A Comparison of the Dynamics of Continuous and Discrete Event Systems

XI-REN CAO, MEMBER, IEEE

Invited Paper

Perturbation analysis of discrete event dynamic systems (DEDSs) is a new technique which provides estimates of the sensitivity of system performance based on one sample path of a system. This paper reviews the results of perturbation analysis pertaining to closed queueing networks, and shows that the basic principles of perturbation analysis can be explained using a dynamic point of view. It is shown that perturbation generation and perturbation propagation rules can be viewed as counterparts of the linearization theory of nonlinear continuous variable dynamic systems; the concept of perturbation realization describes the steady-state effect of a perturbation, and reflects the special dynamic feature of a closed queueing network. The realization probability can be used to calculate the sensitivity of steady-state throughputs and some other sensitivities. Exploring the dynamic properties of DEDSs opens a new area for system analysts.

I. INTRODUCTION

Discrete event systems such as queueing networks can be viewed as dynamic systems. Based on this point of view, a new technique, called perturbation analysis of discrete event dynamic systems (DEDSs), has recently been developed. By utilizing the dynamic feature of a DEDS, this technique provides an efficient method of estimating the performance sensitivity.

The basic idea of perturbation analysis is that, for any given sample path of a DEDS with a given parameter $\theta$, we can construct a corresponding sample path of the DEDS with a slightly perturbed parameter $\theta + \Delta \theta$. This constructed sample path is called a perturbed sample path. Based on the perturbed path, we can further calculate the performance of the DEDS with the perturbed parameter $\theta + \Delta \theta$, and hence obtain an estimate for the sensitivity of the performance with respect to $\theta$. The analysis reveals an intrinsic fact of DEDSs: A sample path of a DEDS may contain not only the information of the performance of the system, but also that of the sensitivity of the performance. This fact is of importance in practice: To estimate performance sensitivity, we only need one simulation or observation of a real system. This is possible only by utilizing the dynamic features of the system.

Perturbation analysis is a successful example of applying the concepts and techniques for continuous dynamic systems to DEDSs. Ho [9] provides a nice conceptual survey of perturbation analysis and outlines some open problems in this area. The purpose of this paper is to introduce perturbation analysis by comparing the dynamics of a queueing network with that of a continuous variable dynamic system. We shall review the three main concepts of perturbation analysis: perturbation generation, perturbation propagation, and perturbation realization. These concepts will be explained by using the dynamic point of view. The first two concepts can be viewed as the counterparts of the linearization theory of a nonlinear dynamic system, and are used in constructing a perturbed path. The last concept, perturbation realization, is special to closed queueing networks. It reflects the strong interaction between servers in a closed network. Mathematical formulas for performance sensitivities based on perturbation realization will also be reviewed and compared with the formulas for continuous dynamic systems.

II. DYNAMICS OF A CONTINUOUS VARIABLE SYSTEM

Consider a linear continuous dynamic system described by the following differential equation:

$$\dot{x}(t) = A(t)x(t) + B(t)u(t)$$

(1)

where $x(t)$ is an $n$-dimensional state variable, and $u(t)$ is an $m$-dimensional control variable. Let $x_0$ be the initial state of the system at time $t = 0$, then

$$x(t) = \phi(t, 0)x_0 + \int_0^t \phi(t, s)B(s)u(s)\, ds$$

(2)

where $\phi(t, s)$ is the state transition function of the system.

The dynamic property of linear systems can be used to determine the effect of small perturbations in a system parameter. Consider a nonlinear system

$$\dot{x} = f(x, \theta(t), t)$$

(3)

where $\theta(t)$ is a time varying parameter. Suppose that $\theta(t)$ changes to $\theta(t) + \Delta \theta(t)$, $|\Delta \theta(t)|/\theta(t) \ll 1$. Then

$$\Delta \dot{x} = \frac{\partial}{\partial x} \{ f(x, \theta(t), t) \} \Delta x + \frac{\partial}{\partial \theta} \{ f(x, \theta(t), t) \} \Delta \theta(t)$$

(4)
The higher order terms of $\Delta x$ and $\Delta \theta$ are omitted in the above equation. This equation is also known as the linearization theory of a nonlinear system.

Assuming $\Delta x_0 = 0$ and applying (2) to the linear equation (4) we have

$$\Delta x(t) = \int_0^t \phi(t, s) \left[ \frac{\partial f(x, \theta(s), s)}{\partial \theta} \Delta \theta(s) \right] ds$$

(5)

where $\phi(t, s)$ is the state transition function corresponding to $\partial f(x, \theta(t), t)$.

Let

$$d\theta(s) = \frac{\partial f(x, \theta(s), s)}{\partial \theta} \Delta \theta(s) ds.$$  

(6)

Then (5) can be rewritten as the following form:

$$\Delta x(t) = \int_0^t \phi(t, s) d\theta(s).$$  

(7)

The equation has the following intuitive explanation: $d\theta(s)$ is the perturbation generated in $[s, s + ds]$ because of the change of $\Delta \theta(s)$; $\phi(t, s) d\theta(s)$ is the effect of $d\theta(s)$ to $\Delta x(t)$; finally, $\Delta x(t)$ is the sum of all these effects in interval $[0, t]$.

Let the performance index for the dynamic system (3) be

$$J = h[x(T)] + \int_0^T c(x, \theta, t) dt.$$  

(8)

The change in performance because of the change in $\theta(t)$, $\Delta \theta(t)$, is

$$\Delta J = \frac{\partial h}{\partial x} \Delta x(T) + \int_0^T \left( \frac{\partial c}{\partial x} \Delta x + \frac{\partial c}{\partial \theta} \Delta \theta(t) \right) dt.$$  

(9)

Therefore, the performance sensitivity $\Delta J/\Delta \theta$ can be obtained using (7) and (9). Note that in this method it is not required to solve the nonlinear differential equation (3) for $\theta$ and $\Delta \theta$.

Discrete event dynamic systems possess similar dynamic properties as those described above for continuous systems. Of course, because of the specific nature of a DEDS, such as the randomness involved, and the strong interconnection between parts and resources, there are some inherent properties associated with each individual DEDS. These properties are distinguishable from that of continuous variable systems. As an example, in the next section we shall study a closed queueing network with single-class customers and single-server nodes.

III. DYNAMICS OF A CLOSED JACKSON NETWORK

One common feature of DEDSs is the existence of stochastic disturbance and unexpected events. These stochastic phenomena can be characterized by random variables. In this paper, a DEDS is represented by the pair $(\theta, \xi)$, where $\theta \in \mathbb{R}^n$ is an $n$-dimensional vector of parameters of the system, $\xi$ is an $\mathbb{R}^m$ valued random vector defined on a probability space $(\Omega, \mathcal{F}, \mathbb{P})$. The performance measure of the system is defined as a function $J(\theta, \xi): \mathbb{R}^n \times \mathbb{R}^m \rightarrow \mathbb{R}$. For any $\theta$, $\mathbb{F} \in \mathcal{F}$ is a random variable on $(\Omega, \mathcal{F}, \mathbb{P})$. For convenience, we assume $\theta$ is a scalar.

Consider a closed Jackson queueing network containing $M$ single-server nodes and $N$ single-class customers. Each server has a buffer with an infinite size. The service discipline is first come, first served. The customers circulate among servers, and at the completion of its service at server $i$, a customer goes to server $j$ with probability $q_{ij}$. The service time required by a customer at server $i$ is exponentially distributed with mean $\lambda_i = 1/\mu$. The service time of the $k$th customer at server $i$ (counting from time $t = 0$) can be expressed as

$$s_{ik} = -\ln(1 - r_k) / \lambda_i,$$

(10)

where $r_k$ are independent random variables uniformly distributed over $[0, 1]$. The system parameters are $\lambda_i$ and $q_{ij}$, $i, j = 1, 2, \cdots, M$.

The random effect involved can be represented by the random vector $\xi = (\xi_1, \xi_2, \cdots, \xi_M, \xi_{M+1}, \cdots, \xi_{M+k})$, where $\eta_i = \{r_1, r_2, \cdots\}$ is a sequence of random numbers which determine the destinations of customers in server $i$, and $\xi$ is a random vector representing the initial condition, i.e., the numbers of customers in each server at $t = 0$.

We observe the system until one server, for example server $u$, has served $g$ customers. Let $g_i$ be the number of customers served by server $i$ in the observation period, let $L = \sum_{i=1}^M g_i$ be the total number of customers served by all servers in the system. Note that $g_i = g$. We choose the length of this observation period, $T_i(\theta, \xi)$, as the system performance measure.

Any value of $(\theta, \xi)$ determines a sample path of the system. Fig. 1 shows an example of a pattern of such a sample path. The arrows in the figure indicate the customer transitions between servers. The transition instants are denoted by $T_i = 1, 2, \cdots, L$. The system state can be represented by $n = (n_1, n_2, \cdots, n_M)$, where $n_i$ is the number of customers in queue $i$ including the customer being served. The numbers above each line in Fig. 1 denote these $n_i$'s. The system can be described by a pure-jump right-continuous Markov process $\mathcal{S}(\xi, \theta)$ representing the system state at time $t$. Each time a customer transfers from one server to the other, the system changes its state. Let $S_t = S_{t-1} + T_i$ be the time duration the system remains in its $i$th state. The sample path goes through a total of $L$ states in the observation period.

Let $t_{i, k}$ be the service completion time of $k$th customer of server $i$ since the observation starts (i.e., $t = 0$). Note that the set $\{T_i, i = 1, 2, \cdots\}$ is just the same as the set $\{t_{i, k}, i = 1, 2, \cdots, M, k = 1, 2, \cdots\}$. A sample path is completely determined by the set $\{t_{i, k}, i = 1, 2, \cdots, M, k = 1, 2, \cdots\}$, the initial state, and the customer transitions. The idle period of server $i$ (i.e., during which $n_i = 0$) plays a dominant role on the dynamics of the queueing network. It represents the interaction between servers. Suppose that server $i$ experiences an idle period after its $(k - 1)$th customer leaves it; then the number of customers in server $i$ at $t_{i, k-1} + \tau = n_i(t_{i, k-1} + \tau) = 0$. If, after this idle period, server $i$...
receives a customer from server \( j \) (who was the \( h \)th customer of server \( j \)) to start a new busy period, we say that the idle period is terminated by the \( h \)th customer of server \( j \). The following recursive formula holds:

\[
t_{i,k} = \begin{cases} 
  t_{i,k-1} + s_{i,k} & \text{if } n(t_{i,k-1}^+) \neq 0, \\
  t_{i,h} + s_{i,k} & \text{if } n(t_{i,k-1}^+) = 0.
\end{cases}
\]

This equation simply says that if a customer starts a new busy period, then its service starting time is decided by its arrival time to the server \( t_{i,k} \); otherwise it is decided by the service completion time of the previous customer.

The recursive equation (11) and (10) can be viewed as the counterpart of the dynamic equation (3) for continuous variable systems. Note that the random variables \( t_{i,k}, i = 1, 2, \ldots, M, k = 1, 2, \ldots, \) are involved in these equations. Even for a fixed realization of \( t_{i,k} \), (11) is not linear in \( s_{i,k} \) because idle periods may occur at different times for different servers. In fact, for a fixed realization of \( t_{i,k} \), the service completion time \( t_{i,k} \) is a piecewise linear function of \( s_{i,k} \). This linear function is a special feature of closed Jackson networks, which makes the linearization equation exact in a small neighborhood rather than approximate to the first order of changes.

Now suppose one parameter, \( \theta = \theta_0 \), changes by a small amount \( \Delta \theta = \Delta \theta_0 \). We shall call the system with \( \theta = \theta_0 + \Delta \theta \) the nominal system (perturbed system), and a sample path of the nominal system (perturbed system) a nominal path (perturbed path).

Let \( s_{i,k} \) be the service time of the \( k \)th customer of server \( i \) in the perturbed system since the observation starts, and \( t_{i,k} \) be the service completion time of this customer. For the perturbed system we have the following two equations which are similar to (10) and (11).

\[
s_{i,k} = -s_{i}^* \ln (1 - t_{i,k}),
\]

\( i = 1, 2, \ldots, M, k = 1, 2, \ldots \)

where \( s_{i}^* \) is the mean service time of the perturbed system, \( s_{i}^* = s_{i}^* \) if \( i \neq v \), and \( s_{i}^* = s_{i} + \Delta s_{i,v} \), and

\[
t_{i,k} = \begin{cases} 
  t_{i,k-1} + s_{i,k} & \text{if } n(t_{i,k-1}^+) \neq 0, \\
  t_{i,h} + s_{i,k} & \text{if } n(t_{i,k-1}^+) = 0.
\end{cases}
\]

Note that the same random variables \( t_{i,k} \) are used for both the nominal and the perturbed systems. Let \( \Delta s_{i,k} = s_{i,k}^* - s_{i,k} \). \( \Delta s_{i,k} \) is called the perturbation generated during the corresponding service period. From (10) and (12), we have

\[
\Delta s_{i,k} = \begin{cases} 
  0 & \text{if } i \neq v, \\
  -s_{i}^* \ln (1 - t_{i,k}) & \text{if } i = v.
\end{cases}
\]

Although the form of (13) looks the same as that of (11), the equations for a particular \( (i, k) \) may be different for both systems. For instance, server \( i \) may meet an idle period after it serves its \( (k - 1) \)th customer in the nominal system (the second equation in (11) holds for this \( (i, k) \)); this idle period may not exist in the perturbed path if the \( k \)th customer in the perturbed system enters server \( i \) earlier than that in the nominal system (hence the first equation in (13) holds for this \( (i, k) \)).

Let \( \Delta t_{i,k} = t_{i,k} - t_{i,k}^* \). \( \Delta t_{i,k} \) is called the perturbation of server \( i \) at \( t_{i,k} \), or the perturbation of the \( k \)th customer of server \( i \). Because (13) and (11) are different for a particular pair \( (i, k) \), we have to make some restrictions before we can derive an equation for \( \Delta t_{i,k} \).

Let \( X_i = \mathbb{X}(T_i, \xi) \). Then \( X_i = \{ X_j \}_{j=0}^{\infty} \) is the Markov chain embedded in \( \mathbb{X}(t, \xi) \). We define:

**Definition 1**: Two sample paths are said to be similar, in \([0, T_i] \), if the sequences \( \{ X_j \}_{j=0}^{\infty} \) are the same for both paths.

If the nominal and the perturbed paths are similar, then for any pair \( (i, k) \), (11) and (13) are exactly the same. Suppose that the initial states of both systems are the same, then \( n(t_{i,k}^*) = n(t_{i,k}) \) for all \( (i, k) \). Thus, we can take the difference between the two sides of these two equations and obtain

\[
\Delta t_{i,k} = \begin{cases} 
  \Delta t_{i,k-1} + \Delta s_{i,k} & \text{if } n(t_{i,k-1}^+) \neq 0, \\
  \Delta t_{i,k} + \Delta s_{i,k} & \text{if } n(t_{i,k-1}^+) = 0.
\end{cases}
\]

where \( \Delta s_{i,k} = s_{i,k}^* - s_{i,k} \) is the perturbation due to the change of \( s_{i,k}^* \).

It was proved in Cao [5] that for any nominal path and finite \( l \), there exists, with probability one, a small \( \delta > 0 \) such that if \( |\Delta s_{i,k}^*| < \delta \), then the perturbed path is similar to the nominal one. Therefore, (15) holds with probability one in a neighborhood of \( \delta_{i,k} \).

Equations (14) and (15) determine a linear relation between \( \Delta t_{i,k} \) and \( \Delta s_{i,k} \). This is analogous to (4) for continuous variable systems.

Now we have established the linear equation for the perturbation \( \Delta t_{i,k} \) in a closed queueing network. To show the similarity between DEDSs and continuous variable dynamic systems (CVDs), we summarize below the steps of obtaining a perturbed path for both DEDSs and CVDs.

1) For CVDs:
   a) Solve (3) for a nominal path \( x(t) \) in \( t \in [0, T] \).
   b) Calculate \( \Delta x(t)/\Delta \) along the nominal path. (\( \Delta x(t) \) is the perturbation generated at time \( s \) because of \( \Delta \phi(s) \)).
   c) Calculate \( \Delta x(t), t \in [0, T] \) by (7). (\( \Delta x(t) \) is the effect of \( \Delta \phi(s) \) generated at time \( s \) on \( x(t) \), or the perturbations propagated to time \( t \) from \( \Delta \phi(s) \). \( \Delta x(t) \) is the accumulation of \( \Delta \phi(s) \).
   d) \( x(t) + \Delta x(t), t \in [0, T] \) is the perturbed path of the CVDs for \( \theta + \Delta \theta \).

2) For DEDS:
   a) Simulate the queueing network to obtain a nominal path with transition times \( t_{i,k} = T_i, t \in [1, 2, \ldots, l] \).
   b) Calculate \( \Delta s_{i,k} \) by (14). \( \Delta s_{i,k} \) is the perturbation generated at \( t_{i,k} \) because of \( \Delta \theta \).
   c) Calculate \( \Delta t_{i,k} \) for all \( i, k \) such that \( t_{i,k} \leq T_i \), using (15). (Equation 15 gives the accumulation of the perturbations propagated to \( T_i \)).
   d) \( t_{i,k} = t_{i,k} + \Delta t_{i,k} \) is the service completion times of the perturbed path.

Equations (14) and (15) can be verbally described as simple rules. Equation (14) is called the perturbation generation rule, which says that a perturbation generated in a service period \( s_{i,k} \) for an exponential server is proportional to the service length. If there is no perturbation generated in \( s_{i,k} \), then the perturbation of \( t_{i,k} \) is

\[
\Delta t_{i,k} = \begin{cases} 
  \Delta t_{i,k-1} & \text{if } n(t_{i,k-1}^+) \neq 0, \\
  \Delta t_{i,k} & \text{if } n(t_{i,k-1}^+) = 0.
\end{cases}
\]
This equation can be stated as the following perturbation propagation rules:

1) A perturbation will be propagated to the next customer in the same busy period ($\Delta_{i-1} = \Delta_{i-1} - \pi$).

2) If a customer coming from server $j$ starts a new busy period at server $i$, then the perturbation of server $j$ will be propagated to server $i$ ($\Delta_{i-k} = \Delta_{i-k}$).

Note that in the second propagation rule if $\Delta_{i-1} > 0$ and $\Delta_{i-2} = 0$, then, after propagation, $\Delta_{i-1} = 0$. In this case, we say that the perturbation $\Delta_{i-1}$ is canceled or lost. This is a special case of perturbation propagation.

If there is a perturbation $\Delta_{i-1}$ generated in $s_{i-1}$, then the perturbation is propagated to $t_i$, which is determined by (16).

We have introduced two main concepts of perturbation analysis, i.e., perturbation generation and perturbation propagation. We have shown how these concepts are related to the linearization theory of a continuous variable system. In the next section, we shall study the performance and some special properties of a queueing network.

IV. PERFORMANCE SENSITIVITY

A. Transient Throughput

There are $L$ service completions in $[0, T_i]; T_i$ is a function of $S_i$ and $\xi$. Thus, the system throughput is defined as

$$TP_i(S_i, \xi) = \frac{l}{T_i(S_i, \xi)}.$$ \hspace{1cm} (18)

The change of $TP_i$ because of $\Delta S_i$ (provided that $\Delta T_i$ is small) is:

$$\Delta TP_i(S_i, \xi) = -\frac{l}{T_i(S_i, \xi)} \Delta T_i(S_i, \xi).$$

For convenience, we shall study the sensitivity of the relative change of the throughput with respect to the relative change of a mean service time, i.e., the elasticity of throughput with respect to the mean service time defined as follows:

$$\frac{\Delta TP_i(S_i, \xi)}{TP_i(S_i, \xi)} = -\frac{S_i}{T_i(S_i, \xi)} \frac{\Delta T_i(S_i, \xi)}{\Delta S_i}. \hspace{1cm} (17)$$

$\Delta T_i$ can be obtained by using the perturbation generation and perturbation propagation rules described in the last section. We shall derive an equation similar to (9) for the continuous variable systems. For this purpose, it is necessary to introduce another concept, the realization index of a perturbation.

The realization index measures the final effect of a perturbation generated in $[0, T_i]$ on $T_i$. As described above, a perturbation will be propagated from one server to another server, or it will be canceled after an idle period. Let $\Gamma = \{1, 2, \cdots, M\}$, and $V \subseteq \Gamma$ be a subset. Recall that $T_i = t_{i, n, P, i}$, i.e., the last service completion happens at server $i$.

Suppose that at the $h$th state (i.e., at $T_i$), the Markov process $\Omega(t, \xi)$ is an arbitrary service process and only servers in a set $V$ have a perturbation (with the same size which is small enough to keep the perturbed path similar to the nominal one). Then the set $V$ is called the perturbation set at the $h$th state. On a sample path, we define:

$$R_i(n, V, l) = \begin{cases} 1 & \text{if the perturbation in } (n, V, l) \text{ is propagated to server } u \text{ at time } T_i \\ 0 & \text{if not.} \end{cases} \hspace{1cm} (19)$$

$R_i(n, V, l)$ is called the realization index of the perturbation at the $l$th state. If $V$ contains only one server $v$, we simply write $V = v$ and $R_i(n, V, l) = R_i(n, v, l)$.

Consider a customer’s service period at a server. Assume that the customer starts receiving service from server $v$ at time $T_i$. During the service period, customers in other servers may transfer from one server to the other. Let $T_{i-1} < T_{i-1} < \cdots < T_{i-1} < T_{i-1}$ be the transition times in the service period. At time $T_{i-1}$, the customer leaves server $v$. If $n(l) = n(l) + 1$, the states visited by the system after this period is the length of this customer’s service duration is its service time $s$. It can be decomposed into

$$s = S_i + S_i + \cdots + S_i,
$$

where $S_i$ is the time the system stays in $n(l + 1)$ because of the change in service time of server $v$ at the end of the service period. The final contribution of this perturbation to $\Delta T_i$ is the sum of all perturbations obtained by server $v$ at the end of the service period. The final contribution to $\Delta T_i$ is the sum of all perturbations obtained by server $v$ at the end of the service period. The final contribution to $\Delta T_i$ is the sum of all perturbations obtained by server $v$ at the end of the service period. The final contribution to $\Delta T_i$ is the sum of all perturbations obtained by server $v$ at the end of the service period.

$$\lambda_{S_i} R_i(n(l) + 1), v, l + 1) + \lambda_{S_i} R_i(n(l) + 1), v, l + 1) + \cdots + \lambda_{S_i} R_i(n(l) + 1), v, l + 1). \hspace{1cm} (19)$$

In (19), $\lambda_{S_i} R_i(n(l) + 1), v, l + 1)$ is the perturbation propagated to $T_i$ due to a perturbation of size $\lambda_{S_i} R_i(n(l) + 1), v, l + 1)$, and $\lambda_{S_i} R_i(n(l) + 1), v, l + 1)$ is the perturbation propagated to $T_i$ due to a perturbation of size $\lambda_{S_i} R_i(n(l) + 1), v, l + 1).$ Equation (19) says that the effect of $\lambda_{S_i}$ on $T_i$ equals the sum of the effects of all the $\lambda_{S_i} R_i(n(l) + 1), v, l + 1)$ on $T_i$. This means that, as far as the final contribution to $\Delta T_i$ is concerned, a perturbation $\Delta S_i = \lambda_{S_i}$ of a customer’s service time is equivalent to a series of perturbations of the same sizes that the system stays in states $n(l) + 1, i = 0, 1, 2, \cdots, r$.

From the above discussion, the definition of realization index, and the perturbation generation and perturbation propagation rules, $\Delta T_i$ has the following value:

$$\Delta T_i = \sum_{k, v \in V} \sum_{l_i \leq l} S_i R_i(n(l), v, l_i)$$

where $n(l)$ is the last state visited in the service period of the $k$th customer of server $v$. In the equation, $S_i R_i(n(l), v, l_i)$ is the perturbation generated in the $k$th customer service period at server $v$. $S_i R_i(n(l), v, l_i)$ is the perturbation propagated to $T_i$ due to $\lambda_{S_i} R_i(n(l), v, l_i)$. $\Delta T_i$ equals the sum of the perturbations propagated to $T_i$ due to all the perturbations generated in

$\lambda_{S_i} R_i(n(l), v, l_i)$.
[0, T_i]. By (19), this is equivalent to

$$\Delta T_i = \lambda \sum_{i=1}^{l} S_j R_l(n, v, l)$$

where (X_i) is the rth component of X_v, i.e., the number of customers in server v at T_i. The indices l such that (X_l) = 0 are excluded in the above expression because there is no perturbation generated in idle periods of server v. However, if (X_l) = 0, then R_l(n, v, l) = R_l(X_l, v, l) = 0. Thus, we can rewrite the above equation as

$$\Delta T_i = \lambda \sum_{l=1}^{L} S_j R_l(n, v, l). \tag{20}$$

Therefore,

$$\lim_{\Delta T_i \to 0} \frac{S_j}{\Delta S_j} \Delta T_i = \lim_{\Delta T_i \to 0} \frac{\Delta T_i}{T_i} = \frac{1}{T_i} \sum_{l=1}^{L} S_j R_l(n, v, l). \tag{21}$$

By (17) and (21), the sample elasticity of TP with respect to S_j is

$$\frac{\partial TP(S_j, \xi)}{\partial S_j} = \lim_{\Delta T_i \to 0} \frac{\Delta T_i}{T_i} = \frac{1}{T_i} \sum_{l=1}^{L} S_j R_l(n, v, l). \tag{22}$$

Equation (20) or (22) for a closed queueing network plays the same role as (9) does for a continuous variable system. Since the performance measure T_i depends only on the value at the end of the observation period, (22) corresponds to the case e x = 1, t = 0 in (9). Equation (22) is a fundamental equation for practical algorithms using perturbation analysis to estimate throughput sensitivities.

B. Steady-State Throughput

We have discussed the similarity between perturbation analysis of a DEDS and the linearization theory of a nonlinear continuous variable system. As one may expect, a DEDS, such as a closed queueing network, also possesses its own structure and properties. One distinguishing feature of a queueing network is the strong connection between servers in the network. It is the strong interaction between servers that governs the evolution of the system state. This strong interaction determines the final effect of a perturbation. The accumulation of all final effects of all perturbations determines the sensitivity of the steady-state throughput.

Let us first introduce some terminologies. Consider a perturbed queueing network. Suppose that at time t = 0, the network is in state n and the perturbation set is V, i.e., all and only servers in V have a perturbation with the same small size. We say that the system is in a perturbed state (n, V). The perturbation in set V may be propagated to servers not in V or may be canceled, as the system evolves. Thus, V depends on time t, and can be denoted as V(t).

Note that $\Gamma = \{1, 2, \ldots, M\}$, and the null set $\Phi$ is two absorbing states of V(t) under the perturbation propagation rules. That is, if at some time $t$, $V(t) = \Gamma$ (or $\Phi$), then $V(t) = \Gamma$ (or $\Phi$) for all $t > t$. If at some time $t > 0$, $V(t) = \Gamma$ (or $\Phi$) on a sample path, we say that the perturbation in (n, V) is realized (or lost) by the system on this sample path.

Now we introduce the notion of irreducible networks. Some queueing networks can be decomposed into several small subnetworks which do not communicate with each other at steady-state. In such a network, any perturbation in one subnetwork cannot affect servers in other subnetworks. The study of such a network can be replaced by the study of its subnetworks. Thus, without loss of generality, we only consider irreducible networks. An irreducible network is one whose transition matrix $Q = [q_{ij}]$ is irreducible. In an irreducible network, a customer in any server may reach any other server in the network with probability one, either directly, or by going through some other servers.

Cao [2] proved that in an irreducible closed Jackson network a perturbation will, with probability one, be either realized or lost. This property indicates that there is very strong interaction between servers in an irreducible closed queueing network. In such a network, if, somehow, one server obtains a perturbation $\Delta$, then either this perturbation will eventually be lost, in which case the perturbed path will, after a while, be exactly the same as the nominal one as if nothing had happened; or all other servers will eventually possess the same perturbation, in which case the perturbed path will, after a while, look the same as the nominal one, except that every event is delayed (or advanced) by the same amount $\Delta$. Based on this property, we define:

**Definition 2:** The probability that a perturbation in (n, V) is realized in an irreducible network is called the realization probability of the perturbation.

The realization probability of (n, V) is denoted as $f(n, V)$. The following theorems were proved in Cao [2]:

**Theorem 1:**

i) $f(n, i) = 0$, if $n_i = 0$. \tag{23}

ii) if $V_i \cap V_j = \Phi$, and $V_i \cup V_j = V_0$, then $f(n, V_i) + f(n, V_j) = f(n, V_0)$. \tag{24}

iii) $\sum_{i=1}^{M} f(n, i) = 1$. \tag{25}

**Theorem 2:** The realization probabilities in an irreducible closed Jackson network satisfy the following equations:

$$\begin{align*}
\left\{ \sum_{i=1}^{M} e(n_i) \nu_i \right\} f(n, k) \\
= \sum_{i=1}^{M} \sum_{j=1}^{M} e(n_i) q_{ij} f(n_j, k) + \sum_{i=1}^{M} \left[ 1 - e(n_i) \right] \mu_i q_{ik} f(n_k, k) \\
&\quad \text{if } n_k > 0, k \in \Gamma
\end{align*} \tag{26}$$

where $n_j = (\cdots, n_i - 1, \cdots, n_j + 1, \cdots)$ is a neighboring state of $n$, and $e(n) = 1$ if $n > 0$, $e(n) = 0$ if $n = 0$.

These equations are similar to the flow balance equation for the steady-state probabilities of a queueing network. The last term on the right-hand side of (26) represents the effect of perturbation propagation. Solving (23)–(26), we can obtain all the realization probabilities for a closed Jackson network.

The realization probability can be used to calculate the sensitivity of the steady-state throughput with respect to a mean service time. In fact, we have the following theorem:

**Theorem 3:**

$$\lim_{t \to \infty} \frac{1}{T_i} \sum_{i=1}^{L} S_j R_l(n, v, l) = \sum_{n \in \Phi} f(n, v) \text{ w.p.1.} \tag{27}$$
A rigorous proof of this theorem is in Cao [2]. An intuitive explanation of (27) is as follows: For any \( n \) we define

\[
\chi_i(n) = \begin{cases} 
1 & \text{if } X_i = n, \\
0 & \text{if } X_i \neq n.
\end{cases}
\]

Then

\[
\lim_{L \to \infty} \frac{1}{L} \sum_{i=1}^{L} \{ S_i R_i(n, v, l) \} = \sum_{n} \left( \lim_{L \to \infty} \frac{1}{L} \sum_{i=1}^{L} S_i \chi_i(n) R_i(n, v, l) \right). \tag{28}
\]

As \( L \to \infty \), \( R_i(n, v, l) \) goes to 1, if the perturbation in \( (X_i, i) \) is realized; and goes to 0, if the perturbation is lost. The probability that \( R_i(n, v, l) \) goes to 1 is \( f(n, v) \). Note that by the ergodicity of \( \mathcal{E}(l, \xi) \), \( \lim_{L \to \infty} \frac{1}{L} \sum_{i=1}^{L} S_i \chi_i(n) R_i(n, v, l) \) converges to \( p(n, f(n, v, l)) \). This explains (27).

From (22) and (27), we get

\[
\lim_{L \to \infty} \frac{S_i}{TP_i(S_i, \xi)} \frac{\partial TP_i(S_i, \xi)}{\partial S_i} = -\sum_{n} \rho(n) f(n, v). \tag{29}
\]

Let \( TP_i(s_i) = \lim_{L \to \infty} TP_i(S_i, \xi) \) be the steady-state throughput. By the ergodicity of the Markov process \( \mathcal{E}(l, \xi) \), this limit does not depend on \( \xi \). A natural corollary of (29) is

\[
\frac{S_i}{TP_i(s_i)} \frac{\partial TP_i(s_i)}{\partial s_i} = -\sum_{n} \rho(n) f(n, v). \tag{30}
\]

This is an analytical formula for the sensitivity of the steady-state throughput of a closed Jackson network with single server nodes and single class customers. For more discussion of this formula, see Cao [2], [5].

Equation (29) integrates the concepts of linearization and realization. The former is a counterpart of the similar concept of continuous variable systems, and the latter is based on the special structure of a closed queueing network.

The concept of realization probabilities has been extended to multiclass queueing networks [6]. It can also be used to calculate some other performance sensitivities [4].

V. DISCUSSION AND CONCLUSION

In this paper, we made a comparison between the dynamics of a continuous variable system and a closed queueing network. It was shown that the perturbation analysis of queueing networks is, in a sense, similar to the linearization theory of nonlinear dynamic systems. Based on this point of view, we explained three main concepts of perturbation analysis of a closed queueing network: perturbation generation, perturbation propagation, and perturbation realization.

We also illustrated that because of the special structure, a closed queueing network has its own dynamic properties. These properties are due to the strong interaction between servers in an irreducible network. In regard to perturbation propagation, this strong interaction leads to the fact

\[
\lim_{L \to \infty} R_i(n, v, l) = 1 \quad \text{or} \quad 0,
\]

with probability one as \( t \to \infty \). Of course, not many continuous systems possess this property.

The principle discussed in this paper for closed queueing networks certainly applies to other DEDSs. Different perturbation generation and propagation rules must be developed for different DEDSs, and special effects are required for investigating specific properties of a particular DEDS. Suri [15] describes the rules for a class of general DEDSs. Some progress has been achieved recently in extending perturbation analysis to more general systems (see e.g., Cao [1], [3], Gong and Ho [8], and Ho and Li [14]).

Finally, it is worthwhile mentioning an important fact. Although the comparison of the dynamics made in this paper looks quite natural, it is only incidental. The development of perturbation analysis was not stimulated by the idea of applying the linearization theory to DEDSs. Perturbation analysis originated in solving a practical problem (Ho, Eyler, and Chien [13]). Since then, many experimental as well as theoretical analyses have been done (see e.g., Ho and Cao [11], Cao [3], Suri and Zanazis [16], and Cassandra and Ho [7]). It is only after these works that the comparison in this paper became clear. There are still many open problems remaining in this area.

REFERENCES

Xi-Ren Cao (Member, IEEE) received the M.S. degree in engineering and the Ph.D. degree in applied mathematics from Harvard University, Cambridge, MA, in 1981 and 1984 respectively.

From January 1984 to April 1986, he was a research fellow at the Division of Applied Sciences, Harvard University. He is currently with Digital Equipment Corporation, Marlboro, MA. His current research interests are in the areas of modeling and performance evaluation of computer systems, theory and applications of queueing networks and discrete event systems, and stochastic approximation and optimization.

Dr. Cao received the Outstanding Transactions Paper Award from the IEEE Control System Society in 1987, and is Associate Editor of the IEEE TRANSACTIONS ON AUTOMATIC CONTROL.