DEAD-TIME LOSSES IN A
BUFFERED DATA RECORDING SYSTEM

H. Øverås^o
CERN, Geneva, Switzerland

ABSTRACT

Expressions are derived for the average fraction of events that will be stored (i.e. not lost) when completely random events (i.e. with exponentially distributed interarrival times) are fed through a buffer with capacity n events into a data store with constant dead-time (service time). The data store can be synchronized with a clock of any cycle time. Transfer times are neglected, except between the buffer and the data store.

Geneva - 7 April 1972
(Submitted to Nuclear Instrum. Methods)

1. INTRODUCTION

If completely random events of some sort with a mean rate λ are counted directly by a device with a dead-time δ , then the fraction R of events counted is given by the well-known formula

$$R = \frac{1}{\lambda\delta + 1} \quad . \quad (1)$$

The fraction of events lost is $1 - R$. Instead of just counting, the device may be storing a coded description of the event. Then δ is the time it takes the device, the data store, to receive and store the description and get ready for a new event.

In this paper more general expressions are derived for the fraction of stored events, applicable when the description of up to n events can be kept temporarily in a buffer while waiting to enter the data store. The problem seems not to have been treated in the literature, maybe because it was of little interest as long as reliable fast electronics was not readily available.

Figure 1 shows a buffer of capacity n events, or, in other words, containing n event buffers. The events occur completely at random with a mean rate λ . The mean rate of the stored events is $\lambda_{\text{stored}} = \lambda - \lambda_{\text{lost}}$. Obviously, $\lambda_{\text{stored}} \leq \lambda$ and also $\leq 1/\delta$. The first limit is approached for small λ and the second for large λ . If $n = 0$ there is no buffer. We shall ignore all transfer times in the system, except between the buffer and the data store. We assume the buffer is asynchronous, so it responds without delay to any request for input (if not full) or output (if not empty) of an event. Events occurring when the buffer is full are lost. The transfer of an event from the buffer into the data store can start only at given times separated by τ , the cycle time of the data store. The free capacity of the buffer increases by one event buffer a time $\eta \leq \delta$ after the transfer has started. We call η the buffer overlap. Owing to the synchronization, the effective dead-time of the data store, which we shall call ξ , can be different from δ . In fact,

$$\xi = \left\{ \text{lowest integer} \geq \frac{\delta}{\tau} \right\} \tau \quad . \quad (2)$$

By definition then, τ , δ , and η are $\leq \xi$. Figure 2 illustrates the case of $n = 1$ and the timings involved.

As we shall soon see, the decisive quantities are the average number of events occurring during these times, so we define

$$x = \lambda \xi, \quad t = \lambda \tau = \frac{\tau}{\xi} x, \quad y = \lambda \eta = \frac{\eta}{\xi} x. \quad (3)$$

The probability of v events occurring in a given time interval, for example ξ , is given by the Poisson distribution :

$$f_v = \frac{x^v}{v!} e^{-x}, \quad \sum_{v=0}^{\infty} f_v = 1. \quad (4)$$

The aim is now to find a generalization of Eq. (1):

$$R = R_n(t, y) = \frac{\lambda_{\text{stored}}}{\lambda} \quad (5)$$

as a function of n , t , y and, of course, x which we do not indicate explicitly.

Equation (1) then expressed the special case

$$R_0(0, 0) = \frac{1}{x + 1}.$$

2. TWO SPECIAL CASES

Before we attack the problem in general, we shall consider the special case of $y = 0$, i.e. no buffer overlap, and only for two special values of t , namely 0 and x . The value $t = 0$ corresponds to what is sometimes called asynchronous data store, and $t = x$ to a synchronous data store, although the latter term could equally well be used for $0 < t < x$.

2.1 Asynchronous data store and no overlap, $t = y = 0$, $\xi = \delta$, $x = \lambda \delta$

The system has no-long term time dependence, and in this case also no periodicity. We can then introduce, as is common in queuing theory¹⁾, a time-independent long-term average of the probability of each of its possible states:

<u>State</u>	<u>Average probability</u>
No event in the buffer and store free	P_0
i events in the buffer and store busy	P_{i+1}

States with events in the buffer and the store free are not possible. Obviously, $0 \leq i \leq n$.

As normalized probabilities the P_j obey

$$\sum_{j=0}^{n+1} P_j = 1 \quad . \quad (6)$$

Further equations are obtained by comparing an arbitrarily chosen pair of times, separated by δ . Since the P_j were time-independent, and a transfer to the data store must have taken place sometime during this interval unless there was no event in the buffer and no event occurred, we have

$$\begin{aligned} P_0 &= P_0 f_0 + P_1 f_0 \\ P_1 &= P_0 f_1 + P_1 f_1 + P_2 f_0 \\ &\dots\dots\dots \end{aligned}$$

or in general the n equations

$$P_{j-1} = P_0 f_{j-1} + \sum_{i=0}^{j-1} P_{j-i} f_i \quad \text{for } j = 1, \dots, n \quad . \quad (7)$$

If $n = 0$, there are no equations of this type; f_i is given by Eq. (4).

Finally we have two equations involving R [defined by Eq. (5)]:

$$R = \frac{1 - P_0}{x} \quad , \quad (8)$$

and

$$R = 1 - P_{n+1} \quad . \quad (9)$$

To solve these equations we observe that the first k of the homogenous Eqs. (7) allow us to determine a quantity A_k defined by

$$\sum_{j=0}^k P_j = A_k P_0 \quad k = 0, \dots, n \quad . \quad (10)$$

Then from Eqs. (6), (8), and (9):

$$R = \frac{1 - P_0}{x} = 1 - P_{n+1} = \sum_{j=0}^n P_j = A_n P_0$$

or

$$R_n(0,0) = R = \frac{1}{x + A_n^{-1}} \quad . \quad (11)$$

In order to find A_k we sum on both sides of Eq. (7):

$$\sum_{j=1}^k P_{j-1} = P_0 \sum_{j=1}^k f_{j-1} + \sum_{j=1}^k \sum_{i=0}^{j-1} P_{j-i} f_i$$

or

$$\sum_{j=0}^{k-1} P_j = P_0 \sum_{j=0}^{k-1} f_j + \sum_{i=0}^{k-1} \sum_{j=i+1}^k P_{j-i} f_i = \sum_{i=0}^{k-1} f_i \sum_{j=0}^{k-i} P_j \quad .$$

Using now Eq. (10), this becomes

$$A_{k-1} = \sum_{i=0}^{k-1} f_i A_{k-i} \quad , \quad (12a)$$

or in a form more convenient for computation:

$$f_0 A_k = A_{k-1} - \sum_{i=1}^{k-1} f_i A_{k-i} \quad 1 \leq k \leq n \quad . \quad (12b)$$

Equation (10) gives $A_0 = 1$, and starting with this value we can successively compute A_k for $k = 1, \dots, n$. Equation (10) moreover states that $A_k = 0$ for $k < 0$.

It is also useful to have an explicit expression for A_k . The solution, which can be readily verified by substitution into Eq. (12a) and carrying out one summation, is

$$A_k = \sum_{l=0}^k \frac{(l-k)!}{l!} e^{(k-l)x} x^l \quad \text{for } 0 \leq k \leq n. \quad (13)$$

To obtain the correct value $A_0 = 1$ from Eq. (13) we must interpret 0^0 as 1.

The first few A_k are:

$$\begin{aligned} A_0 &= 1 \\ A_1 &= e^x \\ A_2 &= e^{2x} - xe^x \\ A_3 &= e^{3x} - 2xe^{2x} + \frac{x^2}{2} e^x \\ A_4 &= e^{4x} - 3xe^{3x} + 2x^2 e^{2x} - \frac{x^3}{6} e^x. \end{aligned} \quad (14)$$

It may be useful to observe that for $x = 1$ and $n \geq 2$, $2n + \frac{2}{3}$ is a good approximation for A_n , e.g. $4 + \frac{2}{3} \approx 4.667$ while $A_2 \approx 4.671$.

Figure 3 shows amongst other curves $R_n(0,0)$ as a function of x for some values of n .

For large x $A_n \rightarrow e^{nx}$, so

$$\lim_{x \rightarrow \infty} R_n(0,0) = \lim_{x \rightarrow \infty} \frac{1}{x + e^{-nx}} = \lim_{x \rightarrow \infty} \frac{1}{x} = 0, \quad (15)$$

as one would expect.

For small x we can find a series expansion:

$$A_k = \sum_{\mu=0}^{\infty} C_{\mu}(k) x^{\mu}, \quad (16a)$$

where

$$C_{\mu}(k) = 1 - \sum_{\ell=k+1}^{\mu} \frac{(-1)^{\ell} (k - \ell)^{\mu}}{\ell! (\mu - \ell)!} , \quad (16b)$$

$$\text{i.e. } C_{\mu}(k) = 1 \quad \text{for } \mu \leq k .$$

We then have

$$A_k = \frac{1 - x^{k+1}}{1 - x} + x^k \sum_{\omega=1}^{\infty} C_{k+\omega}(k) x^{\omega} , \quad (17)$$

which for $x < 1$ gives

$$\lim_{k \rightarrow \infty} A_k = \frac{1}{1 - x} \quad (18)$$

or, from Eq. (11),

$$\lim_{n \rightarrow \infty} R_n(0,0) = \frac{1}{x + 1 - x} = 1 . \quad (19)$$

That is, for $x < 1$ and $n \rightarrow \infty$ no events are lost. The buffer smooths out the data stream completely before it enters the data store.

For finite n we find

$$R_n(0,0) = 1 - \frac{x^{n+1}}{(n+1)!} - \frac{x^{n+2}}{(n+2)!} \left[2^{n+2} - 3(n+2) \right] + \dots . \quad (20)$$

This expansion illustrates the behaviour of $R_n(0,0)$ for very small x , but converges too slowly to be of much practical use.

2.2 Synchronous data store and no buffer overlap, $t = x$ and $y = 0$

In this case the cycle $\tau = \delta = \xi$ of the data store must be expected to create a periodic behaviour also in the average probabilities. However, we need only to know these probabilities at a particular phase of each period, namely immediately after a transfer to the data store has or could have started. When we redefine P_j in this way, and consider two such neighbouring phases, Eqs. (7) still hold, and so do Eqs. (6) and (8), whereas Eq. (9) must be replaced by

$$P_{n+1} = 0 . \quad (21)$$

This is because the transfer takes no time, so there is zero probability of having n events in the buffers immediately after a transfer to the store has started.

Equations (6), (10), and (21) give $A_n P_0 = 1$, which with Eq. (8) gives

$$R_n(x, 0) = R = \frac{1 - A_n^{-1}}{x} . \quad (22)$$

Here A_n is the same as before, and is given by one of Eqs. (12) or (13).

Since $A_0 = 1$, $R_0(x, 0) = 0$, so the synchronous data store cannot work without a buffer.

Figure 3 shows that $R_1(x, 0)$ lies between $R_0(0, 0)$ and $R_1(0, 0)$.

Using Eq. (18) we get for $x < 1$:

$$\lim_{n \rightarrow \infty} R_n(x, 0) = \frac{1 - 1 + x}{x} = 1 , \quad (23)$$

as for the synchronous case.

For large x we have again, as expected,

$$\lim_{x \rightarrow \infty} R_n(x, 0) = \lim_{x \rightarrow \infty} \frac{1 - e^{-nx}}{x} = \lim_{x \rightarrow \infty} \frac{1}{x} = 0 . \quad (24)$$

For small x we get the series expansion

$$R_n(x, 0) = 1 - \frac{x^n}{(n+1)!} - \frac{x^{n+1}}{(n+2)!} \left[2^{n+2} - 3(n+2) \right] + \dots . \quad (25)$$

We see that the lowest power of x is n rather than $n + 1$ as in the expansion of $R_n(0, 0)$, Eq. (20).

3. A MORE GENERAL CASE

We shall now consider the case of arbitrary τ and η , whose meaning was explained in Fig. 2. ξ is related to τ and δ by Eq. (2), and as before $x = \lambda\xi$, $t = \lambda\tau$, and $y = \lambda\eta$. We shall use the abbreviations

$$f_v = \frac{x^v}{v!} e^{-x}, \quad g_v = \frac{y^v}{v!} e^{-y}, \quad h_v = \frac{(x-y)^v}{v!} e^{-(x-y)}, \quad q_v = \frac{t^v}{v!} e^{-t}. \quad (26)$$

Whenever $v < 0$ occurs, the corresponding Poisson probability is zero.

In contrast to what we did in Section 2, we shall not introduce state probabilities defined either at all times (Section 2.1) or at fixed equidistant times (Section 2.2), but rather at times related to a given phase of each storage operation actually taking place. This seems more convenient when many parameters are involved, but it means that relations such as (8) or (9) cannot be directly formulated.

Let Q_j be the average (over very many storage operations) probability of a possible state j of the system at the moment immediately after a transfer to the data store has started. In fact, the only states possible are j events in the buffer, with $1 \leq j \leq n$. We define $Q_j = 0$ for all other j , also $j = 0$ since it is here (unlike in Section 2.2) more convenient to define "immediately after" as $< \eta$ even in the limit of $\eta \rightarrow 0$.

Assuming $n \geq 1$ we have the following equations:

$$\sum_{j=1}^n Q_j = 1, \quad (27)$$

$$Q_{j-1} = Q_1 f_0 \phi_{j-1} + \sum_{i=0}^{j-1} Q_{j-i} f_i - \delta_{j1} Q_1 f_0 \quad \text{for } j = 0, 1, \dots, n-1 \quad (28)$$

and

$$Q_{n-1} = Q_1 f_0 \phi_{n-1} + \sum_{i=0}^{n-1} Q_{n-i} G_i - \delta_{n1} Q_1 G_0. \quad (29)$$

Here δ_{jk} is the Kronecker symbol.

$$\phi_v = \begin{cases} 0 & \text{for } v \leq 0 \\ \frac{q_v}{1 - q_0} & \text{for } v \geq 1 \end{cases} \quad (30)$$

is the probability of v events occurring in a data-store cycle provided at least one event has to occur, as is the case for the cycle preceding the first storage transfer after the buffer has been empty.

$$G_i = h_0 \sum_{v=i+1}^{\infty} g_v + f_i = h_0 \left(1 - \sum_{v=0}^i g_v \right) + f_i \quad (31)$$

takes care of the fact that when the buffer is full, the occurrence of more events does not change the state of the system. We note that

$$G_0 = h_0 (1 - g_0) + f_0 = h_0 \quad (32)$$

We shall now find an expression for $R = R_n(t, y)$ as defined by Eq. (5). Let $\bar{N}$ be the average number of events lost in connection with each data storage operation. In a time T very much longer than λ^{-1} we then have

$$T\lambda_{\text{stored}} = T\lambda - T\lambda_{\text{lost}} = T\lambda - T\lambda_{\text{stored}} \bar{N} \quad (33)$$

which with $\lambda_{\text{stored}} = R \lambda$ gives $R = 1 - R \bar{N}$

or

$$R_n(t, y) = R = \frac{1}{1 + \bar{N}} \quad (34)$$

$\bar{N}$ must be the average number of events occurring in connection with each data storage operation minus the one event stored:

$$\begin{aligned} \bar{N} &= x + \left(\sum_{v=1}^{\infty} v \phi_v \right) Q_1 \{f_0 (1 - \delta_{n1}) + h_0 \delta_{n1}\} - 1 \\ &= x - 1 + \frac{Q_1 t}{1 - q_0} \{f_0 (1 - \delta_{n1}) + h_0 \delta_{n1}\} \end{aligned} \quad (35)$$

In this expression x is the average number of events occurring during the effective dead-time ξ itself; $\sum_{v=1}^{\infty} v\phi_v$ is the average number occurring during the data-store cycle immediately before the transfer starts, provided the previous transfer ended with the buffer empty, and the probability for this is $Q_1 \{f_0(1-\delta_{n1}) + h_0\delta_{n1}\}$.

We can find Q_1 by solving Eqs. (27), (28), and (29). We have previously stated that A_k as given by Eq. (13) for $k = 0, \dots, n$ is the solution of Eq. (12a) :

$$A_{j-1} = \sum_{i=0}^{j-1} A_{j-i} f_i \quad .$$

We now define

$$A_{j-1}^* = \sum_{k=1}^{j-1} A_k \phi_{j-k} \quad , \quad (36)$$

for which we find the recurrence relation

$$A_{j-1}^* = \sum_{i=0}^{j-1} A_{j-i}^* f_i - \phi_j \quad . \quad (37)$$

From the definition Eq. (36) it follows that

$$A_i^* = 0 \quad \text{for} \quad i \leq 0 \quad . \quad (38)$$

It is now easy to show by substitution that

$$Q_j = Q_1 f_0 (A_j - A_{j-1}^* - \delta_{j0}) \quad j \leq n-1 \quad (39)$$

is the solution of Eqs. (28), with the correct initial conditions, since $A_1 = f_0^{-1}$. It also gives $Q_j = 0$ for $j \leq 0$.

Equations (29) and (39) give

$$Q_n = Q_1 \left\{ (1-g_0)\delta_{n1} - f_0 \sum_{j=1}^{n-1} (A_j - A_{j-1}^*) \left(1 - \sum_{v=0}^{n-j} g_v \right) + f_0 g_0 (A_n - A_{n-1}^*) \right\} \quad . \quad (40)$$

From Eqs. (27), (39), and (40) it then follows that

$$\begin{aligned} Q_1^{-1} &= (1 - g_0) \delta_{n1} + f_0 \sum_{j=1}^n (A_j - A_{j-1}^*) \sum_{v=0}^{n-j} g_v \\ &= \left\{ \sum_{j=1}^n (A_j - A_{j-1}^*) \sum_{v=0}^{n-j} g_v \right\} \left\{ f_0 (1 - \delta_{n1}) + h_0 \delta_{n1} \right\}, \quad (41) \end{aligned}$$

since $A_1 = f_0^{-1} = (h_0 g_0)^{-1}$ and $A_0^* = 0$.

Substituting this into Eq. (35) we get

$$\bar{N} = x - 1 + [b_n(t, y)]^{-1}, \quad (42)$$

where

$$b_n(t, y) = \frac{1}{t} (1 - q_0) \sum_{j=1}^n (A_j - A_{j-1}^*) \sum_{v=0}^{n-j} g_v. \quad (43)$$

Equations (30) and (36) give

$$A_{j-1}^* = \sum_{k=1}^{j-1} \frac{A_k q_{j-k}}{1 - q_0}$$

and thus

$$b_n(t, y) = \frac{1}{t} \sum_{j=1}^n \left(A_j - \sum_{k=1}^j A_k q_{j-k} \right) \sum_{v=0}^{n-j} g_v. \quad (44)$$

We shall define

$$a_j(s) = \sum_{k=1}^j A_k \frac{s^{j-k}}{(j-k)!} e^{-s} = \sum_{k=0}^{j-1} A_{j-k} \frac{s^k}{k!} e^{-s}. \quad (45)$$

We substitute A_{j-k} from Eq. (13), carry out one summation, and get

$$a_j(s) = \sum_{\ell=0}^{j-1} \frac{[(\ell-j)x + s]^\ell}{\ell!} e^{(j-\ell)x-s} . \quad (46)$$

By repeating the procedure with $a_{j-k}(s_1)$ instead of A_{j-k} we get

$$\sum_{k=0}^{j-1} a_{j-k}(s_1) \frac{s_2^k}{k!} e^{-s_2} = a_j(s_1+s_2) . \quad (47)$$

We note in passing that

$$a_j(s+x) = a_{j-1}(s) + \frac{s^{j-1}}{(j-1)!} e^{-s} . \quad (48)$$

Using Eqs. (45) and (47) we can write

$$\begin{aligned} b_n(t,y) &= \frac{1}{t} \sum_{j=1}^n \left\{ A_j - a_j(t) \right\} \sum_{v=0}^{n-j} g_v \\ &= \frac{1}{t} \sum_{j=1}^n \sum_{k=0}^{j-1} \left\{ A_{j-k} g_k - a_{j-k}(t) g_k \right\} \\ &= \frac{1}{t} \sum_{j=1}^n \left\{ a_j(y) - a_j(t+y) \right\} . \end{aligned} \quad (49)$$

Since

$$\sum_{j=1}^n a_j(s) = \sum_{\ell=0}^{n-1} \sum_{v=0}^{\ell} \frac{[(\ell-n)x + s]^v}{v!} e^{(n-\ell)x-s} , \quad (50)$$

we can also write

$$b_n(t,y) = \sum_{\ell=0}^{n-1} p_\ell(t, (\ell-n)x + y) , \quad (51)$$

where

$$p_{\ell}(t, w) = \frac{1}{t} \sum_{v=0}^{\ell} \frac{1}{v!} \left\{ w^v - (w+t)^v e^{-t} \right\} e^{-w} . \quad (52)$$

This leads us to write down a recurrence relation directly in terms of $b_n = b_n(t, y)$. Using Eqs. (47), (48), and (49) we get

$$\sum_{i=0}^{k-1} b_{k-i} f_i = b_{k-1} + p_{k-1}(t, y) , \quad (53a)$$

or more suitable for computation:

$$f_0 b_k = b_{k-1} - \sum_{i=1}^{k-1} b_{k-i} f_i + p_{k-1}(t, y) \quad \text{for } 1 \leq k \leq n . \quad (53b)$$

Starting with

$$b_0(t, y) = 0 \quad (54)$$

we get

$$b_1(t, y) = p_0(t, y) f_0^{-1} = \frac{1}{t} \left(e^{x-y} - e^{x-y-t} \right) , \text{ etc } . \quad (55)$$

In the limit of $t \rightarrow 0$, i.e. the asynchronous case:

$$p_{\ell}(0, w) = \frac{w^{\ell}}{\ell!} e^{-w} , \quad (56)$$

which with Eq. (51) gives

$$b_n(0, y) = a_n(y) , \quad (57)$$

and if there is no buffer overlap, i.e. $y = 0$:

$$b_n(0, 0) = a_n(0) = A_n \quad \text{for } n \geq 1 . \quad (58)$$

This result combined with Eqs. (42) and (34) agrees for $n \geq 1$ with Eq. (11) derived in another way in Section 2.1. Since $b_0(0,0) = 0$ and $A_0 = 1$, we get a wrong result for $n=t=y=0$, but the derivation in this section was not valid for $n=0$. We can now fill in this lacuna by putting

$$R_n(t,y) = \frac{1}{x + [B_n(t,y)]^{-1}} = \frac{B_n(t,y)}{xB_n(t,y) + 1}, \quad (59)$$

where

$$B_n(t,y) = \begin{cases} 1 & \text{for } n=t=y=0 \\ b_n(t,y) & \text{otherwise} \end{cases}. \quad (60)$$

From this and Eqs. (57) and (48) we get for the asynchronous case with complete buffer overlap, i.e. $y = x$:

$$R_n(0,x) = \frac{1}{x + [a_n(x)]^{-1}} = \frac{1}{x + A_{n-1}^{-1}}. \quad (61)$$

Comparing this result with Eq. (11) we note that one event buffer has been "lost", as one could expect.

For $t = x$, i.e. the synchronous case we get from Eqs. (48) and (49)

$$b_n(x,y) = \frac{1}{x} \left\{ a_n(y) - \sum_{v=0}^{n-1} \frac{y^v}{v!} e^{-y} \right\}, \quad (62)$$

and in particular for no buffer overlap, i.e. $y = 0$:

$$b_n(x,0) = \frac{1}{x} (A_n - 1). \quad (63)$$

This result put into Eqs. (59) and (60) gives the same $R_n(x,0)$ as Eq. (22) derived in Section 2.2.

For the case of complete buffer overlap, i.e. $y = x$, we get

$$b_n(x,x) = \frac{1}{x} \left\{ A_{n-1} - \sum_{v=0}^{n-1} f_v \right\} \quad (64)$$

which shows, in contrast to Eq. (61), that in this synchronous case it is not just one buffer "lost". The reason for this is that the buffer, unlike the data store, is always assumed to be asynchronous.

4. EXAMPLES

The meaning and the use of the results derived in the previous sections may become clearer through the following examples.

4.1 Buffered paper tape punch

A detector produces random events with a mean rate of $\lambda = 30$ events/sec. The description of each event takes two characters. As data store we consider a punch with maximum speed 75 ch/sec. It will then take a time $\delta = 2/75 = 26.7$ msec to record one event (see Figs. 1 and 2). The interface of the punch contains a character buffer, which is also part of the data store. It is asynchronous ($\tau = 0$), and busy during the whole punching of a character. Before the data store is one event buffer ($n = 1$). It can accept a new event immediately when the second character has been transferred to the data store. We can ignore all transfer times into and between buffers. Equation (2) gives $\xi = \delta$, and from Eq. (3) we have $x = 0.8$, $t = 0$, and $y = 0.5x = 0.4$. The stored fraction is given by Eq. (59). Using Eq. (55) we get $R_1(0, y) = \{x + e^{-(x-y)}\}^{-1}$ or $R_1(0, 0.4) = 0.680$. To increase this we could either increase the capacity of the buffer to $n > 1$ events, or use a faster punch. We show below the "equivalent punching speed" a punch with one event buffer must have in order to compete with a punch of speed 75 ch/sec having a buffer with capacity n :

n	$R_n(0, 0.4)$	Equivalent punching speed
1	0.680	75 ch/sec
2	0.857	230 ch/sec
3	0.923	375 ch/sec

Increasing the buffer capacity from 2 to 3×2 characters is thus an alternative to speeding up the punch by a factor 5. This result would, of course, not apply if the source of the events would be non-random, for example a computer.

4.2 Data acquisition computer

Let the data store in Fig. 1 be a computer, which receives an interrupt when a new event is ready in the buffer. The computer needs α cycles to get ready for the transfer, β cycles to transfer under DMA control, and γ cycles to clear up before it can accept a new interrupt. According to Fig. 2 and Eqs. (2) and (3) we have $\xi = \delta$, $x = \lambda\tau(\alpha + \beta + \gamma)$, $t = \lambda\tau$, $y = \lambda\tau(\alpha + \beta)$.

The fraction of stored events is given by $R_n(t,y)$ in Eq. (59). For $\tau = 1 \mu\text{sec}$, $\lambda = 2000 \text{ sec}^{-1}$, $\alpha = 60$, $\beta = 100$, and $\gamma = 40$ we get $x = 0.4$, $t = 0.002$, and $y = 0.32$. For $t \ll 1$ we derive by means of Eq. (55) $R_1(t,y) \approx \{x + (1 + t/2)e^{-(x-y)}\}^{-1}$, so the extra loss due to the synchronization with the computer cycle is negligible in this case. In fact $R_1(0.002, 0.32) = 0.7553$, whereas $R_1(0, 0.32) = 0.7558$. $R_2(0, 0.32) = 0.95$ and $R_3(0, 0.32) = 0.99$ show what one would gain by increasing the buffer capacity to 2 and 3.

REFERENCE

- 1) D.R. Cox and W.L. Smith, *Queues* (Methuen, London, 1961).

Figure captions

- Fig. 1 : Data recording system consisting of a data store and a buffer with capacity n events.
- Fig. 2 : Timing diagram for data store and buffer with capacity $n = 1$ event. Both the data store and the buffer are here free when the event occurs.
- Fig. 3 : Graph of the fraction of events stored $R_n(t,y)$ as a function of x for various values of n , t , and y . For definitions see Figs. 1 and 2 and Eq. (3).

Fig. 1

Fig. 2

Fig. 3