SRC TR 87-136

Performance Evaluation and Perturbation Analysis of Discrete Event Dynamic Systems

by

Yu-Chi Ho

Performance Evaluation and Perturbation Analysis of Discrete Event Dynamic Systems

YU-CHI HO, FELLOW, IEEE

Abstract—**This is a conceptual and speculative paper concerning the future development of system and control theory in operational and discrete event systems with particular emphasis to the techniques of perturbation analysis.**

I. Introduction

Picture yourself with the mythical Mr. T. S. Mits (The Scientific Man In The Street) and the task of explaining to him the phenomena and workings of 1) the Gulf Stream and 2) a computer-controlled flexible manufacturing system (FMS). Both phenomena are real and both are not completely understood. However, for task 1) you face an easier task since you can draw upon knowledge of calculus and the differential equation to provide a succinct description of ocean currents and fluid dynamics. For task 2) no such ready-made models are in existence.[1] One is reduced to essentially an algorithmic description not that different from writing a computer program to simulate the FMS. In fact, modern technology has increasingly created dynamic systems which are not easily described by ordinary or partial differential equations. Examples of such systems are production or assembly lines, computer/communication networks, air traffic systems, etc., where the evolution of the system in time depends on the complex interactions of the timing of various discrete events, such as the arrival or departure of a job, the completion of a task or message. The *state* of such dynamic systems changes only at these discrete instants of time instead of continuously. We shall call such man-made systems discrete event dynamic systems (DEDS) as opposed to the more familiar continuous variable dynamic systems (CVDS) in the physical world that are described by differential equations. Although systems governed by difference equations are often referred to as discrete-time systems, conceptually they have more in common with CVDS than with DEDS despite the name similarity. For the purpose of this paper, we shall not distinguish between differential and difference equations. To help fix ideas for DEDS, consider a flexible manufacturing system (FMS) [1], [2], [11] with several work stations each consisting of one or more identical machines. These stations are attended by operators and/or inspectors. Parts belonging to different classes arrive at these stations via computer control following some routing plan. They queue up in the buffers according to some priority discipline until a machine and an operator are available to work on them. Stations are connected by some kind of material handling system (MHS) or automatic guided vehicles (AGV) which transport parts from station to station until they finish processing and leave the system. Typical performance criteria of interest for such an FMS are average throughput (TP), flow or wait time (WT), and work in process (WIP). Note that with some simple changes in terminology from parts to messages, work station to nodes, routes to virtual circuits, MHS to communication links, AGV to packets, fixtures to tokens, etc., the above description could be for a communication network which moves and processes information packets rather than material parts [12]. Multiprogrammed computer operating in the batch and time shared mode is another example fitting this description. The point here is the pervasive nature of such systems in the modern world and the relative lack of good analytical and dynamical oriented models for their description.

The purpose of this paper is twofold: to advocate to the control system theorists and engineers the rich opportunities in the field of DEDS in general, and to outline some more specific open problems in a newly developed technique, called perturbation analysis (PA) for DEDS in particular. DEDS is used here in the narrow sense as a parallel to CVDS in the form of $dx/dt = f(x, u, t)$. We shall not address issues such as implementation, integration, interfaces, and hardware in DEDS. By this we do not imply that such issues are unimportant. But our narrower definition of DEDS is consistent with our past use of the term as well as with the convention adopted in discussions of control theory for CVDS. Conceptually, one can visualize the entire sweep of classical control-theoretic problems, such as, controllability and observability, estimation and identification, information, and control awaiting formulation for DEDS. Mathematically, there are a multitude of specific challenging analytical problems in PA demanding resolution. Although we discuss conceptual and visionary issues, the emphasis is on the subject of performance evaluation of such systems since we believe ultimately it is the practical and useful technology that will drive the development of the field. Consequently, we shall admit at the outset that there is a bias against computationally infeasible tools. Perhaps a separate paper advocating the *pure* theory of DEDS can be published to discuss the intrinsic nature of DEDS without reference to performance issues. Our purpose here is to stimulate, to challenge, to speculate, and to provide one perspective, but not necessarily to record something of archival value.

II. Discrete Event Dynamic Systems

Conceptually, we can visualize such a DEDS consisting of *jobs* (parts) and *resources* (machines, operators, AGV's, MHS, and buffer spaces). Jobs travel from resource to resource demanding and competing for service. The dynamics of the system is determined by the complex interactions of the timings of various discrete events associated with the jobs and resources. In this sense, DEDS are simple. There are only two objects in DEDS, jobs and resources, which interact. When they do, job occupies (receives service from) the resource for a random/deterministic period of time. If we let the number of jobs waiting at a resource be a state variable $x(t)$ and, furthermore, consider the approximation that there are an infinite number of jobs each infinitesimally small, then we can consider the differential equation

$$\begin{aligned} dx/dt &= \lambda - \mu \qquad x>0 \\ &= \lambda \qquad x=0 \end{aligned} \tag{1}$$

Manuscript received September 4, 1986; revised January 6, 1987 and February 23, 1987. Paper recommended by Associate Editor, W. R. Perkins. This work was supported by the National Science Foundation under Grants ECS-82-13690 and CDR-85-001-08, by the Office of Naval Research under Grant N00014-84-K-0465, N00014-79-C-0776, and by the Draper Laboratory.

The author is with the Division of Applied Sciences, Harvard University, Cambridge, MA 02138.

IEEE Log Number 8714812.

[1] We do not regard queueing networks or automata models of an FMS as being accessible to Mr. T. S. Mits. More about these models in Section II.

where λ is the rate of job arrival and μ is the service rate of the resource. This is a deceptively simple model of one component of a queueing system. The complications come in because λ and μ can depend in arbitrary ways on other state variables of the DEDS. In other words, jobs compete and wait for service by the resources in very very complicated ways. Here we have an endless variety of queueing disciplines, priorities, service requirements, routing, resource sharing, and general logical conditions that need to be met for interactions to take place. Any attempts to describe a DEDS by (1), even in cases where the approximation is appropriate, results on the right-hand side of (1) are so complicated as to be useless. This description amounts to the writing of a computer program to simulate the DEDS. In fact, general purpose discrete event simulation languages provide the constructs of jobs, resources, timing of events, and logical tests while the coding in such languages produces the description of the specific systems. At this point we simply have no convenient way to capture these descriptions mathematically with the same degree of efficiency as the case of CVDS with differential equations. Thus, as mentioned earlier, the workings of such DEDS are described through a system of "rules of operation" or "algorithms." This is essentially a brute force approach. However, it is important to emphasize that DEDS are nevertheless "dynamic systems" in the usually understood sense of the term, i.e., it is a quintuple consisting of (input set, output set, state set, state transition map, and output map[2]). See, for example, [77, chs. 6 and 9]. But the specification of these five objects are far from succinct and pristine as is in mathematical system theory (e.g., compare [7, Section 3.1] or [40], [41] versus [8, p. 154]). Nevertheless, the fact that we can implement *general purpose* discrete event simulation languages can be construed as testimony to the "dynamical" nature of these systems.

A. Existing Models for DEDS

Attempts have been made in the past with varying degrees of success to model analytically DEDS. The most basic being the use of *finite state Markov chains or processes* [13]. We assume that the state, input, and output set are *finite* or countable. Consequently, the state transition and output map can be modeled by finite or countably infinite matrix of transition probabilities. This is perfectly general. The only difficulty is computational and structural. The number of states in a typical DEDS can be combinatorially large. For example, consider a serial production line of M stations which can be either in working order or being repaired. Each station has associated with it a buffer (queue space) of size K_i. Then the total number of states of the production line when viewed as a finite state machine is

$$\text{number of states} = \left(\prod_{i=1}^{M} (K_i + 1)\right)(2^M) \tag{2}$$

which can easily reach billions for relatively small M and K_i (two possible states for each of the M machines and $K_i + 1$ states for the ith buffer storage). Secondly, in such a finite state approach, all structural information about the system is lost. The states are completely one-dimensionalized in the sense of being listed one after another with no particular distinctions. It is difficult to establish approximating notions such as neighboring states or to identify a given state with a particular system configuration. Thus, microscopic and finite state approaches such as petri nets, and automata, primarily are more useful for answering qualitative (yes/no) or conceptual rather than quantitative questions. Outside of academic examples, it does not seem hopeful that we can do anything computationally useful for engineering purposes unless a successful theory of aggregation can be developed. At present, this is hopeful but not yet successful. We shall have more to say about this in Sections II-B and IV. In this vein, we must also classify the works on extended state machines [64], [70]. Without general aggregation techniques, they do not at this point appear to be computationally feasible.

On the other hand, the "network of queues" model developed in *Operations Research* in the 1960's and 1970's does preserve structural information and solves the computational problem [14]–[16]. Here, in the simplest case, we take as the "state" the vector with each component representing the number of jobs waiting and being served at each service station. The theory then strives to develop a description for the equilibrium probability distribution of the state vector. This is indeed possible for the so-called "product form" class of networks. A fair amount of modeling successes have been obtained in practice, particularly in the computer system performance analysis area. The models have also been shown to be very robust with respect to many of the assumptions, such as exponentially distributed service time, of the theory. In fact, the results in this area can be derived by way of a completely different set of assumptions not based on probabilistic considerations. This is known as the operational analysis of queueing network [15]. A number of approximate analysis techniques based on this theory have been developed and made precise with respect to the nature of their approximations [14], [15], [16], [17], [35], [36]. For many rough cut analyses, this approach is eminently reasonable and practical; its main limitation being its generality. Features, such as, finite queue limit, state-dependent routing, simultaneous resource sharing (one job demanding several resources or one resource requiring several jobs to begin service, e.g., assembly/disassembly), and nonstandard queue disciplines, are difficult for the theory to handle.

Lastly, queueing theory is primarily a quasi-dynamic approach. While it does take into account jobs competing for resources and the time order of job arrival/departure, it does so only *on the average* in the steady state. To use a rough analog, the product form network of queues theory can be likened to the frequency response theory of linear stationary systems, namely, the technique solves one class of problem analytically, it transforms the time domain behavior of the system to a different domain, it deals only in steady state, and it can be extended to solve other classes of problems approximately but not universally successful in all applications. For example, systems which are totally deterministic and relatively simple often exhibit periodic behavior if breakdowns are ignored. For such systems, we cannot rely on "complexity" to imitate "randomness." Queueing theory is less applicable. However, for such a restricted class of discrete event systems, a relatively new algebra, the so-called minimax algebra, offers an elegant solution for modeling [65], [66]. On the other hand, for more general DEDS, the network is often so complex as to appear random even though the constituent parts all behave deterministically. This has been demonstrated experimentally in a series of papers [73]–[76]. But, for a completely general description and analysis of DEDS, we are left with the only alternative of *simulations*.

In principle, simulation is a completely general tool and a "dynamics oriented" tool. A simulation model can be made as accurate as one desires, limited only by cost and time. To put it simply, it is brute force trial and error experimentation using a computer model of the real thing. Such efforts can be time consuming and expensive, particularly for parametric studies. Consequently, most of the efforts in simulation have to do with the following:

1) good language design and friendly software to make simulation modeling easy;

2) statistical analysis of outputs, such as variance reduction, regenerative simulations, etc., to make the experiments more efficient and thus ameliorating the cost.

Mathematically, the theory of generalized semi-Markov processes (GSMP) has been advanced as a formal model of simulated discrete event process [40], [41]. Its main feature is to distinguish

[2] The transition and the output map for DEDS have to be interpreted to include objects such as time advance and event selection mechanisms commonly found in discrete event simulation languages.

the discrete/countable (e.g., queue contents) and continuous/uncountable (e.g., remaining service times) parts of the state space. The former is defined as the *state* of the GSMP with the latter defined as *event clock readings*. This definition is conceptually useful. At present, however, the theory has not attained the level of succinctness and quantification that the CVDS has with differential equation models. The continued creation of new discrete event simulation languages such as SIMAN, DESFOR, GPSS-85, to name a few, are also evidence of the vitality and needs of the field [3], [78]. It is also worthy to note the almost complete absence of similar developments of simulation languages for the CVDS except for software dealing with a better numerical integration formula. This is another indirect evidence of the lack of a good analytical model for DEDS.

On the other hand, it should be pointed out that during the 1970's, system theorists at Berkeley did launch a major effort for a "dynamics oriented" approach to the study of stochastic point processes [51]-[55]. In fact, to quote Bremaud "... The Martingale calculus was applied to point process systems in much the same way as it had already been applied to Wiener-driven stochastic systems..." [51, p. xvii]. However, they may be ahead of their time. Manufacturing systems and computer communication networks had not achieved the degree of practical importance then as today. As a result, it is fair to say that the effort was more mathematically oriented in establishing a parallel with Wiener-driven processes in stochastic control than building new models to fit real world systems which are the *raison d' etre* for all the discrete event simulation language developments.

This brings us back to the thesis of the relative lack of computationally usable mathematical models, particularly "dynamics oriented" models for the descriptive and prescriptive analysis of such systems. This relative lack of mathematically succinct and computationally feasible models for DEDS cannot be overemphasized. Unlike their counterpart in CVDS where there is a long history of research results starting from calculus and Newtonian physics, DEDS are modern day phenomena that do not have the same well-established vertical intellectual structure. Just as it is most difficult to do mathematics without symbols, it is equally hard to do analysis of DEDS without good mathematical models. We submit that many of the problems of design, operations management, scheduling, and control of DEDS can be dealt with if good dynamic models of DEDS are available. To show this is not idle speculation, we mention the recently developed idea of "perturbation analysis" for DEDS [4]. This technique views DEDS as a stochastic dynamic system evolving in the time domain. By examining and analyzing the trajectory of such dynamic systems, it "linearizes" the dynamics of the system about the particular trajectory in question and answers "what if" questions about perturbations around this trajectory based on this linearization. This is completely parallel to the familiar perturbation analysis used for continuous variable dynamic systems governed by nonlinear ordinary differential equations. Inasmuch as linear system theory represents the most successful part of control theory, the development of perturbation analysis for DEDS so far can be regarded as a possible first step in an effort to create a control system theory and modeling methodology for such systems. Some initial successes have been achieved. Much more can be and remains to be done.

Furthermore, we submit that the development of a theory of DEDS requires serious experimental effort very similar to that of experimental physics in support of the effort of theoretical physics. Our success in controlling aerospace vehicles or other physical systems often obscures the enormous modeling and experimental effort of the last century in developing the differential equation model of continuous variable dynamic systems. It is on the basis of these efforts that our successes with control theory were built. However, no such parallel exists for DEDS which are very recent man-made phenomena. In fact, our experience in the development of the perturbation analysis is an example of the importance of experimentation. Many of the theoretical results of PA (to be discussed in Section III) are inspired by efforts to explain experimental findings first arrived at via intuition, heuristics, insight, and observed experiences. It is not an exaggeration to say that PA would never have been developed if we did not engage in the "observation-conjecture-experiment-validation" cycle of the research effort [5]. Without experimentations, we run the danger of creating "models without data" or doing "as if analysis" as theoretical economists are sometimes accused of doing. It is a trap all too easily fallen into by control theorists after decades of success and the unquestioned assumption of $dx/dt = f(x, u)$ as the starting point of analysis. More importantly, without experiments, it is difficult to determine the correct level of details to be modeled. If control theory is to make headway in DEDS, then this author submits that we must acknowledge properly the importance of experimentation [6]. It is unlikely that progress in DEDS can be made with the purely deductive logic of mathematics. On the other hand, we do not advocate blindly going ahead with data collection and simulation. "Data without model" simply creates an experience-based discipline. Only with the proper combination of experiments—inspired-theory and theory-induced-experimentation can there be meaningful progress.

B. The Nature of the Dynamics of DEDS

As we alluded to at the beginning of Section II, the dynamics of a queue is basically a *flow* of jobs in and out of a storage. The purpose of storage is to *smooth* out fluctuations in the demand for resources. Performance of a queueing system is thus basically a trade-off between utilization of the resources and the delay time experienced by the jobs. 100 percent resource utilization, in general, can be achieved only at the expense of long waiting times for most of the jobs. Conversely, instantaneous service requires a large reserve of resources to meet peak demands. To put it another way, we maintain that the long-term behavior of a DEDS is governed by the concepts of *continuity* and *conservation* of flow. In steady state, the flows in and out of a station or a part of a network balance in some probabilistic sense. For example, take the simple birth-death process in equilibrium. The probability that we find the population is equal to K is a constant. The probability that we move out of the state K due to birth or death is balanced by the probability that we move into the state K from $K - 1$ by birth and from $K + 1$ by death. This is called global balance. In addition, the probability that we move from state K to state $K + 1$ due to birth at a particular node of the system is further balanced by the probability of death at that node. This is called local balance. From these balance equations, we can calculate the expected behavior and the equilibrium distributions in very much the same way as the calculation of flows balance in hydrostatics. On the other hand, in the short run, the purpose of a queue is to *decouple* the behavior of one station from that of another. Thus, we speculate, and queueing network theory supports this with some evidence [13, p. 150], that for "dynamic" control and estimation it may be possible to analyze and control behavior one station at a time as if the arrival and departure are not affected by happenings at other parts of the network. By the same token, it probably will not be very fruitful to estimate the behavior of one part of a DEDS based on observation of that of another [18], [42]. This is very different in CVDS where action in one part of the system instantaneously affects every other part via the differential equation model of the dynamics. In other words there is no computationally easy Kalman-Bucy filter for DEDS.

To illustrate additional parallel developments in CVDS and DEDS, mention must be made to the idea of the Norton's equivalent of a queueing network [42]. This is the analogous idea of replacing a complex electrical network with a simple equivalent source and impedance for the purpose of external behavior calculation. It turns out that the same thing can be done for queueing networks that are "product form." Such an idea of aggregation and equivalence is fundamental to all engineering analysis and design. But what about more general cases? More about this later. In short, we submit that the entire range of control-theoretic problems that have been studied for the past two decades for CVDS have their possible counterpart in DEDS. Only the surface has been scratched [10], [37], [45]. The opportunities and the payoffs seem to be enormous.

In summary, we submit that many of the well-known concepts of system theory can be transplanted to and further developed in DEDS. In order to accomplish this, we need to change some well-entrenched notions of system theory as a purely deductive mathematical discipline. Rigorous experimentation must accompany mathematical modeling if we are to make substantial inroads to the intelligent control of such discrete event systems.

III. Perturbation Analysis

Perturbation analysis (PA) is an analytical technique that calculates the sensitivity of performance measure of a DEDS with respect to system parameters by analyzing its sample path. Ordinarily, unless closed-form formulas are available, performance sensitivity is calculated by brute force using two different experiments each with only the parameter value being different. This can, of course, be very time consuming, expensive, and numerically difficult. PA, in effect, is a method of reconstructing the perturbed performance value from the nominal (original) experiment or sample path of the DEDS without the need of actually carrying out the perturbed experiment. This point deserves emphasis. PA is simply an analytical means to process information inherent in the sample path of an experiment. It makes no difference whether or not the experiment deals with steady state or transient performance. To the extent PA is applicable to the particular experiment, it can be used for the determination of either steady state or transient performance gradients.

The starting point of PA is the recognition that the *timing of events* is the most basic element in the description of the behavior of a DEDS. (The "state" of a DEDS changes only at an event. Nothing of consequence occurs between two successive events.) It approaches the problem by decomposing the calculation of the system sensitivities into the following three parts.

1) How does the change in the value of a system parameter θ change the timing of various system events t_i, i.e., $\partial t_i/\partial\theta$? This is often referred to as the *perturbation generation* rules.

Example: Changing the mean service time of a resource will induce a series of changes in the service termination time of the resource.

2) How does the change in the timing of one event t_i change the timing of another t_k, i.e., $\partial t_K/\partial t_i$? This is often referred to as the *perturbation propagation* rules.

Example: If server A completes a job earlier and sends it onto a waiting server B, then server B will be able to start, and hence complete, service earlier.

3) How does the change in the timing of events t_K change the system performance PM, i.e., $\partial PM/\partial t_K$?

Example: The change in the time required to finish 100 000 jobs will change the average throughput of the system.

Questions 1) and 3) are parameter and performance measure specific. In the context of DEDS performance analysis, they can usually be resolved via ordinary calculus and probability [46]. However, question 2) is generic to the sample path of all DEDS. The answer for 2) is based on a very simple notion, namely that of the *critical timing path* (CTP) which couples the timing of one event to that of another. PA can be visualized as a very efficient method of keeping track of a large number of complex CTP's. In the parlance of discrete event simulation, the CTP is simply the "*event scheduler or future event list*" found in all general purpose simulation languages. Putting 1)–3) together, we get

$$dPM/d\theta = \sum_i (\partial PM/\partial t_K)(\partial t_K/\partial t_i)(\partial t_i/\partial\theta). \qquad (3.1)$$

The resemblance of (3.1) to the well-known equation of control theory

$$dPM/d\theta = \int (\partial J/\partial x(t_f))\Phi(t_f, t)(\partial f/\partial\theta)\, dt \qquad (3.1)'$$

for the system $dx/dt = f(x, \theta, t)$ and $PM = J(x(t_f))$ is no coincidence. To carry out (3.1) efficiently, we take advantage of the fact that the CTP of the nominal and the perturbed sample path is in fact the same for a sufficiently small perturbation in the parameter value. Thus, we can use the nominal sample path to reconstruct the perturbed sample path and performance. This is fine as long as we limit ourselves to the experiment of finite length and parameter values that are continuous. In such cases, we can always visualize a conjectured perturbation of small enough size such that the nominal and the perturbed trajectory have the same CTP. This condition is denoted as *deterministic similarity.*

On the other hand, the above argument is a two-edged sword. No matter how small the parameter perturbation, with probability one we can always find a sample path (or for any sample path if one waits long enough) such that the CTP of the nominal will be different from that of the perturbed. In other words, the order of the timing of some events will be changed due to the perturbation. Short of reconstructing the perturbed sample path, how can one *estimate* the perturbed performance based only on the nominal sample path? This problem can be understood in practical terms by considering two examples. First is the case of the sensitivity of throughput to routing probability in a queueing network. Intuitively, it is obvious that this sensitivity is continuous and nonzero in general. However, a naive perturbation analysis immediately encounters a dilemma. If the perturbation is small enough to maintain the same CTP for the nominal and the perturbed sample path, then there will never be any change in the routing of any jobs, and hence any change in the throughput. On the other hand, if a job is ever routed differently, then the nominal and perturbed path of the network from that point on can have vastly different CTP's. Thus, either we implement (3.1) with such a small $\Delta\theta$ so as to produce the incorrect answer of zero sensitivity or we require essentially brute force reconstruction of the perturbed sample path which PA purports to avoid. The situation can be visualized graphically as shown in Fig. 1 which illustrates generically the plot of a performance measure PM, say, sample throughput, with respect to a system parameter θ, such as routing probability. Note that the PM is a function of both θ and the sample path which we denote as ω[3]. For small perturbation in θ there is no change in PM as shown in Fig. 1(a) (or more generally a linear change as in Fig. 1(b) for other PM and θ). However, as θ changes beyond a certain value, a discontinuity in either the value or the slope of the PM results. This corresponds to the point at which the sample path under study will undergo a discontinuous change in its CTP, e.g., two or more discrete events taking place simultaneously are about to switch their order of occurrence, or the creation/destruction of an event such as an idle interval. For a different sample path ω', the plot of $PM(\theta, \omega')$ will have the same general character but different details as to the location of discontinuities and ordinate values. When averaged over ω, ω', and ω''... the resultant average PM takes the intuitively more reasonable shape as shown in Fig. 2 [for the case of Fig. 1(a)]. As the number of samples increases, it is entirely plausible that we obtain a continuous $PM(\theta)$ curve with nonzero slopes. Another example of the same type of phenomenon considers the case of the sensitivity of the average number of customers served in a busy period in a $G/G/1$ queue as a function of the mean service time (this example will be further discussed in Section III-C). Again intuitively we see that if the change in mean service time is sufficiently small, then no two busy periods will ever coalesce together, thus changing the number of customers served in a busy period. Consequently, for any given experiment one can always find $d\theta$ small enough to give rise to zero sensitivity (as illustrated

[3] We can think of ω as representing all the random variables in the DEDS. In simulation, it is a sequence of independent samples from the uniform distribution on [0, 1].

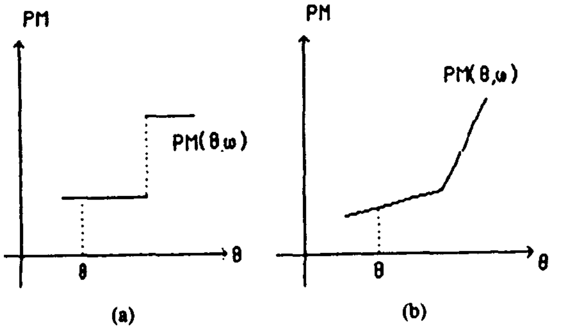

Fig. 1.

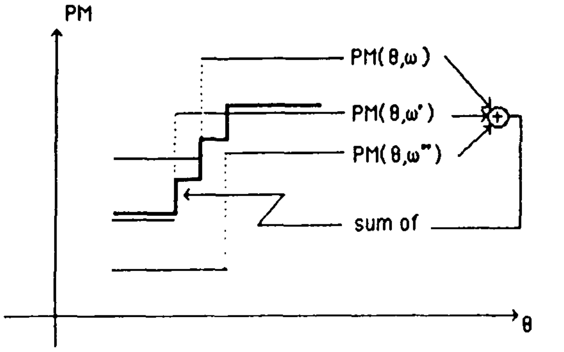

Fig. 2.

in Fig. 1(a). On the other hand, for any given $d\theta$, there always exists ω such that two busy periods will coalesce and produce a discontinuity in the sample performance measure. Again Fig. 2 is generic to this case.

Thus, the problem is how can we obtain information about $PM(\theta)$ by only looking at individual sample paths in the small and without reconstruction of the entire perturbed sample path. Criticisms of the early works on PA rest entirely on this premise [56]. This was actually well understood [24], [29] and PA has developed a three pronged attack which led to some very interesting and challenging theoretical problems in DEDS.

A. Discontinuities and Interchange of Expectation and Derivatives

Perturbation analysis calculates the average value of the sample derivatives of some *PM* with respect to a parameter θ. On the other hand, we are interested in the derivative of the expected value of the *PM* w.r.t. θ. Thus, there is the question as to when are we justified in interchanging the operations of averaging and differentiation, i.e.,

$$dE[PM(\theta, \omega)]/d\theta = ? = E[dPM(\theta, \omega)/d\theta]. \qquad (3.2)$$

Note the left-hand side of (3.2) is typically approximated by $\Sigma_{k=1}^{n}$ $[PM(\theta + \Delta\theta, \omega_k) - \Sigma_{k=1}^{n} PM(\theta, \omega_k)]/\Delta\theta$ via the brute force simulation of the system at two different values of the system parameter. On the other hand the right-hand side is obtained via a *single* Monte Carlo simulation using PA in the manner of Fig. 2. As we can see, these need not be equal. Mathematically, a sufficient condition to ensure the validity of the interchange is the dominated convergence theorem which in simple cases often requires $PM(\theta, \omega)$ to be smooth w.r.t. θ. The detail condition under which this interchange can be effected is addressed in [24] in general terms. The idea is that if for small perturbation in θ, *both* the probability of encountering a discontinuity in $PM(\theta, \omega)$ and the value of the discontinuity are sufficiently small, then such sample paths do not contribute substantially to the averaging process and (3.2) is valid. This may occur, for example, for cases illustrated in Fig. 1(b). Here the error resulting from ignoring event order change is small as the discontinuity is only in the slope of the $PM(\theta, \omega)$ curve. We can get a glimpse of this idea by the following illustration given in Fig. 3. We consider the introduction of a given perturbation of size "delta" into one of the service times of a $G/G/1$ queue. It is clear that the effect of this perturbation has ever decreasing influence on the events in succeeding busy periods. Thus, if we use infinitesimal PA rules which ignore propagation effects across busy periods to compute, say, the throughput sensitivity, then heuristically we can see that the right answer may still be obtained even though on a vanishingly small (as $\Delta\theta \rightarrow 0$) percentage of the time the rule will make errors which also are small in magnitude. On the other hand, the case pictured in Figs. 1(a) and 2 is where the discontinuities are the only things of interest in the averaging process. In such cases PA calculation using only the local information of $PM(\theta, \omega)$ does not give correct information about the slope. In general, the conditions in [24] are not easy to check explicitly. So far, the validity of (3.2) must be established separately for each class of problem. Table I lists what has been proved.

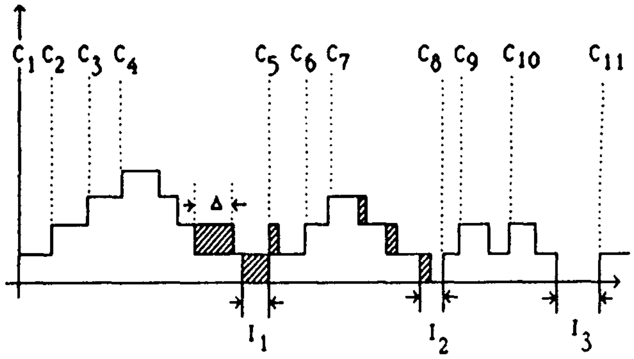

Fig. 3. Introducing a single perturbation in the sample path of a $G/G/1$ queue. C_i = ith customer arrival in the sample path.

- Change in the response time of C_3 and C_4: Δ.
- Change in the response time of C_5, C_6, C_7: max $(0, \Delta - I_1)$.
- Change in the response time of C_8, C_9, C_{10}: max $(0, \Delta - I_1 - I_2)$.
- Change in the response time of C_{11}: max $(0, \Delta - I_1 - I_2 - I_3)$.

B. Extensions of Perturbation Analysis

Of course, in many situations of interest (3.2) simply is not true. In such cases, different approaches are necessary. Two approaches will be outlined below.

1) Extended Perturbation Analysis: The basic idea of PA for DEDS is the reconstruction of an arbitrary perturbed sample path from a nominal path. Under deterministic similarity, the simple infinitesimal perturbation analysis (IPA) rules described in (3.1) apply, and the computation of perturbation propagations is easy since the "critical timing path" or the "future event schedule" between the nominal and perturbed paths remains the same. However, as pointed out before in the limit of path of very long duration or experiment with very large ensembles, deterministic similarity will always be violated. In such cases, the IPA rule, which ignores the order changes of events, gives strongly consistent estimates for performance gradients of only a limited class of DEDS as shown above. In other cases, the key question is "*how can one reconstruct the perturbed path short of a separate new simulation/experiment?*" We now suggest an alternative which we believe to be much more efficient than brute force reconstruction. As a byproduct of the new idea, we extend the applicability of IPA rules to situations where it has been thought to be unworkable.

To first get an intuitive idea of our approach, let us consider Markov systems in steady state. For such a system, a sample path or trajectory can be characterized by a sequence of "states" which the system goes through as it evolves in time. While the time duration between state changes are important for perform-

TABLE I
EXISTING RESULTS ON EXACTNESS OF PA

PM θ	Throughput	Waiting Time
arrival rate	G/G/1 [21]	G/G/1 [21]
service rate	G/G/1 [21]	G/G/1 [21]
service rate	Single class Jackson [31] Networks Tandem Network with Blocking [35]	Single class Jackson [31,34] Networks

State Sequence

ω_1: $s_a, s_g, s_h, s_u, s_z, s_y, s_p, s_a, s_r, s_k, s_e, s_r, s_l, s_a, s_u, s_j, s_y, s_u, s_c, s_h, s_l, s_h, s_o, \cdot$

ω_2: $s_e, s_n, s_p, s_u, s_t, s_o, s_m, s_g, s_z, s_d, s_j, s_l, s_l, s_m, s_a, s_e, s_s, s_o, s_p, s_h, s_l s_a,$

Fig. 4. State sequences of a Markov DEDS.

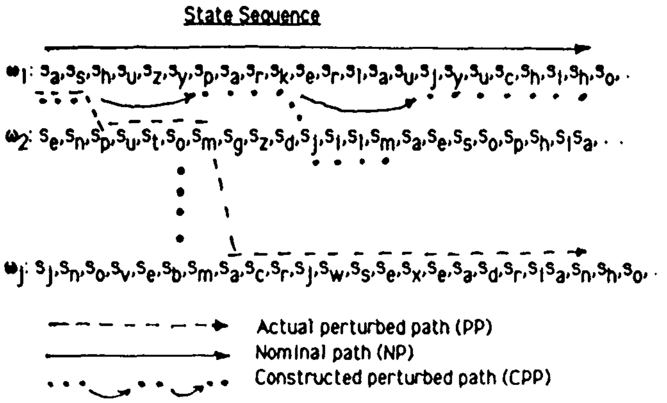
State Sequence
ω1: sa,ss,sh,su,sz,sy,sp,sa,sr,sk,se,sr,sl,sa,su,sj,sy,su,sc,sh,sl,sh,so, ·
ω2: se,sn,sp,su,st,so,sm,sg,sz,sd,sj,sl,sl,sm,sa,se,ss,so,sp,sh,sl sa, · ·
ωj: sj,sn,so,sv,se,sb,sm,sa,sc,sr,sj,sw,ss,se,sx,se,sa,sd,sr,sl sa,sn,sh,so, ·
Actual perturbed path (PP)
Nominal path (NP)
Constructed perturbed path (CPP)

Fig. 5. NP, PP, and CPP.

ance evaluation purposes, we need not be concerned with these values for the moment but concentrate only on the state sequences. Deterministic similarity between two sample paths for Markov systems is equivalent to identical state sequences. Fig. 4 shows two such random state sequences for a typical Markov DEDS.

The crucial point to be made here is that arbitrary interchange of partial state sequences beginning with the same state will leave the underlying stochastic properties of the DEDS invariant. In other words, one can generate a legitimate sample path by "cut and paste" partial state sequences as long as we restrict the interchange to sequences that began with the same state. This fact is clearly guaranteed by the Markov and ergodicity properties of the sequences and is known as "coupling" [67], [69]. Now let us see how can we exploit this coupling notion for perturbation analysis purposes. We know that IPA rules apply as long as the event order sequence between the nominal and perturbed sample path (NP and PP) remain deterministically similar (and hence, the nominal and the perturbed state sequence remain identical). When an event order change occurs, the state sequences of NP and 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). Suppose, for example, the state sequence jumps from s_s on ω_1 to s_p on ω_2 instead of s_h on ω_1 as illustrated in Fig. 5. Subsequent perturbations involving state changes may cause further deviations so that a perturbed path could be made up from segments of state sequences from $\omega_1, \omega_2, \cdots, \omega_j, \cdots$.

The central question for PA is simply how to reconstruct such a perturbed path (PP) from the information furnished by the nominal path (NP), ω_1, alone. We submit that using the idea of "coupling" an equivalent perturbed path can be constructed from the nominal sample path ω_1 by connecting s_s with s_p with $s_j, \cdots$, etc, as they occur on ω_1. Along each segment of this equivalent constructed perturbed path (CPP), IPA rules apply since by construction the NP and CPP are deterministically similar on each segment of the CPP. On the other hand, the CPP and PP are "stochastically similar" by the notion of *coupling*. To establish the connection between different segments of the CPP we use any finite PA rule to establish the state sequence jump. Such a finite PA rule always exists since we can in the extreme employ brute force reconstruction for a very short interval of time. The trick is not to extrapolate the perturbed path indefinitely into the future.

Computationally, we visualize the new method this way. Imagine the following thought experiment. We begin observing a DEDS and its identical twin starting from some initial state $x(0)$ [in the sense of an identical simulation experiment using the same random seed and same $x(0)$]. The identical twin evolves in exactly the same way as the nominal DEDS except for the fact that it has a small perturbation in one of its parameters θ. It is clear that initially the two systems will evolve in a deterministically similar way. During this time, simple rules of IPA apply. However, sooner or later at some transition time, $t_{\theta 1}$ (henceforth, we will use subscript "θ" and "θ'" to denote quantities in the nominal and perturbed path, respectively), an event order change (compared to the NP) may occur in the PP due to accumulated perturbations. After this order changes, the perturbed system may enter into a different "state" than the nominal system. This different state and the immediate resultant perturbations can be predicted exactly by the so-called finite perturbation analysis [9] rules. Let us call this different perturbed state $x_{\theta'}(t_{\theta' 1})$ as distinguished from the state on nominal $x_\theta(t_{\theta 1})$, where $t_{\theta' 1}$ denotes the perturbed time, i.e., $t_{\theta' 1} = t_{\theta 1} + \Delta t_{\theta 1}$. From this point onward, we can no longer use the nominal path to track the evolution of the perturbed path. What we can do is observe the nominal path until at some time $\tau_{\theta 1} > t_{\theta 1}$ when the nominal state $x_\theta(\tau_{\theta 1}) = x_{\theta'}(t_{\theta' 1})$. This will occur as long as the transition probability matrix of the Markovian system is irreducible. At that point, we can once again use the nominal path to track the evolution of the perturbed system starting from $t_{\theta' 1}$ with $x_{\theta'}(t_{\theta' 1})$. The reasoning being that for the Markovian system, the present state determines the future independent of the past. Hence, from a statistical point of view, the evolution of nominal path from $\tau_{\theta 1}$ with state $x_\theta(\tau_{\theta 1})$ is statistically indistinguishable from that of the perturbed path from $t_{\theta' 1}$ with $x_{\theta'}(t_{\theta' 1})$ as long as no further event order change takes place. Thus, we can use the nominal path from $\tau_{\theta 1}$ onwards to track the perturbation along the perturbed path using IPA rules until at some time $t_{\theta 2}$ when we again observe the need for consideration of an event order change. At that time, we must again use a finite PA rule to compute the new perturbed state $x_{\theta'}(t_{\theta' 2})$, and the whole process described above repeats.

Thus, what we claim is this: By selectively discarding time segments of the nominal path and connecting the remaining disjoint segments of nominal path using finite order PA rules and the condition $x_\theta(\tau_{\theta i}) = x_{\theta'}(t_{\theta' i})$, we can reconstruct a perturbed path which is statistically similar to what could have been generated if we had decided to reconstruct the perturbed path by brute force. This constructed perturbed path (CPP) is illustrated in Fig. 5. Given this statistically equivalent CPP, gradient information can be calculated in an obvious way. We submit that this approach is computationally more efficient than brute force reconstruction of the PP using a separate experiment particularly when "θ" is multidimensional. Segments of the NP discarded for one component of the "θ" vector in the construction of one CPP can be used for the construction of another CPP. Overall, the computational advantage of N:1 will be preserved.

This idea has many points in common with regenerative simulation where we must identify particular states along a sample path. In regenerative simulation, we often look for approximate regeneration. In other words, instead of looking for exact match or regeneration of states, we only search for approximate regeneration using trap intervals, etc., [71]. Similar approximations can be employed here. In fact, in this framework we see the finite PA rule discussed in [9] can be viewed as an extreme approximation of this idea of the extension of IPA rules. Namely, we simply ignore all requirements of a matching state and let "statistical averaging" take care of everything. Experimentally, we find even in such a drastic approximation mode, the results are highly encouraging. In fact, a very large scale statistical experiment involving random generation of systems, routes, parameters, and initial conditions shows that such crude finite perturbation analysis rules do predict sensitivities more accurately than infinitesimal rules [47]. Additional details of exact and partial matching using this approximate coupling can be found in [68]. The heuristic reasoning behind the experimental successes of partial or zero state matching can be stated as the following.

Statistical Similarity Assumption: Once a perturbation (finite or otherwise) has been introduced, the types of future interaction it may encounter and induce among customers and servers in the DEDS along the actual perturbed path is statistically similar to that of the nominal sample path.

In other words, we assume that for a small system parameter change, the change in the distribution of various interactions among customers and servers should be continuous from the nominal to the perturbed path. Even though finite or large perturbations may be generated along the sample path, the propagation of these perturbations along the nominal and the perturbed sample path will be essentially the same *on the average*. As long as we are only interested in the average PM's, we can use the nominal path to calculate the propagations. As long as we implement accurately the short-term effect of the perturbations on each and every interaction (state change), their long-term accumulated average effects are essentially the same whether we use the perturbed or the nominal path for calculation. However, a rigorous proof of this assertion seems difficult at this stage and must await further development. However, the idea discussed here and in [68] provides a framework for such an investigation of the idea of approximate coupling. There are also other cases of finite PA where only short-term reconstruction of the sample path is required [62].

2) Smoothed Infinitesimal Perturbation Analysis: The discussions in the above sections [Figs. 1 and 2, (3.2)] showed that the basic difficulty in the application of infinitesimal perturbation rules is the possible *discontinuities* in the sample performance with respect to parameter perturbations. Another simple remedy is then to smooth out these discontinuities. Mathematically, let us rewrite (3.2) as

$$dE[PM(\theta, \omega)]/d\theta = dE_z E_{/z}[PM(\theta, \omega)]/d\theta$$

$$= ? = E_z \lim_{\Delta\theta \to 0} [E_{/z}[\Delta PM(\theta, \omega)]/\Delta\theta]. \quad (3.3)$$

In other words, we decompose the "expectation" into a conditional expectation on L first followed by expectation on the conditioning variable z. We expect $E_{/z}[PM(\theta, \omega)] \equiv PM(\theta, z)$ to be smoother than $PM(\theta, \omega)$; and hence may make the interchange between E_z and $d/d\theta$ possible (e.g., instead of Fig. 1(a) we have Fig. 1(b)]. This is workable provided $PM(\theta, z)$ is computable, i.e., z must be based on data available on the sample path. Let us consider the second example discussed in the introduction of this section, that of estimating the sensitivity of the average number of customers served, $E[n]$, in a busy period with respect to mean service time s for a $G/G/1$ queue. Fig. 6 illustrates the typical situation.

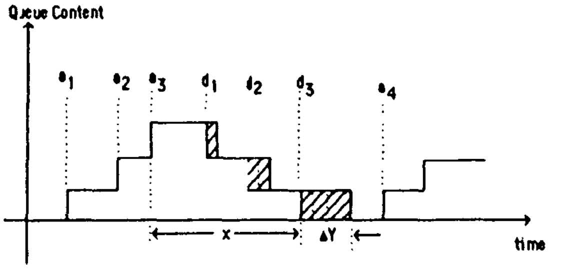

Fig. 6.

At time d_3, we have accumulated perturbation $\Delta Y = \Delta s_1 + \Delta s_2 + \Delta s_3$ representing the total change in d_3 due to mean service time change on customers C_1, C_2, and C_3. Also, x units of time have elapsed since the last arrival at time a_3. Whether or not we will encounter a discontinuity in the sample performance (or the busy period will coalesce with the next) depends on the size of ΔY and the duration of x, i.e., $z = \{\Delta Y, x\}$. The conditional expectation of the change in the number of customers served is given by the conditional probability of coalescing and the expected number of customers added due to the coalescing. Thus,

$$\Delta E[n] = E_{\Delta Y, x}\,\{\text{Prob [coalescing takes place}/\Delta Y, x]^*$$
$$E[\text{number of customers served in a busy period}]\}$$
$$= E_{\Delta Y, x}\{[g(x)/1 - G(x)]^* \Delta Y\}^* E[n] \quad (3.4)$$

where $g(x)$ and $G(x)$ are, respectively, the density and distribution function of x and $g(x)/1 - G(x)$ is the well-known hazard rate in reliability theory which is the rate failure may occur given that it has not occurred in x units. If we further specialize to the case of an $M/M/1$ queue with arrival rate λ and service rate $1/s$, then (3.4) can be explicitly evaluated to give

$$\Delta E[n]/\Delta s = [E(n)]^{2*}\lambda \quad (3.5)$$

which agrees with the well-known formula [13, p. 213]. For more details and further examples of this approach, see [49].

C. Other Approaches to Sensitivity Analysis

There are other approaches to computing sensitivities by using only one sample path; the most prominent one being the approaches developed by Reiman and Weiss [48], Rubenstein [50] and Glynn and Sanders [57]. Briefly, the idea is as follows.

Suppose for the moment that θ affects the performance measure $E[PM]$ only through the distribution of the random variables in the experiments. We write,

$$E[PM(\theta, \omega)] = \int PM(\omega)p(\theta, \omega)\, d\omega$$

$$E[PM(\theta + \Delta\theta, \omega)] = \int PM(\omega)p(\theta + \Delta\theta, \omega)\, d\omega$$

which can be rewritten as

$$= \int \{PM(\omega)[p(\theta + \Delta\theta, \omega)/p(\theta, \omega)]\}\, p(\theta, \omega)\, d\omega$$

$$= \int \{\cdots\}\, p(\theta, \omega)\, d\omega = E[\{\cdots\}]. \quad (3.6)$$

Note that (3.6) is completely general. It only requires the ability to evaluate the expression in $\{\cdots\}$ along the original sample path or experiment. In particular, we can write

$$\lim_{\Delta\theta \to 0} E[(PM(\theta + \Delta\theta, \omega)] - E[PM(\theta, \omega)])/\Delta\theta]$$
$$= \int PM(\omega)[(dp/d\theta)/p(\theta, \omega)]\, p(\theta, \omega)\, d\omega$$
$$= E[PM(\omega) d \ln p(\theta, \omega)/d\theta]. \quad (3.7)$$

For the case where $p(\theta, \omega)$ is the exponential distribution and θ the mean, we get $d \ln p(\theta, \omega)/d\theta = \theta^{-1} - \omega$ and

$$E[dPM(\omega)/d\theta] = E[PM(\omega)(\theta^{-1} - \omega)] \quad (3.8)$$

which can be evaluated along the nominal path. Note (3.7) also requires the interchange of an expectation with $d/d\theta$. However, the conditions here are generally easier to verify than in the PA case. But this approach requires that the density function of the random variables be expressed analytically. The variance of the estimate also increased with the number of random variables involved. Thus, in general, it tends to work well only with short regenerative cycles. See [23] for an analysis of this approach.

IV. Other Open Problems

Routing Probability Sensitivity

There are other ad hoc ways to overcome the problem mentioned in Section III-A. One possibility is to find another parameter θ', which is deterministically related to θ, and for which (3.2) holds. Then we can compute the sensitivity w.r.t. θ by transforming the equivalent sensitivity w.r.t. θ'. This is the case for the example with the routing probability sensitivity [25]. We showed that Δ[routing probability] can be related to Δ[visit ratio] which can be equivalently expressed as Δ[mean service time] for certain classes of queueing networks. In fact, from the discussions in Section III and above, it is tempting to conjecture that if the performance curve $PM(\theta)$ of a system is differentiable, a way can be found to use the idea of perturbation analysis to compute a consistent estimate of it. In other words, a single sufficiently long sample path should contain all relevant information about the stochastic process. The only question is how to extract the information efficiently (see the discussion in Section III-B-1 again). This is a general open problem waiting for concrete treatment. The approach of PA is basically sample path analysis with its accompanying advantage of distribution free analysis. However, one is not restricted from using probablistic arguments whenever appropriate.

Aggregation, Decomposition, and Simulation Modeling

The idea of aggregation is so fundamental to modeling and engineering analysis that we are not often aware of its existence. Certainly, to claim that $dx/dt = f(x, u, t)$ is a good model for a large class of dynamical systems implies the validity of the assumption that many real-world details can be aggregated and only certain "state" variables are sufficient for analysis and control purposes. Other examples abound in all phases of engineering design (see [77]). Furthermore, a good model satisfies in addition the requirement of "external independence," i.e., the model does not change if we connect it to other systems. Otherwise, the utility of the model is greatly diminished if we cannot use it to help analyze its effect with others. In the case of DEDS we know that the Norton theorem holds for product form networks as mentioned in Section II-B. However, it turns out [39] that the analogy breaks down in more general cases. It is no longer possible to aggregate a portion of a network (i.e., represent the portion by some simpler network) without considering what constitutes the rest of the network. In other words, the equivalent network cannot be dependent only on the part that is being aggregated. On the other hand, aggregation is a well-developed engineering concept in system design and analysis. Every system designer and analyst uses "aggregation" at all stages of the analysis whether or not he is using analytical or simulation tools. The aggregation process is not only intuitive and heuristic but often successful. More recently experimental study of the PA technique seems to indicate that "aggregation" is possible approximately for general networks [44]. However, a more insightful analysis of the approximation process is lacking at present. In fact, since we know [51, p. 30] that *any* portions of a general queueing network can be replaced by an "externally dependent" equivalent $M/M/1$ server, it would seem to offer the possibility of applying PA techniques to analyze problems in DEDS where it is not applicable if the original nonaggregated DEDS were used. Thus, in addition to computational saving in the case of simulation, aggregation may considerably extend the domain of applicability of the PA technique. Viewed in the context of "engineering modeling," there is every reason to be optimistic with respect to the study of aggregation in DEDS. Concommitant with aggregation is the concept of "decomposition" which has been extensively developed by Courtois [19]. Again this idea has its parallel in continuous variable system analysis. An excellent survey can be found in [20].

Light Traffic Theory as PA with Zero Nominal

Reiman and Simon [58] recently developed a light/heavy traffic theory approach to the determination of the performance of certain queueing networks. The idea is that performances under light (zero customer arrival rate) and heavy (100 percent utilization) conditions are often easy to determine for certain networks. In addition the derivatives of $PM(\theta)$ with θ = arrival rate under light traffic conditions can also be determined by adding one single customer to the zero rate arriving stream. Thus, given the two data points and a derivative, the entire curve $PM(\theta)$ can be estimated with remarkable accuracy. From the PA viewpoint, the light traffic derivative theory is simply a special case of PA where the nominal sample path is trivially easy to implement, namely, the path with zero customers. However, by the same token, if another path is available, then PA can be performed to obtain the derivatives at any point other than $\theta = 0$ [72]. In fact, the idea of adding a customer to a sample path goes as far back as the 1977 thesis of Bello [59]. Suri and Cao have also employed a similar idea in their marked and phantom customer approach to PA [60]. This approach to the determination of the entire $PM(\theta)$ curve appears to be very promising and intriguing, particularly in the case when θ is multidimensional.

Sample Path and Operational Analysis Approach to DEDS

Both PA and operational analysis are sample path based techniques for the study of DEDS. There are many points of similarity between the two approaches. In fact, results of PA can be derived from an operational analysis framework [32]. On a heuristic basis, engineers have routinely used simulation often without much statistical justification in the design and optimization of DEDS. It is natural to ask the question: "Does there exist a nonprobabilistic or experimental approach to the control and optimization of DEDS?" Various evidences such as "simulated annhealing," "generalized stochastic approximation," etc., suggest that this is not necessarily an outlandish suggestion.

V. Conclusion

Perturbation analysis is less a specific technique but more a state of mind or a framework for the study of DEDS in the time domain. It complements but does not replace the probabilistic approach which has dominated the study of queueing networks for the past two decades. It is also a very natural tool for the system engineer. The ultimate goal for the system engineers/theorists is the dynamic optimal stochastic control of DEDS. As mentioned earlier in the Introduction, we see the entire expanse of traditional control theoretic concepts and techniques awaiting parallel developments in DEDS. We are at the dawn of a new era of control system analysis and synthesis.

References

[1] J. Cassidy, T. Z. Chu, M. Kutcher, S. Gershwin, and Y. C. Ho, Eds., "IEEE task force report on research needs in manufacturing systems," *IEEE Contr. Syst. Mag.*, Aug. 1985.

[2] D. Yao and J. Buzacott, "Modeling the performance of flexible manufacturing systems," *Int. J. Production Res.*, 1985.

[3] J. D. Henriksen, "The development of GPSS-85 simulation language," in *Proc. 18th Annual Simulation Symp.*, Mar. 1985.

[4] Y. C. Ho, "On the perturbation analysis of discrete event dynamic systems," *J. Optimiz. Theory Appl.*, vol. 46, pp. 535-554, Aug. 1985.

[5] Y. C. Ho, "Is it applications or is it experimental science?" Editorial, *IEEE Trans. Automat. Contr.*, vol. AC-27, Dec. 1982.

[6] R. M. Hogarth, "Why bother with experiments?" presented at the NSF Conf. Generalization, Boulder, CO, June 1985.

[7] J. Banks and J. S. Carson, *Discrete Event System Simulation.* Englewood Cliffs, NJ: Prentice-Hall, 1984.

[8] R. E. Kalman, "Mathematical description of linear dynamical systems," *SIAM J. Contr.*, vol. 1, pp. 152-192, 1963.

[9] Y. C. Ho, X. Cao, and C. Cassandras, "Finite and infinitesimal perturbation analysis."

[10] S. Stidham, "Optimal control of admission to a queueing system," *IEEE Trans. Automat. Contr.*, vol. AC-30, pp. 705-713, Aug. 1985.

[11] J. A. Buzacott and D. D. Yao, "Flexible manufacturing systems: A review of analytical models," *Management Sci.*, to be published.

[12] B. W. Stuck and E. Arthurs, *A Computer and Communication Network Performance Analysis Primer.* Englewood Cliffs, NJ: Prentice-Hall, 1984.

[13] L. Kleinrock, *Queueing Systems, Vol. I: Theory.* New York: Wiley, 1975.

[14] F. Basket, K. M. Chandy, R. R. Muntz, and F. G. Palacios, "Open, closed, and mixed network with different classes of customers," *J. Ass. Comput. Mach.*, vol. 22, pp. 248-260, Apr. 1975.

[15] P. Denning and J. Buzen, "The operational analysis of queueing network models," *Comput. Survey*, vol. 10, pp. 225-261, Sept. 1978.

[16] M. Reiser and S. S. Lavenberg, "Mean value analysis of closed multichain queueing networks," *J. Ass. Comput. Mach.*, vol. 27, pp. 313-322, Apr. 1980.

[17] R. Suri, "Robustness of queueing network formulas," *J. Ass. Comput. Mach.*, vol. 30, pp. 564-594, July 1983.

[18] J. Walrand, "Filtering formulas and the ./M/1 queue in a reversible network," *Stochastics*, vol. 6, pp. 1-22, 1981.

[19] P. J. Courtois, *Decomposability:Queueing and Computer System Applications.* New York: Academic, 1977.

[20] ——, "On time and space decomposition of complex structures," *Commun. Ass. Comput. Mach.*, vol. 28, pp. 590-603, 1985.

[21] R. Suri and M. Zazanis, "Perturbation analysis gives strongly consistent sensitivity estimates for the M/G/1 queue," *Management Sci.*, 1986.

[22] X. Cao and Y. C. Ho, "Perturbation analysis of sojourn time in queueing networks," presented at the IEEE Conf. Decision Contr., San Antonio, TX, Dec. 1983; also in *J. Opt. Theory Appl.*, 1987.

[23] X. Cao, "Sensitivity estimators based on one realization of stochastic system," *J. Statist. Computation Simulation*, 1986.

[24] ——, "Convergence of parameter sensitivity estimates in a stochastic environment," in *Proc. IEEE Conf. Decision Contr.*; also in *IEEE Trans. Automat. Contr.*, vol. AC-30, no. 9, pp. 845-885, 1985.

[25] Y. C. Ho and X. Cao, "Performance sensitivity to routing changes in queueing networks and flexible manufacturing systems using perturbation analysis," *IEEE J. Robotics Automat.*, 1986.

[26] C. M. Woodside, "Response time sensitivity measurement for computer systems and general closed queueing networks," *J. Performance Evaluation*, vol. 4, pp. 199-210, 1984.

[27] M. Zazanis and R. Suri, "Comparison of perturbation analysis with conventional sensitivity estimates for regenerative stochastic systems," Harvard Univ., Dec. 1984; and *Oper. Res.*, submitted for publication.

[28] ——, "Estimating second derivatives of performance measure for G/G/1 queues using a single sample path," Harvard Univ., Cambridge, MA, Apr. 1985.

[29] X. Cao, "First order perturbation analysis of a simple multiclass finite source queue," in *Proc. 1985 IEEE Conf. Decision Contr.*; also in *Performance Evaluation*, submitted for publication.

[30] R. Rubenstein and T. Szidarovszky, "Convergence of perturbation analysis estimates for queueing networks: A general approach," *IEEE Trans. Automat. Contr* , submitted for publication.

[31] X. Cao, "On the sample functions of queueing networks with application to perturbation analysis," *Oper. Res.*, submitted for publication.

[32] X. Cao and Y. Dallery, "An operational approach to perturbation analysis of closed queueing networks," Harvard Univ., Cambridge, MA; also in *Proc. 1986 ACC Conf.*

[33] X. Cao and Y. C. Ho, "Perturbation analysis of sojourn times in queueing networks," in *Proc. IEEE Conf. Decision Contr.*, San Antonio, TX, 1983; revised version in *J. Optimiz. Theory Appl.*, 1987.

[34] X. Cao and Y. C. Ho, "Perturbation analysis of sojourn time in closed Jackson networks," *Oper. Res.*, submitted for publication.

[35] Y. Bard, "Some extensions to multiclass queueing network analysis," in *Performance of Computer Systems*, M. Arato, Ed. Amsterdam, The Netherlands: North-Holland, 1979.

[36] P. Schweitzer, "Approximate analysis of multiclass closed network of queues," in *Proc. Int. Conf. Stochastic Contr. Optimiz.*, Amsterdam, 1979.

[37] A. Lazar, "Optimal flow control of a class of queueing networks in equilibrium," *IEEE Trans. Automat. Contr.*, vol. AC-28, pp. 1001-1007, 1983.

[38] A. Brandwajn, "Equivalence and decomposition in queueing systems—A unified approach," *Performance Evaluation*, vol. 5, pp. 175-186, 1985.

[39] X. Cao and Y. C. Ho, "Sensitivity estimator and optimization of production line with blocking," *IEEE Trans. Automat. Contr.*, submitted for publication.

[40] W. Whitt, "Continuity of generalized semi-Markov processes," *Math. Oper. Res.*, vol. 5, pp. 494-501, 1980.

[41] R. Schassberger, "On the equilibrium distribution of a class of finite state generalized semi-Markov processes," *Math. Oper. Res.*, vol. 1, pp. 395-400, 1986.

[42] Y. C. Ho and W. B. Gong, "A note on filtering in queueing networks and discrete event dynamic systems," *IEEE Trans. Automat. Contr.*, vol. AC-31, July 1986.

[43] K. M. Chandy, U. Herzog, and L. Woo, "Parameteric analysis of queueing networks," *IBM J. Res. Develop.*, vol. 19, pp. 36-42, 1975.

[44] Y. C. Ho and P. Q. Yang, "Equivalent networks, load dependent servers, and perturbation analysis—An experimental study," *Teletraffic Analysis and Computer Performance Evaluation*, O. J. Boxma, J. W. Cohenm, H. C. Tijms, Eds. Amsterdam, The Netherlands: North-Holland, 1986.

[45] R. K. Boel and J. H. Van Schuppen, "Overload control for switches of communication systems—A two phase model for call request processing," in *Teletraffic Analysis and Computer Performance Evaluation*, O. J. Boxma, J. W. Cohenm, H. C. Tijms, Eds. Amsterdam, The Netherlands: North-Holland, 1986.

[46] R. Suri, "Infinitesimal perturbation analysis of discrete event dynamic systems—A general theory," in *Proc. 22nd IEEE Conf. Decision and Contr.*, San Antonio, TX, pp. 1030-1039.

[47] Y. C. Ho and J. Dille, "Large scale statistical study of infinitesimal and finite perturbation analysis," presented at the TIMS/ORSA National Meet., Los Angeles, CA.

[48] M. I. Reiman and A. Weiss, "Sensitivity analysis for simulations via likelihood ratiors," Preprint 1986.

[49] W. B. Gong and Y. C. Ho, "Smoothed perturbation analysis of discrete event dynamic systems," *IEEE Trans. Automat. Contr.*, to be published.

[50] R. Y. Rubinstein, "Sensitivity analysis and performance extrapolation for computer simulation models," Harvard Univ., Cambridge, MA, 1986.

[51] Bremaud, *Point Processes and Queues—A Martingale Approach,* (Springer-Verlag Series in Statistics). New York: Springer-Verlag, 1981.

[52] R. Boel and P. Varaiya, "Optimal control of jump processes," *SIAM J. Contr.*, vol. 15, pp. 92-119, 1977.

[53] R. Boel, P. Varaiya, and E. Wong, "Martingales on jump processes; Part 1 representation results, Part 2 applications," *SIAM J. Contr.*, vol. 13, pp. 999-1061, 1977.

[54] P. Varaiya and J. Walrand, "Flows in queueing networks, A Martingale approach," *Math. Oper. Res.*

[55] J. Van Schuppen, "Filtering, prediction and smoothing for counting process observations, A Martingale approach," *SIAM J. Appl. Math.*, vol. 32, pp. 552-570, 1977.

[56] P. Heidelberger, "Limitations of infinitesimal perturbation analysis," IBM Res. Rep. RC 11891 5/20/86.

[57] P. Glynn and J. L. Sanders, "Monte Carlo optimization in manufacturing systems: Two new approaches," in *Proc. ASMECIE Conf.*, Chicago, IL, 1986.

[58] M. I. Reiman and B. Simon, "An interpolation approximation for queueing systems with Poisson input," *Oper. Res.*, submitted for publication.

[59] M. Bello, "The estimation of delay gradient for purpose of routing in data communication networks," S.M. thesis, Dep. Elec. Eng., Mass. Inst. Technol., Cambridge, 1977.

[60] R. Suri and X. Cao, "The marked and phantom customer method for optimization of closed queueing networks with blocking and general service times," *ACM Performance Evaluation Rev.*, vol. 12, pp. 243-256, 1983.

[61] P. Whittle, "Equilibrium distribution for an open migration process," *J. Appl. Prob.*, vol. 5, pp. 567-571, 1968.

[62] C. G. Cassandras, "Sensitivity analysis of a simple routing strategy," in *Proc. 1985 IEEE Conf. Decision Contr.*, 1985.

[63] A. Lazar and M-T. Hsiao, "Network & user optimal flow control with decentralized information," Commun. Res. Center, Columbia Univ., New York, NY, 1986.

[64] P. J. Ramadge and W. M. Wonham, "Supervision of discrete event processes," in *Proc. 1982 IEEE Conf. Decision Contr.*, 1982, pp. 1228-1229.

[65] G. Cohen, D. Dubois, J. P. Quadrat, and M. Voit, "A linear-system-theoretic view of discrete event process and its use for performance evaluation in manufacturing," *IEEE Trans. Automat. Contr.*, vol. AC-30, pp. 210-220, 1985.

[66] G. Olsder, "Some results on minimal realization of discrete event dynamic systems," Dep. Math. Inform., Delft Inst. Technol., Delft, Tech. Rep. 85-36, 1985.

[67] S. M. Ross, *Stochastic Processes.* New York: Wiley, 1983.

[68] Y. C. Ho and S. Li, "Extensions of the perturbation analysis techniques for discrete event dynamic systems," submitted for publication.

[69] E. Cinlar, *Introduction to Stochastic Processes.* Englewood Cliffs, NJ: Prentice-Hall, 1975.

[70] P. J. Ramage, "Control and supervision of discrete event processes," Ph.D. dissertation, University of Toronto, Toronto, Ont., Canada, 1983.

[71] M. A. Crane and A. J. Lemoine, *An Introduction to the Regenerative Method for Simulation Analysis.* New York: Springer-Verlag, 1977.

[72] B. Simon, "Light traffic perturbation of queueing systems," *Oper. Res.*, submitted for publication.

[73] J. J. Solberg, "A mathematical model of computerized manufacturing systems," in *Proc. 4th Int. Conf. Production Res.*, Tokyo, Japan, 1977.

[74] R. Suri and R. Hildebrand, "Modeling flexible manufacturing systems using the mean value analysis," *J. Mfg. Syst.*, vol. 3, no. 1, pp. 27–38, 1984.

[75] S. Gershwin, R. Hidlebrand, R. Suri, and S. Mitter, "A control theorist's perspective on recent trends in manufacturing systems," in *Proc. 1984 Decision Contr. Conf.*, Las Vegas, NV, 1984.

[76] G. Shantikumar and J. Buzacott, "Open queueing network models of dynamic jobshop," *Int. J. Production Res.*, vol. 19, pp. 255-266, 1981.

[77] B. Zeigler, *Theory of Modelling and Simulation.* New York: Wiley, 1976.

[78] R. F. Garzia, M. R. Garzia, and B. P. Zeigler, "Discrete event simulation," *IEEE Spectrum*, pp. 32-36, Dec. 1986.

Yu-Chi Ho (S'54-M'55-SM'62-F'73) received the S.B. and S.M. degrees from the Massachusetts Institute of Technology, Cambridge, and the Ph.D. degree from Harvard University, Cambridge, MA.

Except for three years in industry, he has been with Harvard University, where he is currently Gordon McKay Professor of Engineering and Applied Mathematics and Chairman of the Committee on Higher Degrees in Business Studies.

Dr. Ho was a Guggenheim Fellow in 1970 and USA-USSR Senior Exchange Fellow in 1973. He was elected to distinguished membership of the IEEE Control Systems Society in 1985 and to membership in the National Academy of Engineering in 1987.