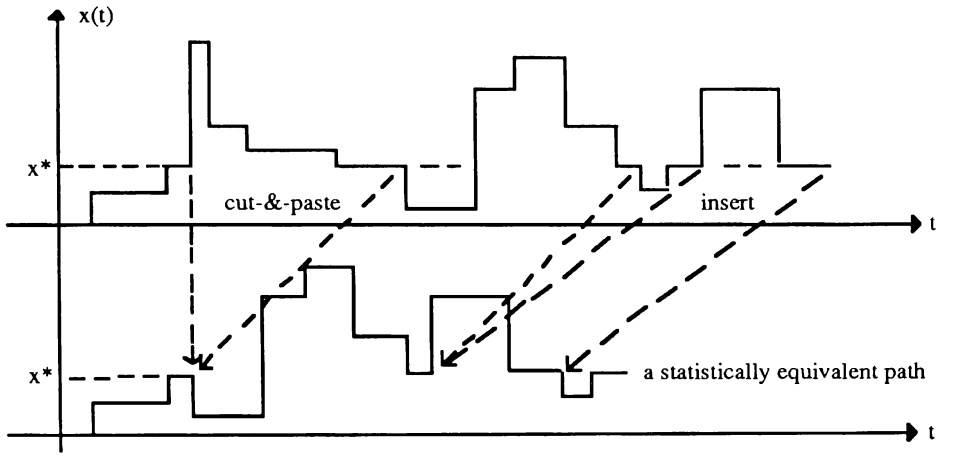

Figure 2 Markov Trajectory under "cut, insert and paste"

Basically, we view a sample path generated under the system parameter value of θ as made up of many segments each of which operates under state sequence invariance, i..e., so long as the states of a DEDS experiment are the same, we can append or cut pieces of the trajectory arbitrarily. By appropriately "cut, insert, and paste-ing" together different segments one can in fact regenerate sample paths which are statistically indistinguishable from what would be generated under $\theta+\Delta\theta$. Fig.2 pictorially illustrates this for a Markov trajectory.

The "cut-&-paste" idea can also be viewed as a dynamic system extension of the rejection method of random variable generation (Fishman 1988). In this view, trajectory generation for a number of **structurally similar but parametrically different systems** can be efficiently carried out in parallel minimizing duplications. In particular, massively parallel SIMD (Single Instruction Multiple Data) computers are ideally suited to carry out such parallel simulations. Since experiments rather than the simulation procedure is distributed or parallelized, no synchronization problem (Fujimoto 1989, Rego & Sunderam 1992) exists and scalability approaches the theoretical maximum. This leads to the idea of the **Standard Clock** (SC) approach to parallel simulation which is a radical departure from traditional simulation methodology.

5. STANDARD CLOCK AND PARALLEL SIMULATION

Starting with the assumption that all random phenomena in the DEDS are exponentially distributed, we generate a single Markov clock, called the standard clock at the rate

$$\Lambda = \sum_i \lambda_i$$

where λ_i is the rate of the ith event type (arrival, service, etc.). In other words, this is the maximal rate at which events can possibly happen in the system. Now this maximal rate must be THINNED or filtered due to perturbations and/or feasibility. For example, no departure is allowed from a server if it is idle. Procedurally, the method starts by picking up an event from the SC stream. We use a ratio yardstick and a random number $u \in [0,1)$ to determine the type of this event as in Fig. 3

Once the event type is determined, we simply check against the feasible event set, $\Gamma(x)$, for the feasibility of this event in the current state. If feasible, we accept the event and use it to trigger the transition to the next state according to the state transition function, $x_{next}=f(x_{now}$, triggering event). The cycle repeats. Note that this procedural cycle is the same regardless of which experiment we are doing. In other words, we can do parallel experiments using an SIMD massively parallel computer. Thousands of replications or parametrically different but structural similar experiments can be run in parallel with no synchronization problems and taking only as much time as a single run. We only need to keep a separate state for each separate experiment/replication. Perturbation in system parameters will produce different states which will cause the same event type to be accepted or rejected by different experiments. Note also if rate perturbation on λ_i is desired, we only need to introduce a perturbed yardstick similar to Fig.3 resulting in different event type determinations for different experiments. No other changes are needed. Fig.4 compares this SC approach with the traditional simulation approach on sequential computers. Note that here we take maximal advantage of the separation and independence of the clock mechanism from the state transition function in the GSMP formalism.

Two additional observations:

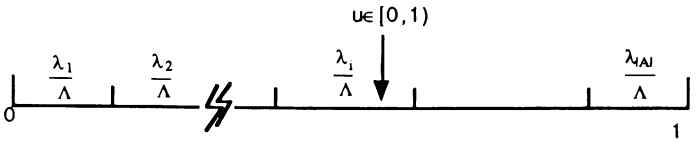

Figure 3 Event type Determination via the Ratio Yardstick

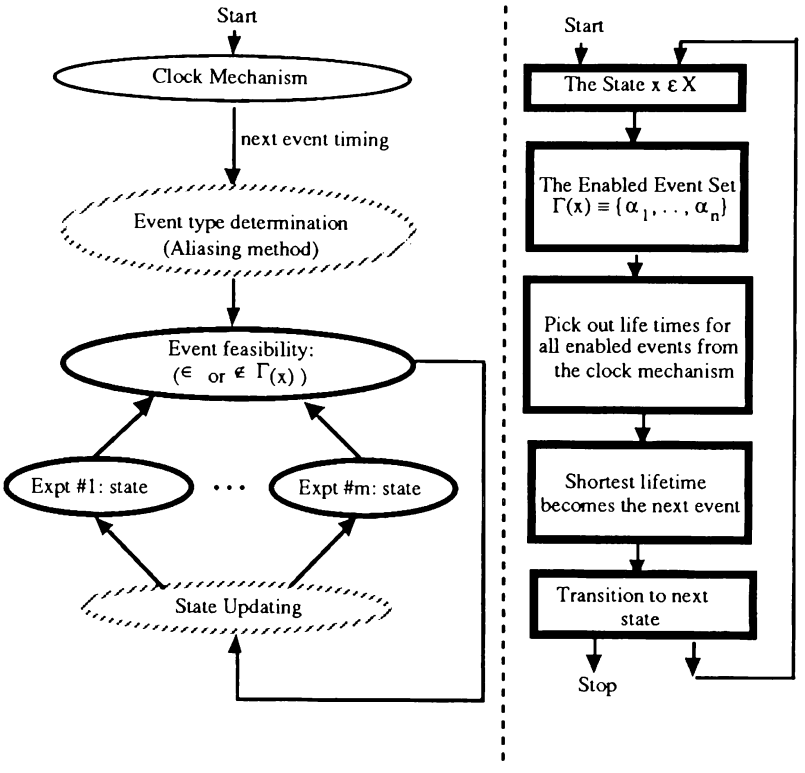

Figure 4 SC and Traditional Simulation Methodology

(i) The Markov clock assumption, while fundamental, can be relaxed via approximation. We can use the state not only to accept/reject events according to Γ(x) but also to shape the distribution such that the means and variances of the resultant event streams accepted can be arbitrarily matched. (This is simply an efficient way of implementing the "method of stages" approximation to arbitrary distributions.) The point is that in SC the rule of system operation, namely the state transition function, can be totally arbitrary and used to suit our simulation. We submit that being able to accommodate complex rules is much more important in simulation than in matching exactly the distributions of various event types.

(ii) The SC is also suited for real time application, i.e., using the event traces of a real world operation to generate in parallel other "what-if" experiments as the real system evolves. Thus, on-line monitoring and control becomes a real possibility. The details of SC can be found in Ho & Cao (1991 §7.1), Vakili (1989,1990ab), Ho & Li & Vakili (1988), Ho & Cassandras & Makhlouf (1992), and Vakil & Mollamustafaoglu &Ho (1992).

6. STOCHASTIC AND ORDINAL OPTIMIZATION

The raison d'etre for PA is, of course stochastic optimization via simulation. For a fixed computing budget, the key issue faced by any kind of iterative hill climbing scheme other than theoretical convergence is the trade-off between estimating accurately the gradient vs taking more steps of the iterative process. Experimental evidence seems to favor taking more steps. There is a large literature in general . For the present context, see Suri-Leung (1989) and Chong & Ramadge (1992).

Finally, it is worthwhile to emphasize that the purpose of design and performance evaluation of systems is to find good, better, or best designs first. Then we worry about "how good" is the selected design or designs. In short, **ordinal** optimization comes before **cardinal** optimization. The main virtue of ordinal optimization is this:

The relative order of performance of a system as a function of parameter values is

insensitive to errors in the estimation of the performances (Ho-Sreenivas-Vakili 1992)

For example, if we estimate the performance of a system under 200 different parameter designs very approximately via simulation and pick the designs corresponding to the top-12 **observed** performances, then there is 50-50 chance that at least one **actual** top-12 design belong in the picked set even if the performance estimation error has infinite variance.

Given this fact, *instead of trying to successively improve upon a design sequentially (e.g., by the traditional way of hill climbing/steepest descent), and spending a large effort to insure the accurate estimation of these intermediate results which will be eventually discarded, we propose to simultaneously and approximately evaluate many designs in concert with simultaneous simulation experiments as discussed above in § 4.* Not only does this approach provide us with a global view of the response surface often ignored in local hill-climbing types of procedures and permits quick localization of promising search regions, but it also allows for taking maximal advantage of ideas, such as approaches based on order statistics, genetic-like search , etc. See Deng & Ho & Hu (1992) , Garai and Ho and Sreenivas (1992) Vakili,Mollamustaflaoglu & Ho (1992). Space limitation prevents us from elaborating on this. But it is an integral part of doing efficient simulation for performance optimization.

7. CONCLUSION AND RESOURCES FOR FURTHER STUDY

We submit that the subjects of simulation and stochastic optimization are entering a new age. No less than four major federal agencies have identified "Simulation and Modeling" as a critical technology for the '90s (U.S. Department of Commerce, *Emerging Technologies: A Survey of Technical and Economic Opportunities*, Spring, 1990; U. S. Department of Defense, *Critical Technologies Plan*, March 15, 1990; Council on Competitiveness, *Gaining New Ground: Technology Priorities for America's Future*, 1990; Office of Science and Technology Policy, *Technology Critical to Economic Prosperity and National Security*, April 25, 1991.) Instead of merely being a subset of statistics and emphasizing output analysis, a whole range of new conceptual and analysis problems in simulation and modeling taking into account the dynamics of DEDS, the impact of new hardware technology, and a new mind-set are awaiting exploration and solution.

For further study on PA, the books by Glasserman (1990), and Ho & Cao (1991) and the tutorial article by [Suri (1989) and the taxonomy article (Ho & Strickland 1991) should be the first sources. Glasserman & Glynn (1992 this proceeding) has a turorial on advanced aspects of IPA/LR. These sources contain extensive references to other articles on PA which now total over 100 not all listed below. The Journal on DEDS, and the IEEE Transactions on Automatic Control regularly publish new articles on PA. An e-mail bulletin board on PA/DEDS also exists c/o padeds@virginia.edu.

ACKNOWLEDGEMENT

The work in this paper is supported by NSF grants CDR-88-03012, ECD-90-44673, ONR Contracts N00014-89-J-1023 and 90-J-1093, and Army Contracts MIT-GC-R-102337 and DAAL-91-G-0194

REFERENCES

Bello, M.,"The Estimation of Delay Gradient for Purpose of Routing in Data Communication Networks", S.M. Thesis, Electrical Engineering Department, M.I.T., 1977.

Bremaud, P., and F. J. Vazquez-Abad, ,"On the pathwise computation of derivatives with respect to the rate of a point process: the phantom RPA method,"*Queueing Systems* , 10, 1992 249-270,.

Bremaud, P. and Gong, W.B., "Derivatives of Likelihood Ratios and Smoothed Perturbation Analysis for The Routing Problem" submitted to*ACM Transactions on Modeling and Simulation.* also Rapports de Recherche, No. 1495, INRIA-Sophia Antipolis, 1991

Cao, X. R.,"Convergence of Parameter Sensitivity Estimates in a Stochastic Environment", *IEEE Transactions on Automatic Control* AC-30, 834-843, 1985.

Cassandras, C.G., and Strickland, S.G., "Sample Path Properties of Timed Discrete Event Systems", in *Discrete Event Dynamic Systems* (Y.C. Ho, Ed.), pp. 21-33, IEEE Press, 1991.

Cassandras, C.G., Abidi, M.V., and Towsley, D., "Distributed Routing with On-Line Marginal Delay Estimation", *IEEE Trans. on Communications,* COM-38, 3, pp. 348-359, 1990.

Cassandras, C., Gong, W.B., & Pan, J."The RIPA Algorithm for M/G/1(∞,K) Queue", *1990 IFAC Congress Proceedings*, Pergamon Press.

Chong, E. and Ramadge, P., "Convergence of Recursive Optimization Algorithms Using Infinitesimal Perturbation Analysis Estimates", *J. of DEDS*, 1,4, 339-372, 1992

Dai, Li-Yi, and Ho, Y.C.,"Structural Infinitesimal Perturbation Analysis", *IEEE Trans. on Automatic Control* , 1992 submitted.

Deng, M., Y.C. Ho, and J.Q. Hu,"Effect of Correlated Estimation Error in Ordinal Optimization", Proceedings of the Winter Simulation Conference , 1992.

Fishman, G. S., *Principles of Discrete Event Simulation*, Wiley , 1978.

Fu, M. and Hu, J.Q. "Extensions and Generaliuzation of Smoothed Perturbation Anal;ysis in a GSMP Framework", *IEEE Trans. on Auto. Control* , 1992 to appear

Fu, M. Sample Path Derivatives for (s,S) Inventory Systems", *Operations Research* , submitted 1990.

Fujimoto, R.M.,"Parallel Discrete Event Simulation", *Comm. of ACM* 33(10), 31-53, 1990.

Gaivoronski, A., Shi, Leyuan, Sreenivas, R., "Augmented Infinitesimal Perturbation Analysis: An alternate explanation" *J. of Discrete Event Dynamic Systems* , 1992 to apprear.

I. Garai, Y. C. Ho, and R. Sreenivas,"Hybrid Ordinal Optimization", *Proc. of IEEE Conference on Decision and Control* , 1992.

P. Glasserman, J.Q. Hu, and S.G. Strickland, "Strong Consistency of Steady State Derivative Estimations," *Probability in the Engineering and Information Sciences*, Vol. 5, pp. 391–413, 1991.

Glasserman, P. and Yao, D.,"Algebraic Structure of Some Stochastic Discrete Event Systems with Applications", *Journal of Discrete Event Dynamic Systems* 1, 1, 1991.

Glasserman, P. and Gong, W.B.,"Derivative Estimates from Discontinuous Realizations: Smoothing Techniques", *Proceedings of the Winter Simulation Conference* ed E. A. MacNair, K. J. Musselman, and P. Heidelberger, 381-389, 1989a.

Glasserman, P., *Gradient Estimation via Perturbation Analysis*, Kluwer Academic Publisher, 1990

Gong, W.B. and Ho, Y.C.,"Smoothed Perturbation Analysis of Discrete Event Dynamic Systems", *IEEE Transactions on Automatic Control* AC-32, 10, 858-866, 1987.

Gong, W.B. and Hu, J.Q.,"The Light Traffic Derivatives for the GI/G/1 Queue", *Journal of Applied Probability* , 1991.

Heidelberger, P., Cao, X. R., Zazanis, M. R. & Suri, R. "Convergence Properties of Infinitesimal Perturbation Analysis Estimates", *Management Science*, 34 11, 1281-1302, 1988.

Ho, Y.C., Sreenivas, R., Vakili, P.,"Ordinal Optimization of Discrete Event Dynamic Systems", *J. of* Discrete Event Dynamic Systems 2(2), 1992, 61-88.

Ho, Y. C., Li, S., and Vakili, P.,"On the Efficient Generation of Discrete Event Sample Paths under Different Parameter Values", *Mathematics and Computation In Simulation* 30, 347-370, 1988.

Ho, Y.C. and Li, S.,"Extensions of the Perturbation Analysis Techniques for Discrete Event Dynamic Systems", *IEEE Transactions on Automatic Control* AC-33(5), 427-438, 1988.

Ho, Y.C., Eyler, A., and Chien, T. T.,"A Gradient Technique for General Buffer Storage Design in a Serial Production Line", *International Journal on Production Research* 17(6), 557-580, 1979.

Ho, Y.C., Cassandras, C., Makhlouf, M.,"Parallel simulation of Real Time System via the Standard Clock Approach", *Mathematics and Computers in Simulation* , 1991.

Ho, Y.C. and Strickland, S. "A Taxonomy of PA Techniques", in *Introduction to Discrete Event Dynamic Systems*, IEEE Press 1991

Ho, Y. C. & Cao, X.R. *Perturbation Analysis of Discrete Event Dynamic Systems*, Kluwer Academic Publishers, June 1991

Rego, V.J., Sunderam, V.S.,"Experiments in concurrent Stochastic Simulation: The EcliPSE Paradigm", *Journal of Parallel and Distributed Computing* 14, 66-84, 1992.

Shi, Leyuan, "Discontinuous Perturbation Analysis", submitted *IEEE Trans. on Automatic Control* , 1992 see also Ph.D. Thesis of L. Shi, Division of Applied Science, Harvard University 1992

Suri, R. "Perturbation Analysis: The State of the Art and Research Issues Explained via the G1/G/1 Queue" *Proceedings of the IEEE*, 77, 114-137, 1989

Suri, R. & Zazanis, M. "Perturbation Analysis Gives Strongly Consistent Sensitivity Estimates for the M/G/1 Queue," *Management Science* , 34 (1), 39-64, 1988

Suri, R. and Leung, Y.T. "Single Run Optimization of Discrete Event Simulations - An Empirical Study using the M/M/1 Queue" *IIE Transactions*, 21, 1, 35-49, 1989

Vakili, P.,"Three topics on Perturbation Analysis of Discrete-Event Dynamic Systems", Ph. D. Thesis, Harvard University , 1989.

Vakili, P.,"Using Uniformization for Derivative Estimation in Simulation" *Proceedings of the American Control Conference* , 1034-1039, 1990a.

Vakili, P. "A Standard Clock Technique for Efficient Simulation", *Operations Research Letters* , 10, pp. 445-452, 1991.

Vakili, P. Mollamustafaoglu, L., Ho, Y.C.,"Massively Parallel Simulation of a Class of Discrete Event Systems", *Proc. of the IEEE Symposium on the Frontier of Massively Parallel Computation* , 1992.

W. Wardi and J.Q. Hu, "Strong Consistency of Infinitesimal Perturbation Analysis for Tandem Queueing Networks," *Journal of Discrete Event Dynamic Systems*, Vol. 1, No. 1, pp. 37-59, 1991

Woodside, C.M.,"Response Time Sensitivity Measurement for Computer Systems and General Closed Queueng Networks", *J. of Performance Evaluation*, 4, 199-210, 1984.

AUTHOR BIOGRAPHY

Yu-Chi Ho received his S.B. and S.M. degrees in Electrical Engineering from M.I.T. and his Ph.D. in Applied Mathematics from Harvard University. Except for three years of full time industrial work he has been on the Harvard faculty where he is the T. Jefferson Coolidge Chair in Applied Mathematics and Gordon McKay Professor of Engineering..

He has published over 100 articles and three books, and is the recipient of various fellowships and awards including the Guggenheim (1970) and the IEEE Field Award for Control Engineering and Science (1989). He is a fellow of IEEE, a Distinguished Member of the Control Systems Society, and a member of the U.S. National Academy of Engineering.