A regression model for predicting the response time of a disc I/O system*

H. G. Perros

Dept. of Quantitative Methods, University of Illinois at Chicago Circle, Chicago, Illinois, USA

In a disc I/O system, the execution time of a read or write request depends partially upon other requests being executed at the same time. This concept is developed in this paper, and it is then used together with some standard results from queueing theory to produce a simple nonlinear regression model for predicting the response time of a disc I/O system. The model is of wide applicability although it was developed originally for a specific system.

(Received March 1979)

1. Introduction

Disc I/O systems are of interest as their performance affect the throughput of the system they support. Various studies related to disc I/O systems have been reported in the literature. For example, different disc scheduling policies, the use of multiple arm services and other strategies aiming at the reduction of the disc I/O response time have been studied (Cussens, 1971; Fife, 1965; Frank, 1966; Gotlieb and MacEwen, 1973). Models valid under various assumptions for estimating the response time of a disc I/O system have also been reported (Perros, 1977; Seaman, 1966). In this paper, a nonlinear regression model for predicting the mean response time of a disc I/O system is described. The response time is defined as the *file access time*, which is the time elapsing from the moment a read or write request is issued to the moment it is fully executed. The regression model constructed is not a result of purely fitting empirical data. Rather, it is a combination of an analytic approach and empirical techniques. In particular, the model is based on an analytic model which is easily related to the physical system, yet the final form of it is derived using nonlinear regression techniques. The model is useful when evaluating the performance of an existing disc I/O system. It does not require knowledge of the seek, search and transmission distributions. The model was developed originally for the disc I/O system which supports the real time seat reservation system, Astral, of Aer Lingus-Irish. It is applicable to any disc I/O system.

In the following section a brief description of the Astral disc I/O system is given. In Section 3, the concept of *partially shared servicing* is introduced. This deals with the execution time of a read or write request. It is demonstrated that this time depends *partially* upon other requests being executed at the same time. In Section 4, this concept is used together with some standard results from queueing theory to develop the regression model.

2. Queueing in the Astral disc I/O system

The Astral disc I/O system at the time of the investigation consisted of two IBM 2314 disc storage facilities (IBM, 1971). Each facility consisted of nine discs, seven of which were active at any time, and a control unit. The two control units were served by one channel. The same channel also served six magnetic tape drives. The system operated in buffer mode. This meant that the two control units were not driven entirely by the channel, but were to a great extent autonomous units. The channel simply undertook the transmission of data, along with channel programs appropriately chained, from core to the appropriate control unit. The rest of the steps necessary for the execution of a read or write request were initiated and supervised by the control unit itself.

Read or write requests (hereafter referred to as requests) issued by application programs or system programs were queued so that all requests that were to be carried out on the same disc were placed in the same queue, known as module queue. There were, therefore, as many module queues as active discs, i.e. 14. The service discipline of these queues was first-in-first-out. A request became eligible for execution when it reached the top of its module queue. Only one request from each module queue was executed at any time. Therefore, the maximum number of requests that could be executed at any time was equal to the number of module queues, i.e. 14. The execution of the requests was carried out by a number of servers, namely the channel, the control units and the 14 discs. We shall refer to the collection of these servers as the service mechanism, and to the execution time of a request as the service time. In **Fig. 1** the module queues and the service mechanism are shown. The major components of the service provided by this mechanism are shown in **Fig. 2.** The file access time of a request, therefore, consisted of the time the request spent waiting in the appropriate module queue and of its service time. All active data sets and programs were distributed over the 14 discs so that all discs were equally utilised. In view of this, it was assumed that all module queues were symmetrical. Therefore, the mean file access time of the disc I/O system coincided with the mean waiting time of a request in any of the above module queues, service included.

3. Partially shared servicing

We now proceed to examine further the type of service a request received. This service consisted of various services offered by the channel, the control units, and the discs. As mentioned in Section 2 the system was organised so that only one request from each module queue was in the service at any time. Because of this, any disc was sought by at most one request at any time, as opposed to the channel and the control units which were sought by more than one request at any time. A particular request, therefore, had to compete with other requests for services provided by the channel and the control units but not for the service provided by a disc. In view of this, it was

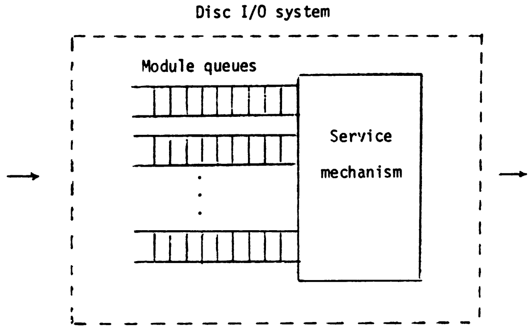

Fig. 1 The module queues and the service mechanism

*This research was partially supported by a grant from the Research Board, University of Illinois at Chicago Circle.

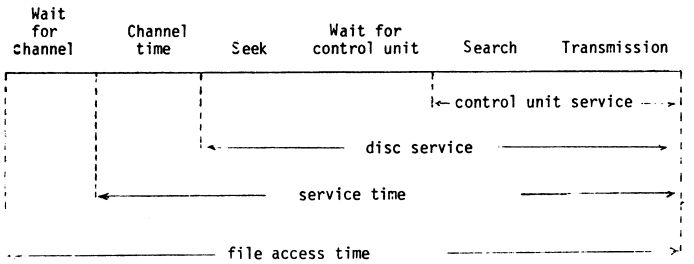

Fig. 2 The major time components of the file access time

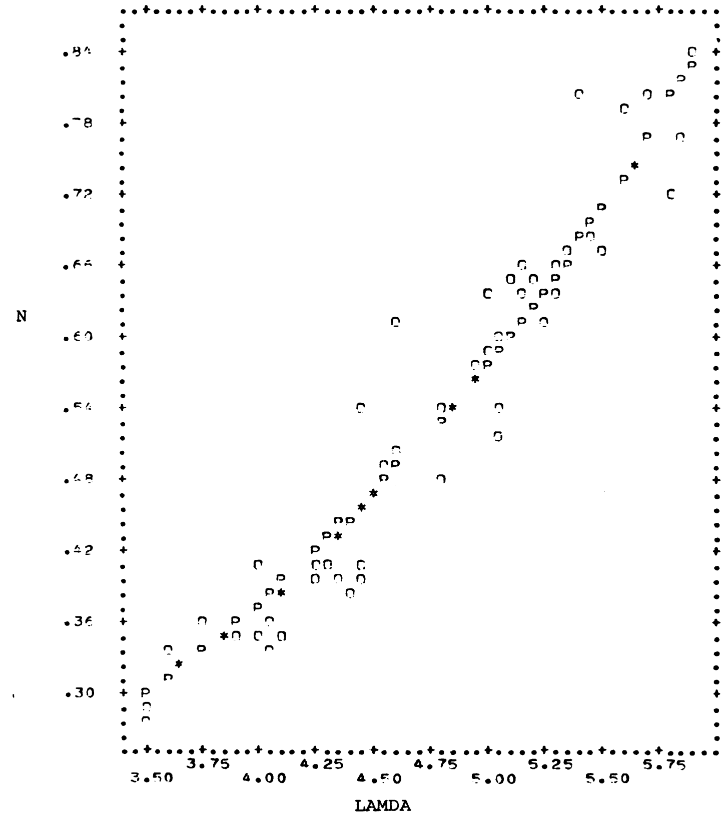

Fig. 3 A plot of variable λ versus the predicted and observed variable *N*

reasonable to assume that the total service a request received depended *partially* upon other requests in service. We define this type of servicing as *partially shared servicing*. Let μ be the rate at which a request is served at a time no other requests are in service. Let λ be the rate at which requests are issued. Then, the rate μ^* at which requests are served at any time can be expressed as follows

$$\mu^* = \mu - \alpha\lambda ,$$

where α is a non-negative quantity that expresses the interference due to other requests in service. A lower and an upper bound of α are derived below. These bounds represent two extreme situations of the partially shared servicing.

1. *Lower bound of* α

The lower bound of α can be obtained by considering the case in which a request suffers a service which is entirely *independent* from other requests in service. In this case, the queueing structure shown in Fig. 1 is reduced to 14 independent queues in parallel served by independent servers. Obviously, $\mu^* = \mu$ and, therefore, we have $\alpha = 0$.

2. *Upper bound of* α

An upper bound of α can be obtained by considering the case in which a request suffers a service which is entirely *dependent* upon other requests in service, In particular, let us consider any request in service. (We shall refer to this request as the 'tagged unit'.) If there is no other request in service, then the tagged unit is processed at the rate μ. If there are k-1 requests constantly being served with it, then it is processed at the rate μ/k. Define J to be the average number of requests in service conditioned upon the tagged unit being in service. Then, the rate at which the tagged unit is processed at any time is $\mu^* = \mu/J$.

Let τ_i be the traffic intensity of the ith module queue. Clearly, the probability that this particular queue is busy at any time is τ_i. Also, we have that $\tau = \tau_i$, for $i = 1, 2, \ldots, 14$, seeing that all module queues are symmetrical, and $\tau = \lambda/14\mu^*$ or

$$\tau = (\lambda/14)(J/\mu) . \qquad (3.1)$$

Now, in Section 2 it was mentioned that only the top member of each module queue is in service. Therefore, the number of requests in service at any time coincides with the number of busy module queues. The average number of requests in service conditioned upon the tagged unit is in service is therefore,

$$J = 1 + 13\tau . \qquad (3.2)$$

Using expression (3.1) in (3.2) we obtain

$$J = 1/(1 - (13\lambda/14\mu)) . \qquad (3.3)$$

Now, seeing that $\mu^* = \mu/J$ we have

$$\mu^* = \mu - (13/14)\lambda$$

and, therefore, we obtain $\alpha = 13/14$.

4. The regression model

The mean file access time of the system under investigation was defined in Section 2 to be the mean time a request spends in a module queue. Let us consider, therefore, any module queue. An expression for the mean number, N, of requests in the queue (including the one in service) can be obtained by treating the queue as an $M/G/1$ system. The quantity N is then given by the

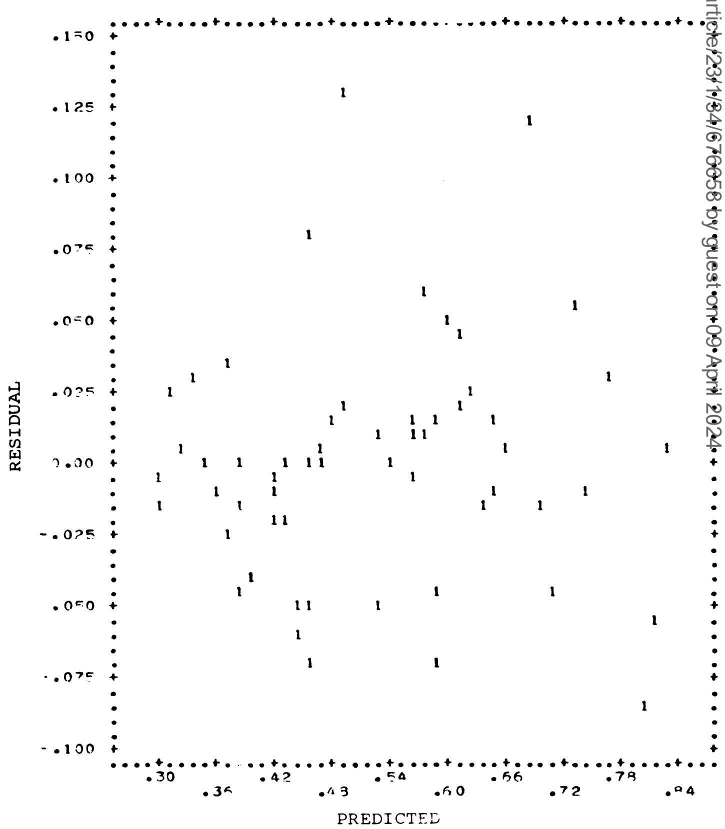

Fig. 4 A plot of the predicted variable *N* versus residuals

Pollaczek-Khintchine formula (see Lee, 1968, page 34). We have

$$N = \tau^2(1 + c^2)/2(1 - \tau) + \tau \,, \qquad (4.1)$$

where c^2 is the coefficient of variation, and τ is the traffic intensity of the queue, given by $\tau = \lambda'/\mu^*$ or

$$\tau = \lambda'/(\mu - \alpha(14\lambda')) \,, \qquad (4.2)$$

where $\lambda' = \lambda/14$. The parameters μ, α and c^2 were estimated from performance data using the nonlinear regression program of the Biomedical Computer Programs. The performance data were collected through a file collector package and related module queue sizes to input rates. The independent variable was the input rate λ' and the dependent variable was the module queue size N, which except for random fluctuation in N, was related to λ' by expression (4.1). The estimates obtained were: $\hat{\mu} = 20{\cdot}42$, $\hat{\alpha} = 0{\cdot}058$ and $\hat{c}^2 = 1{\cdot}00$. The residual sum of squares was 0·098. A line printer plot of λ' versus the predicted and observed variable N is given in **Fig. 3.** A plot of the predicted variable N versus residuals is given in **Fig. 4.**

In the light of these estimates the following simpler model was tested. Parameter α was taken to be equal to $1/14$ ($= 0{\cdot}07$) and c^2 was taken to be equal to one. Expression (4.1) was then reduced to the following simple expression

$$N = \tau/(1 - \tau) \,, \qquad (4.3)$$

where

$$\tau = \lambda'/(\mu - \lambda') \,. \qquad (4.4)$$

(Expression (4.3) gives the mean number of units in a $M/M/1$ system.) The obtained estimate was: $\hat{\mu} = 18{\cdot}6$. The residual sum of squares was 0·191. The obtained plots of λ' versus the predicted and observed values of N, and of the predicted values of N versus the residuals are not given here as they were similar to those shown in Figs. 3 and 4.

The mean file access time, R, can be easily obtained by employing Little's relation. We have $R = N/\lambda'$ or using expressions (4.3) and (4.4) we obtain

$$R = 1/(\mu - (\lambda/7)) \,,$$

where λ is the rate at which requests are issued, and μ was estimated to be equal to 18·6.

The quantity μ was defined as the rate at which a request is served at a time no other requests are in service. It was possible to calculate this quantity by simply adding the mean channel time, mean seek time, and mean search and transmission time. Information about each individual mean time was readily available, except for the mean seek time. This was easily estimated by simulation. The quantity μ was calculated to be 18·9 per second. We note that the estimate $\hat{\mu} = 18{\cdot}6$ compares well with this exact figure.

5. Conclusion

In this paper a simple nonlinear regression model for predicting the mean file access time of a disc I/O system was presented. The model was not constructed by purely fitting empirical data. Rather, it was a combination of an analytic model and empirical techniques. The analytic model was based on the idea that the execution time of a read or write request depends partially upon other requests in service. This idea is conceptually very simple and it has wider applicability than that considered here (*cf.* Foster and Perros, 1979). The combination of an analytic approach and empirical techniques yields useful results and it provides an important complement to analytic tools (*cf.* Svobodova, 1976).

References

Cussens, C. J., Mosses, R. N. and Pretty, J. M. (1971). Report on the Modelling Experiment, ICL SEDD Report TRG/S71.

Fife, D. W. and Smith, J. L. (1965). Transmission capacity of Disk Storage Systems with concurrent Arm Positioning, *IEEE Trans.*, EC-14 Vol. 4, pp. 575-582.

Foster, F. G. and Perros, H. G. (1979). Hierarchical Queue Networks with Partially Shared Servicing, *J. Opl. Res. Soc.*, Vol. 30, pp. 157-166.

Frank, H. (1966). Analysis and Optimization of Disk Storage Devices for Time-Sharing Systems, *JACM*, Vol. 16, pp. 602-620.

Gotlieb, C. C. and MacEwen, G. H. (1973). Performance of Movable-Head Disk Storage Devices, *JACM*, Vol. 20, pp. 604-623.

IBM (1971). System/360 Components Description-2314 Direct Access Storage Facility, form A26-3599. IBM Syst. Ref. Library.

Lee, A. M. (1968). *Applied Queueing Theory*, MacMillan, London.

Perros, H. G. (1977). Modelling a Disc I/O System: A Queue Network Approach, *EJOR*, Vol. 1, pp. 139-145.

Seaman, P. H., Lind, R. A. and Wilson, T. L. (1966). An Analysis of Auxiliary-Storage Activity, *IBM Systems J*, Vol. 5, pp. 158-170.

Svobodova, L. (1976). *Computer Performance Measurement and Evaluation Methods: Analysis and Applications*, American Elsevier.