'91 韓國自動制御學術會議論文集 1991. 10.22~24

EXACT PERTURBATION ANALYSIS TECHNIQUE AND OPTIMAL BUFFER STORAGE DESIGN FOR TANDEM QUEUEING NETWORKS

Wook Hyun Kwon, Hongseong Park and B.J. Chung

Engineering Research Center for

Advanced Control and Instrumentation

Seoul National University

Sillim_Dong Kwanak_gu, Seoul 151-742, KOREA

TEL.(822) 887-0040 FAX.(822) 871-7010

ABSTRACT

In this paper, we suggest the exact perturbation analysis(Exact_PA) technique with respect to the buffer storage in tandem queueing networks, through which the optimal buffer storage design problem is considered. The discrete event dynamic equations for the departure time of a customer are presented together with the basic properties of Full Out(FO) and No Input(NI) with respect to the buffer storage. The new perturbation rules with respect to the buffer storage are suggested, from which the exact perturbed path can be obtained. The optimal buffer storage problem is presented by introducing a performance measure consisting of the throughput and the buffer storage cost. An optimization algorithm maximizing this performance measure is derived by using the Exact_PA technique. The proposed perturbation analysis technique and the optimization algorithm are validated by numerical examples.

1. INTRODUCTION

The perturbation analysis(PA) technique is considered to be the combination of the analytic and the simulation method. Thus this method is known to be less costly and to have less assumptions. Many researchers have applied the PA methods to the sensitivity analysis of some performance measures such as the system throughput and the waiting time[2-8]. While most papers deal with PA techniques with respect to continuous parameters such as service time, only a few papers[1,3-7] have discussed PA techniques w.r.t. discrete parameters such as buffer storage.

The estimation accuracy is important in the PA technique. Generally, the performance measure such as the throughput cannot be accurately estimated if the amount of the parameter perturbation is not small. In the case of a discrete parameter, the amount of the parameter perturbation cannot be small as seen in the buffer storage of general queueing systems. Therefore, the accurate estimation of the performance measure with the discrete parameter perturbation is difficult through conventional PA techniques since conventional PA techniques require the small parameter perturbation for accurate estimation of performance measure. In this paper, we present a new technique called Exact_PA technique, through which the performance measure can be accurately estimated in the case that the discrete parameter such as the buffer storage is perturbed. This result is obtained from the special structure of tandem queueing networks w.r.t. the buffer storage.

The proposed exact PA technique will be shown to be an on_line algorithm in a sense that the exact performance measure can be estimated only with the observed data. Most PA techniques are off_line methods since they require a little of future information to predict interactions between servers. The on_line PA technique seldom exists. In [6], the on_line PA technique w.r.t. the buffer storage was studied for the system with only a single server(G/G/1/T). This on_line PA technique does not generally generate the exact estimation and cannot be applied to the tandem queueing network.

The Exact_PA technique will be utilized to the optimal buffer storage problem. In [1], an optimal buffer storage was obtained from a gradient technique in which the gradient is estimated from the conventional PA technique under the strong conditions that the sum of the buffer size is fixed and the cost per unit buffer storage in all servers is identical. The Exact_PA technique can be also utilized to estimate the gradient. But the gradient method based on the conventional PA technique[1-3] as well as the Exact_PA technique requires the careful adjustment of weighting factors to guarantee convergence. Since the exact estimation of the performance measure can be obtained, we suggest a gradient-free search algorithm utilizing Exact_PA technique, which is simple and easily implemented. The conventional PA technique can be hardly applicable for the gradient-free search algorithm since it has the large estimation error. In addition, the suggested optimization algorithm does not

require the strong conditions in [1].

This paper is organized as follows. In Section 2, discrete event dynamic equations and basic properties of Full Out(FO) and No Input(NI) with respect to the buffer storage in the tandem queueing network are presented. In Section 3, the new perturbation rules for the buffer storage are derived. The exact PA algorithm is presented to estimate the exact performance measure. In Section 4, the proposed PA technique is applied to an optimization problem to obtain the optimal buffer storages. In Section 5, numerical examples are shown. Conclusions are given in Section 6.

2. BASIC PROPERTIES

Let the event be the customer's departure from a server. The i-th server S_i is always in one of three possible states : 1) busy ; 2) blocked, also called Full Output(FO) ; 3) idle, also termed No Input(NI). Let B_i be the buffer size of S_i. The open tandem queueing network with finite queues such as a production line is shown in Fig. 2.1.

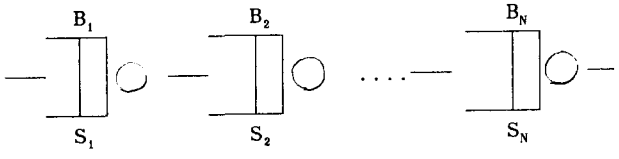

Fig. 2.1 Tandem Queueing Network

It is assumed that each server has no internal buffer and the initial number of customers in the buffer of each server is 0. It is also assumed that each server is a error-free machine. That is, the breakdown of the server is not considered. For the operations of the tandem queueing network, the following is assumed in the case that the service of the customer is completed at the server S_i. If the next server, S_{i+1} is free or idle, the service for the customer begins immediately at the server S_{i+1}. If the server S_{i+1} is not free and the buffer storage in the server S_{i+1} is not full, the customer is waiting in the buffer storage of the server S_{i+1}. If the buffer storage in the server S_{i+1} is full, the customer is waiting in the server S_i. The j-th customer in S_i, denoted by $C_{i,j}$, becomes the j-th customer in S_{i+1}, i.e. $C_{i+1,j}$, in the tandem queueing network. Let $d_{i,j}$ be the departure time of $C_{i,j}$ from S_i and $\underline{d}_{i,j}$ be the service completion time of $C_{i,j}$, and let b(i) be defined to indicate the server who blocks the i-th server S_i. Generally, b(i) is greater than i. Consider the case that the customer $C_{i,j}$ departs from the server S_i. If the blocking does not occur in the server S_{i+1} and the blocking occurs in the server S_i(that is, the buffer storage of S_{i+1} is full when $C_{i,j}$ departs from S_i), b(i) is (i+1). If the blocking does not occur in the server S_{i+3} and the blockings occur in three servers S_{i+2}, S_{i+1} and S_i(that is, the buffer storage of S_{i+3}, S_{i+2} and S_{i+1} are full when $C_{i,j}$ departs from S_i), it can be considered that the blocking in S_i occurs due to S_{i+3}. Thus b(i) is (i+3). General conditions for Full Out(FO) and No Input(NI) with respect to the buffer storage in the tandem queueing network are given as follows:

Fact 1(FO case) : If $\underline{d}_{i,j} < d_{b(i),j-T(i+1,b(i))}$ ($T(j,m) = \Sigma_j^m B_l$), FO also occurs.

Fact 2(NI case) : If $d_{i,j} > d_{i+1,j-1}$, NI occurs.

Due to the interaction of customers in the tandem queueing network, the following discrete event dynamic equations can be formulated.

$$\underline{d}_{i,k} = \begin{cases} d_{i-1,k} + t_{i,k} & ; \text{NI case} \\ d_{i,k-1} + t_{i,k} & ; \text{otherwise} \end{cases} \quad (2.1)$$

$$d_{i,k} = \begin{cases} d_{b(i),k-T(i+1,b(i))} & ; \text{FO case, } i<b(i) \\ \underline{d}_{i,k} & ; \text{otherwise} \end{cases} \quad (2.2)$$

where $t_{i,k}$ is the service time of $C_{i,k}$.

The FO time duration, which varies due to other FOs occurred in other servers, is summarized in the following.

Fact 3.(FO Time Duration) : In the case that $\underline{d}_{i,j}$ is less than $d_{b(i),j-T(i+1,b(i))}$ where i<b(i), the i-th server's FO time duration $(d_{b(i),j-T(i+1,b(i))} - \underline{d}_{i,j})$ is given by $(d_{i+1,j-B_{i+1}} - \underline{d}_{i,j})$.

This fact is proved as follows. Because the departure of $C_{i,j}$ is blocked by the effect of $C_{b(i),j-T(i+1,b(i))}$, all customers in servers, S_m, m=i+1,..,b(i)-1, are also blocked. At the departure time of $C_{b(i),j-T(i+1,b(i))}$, this blocking is dissolved. Thus the departure time of $C_{b(i)-1,j-T(i+1,b(i)-1)}$ is the departure time of $C_{b(i),j-T(i+1,b(i))}$ By the similar method,

$$d_{b(i),j-T(i+1,b(i))} = d_{b(i)-1,j-T(i+1,b(i)-1)} = d_{b(i)-2,j-T(i+1,b(i)-2)}$$
....

Thus, $d_{i+1,j-B_{i+1}} = d_{b(i),j-T(i+1,b(i))}$.

The above facts and discrete event dynamic equations are used for deriving the exact PA technique in the next section.

3. PA TECHNIQUE FOR THE PERTURBED BUFFER STORAGE

The PA technique with respect to the buffer storage will be presented for the case that multiple discrete parameters are perturbed at the same time. If the buffer B_i of the server S_i is perturbed by ΔB_i, it directly changes the customer's departure time of S_{i-1}. That is, the perturbations are generated in the S_{i-1}. And these perturbations are propagated to other servers according to some rules which will be described later. The performance measure such as the throughput can be changed in this case. For the general queueing networks, the exact estimation of the performance measure is difficult due to the complex interaction of the events during the period of FO or NI. For the tandem queueing network, the exact estimation of the performance measure can be easily obtained due to the special properties of the tandem queue.

For the perturbation rules, we define the following terminologies:

$NIT_{i,j}$: NI's time duration just prior to the service of $C_{i,j}$ in the nominal sample path.

$PNIT_{i,j}$: NI's time duration just prior to the service of $C_{i,j}$ in the perturbed sample path.

$FOT_{i,j}$: FO's time duration after the service of $C_{i,j}$ was just completed in the nominal sample path

$PFOT_{i,j}$: FO's time duration after the service of $C_{i,j}$ was just completed in the perturbed sample path.

$\Delta_{i,j}$: the amount of perturbations in the departure time, accumulated in S_i until the customer $C_{i,j}$ departs from S_i.

$\delta_{i,j}$: the amount of perturbation in the departure time, generated in S_i at the departure time of $C_{i,j}$.

In the definition of $\Delta_{i,j}$ and $\delta_{i,j}$, "amount of perturbation in the departure time" means the difference between the departure time of the customer $C_{i,j}$ in the nominal path and the departure time of the customer $C_{i,j}$ in the perturbed path. The departure time of the customer $C_{i,j}$ in the perturbed path is denoted by $d'_{i,j}$, which is

$$d'_{i,j} = d_{i,j} + \Delta_{i,j}. \quad (3.1)$$

In order to estimate the exact perturbed departure time, $\Delta_{i,j}$ must be estimated exactly, which can be derived from

$$\delta_{i,j} = (PNIT_{i,j} + PFOT_{i,j}) - (NIT_{i,j} + FOT_{i,j}) \quad (3.2)$$

$$\Delta_{i,j} = \Delta_{i,j-1} + \delta_{i,j}, \quad \Delta_{i,0} = 0 \quad (3.3)$$

It will be shown that Eq.(3.2) and Eq.(3.3) are the exact amount of perturbation generated and accumulated at the departure time of $C_{i,j}$. Suppose that the B_{i+1} of the server S_{i+1} is perturbed by ΔB_{i+1}. First, we will consider the perturbed service completion time of the customer $C_{i,j}$. Since the departure time of the customer $C_{i,j}$ in the perturbed path, $d'_{i,j}$ depends on the departure time, $d'_{i+1,j-B_{i+1}-\Delta B_{i+1}}$, of the customer $C_{i+1,j-B_{i+1}-\Delta B_{i+1}}$, the service completion time of $C_{i,j}$ in the perturbed path, $\underline{d}'_{i,j}$ must be obtained in order to compute $d'_{i,j}$. Fig. 3.1 shows the comparison of the service completion time of the customer $C_{i,j}$ with the departure time of the customer $C_{i,j}$ in the perturbed path.

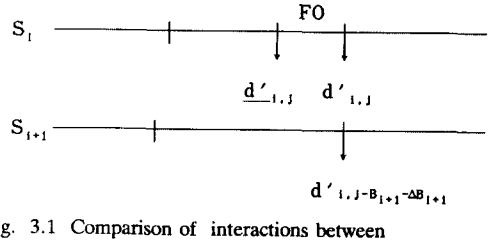

Fig. 3.1 Comparison of interactions between two adjacent servers in FO case

If $\underline{d}'_{i,j}$ is less than $d'_{i+1,j-B_{i+1}-\Delta B_{i+1}}$, the FO occurs in the perturbed path. On the other hand, if $\underline{d}'_{i,j}$ is not less than $d'_{i+1,j-B_{i+1}-\Delta B_{i+1}}$, the FO does not occur. From Fig. 3.1 and Eq. (3.1)-(3.3), the perturbed service completion time of $C_{i,j}$, $\underline{d}'_{i,j}$ is given by $(d_{i,j} + \Delta_{i,j-1} + PNIT_{i,j} - NIT_{i,j} - FOT_{i,j})$. Next, the activities of customers in the server S_i will be considered. The perturbed departure time of the customer $C_{i,1}$, $d'_{i,1}$ is equal to the nominal departure time, $d_{i,1}$, of the $C_{i,1}$ since the initial number of customers in the buffer of each server is zero. The perturbed service completion time of $C_{i,2}$, $\underline{d}'_{i,2}$, is given by $(d_{i,2} + \Delta_{i,1} + PNIT_{i,2} - NIT_{i,2} - FOT_{i,2})$. $NIT_{i,2}$ and $FOT_{i,2}$ are the known values in the nominal path and $PNIT_{i,2}$ is obtained from $(d'_{i-1,2} - d'_{i,1})$, where $d'_{i-1,2}$ and $d'_{i,1}$ are the values calculated in the previous step. From Eq.(3.1)-Eq.(3.3), we obtain that

$$d'_{i,2} = \underline{d}'_{i,2} + PFOT_{i,2}$$

where $PFOT_{i,2}$ is given by $(d'_{i+1,2-B_{i+1}-\Delta B_{i+1}} - \underline{d}'_{i,2})$ if $d'_{i+1,2-B_{i+1}-\Delta B_{i+1}}$ is greater than $\underline{d}'_{i,2}$. The $d'_{i+1,2-B_{i+1}-\Delta B_{i+1}}$ is also obtained in the previous step. Thus $d'_{i,2}$ is derived exactly. Perturbed departure times of other customers, $d'_{i,3}, d'_{i,4}, \ldots$, are calculated exactly by the similar method. Because the exact amount of $NIT_{i,j}$, $FOT_{i,j}$, $PNIT_{i,j}$, and $PFOT_{i,j}$ can be obtained from the above procedure, $\delta_{i,j}$ is obtained from Eq.(3.2). And Eq.(3.3) consists of the amount of the perturbation in the departure time$(\Delta_{i,j-1})$, accumulated until the previous customer $C_{i,j-1}$ departed from the server S_i and the amount of the perturbation in the departure time$(\delta_{i,j})$, generated at the departure time of the customer $C_{i,j}$. Thus, Eq.(3.3) can be written equivalently as

$$\Delta_{i,j} = \sum_{k=1}^{j} \delta_{i,k}.$$

From above equations, $\Delta_{i,j}$ can be obtained. As can be seen above, the departure time of the customer is computed only with the observed data.

In the perturbation amount of the departure time, $\delta_{i,j}$, of Eq.(3.2), $NIT_{i,j}$ and $FOT_{i,j}$ are known values but $PNIT_{i,j}$ and $PFOT_{i,j}$ are unknown values. Thus $PNIT_{i,j}$ and $PFOT_{i,j}$ have to be obtained. First, we will obtain the perturbation accumulated in the preceding server, S_{i-1}, when the succeeding server, S_i, has the perturbed buffer. The perturbation rule applied for this case will be called *"Perturbation rule A"* and is as follows.

Let us consider the FO case in the perturbed path. The perturbation of the buffer storage has much effect on FO. Though the customer $C_{i-1,j}$ meets the FO in the nominal path, this FO can be eliminated or its time duration is varied due to the ΔB_i in the perturbed path(the typical example of the elimination is the case of the first FO for the positive ΔB_i). And, though the customer $C_{i-1,j}$ doesn't meet the FO in the nominal path, the FO can be created due to the ΔB_i in the perturbed path. The amount of perturbations due to FO is obtained from $(PFOT_{i-1,j} - FOT_{i-1,j})$, where $PFOT_{i-1,j}$ is obtained from

If $d_{i-1,j} + \underline{\Delta}_{i-1,j} < d_{i,j-B_i-\Delta B_i} + \Delta_{i,j-B_i-\Delta B_i}$, then

$$PFOT_{i-1,j} = d_{i,j-B_i-\Delta B_i} + \Delta_{i,j-B_i-\Delta B_i} - d_{i-1,j} - \Delta_{i-1,j}$$

else

$$PFOT_{i-1,j} = 0 \qquad (3.4)$$

where $\underline{\Delta}_{i-1,j}$ is defined by $(\Delta_{i-1,j-1}+PNIT_{i-1,j}-FOT_{i-1,j}-NIT_{i-1,j})$.

Next, let us consider the NI case. From the similar way to the FO case, the amount of perturbations due to NI is also obtained. While $C_{i-1,j}$ meets FO in the interaction between S_{i-1} and S_i, $C_{i-1,j}$ meets NI in the interaction between S_{i-2} and S_{i-1}. The amount of the perturbation due to NI is obtained from $(PNIT_{i-1,j} - NIT_{i-1,j})$, where $PNIT_{i-1,j}$ is derived from

If $d_{i-2,j+1} + \Delta_{i-2,j+1} > d_{i-1,j} + \Delta_{i-1,j}$, then

$$PNIT_{i-1,j} = d_{i-2,j+1} + \Delta_{i-2,j+1} - d_{i-1,j} - \Delta_{i-1,j}$$

else

$$PNIT_{i-1,j} = 0 \qquad (3.5)$$

Due to the perturbed buffer storage, it is shown that the NI and FO can be eliminated/created or their time durations can be varied. As $C_{i-1,j}$ can meet both NI and FO, the amount of the perturbation occurred at the departure of $C_{i-1,j}$ is

$$\delta_{i-1,j} = (PNIT_{i-1,j} + PFOT_{i-1,j}) - (NIT_{i-1,j} + FOT_{i-1,j})$$

Finally, the perturbation accumulated in the server S_{i-1} until the customer $C_{i-1,j}$ departs from the server S_{i-1} is

$$\Delta_{i-1,j} = \sum_{l=1}^{j} \delta_{i-1,l} = \Delta_{i-1,j-1} + \delta_{i-1,j}$$

So far, we have analyzed the interaction between two adjacent servers when the succeeding server has a perturbed buffer storage.

Secondly, we will obtain the perturbation accumulated in the preceding server, S_{k-1}, when the succeeding server, S_k, has no perturbed buffer. The perturbation rule applied for this case will be called *"Perturbation rule B"*. This perturbation rule is similar to the *Perturbation rule A* discussed above and is summarized as follows.

1) FO case

If $d_{k-1,j} + \underline{\Delta}_{k-1,j} < d_{k,j-B_k} + \Delta_{k,j-B_k}$, then

$$PFOT_{k-1,j} = d_{k,j-B_k} + \Delta_{k,j-B_k} - d_{k-1,j} - \Delta_{k-1,j}$$

else

$$PFOT_{k-1,j} = 0$$

where $\underline{\Delta}_{k-1,j}$ is defined by

$$(\Delta_{k-1,j-1}+PNIT_{k-1,j}-FOT_{k-1,j}-NIT_{k-1,j}).$$

2) NI case

If $d_{k-2,j+1} + \Delta_{k-2,j+1} > d_{k-1,j} + \Delta_{k-1,j}$, then

$$PNIT_{k-1,j} = d_{k-2,j+1} + \Delta_{k-2,j+1} - d_{k-1,j} - \Delta_{k-1,j}$$

else

$$PNIT_{k-1,j} = 0$$

$$\delta_{k-1,j} = (PNIT_{k-1,j} + PFOT_{k-1,j}) - (NIT_{k-1,j}+FOT_{k-1,j})$$
$$\Delta_{k-1,j} = \Delta_{k-1,j-1} + \delta_{k-1,j}.$$

It is noted that the perturbation rules for the buffer storage are utilized in the preceeding server between two adjacent servers. Conceptually, *Perturbation rule A* corresponds to the combined rule of the conventional generation and the conventional propagation rules. *Perturbation rule B* corresponds to the conventional propagation rule.

To validate the above rule, the following examples are considered. Let's observe the activities of customers in the tandem queueing network with 3 servers in a model of Fig. 2.1, in which each buffer storage size(B_1, B_2, and B_3) is 1. First, an example for the positive perturbation is considered. The nominal sample path of the above system is shown in Fig. 3.2. The perturbed path of $\Delta B_2=1$ is shown in Fig. 3.3. The a_i is the i-th customer's arrival time from the external environment, the dot(.) or the blank represents NI or FO state, ⊣ is denoted by the service completion of customers and ⊥⊥ represents the one unit time. To simplify the notation, a_{k+j}, $d_{i,k+j}$ and $d'_{i,k+j}$ are replaced by a_j, $d_{i,j}$ and $d'_{i,j}$, $k>1$, respectively. To simplify the analysis, it is assumed that

$$a_1 = d_{2,-1} = 0 \quad \text{and} \quad \Delta_{2,0} = \Delta_{2,-1} = \Delta_{2,-2} = 0.$$

In the nominal path, the FO time of the customer $C_{1,1}$, $FOT_{1,1}$, is 1 and the NI time of the customer $C_{1,1}$, $NIT_{1,1}$, is 0. In the perturbed path, the perturbed NI time of the customer $C_{1,1}$, $PNIT_{1,1}$, is 0 by the rule (3.5). And the perturbed FO time of the customer $C_{1,1}$, $PFOT_{1,1}$, is 0 since the service completion time of $C_{1,1}$ in the perturbed path, $d'_{1,1}$, is greater than the perturbed departure time of $C_{2,-1}$, $d'_{2,-1}$. Thus $\Delta_{1,1}$ is -1 by Eq.(3.2) and Eq.(3.3). The perturbed departure time of the customer $C_{1,1}$, $d'_{1,1}$, is 5 by Eq.(3.1). By a similar method, the perturbed departure time of $C_{1,2}$, $C_{1,3}$, and $C_{1,4}$, i.e. $d'_{1,2}$, $d'_{1,3}$ and $d'_{1,4}$, are 7, 11 and 13, respectively. $\Delta_{1,2}$, $\Delta_{1,3}$ and $\Delta_{1,4}$ are -4, -4 and -7, respectively. For the customer $C_{1,5}$, $FOT_{1,5}$, $NIT_{1,5}$ and $PNIT_{1,5}$ are 4, 0, and 4, respectively. Because the service completion time of $C_{1,5}$ in the perturbed path, $d'_{1,5}$, is less than the perturbed departure time of $C_{2,3}$, $d'_{2,3}$, the FO time of $C_{2,3}$ in the perturbed path, $PFOT_{1,5}$, is 1 by the rule (3.4). Thus $\Delta_{1,5}$ is -6 by Eq.(3.3) and $d'_{1,5}$ is 20 by Eq.(3.1). By a similar method, the exact departure time of other customers in the server can be also computed.

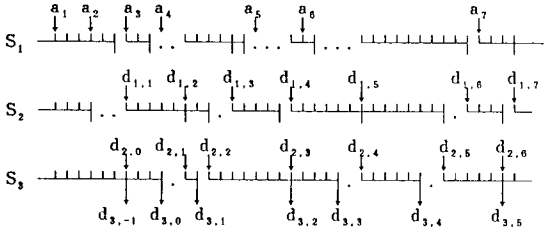

Fig. 3.2 the nominal sample path

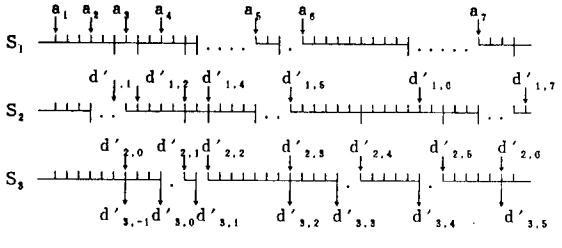

Fig. 3.3 the perturbed path ($\Delta B_2=1$)

So far, the example for the positive perturbation has been considered.

We choose the throughput as the performance measure of interest in this section, which can be defined as

$$TP = N/d_{L,N} \qquad (3.6)$$

where the L-th server is the last server in the tandem queueing network and $d_{L,N}$ is the departure time of the customer $C_{L,N}$. The perturbation of the throughput, ΔTP, resulting from ΔB_i is easily obtained at the end of the observation interval from

$$\Delta TP = N/(d_{L,N}+\Delta_{L,N}) - N/d_{L,N}. \qquad (3.7)$$

The presented perturbation rules are summarized in the following algorithm for the general tandem queueing network. $\Delta \mathbf{B}$ is denoted by the row vector $(\Delta B_1, \Delta B_2, \ldots, \Delta B_N)$. Assume that the total number of servers is N. The tandem queueing network is started empty. That is, all the buffer storages are empty in the initial state. The total number of customers to be served is M. The performance measure of interest in this algorithm is the throughput.

Exact PA Algorithm For perturbed buffer storage ΔB

1) $j = 1$, $PNIT_{j,0} = \Delta_{i,0} = d'_{i,1} = 0$; $i = 1,2,\ldots,N$ and $l < 0$.

2) Calculate the amount of accumulated perturbations and the perturbed departure time.
For $i = 1,2,\ldots,N$
$\Delta_{i,j} = \Delta_{i,j-1} - (NIT_{i,j} + FOT_{i,j}) + PNIT_{i,j}$
$\underline{d}'_{i,j} = d_{i,j} + \Delta_{i,j}$.

3) Calculate the FO's time duration in the perturbed path
For $k = N,\ldots\ldots,2$
if $\underline{d}'_{k-1,j} < d'_{k,j-B_k-\Delta B_k}$, then
$PFOT_{k-1,j} = d'_{k,j-B_k-\Delta B_k} - \underline{d}'_{k-1,j}$
else
$PFOT_{k-1,j} = 0$
$\Delta_{k-1,j} = \Delta_{k-1,j} + PFOT_{k-1,j}$
$d'_{k-1,j} = d_{k-1,j} + \Delta_{k-1,j}$.

4) Evaluate the NI's time duration in the perturbed path ($d'_{0,l}$ is the lth arrival from the external environment.)
For $k = 1,2,\ldots,N$
if $d'_{k-1,j+1} > d'_{k,j}$, then
$PNIT_{k,j+1} = d'_{k-1,j} - d'_{k,j-1}$
else
$PNIT_{k,j+1} = 0$.

5) If $j = M$ customers, evaluate the performance measures such as ΔTP of Eq.(3.7).
else $j = j + 1$ and repeat step 2) - step 4).

This PA algorithm is an on_line algorithm in the sense that the exact departure time can be estimated via the presented PA rule only with the observed data, as seen in above examples. The exact perturbed departure time of a customer is easily obtained from this on_line PA algorithm. This PA algorithm can be applied to the case that multiple buffer storages are simultaneously perturbed since the suggested rules are utilized in several servers. This exact PA algorithm will be validated by several numerical examples in Section 5. In the next section, the proposed PA algorithm is utilized to solve the optimal buffer problem.

4. OPTIMAL DESIGN OF BUFFER STORAGE

The throughput and the cost of the buffer storage are a function of the buffer storage. The larger the the buffer storage size is, the more the throughput and the buffer storage cost is. But the throughput tend to be saturated as the size of the buffer storage approaches some value.. The throughput is denoted by TP(**B**) and the cost of the buffer storage(ST) is assumed to be given by

$$ST(\mathbf{B}) = \mathbf{C}\,\mathbf{B}^T$$

where $\mathbf{B}=(B_1,\ldots,B_N)$, $\mathbf{C} = (C_1,\ldots C_N)$ and C_i is the unit buffer cost of the server S_i. The optimal buffer storage must be determined to maximize the throughput while minimizing the buffer storage cost. Therefore, the performance measure(P) of interest in this section will be defined as

$$P(\mathbf{B}) = \alpha\, TP(\mathbf{B}) - ST(\mathbf{B}) \qquad (4.1)$$

where α is defined as a weighting factor between the throughput and the buffer storage cost, ST(**B**). Thus the optimal buffer storage is obtained by

$$\max_{all\ \mathbf{B}} P(\mathbf{B}). \qquad (4.2)$$

The vector $\mathbf{B}_0$ denotes the vector **B** for the buffer storage used in the nominal path. If the vector $\mathbf{B}_0$ is perturbed by $\Delta \mathbf{B}$, then from Eq. (4.1),

$$\begin{aligned}\Delta P(\Delta \mathbf{B}) &= P(\mathbf{B}_0+\Delta \mathbf{B}) - P(\mathbf{B}_0) \\ &= \alpha\ \Delta TP(\Delta \mathbf{B}) - ST(\Delta \mathbf{B})\end{aligned} \qquad (4.3)$$

where $\Delta TP(\Delta \mathbf{B}) = TP(\mathbf{B}_0+\Delta \mathbf{B}) - TP(\mathbf{B}_0)$ from Eq.(3.7). Let $\mathbf{B}^*$ be the solution of Eq.(4.2). If $\Delta \mathbf{B}^*$ is given by $(\mathbf{B}^* - \mathbf{B}_0)$, then it is easy to see that $\Delta \mathbf{B}^*$ is the optimal solution of the following equation,

$$\max_{all\ \Delta \mathbf{B}} \{\ \alpha\ \Delta TP(\Delta \mathbf{B}) - ST(\Delta \mathbf{B})\ \}. \qquad (4.4)$$

In optimization techniques, there are several methods such as the gradient method and the gradient-free method[11]. The gradient estimated by conventional PA techniques has been utilized in many optimization problems. Also, the gradient estimated by the Exact_PA technique can be utilized in the optimization problem. But the gradient method based on conventional PA techniques[1-3] as well as the Exact_PA technique requires the careful adjustment of weighting factors to guarantee convergence. The suggested gradient-free method can guarantee convergence when the buffer capacity of each server is finite and the performance measure, $\Delta P(\Delta \mathbf{B})$, with respect to the buffer storage is concave. The suggested gradient-free method is especially better than the gradient method in the sense that it guarantees convergence while avoiding the complicated weighting factor adjustement. And the rule for terminating the optimization algorithm is derived from the concave property of the performance measure, which is discussed later. The suggested gradient-free algorithm is simple and easily implemented since no gradient is necessary and the concave property is utilized.

It is known that the throughput with respect to the buffer storage increases monotonically but will be saturated as the buffer size approaches a certain value[9,10]. From the above property the throughput is likely to have the concave property with respect to the buffer storage in many cases. This tendency will be demonstrated in numerical examples discussed later. If TP(**B**) is concave, the change of the performance measure, $\Delta P(\Delta \mathbf{B})$ of Eq. (4.3), can be shown to be concave. This is possible due to the following fact. As TP(**B**) can be given by $[TP(\mathbf{B}_0)+\Delta TP(\Delta \mathbf{B})]$, $\Delta TP(\Delta \mathbf{B})$ has the concave property. $\Delta P(\Delta \mathbf{B})$ in Eq. (4.3) has the concave property as $CB(\Delta \mathbf{B})$ is an increasing function.

We present the following on_line optimization algorithm for the optimal allocation of the buffer storage by using the Exact_PA algorithm. Consider the sequence

$\{\Delta\mathbf{B}_k\}$ where all points in $\{\Delta\mathbf{B}_k\}$ are generated from the suggested optimization algorithm and $\Delta\mathbf{B}_k$ $\Delta\mathbf{B}$. And let $\mathbf{y}_j$ $\Delta\mathbf{B}$. The direction $\mathbf{d}_j(j=1,..,N)$, denotes a vector of zeros except for a one at the j-th position. It is assumed that $\Delta P(.)$ is obtained from the Exact_PA algorithm and Eq. (4.3).

Gradient-free Optimization Algorithm with Exact PA

1) initial state :
   Let $\mathbf{d}_1,\ldots,\mathbf{d}_N$ be the directions.
   Choose a starting point $\Delta\mathbf{B}_1$. Let $\mathbf{y}_1 = \Delta\mathbf{B}_1$ and $k = j = 1$.
2) obtain $\Delta P(\mathbf{y}_j + \mathbf{d}_j)$ and $\Delta P(\mathbf{y}_j)$ from the Exact_PA algorithm and Eq. (4.3).
3) If $\Delta P(\mathbf{y}_j + \mathbf{d}_j) > \Delta P(\mathbf{y}_j)$, let $\mathbf{y}_{j+1} = \mathbf{y}_j + \mathbf{d}_j$ and go to step 4.
   else obtain $\Delta P(\mathbf{y}_j - \mathbf{d}_j)$ from the Exact_PA algorithm and Eq. (4.3).
   if $\Delta P(\mathbf{y}_j - \mathbf{d}_j) > \Delta P(\mathbf{y}_j)$, let $\mathbf{y}_{j+1} = \mathbf{y}_j - \mathbf{d}_j$ and go to step 4.
   else let $\mathbf{y}_{j+1} = \mathbf{y}_j$ and go to step 4.
4) If $j < N$, $j = j + 1$ and go to step 2.
   else if $\Delta P(\mathbf{y}_{N+1}) > \Delta P(\Delta\mathbf{B}_k)$, go to step 5.
   else stop. That is, $\Delta P(\Delta\mathbf{B}_k)$ is greater than all $\Delta P(\Delta\mathbf{B}_k + \mathbf{d}_j)(j=1,..N)$. Thus, $\Delta\mathbf{B}_k$ is the optimal solution.
5) Let $\Delta\mathbf{B}_{k+1} = \mathbf{y}_{N+1}$, and $\mathbf{y}_1 = \Delta\mathbf{B}_{k+1}$, and $k = k+1$ and $j = 1$, and go to step 2.

In the optimization problem utilizing the gradient method, weighting factors should be carefully chosen so that it guarantees convergence. In this algorithm, no weighting factors exist. The advantages of the algorithm presented are its simplicity and ease of implementability. Because the Exact_PA technique provides the exact estimation of the performance measure, the above algorithm gives the global optimal solution if the performance measure of interest is concave. Otherwise, this algorithm may give the local optimal solution. We consider several numerical examples for the validation of the above optimization algorithm in the next section.

5. EXAMPLES FOR Exact_PA ALGORITHM and OPTIMAL BUFFER DESIGN

In Example 1, it will be shown that the estimated value by the Exact_PA technique is the same as the actual value which is obtained by the brute force simulation. The Exact_PA method will be shown to be better than the Finite Perturbation Analysis(FPA) method[5]. It will be also shown that the throughput in this example has the concave property with respect to the buffer storage.

Example 1. Nominal System Model :
number of servers : 3

	S_1	S_2	S_3
buffer storage size :	1	3	2
mean service time :	1	2	3
mean arrival time :	1		

The number of customers served in this example is 9000. The service time of each server is assumed to be exponentially distributed. The nominal path is generated by the simulation with the above parameters. Fig. 4.1 shows the perturbed throughput, ΔTP, obtained from the Exact_PA method and the FPA method when the buffer storages, B_2 and B_3, are simultaneously perturbed. From this figure, we can observe that the actual value can be obtained via the Exact_PA algorithm and that the throughput has the concave property with respect to the buffer storage. As can be seen in the Fig. 4.1, the Exact_PA method is better than the FPA method. The Exact_PA method takes almost the same simulation time as the FPA method. But the Exact_PA method provides the exact value of the performance measure while the FPA method provides the estimated value of it.

Next, two numerical examples will be considered for the validation of the presented optimization algorithm. As in [1], the same cost is assigned to each buffer storage in Example 2. The number of customers to be served in these examples is 9000. The service time of each server is assumed to be exponentially distributed.

Example 2. Nominal System Model :
number of servers : 4

		S_1	S_2	S_3	S_4
buffer storage size :	1	1	1	1	
mean service time :		2	3	2	3
mean arrival time :		1			
buffer cost,C_i :		3	3	3	3
weighting factor for TP,α :		1000			

The nominal path is generated by simulation with the above parameters. From the presented optimization algorithm, the maximum ΔP and the perturbed optimal buffer storage($\Delta\mathbf{B}^*$) are given as follows:
$\max(\Delta P) = 58.16$ and $\Delta B^*_1=0$, $\Delta B^*_2=3$, $\Delta B^*_3=4$ and $\Delta B^*_4=5$
(that is, $B^*_1=1$, $B^*_2=4$, $B^*_3=5$ and $B^*_4=6$)
Fig. 4.2 shows the values obtained from the brute force simulation when B_2, B_3 and B_4 are simultaneously perturbed. From the brute force simulation, the following optimal buffer storages,
$\Delta B^*_1=0$, $\Delta B^*_2=3$, $\Delta B^*_3=4$ and $\Delta B^*_4=5$,(that is, $B^*_1=1$, $B^*_2=4$, $B^*_3=5$ and $B^*_4=6$)
are obtained. These values are identical to ones obtained from the suggested optimization algorithm. In Example 3, we will consider the case that the cost of each unit buffer storage is different.

Example 3. Nominal System Model :
number of servers : 3

	S_1	S_2	S_3
buffer storage size :	1	3	2
mean service time :	2	2	3
mean arrival time :	1		
buffer cost,C_i :	1	3	2
weighting factor for TP,α :	500		

The nominal path is generated by simulation with the above parameters. The maximum ΔP and the perturbed optimal buffer storage($\Delta\mathbf{B}^*$),
$\max(\Delta P) = 1.129$ and $\Delta B^*_1=0$, $\Delta B^*_2=0$ and $\Delta B^*_3=1$,
(that is, $B^*_1=1$, $B^*_2=3$ and $B^*_3=3$)
are obtained from the presented optimization algorithm. In the case that B_2 and B_3 are simultaneously perturbed, the values obtained from the brute force simulation is shown in Fig. 4.3, from which we can get the optimal buffer storage,
$\Delta B^*_1=0$, $\Delta B^*_2=0$ and $\Delta B^*_3=1$
(that is, $B^*_1=1$, $B^*_2=3$ and $B^*_3=3$).
This simulation validates the suggested optimization algorithm.

Above examples show that the optimal buffer storage is obtained from the proposed optimization algorithm regardless of the cost of each buffer storage and the buffer storage size.

6. CONCLUSION

In this paper, we have attempted to provide the exact perturbation analysis technique with respect to the buffer storage and to solve the optimization problem via the suggested perturbation analysis technique in the tandem queueing network.

Discrete event dynamic equations and basic properties of Full Out(FO) and No Input(NI) with respect to the buffer storage in the tandem queueing network are presented. The perturbation rules with respect to the buffer storage are derived from these basic properties and dynamic equations. The on_line Exact_PA algorithm is derived from these rules. The performance measure, such as the throughput, is obtained from the proposed exact perturbation analysis algorithm for the case of the perturbation of multiple discrete parameters. This on_line exact perturbation analysis algorithm can be easily implemented. The optimal buffer storage problem is presented by introducing a performance measure consisted of the throughput and the buffer storage cost. It is shown that the optimal buffer storage can be easily obtained from the proposed gradient-free optimization algorithm regardless of the buffer storage cost of each server and the size of the buffer storage. The Exact_PA algorithm and the gradient-free optimization algorithm are validated by numerical examples.

The global optimal buffer storage is guaranteed in the suggested gradient-free optimization algorithm if the throughput is concave. Otherwise, it may give the local optimal buffer storage. There are many cases in tandem queueing networks, where the throughput is believed to be concave with respect to the buffer storage. It is necessary to investigate general conditions under which the throughput is concave.

The Exact_PA technique may be extended to general queueing networks with respect to the buffer storage. And the method suggested in this paper may also be applied to the performance measures with respect to other discrete parameters such as the number of servers.

REFERENCES

[1] Y.C.Ho, M.A.Eyler, and T.T. Chien,"A gradient technique for general buffer storage design in a production line," *Int. J. Prod. Res.*,vol.17,pp.557 -580,1979.

[2] Y.C. Ho and X.Cao,"Perturbation Analysis and Optimization of Queueing Networks,"*J. Optimiz. Theory Appl.*,vol.40,pp.559-582,1983.

[3] X.Cao and Y.C.Ho,"Estimating the Sojourn Time Sensitivity in Queueing Network using Perturbation Analysis,"*J. Optimiz. Theory Appl.*,vol.53,pp.353 -375,1987.

[4] R.Suri,"Infinitesimal Perturbation Analysis of Discrete Event Dynamic System - A General Theory," *J. of ACM.*,vol.34,pp.686-717,1987.

[5] Y.C.Ho, X.Cao, and C. Cassandras,"Infinitesimal and Finite Perturbation Analysis for Queueing Network," *Automatica*,vol.19,pp.439-445,1983.

[6] C.G.Cassandras,"On-Line Optimization for a Flow Control Strategy," *IEEE Trans.AC.*, vol.AC -32,pp.1014-1017,1987.

[7] C.G.Cassandra and S.G.Stickland,"Observable Augmented Systems for Sensitivity Analysis of Markov and Semi-Markov Processes," *IEEE Trans. AC.*,vol.AC-34,pp.1026-1037,1989.

[8] X.Cao and Y.Dallery,"Sensitivity Analysis of Closed Queueing Networks : An Operation Approaches,"*Proc. of ACC.*,pp.2034-2039,1986.

[9] J.Wijngaard,"The Effect of Interstage Buffer Storage on the Output of Two Unreliable Production Units in Series, with Different Production Rates," *AIIE TRANSACTIONS*, vol.11,pp.42-47,1979.

[10] K.Okaura, H.Yamashina, and M.Ishihara,"Computer Simulation of the Effect of Buffer in the Multi -stage Line Systems,"*Precision Instrument*,vol.43,pp.45 -50,1977.

[11] M. Bazaraa and C.M. Shetty, Nonlinear Programming,John Wiley & Sons,New York,1979.

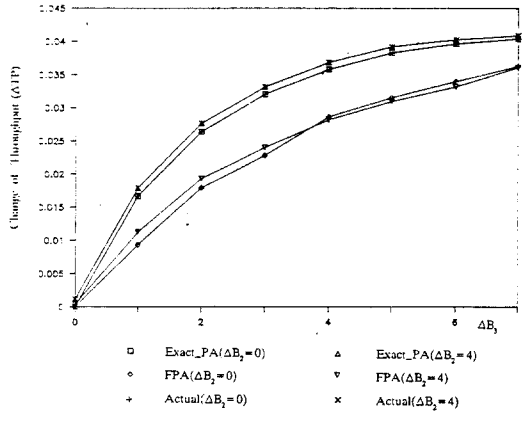

Fig. 4.1 Change of Throughput w.r.t. perturbed buffer storage in Example 1

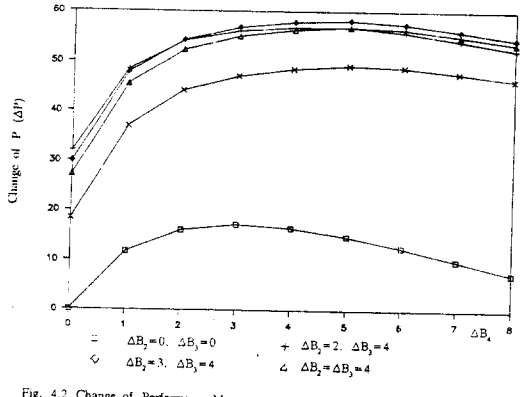

Fig. 4.2 Change of Performance Measure w.r.t. perturbed buffer storage in Example 2

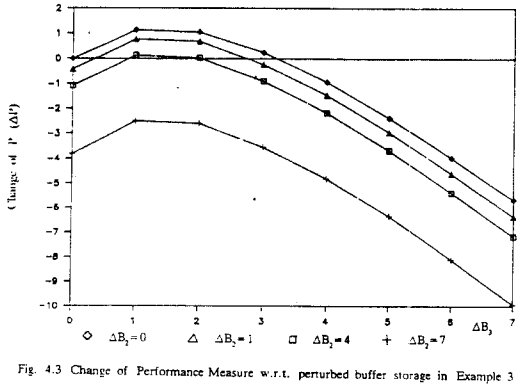

Fig. 4.3 Change of Performance Measure w.r.t. perturbed buffer storage in Example 3