ESD-TR-72-327

MTR-2434

COMPARISON OF REQUEST HANDLING CAPABILITY OF SOME AIRBORNE DRUM MEMORIES

Norman B. Sutherland

DECEMBER 1972

Prepared for

DEPUTY FOR COMMAND AND MANAGEMENT SYSTEMS

ELECTRONIC SYSTEMS DIVISION
AIR FORCE SYSTEMS COMMAND
UNITED STATES AIR FORCE
L. G. Hanscom Field, Bedford, Massachusetts

Project 6700

Prepared by
THE MITRE CORPORATION
Bedford, Massachusetts

Contract 19628-71-C-0002

ESD-TR-72-327

MTR-2434

COMPARISON OF REQUEST HANDLING CAPABILITY OF SOME AIRBORNE DRUM MEMORIES

Norman B. Sutherland

DECEMBER 1972

Prepared for

DEPUTY FOR COMMAND AND MANAGEMENT SYSTEMS

ELECTRONIC SYSTEMS DIVISION
AIR FORCE SYSTEMS COMMAND
UNITED STATES AIR FORCE
L. G. Hanscom Field, Bedford, Massachusetts

Project 6700

Prepared by

THE MITRE CORPORATION
Bedford, Massachusetts

Contract 19628-71-C-0002

FOREWORD

The work described in this report was carried out under the sponsorship of the Deputy for Command and Management Systems, Project 6700, by The MITRE Corporation, Bedford, Massachusetts, under Contract F19628-71-C-0002.

REVIEW AND APPROVAL

Publication of this technical report does not constitute Air Force approval of the report's findings or conclusions. It is published only for the exchange and stimulation of ideas.

Paul A. Butler

PAUL A. BUTLER, Col, USAF
Deputy Program Manager, (AABNCP)
Deputy for Command and Management Systems

ABSTRACT

A method is described for developing a consistent framework for comparing the request handling capabilities of various drum memories. The method permits one to estimate the request capacity of a drum, given its physical characteristics together with a number of assumptions regarding such factors as data organization, blocking, average quantity of data transferred per request, and effective latency time. The method developed is used to compare the capability of several existing or proposed airborne drums. The effect of a number of possible modifications to a particular drum (e.g., increase density, increased rotational speed, reduction of number of overhead bits) is also examined.

TABLE OF CONTENTS

LIST OF ILLUSTRATIONS

LIST OF TABLES

SECTION I

INTRODUCTION

This paper describes a method that was used to consistently compare the potential request-handling capability of a number of magnetic drum memories. A request here is considered to be a user or program generated call for data base interaction which requires reading or writing one or more blocks of information on the drum. Typical requests include queries and file update actions. The basic operations postulated for processing a typical request are discussed below in Section III. As noted in that discussion, the time to process a request is considered to be approximated by the data access and transfer times on the assumption that internal request processing time will be negligible when compared with drum I/O time. Thus, the method described involves estimating the data access delays and data transfer times when processing a typical request. The request handling capability is an estimate of the maximum number of such requests per minute which the drum can accommodate.

Average request processing time (excluding internal processing time) is a function of:

1. the average quantity of data transferred per request;
2. the data transfer rate;
3. the average rotational delay per request.

The average quantity of data transferred per request must be determined for the particular application, and consists of a weighted average based on the average quantity of data transferred for each type of request, and the relative frequency of the various request types.

The data transfer rate is determined by the physical characteristics of the drum, and by the controller, I/O channel arrangements and channel capacity. Physical parameters to be considered when determining the effective data transfer rate include:

Drum diameter

Rotational speed

Recording density

Number of overhead (non-data) bits required

Number of paralleled tracks

The average rotational delay per request is a function of drum rotational speed, of data file structure, organization and allocation to the drum, and accessing techniques. The average rotational delay per request consists of a weighted average based on the average drum latency time for each type of request, and the relative frequency of the various types. Average rotational delay or latency is typically estimated as 1/2 of drum revolution time for each random access to a data item or block of data, but this may be reduced by various optimization schemes.

SECTION II

GENERAL EXPRESSION FOR REQUEST HANDLING CAPACITY

Let $\overline{Wq}$ be the weighted average number of words transferred per request.

The data transfer rate is a function of the number of words that can be read or written per revolution (W_T = no. of words per track, or per band if two or more tracks are paralleled), and of the rotational speed (V_{rpm}).

The expression for W_T is

$$W_T = \frac{(\pi DBN_p)\rho}{\ell} \text{ wds/track (or band)} \qquad (1)$$

where D = drum diameter in inches

B = recording density in bits/inch

N_p = no. of paralleled tracks per band (no. of tracks which are simultaneously read)

ℓ = no. of data bits per word

ρ = reduction factor to account for overhead bits. The value of ρ is determined by the ratio of the number of data bits per track to the total number of bits (data and overhead) per track.

The expression for the data transfer rate is

$$R = \frac{W_T V_{rpm}}{60} \text{ wds/sec}$$

where W_T = no. of words per track (or band)

V_{rpm} = rotational speed of drum in revolutions per minute

The weighted average rotational delay or latency time per request is given by

$$\bar{T}_L \doteq \bar{N}_d t_d \text{ seconds} \tag{3}$$

where $\bar{N}_d$ = average number of blocks accessed per request for which there is a rotational delay, weighted by the relative frequency of the various request types

t_d = average rotational delay in seconds

The average rotational delay t_d is often expressed as a fraction of drum revolution time (e.g., average access time equals 1/2 drum rotation time). Letting μ equal the fraction and substituting μT_{rev} for t_d in (3) above, we have

$$\bar{T}_L \doteq \bar{N}_d \mu T_{rev} \tag{4}$$

The average processing time per request is given by

$$T_q = \bar{W}_q/R + \bar{T}_L \text{ seconds} \tag{5}$$

Substituting from (2) and (4), we have

$$T_q = \frac{\bar{W}_q \; 60}{W_T \; V_{rpm}} + \bar{N}_d \mu T_{rev}$$

but $$T_{rev} = \frac{60}{V_{rpm}}$$

$$T_q = \frac{\bar{W}_q}{W_T}\frac{60}{V_{rpm}} + \frac{\bar{N}_d \mu 60}{V_{rpm}}$$

$$T_q = \left(\frac{60}{V_{rpm}}\right)\left(\frac{\bar{W}_q}{W_T} + \mu \bar{N}_d\right) \text{ seconds} \qquad (6)$$

The request handling capacity is expressed as

$$C = \frac{60}{T_q} \text{ requests/minute}$$

Substituting for T_q from (6)

$$C = \frac{V_{rpm}}{\bar{W}_q/W_T + \mu\bar{N}_d} \text{ requests/minute} \qquad (7)$$

SECTION III

ASSUMPTIONS FOR AABNCP APPLICATION

In order to make a consistent comparison of the request handling capacity of several drums, it is first necessary to make some application-specific assumptions. In particular, assumptions must be made that will enable one to determine $\overline{W}_q$, μ, and $\overline{N}_d$. These assumptions, together with the physical parameters of the various drums, may then be used to estimate the request handling capacity of the drums for the given application.

The application for which the comparisons in this paper were made is within the Advanced Airborne Command Post (AABNCP) data processing system. The on-board AABNCP data processing system will require a very large (approximately 300 million bit) auxiliary store to hold programs and data. Approximately 18 users may be simultaneously interacting with this data base from on-line terminals. At the same time, the data processing system will be handling a heavy volume of message traffic from external sources, much of which will result in the generation of requests for data base update.

In order to provide the required storage capacity, it has been suggested in the past that two Mass Memory Units (MMU) of the type used in PACCS ADA should be considered.

The PACCS ADA MMU is a magnetic drum mass memory system that was built by the Magne Head Division of General Instrument Corporation,

and is currently being flown by SAC in an EC-135C aircraft as part of an experimental testbed for the AABNCP. Each MMU has a data capacity of 100-million bits (see Table I for other MMU characteristics).

In an internal project document on the subject of AABNCP Software Design, J. Glore makes some rough estimates of the request handling capacity of the PACCS ADA MMU for the AABNCP application. He concludes that if two PACCS ADA-like MMUs are employed as AABNCP auxiliary storage, the rates at which information can be read from and written into this auxiliary storage can be expected to severely limit system performance.

For this reason, this paper compares the request handling capability, as estimated for the AABNCP application, of the PACCS ADA MMU with that of a number of other existing or proposed airborne drum memories. The effect on request handling capacity of a number of possible modifications to the PACCS ADA MMU is also examined.

For the AABNCP application, the following assumptions were made, using Glore's estimate as a general guide. They are presented here so that one may, if he so desires, modify the assumptions and get different results which are otherwise consistent.

REQUEST TYPES

Two primary request types are considered; Retrieval and Update. A retrieval request in general, requires searching a file (or files) and extracting items which qualify according to a conditional expression contained in the retrieval request. An update request in general,

requires locating in a file (or files) those items requiring updating, loading them, updating them, and rewriting them onto auxiliary storage.

REQUEST PROCESSING OPERATIONS

It is assumed that processing of a request involves the following basic operations:

1. Initial examination of the request
2. Loading appropriate request handling programs from drum
3. Translation of the request
4. Loading key file(s) from drum
5. Searching key file(s) for qualifying records
6. Loading qualifying records from drum
7. If updating:
    a. Modification of qualifying records
    b. Writing updated records back on drum
8. Formatting of output messages
9. Presentation of output in response to the request.

The assumption is made that the time required for internal processing will be very small in comparison with the data access and transfer times, and may be considered as negligible for our purposes. Thus, the time to process a request may be approximated by the sum of the times to perform operations 2, 4, and 6, above, with the additional time to perform operation 7b in the case of updating. These operations are examined in more detail below.

Program Loading

This operation involves loading from drum the programs necessary for processing a request. The following assumptions were made regarding program loading:

1. 5000 words of request processing programs must be loaded.
2. Word length is 36 data bits.
3. Programs are stored on the drum in blocks of 500 words each.
4. It is not assumed that blocks of programs are contiguously stored. Therefore, a rotational delay for each block is assumed.
5. The average latency, or rotational delay per block, is assumed to be equal to one half of the drum rotation time ($\mu = .5$).

Key File(s) Loading

This operation involves loading from drum the key file or set of key files, i.e., the file(s) against which the search is made for qualifying records. The following assumptions are made concerning the key file(s):

1. File structure is parallel by word.
2. Single-word records, with 36 bits/word.
3. The one or more key subfiles loaded comprise a total of 15,000 words.

4. Each parallel subfile is stored in one or more blocks of approximately 500 words per block.
5. The average rotational delay in accessing each block is assumed to be equal to one half of the drum rotation time (μ = .5).

Sequential search of at least one key subfile is assumed necessary to satisfy an update or retrieval request, because only one of a set of parallel subfiles can be arranged to permit faster searching. A typical search request is also assumed to require all (rather than any) of the set of qualifying values, implying a need to search an entire key subfile.

The assumption of a one-half rotation time latency per block is a compromise. If blocks of a given file are stored consecutively, once the first block has been located, subsequent blocks may be loaded with no access delay provided that the processing of each block can be completed within the time that it takes to load the next block. However, for faster drums, especially in a multiprogrammed environment, this may not always be possible. In that case, if blocks are consecutively stored, an access delay of up to a full revolution would occur between blocks. Various optimization schemes may be postulated for reducing the effective latency. Rather than assume a particular optimization technique, the compromise value of one-half rotation time delay per block is assumed.

Additional Retrieval

This operation involves loading those records (in parallel subfiles) which correspond to the key subfile records that qualified during the key subfile search and which contain desired data not included in the key subfile(s). The following assumptions are made:

1. 1% of the key subfile records will qualify during the search.
2. For each qualifying record in a key subfile, a corresponding record from an unsearched parallel subfile must be retrieved.
3. The records to be retrieved will be distributed randomly and uniformly throughout the blocks of the unsearched subfile. Since a block length of 500 words is assumed, an average block in the unsearched subfile would contain at least 5 records (1% of 500) to be retrieved. With a uniform distribution one would expect it would be necessary to load nearly every block of the unsearched subfile in order to retrieve the qualifying records, since the probability that a block contains no qualifying record is quite low. Therefore, retrieval will require loading approximately the same number of blocks as for searching, or about 15,000 words.
4. Latency time of one-half drum rotation time is assumed necessary for locating only the first block of the unsearched subfile. It is assumed that subsequent blocks are stored consecutively and may be loaded and processed without additional

rotational delay, since no searching is required and the number of records to be retrieved per block is small.

Update Rewrite

This operation involves writing back on drum records which have been retrieved and modified. Update rewrite is essentially the reverse of the Additional Retrieval Operation above, and therefore it is assumed that:

1. Retrieved and updated blocks comprising approximately 15,000 words must be rewritten.
2. Latency time of one-half drum rotation time is necessary for locating only the first block of the subfile being updated.

RELATIVE FREQUENCY OF REQUEST TYPES

Based on the results of modeling activity, the assumption is made that update requests will occur four times as often as straight retrieval requests.

SECTION IV

GENERAL EXPRESSION MODIFIED BY AABNCP ASSUMPTIONS

Using the assumptions for the AABNCP application from the previous section, the general expression for request handling capacity (see Section II) may be modified to fit the AABNCP application.

For a retrieval type request, the average number of words transferred from drum includes 5,000 words of program, 15,000 words of key subfile and 15,000 words of unsearched subfile. Therefore:

$$W_{q_1} = 5{,}000 + 15{,}000 + 15{,}000 = 35{,}000 \text{ words}$$

For an update type request, the average number of words transferred to and from the drum is the same as for a retrieval request, except that an additional 15,000 words are written back onto drum. Therefore:

$$W_{q_2} = 5{,}000 + 15{,}000 + 15{,}000 + 15{,}000 = 50{,}000 \text{ words}$$

Since updates are assumed to occur four times as often as retrievals, the weighted average number of words transferred per request is:

$$\begin{aligned} \overline{W}_q &= (W_{q_1} + 4\, W_{q_2})/5 \\ &= (35{,}000 + 200{,}000)/5 \\ &= 47{,}000 \text{ words.} \end{aligned}$$

For a retrieval type request, the number of blocks for which rotational latency is assumed includes all blocks of programs to be loaded, all blocks of the key subfile(s) to be searched, and the first block of the unsearched subfile from which additional retrieval is made. Since 500-word blocks are assumed:

$$N_{d_1} = \left(\frac{5,000}{500} + \frac{15,000}{500} + 1\right) = 41 \text{ blocks} \qquad (8)$$

For an update request, the number of blocks for which rotational latency is assumed will be the same as N_{d_1}, except for an additional delay in accessing the first block of the subfile being rewritten. Therefore:

$$N_{d_2} = \left(\frac{5,000}{500} + \frac{15,000}{500} + 1 + 1\right) = 42 \text{ blocks} \qquad (9)$$

The weighted average number of blocks per request for which rotational delay is assumed is:

$$\overline{N}_d = (N_{d_1} + 4N_{d_2})/5 \qquad (10)$$

$$= (41 + 168)/5$$

$$= 209/5 = 41.8 \text{ blocks}$$

substituting in (7) the values

$$\overline{W}_q = 47,000$$

$$\overline{N}_d = 41.8$$

$$\mu = .5$$

the modified general expression for request handling capacity becomes

$$C = \frac{V_{rpm}}{47{,}000/W_T + 20.9} \quad \text{requests/minute} \qquad (11)$$

which now reflects the assumptions made for the AABNCP application.

SECTION V

CALCULATION RESULTS AND DISCUSSION

Using the modified expression (11) of the previous section, the request handling capacity of several drums was calculated and presented in Table I.

The word length in each case was assumed to be 36 data bits. This is consistent with the current PACCS ADA drum configuration.

Recording or reading from more than one track simultaneously is not assumed; i.e., N_p is assumed to be 1.

A value of .93 as overhead reduction factor (ρ) was assumed for the proposed RCA and Magne-Head drums. This value is consistent with the reduction factor for the IBM 47 Mass Memory, and with estimates given by Magne-Head representatives (between 5% to 10% reduction for overhead bits).

Also shown in Table I is the theoretical request capacity with zero latency, to give an upper bound for each drum, subject to the assumptions of Section III, of course. If latency is reduced to zero, the request capacity is then directly proportional to the data transfer rate. The percentage improvement in request capacity when latency becomes zero is directly proportional to the number of data words per track.

From equation (11)

$$C = \frac{V_{rpm}}{47{,}000/W_T + 20.9}$$

Table I

Request Handling Capacity of Several Drums

	Magne-Head PACCS ADA	IBM 4π	Hughes	RCA (Proposed)	RCA (Proposed)	Magne-Head (Proposed)
Drum Diameter (inches)	18	6.5	6.625	8.4	8.4	12
Rotational Speed (rpm)	1160	4800	2120	2400	3600	3000
Recording Density (bits/inch)	1254	1960	2905	2200	2200	2985
Capacity (bits)	133 x 10^6	15 x10^6	15.48 x10^6	50 x10^6	50 x10^6	172.8 x10^6
Track Capacity (bits/track)	70,922	39,960	60,475	58,080	58,080	112,500
Word Length (data bits)	36	36	36	36	36	36
Overhead Reduction Factor (ρ)	.766	.926	.889	.93*	.93*	.93*
Data Words per Track	1509	1028	1494	1500	1500	2907
Rotation Time (seconds)	.0517	.0125	.0283	.025	.01667	.020
Maximum Transfer Rate (bits/second)	1.372 x10^6	3.192 x10^6	2.139 x10^6	2.323 x10^6	3.484 x10^6	5.625 x10^6
Data Transfer Rate (words/second)	29.2 x10^3	82.24 x10^3	52.8 x10^3	60 x10^3	90 x10^3	145.35 x10^3
Request Capacity (req/minute)	22.3	72	40.4	45.8	68.8	80.8
Zero Latency Request Capacity (req/minute)	37.2	105	67.3	76.5	115	185.5
Percentage Improvement by Reducing Latency to Zero	66.8%	45.8%	66.5%	67%	67%	129.5%

*Assumed

it may be seen that the request capacity can be improved by increasing either the rotational speed V_{rpm} or the number of data words per track W_T, or both.

In discussions with representatives of Magne-Head (manufacturer of the PACCS ADA drum) we were told that an increase in rotational speed of about 60% would be feasible. The effect of such an increase is shown in Table II.

One way to increase the number of words per track is to increase the recording density. The Magne-Head representatives indicated that improvements in technology since the PACCS ADA system was built would permit increasing the recording density of that system by 60%. The effect of a 60% density increase is calculated and shown in Table II.

Another way to increase the number of data words per track is to reduce the number of overhead bits. The PACCS ADA system requires a very high percentage of overhead bits (11 overhead bits per 36-bit word) in order to provide addressability to the level of individual words. The number of overhead bits can be reduced by increasing the size of the smallest addressable unit to a block of words (this would require hardware control logic changes which have not been discussed with the manufacturer). The improvement in request capacity that would result from increasing the overhead reduction factor (ρ) from .766 to .93 is calculated and shown in Table II.

Table II

Effect on Request Capacity of Various Modifications

	PACCS ADA (Present)	PACCS ADA 60% Increase in rpm	PACCS ADA 60% Increase in density	PACCS ADA Reduced Overhead	PACCS ADA 1500 words per block	PACCS ADA combination of b, c, e
	(a)	(b)	(c)	(d)	(e)	(f)
Drum Diameter (inches)	18	18	18	18	18	18
Rotational Speed (rpm)	1160	1860	1160	1160	1160	1860
Recording Density (bits/in)	1254	1254	2000	1254	1254	2000
Capacity (bits)	133_6 x10^6	133_6 x10^6	213_6 x10^6	133_6 x10^6	133_6 x10^6	213_6 x10^6
Track Capacity (bits/track)	70,922	70,922	113,040	70,922	70,922	113,040
Word Length (data bits)	36	36	36	36	36	36
Overhead Reduction Factor (ρ)	.766	.766	.766	.93	.766	.766
Data Words per Track	1509	1509	2405	1832	1509	2405
Rotation Time (Seconds)	.0517	.032	.0517	.0517	.0517	.032
Maximum Transfer Rate (bits/second)	1.372 x10^6	2.216 x10^6	2.185 x10^6	1.372 x10^6	1.372 x10^6	3.532 x10^6
Data Transfer Rate (words/second)	29.2 x10^3	46.78 x10^3	46.5 x10^3	35.42 x10^3	29.2 x10^3	75.15 x10^3
Request Capacity (req/minute)	22.3	35.7	28.6	24.9	30	68.5
Percentage Improvement Over Present PACCS ADA	---	60.0%	28.2%	11.7%	34.6%	207%
Zero Latency Request Capacity (req/minute)	37.2	59.6	59.4	45.2	37.2	95.2

As can be seen from Table I, the upper limit of request capacity is significantly degraded due to latency time. Referring to expression (4) for latency time

$$\overline{T}_L = \overline{N}_d \, \mu \, T_{rev} \quad ,$$

assuming that drum rotation rate remains constant, the average latency time may be decreased by decreasing $\overline{N}_d$, μ, or both.

One way to reduce $\overline{N}_d$ is to assume a larger block size (it is acknowledged that this may be in conflict with other considerations in determining optimum block size, and would have to be considered in a block size trade-off analysis). Assume, for example, that the block size for the AABNCP application was increased from 500 to 1,500 words per block. Expression (8) becomes

$$N_{d_1} = \left(\frac{5,000}{1,500} + \frac{15,000}{1,500} + 1\right) = 14.33 \text{ blocks}$$

Expression (9) becomes

$$N_{d_2} = \left(\frac{5,000}{1,500} + \frac{15,000}{1,500} + 1 + 1\right) = 15.33 \text{ blocks}$$

From (10),

$$\begin{aligned} \overline{N}_d &= (N_{d_1} + 4N_{d_2})/5 \\ &= [14.33 + (4)(15.33)]/5 \\ &= 15.13 \text{ blocks} \end{aligned}$$

Again, substituting in (7) the values

$\overline{W}_q$ = 47,000 words

$\overline{N}_d$ = 15.13 blocks

μ = .5

the modified expression for request handling capacity becomes

$$C = \frac{V_{rpm}}{47{,}000/W_T + 7.565} \quad \text{requests/minute} \qquad (12)$$

The request capacity for the PACCS ADA drum using the larger block size reflected in (12) is calculated and shown in Table II.

The combined effect of a 60% increase in density and rotational speed plus the use of a larger (1,500 word) block size is calculated and shown in Table II.

SECTION VI

USE OF DRUM ANGULAR POSITION INFORMATION TO REDUCE LATENCY

Another possible approach to reducing the average drum access time is through the use of drum angular position information, provided by a program-readable register that is updated continually by hardware as the drum revolves. The register would reflect at all times the next addressable word or block position on the drum.

The PACCS ADA drum controller has an internal 11 - bit register which always reflects the current rotational position of the drum. This register is reset to zero at the beginning-of-track origin point, and is incremented by one as each word position is passed. At present, the register is not program readable. Magne-Head representatives indicated that only minor hardware changes would be required to permit software readout of this register.

USE OF ANGULAR POSITION WITH FULL TRACK BLOCKS

Assuming that a program-readable angular position register was available, and that data was recorded on the drum in full track blocks, the angular position data might be used in the following way to reduce the effective latency time:

An in-core table is used to determine the track address of the desired block. The angular position register is read to determine

the current drum position within track. The present drum position (plus a delta to allow for delay in setting up and issuing I/O commands) is used to set up and issue a command to read from present position (plus delta) to the end of track. A second command is set up and chained to the first, to read from the beginning of the track up to the starting location of the first command. Since full track blocks are assumed, a block is completely read in one revolution. The use of the angular position data permits the immediate commencement of reading without having to wait until a beginning-of-track origin point is passed. Thus, the effective latency is nearly zero.

USE OF ANGULAR POSITION FOR LESS THAN FULL TRACK BLOCKS

The same general scheme can be used to read blocks that are less than a full track in length. The in-core block address table entry can specify the track address of the desired block, and the location and length of the block within the track. The operating system can use this information to compose I/O commands to access only the desired portion of the track.

When this method is used to access full track blocks, the rotational access delay per block, as previously noted, is nearly zero. When less than full track blocks are used, this is no longer true. The remainder of this section examines the relationship between block length and average rotational delay per block, when using angular position information in the manner described in the preceding paragraph.

For this discussion we shall express the block length (f) as a fraction of a full track. That is:

$$0 < f \leq 1$$

When a request is issued, the probability of the drum being positioned so that some portion of the requested block is already under the read head is:

$$p_a = f$$

The probability of the drum being positioned so that no part of the requested block is under the read head is:

$$p_b = (1-f)$$

$(p_a \times 100)\%$ of the time, the requested block will already be under the read head, and reading can begin immediately, from current position to end of block. However, in order to read the front end of the block, there will be a delay while passing over (1-f) track in order to reach the beginning of block. Therefore, $(p_a \times 100)\%$ of the time, the average delay per block will be

$$(1 - f) \text{ revolutions}$$

$(p_b \times 100)\%$ of the time, reading must be delayed until the beginning of block is encountered, since the read head is within the "non-block" area. The length of the non-block area is (1 - f) track,

and on the average, there will be a delay sufficient to pass one-half of the non-block area. Therefore, (p_b x 100)% of the time the average delay per block will be:

$$\frac{(1-f)}{2} \text{ revolutions}$$

The combined average delay (M) is the sum of each of the average delays above multiplied by its probability. Thus:

$$M = p_a(1-f) + p_b \frac{(1-f)}{2}$$
$$= (f)(1-f) + (1-f)\frac{(1-f)}{2}$$

which simplifies to:

$$M = \tfrac{1}{2}(1-f^2) \text{ revolutions}$$

Table III shows the average delay per block (M), expressed as a fraction of a revolution, for block sizes ranging from zero to full track; that is, for values of (f) from zero to one. As the table shows, the average delay varies from a minimum of zero for full track blocks to a maximum that approaches .5 revolutions for very short blocks. This relationship is shown graphically in Figure 1.

As seen above, the shorter the block length the longer the average access delay per block, when assuming the use of angular position information as previously discussed. To retrieve a given number of words, a shorter block length also means more blocks to be

Table III

Access Delays Vs. Block Length,

Assuming Use of Angular Position Data

Block Length as Fraction of Full Track	Delay per Block, in revolutions	Number of blocks (assuming 47,000 words, 1500 words per track)	Total Access Delay in revolutions (assuming use of angular position)	Total Access Delay in revolutions (assuming no angular position data	Percent Decrease in Total Access Delay when angular position data used
(f)	$M = \frac{1}{2}(1-f^2)$	$N_B = \left\lceil \frac{47,000}{(1500)(f)} \right\rceil$	$(M)(N_B)$	$(.5)(N_B)$	
1.00	0	32	0	16	---
.875	.117	36	4.2	18	76.6%
.75	.219	42	9.2	21	56.2%
.625	.305	51	15.6	25.5	38.8%
.50	.375	63	23.6	31.5	25.1%
.375	.430	84	36.2	42	13.8%
.25	.469	126	59.0	63	6.35%
.125	.492	251	123.5	125.5	0.16%
0	.500	∞	∞	∞	0

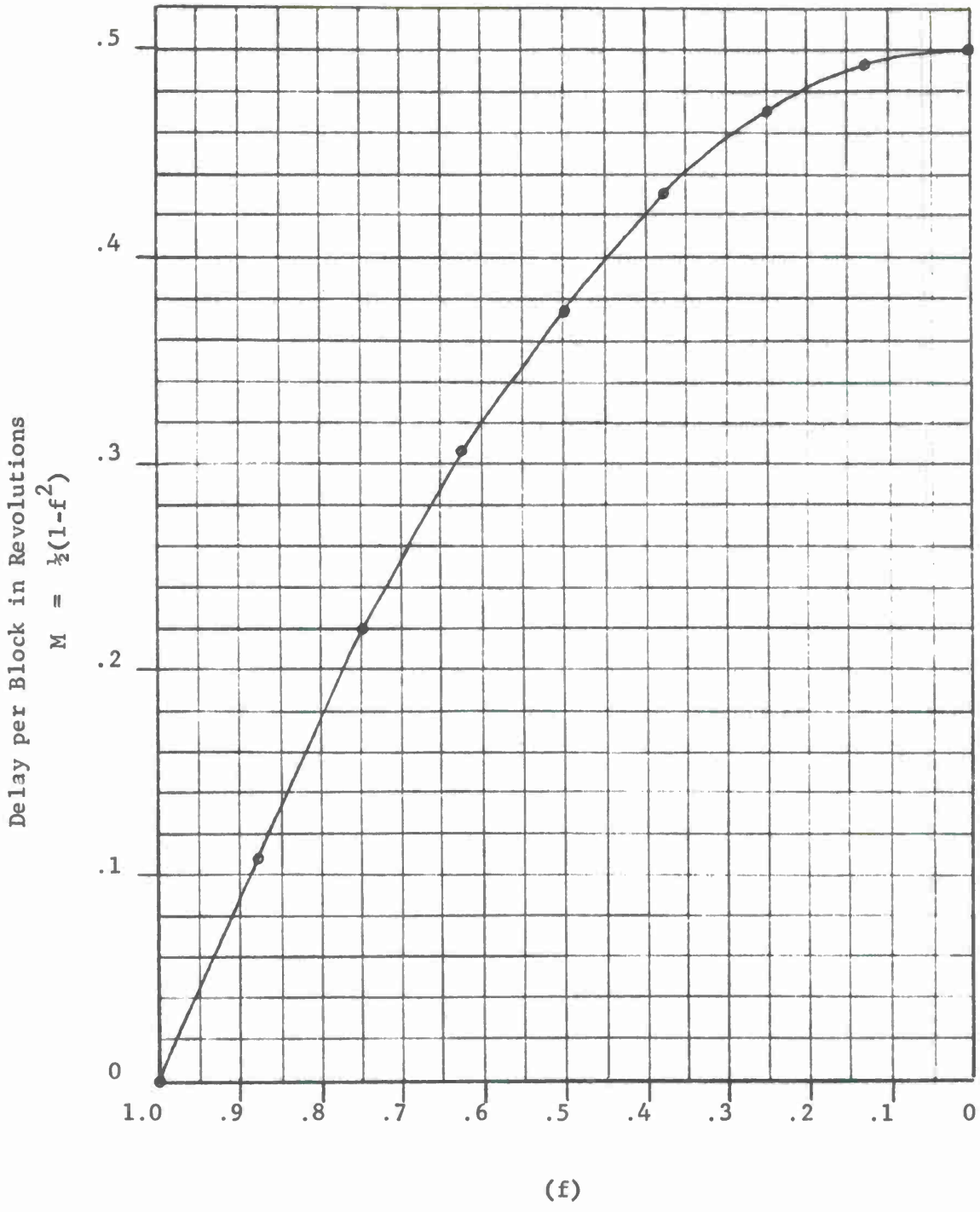

Figure 1. Access Delay per Block vs. Block Length

accessed. For the worst case, assume that each block is accessed independently, so that total access delay for N_B blocks is the product of N_B and the average delay per block.

Assuming retrieval of 47,000 words and a track length of 1500 words, Table III shows the number of blocks required to contain 47,000 words, for block lengths ranging from zero to full track. Also shown for each case is the total access delay for that number of blocks, expressed as numbers of revolutions. This is found by multiplying the number of blocks (N_B) by the average delay per block(M) for that particular block length. The total access delay (for 47,000 words) versus the block length is shown graphically in Figure 2.

It is interesting to now compare the total access delay as developed above with the average access delay that might be expected if angular position information was not available. In the latter case, the average access delay per block, assuming independent access to each block, is one-half revolution, regardless of block length. Table III shows the total access delay, found by multiplying the number of blocks (N_B) by .5 revolutions for each of the various block lengths. These values are presented graphically in Figure 2 as a comparison against the corresponding access delays when angular position information is available.

It can be seen from Figure 2 and Table III that the improvement resulting from the use of angular position information is marginal unless rather long blocks are used. For block length equal to

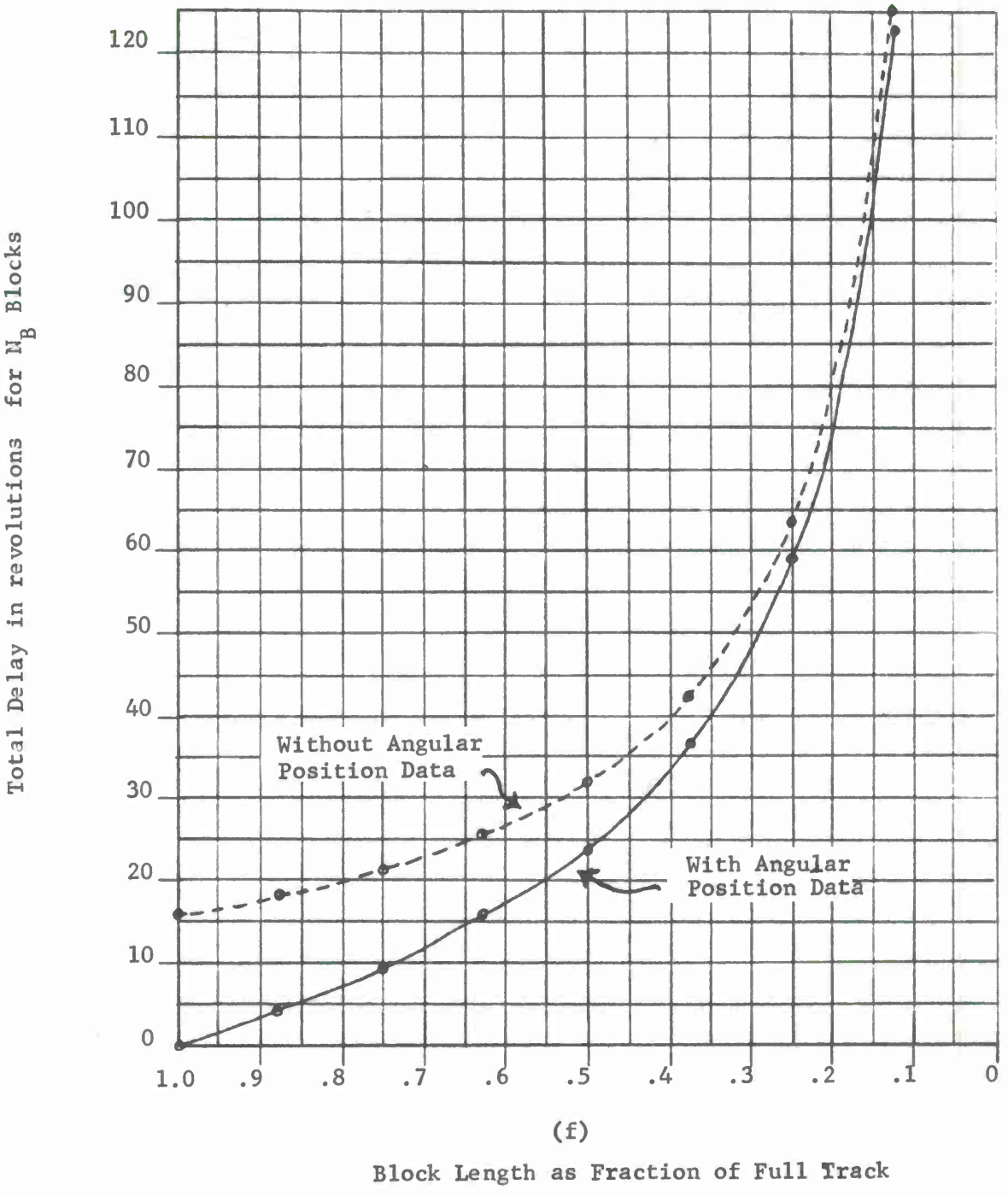

Figure 2. Access Delay With and Without Use of Angular Position Data

one-fourth of a track, the total access delay will be reduced by only 6.35% by using angular position data. For block length equal to one-half of a track, the reduction is 25.1%, and for block length of three-fourths of a track the reduction is 56.2%.

USE OF ANGULAR POSITION WITH SECTOR QUEUES

The preceding discussion of the use of angular position data assumes that drum access requests will be honored strictly on a first-come-first-served basis. This does not necessarily have to be the case.

Consider a drum that is subdivided into a number of equal-length angular sectors, and a program-readable angular position register that indicates which sector is currently under the read/write heads. If block length is made equal to sector length, a request for a block can be translated into a request for access to a specific track and sector.

A central routine, managing all drum requests, could place all block requests in sector queues, and, by using the current angular position information, service the requests from each queue as the corresponding sector passes under the read/write heads. With this scheme, the average access time per block is dependent upon the request load, decreasing as the request load increases, "until (in the limit) at least one outstanding block request exists at all times for each distinguishable drum sector. Under such limiting conditions,

average drum access time approaches rotation time divided by twice the number of such sectors."

SECTION VII

EFFECT OF ADDING A SECOND DRUM

As noted in Section III, it has been suggested that two PACCS ADA - like Mass Memory Units (on separate I/O channels) be used to provide a data storage capacity of 200-million bits. This section discusses the effect on request handling capacity of adding a second drum.

Assume that a single drum is capable of reading or writing one block in t seconds. Assume also that a query generates n block requests which must be satisfied before the query can be responded to. The minimum time to process one query is (nt) seconds, and the maximum query processing capacity of the single drum is $\left(\frac{60}{nt}\right)$ queries/ minute.

Assume a second drum is added, with the further condition that:

1. a given block request must be handled by a specific drum; it cannot be handled by either
2. the block requests will be evenly distributed between the two drums.

Consider two queries being processed simultaneously. In the ideal case, there would be no contention for a drum, processing of the queries would be completely overlapped, and 2n block requests would be handled in the same time that the single drum could handle n block requests.

In the worst case, there would be contention for a drum on every block request, so that no overlapped processing would occur. In this case, the time to process two queries (2n block requests) would be twice the time required by a single drum to process one query, giving no improvement at all.

Because block requests are assumed to be distributed evenly between the two drums, one would expect contention to occur on about one half of the block requests. The time required to process two queries is estimated as follows:

The two queries comprise a total of 2n block requests. Half of these will be processed singly because of contention, requiring (1/2)(2n)(t) seconds. The remaining n block requests will not encounter contention, and will be served two at a time, requiring (n/2)(t) seconds.

Therefore, the time to process <u>two</u> queries becomes

$$(1/2)(2n)(t) + (n/2)(t)$$

$$= (nt) + (1/2)(nt) = 3/2\ nt \text{ seconds.}$$

The average time to process <u>one</u> query becomes

$$(1/2)(3/2\ nt) = 3/4\ nt \text{ seconds.}$$

The combined capacity of two drums, measured in queries per minute, becomes

$$C = 60 \div 3/4\ nt$$

$$= 60 \cdot \frac{4}{3\ nt} = \frac{80}{nt} \text{ queries/minute}$$

Letting the request capacity of a single drum equal C_1 and the request capacity of two drums equal C_2, the ratio of C_2 to C_1 is then:

$$\frac{C_2}{C_1} = \frac{\frac{80}{nt}}{\frac{60}{nt}}$$

$$\frac{C_2}{C_1} = \frac{4}{3} \quad \text{or} \quad C_2 = 4/3\ C_1$$

Using the above expression, the estimated request capacity of the present PACCS ADA drum as shown in Table I would be increased from 22.3 to 29.7 requests per minute by the addition of a second drum.

Security Classification

DOCUMENT CONTROL DATA - R & D

(Security classification of title, body of abstract and indexing annotation must be entered when the overall report is classified)

1. ORIGINATING ACTIVITY (Corporate author) The MITRE Corporation P.O. Box 208 Bedford, Mass. 01730	2a. REPORT SECURITY CLASSIFICATION UNCLASSIFIED 2b. GROUP

3. REPORT TITLE

COMPARISON OF REQUEST HANDLING CAPABILITY OF SOME AIRBORNE DRUM MEMORIES

4. DESCRIPTIVE NOTES (Type of report and inclusive dates)

5. AUTHOR(S) (First name, middle initial, last name)

Norman B. Sutherland

6. REPORT DATE	7a. TOTAL NO. OF PAGES	7b. NO. OF REFS
DECEMBER 1972	39	0

8a. CONTRACT OR GRANT NO.	9a. ORIGINATOR'S REPORT NUMBER(S)
F19628-71-C-0002 b. PROJECT NO. 6700 c. d.	ESD-TR-72-327 9b. OTHER REPORT NO(S) (Any other numbers that may be assigned this report) MTR-2434

10. DISTRIBUTION STATEMENT

Approved for public release; distribution unlimited.

11. SUPPLEMENTARY NOTES	12. SPONSORING MILITARY ACTIVITY
	Deputy for Command and Management Systems Electronic Systems Division, AFSC L.G. Hanscom Field, Bedford, Mass.

13. ABSTRACT

A method is described for developing a consistent framework for comparing the request handling capabilities of various drum memories. The method permits one to estimate the request capacity of a drum, given its physical characteristics together with a number of assumptions regarding such factors as data organization, blocking average quantity of data transferred per request, and effective latency time. The method developed is used to compare the capability of several existing or proposed airborne drums. The effect of a number of possible modifications to a particular drum (e.g., increase density, increased rotational speed, reduction of number of overhead bits) is also examined.

DD FORM 1 NOV 65 1473

Security Classification

14. KEY WORDS	LINK A		LINK B		LINK C	
	ROLE	WT	ROLE	WT	ROLE	WT
COMPUTER STORAGE						
DRUM MEMORIES						
MAGNETIC DRUMS						
REQUEST-HANDLING, COMPUTER						