IBM CONFIDENTIAL

SDD - Endicott Laboratory
Department F64/6a56

February 14, 1973

MEMORANDUM TO: Distribution

SUBJECT: FS Sensor Base
Objectives and Strategy

The attached FS sensor Base Objectives and Strategy document is provided for your information, review and comment. For those who desire additional back-up information, the Appendix is available on request. By March 31st, technical specifications for sensor base support in FS and initial sensor base forecast assumptions will be complete. We would like to include your input.

To aid in this effort I am also asking those people identified by an asterisk before their name on the distribution list to make available, for their area of responsibility and within the context of the "Layered" strategy approach (by layer), their:

1. Market Requirements
   -function, capability, performance, etc.

2. Best guess sizing of the total sensor base market potential.

Please direct your information and comments to Mr. R. Guyette or myself by March 12th.

Larry Eischen
L. D. EISCHEN

dlc

attachments

DISTRIBUTION LIST:

	location	copy number
ASDD		
F. Oudheusden	Mohansic	1
DPG		
A. D. Pendleton	732/Harrison	2
T. F. Wagner	730/Harrison	3
GSD		
R. E. Carty	28Q/Boca Raton	4
W. R. Hamil	536/Rochester	5
J. Stuehler	889/Atlanta	6
*C. R. White	28Y/Boca Raton	7
WTC		
*J. R. Mueller	419/White Plains	8
SPD		
A. Esslinger	Boeblingen	9
C. S. Gurski	F51/Endicott	10
R. G. Hartmann	F65/Endicott	11
R. R. Highwart	F62/Endicott	12
A. A. Magdall	F40/Endicott	13
E. J. O'Neil	812/Poughkeepsie	14
G. E. Russell	B12/Poughkeepsie	15
F. B. Sheehy	06K/Fishkill	16
J. J. Webster	B65/Poughkeepsie	17
SDD		
G. P. Becker	F44/Endicott	18
B. W. Landeck	D56/Poughkeepsie	19
M. J. Marcus	B76/Poughkeeisie	20
J. W. Marks	F62/Endicott	21
W. McGovern	D56/Poughkeepsie	22
I. Presson	G31/Raleigh	23
T. Sasala	G10/Endicott	24
*H. R. Hansen	589/Kingston	25
*J. P. Murphy	B14/Poughkeepsie	26

SDD (continued)

	location	copy number
*F. L. Becker	R21/Endicott	27
R. P. Case	B50/Poughkeepsie	28
R. J. Cypser	B16/Poughkeepsie	29
C. M. Davis	D87/Poughkeepsie	30
P. Fagg	D70/Poughkeepsie	31
D. H. Furth	D57/Poughkeepsie	32
R. B. Hanrahan	D99/Poughkeepsie	33
*R. A. Hase	F62/Endicott	34
J. P. Hogan	D86/Poughkeepsie	35
O. W. Johnson	D28/Poughkeepsie	36
J. F. Keeley	D67/Poughkeepsie	37
A. S. Lett	B08/Poughkeepsie	38
W. F. O'Connell	730/Harrison	39
G. D. Pirtle	D19/Poughkeepsie	40
S. H. Pitkowsky	D93/Poughkeepsie	41
J. M. Pugliese	D86/Poughkeepsie	42
G. Radin	D74/Poughkeepsie	43
P. R. Schneider	B52/Poughkeepsie	44
C. O. Smarling	D86/Poughkeepsie	45
W. Y. Stevens	B16/Poughkeepsie	46
E. H. Sussenguth	E92/Raleigh	47
R. B. Talmadge	D99/Poughkeepsie	48
W. G. Thornton	H14/Poughkeepsie	49
H. F. Walsh	B64/Poughkeepsie	50
*R. C. Williams	R31/Endicott	51

DPD

*L. Bowman	822/White Plains	52
*B. Carris	Princeton	53
*F. Elser	Bethesda	54
*D. R. Foster	882/White Plains	55
*E. Green	Bethesda	56
*A. Guerin	Bethesda	57
*W. Haulihen	Princeton	58
*D. Jagger	Bethesda	59
*H. L. Kitchens	823/White Plains	60
*G. P. Kaplan	730/White Plains	61
*E. B. Levine	794/White Plains	62
*K. Y. Lilienfeld	839/White Plains	63
*J. Michel	Mohansic	64
*J. E. Morrison	773/White Plains	65
*J. McDermott	Bethesda	66
*P. J. O'Neil	823/White Plains	67
*G. Pavloff	2X16/White Plains	68
*C. Rowe	Princeton	69
*C. Sams	Princeton	70

COPY NUMBER: 48

FS SENSOR BASE OBJECTIVES AND STRATEGY

Advanced Products Analysis
Advanced Systems Market Development

January, 1973

Direct Questions To:

R. R. Guyette
Department F64/6a56
SDD Endicott Laboratory
Extension: 252+3434

THIS DOCUMENT IS NOT TO BE REPRODUCED. ALL INFORMATION HEREIN IS TO BE KEPT IBM CONFIDENTIAL

ABSTRACT

The 1970 Group Potentials Study showed 25% of the 1978 net potential as sensor base using systems. The establishment of a successful plan to penetrate this potential has suffered in the past due to the traditional application description approach to objectives. This document takes the approach that Sensor Base support is a system function analogous to TP or DB/DC, rather than an application or set of applications.

The objectives given here for support of Sensor Base as a system function are structured into a layered support strategy that reflects both our customers' needs and IBM's capabilities. Major consideration is given to such trends as low cost minicomputers, LSI, increasing cable installation costs and the establishment of computer hierarchies. The support strategy proposed in this document is structured into the following four levels of support:

1. Homogeneous Hierarchical Communication
2. Nonhomogeneous Hierarchical Communications
3. Automatic Data Collection
4. Non DP Device Support via Direct Sensor I/O

January 1973

IBM CONFIDENTIAL

1.0 Overview

1.1 Introduction

This document presents a strategy and objectives for a layered approach to FS Sensor Base support. The support levels called for are (in priority order):

1. Homogeneous Hierarchical Communications
2. Nonhomogeneous Hierarchical Communications
3. Automatic Data Collection
4. Other Non DP Devices

Section 1 of this document is concerned with the motivation and justification for such an approach.

Section 2 sets forth the characteristics of the application environment to be supported by each level.

Section 3 presents a first pass at technical objectives for supporting each level.

Section 4 contains some architectural considerations for Sensor Base and other Response Oriented Systems together with some initial comments on the current FS definition *vis a vis* Sensor Base.

1.2 Motivation

FS is the vehicle through which IBM must meet its growth objectives in the late 70's and early 80's. Neither price/performance improvements nor GNP growth will serve to meet our revenue objectives. Therefore, one of the key objectives of Advanced Systems is to seek out new functions and applications for FS which will provide the needed new sources of revenue. Funding solutions in new areas will not be easy. It will require a change in the way we traditionally think about our systems and policies and will lead us into areas where we may lack the kind of expertise we enjoy in our traditional business. However, if we are to meet our objectives and maintain our leadership in information processing, we must expand into nontraditional applications where others are already making inroads.

Although they have been around for well over a decade, sensor based applications are certainly nontraditional from the standpoint of IBM's mainline systems. IBM's past sensor based systems (1710, 1800, M44, S/7) and the minicomputer explosion led by Digital Equipment Corporation in the sixties have led the sensor oriented user away from mainline data processing. Users necessarily made different trade-offs. They elected for responsiveness and performance instead of throughput and integrity; capital equipment cost was more important than function and convenience.

At this point in time, mainline data processing and sensor based applications have both matured to the point where we can see some trends indicating a new generation where sensor base as a function will become integrated into our customer's total system together with the more traditional mainline functions. We appear to be at the point where teleprocessing was in the middle sixties.

In the past ten years we have seen an accelerating increase in the automating and computerizing of the value adding process. This has been especially true since the advent of integrated circuit minicomputers. These systems have been for the most part dedicated, stand alone computers concerned with optimizing one or at most a few related functions.

1.3 Trends

The trend now and for the future is to interconnect these stand alone computers into hierarchies of minicomputers. There are two reasons for this:

1.3.1 Technology

The first reason is the continuing rapid decrease in the cost of the mini systems. The customer can now make price/performance tradeoffs at a lower level than ever before. He can move intelligence in the form of the improved price/performance mini systems out nearer to where the sensors are. By connecting these systems into hierarchies, he can improve his total operation in the areas of RAS, cabling costs, and integrity, without giving up responsiveness and performance. Communication and coordination is handled by connecting several of these mini systems to a central "host" system. The host performs administrative, I/O, and supervisory functions on behalf of the subsystems. The host also is used for program development for the subsystems by way of cross assemblers and compilers. We expect this trend toward increasingly distributed intelligence to expand greatly with the advent of LSI.

1.3.2 Increasing Control Needs

The other reason for the increase in the number of sensor based hierarchies is the growing need for tying dissimilar sensor based systems together for the purposes of management. This need has been developed mainly in two areas. One is the increasing size and complexity of production operations; the other is the need for increasing productivity in the face of rising costs. Together they require tighter management control of the production process. This is achieved by tying sensor base control systems to higher level systems - communicating operational data upwards and supervisory or optimizing commands downward. We expect that this trend, too, will continue with the increasing integration and optimization of the production and planning processes.

These trends emphasize the fact that pieces of the ultimate corporate system are being designed and implemented today and that sensors and final control elements are legitimate I/O devices for an information processing system. If we are to provide solutions to our customers total needs with FS, sensor based functions must be incorporated into our systems.

1.4 Justification for a Layered Approach to Sensor Base Support in FS

Traditionally, sensor base has been considered an exotic, technically difficult application area that is far away from the main stream of IBM's expertise. Real time response and data rate requirements can be found that cover the entire spectrum from "any old time" to beyond the state of the art at any time. The complexity of sensor based applications, too, cover nearly as broad a range: from the smallest OEM mini buried inside an instrument, to duplexed/tandem M75's controlling a space mission. Worse, the complexity or size of the application (i.e., number of sensors/actuators, amount of data storage, and processing power required) bears almost no relationship to the response time and data rate required. For example, it is not unusual to find very powerful processors handling 1-4 hybrid simulators (e.g. M91 - Johns Hopkins) while an IBM 1800 monitors 840 machines in a GM plant (over 2000 digital input points and half that many outputs).

In the past, efforts have been made to try to group sensor based applications in order to have a more manageable number to deal with for planning, forecasting, development, education, etc. It is not unusual to find such application breakdowns as the following:

Plant Automation

- Make
- Move
- Test

Process Control

- Continuous
- Batch Sequence Control

Laboratory Automation

- Research
- Quality Control
- Clinical Lab

High Speed Data Acquisition

- Nuclear
- Telemetry
- Spectroscopy

When applications within a specific group are examined, we find that there are indeed similarities; but the range of performance and function required is still frustratingly large. We are still left with the situation where a "typical" configuration yields little insight into the true needs of the applications consigned to that grouping.

Figure 1 shows a matrix of requirements derived from the table that appears in FS Sensor Base Objectives AS-PL/PLNG/OBJ - POK-0344-0.0 dated 9/22/72 (Figure 2). The X's in the matrix represent the occurrence of a mention in the corresponding category. It can be seen that not only is there considerable overlap in most categories, but that the range is extremely broad in many.

In sum, the application description approach seems to obscure at least as much as it clarifies with the result that little has been accomplished in the sensor based area to date.

The layered approach described in this document provides the solution to these shortcomings. This will enable us to effectively penetrate what is a known, large potential (297 million application points in 1978 : 1970 Group Potential Study). The approach is to treat sensor base as a system function in much the same way that teleprocessing is treated as a system function. That is, provide a means for implementing a particular kind of a system rather than an application in itself. By taking this approach, we can begin to structure the kind of sensor base support needed, based on its integration into an application system.

From the point of view of FS, one of the most important developments in sensor based application systems is the emergence of computer hierarchies. It is important because it gives a structure to sensor based systems that IBM can support to advantage. These hierarchical systems have the characteristic that extreme data rate and response time requirements tend to be found at the lowest level of the hierarchy while the extreme in computational, storage and DP I/O requirements tend toward the upper end. It is precisely this structure that makes feasible a layered approach to sensor base support which takes advantage of IBM's strengths in the systems area. That is, it is possible to structure a support plan that starts with those areas of sensor based hierarchical support that are most in line with our systems expertise and proceeds in an orderly fashion in the direction of increasing involvement with the more unique areas of sensor base support.

This layered approach to sensor base support is a top down support plan which ensures that even in the event the entire plan is not implemented, there will be a consistent portion of our customer's problem that FS can address.

		Plant Automation	Process Control	Laboratory Automation	Data Acquisition and Control Systems	Health Ctrs. & Medical Schools
Operation Mode	LC	XXX	X	X	X	X
	TC	XX	X	X	XXX	
	SA	XX	X	X	XXX	X
Number of Units Attached to Host System	1-5	XX	X	XXXXXXX	XX	X
	6-10	XXX	X	XXXXXXXX	XX	X
	11-50	XXX	X	XXXXX	X	X
	51-100	XX			X	
	101-100	XX			X	
	> 200	X			XX	
Av Dist. Farthest Unit (KFT)	5-1				X	
	1-5					
	5-10	XXX	X	XXXX	XX	(20 Miles)
Av Data Rate per Unit (KB)	0-10	X			XX	
	1-10	X	X		X	
	10-100	X		X	X	X
	100-500				X	
	1000-10000				X	
Aggregate Data Rate to Host (KB)	10-50		X			
	50-100					
	100-500	X		X		X
	500-1000	X				
	1000-10000	X		X	XX	
Resp. Time	Micro Sec	X		X	XX	X
	Milli Sec	X	X			
	Seconds	X				

FIGURE I

FIGURE 2

Major App'l Area	Sub App'l Areas	Industry W/App'l.	Oper* Mode	# of Units Per Host	Distance Between Units Max	Distance Between Units Avg.	Host to Farthest Unit Max	Host to Farthest Unit Avg.	Data Rates Per Unit	Data Rates Aggregate	Response Time Lower Limit
Plant Auto	Make	Mfgr Proc	LC TC SA	Small 1-32	3K'	1K'	10K'	5K'	Max 10KB Avg 1KB	Small 100KB	Milli Secs
	Move	Mfgr Proc Distr	LC	Intermediate 4-128	3K'	1K'	10K'	5K'	Max 1KB Avg 100 bytes	Interm 500KB	Secs
	Test	Mfgr Proc Federal	LC TC SA	Large 8-256	3K'	1K'	10K'	5K'	Max 500KB Avg 10KB	Large 1MB	Micro Secs
Process Control	Petro-Chem Paper Metal Rubber Glass	Proc Mfgr	LC TC SA	1-16	4K'	2K'	10K'	7.5K	Max 3KB Avg 1KB	15KB to 40KB	Milli Secs
Lab	Research	Mfgr Fed Medical Educ Proc	LC TC SA	1-24 4-10 (Avg)	3K'	2K'	10K'	7.5K'	Max 200KB Avg 40KB	Max 1-2 MB Avg 200KB	Micro Secs
	Analytical			1-12 2-6 (Avg)	2K'	1K'	10K'	5K'			
	Clinical			1-24 8-12 (Avg)	3K'	2K'	10K'	7.5K'			
	QC			1-12 2-6 (Avg)	2K'	1K'	10K'	5K'			
Data Acq and Ctl	HS Hybrid Telem Nuc	Fed Mfgr Proc Educ	TC SA	1-10	0.5K'	0.2K'	1.5K'	0.5K'	8MB	8MB	Micro Secs
	NS Many	All	TC SA LC	1-256	Vary	Vary	10K'		Vary	8MB	Micro Secs
	LS Pollution Monitoring Weather	Mfgr Proc Fed	SA LC	256			2 miles		Very slow		
Health Ctrs. & Medical Schools	Clinical Lab Electrod Diagnostics Patient Monitoring Multiphasic Screening Signal Acq. & Anal. Automated Vision Hybrid Research	Med.	LC SA	1-24	50 mi	5 mi	100 mi	20 mi	10KB	200KB	Micro Secs.

* LC Loosely Coupled TC Tightly Coupled SA Stand Alone

2.0 Characteristics of the FS Layered Approach to Sensor Base

2.1 Introduction

In order to establish a layered or phased approach to sensor base support, the first step is to set up selection criteria. The selection criteria will then be used to establish the content and priority of each of the four support levels.

2.1.1 Selection Criteria

2.1.1.1 IBM Capabilities

The most important criteria is "IBM capabilities". When setting out to break new ground, it is best to attack those problems which are most well understood. Later, as experience and understanding increases, the more unfamiliar aspects of the problem can be addressed. This will insure that we get the best possible "return on investment", e.g., for a given investment a greater portion of the total job required will be achieved if it is spent on those areas that are in line with existing capabilities. By structuring a support plan that is closely in line with IBM resources, the cost should be lower, the total development time shorter, and, most importantly, the probability that the plan will be executed should be much higher. This last point is a key one. One of the major reasons for the demise of the S/370 Sensor Base Support Plan (BCA/PI) was that it was a very expensive (30 million development cost), all-or-nothing plan that relied heavily on new, unfamiliar, hardware and software technology for its success.

2.1.1.2 Compatability with FS

The second most important criterion is compatability with FS. It is obvious that we cannot afford a separate product line to support sensor base in its entirety. One of the important objectives is to bring about the integration of sensor based functions into mainline data processing in order to solve the total problem.

For the most cost effective solution, we must utilize standard hardware and software facilities wherever possible. Where changes are justified for performance, they should benefit the entire system and not just sensor base. Where new facilities are required for sensor base, we should attempt to extend these increased capabilities to support other parts of the system as well.

2.1.1.3 Market Size

The third criterion is market size. We should provide hardware and software support for those items which will give us greatest penetration. Since our main objective is revenue growth, we should not overlook the fact that there is more potential revenue involved than just the revenue from sensor base related facilities. For example, bringing sensor input data to a plant computer generates a requirement for large file space to hold it and the related programs, as well as extra compute power to store, retrieve, manipulate and report it. Also, such a system generally requires higher availability and fast response times which justify more system resources than a batch system.

2.1.1.4 IBM Marketing Strategies

The fourth criterion is that sensor base must conform to IBM's marketing strategies.

2.1.1.5 Complete Sensor Base Solution

The last criterion would be the need for FS to provide a complete solution to sensor base. This would require FS hardware and software support for the entire hierarchy including mini-processors to directly attach sensors. It would also imply little (if any) coordinaton with GSD.

The result of applying this weighted set of criteria to the problem of support for sensor based hierarchies is the four support levels named in Section 1.1.

2.1.2 Importance of Software

2.1.2.1 System Functional Support

The overall objective is to provide system functional support for sensor based applications. Hardware is important since it is the skeleton upon which the support will be built; but it is the software that provides form and movement to the skeleton.

The story of the IBM 1800 is a case in point. Announced in 1965, it is generally accepted that the 1800 was not competitive in either price or performance with its contemporaries (with the possible exception of the CDC 1700). The 1800 was, however, a relatively successful program (peak inventory of about 1100 systems) due mainly to its strong software support. The level of software function provided first by TSX and later MPX, was virtually unknown in that market at that time. It set a standard for software support such that today it is de riguerur for a viable sensor based minicomputer to offer a "Real Time Executive" to manage system resources.

2.1.2.2 Real Time Monitor

Another example of the importance customers place on software support in the sensor base area is RTM. RTM is a (hypervisor)

Running in the problem state is not essential to "Hypervision"

support package created by DPD to support the "real time channel" RPQ (2909) on S/360 M65 and up. This software supports the 2909 but since it hypervises OS (running it in problem state) it causes severe degradation to OS and OS jobs. We are told of one such system that was unable to run its printer at full speed. The important point is that customers were willing to settle for such an unsatisfactory system in order to have both real time support and OS functions co-existent.

2.1.2.3 Programmed Airlines Reservation System

1) PARS is not a sensor-based application
2) it could get OS fns by RTSX-like hypervisor.

The third illustration is a negative one. In order to support large airline reservation systems, PARS was written as a tuned system to handle its transactions as efficiently as possible. A major deficiency is that it cannot run background jobs even though the system is only 50% utilized on a 24 hour basis. We understand that PARS customers are unhappy that they cannot have the rest of the OS functions and that the PARS group is now planning to implement OS functions in PARS.

2.1.3 Categories and Support Levels

For the purposes of the discussion to follow, the four levels of support are consolidated into two categories. The first is hierarchical systems communication and the other is non-DP device support. In some cases, especially at the lower levels in the hierarchy, it may be difficult to distinguish between a system and a device. By our definition, a system is capable of being programmed dynamically to perform different functions (or the same functions in a different manner), while a device is characterized by fixed programs and fixed functions. In addition, systems communicate in an interactive mode while devices communicate in a simpler demand/response mode.

2.2 Hierarchical Systems Communications

2.2.1 General

2.2.1.1 Definition

Hierarchical systems communication is defined as communication between two systems that are dependent on each other for services that are necessary for each to complete its respective task.

2.2.1.2 Hierarchical Organization

A computer hierarchy is generally sketched schematically in a manner similar to a management organization chart. There is good reason to do this since the tasks performed at the various levels tend to follow a traditional division of labor. The more specialized detail functions are performed at the lower levels while the more generalized global functions are performed at the upper levels. Hierarchical communications are between two systems only, except in unusual cases (backup). A lower level system is connected to only one higher level system. A higher level system may be connected to several lower level systems but the lower level systems are not aware of each other. A lower level system never "goes around" its immediate superior. When data is communicated upward in the hierarchy, it is retransmitted by each system to the next higher level. Often the data will be manipulated, combined with other data, or summarized at a level before transmission upward. Similarly, data that is communicated downward in the hierarchy is handled in chain-of-command fashion. Again, operations on the data may be performed in order to put it in a form that is appropriate for the next level.

Hierarchically communicating systems are dependent on each other in the sense that each owes its existence, at least in part, to the other. The higher level depends on the lower level for its input. Input may be data to be processed and acted upon, stored or passed upward or requests for data (or programs) to be passed downward. A lower level system may be dependent on a higher one for DP I/O or supervisory services (optimizing extensive computations, scheduling data, etc.).

2.2.1.3 Hierarchies and Networks

An important difference between hierarchies and networks exists. In a hierarchy, the two systems communicating are known to each other. There is usually a single, permanent, open path between the two and the transaction types are relatively few and well defined. That is, this type of communication is "routine". Often, there is a great deal of traffic. In networks of systems, on the other hand, there are often multiple paths between the source system and the destination system. In many cases the path may be by way of intermediary systems acting passively as switches. The sending system may have little control over the actual path taken by its message. The types of traffic tend to be more varied. Communication may be set up in a fashion similar to a terminal session where a temporary logical (and sometimes physical) link is established.

It is obvious that network communications require a much more formal protocol than hierarchical system communications. Such a protocol is unnecessary and would only get in the way of routine "business as usual" traffic in a hierarchy.

2.2.1.4 Reasons for the Growth of Hierarchies

2.2.1.4.1 Data Integration

Why are hierarchies coming into being? Some of the reasons were set forth in the overview. Discussions with IBM internal users in our plants have raised "data integration" as the overriding issue. According to SPD Technical Services 30% of the 8 million points installed in our plants is in support of sensor based applications. Much of it is supporting the automation of production and test operations but the largest payoff is from integrating the data that is a byproduct of computerized operations. Engineers in Burlington directly credit their information system (a four level hierarchy) with a fourfold increase in yield. The reason for this is a data base of raw process data that allows them to track individual lots through the whole manufacturing process and then to relate test results back to process parameters at each step. Not only has this system been responsible for a factor of four increase in yield but they also state that the availability of raw history data shortens by months the identification of quality control problems.

The Burlington data base of routine operational data has also produced another unexpected benefit. Since they track all of the lots through their plant with better than 99% accuracy, the data base is accepted by Price-Waterhouse in lieu of a physical count. This results in a saving of one million dollars per year.

All other IBM plants have similar types of hierarchies installed.

Other important areas of data integration exist in diagnostic monitoring of machines, maintenance scheduling, facilities monitoring, load balancing, etc. The net is that the strongest argument for hierarchies is the integration of data. The greatest potential for improved operations results when all data is accessible and correlatable. This also means that sensor data must be able to be integrated into the system DB/DC capabilities.

2.2.1.4.2 Economy

Another key reason, according to published descriptions, for establishing computer hierarchies is economy. It was also a frequently mentioned reason given by users interviewed by Stanford Research Institute during their study of hierarchical computer systems.

2.2.1.4.3 I/O Costs

Since the cost of I/O devices is quite large with respect to the cost of a minicomputer, we often find several minis (performing related tasks) attached to a "host" system (often a minicomputer itself). The host provides I/O services to the attached minis. Data and program storage sufficient for all can be more economically attached to one system rather than a smaller amount of I/O attached to each individual mini. Printers, too, can be shared in this way as well as logging tapes, etc. As stated previously, hierarchies will continue to be established for this reason. I/O devices have many mechanical components and we can therefore, conclude that the rate of decrease in cost will be much lower for I/O devices than for the non-mechanical parts of the system.

2.2.1.4.4 Cable Costs

Another important economic reason for hierarchies is cable cost. It is often cheaper to install multiple minicomputers each close to its sensor than to run long cables to a single central system. One example of this was reported in Control Engineering, Vol. 19, Number 12. An article on Ford's Rawsonville, Michigan plant describes a hierarchy of three Interdata Model 5's attached to a fourth Model 5. The article states that one Interdata Model 5 has the capability of performing the entire job but that the hierarchy of four Model 5's with their interconnection was cheaper. The difference is due to the large cost of cabling all the sensors to a single location. It should be noted that cable installation costs generally fall in the range of $5 to $10 per foot not including the cost of the cable itself. Another example of cable costs justifying a hierarchy was discovered on a visit to Caterpillar in 1971. They had a pilot installation for a production monitoring system consisting of an 1800 connected to a Weltronics Plant Central system. Based on their experience with that system, their plan for the full monitoring system was to include several S/7's on the plant floor (the 1800 was in the Plant Central Control Room). The S/7's were justified entirely on cable costs that would be saved by using them as "concentrators".

2.2.1.4.5 Ease of Implementation

There exist non-economic reasons for hierarchies, too. One has to do with phased installation growth. It is easier to install several smaller systems one at a time than trying to implement the entire system at one time. In addition, is the sheer inability to do the whole job at once. In an article entitled, "Integrated Manufacturing Systems: Architectural Considerations," in the IBM Journal of Research and Development, Vol 14, Number 6, C. Kinberg and B. Landeck state the following:

"The Time and investment required to attain the level of a properly operating integrated system are considerable; five years seems like an almost unattainable lower limit and periods of seven to twelve years seem more realistic."

It is often possible to computerize some areas sooner with a small individually justified system than to try to justify a large system handling many areas. As the several small stand alone systems come into operation, the benefits or necessity to interconnect them is often seen as the next step in improving plant efficiency and management control.

2.2.1.4.6 Application Autonomy

A point that should not be overlooked is the reality of application autonomy. An engineering group within a plant that has responsibility for a particular function is naturally reluctant to give up part or all of the control over the way that function is computerized. They would rather have complete control of a smaller system handling just that function as opposed to being serviced as part of the function of a larger system in a remote part of the plant.

These small, single application systems whether conceived initially as part of a hierarchy or stand alone, will ultimately need to be coordinated within a plant and in some cases, between plants. IBM plants are already doing some of this interplant coordination and are currently planning to increase this function.

2.2.1.4.7 RAS

The area of RAS is an important factor in the decision to establish a hierarchical system as opposed to a large stand alone system. Increasingly, as more and more parts of a plant are brought under computer control, the concerns about reliability, availability and serviceability are gaining in importance. Our customers are reluctant to rely completely on only one system when it is involved in the production process itself. Cost penalties for failures in this area of a business can be very large indeed. Not only are there large costs accrued in idle manpower and capital equipment as a result of outages, but the income that would be generated by the plant is lost while production is curtailed.

When the control of the production process is spread out over many smaller units as in a well distributed plant hierarchy, the failure of a single system or device has less impact on the plant as a whole. The cost per unit failure is lower due to the smaller quantity of resources involved, fewer men and machines are idled, and there is less scrap and lost production.

It should also be noted that among reliability, availability and serviceability the item of availability is by far the most important. One reason for this is that automated systems are frequently difficult or expensive to start up again once they have been shut down for a period of time. The reason that the duration of an outage is so important is because sensor based applications all involve physical processes. In many cases, these physical processes can continue to run in a degraded manner, or to ride through a short outage due to a float of finished production that is waiting to be processed by a subsequent process. If operation can be restored soon enough, the outage may have little overall effect on the plant. There is generally a critical time period after which the cost of the outage takes a dramatic step up. The length of the initial permissive period obviously depends on the process as does the initial cost. The step in the cost of an outage is usually related to secondary effects. That is, it may represent other downstream units being shut down for a lack of material or it may be due to the cost of purging and restarting the unit that is down.

The following is a dramatic illustration of the outage cost step function. A chemical plant that used live steam to keep the material involved in a particular process in a molten state experienced a failure in the boiler feedwater.

The boiler had sufficient capacity to maintain stream for 15 minutes. If feedwater to the boiler was not restored or the boiler shut down, the boiler would destroy itself in a fairly spectacular way. On the other hand, if the boiler was shut down within the 15 minute period, there would not be time enough to clear the process material from the plant's vessels and pipes. Once steam was removed, the process material would quickly and irreversibly solidify.

If the material solidified, the vessels, pipes, pumps, etc., containing it would have to be torn out, scrapped and replaced with new ones. In this case, those 15 minutes were the critical period. The moral is that in most cases more, shorter outages are more easily tolerated than fewer, longer ones.

2.2.2 Level 1 Support: Homogeneous Hierarchies

Homogeneous hierarchical communications is defined here to mean communications between two systems having the same architecture. In this case it means hierarchical communications between two FS systems. Homogeneous hierarchies of non-FS systems (e.g., two S/7's) will not be addressed. Since we are talking about two FS systems, it is clear that the first support level is concerned with the top few levels of the total hierarchy. The reason that this kind of support is given first priority is a result of applying the selection criteria described in Section 2.1.1, i.e., this is the area of greatest interest and capability for FS.

At these top levels of the hierarchy, sensor based data traffic between systems is probably indistinguishable from other traffic. The only functional distinction that can be made is the hierarchical nature of the communication. That is, the data is transmitted regularly as a routine function and it always has the same destination.

Projections by others in Advanced Systems show that DP resource consolidation will continue into the FS time frame. Therefore, the typical establishment will tend to have only one FS system. This implies that communications between FS systems will typically be between establishments via common carrier telecommunications.

These communications will, most likely, consist of high level, summary-type data integrated with other nonsensor based data. It will be very data base oriented.

With characteristics such as these, it is likely that Level 1 of the sensor base hierarchical support can be satisfied by incorporating it under DB/DC. The only consideration indicated is an increase in hierarchical type traffic. That is, those systems that incorporate a sensor based hierarchy have readily available a large amount of hard data taken directly from sensors on the plant floor.

NOTE: A sensor based hierarchy within a customer's plant will generate a new additional load on top of his normal DB/DC requirements.

2.2.3 Level 2 Support: Nonhomogeneous Hierarchies

Nonhomogeneous hierarchical communications is defined as communications between two systems of dissimilar architecture interconnected in a hierarchy. Assuming one FS system per location, this type of communication represents the bulk of the interlevel communications within a plant hierarchy.

2.2.3.1 Types

From the point of view of FS, nonhomogeneous hierarchies can be broken down into these three main types:

FS system to FS subsystem.
FS system to non-FS system.
FS system to non-IBM system.

The term "FS System" above should be construed to include inboard or outboard subsystems that are neither the source nor destination of a data transmission.

The essence of nonhomogeneous hierarchical communications is end-to-end communications between two intelligent processors programmed to support the functions required in a sensor based hierarchy. This usually (but not necessarily) means customer code resides in each processor.

2.2.3.2 Data Link Performance Considerations

Functionally, there is no difference among the three hierarchical types described above. Physically, all of the systems in the hierarchy will tend to be located in a place that is appropriate to their function in the hierarchy. Location will be determined first on the basis of cost and second on convenience. It is reasonable to expect that in the FS time frame, the order of priority of cost considerations will be:

Cabling and Transmission Costs
Electro-mechanical I/O Devices
Processors

2.2.3.2.1 Cabling and Transmission Costs

Much more multiplexing of data at lower levels in the hierarchy together with higher traffic requiring higher data rates on the fewer cable runs can be expected. An example of this kind of tradeoff at Ford was given in Section 2.2.1.5.2. In line with the above, there is a trend toward more digital sensors and line sharing. Not only is digital data less sensitive to noise than analog, but the possibilities for line sharing to reduce cabling is becoming more important than the cost of analog to digital conversion at (or close to) the sensor.

2.2.3.2.2 Electro-Mechanical I/O Device Costs

The second most cost sensitive area in the construction of sensor based hierarchies is the cost of I/O devices.

Many hierarchies are justified on the basis of the upper level system supplying I/O services to the attached lower level systems. Many others will provide these services along with their normal traffic of production counts, machine status, and supervisory control messages.

A crucial factor in the viability of a hierarchy that uses the upper level for I/O is the ability of the lower level system to retrieve data from its host's file in about the same time as it would be able to from a locally attached file. This will be an important factor in the kind of data link that will be required to support these hierarchies.

In our own plants, hierarchical communications for I/O sharing are carried out using an SMD designed 277K byte per second data link. This link (available as an RPQ on S/360-370 for attaching S/7's) can attach systems up to a mile away.

2.2.3.2.3 Distributed Systems Applications

Applications which are too large for one mini computer to handle are configured with multiple minis. Each mini handles one functional part of the whole job. Coordination and overall control of the application is the responsibility of the host system to which the minis are subordinate. In order for a host system to exercise effective control over its subsystems in such a system, it needs a high performance data link. The reason for this is that the application supervisor must be able to supply programs, data, and control messages to its subsystems with the same kind of efficiency as if the subsystem functions were being handled directly from host resident code.

2.2.3.3 Importance of Level 2 Support

The reason that nonhomogeneous hierarchy support is given second priority is that, on the one hand, it is more difficult to provide due to the dissimilarity of the lower level systems with FS and the multiplicity of different types, while, on the other hand, it is of vital importance in order to give FS a "hook" into the bulk of the hierarchy.

2.2.3.3.1 Non-IBM Subsystems

A question that arises: If it is difficult to support non-FS IBM systems, why provide support for non-IBM minis? The answer is that if we want any significant degree of penetration for FS in the sensor based hierarchy market, we must.

2.2.3.3.1.1 Installed Minis

International Data Corporation's EDP Industry Report, Vol 8, Number 2, projects the installed base of minicomputers used in the application areas of data conversion, data collection monitoring and control at over 67 thousand in 1975. This is triple the number installed in 1971. Most of these systems will be candidates for hierarchical attachment, if they are not already in one. Most of them are purchased (minicomputer vendors need their capital back) and an integral part of productive operations so they are not displaceable. And note: most of them will not be IBM products; S/7 penetration is currently running in the neighborhood of 5%.

2.2.3.3.1.2 Ingested Minis

Besides the already installed minis that will have to be attached, there are the ingested minis. These are systems that are sold as an integral part of some piece of production or test equipment. These systems are not competitors of IBM. The buyer of such equipment generally has no choice in which mini is used to control the equipment he is buying. The Stanford Research Institute's study of minicomputer instrumentation systems in six application areas projects that only 15% of the mini systems installed in 1978 will be general purpose minicomputers integrated into the application by the customer (Figure 3 and 3A). The remainder will either be vendor integrated or ingested.

2.2.3.3.1.3 Special Purpose Minis

We also expect that LSI will cause the number of different "brands" of minicomputers on the plant floor to proliferate. With LSI it will be feasible for machine tool builders, instrument makers and production equipment suppliers to "roll their own" mini to suit their needs rather than accept sub-optimal hardware from the established mini vendors.

2.2.3.3.1.4 Midi Computers

Another reason that it is imperative that we support as many non-IBM minis as we can is to head off the growing trend to midi computers as hosts in hierarchies. Several vendors have seen the requirement to support the small minis with the more sophisticated I/O and computational services of a midi. DEC, for one, is currently touting its DEC System 10 as a host system in a hierarchy. Needless to say, the DEC System 10 (a competitor of S/370 M135 and M145) is within the range of FS.

2.2.3.3.3 Current Marketing Policy

On the negative side, an IBM employee in Atlanta, tells the story of a proposed system for new appplication in a Motorola plant. The system would have required five S/7's

IBM CONFIDENTIAL

STANFORD RESEARCH INSTITUTE
MINICOMPUTER INSTRUMENTATION
STUDY RESULTS

INSTRUMENTATION SYSTEMS DEFINITION

Class I - No Computer
Class II -

Type 1 - User performs integration of a general purpose computer.

Type 2 - Vendor performs integration of a general purpose computer (turnkey installation).

Type 3 - Complete functional product provided including ingested minicomputer

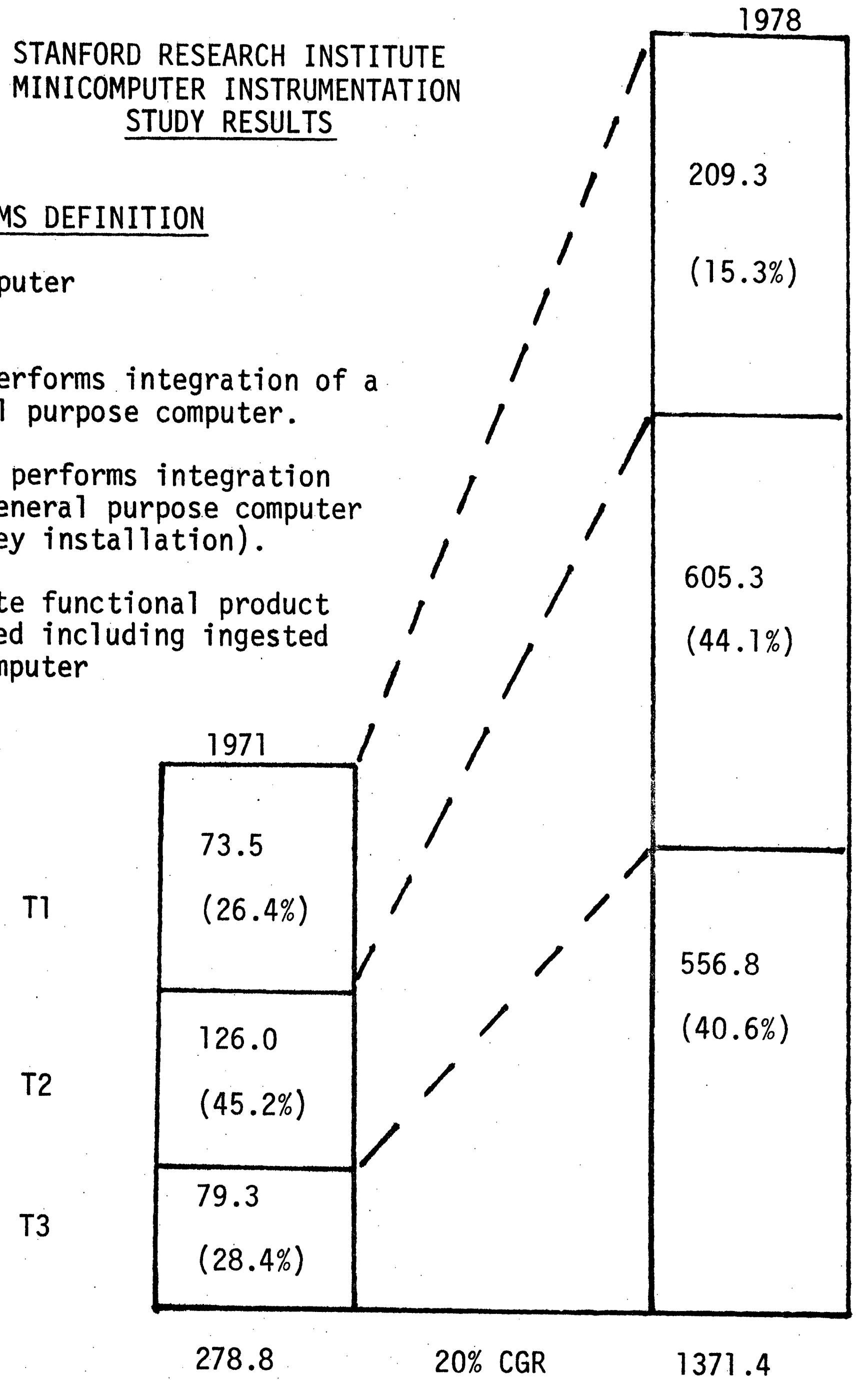


FIGURE 3

FIGURE 3A

Table

COMPARISON OF SIX SELECTED MARKETS FOR IC-BASED SYSTEMS
(Millions of Dollars)
1971-1978

	LAI			B/M			ENV			ETM			P/C			MIG		
	1971	1974	1978	1971	1974	1978	1971	1974	1978	1971	1974	1978	1971	1974	1978	1971	1974	1978
Type 1																		
Class II	$ 6.1	$ 8.7	$ 8.7	$ 1.7	$ 7.7	$ 9.4	$Nil	$Nil	$ Nil	$ --	$ --	$ --	$ 5.7	$ 19.2	$ 37.1	$ 1.3	$ 3.1	$ 6.7
M/C equipment	18.2	23.6	26.9	10.7	24.5	31.8	Nil	Nil	Nil	20.8	27.4	41.5	6.6	18.7	32.1	2.4	7.0	14.5
M/C services	0.1	.3	.3	1.5	3.0	3.5	Nil	Nil	Nil	2.6	3.9	7.2	1.1	3.3	4.3	.5	1.3	2.6
Subtotal	$24.4	$ 32.6	$ 35.9	$13.9	$ 35.2	$ 44.7	$Nil	$Nil	$ Nil	$23.4	$ 31.1	$ 48.7	$13.4	$ 41.2	$ 74.1	$ 4.2	$ 11.4	$ 23.8
Type 2																		
Class II	$ 6.3	$ 6.6	$ 5.8	$ 1.5	$ 6.5	$ 13.7	$2.3	$3.1	5.0	$39.5	$ 78.8	$173.7	$11.7	$ 28.9	$ 78.9	$ 3.9	$ 17.2	$ 89.8
M/C equipment	16.4	23.8	40.2	5.8	26.9	47.9	1.0	1.4	3.5	27.8	42.9	68.5	6.0	13.7	38.8	3.8	11.7	39.5
M/C services	4.8	7.4	13.4	2.0	11.4	25.0	.3	.5	1.2	7.3	15.0	30.3	3.6	11.8	24.6	1.2	3.8	13.3
Subtotal	$27.5	$ 37.8	$ 59.4	$ 9.3	$ 44.8	$ 86.6	$3.6	$5.0	$ 9.7	$74.6	$136.7	$272.3	$21.3	$ 51.4	$142.3	$ 8.9	$ 32.7	$142.6
Type 3																		
Class II	$15.3	$ 35.4	$ 50.6	$ 5.4	$ 27.0	$ 41.9	$Nil	$1.0	$ .6	$ --	$ --	$ --	$10.9	$ 29.9	$109.9	$11.9	$ 36.3	$ 96.4
M/C equipment	12.4	27.3	40.1	2.7	10.5	19.6	Nil	.5	.4	--	--	--	5.5	15.0	58.1	15.2	60.9	140.2
M/C services	4.2	8.9	13.6	1.0	4.6	10.2	Nil	.1	.1	--	--	--	1.8	6.7	22.5	8.8	21.4	17.8
Subtotal	$31.9	$ 71.6	$101.3	$ 9.1	$ 42.1	$ 71.7	$Nil	$1.6	$ 1.1	$ --	$ --	$ --	$18.2	$ 51.6	$190.3	$35.9	$121.6	$2[illegible]1.1
Total	$83.8	$142.0	$199.6	$32.3	$122.1	$203.0	$3.6	$6.6	$10.8	$98.0	$167.8	$321.0	$52.9	$147.2	$406.9	$49.0	$165.7	$[illegible]0.8
Type 4																		
M/C equipment	$ --	$ --	$ --	$ 2.6	$ 8.1	$ 17.9	$0.1	$0.2	$ 0.4	$ Nil	$ Nil	$ Nil	$ Nil	$ Nil	$ Nil	$ 3.0	$ 15.5	$ 30.0
M/C services	--	--	--	1.0	3.8	9.3	Nil	Nil	Nil	Nil	Nil	Nil	Nil	Nil	Nil	1.2	5.5	10.5
Subtotal	$ --	--	--	$ 3.6	$ 11.9	$ 27.2	$0.1	$0.2	$ 0.4	$ Nil	$ Nil	$ Nil	$ Nil	$ Nil	$ Nil	$ 4.2	$ 21.0	$ 40.5
Grand total	$83.8	$142.0	$199.6	$35.9	$134.0	$230.2	$3.7	$6.8	$11.2	$98.0	$167.8	$321.0	$52.9	$147.2	$406.9	$53.2	$186.7	$491.3

attached to a S/370 M135 as a host system. This sale was lost to DEC because IBM refused to attach to other already installed systems (60 computers) within the plant. This customer had a legitimate need to incorporate these systems into a hierarchy. By not satisfying that need, IBM lost not only the S/7's and the M135 but no doubt, a substantial increase in the host system to support those other minis.

Figure 4 reproduces a situation report detailing a similar case with respect to General Motors. These cases are merely two at hand but are indicative of increasing problems. It is clear that either IBM must meet these needs, or face the prospect of giving up business in areas of traditional strength. The areas referred to include both products and customers. The rough treatment given GM bodes ill for our future relationship with this large customer.

The level of support being provided for hierarchical systems today by IBM is minimal. Aside from TP support (which in itself is poor) we only provide S/360-370 to S/7 support, and that is by the Sensor Base Control Unit (SBCU), an RPQ which the customer must program at the EXCP level. We are promised a support package from the Palo Alto Development Center 10/73 to run under OS/VSI (virtual = real only) and a DOS/VS (virtual = virtual) version in 9/74.

It is our contention that the reason we do not have large numbers of hierarchies being installed or planned today is that the level of support being provided is inadequate. Only our largest customers have the resources required to do the necessary systemitizing and support programming. This is an unhappy situation because those are just the things that IBM can do best. For FS we must do a lot better in order to get the business.

When it comes to non-IBM systems in the hierarchy, not only are we providing <u>no</u> support but we are giving <u>negative</u> support as witnessed by the GM situation report cited above.

SITUATION REPORT

CUSTOMER: General Motors Research

SUBJECT: SBCU OEM Interface

BACKGROUND: This situation was brought to our attention on October 17, 1972 by Hal Renken during a review of MIM's strategy. I subsequently attended a mobilization meeting in Poughkeepsie called by Bryan Mayo, MIM Systems Center.

SITUATION: Following the announcement of the Sensor Based Control Unit (SBCU) 5908-N05 GMR was given a demonstration of the IBM plant site PC/OS, TCU/TCA systems which was the basis for the release of the SBCU which utilizes the design of the TCU (Transmission Control Unit). The customer placed an order for a 370/135, an SBCU and a System 7. In subsequent meetings with the customer the B/O determined that it was the customer's desire to tie the SBCU directly to a number of OEM-Mini's in addition to the System 7. GSD would not accept an RPQ for the adapters necessary. Further they would not provide the interface data necessary for the customer to do it and raised the question of possible patent infringement if the customer did develop his own attachment.

STATUS: I alerted Patent Counsel at Endicott of the patent question. They reviewed the situation with MWR Counsel and GSD and concluded that there was no basis for raising the patent issue and it should not have been done. This position was documented by G. Clark.

Wayne Adams is currently escalating the SBCU attachment policy which restricts the termination of the SBCU at System 7 only. The argument for this policy change is that there are thousands of OEM-Mini's in manufacturing. Many customers are now considering the consolidation of data from these systems into a larger "host" computer. The SBCU offers a highly desirable method of developing a hierarchial systems architecture when used as we in IBM use it in our plants. If, however, we continue to require System 7 termination of the SBCU lines the cost rapidly becomes prohibitive in retrofit situations and we will loose the "host" business.

FIGURE 4

At my request Gene Busen has assigned Wayne Vlack to investigate the plants TCA (Transmission Control Adapter) technology and experience so that we may be in a position to respond to an RPQ if Adams is successful. I have also initiated a resurrection of some OEM-Mini interface effort that was done by Amal Lueppert for review.

TIMING: The GMR order is scheduled for delivery in Mid January, 1973. The attachment of foreign devices is not required until approximately August, 1973.

W. R. Couch
11/06/72

FIGURE 4 (continued)

2.3 Non Data Processing Device Support

2.3.1 General

2.3.1.1 Definition

The second category of support to be discussed is that for non data processing I/O devices. Non data processing devices in this context are those that input or output to transducers rather than media or people.

The importance of these devices in a sensor based hierarchy is central. They are the interface with the external devices that put the word "sensor" in sensor base. We are speaking here of devices that directly attach to FS systems as devices. Non DPIO devices that attach to non-FS systems in the hierarchy will not be treated here since, from the point of view of FS, I/O relating to those devices comes under the heading of nonhomogeneous hierarchical communications with the system or subsystem to which they are directly attached.

2.3.1.2 Applications

Although we would expect to find most of the transducers in a hierarchy attached at the lower levels, there are some applications where direct attachment to FS is appropriate. One of the traditional applications for mainline systems is plant floor data collection. In the past, data collection has been almost an entirely manual operation. The future will certainly mean more automation of this function for three reasons:

- °Increasing labor costs will enforce more efficient use of manpower: Why pay a man to stand at a terminal if a machine can do it quicker and cheaper.
- °Increasing management control for production efficiency will generate a need for a greater volume of data from the plant floor which would tie up productive manpower in data entry operations.
- °Increasing the productivity of men and machines means that production managers will need to know plant status quickly and continuously if they are to manage the plant in "real time".

The last two points above also create a need to capture data closer to the point at which it is generated.

Areas of the plant that are not computerized are candidates for this kind of support. Manual and fixed sequence machines fixed program controllers ala the PDP 14, weighing stations, plant facilities and environment, pump (billing or inventory control applications), and material balance for scrap and shrinkage control are some examples.

2.3.2 Level 3 Support: Automatic Data Collection

2.3.2.1 Definition

Automatic Data Collection is defined as the automation of manual data collection. Its place and function in the sensor based hierarchy is much the same as that of manual data collection. In fact, it will be found together with manual terminals serving those functions that it is not feasible to automate (e.g., clerical functions). A fundamental assumption is that automatic terminals will be used in addition to people oriented terminals and will almost never be found alone. The reason for this assumption is that Level 3 support is aimed at installations that are evolving from, or adding to, a manual plant data collection system. The more esoteric forms of direct transducer I/O are treated by Level 4.

2.3.2.2 Selection Considerations for Level 3

The reasons that automatic data collection was chosen as third priority in the layered support plan are several:

°In the first place, data collection is a known system function in IBM. As such, it should be relatively easy to identify and quantify the hardware and software requirements in this area.

°Second, it is evolutionary. We already have a system that has a small amount of automatic capability: The 2790.

°Automatic data collection should be easy to sell and install since its relationship with manual data collection applications and support make it familiar.

°Thirdly, it does not place much of a burden on our traditional systems concepts. That is, it does not require extreme data rates or microsecond response times in order to be viable.

°Lastly, if the support plan is terminated after Level 2 and no automatic data collection support exists on FS, then the requirement probably will be satisfied by support on non-FS systems (IBM and others) attached to FS by way of Level 2 support.

2.3.2.3 Characteristics

The characteristics that automatic data collection requires to be viable are much the same as for manual data collection:

°The terminals must be inexpensive and simple since there are likely to be a large number of them in an installation. That is, they have to go where the information is rather than waiting for a person to bring it to them.

°For the same reason, they must be rugged and have a high tolerance for environmental hostility.

°Above all, they must be inexpensive to install and maintain, both of which have a high labor content.

One example of a recent effort to provide automatic data collection functions is the IBM Industrial Translator System produced by the Sensor Base Custom and Application Systems area of GSD. This system, designed to attach to S/7, can attach more than 4000 sensors up to a mile away. The terminal has a $70 purchase price. Its failure rate is claimed to be .0001% per thousand hours and it is serviced like a light bulb.

2.3.3 Level 4 Support: Other Non DPIO Devices

2.3.3.1 Definition

Level 4 support could be defined as everything not covered by the other three. Generally speaking, it is meant to cover the area that has been known in the past as "direct sensor I/O".

2.3.3.2 Selection Considerations for Level 4

The reason that it has lowest priority has more to do with the selection criterion of "IBM capabilities" than with a lack of market. In 1971 the high performance direct sensor I/O market was credited by both GSD and SDD to be 30% of the total. Today, this slice would probably be pared due to the emerging importance of hierarchies but still represents a substantial opportunity. The reason that direct sensor I/O (DSIO) support is Level 4 has a great deal to do with the purpose behind trying to find a structure for the sensor base function that would allow a layered support plan. Prior to this point, the high performance DSIO requirement has been so intertwined with the total sensor base support requirements that there has been little progress due to the technical difficulties involved. By putting DSIO support at Level 4, it has not been relegated to a position of unimportance. Rather, we have skimmed off the easier more mainline-like functions so that they are out of the way. The result is that DSIO support may be addressed separately and clearly without the intrusion of these other issues.

Except that if the support does not anticipate (or deliberately precludes) the L4 needs, it can never be part of an integrated system.

2.3.3.3 Technical Problems

There is no question that this area presents technical difficulties for mainline systems. Many of the applications for DSIO place severe performance requirements on the driving system. These requirements may be for extremely high data rates or very fast response times or both.

This is where the problem lies. Mainline data processing systems have traditionally been designed to meet only the requirements of data processing I/O device rates and response requirements. The System 360 (and 370) I/O architecture was specifically designed to support only non time dependent I/O devices. (Exceptions such as inclusion of the real time control of the 1419 check sorter were costly in effort and efficiency).

By and large the System 360 and I/O architecture has been adequate for the purposes for which it was designed. In order to support high performance sensor I/O, however, increasingly drastic changes were required. The 2909 mentioned earlier was provided with separate CAW/CSW locations to avoid contention with other I/O for the normal locations. It has a "store status" command to avoid extra SIO instructions and supervisor overhead. In addition, it has a kind of priority interruption capability to assist in shortening the queueing time for high priority events.

The BCA/PI architecture developed (but never implemented) for System 370 sensor base support was a high speed multiplexor (as was the 2909) incorporating a true priority interruption mechanism (unforunately not integrated with the rest of the interrupt structure). It went a step further than the 2909 by providing separate status areas for each operation in progress.

Pfui!

It is interesting to note that this idea of Operation Request Blocks has been incorporated into the FS definition.

Fortuitously for Level 4 support, an objective of FS is to produce a highly responsive architecture. If the architecture can be made sufficiently responsive to meet the needs of Level 4 support, then all parts of the system will benefit. In any case, we believe that adequate responsiveness will require a non-traditional approach. This point is discussed further in Section 4.2.

But this is (albeit another tradition)!

2.3.3.4 Support Characteristics

2.3.3.4.1 Traditional View

The number of different kinds of devices to be supported by DSIO is, of course, large. They can be put into four main classes: digital input and output, and analog input and output. None of this is news to anyone who has been even remotely associated with sensor base. The unfortunate thing about such a traditional classification of sensor I/O is that IBM and other vendors have allowed it to dictate the way hardware and software have been designed. The 1800 was a prime offender in this respect and the S/7 is not much better. It is another example of solving IBM's problems rather than our customers'. An implicit assumption in these systems is that the classes of sensor I/O have relative importance but that specific sensor inputs and outputs do not. The only exception is inputs that directly trigger interruptions. The concept that an entire class of sensor I/O devices should be driven by a single piece of serial code and have its interruptions serviced at a single priority level completely ignores the fact that each of the transducers is attached to a different part of the physical plant, and may perform a different application function, having different servicing and priority needs. Such systems are only adequate where utilization is so low that contention for priority is not a factor. The point to be made is that even in these systems designed specifically for "real time" responsiveness, improvements can be made in organization.

2.3.3.4.2 Customer's View

The customer's view of his sensor I/O is entirely different. To a customer it is the machine or the furnace, or the boiler, or the digester that is the I/O device that he is interested in. Each has its own requirements for control. Each may have several different transducers of different classes having different service requirements. The action of all of these must be coordinated in order to perform the function that the customer wants, i.e., the operation of his "device". The sensor base customer's problem is analogous to taking away all the control unit functions from S/370 and having them performed directly by IOS!

By taking a hard look at what the DSIO customer really wants to do, we believe that a great deal of support for Level 4 can be supplied in FS architecture that has the required net performance.

3.0 Technical Objectives for Layered Sensor Base Support

3.1 Introduction

The purpose of this section is to state the technical objectives for the support of sensor base hierarchies in FS. These objectives are broken down into the four support levels to which they apply. As before, each support level pre-supposes the implementation of the preceeding levels.

What follows is considered a technical objective because it deals with the hardware and software function, performance and RAS characteristics needed to support sensor base hierarchies. It is considered an objective since the sensitivity of the marketplace to the various items has not been measured.

These technical objectives are quantitative wherever possible given the current level of understanding. In some cases, they specify an implementation.

Items which are not considered or only implicitly considered are development cost, prices, and marketing plans and capabilities.

The underlying assumptions include some form of S/370 support for sensor based hierarchies and the existance of S/7 follow on products in the FS time frame.

3.2 Support Objectives for Homogeneous Hierarchies

As stated in section 2.2.2, communications between two FS systems in a sensor based hierarchy consists mainly of high level summary type data. Given the power of FS systems, these communications will nearly always be between plant sites, or a plant site and a headquarters location. Therefore, virtually all these systems will be connected by way of common carrier data links.

3.2 Continued

Hierarchies incorporating multiple FS systems will be found mainly in our medium and large customers' operations. The high degree of integration in these businesses will require data link capabilities up to the fastest then generally available. This means that FS to FS communications must support the T1 carrier (1.5 mega bits/second).

The software facilities needed to support homogeneous hierarchical communications are assumed to be the same as those for DB/DC generally. The reason is that at these upper levels of the hierarchy, the data is more business oriented than sensor oriented. The specification of any additional facilities that may be required must wait until the DB/DC support plan is established.

3.3 Support Objectives for Non-Homogeneous Hierarchies

The support for non-homogeneous hierarchies in FS is critical. Assuming that FS to FS communications will be adequately supported by the DB/DC plan, support for FS to non FS system communications is the first place where development dollars must be spent specifically for sensor based hierarchies. The return on this investment stated simply, is the capability to have FS systems in a sensor based hierarchy at all.

3.3.1 Attachment of Lower Level non FS systems

The ability to incorporate FS systems into a sensor based hierarchy must not be taken lightly for competitive reasons. All sensor based hierarchies demonstrate requirements for the traditional data processing functions. These requirements increase as one moves up in the hierarchy. These functions are the ones that IBM can best provide. In order to be able to propose, sell and install systems to provide these functions in the hierarchy, we must have the capability to attach the lower level non FS systems.

3.3.1 Continued

If IBM does not have this capability other vendors will make inroads into our traditional market.

A strong support plan for non FS systems can have a powerful impact on FS acceptances in two ways.

First, if FS can be a host system to the lower levels of of a sensor based hierarchy, it could well justify the existance of a plant site (FS) computer. Without this capability, the plant may be able to get along with a remote batch terminal to an off-site system for its normal data processing requirements. With a sensor based hierarchy to support, however, there must be a local system due to the data volumes and response times required.

Second, strong non FS system support can justify the migration of data processing I/O devices upward in the hierarchy to the FS system. This means more FS peripherals and fewer minicomputer peripherals. The hardware facilities required to support non-homogeneous hierarchies are summarized in Figure 5.

3.3.2 Data Link Capability

There are two types of data links required. Off-site communications will require attachment to a common carrier facility. The speed of these data links will be governed by the tarrifs expected to be available. They will be the highest speed that can be economically justified because they are system to system and therefore, not subject to the constraints of people-to-machine communication.

The other type of data link is for in-plant communications. In-plant communications need not be constrained by common carrier requirements. There are two kinds of in-plant data links required.

3.3.2.1 High Performance Link

The first is for use in very tightly coupled coupled hierarchies where the satellite systems are very dependent on host compute power and I/O facilities. This kind of data link will be a functional replacement for the high speed in-plant transmission capability currently used within IBM (the TCU) and available as an RPQ for S/370 to S/7 attachment (SBCU). As such, the throughput of this link must be comparable with the capabilities of miniperipherals attached locally to the satellite systems.

	DATA LINK	SPEED	DISTANCE	MAX. #
Off Site	Common Carrier	Common Carrier	N/A	~10
In Plant High Performance	Serial Coax	500 KB/S	6K Ft	~200
Low Cost	Serial	1 K B/S	20K Ft	~200

FIGURE 5

3.3.2.1 Continued

Since this kind of link will be used mainly to replace local files, the record lengths involved will tend to cluster around two sizes. Programs and data being transmitted will produce records that are of the order 1-10K bytes, while control transactions (e.g. to ask for a specific program to be transmitted) will be less than 100 bytes. In our own plants, records up to 25K bytes are routinely transmitted (Kingston). Given the record lengths involved and the delays introduced by transmission and host software it is estimated that a transmission speed of the order of 500 K bytes per second will be required in order to be competitive with a 200K byte per second local file.

3.3.2.2 Low Cost Link

The second type in-plant data link required is a low cost, low speed version that uses twisted pair wires. The main reason for twisted pair attachment is the ability to use wires that are already installed in the plant. This capability can make the difference is justifying a satellite system when the installation cost of new cable would be prohibitive.

The data rate of 1K bytes per second is called for on this data link. This speed is easily achievable on twisted pair wire. It is also comparable with hi speed paper tape readers (1K cps), 600 line/min printers, and about 3 times the current speed of digital tape cassettes. Therefore, it should be a viable alternative to these devices in many instances.

The cable distances specified in Figure 5 are different for good reason. Coaxial cable runs of over 6K feet probably cannot be justified over alternative means. For example, at the rate of 10 dollars per installed foot, 6 thousand feet would cost 60 thousand dollars. This is equivalent to 1.2 million bytes of local disk storage at a $.005 per byte purchase price. Also, 6K feet is adequate for most plant sites (the SBCU/TCU can attach S/7's up to a mile away). The low speed, twisted pair link is specified as 20K feet because the use of previously installed wire will undoubtedly require a circuitous routing.

3.3.3 Buffered Transmission Control Subsystem

In view of the transmission speed differential of even the high speed link with the proposed FS loop (2 MB), it appears that a buffered transmission control subsystem attached to the loop is indicated. This controller should have very high performance to match the loop capabilities. It should also be able to handle all in-plant data links. This means that the low speed lines at least, must be multiplexed among themselves and the low speed lines multiplexed with high speed lines. High speed lines need not be multiplexed among themselves.

3.3.3.1 Record Lengths

Since the transmission of storage loads to the satellites is a normal function of the transmission subsystem, long logical records will need to be broken down into smaller physical records for transmission. Smaller physical records will facilitate better utilization of buffer space within the transmission subsystem. More importantly, shorter transmission blocks will improve the degradation due to retransmission for error recovery particularly for the twisted pair lines. Physical record length should be chosen such that the net data rates are no less than 300 KB/S and 600 B/S for the high performance and twisted pair links, respectively.

3.3.3.2 Signal and Line Discipline Characteristics

The signal characteristics and line discipline to be implemented are not specified here. However, since in this marketplace IBM is not the leader, we are in a position somewhat similar to the PCM I/O vendors. FS must have a satellite or intelligent subsystem interface that will make it easy for us to attach to them. In this regard, first attention should be given to the possibility of building to existing standards. If existing interface standards (e.g., EIA or MIL spec) are not suitable, we should develop our own such that the ease of implementation by non IBM vendors is an important consideration. By doing this, IBM will offer a more attractive solution to our customers, thereby enhancing FS.

IBM CONFIDENTIAL

3.3.3.3 Preemptive-Resume Priority Queueing

An underlying assumption that should be noted is that preemptive-resume priority queueing techniques will be implemented throughout wherever feasible when there is the possibility for queues to build up. This applies to both hardware and software. More on this subject can be found in Section 4.2.

No? Not from a programming point of view

3.3.4 Data Link Alternative

The following alternative to the above coaxial cable and twisted pair data links should also be investigated. An increasingly higher percentage of the cost of creating sensor based hierarchies will be due to the cost of installing the many cables that are required in order to link the elements of the hierarchy together. The reason for this has been stated earlier but bears repeating here. The costs of all the other components of a working hierarchy are expected to decrease - even programming. Cable installation is very labor intensive and can therefore, be expected to greatly increase in cost. This fact creates drag on the rate at which hierarchies will come into existance. It is also an opportunity for IBM. At the current installed cost of 6 to 12 dollars per foot, a great deal of IBM hardware can be justified as a replacement for laying cable (and re-laying as systems are moved about in the plant).

The objective is to replace a labor intensive communication link with a capital intensive one.

The possibility for introducing wireless or partially wireless data links within the plant should be pursued. By use of multiple, low power, stand alone UHF relays, it should be possible to keep transmission distances down to a few hundred feet. There should also be adequate bandwidth to handle most, if not all, in plant data communications. With high frequency and low power, there should be no problem keeping transmissions within the walls of the plant.

To IBM, these data links become an entirely new source of revenue since we get nothing for cable installed by the customer. For the customer they represent continuing savings.

3.3.4 Continued

That is, not only does he save on labor cost for initial installation over installing miles of cable, but he saves again every time he must move equipment around in his plant. This last aspect is continuous in most manufacturing plants.

Once the physical wire link between system and subsystem or terminal has been broken, other advantages accrue. One is the independence of physical location. That is, if a subsystem or terminal is moved, not only does it not have to be re-cabled, but there is no disruption in the rest of the system - either hardware or software. Another may be the possibility for portable terminals to be used on the plant floor.

The main problem to be attacked, however, is the one of increasing cost for cable installation. It appears likely that this problem will become so severe in the FS time frame, that alternatives will be found. The opportunity for new revenue to IBM should not be ignored.

3.3.5 Software Objectives

The software functions that will be required to support communications in a nonhomogeneous hierarchy are, at this point in time, considered to be the same as those developed by the recent joint SDD/SPD/GSD task force on intelligent subsystems. The task force report is included in the appendix, and the functions are reproduced here.

3.3.5.1 Subsystem Program Load

A. Initial (bootstrap)

1. At request of:

   a. Host application

   b. Target subsystem application

   c. Different subsystem application

   d. Target subsystem Power on or IPL button.

3.3.5.1 Continued

2. Source of Program Load is:
   a. Host library
   b. Target subsystem library
3. Controlled Termination

B. Program Load (Overlay)

we want to eliminate necessity for overlay planning on description.

1. A Request of:
   a. Host Application
   b. Target subsystem application
   c. Different subsystem application
2. Source of Program Load is:
   a. Host library
   b. Target subsystem library
3. With or without execution
4. With or without associated data transmission.

3.3.5.2 Invoke /Synchronize Program Execution

A. In host from subsystem application

B. In host from host application.

C. In target subsystem from host application

D. In target subsystem from another subsystem application.

E. Without Partition/Region Constraints
   1. Sub tasks of real time job
   2. Other "On-line" partitions/Regions
   3. Background (Batch)

should be no distinction between real-time job & batch/background in description (only in how they actually behave with respect to time).

3.3.5.2 Continued

F. With or without associated data transmission.

G. Transparent to paging, library, references, etc.

H. Program controlled event posting.

I. Timed interval/event basis.

3.3.5.3 Timer Support

A. Invoke/Synchronize Host/Subsystem Application Program

1. Starting at "n" o'clock.

2. Every "n" time units

3. Until terminated by

a. "M" Repetitions

b. User

B. Using parameters initialized/modified by host/subsystem/application programs.

C. With access to time-of-day clock from host/subsystem/application program.

D. Synchronized to external user clock.

E. Watchdog timer services.

3.3.5.4 Data Interchange

A. Between User, Industry, or SCP Programs

1. In host

2. In host and subsystem

B. Between Named Program and Named Data Set

1. In host

2. In host and subsystem

3. In subsystem

C. Between Host or Subsystem User or Industry Programs and Host Data Base.

D. From Subsystem Application Program to "System Output File" (card/printer I/O).

1. Open and close file (release to punch/print)

E. With option for program execution.

3.3.5.5 Data Set/Program Library Maintenance

A. Create/Delta/Maintain named data sets

1. On host at request of host application

2. On host at request of subsystem application

3. On subsystem at request of subsystem application.

4. Main storage and secondary storage resident files.

B. Create/Delta/Maintain subsystem program library

1. On host at request of host application application.

2. On host at request of subsystem application

3. On subsystem at request of host application

4. On subsystem at request of subsystem application.

3.3.5.6 Connection Modes/Speeds

A. Support full range

1. Common carrier

2. In-Plant

B. Allow integrated and standalone control units

C. Provide system responsiveness commensurate with line speed.

D. Mode/Speed Transparency required.

3.3.5.7 Operational Requirements

A. 24 hour, 7 day environment; RAS, OLT, Recovery Requirements.

B. Unattended subsystem operation.

C. High volume, multiple subsystems on host.

D. High performance target: Host response equal or faster than subsystem disk.

E. User interface consistent with high level languages.

F. Dynamic library and data file replacement.

G. Security: unauthorized access control.

Although the task force report on intelligent subsystem support requirements is aimed at S/370, it is relevant to FS. The functions required are the same and FS plans must be consistant as a S/370 follow on. A more detailed explantation of the software requirements listed above can be found in section 3 of the task force report in the appendix.

3.4 Support Objectives for Automatic Data Collection

As stated in Section 2.3.2 automatic data collection is the sensor based part of the overall data collection requirement. It is an important objective that it be incorporated with the manual data collection subsystem rather than with the traditional sensor I/O facilities. The reason for this is that data collection is installed for management purposes. This means that it is generally on line to the plant data processing system rather than a sensor based system. This statement may appear erroneous in view of the popularity of the S/7 as opposed to the 2715 as a controller for the 3790 plant communications system.

However, it is explained by the difference in cost between the 2715 and the S/7, rather than any functional difference in application. Also, data collection is usually installed on a plant data processing system before it becomes a host system in a sensor based hierarchy.

3.4.1 Hardware Considerations

Automatic data collection terminals differ substantially from manual terminals as would be expected. They are required to exist in hostile environments and are found close to, or physically attached to, the customer's machines from which they are collecting data. Since each of these terminals is dedicated to a single customer machine, it requires only limited input/output capability. By the same token there must be the capability to attach thousands of them to the system. The industrial translator system mentioned in Section 2.3.2 has the capability for more than four thousand such terminals.

3.4.1.1 Input/Output Interface

The characteristics of automatic data collection are such that the number of inputs required dominates the outputs. Further, inputs and outputs are almost always digital. Inputs are required for data; outputs are generally only required for control functions external to the terminal. Inputs must be capabile of sensing the status of:

a. Dry contacts (limit switches, relay status, etc.)

b. Logic levels (external digital meters, etc.)

c. AC power (motors, lights, solenoids, etc.)

3.4.1.1 Continued

Outputs are required mainly to switch 120V AC for relays and indicator lamps. Input and output interfaces should be provided with optical insolation from the data transmission part of the system and protected from accidental shorts and high voltage being applied.

The number of inputs and outputs required per terminal is small but variable. A simple manually operated machine such as a stamping press may require 1-4 bits of input and 0-2 bits of output. A more complex, automatic machine may need up to four times as many. Terminals attached to sources of coded data such as weighing, measuring, or testing machines can be expected to require the largest number of input bits. These fall in the range of 2-4 bytes. Requirements beyond this will probably mean that there is much variable data involved that needs to be entered by a human operator at a manual terminal.

There are two kinds of digital data to be handled: static and dynamic. Static data is the most common. It consists of uncoded status bits and coded numbers from external meters, thumbwheels, and counters. Dynamic data is represented by piece and cycle counts entered by the continuous cycling of contacts connected to the digital inputs. Due to the mechanical nature of the events being counted, cycle rates of more than 3 per second are seldom encountered.

3.4.1.2 <u>Data Link</u>

Because the cost of cabling thousands of terminals back to a central system can be enormous, special attention must be paid to it. Line sharing, submultiplexing and line concentrating must be pursued vigorously in order to lower these costs to a minimum. As far as the type of cable to be used is concerned, it should be noted that if a <u>new</u> cable must be installed, the cost is relatively insensitive to the type of cable.

3.4.1.2 Continued

The kind of physical link implemented must also take into account the fact that customers continually move machines and terminals after initial installation. In this mode of operation, loops for example, can be very expensive even though a loop seems to offer the minimum amount of cabling. Moving a few terminals around in a plant may require that nearly the entire loop must be reinstalled at high cost. It may also mean that the entire plant data collection system must be down one or more times in order to move the terminals for one department to the other side of the plant.

3.4.1.3 The Need for an Intelligent Controller

It is pertinent to both the hardware and software objectives to note a major difference between manual and automatic data collection. Manual data collection is transaction oriented. A relatively large amount of data (10-100 bytes) is entered interactively by an operator with long periods between transactions (except attendance recording). In automatic data collection, data is entered in small amounts in a relatively continuous manner, asychronous to any polling by the data collection system. One way to take advantage of this difference would be to bring data from automatic terminals only as far as an intelligent subsystem of the UC variety. This controller can then deliver data to the host system on demand by the host application or by exception, based on user supplied tables loaded from the host.

3.4.1.3 Continued

These same tables can also be used to cause an immediate response from the controller to the same or different terminal in order to operate an alarm or shut off a motor, etc. This would provide a kind of reflex action which can be followed up later by a more comprehensive procedure from the host system. With the provision for simple table driven reflex action by a controller, the host response requirements should be no more stringent than for manual terminals. Reflex response from the controller should reach the terminal less than .5 seconds after the triggering status change in the terminal.

The automatic data collection controller would appear as a manual terminal to the host. The continuous updating of information from the automatic terminals would then be handled by the controller, completely hidden from the host system.

3.4.2 Software Considerations

A major software objective is that automatic data collection support be integrated into, and have the same interface as manual data collection.

In addition, the following capabilities are necessary:

- Access to data from an individual terminal on demand.
- Access to data from a group of terminals together (department) on a single demand.
- Ability to associate one or more terminals as a logical, accessible, named entity so that application programs can deal with functional plant floor units.
- Ability to extract and summarize data relating to multiple plant floor units. (reports and displays)
- Ability to have a functional view of the plant floor regardless of physical location or connection method of the actual machines and terminals.
- The ability to logically set plant floor units "off-line" and "on-line". (In order to accrue downtime and ignore cycling during setup and maintenance.)

3.4.2 Continued

- The ability to set and change on a functional unit basis, the period for updating status and counts in the host system.

All of the above can be summarized by the requirement for the application programmer to be able to construct a view of the plant floor as a set of logical machines or devices. He should be able to have multiple views of the same data/ machines; i.e., a department as a whole, or individual machines within the department. This is the customer's equivalent of having varying degrees of visible device dependency. The application programmer should not be required to know about more "device dependency" than he needs to; i.e., he should not have to program or re-program for the way physical connections are made or changed.

3.5 Support Objectives For Other Non Data Processing Devices

Section 2.3.3.1 defines "other non data processing devices" as direct sensor I/O other than automatic data collection. The reason for this is that many customers have the need to to attach "devices" to computer systems and no computer vendor provides these individual type of attachments.

The orientation of this section is toward identifying objectives for the support of customer supplied, non traditional devices. This differs from past efforts that aimed at just defining transducer to computer interfaces and support.

3.5.1 Attachment Environment

The fundamental assumption for this level of the support strategy is that the customer attaches his devices to an FS system because no lesser system has the power to handle it. This means that if the device does not need the data rate, compute power or storage capabity of an FS system, it will be attached to some other, lower level, less expensive system. Therefore the usual slow to medium speed "process control" like functions will not be handled directly from an FS system.

The kinds of applications that will require direct attachment of external devices via sensor I/O are those that are considered leading edge or state of the art today. These include the following examples:

- Hybrid Simulation
- Telemetry Data Processing
- Nuclear/Other High Speed Data Acquisition
- Missile Launch and Flight Control
- Missile Defense
- High Speed Film Digitizers
- Gapless Digital or Analog Tape

Data will arrive at the system either "raw" directly from sensors or be delivered by a pre-processor that is hardwired or only minimally "intelligent" as far as the application in the host system is concerned. There are just two main classes of problems to be solved. All applications are either subsets or a mixture of these two. The first of these is the hybrid simulation problem. It is characterized by repetitive, very fast cycles of input-compute-output that must be of constant duration and synchronized with external events. The other is the gapless tape problem.

3.5.1 Continued

In this type of application data arrives in a continuous stream at a very high, uniform rate and must be processed on the fly. In either case, failure to keep up results in an unrecoverable data overrun. The consequences of a data overrun vary with the application. At a minimum, the computer run will have to be restarted, usually from the beginning. On-line experiments are usually the most expensive to re-run due to the often large amount of equipment and manpower involved outside the computer room.

3.5.2 The Hybrid Simulation Problem

The key to hybrid simulation is fast response by the digital computer system. In hybrid simulation, an analog computer is directly attached to a digital computer. Analog computers are very fast since they process everything in parallel. They have two drawbacks for the purposes of simulating complex systems, however. Extreme accuracy is often impossible or prohibitively expensive and they do not deal well with mathematical functions that are not smooth. These two aspects can be handled to any degree required by a digital computer. Ergo, a hybrid computer system is required, combining both analog and digital computers.

The problem is that the two must be synchronized In operation, the analog computer processes its portion of the calculations and passes the results to the digital computer. The digital system uses the analog computer's results to perform its part of the calculations and passes the results back to the analog computer. The analog computer uses these data for new calculations and again passes the results back. Ad infinitum. Since the purpose of the simulation is to compute the actions of a physical system, these iterations must be clocked at a specific, known rate in order to provide the time base for the simulation. If the hybrid systems fail to keep in synchronization, the simulation produces erroneous results and the run must be stopped.

3.5.2 Continued

In order to establish the time base for the simulation, the activity within the digital computer is organized into externally selected time slots called frames. The analog computer is always much faster than the digital computer so the frame time is chosen as the shortest period in which the digital computer can reliably perform the required input-compute-output cycle. While frame times are generally of the order of a few milliseconds, the key to success in this application is how short the frame time can be. The length of the frame represents throughput to the customer. That is, the shorter the frame, the more simulating he can do per unit time. To this end, simulations of real systems are run at many (thousands in some cases) times "real time".

In summary, this application is highly response oriented, synchronous, and overrunnable. Response requirements are usually stated as "as fast as possible" or "microseconds".

It is normally carried out on a large general purpose system rather than a special purpose one because:

(a) The cost of such a powerful, special purpose computer would be prohibitive.

(b) Even a general purpose system can only be justified by being able to do more computing tasks when it is not performing hybrid simulations.

Figure 6 shows a schematic of a frame. Within the frame there are four subdivisions or phases:

- INPUT
- COMPUTE
- OUTPUT
- CONTINGENCY

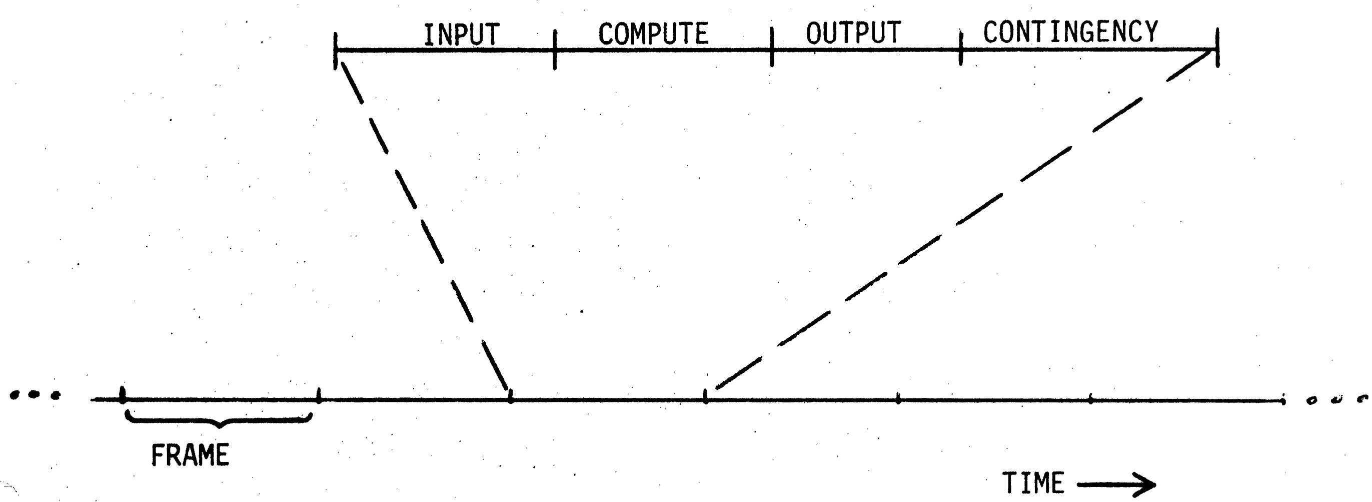


FIGURE 6

3.5.2 Continued

Since this is a synchronous process it is not possible to overlap any of these periods. The computing cannot begin until the data has been inputted; the output cannot be done until the values have been computed. (Note that is is possible, of course, to overlap input and output with other simulations that may be going on concurrently.)

If it is not possible to overlap the sub intervals, then each must be shortened as much as possible. For example the length of the compute interval is dependent on the MIP rate of the processor. If we assume the customer has the most powerful processor he can afford, complete with floating point hardware (most of the calculations are floating point), then the problem is one of reducing the overhead of non productive (to the application) processing time. This is the main reason that very fast task switches are called for in this application and why standard OS cannot be used. Unproductive overhead robs MIPS from the application.

3.5.2 Continued

It would appear that one way to shorten the I/O intervals is to raise the data rate. This is only marginally effective. Hybrid simulations generally require less than 100 bytes of data to be exchanged between digital and analog computers. In this case, again, the overhead in I/O is the major burden. The difference between a 1 megabyte data rate and 2 MB is only 50 microseconds for a 100 byte transfer. The time between Input request and the start of computation is far greater than this.

The contingency interval should also be investigated for possible shortening. Contingency is included to allow for variations in response time. Variations arise due to the random effects of interruption disable time, channel interference, hardware and software queues and variances in instruction execution. A system organization that makes system action more predictable and uniform can result in a shorter contingency and shorter frames.

this problem is the origin of the RTXP approach

It goes without saying that any improvements in these areas have a direct effect on the entire system, particularly for response oriented applications.

3.5.3 The Gapless Tape Problem

Gapless tape type applications (also known more elegantly as data streaming) have been around for many years. They are on the increase due mainly to the more recent requirements for analysis of vast amounts of continuously recorded data from such sources as telemetry, radar, and seismic recordings.

Data streaming applications are run both on-line and off-line. On-line applications are represented for example, by the direct processing of live telemetry data from a weather or other surveillance satellite in real time. One such installation run by the U.S Air Force on the west coast is comprised of a S/360 M40 pre-processor coupled to a M75J for analysis. This system accepts data continuously at a 1 megabit rate and pushes the Model 75 to over 90% CPU utilization while processing "interesting" data.

3.5.3 Continued

Off-line data streaming applications are those where it is either inconvenient to bring data directly to the computer in real time, or where there is a data rate mismatch with the computer that precludes doing so. An example of the former is seismic data reduction where the raw soundings are recorded on magnetic tape at the field site for later processing. The latter case of data rate mismatch occurs when either the data is generated at a faster rate than any computer could process it or when it is generated at such a slow rate that it would be inefficient to process it in real time.

In all these applications, data rates are adjusted such that data is delivered continuously as fast as the system can accept it. If the source of the data is analog, it is generally converted to digital as it is read in.

The problem in handling these applications is that the input record is "infinitely" long. The word "infinite" is used to convey the fact that not all the data can be read into main storage before processing must begin. At the very least, the incoming data must be moved out to external storage where it can be called back and processed a piece at a time at leisure. In some cases, particularly on-line applications, the data must be partially ("quick look") or completely processed while input continues.

what single-level store?

Given the FS single level store with reasonable performance, the outputting to external storage appears to be completely solved for this application. The ability to handle infinitely long, high data rate input records remains a problem.

The maximum data rate for these applications is limited by

(a) The speed of analog to digital conversion.

(b) Serial transmission speed of digital data (telemetry)

(c) Computer throughput.

3.5.3 Continued

The current state of the art for analog to digital conversion is in the 200-500K samples per second range. This yeilds a data rate of 1 MB or less. Telemetry data rates are currently in the .1 to .5 MB range. It should be noted that many of these systems are required to process multiple data streams concurrently such that aggregate data rates up to 5 MB may be required.

The key to success in data streaming applications is not the ability to handle extremely high data rates. The key is the ability to handle data continuously at a fairly fast rate. The problem with S/360-370 I/O architecture for this application is that channel re-instruct takes data cycles. This means that the input data rate is limited by the necessity to allow time for a channel re-instruct between each data cycle. The reason for this is that once started, the input stream must continue at a constant rate. Therefore, even on the largest S/360-370 where command chaining takes only one data cycle, the data streaming rate must be set to only half the capability of the channel. Even though the speed capability is there, the I/O structure does not allow the application to use it.

No implementation is suggested here since these objectives can impact very basic structures in the FS architecture and there are many ways to do the trade-offs.

4.0 Architectural Considerations for Sensor Based and Other Response Oriented Systems

4.1 Introduction

This section is included as an on going forum for insights and observations on the subject of Sensor Based and Other Response Oriented Systems. As such, it relates to the nature and attributes of these systems in general, rather than the more specific items covered in the rest of this document. It is planned that this section will be added to over time. Contributions are solicited.

4.2 Sensor Based and Other Response Oriented Systems Considerations

4.2.1 The Effect of Overhead on Response Time

The simplest model for a system handling real time events is a single server handling random arrivals with random service times using a FIFO Queue discipline with no losses: denoted $M/M/1/\infty$

For this model, the mean response time (T_Q) is given by:

$$T_Q = \frac{T_S}{1-\rho}$$

where T_S = mean service time

and ρ = server utilization

To see the effect of overhead in the server (system) let us assume that the mean service time T_Q is all useful work and that there is a fixed amount of overhead added to each service time. Let the overhead be a percent of the mean service time:

Added overhead: $T_{OH} = rT_S, \quad 0 \leq r \leq 1$

The new mean service time is therefore:

$$T_S = T_S + T_{OH} = T_S + rT_S = T_S\ (1+r)$$

By definition the server utilization (ρ) is:

$\rho = \lambda T_S$, $\quad \lambda$ = # events per unit time

Then the new utilization ρ' is

$$\rho' = \lambda T'_S = \lambda T_S\ (1+r) = \rho(1+r)$$

Therefore, for the queue with added overhead the response time is given by:

$$T'_Q = \frac{T_S'}{1-\rho'} = T_S\ \frac{(1+r)}{1-\rho(1+r)}$$

which reduces to the original T_Q when r=0.

4.0 Architectural Considerations for Sensor Based and Other Response Oreented Systems

4.1 Introduction

This section is included as an on going forum for insights and obeervations on the subject of Sensor Based ond Other Repponse Oriented Systems. As such, it relates to the nature and attributes of these systems in general, rather than the more specific items covered in the rest of this document. It is planned that this section will be added to over time. Contirbutions are solicited.

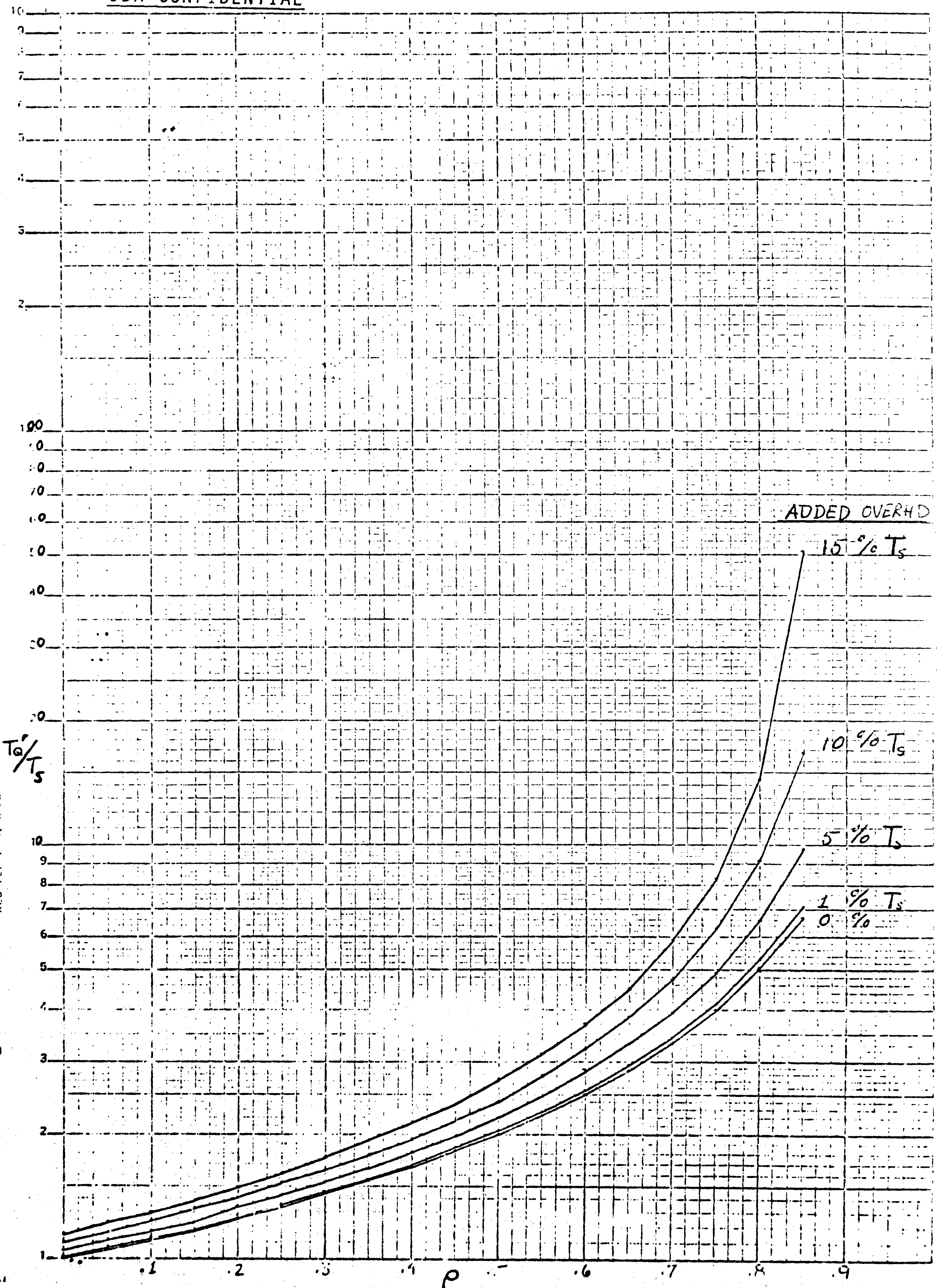
The response time is generally more useful if given in "service time units" which are independent of actual time units, thus,

$$\frac{T_Q'}{T_S} = \frac{1+r}{1 - \rho(1+r)}$$

We can now examine the effect on response time (T_Q') for this model, of different values of overhead (rTs). The plot shows the result for r=0, .01, .05, .10 and .15.

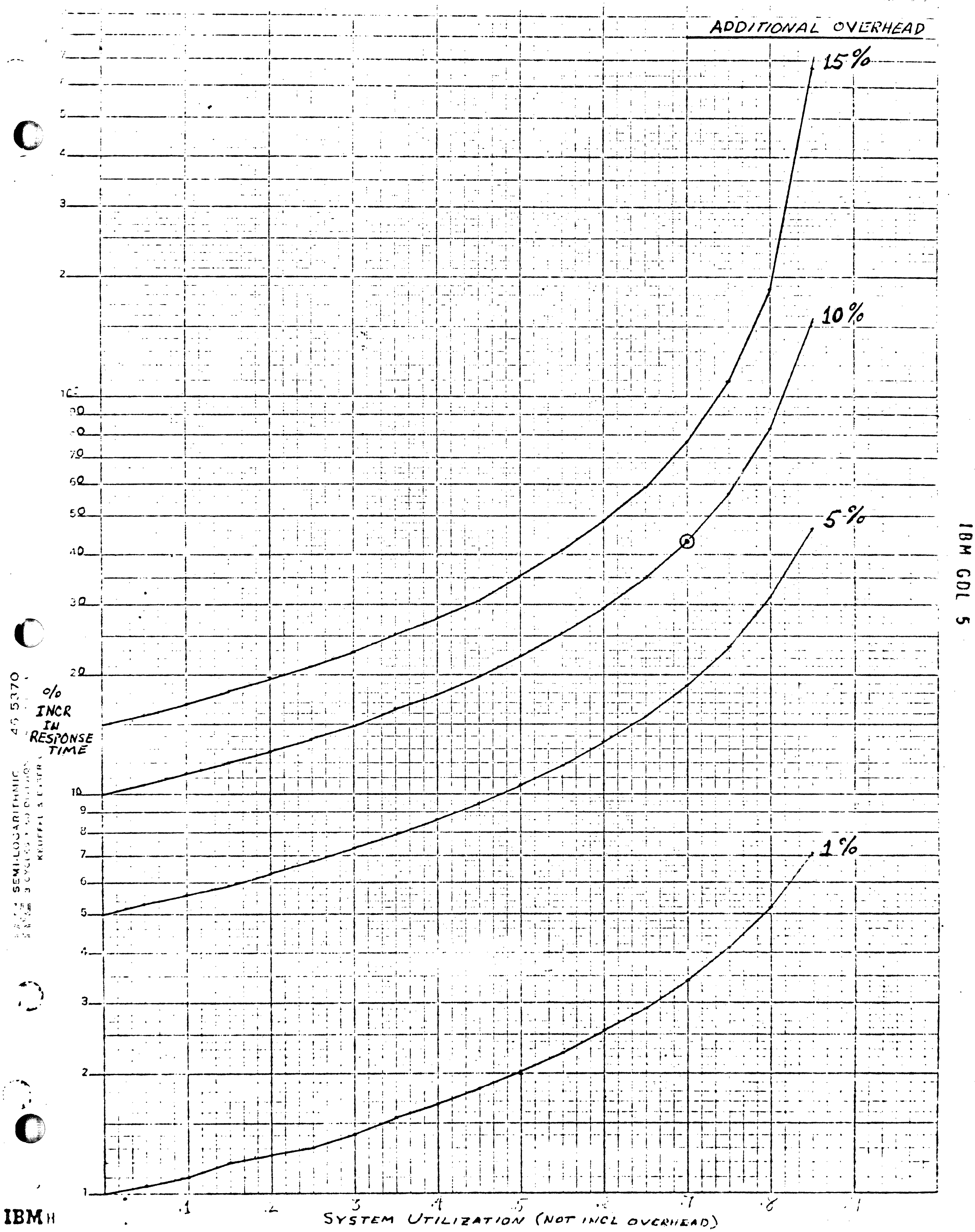
Note that at 70% utilization, 10% added overhead yields:

$$\frac{(T_Q')\quad 10\%}{(T_Q')\quad 0\%} = \frac{4.78}{3.33} \approx 1.43 \rightarrow 43\%$$ increase in response time


ADDED OVERHD
15 % T_s
10 % T_s
5 % T_s
1 % T_s
0 %
T_Q'/T_s
ρ
.1
.2
.3
.4
.6
.7
.8
.9

T_Q'/T_S

ρ	r=0	r=.01	r=.05	r=.10	r=.15
0.05	1.052631579	1.063717746	1.10817942	1.164021164	1.220159151
0.1	1.111111111	1.123470523	1.173184358	1.235955056	1.299435028
0.15	1.176470588	1.190335887	1.246290801	1.317365269	1.389728097
0.2	1.25	1.26566416	1.329113924	1.41025641	1.493506494
0.25	1.333333333	1.351170569	1.423728814	1.517241379	1.614035088
0.3	1.428571429	1.449067432	1.532846715	1.641791045	1.755725191
0.35	1.538461538	1.562258314	1.660079051	1.788617886	1.924686192
0.4	1.666666667	1.694630872	1.810344828	1.964285714	2.12962963
0.45	1.818181818	1.851512374	1.990521327	2.178217822	2.383419689
0.5	2	2.04040404	2.210526316	2.444444444	2.705882353
0.55	2.222222222	2.272215973	2.485207101	2.784810127	3.129251701
0.6	2.5	2.563451777	2.837837838	3.235294118	3.709677419
0.65	2.857142857	2.940320233	3.307086614	3.859649123	4.554455446
0.7	3.333333333	3.447098976	3.962264151	4.782608696	5.897435897
0.75	4	4.164948454	4.941176471	6.285714286	8.363636364
0.8	5	5.260416667	6.5625	9.166666667	14.375
0.85	6.666666667	7.137809187	9.76744186	16.92307692	51.1111111


ADDITIONAL OVERHEAD
15%
10%
5%
1%
% INCR IN RESPONSE TIME
SYSTEM UTILIZATION (NOT INCL OVERHEAD)
IBM GDL 5

RATIO

$T_Q' / (T_Q)_{r=0}$

ρ	$r=0$	$r=.01$	$r=.05$	$r=.10$	$r=.15$
0.05	1	1.010531859	1.052770449	1.105820106	1.159151194
0.1	1	1.011123471	1.055865922	1.112359551	1.169491525
0.15	1	1.011785504	1.059347181	1.119760479	1.181268882
0.2	1	1.012531328	1.063291139	1.128205128	1.194805195
0.25	1	1.013377926	1.06779661	1.137931034	1.210526316
0.3	1	1.014347202	1.072992701	1.149253731	1.229007634
0.35	1	1.015467904	1.079051383	1.162601626	1.251046025
0.4	1	1.016778523	1.086206897	1.178571429	1.277777778
0.45	1	1.018331806	1.09478673	1.198019802	1.310880829
0.5	1	1.02020202	1.105263158	1.222222222	1.352941176
0.55	1	1.022497188	1.118343195	1.253164557	1.408163265
0.6	1	1.025380711	1.135135135	1.294117647	1.483870968
0.65	1	1.029112082	1.157480315	1.350877193	1.594059406
0.7	1	1.034129693	1.188679245	1.434782609	1.769230769
0.75	1	1.041237113	1.235294118	1.571428571	2.090909091
0.8	1	1.052083333	1.3125	1.833333333	2.875
0.85	1	1.070671378	1.465116279	2.538461538	7.666666667

4.2.2 Pre-emptive - Resume Priority Queueing

Pre-emptive resume priority queueing is the discipline used in the OS task dispatcher and in what is commonly called "Priority Interrupt" (PI) when implemented in hardware for system control. There are two reasons that make pre-emptive resume priority queueing attractive in response oriented systems (and indeed in computer systems in general).

1. faster response time for important events
2. better throughput for events with short service times

The first of these two is the one that is usually thought of by the sensor base user. The second has become increasingly important to those concerned with task scheduling and dispatching algorithms.

In the terminology of Queueing Theory, not only is a high priority customer placed ahead of all lower priority waiting customers in a queue, but if a lower priority customer is currently being serviced, he is preempted and service is immediately given the new arrival. This means that if levels of priority can be assigned to events within a system, the system can be assured that it is working at the most important task at all times.

It is an extreme simplification to view any computer system as being a single server with a single queue. In reality any system is a network of interconnected hardware and software queues, some with multiple servers. This makes these systems very difficult to analyze and so to date, though progress is being made, we have very little detailed understanding of system operation.

Simple systems are easily analyzed, however. What follows is an effort to gain insights into the complex problem by use of what the simplest preemptive-resume priority queueing model.

The model consists of a single, lossless queue and single server. Random arrivals and negative exponetial service times are assumed.

If there are m levels of preemptive-resume priority, then the mean queueing time for priority level j (TQj) is given by:

$$T_{Qj} = \frac{1}{1-u_{j-1}} \left\{ b_{1j} + \frac{\sum_{i=1}^{j} \lambda_i b_{2i}}{2(1-u_j)} \right\} ; \quad u_0 = 0$$

(Note: j = 1, is the highest priority level)
where:

b_{1j} = 1st moment of service time for level j

b_{2j} = 2nd moment of service time for level j

For negative exponetial service times,

$b_{1j} = T_{Sj}$

$b_{2j} = 2T^{2}_{sj}\,2$

The fraction of server utilization due to traffic on the jth priority level is given by:

$$\rho_j = \lambda_j \times b_{1j}$$

The cumulative utilization due to traffic on the highest j levels is designated uj, where:

$$u_j = \sum_{i=1}^{j} \lambda_i b_{1i} \quad ; \quad uo = 0$$

Results using this model were obtained for 1-32 priority levels. A one level preemptive-resume queue is the same as the FIFO queue studied earlier.

The workload chosen for study was .6 total utilization which is neither high nor low for such a system. The total workload was distributed among the priority levels in three different ways as shown in Figure 1.

Case 1: the total workload is distributed uniformly with an equal amount handled by each of the available priority levels:

$$\rho_i = \rho_{TOT} \frac{1}{m} = \text{constant}$$

Case 2: The total workload is distributed linearly with the smallest utilization on the highest priority level:

$$\rho_i = \rho_{TOT} \frac{2i}{m(m+1)}$$

Case 3: The total workload is distributed exponentially with the smallest utilization on the highest priority level:

$$\rho_i = \rho_{TOT} \frac{2^{i-1}}{2^m - 1}$$

The results for the mean response time by level for 1-32 levels are shown in Figures 2, 3, 4 for cases 1, 2, 3 respectively. Response times are in mean service time units.

Discussion:

This study is not yet complete so discussion will be deferred until that time.

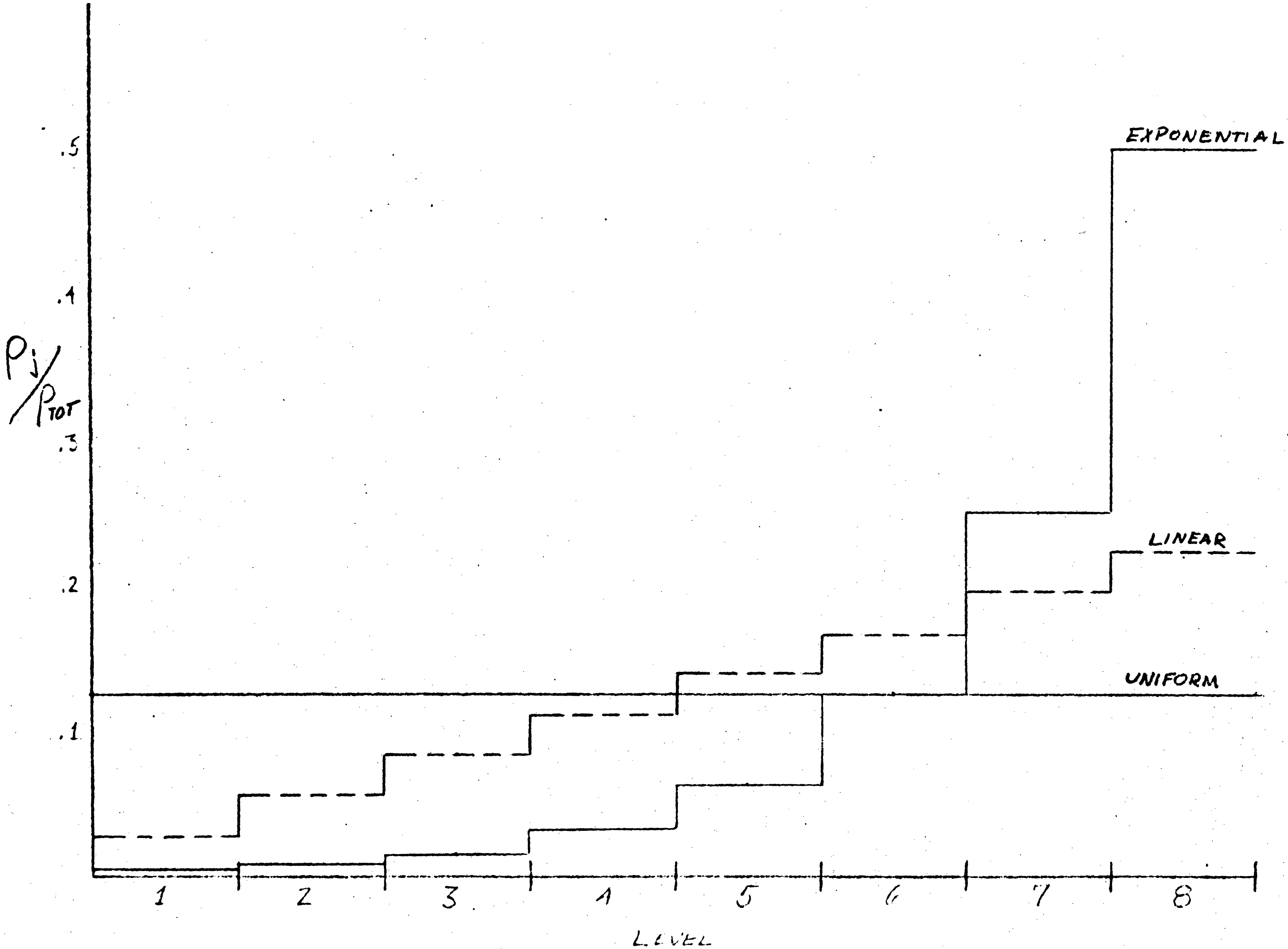


FIGURE 1

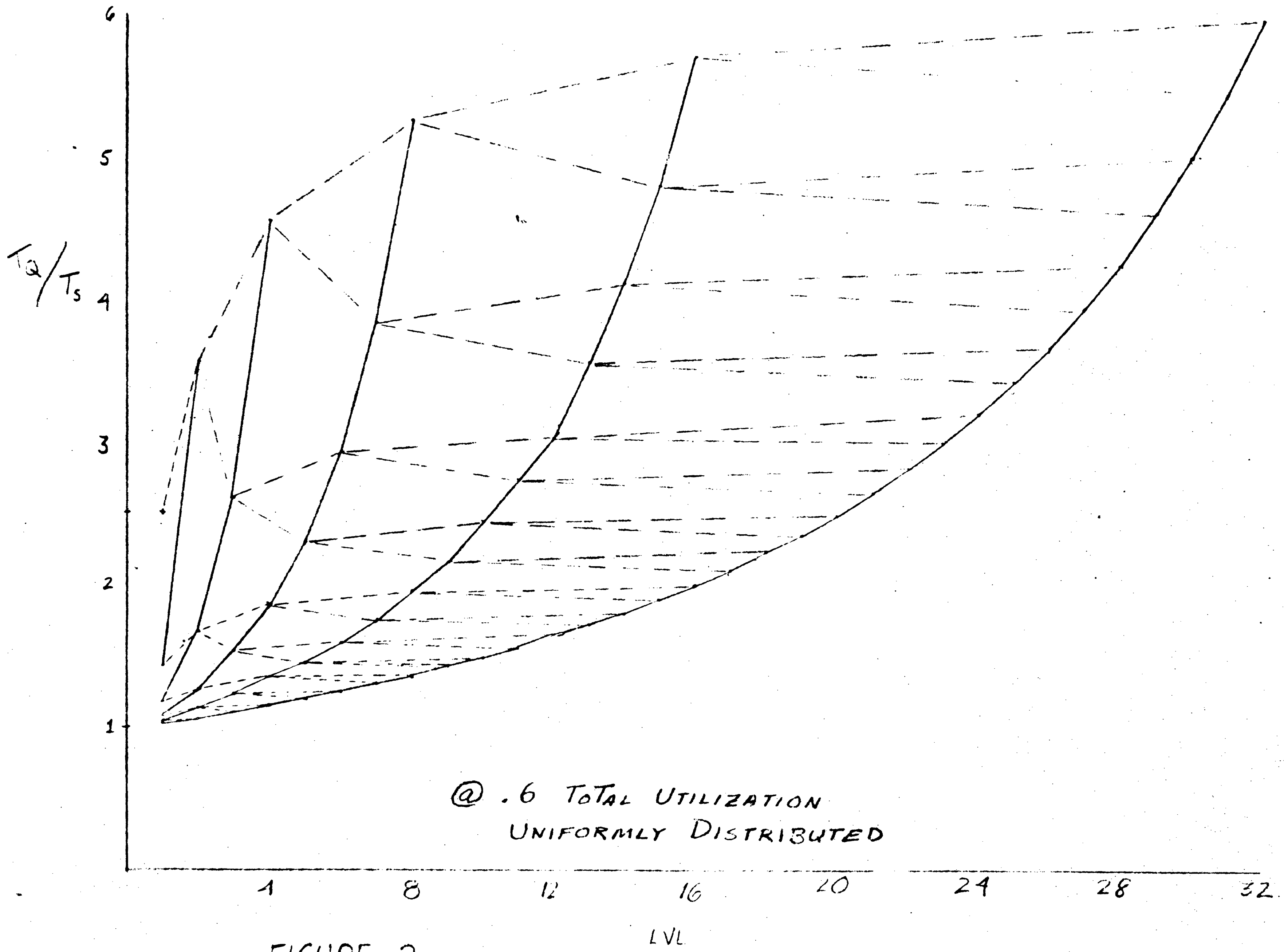

6
5
4
3
2
1
TQ/Ts
@ .6 TOTAL UTILIZATION
UNIFORMLY DISTRIBUTED
4
8
12
16
20
24
28
32
LVL

FIGURE 2

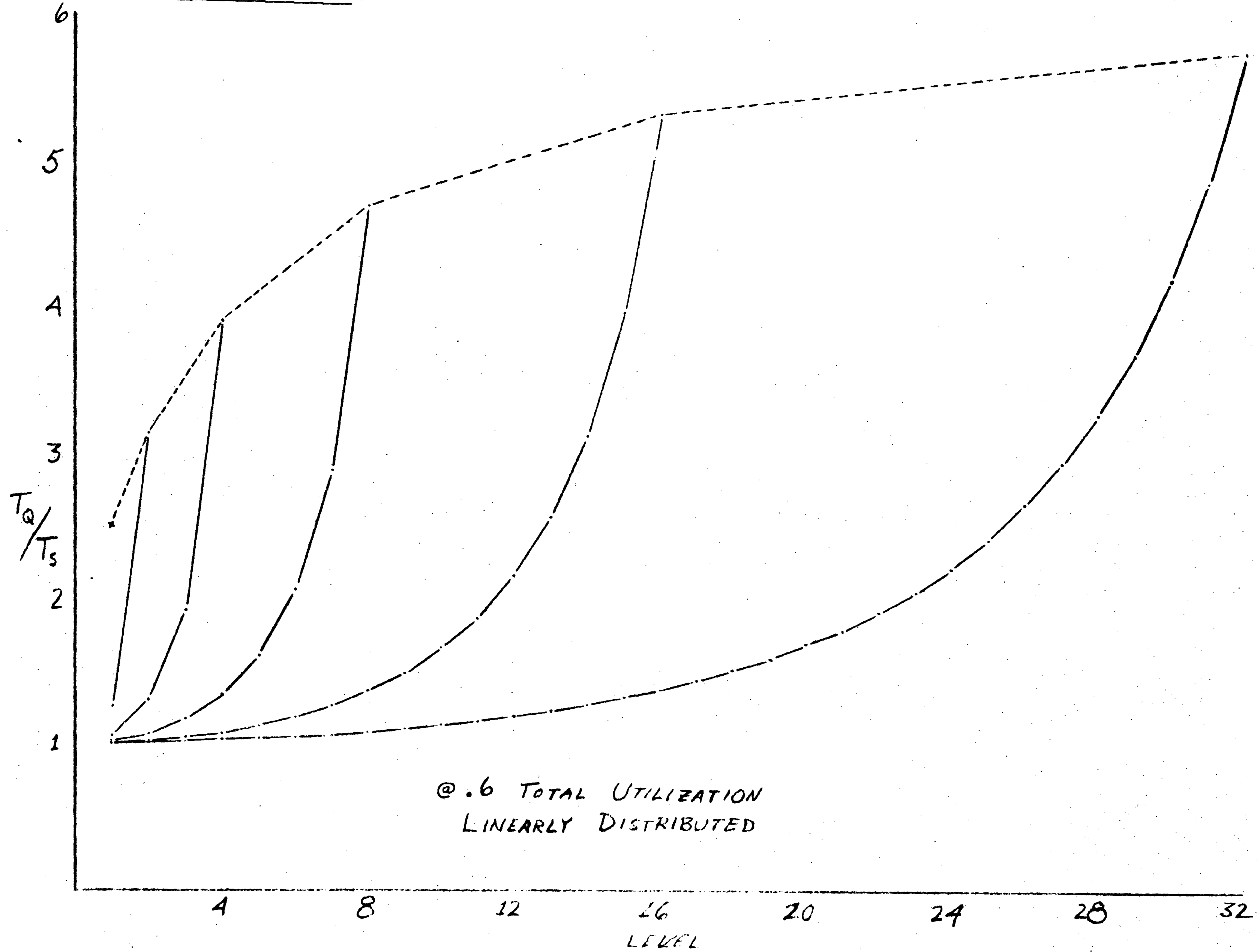
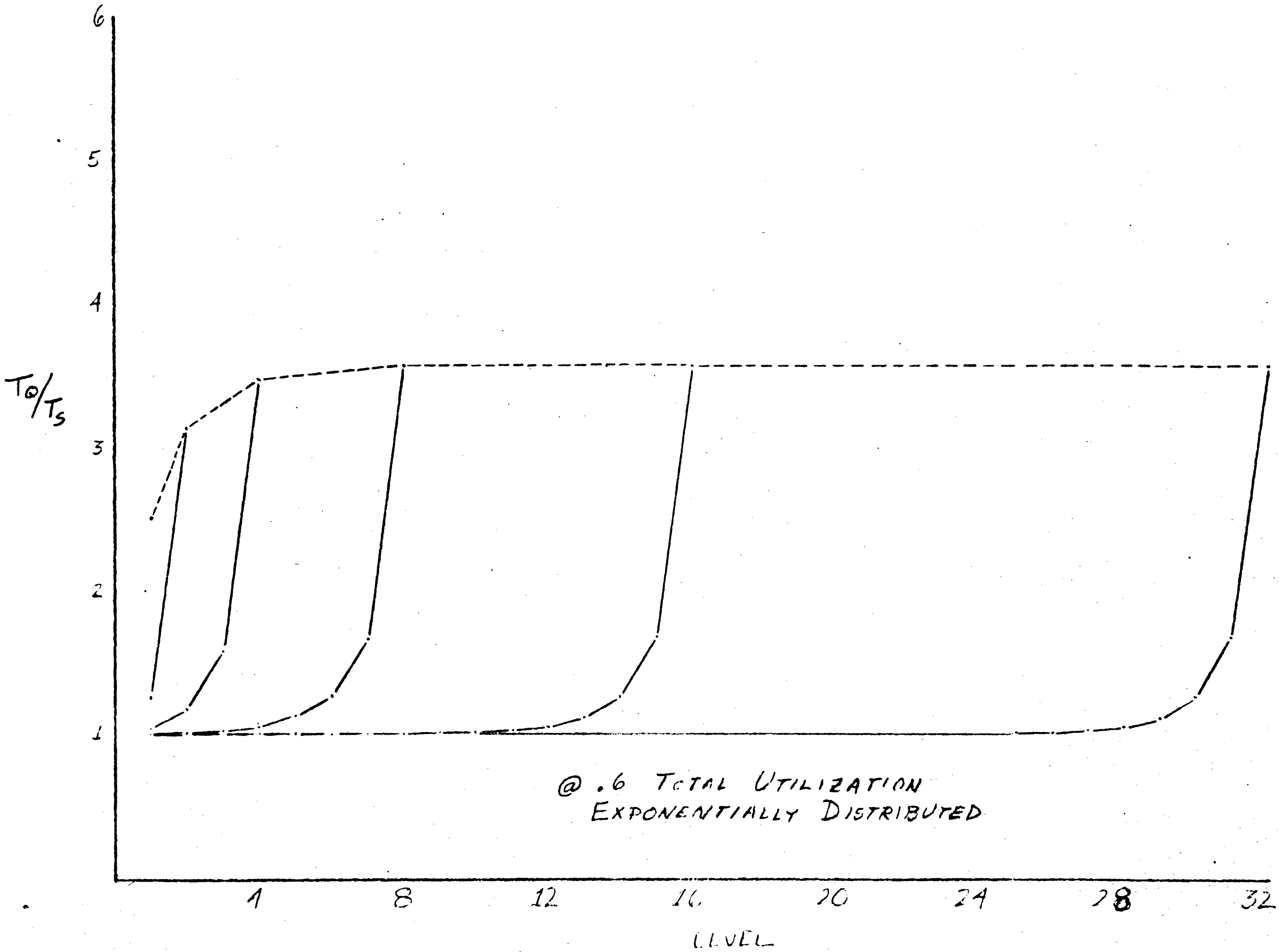

6
5
4
3
T_Q / T_S
2
1
@.6 TOTAL UTILIZATION
LINEARLY DISTRIBUTED
4
8
12
16
20
24
28
32
LEVEL

FIGURE 3



6
5
4
3
2
1
To/Ts
@ .6 Total Utilization
Exponentially Distributed
4
8
12
16
20
24
28
32
LEVEL

FIGURE 4

4.3 Comments on the Current FS Definition

4.3.1 Networks/Routing

Do not buffer entire message before retransmission

- extra delays will be introduced, especially if retransmission is done several times
- such structure makes the support for naturally long records (programs) difficult

4.3.2 Support for hierarchical computer systems

. enhances the customer's view of RAS. A customer does not want his plant to depend on one box or the boxes in one room.

4.3.3 Plant Data Communications

. a common interface and transmission scheme for all in plant data communications enhances IBM and customer flexibility. This is a "granularity" item.

. low speed communications should be built around line sharing techniques wherever feasible.

. need a single high performance subsystem to support attachment to FS of:

- -terminals
- -data collection
- -sensor base subsystems
- -high performance direct sensor devices

4.3.4 Task Switch Time

. more important than S/370

- -FS is a response oriented system
- -many more task switches per unit time
  - *faster processing = less time to next I/O
  - *many more on line, interactive users than 370
  - *increased parallelism possible with architecture
  - *event driven system

4.3.5 Batch Performance

. highly responsive systems require low disable time and low overhead. Both these items improve batch performance

4.3.6 TP Performance

. sensor base requirements are more stringent than TP, therefore, a structure that is adequate for sensor base will be a swinger for TP.

4.3.7 Sensor Based Data

. future trend in sensor base is to much more digital data transmission
. analog sensors will ultimately disappear.

4.3.8 Plant Floor Displays

. need automatic backup for plant floor logging devices
. alternate device support for inquery and alarm display devices
. automatic save and optional retransmit for messages to Not Ready devices (e.g., remote logging printer runs out of paper)

4.3.9 Source/Sink Architecture

. if all S/S instructions are syschronous, then there must be provision for control of complex logical devices
- an I/O operation that may take a long time to complete (secs to hrs)
- a complex logical I/O operation that may require control and data transmission to more than one physical piece of hardware or path.

4.3.10 RPQ's

. systems supporting sensor based devices or sub-systems must have the capability to give full support to RPQ's on these devices.

- one or more RPQ's are installed on

  *83% of IBM 1800's
  *73% of IBM S/7's

- RPQ's are a way of life for SB systems - a viable design must make provision for their support.

APPENDICES

Available on request:

Dick Guyette
252+3434

IBM CONFIDENTIAL