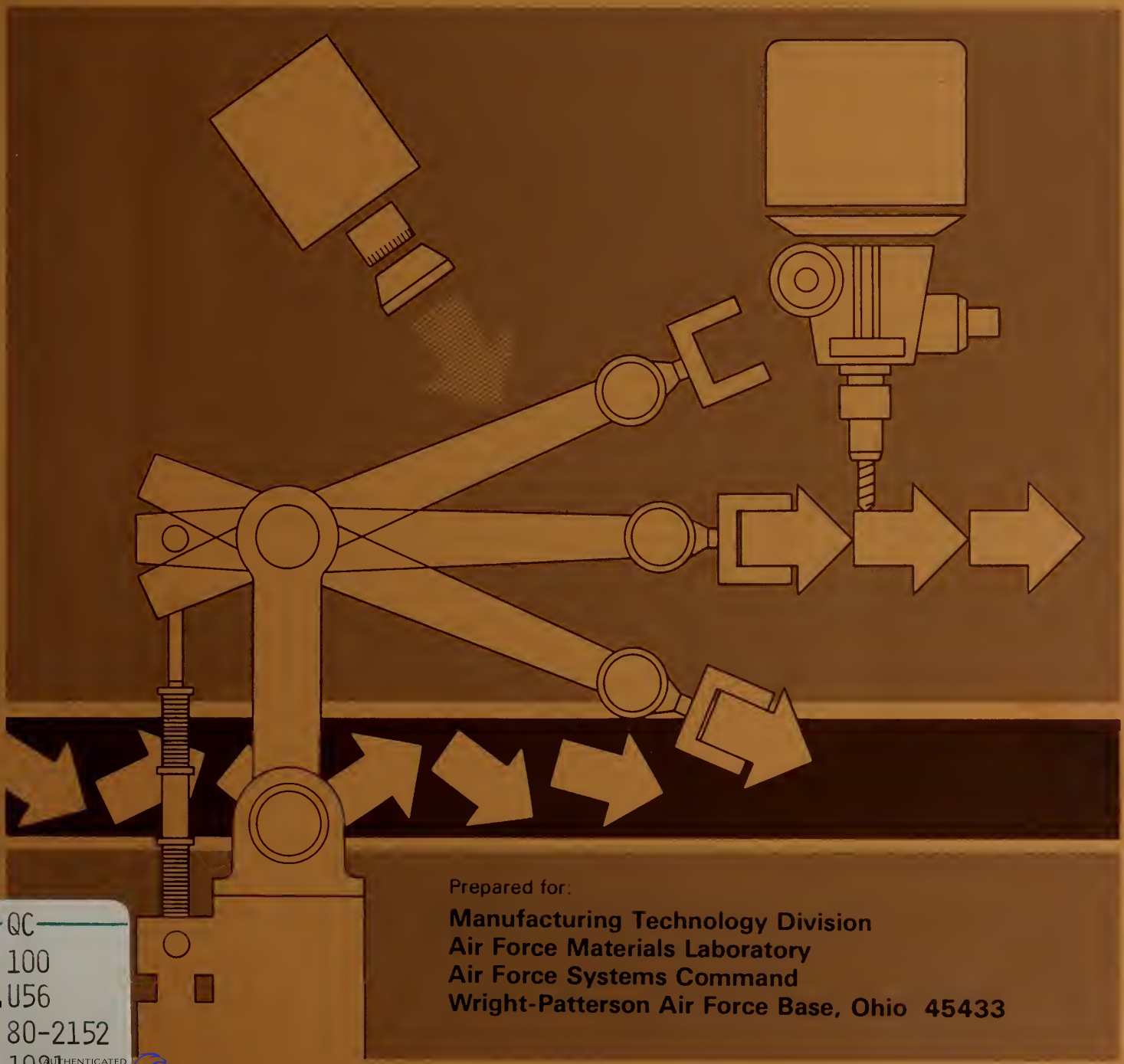
NAT'L INST. OF STAND & TECH R.I.C.
A11105 354200
IBS
ubli-
ations
REFERENCE
NIST
PUBLICATIONS

U.S. Department
of Commerce
National Bureau
of Standards

Proceedings of NBS/Air Force ICAM Workshop on Robot Interfaces

January 1980

NBSIR 80-2152



Prepared for:
Manufacturing Technology Division
Air Force Materials Laboratory
Air Force Systems Command
Wright-Patterson Air Force Base, Ohio 45433

QC
100
.U56
80-2152
198
AUTHENTICATED
U.S. GOVERNMENT
INFORMATION
GPO

National Bureau of Standards
Library, E-01 Admin. Bldg.
FEB 27 1981
Not circ. - Ref.
QC100
.U56
NO. 80-2152
1981

NBSIR 80-2152

PROCEEDINGS OF NBS/AIR FORCE ICAM WORKSHOP ON ROBOT INTERFACES

January 1981

Prepared for:
Manufacturing Technology Division
Air Force Materials Laboratory
Air Force Systems Command
Wright-Patterson Air Force Base, Ohio 45433

U.S. DEPARTMENT OF COMMERCE, Philip M. Klutznick, *Secretary*

Jordan J. Baruch, *Assistant Secretary for Productivity, Technology, and Innovation*

NATIONAL BUREAU OF STANDARDS, Ernest Ambler, *Director*

NOTICE

When Government drawings, specifications, or other data are used for any purpose other than in connection with a definitely related Government procurement operation, the United States Government thereby incurs no responsibility nor any obligation whatsoever; and the fact that the government may have formulated, furnished, or in any way supplied the said drawings, specifications, or other data, is not to be regarded by implication or otherwise as in any manner licensing the holder or any other person or corporation, or conveying any rights or permission to manufacture, use or sell any patented invention that may in any way be related thereto.

Copies of this report should not be returned unless return is required by security considerations, contractual obligations, or notice on a specific document.

For Early Domestic Dissemination

Because of its significant early commercial potential, this information, which has been developed under a U.S. Government program, is being disseminated within the United States in advance of general publication. This information may be duplicated and used by the recipient with the expressed limitations that it not be published nor released to foreign parties without appropriate licenses. Release of this information to other domestic parties by the recipient shall be made subject to these limitations. This legend shall be marked on any reproduction of this data in whole or part. These limitations shall be considered void two years after the release date on the data.

ROBOTICS SUPPORT PROJECT
FOR THE AIR FORCE ICAM PROGRAM

Proceedings of the NBS/Air Force ICAM Workshop on Robot Interfaces

June 4-6, 1980

Editors
Thomas E. Wheatley
James S. Albus
Roger N. Nagel

National Bureau of Standards
National Engineering Laboratory, CDPT
Industrial Systems Division
Programmable Automation
Washington, DC 20234

Prepared under
MIPR SY1455-78-00003

This is to certify that this report was prepared by U.S. Government employees as part of their official duties and is therefore a work of the U.S. Government and not subject to copyright.

TABLE OF CONTENTS

Page

ABSTRACT 1

WORKSHOP ATTENDEES 2

INTRODUCTION 7

WORKSHOP REPORTS 9

SIMPLE SENSOR INTERFACE 9

WRIST INTERFACE 14

COMMON ROBOT CONTROL INTERFACE 18

COMPLEX SENSOR INTERFACE 34

FUTURE GUIDELINES TOWARDS INTERFACES 40

GENERAL RECOMMENDATIONS AND CONCLUSIONS 46

APPENDIX I 47

ABSTRACT

This paper describes the proceedings of a workshop on robot interfaces held at the National Bureau of Standards on June 4-6, 1980. Five possible areas for standardization of interfaces were discussed: the Simple Sensor interface between simple peripheral devices and a robot control system; the Wrist Interface, between the robot wrist and the end effector; the Complex Sensor Interface that covers vision, complex touch, and other such sensors; the Common Robot Control interface, providing robot independent trajectory descriptions; and Future Guidelines towards Interfaces, covering data base, offline programming, and system integration interfaces. The goal was to define the areas ready for current standards, and those for which standards would be considered an impediment to developing technologies.

This workshop was jointly sponsored by NBS and the Air Force ICAM project under MIPR SY1457-00003 "Robotics Support Project for the Air Force ICAM Program".

WORKSHOP ATTENDEES

James S. Albus
Programmable Automation
National Bureau of Standards
Bldg. 220, Rm. A123
Washington, DC 20234
(301) 921-2381

Carl V. Ames
General Dynamics
P. O. Box 748
Ft. Worth, Texas 76101
(817) 732-4811 X5475

Henry S. Baird
RCA Laboratories
Room N201, US Route 1
Princeton, NJ 08540
(609) 734-3150

Anthony J. Barbera
Programmable Automation
National Bureau of Standards
Bldg. 220, Rm. A123
Washington, DC 20234
(301) 921-2381

Richard Becker
Cheeseborough-Pond's, Inc.
John Street, Potts Bldg.
Clinton, CT 06413
(203) 669-8601, x557

Sidney Berkowitz
David W. Taylor Naval Ship
R&D Center, Code 1824
Bethesda, MD 20084
(202) 227-1889

Edwin R. Bowerman
GTE Laboratories
40 Sylvan Rd.
Waltham, MA 02154
(617) 890-8460

William S. Brower
Control Data Corporation
124 Hebron Avenue
Glastonbury, CT 06033
(203) 633-6781

Donald O. Caughman
Lockheed-Georgia Co.
D47-11 Z80
86 S. Cobb Dr.
Marietta, GA 30063
(404) 424-3859

Hans G. Colleen
ASEA Inc.
4 New King Street
White Plains, NY 10604
(914) 428-6000 X345

Bryan L. Dawson
Cincinnati Milacron
4701 Marburg Avenue
Cincinatti, Ohio 45209
(513) 841-8767

John D. Dexheimer
General Electric
1 River Road, Bldg. 10A-115
Schenectady, NY 12345
(518) 385-0689

Maurice J. Dunne
Unimation Inc.
Shelter Rock Lane
Danbury, CT 06810
(203) 744-1800 X305

G. E. Ennis
McDonnell Douglas Corportion
P. O. Box 516
St. Louis, MO 63166
(314) 232-4021

Bruce D. Ernst
Prab Conveyors, Inc.
5944 E. Kilgore Rd.
Kalamazoo, MI 49003
(616) 349-8761

Patrick E. Hacker
Boeing Commercial Airplane Co.
M/S 9R-38, P. O. Box 3707
Seattle, WA 98124
(206) 237-5520

Richard J. Handwerb
General Dynamics
Box 748, MZ 6459
Ft. Worth, TX 76101
(817) 732-4811 X5475

T. Kasvand
National Research Council
Bldg. M50
Ottawa, Ontario, Canada
(613) 993-2003

Arkady G. Makhlin
Westinghouse R&D Center
1310 Beulah Rd.
Pittsburgh, PA 15235
(412) 256-3982

Gordon Mayer
U.S.A.F. (ICAM)
AFWAL/MLTC
Wright Patterson Air Force Base
Ohio 45433
(513) 255-2562

V. W. McClure
Texas Instruments Inc.
P. O. Box 225621, M/S 452
Dallas, TX 75265
(214) 238-3479

Lori Mei
Society of Manufacturing Engineers
P. O. Box 930
1 SME Drive
Dearborn, MI 48128
(313) 271-1500 x407

Roger N. Nagel
Programmable Automation
National Bureau of Standards
Bldg. 220, Rm. A123
Washington, DC 20234
(301) 921-2381

Wallace J. Plumley
Lockheed-Georgia Co.
D47-11 Z80
86 South Cobb Dr.
Marietta, GA 30063
(404) 424-3796

Louis E. Potter
RCA Corporation
New Holland Ave.
Lancaster, PA 17604
(717) 397-7661 X2208

Donald Seltzer
C. S. Draper Laboratory, Inc.
555 Tech. Square
Cambridge, MA 02139
(617) 258-1368

Joseph Sheehan
David W. Taylor Naval Ship
R&D Center
Bethesda, MD 20084
(202) 227-1302

Bradford M. Smith
Control Software
National Bureau of Standards
Bldg. 220, Rm. A123
Washington, DC 20234
(301) 921-2381

Albert C. Snowdon
EATON
Operations & Technical Center
4201 N. 27th Street
Milwaukee, WI
(414) 442-7800 X2985

Charles Spalding
Unimation Inc.
188 S. Whisman Rd.
Mountain View, CA 94041
(415) 965-0557

Gordon J. VanderBrug
Automatix Inc.
217 Middlesex Turnpike
Burlington, MA 01803
(617) 273-4340

John Vranish
Naval Surface Weapons Center
Code E-1A, Rm. 4-174
White Oak
Silver Spring, MD 20910
(202) 394-3256

Jerome F. Walker
Nordson Corporation
555 Jackson Street
Amherst, Ohio 44001
(216) 988-9411

Thomas E. Wheatley
Programmable Automation
National Bureau of Standards
Bldg. 220, Rm. A123
Washington, DC 20234
(301) 921-2381

Jack Wiley
Planning Office
National Bureau of Standards
Bldg. 101, Rm. A929
Washington, DC 20234

INTRODUCTION

On June 4 - 6, 1980, a workshop on the need for standardization of robot interfaces was held at the National Bureau of Standards (NBS). Co-sponsored by the Air Force ICAM office and the NBS Programmable Automation group, the workshop was co-chaired by Dr. James Albus and Dr. Roger Nagel of the NBS. Mr. Gordon Mayer of the Air Force ICAM presented an over-view of the Air Force program in robotics both present and future. There were 35 attendees representing industry, academia, and the government. During the workshop, the attendees were assigned to one or more of five separate working groups, each of which met on two occasions. These five areas were: the Simple Sensor interface between simple peripheral devices and a robot control system; the Wrist Interface, between the robot wrist and the end effector; the Complex Sensor Interface that covers vision, complex touch, and other such sensors; the Common Robot Control interface, providing robot independent trajectory descriptions; and Future Guidelines towards Interfaces, covering data base, offline programming, and system integration interfaces. There were general sessions held each day both to charge the separate working groups with tasks and to hear reports on their results so that the workshop could truly function in an interactive working fashion.

On the last day of the workshop, a general session was held. At that session it was determined that with respect to the Simple Sensor interface, connecting robots with other devices in their environment, the time has come when standards ought to be created. Similarly, with respect to the Wrist interface, the connection between the robot wrist and its end effector, we are also ready for a standard. In both of these areas reports were written recommending the beginning of a standards effort.

With respect to the Complex Sensor and Common Robot Control interfaces, it was determined that it is still too early to begin a standards effort. However, position papers arguing for particular directions and general awareness of the community on these topic areas have been written and are included in this report. These position papers are designed to focus attention on the issues and suggest paths as to how they may be addressed.

In the fifth topic area, Future Guidelines towards Interfaces, the two main themes covered were robot programming languages and the integration of robots into robotic systems for automated manufacturing. While the discussions were

interesting and instructive, it was clear that it is much too early in the development of these areas to offer standards. The general feeling was that the field is beginning to accelerate and that it would be worth re-investigating these topics as well as some of the other topics covered in the workshop in approximately one year. Accordingly, the Bureau of Standards and the Air Force ICAM office are looking into the possibility of a second conference to take place next June.

WORKSHOP REPORTS

During the introductory workshop session, guidelines were drawn up for a format for each workshop session to follow. Various topics and questions were to be presented to the group members, and answered as best befits that interface. The topics decided upon for discussion were:

A. Scope and Definition of Interface
B. Is There a Need for a Standard?
C. Standardization
D. Timing of Effort
E. Recommendations

Following these initial discussions, the overall group broke out into the individual working groups. The session chairpersons were charged with reporting the activities of their particular group with respect to these questions, and to draw up a position paper on their conclusions and recommendations at the close of the workshop. Their reports are below in the following order: the Simple Sensor interface; the Wrist interface; the Common Robot Control interface; the Complex Sensor interface; and the Future Guidelines towards Interfaces report.

SIMPLE SENSOR INTERFACE

I. GROUP MEMBERS

Bryan Dawson *	-	Cincinnati- Milacron
Richard Becker	-	Cheeseborough - Ponds, Inc.
Edwin Bowerman	-	GTE Laboratories
Donald Caughman	-	Lockheed - Georgia Company
John Dexheimer	-	General Electric
Bruce Ernst	-	Prab Conveyors, Inc.
Patrick Hacker	-	Boeing
Ward McLure	-	Texas Instruments

* Chairperson

II. INTRODUCTION

In the general introduction to the workshop, it was stated that this type of interface was probably one of those most

ready for standardization. This group session was, therefore, charged with addressing certain specific issues with regard to the possible standardization of robot-to-peripheral device interfaces and to offer guidelines towards topics to be addressed in such a standard.

III. DEFINITION OF A PERIPHERAL DEVICE

A peripheral device in this context was defined as a piece of equipment or device used in conjuction with an industrial robot in the implementation of an application, and which communicates with the robot by means of binary signals only. To set the record straight, it does not include what are referred to as computer peripheral devices such as floppy discs, printers, and other such devices.

The definition should also not be construed to include higher frequency binary signals which constitute a data format, nor analog signals. However, many of the areas discussed by this group could also apply to such interfaces. Examples of peripheral devices would be machine tools, spot weld equipment, conveyors, etc.

IV. IS THERE A NEED FOR A STANDARD?

The consensus of the group was "yes" and the time for implementation is just right or may be even too late. The following reasons for this conclusion were offered:

A. Many robot users require the interfacing of robots to older, existing in-plant equipment. It is not until the actual interfacing takes place that snags, such as the following, are found:

1. Lack of interface points on old equipment.

2. Lack of control devices such as limit switches on existing machines.

3. Lack of complete cycle control on existing equipment, e.g., no way of initiating an automatic cycle nor of generating a cycle complete signal.

In other words, up until just recently, machines and equipment have not been built with robots in mind.

B. To offset these problems, there is a need to aid the user, robot manufacturer, and peripheral equipment manufacturer by providing guidelines to indicate the requirements for adequate marriage of the equipment.

C. It was felt that although new peripheral equipment is being manufactured to be used with robots, there is still a need to standardize logic levels of signals.

D. Although new equipment is being made suitable for use with robots, it is envisaged that for at least the next 20 years, robots will still be required to interface with the "old" equipment.

V. RECOMMENDATIONS

The group decided to limit the scope of guidelines to include the following subjects: signal level and type; noise and means of elimination; connector configurations; minimum interlock signal requirements; and incorporation of existing standards.

A. Signal Level and Type

Existing equipment and plants often use 110 VAC relay logic type control. Naturally, it is usually not possible to change these. Therefore, a 110 VAC interface appears to be necessary. New equipment (and programmable controllers) tend towards the use of lower voltage solid state logic; therefore a 24 VDC interface also appears to be necessary. It was felt that for this type of interface, levels such as the 5 VDC were too low and included possible noise and line attenuation problems. Thus (in conjunction with common practice) the two logic levels, 24 VDC and 110 VAC, were put forth as guidelines.

B. Noise and means of elimination

Precautions must be taken to eliminate the effects of noise, voltage transients, contact bounce, etc. The following points were indicated as guidelines to eliminating these problems.

1. Adequate grounding to eliminate ground loops.

2. Suitable shielding, e.g., basketweave, twisted shielded pair signal lines, etc.

3. Eliminating contact bounce and similar effects in devices such as relays and limit switches by suitable damping or by time delay concepts in robot controls.

4. Devices need to be isolated, e.g., by optical coupling.

5. Noise from switching devices can be eliminated by

suitable arc suppression.

6. Differential communication, i.e., floating zero potential on one signal line (not tied to ground).

7. Control power should be from isolated utilities.

8. Solid state devices may require such things as load resistors to ensure proper signal recognition.

C. Connector Configurations

Field wiring to terminal strip interface boxes was recommended. The interface terminal boxes must be mounted to the robot control and peripheral equipment such that the terminal strip access is convenient. The field wiring should be done in hard conduit for noise rejection. Use of individual pairs of wires from the robot to different parts of peripheral equipment was not recommended.

Also recommended was the use of pre-assembled flexible conduit and connectors for connections to and from junction boxes. Such connectors in common use are Pyle National, Amphenol, Cannon, etc., and are usually user specified.

A combination of the above may be used. If a combination of 110 VAC and 24 VDC is used, the signal lines must be adequately separated.

D. Minimum Interface Requirements

A minimum interlock requirement is probably necessary. The attached diagram (Figure 1) shows such interface signals. However, the signals and their functions are very application-oriented and each application must be examined carefully.

E. Existing Standards

Wherever possible, existing standards should be used, if not in their entirety, at least as guidelines. For instance, some robot manufacturers and users have already formulated their own standards. Programmable controller users and manufacturers similarly may have their own common practices. A standards search for such information is recommended.

VI. GENERAL COMMENTS AND CONCLUSIONS

It was recognized that in many cases effective use of programmable controllers would bridge the gap between "old" equipment and robots, and would also allow "add on" cycle

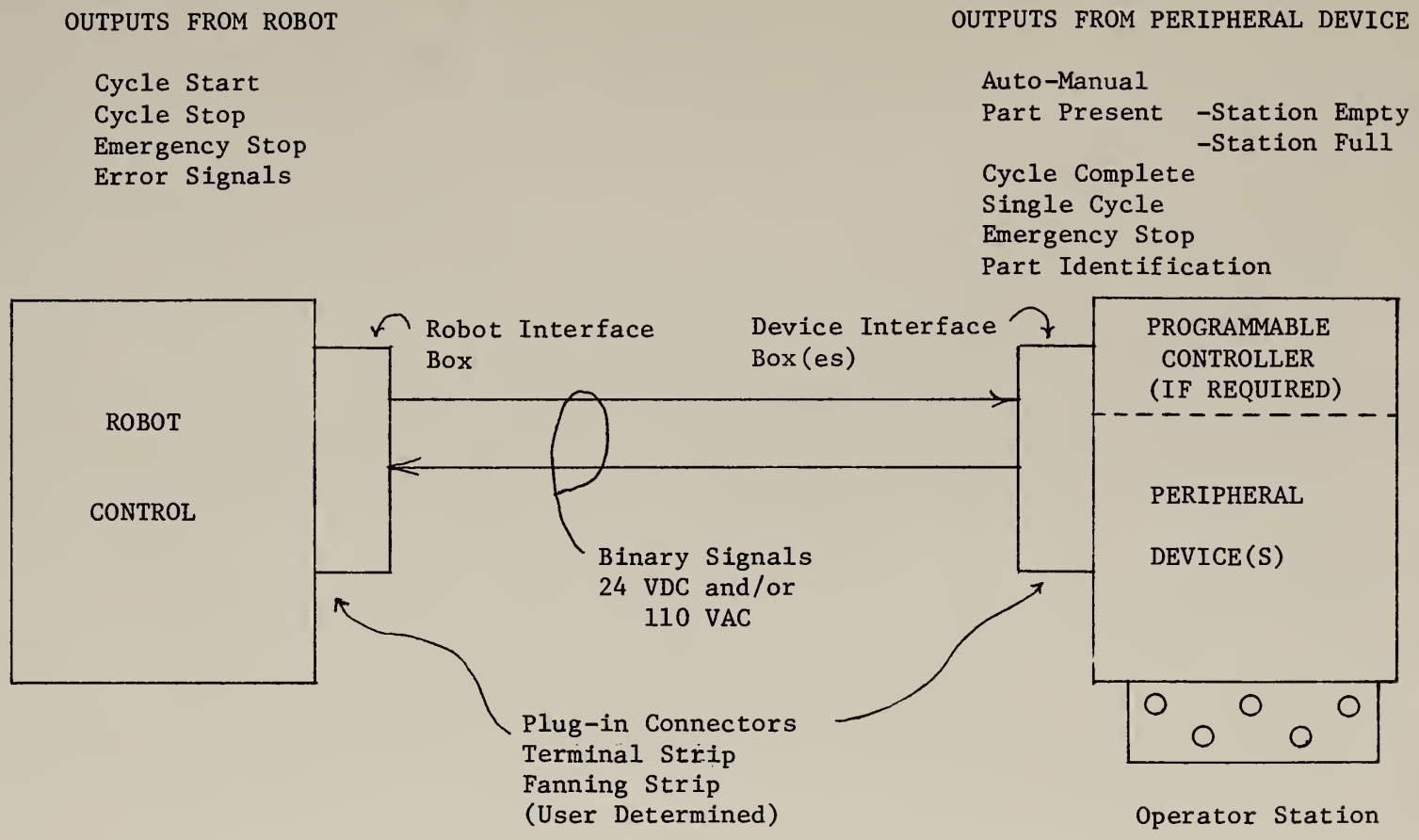


FIGURE 1

Minimum Robot/Peripheral Device
Interface Guidelines

control to such equipment. However, the guidelines discussed above still apply. Many of these guidelines could also apply to the other interfaces discussed in the workshop as well as to analog signals between equipment.

WRIST INTERFACE

I. GROUP MEMBERS

Dunne, Maurice*	-	Unimation
Smith, Bradford M.*	-	National Bureau of Standards
Ennis, Jerry	-	McDonnell Douglas
Ernst, Bruce	-	Prab Conveyors, Inc.
Handwerg, R. J.	-	General Dynamics
Potter, Louis	-	RCA
Vranish, John	-	Naval Surface Weapons

*Co-Chairperson

II. INTRODUCTION

Robots have typically been applied in dedicated, high-volume tasks where only a single gripper or end effector has been required. The end effector is generally a unique design and is specifically built for the chosen task by either the manufacturer or the end user. If subsequent end effectors are needed, the user is often hard put to find sources with reasonable leadtimes and costs. Manufacturers are necessarily more interested in expanding their sales of similar robots than they are in taking on more design and fabrication of different end effectors. Other design engineering firms are unable to develop cost-effective, generic end effectors since mounting techniques vary across different robot models and the volume of any one configuration is fairly small.

Present mounting schemes allow for manual replacement of an end effector and actuation of its operating mechanism. These schemes generally involve a bolt-on technique with facilities for hydraulic, pneumatic, and/or electrical signals to and from the end effector. One problem sometimes encountered with these designs is that an end effector cannot be replaced in exactly the same position and orientation that it had originally. This forces a user to slightly reprogram his robot every time an end effector such as a gripper or spot weld gun is taken off for maintenance. Another problem seen by users of a variety of different robots is that end effectors are not exchangeable across like capability manipulators. Often a spare unit must be stocked for each different production robot.

Recent trends have seen robots used for a greater variety of small-volume workloads. Here, end effectors are often changed to meet new job requirements or to make use of a variety of tools for the same job. Increased emphasis is

placed on the interface between the robot wrist and the end effector. Designs are required whereby a robot can change its own tools on command. Several implementations are known to exist with at least one in the public domain.

III. NEED

The Group concluded early in their discussion that it was timely to initiate a standardization activity to address the interface between an industrial robot and the variety of end effectors. Areas to be covered include mechanical registration; fastening; and facilities for air, fluid, and electrical signals to and from the end effector. Furthermore, a standard for the mounting surface would provide the following benefits:

-A user can put the same gripper on any of his robots without redesign and rework.

-A user can replace a worn or damaged gripper without editing or reprogramming.

-A user can purchase a gripper from a specialty supplier with assurance.

IV. INTERFACE REQUIREMENTS

The wrist/end effector interface design addresses a variety of engineering concerns:

A. Mechanical Fastening

The wrist mounting surface, to which the end effector is attached, must withstand rated static and dynamic loads.

B. Registration and Orientation

Facilities must be provided for locating the end effector in the same attitude each time it is mounted.

C. Electrical and Pneumatic

Capability for actuators, powered tools and sensors on the end effector will require connections for electrical, electronic, hydraulic and pneumatic lines.

D. Replaceable or Quick Change

All applications require means for users to remove an end effector; some require the robot to change tools itself.

V. STANDARDIZATION

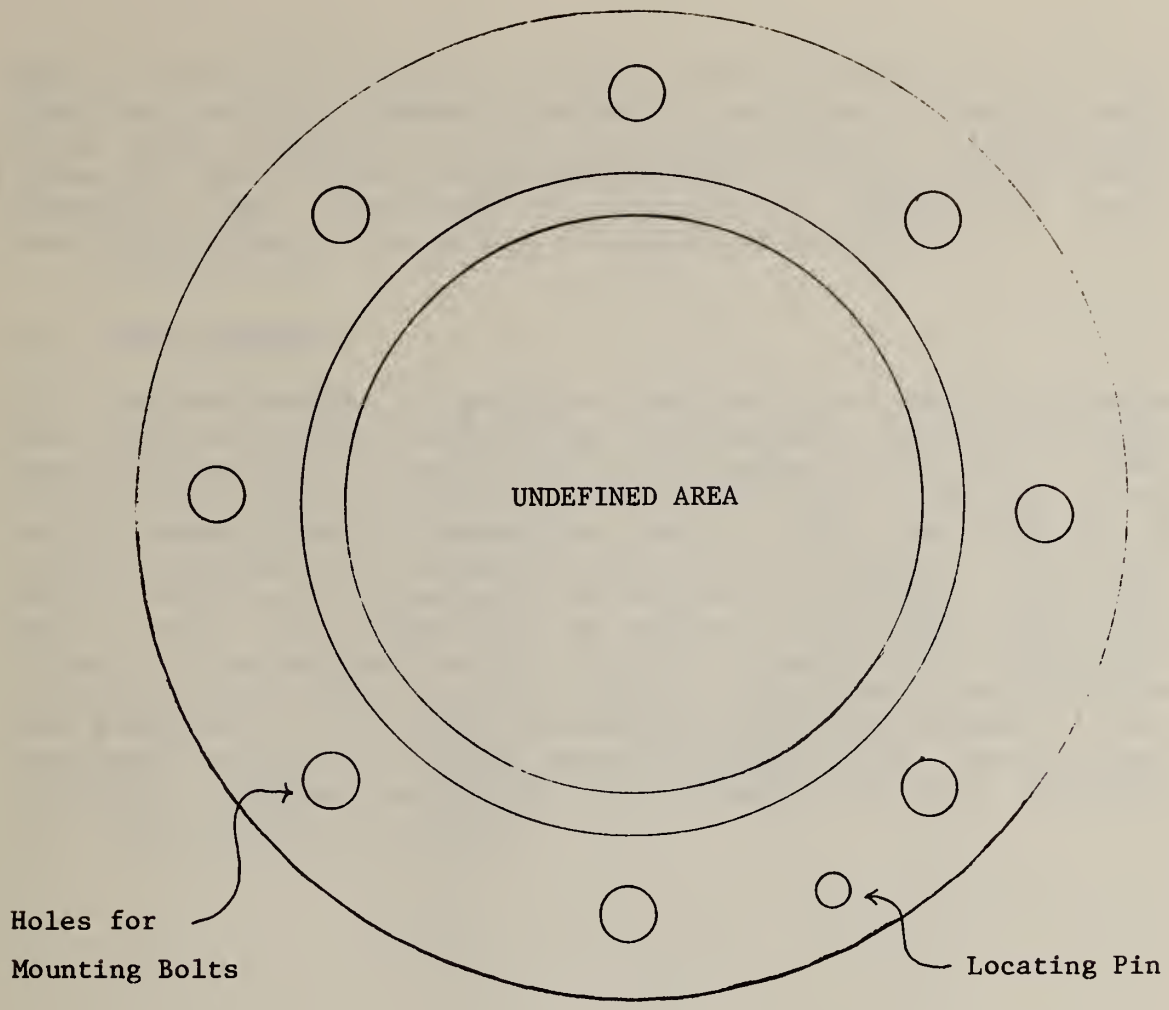
A logical stepwise progression was developed by the working group through which standardization of the wrist interface could be obtained. The strategy was suggested to start with the development of a common mounting surface for functionally equivalent robots. The surface will provide for fastening, registration and indexing of end effectors. Envisioned here is perhaps the disk shaped mounting surface shown in Figure 1 which incorporates a bolt circle for fastening, a land for shear strength and registration, and a pin for indexing. The bolt circle might be on the order of six inches in diameter for a large class of robots and include a four-inch and a two-inch size to cover smaller classes of robots. It is important to note that the interior of the circle is undefined allowing manufacturers leeway in incorporating this design into their product.

A second step in the strategy is to attack the problem of passing signals and power through the interface. Various alternatives were suggested leading to the only conclusion that the mechanical concept imposed no unrealistic constraints on the electrical/pneumatic problem and the latter was best left to a working group of design engineers under the standardization effort.

The last step in the strategy is to address the quick change problem. At least three designs are known to exist with one developed under Air Force ICAM funding and being available to any user. However, it was thought premature at this time to initiate standardization activity in the quick change area since the variety of signals to and from the end effector are so diverse in present applications that a consensus for standardization does not yet exist. A conclusion was made that a future standard for a quick change device is highly desirable and it is hoped that the experience gained in the above two areas by the standards group will pave the way for needed work on the quick change mechanism.

VI. RECOMMENDATIONS

To initiate standardization activity in this area the working group developed a letter request to the Robotics Institute of America for sponsorship of a group to develop a common interface between the robot and the end effector for functionally equivalent classes of industrial robots. The letter outlined the scope, overview and need for technical work. It was recommended initially that activity be directed at establishing a bolt-on mounting surface which will provide registration and indexing. Consideration could then be given to future requirements for utility services and quick change requirements.



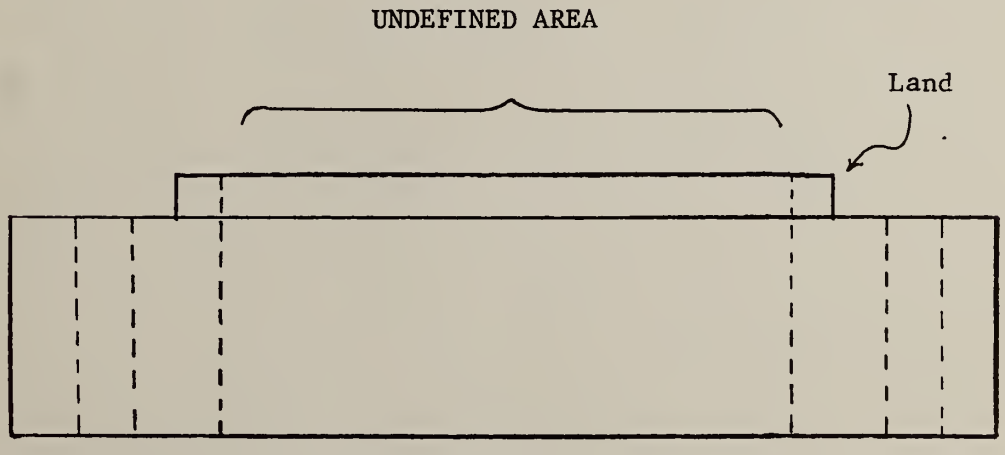


FIGURE 1

Typical Mounting Surface Geometry

One of the first tasks of any such effort would be to gather data on all U.S. Robotic mounting surfaces presently used. In an effort symbolic of the perceived need for standardization in this area, representatives from each robot vendor offered to make available the engineering drawings of their mounting surface. These are included as Appendix I.

VII. CONCLUSIONS

The present variety of grippers and end effectors is extreme and will probably continue to proliferate. While the robot manufacturer will provide design and manufacture of a special gripper as an initial service, his interests lie in the serial production of robots and not in the fabrication of single quantities of widely varying end effectors. The burden is falling increasingly on the user to design his own. While the rapid expansion of robots in the work force provides a total volume which could support an independent business concerned with the design and manufacture of robot end effectors, it is not yet viable since the method of attachment varies so widely among manufacturers. An adequate interface standard would be a most timely solution benefiting both vendors and users.

COMMON ROBOT CONTROL INTERFACE

I. GROUP MEMBERS

Barbera, Anthony J.*	-	National Bureau of Standards
Spaulding, Charles*	-	Unimation
Ames, Carl V.	-	General Dynamics
Baird, Henry S.	-	RCA
Colleen, Hans	-	ASEA
Dawson, Brian	-	Cincinnati-Milacron
Dexheimer, John	-	General Electric
Dunne, Maurice	-	Unimation
Makhlin, A. G.	-	Westinghouse
Mayer, Gordon	-	Wright-Patterson
Plumley, Wally	-	Lockheed-Georgia
Seltzer, Donald	-	Charles S. Draper
VanderBrug, Gordon	-	Automatix
Wheatley, Thomas	-	National Bureau of Standards

* Co-Chairperson

II. INTRODUCTION

This working group considered the specifications of a robot-independent control interface that would allow for real-time trajectory control of a robot. Such an interface would provide the option of controlling the robot with a user program generated external to the manufacturer's controller, where the robot's motions would be specified in some robot-independent form (e.g. Cartesian descriptions of the end effector's position and orientation). In considering this interface, the following topics were discussed:

1. The need for such an interface.
2. The level within the control structure at which the interface should occur.
3. The scope of the interface specification.
4. The possible standardization of the interface.
5. The timing of the interface specification effort.

III. THE NEED FOR THE INTERFACE

To better understand the question of the need for this type of robot interface, some background information is included here.

A. Background

Industrial robots are mechanical manipulators composed of some combination of prismatic (telescoping or sliding) and rotary-jointed linkages. A minimum of six joints is required to be able to position the robot's end effector at any point in space with any orientation. The interface discussed here is meant to provide a means of controlling these joints. To establish common dialog for the interface discussion different levels of control capabilities will be described.

1. Servo System

The more general-purpose industrial robots have their joint motions under servo control. The inputs to the servo control are the desired joint positions of the robot. These are compared to the feedback from the joint position indicators. If these values are different, a drive signal is generated to move each joint until the position error is nulled in a smooth, stable manner. This bottom level of control exists in all general purpose, programmable robots.

2. Joint-Space Coordinates

The joint position values that are input to the servo system are the coordinates in the joint space of the robot which describe some desired position and orientation of the manipulator.

Different robots within and between manufacturers are made with different size links and different combinations of prismatic and rotary joints. Thus two different robots capable of performing the same task of manipulating an end effector through the same trajectories, positions, and orientations could have very different mechanical configurations. (See Fig 1.) Therefore, the sequence of joint position coordinates for two such robots would be very different even though they were moving an end effector through the same trajectory in space. Even two robots of the same type from the same manufacturer would require different joint position coordinates due to lack of exactly the same zero location of position indicators, different tolerances in linkage alignment, slightly different

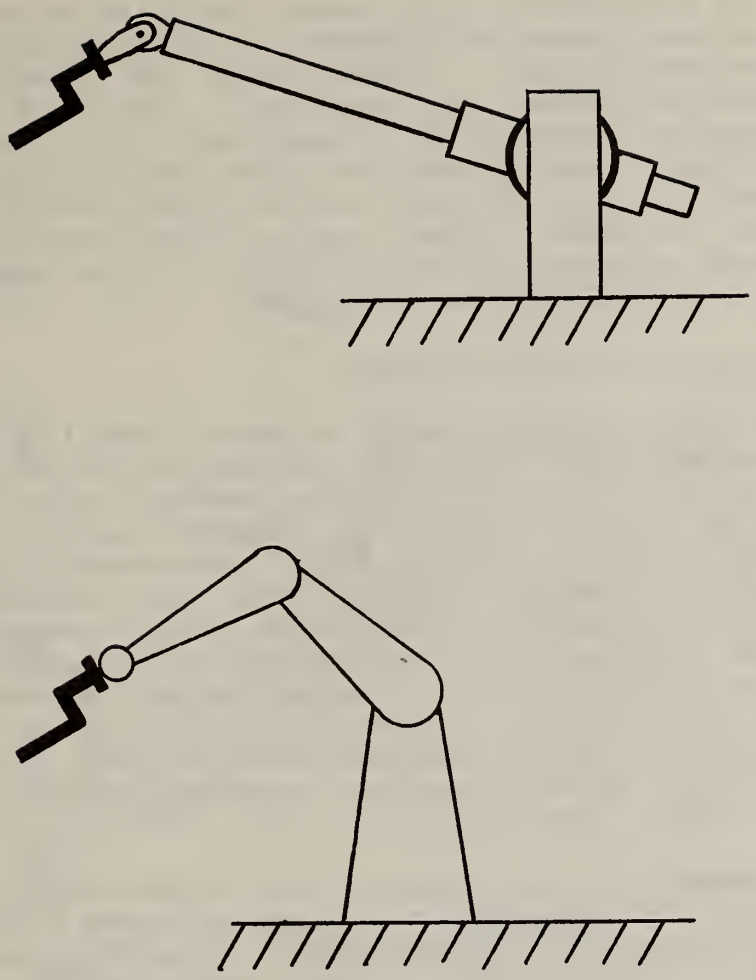


FIG 1 Example of how two different robot configurations can position an end effector in the same location and orientation. The joint values are very different between the two robots while the end effector's location in both instances can be described by the same set of robot-independent work space coordinates.

linkage length, etc.

The lack of compatibility between the joint-space coordinates of one robot with those of another robot does not pose a problem with robots used in stand-alone, teach-record-playback applications. This is because the joint-space coordinates played back to the robot are the values recorded off the same robot while it was being physically led through the task motions. However, these recorded joint values could not be played back through another robot and result in the same trajectories for the reasons mentioned.

3. Cartesian-Space Coordinates

The joint-space coordinates of a robot describe motions of the rotary and sliding joints of the particular robot involved. There are times, however, when it is advantageous to be able to command motion of the robot's end effector in some other coordinate system. Programming, trajectory control, and path modification using sensory feedback are all more easily commanded in coordinate systems other than joint coordinates of the robot. For example, it is simpler for an operator to move the end effector to desired locations if she is commanding end effector motions in an X,Y,Z frame of reference (e.g. with a joystick), rather than trying to move each joint. (See Fig 2.)

Motion commands can be given in any arbitrary coordinate frame of reference as long as there exists the necessary coordinate transformation algorithms to calculate the corresponding joint position values that will result in the proper joint motions. Each robot must have a coordinate transformation routine particularized to its linkage and joint configuration, as well as the zero points and resolution of the joint position indicators.

A set of values in an arbitrary coordinate system, such as a Cartesian coordinate system based in the workplace, is a robot-independent description of the position and orientation of the robot's end effector. (See Fig 3.) The coordinate transformation routine built into a robot control system will take these Cartesian values and transform them into the corresponding set of joint position values for that robot.

B. Controllers and Control Systems

When a user purchases an industrial robot, he also receives a controller which contains some form of control system that

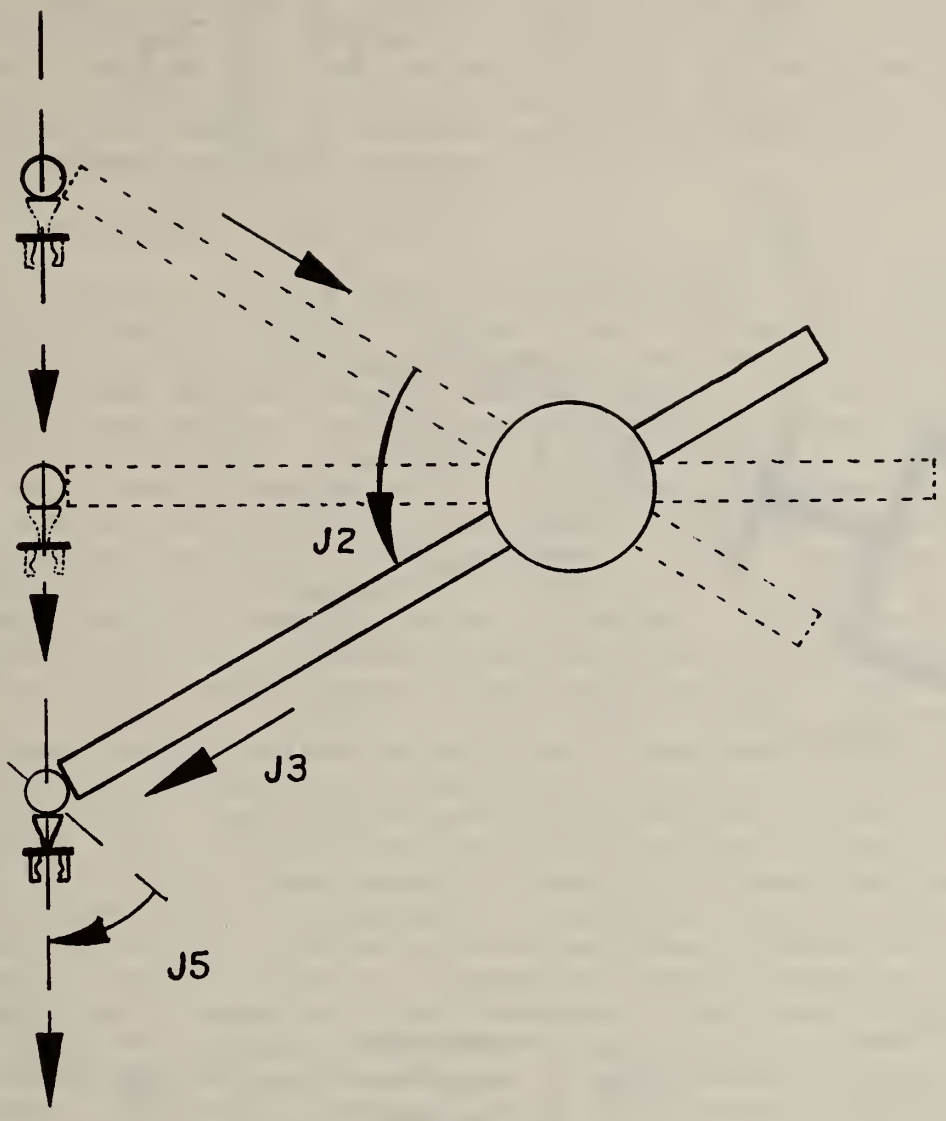


FIG 2 This figure illustrates the required coordination of a number of joints in order to obtain a straight line motion of the hand. For each incremental move along the straight line path, joints J2 (elevation), J3 (boom extension), and J5 (wrist flex) must be adjusted and coordinated.

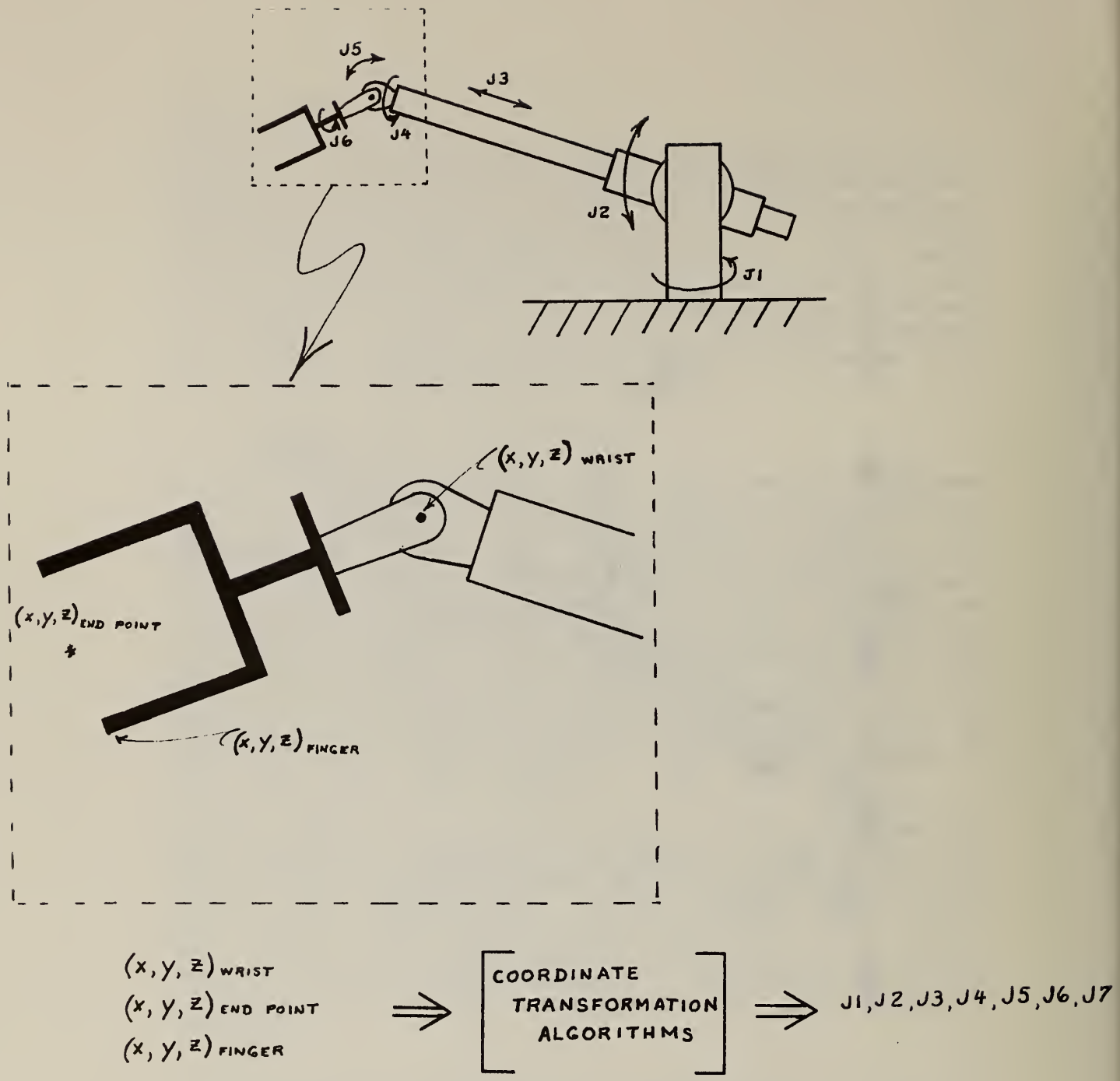


FIG 3 This figure depicts the specification of the position and orientation of the end effector by 3 sets of workspace robot-independent Cartesian coordinates. There is an x,y,z location to specify the wrist point, an x,y,z location to specify the end point and an x,y,z location to specify the finger tip. These would be the same values regardless of the type of robot used. The coordinate transformation algorithms will calculate the set of joint values (J1,J2,J3,J4,J5,J6,J7) required to position this particular robot in the correct way to put the end effector in the specified configuration.

includes a programming and execution capability. The user does not have direct input capability to the actuators, the servos, or to a coordinate transformation routine. Rather, the robot manufacturer has provided in the controller a user interface for programming a task to be performed. Once a task is programmed, the controller can be put into an execution mode in which the robot moves through the recorded trajectories. The following describes the type of programming required and the level of control capability available with typical systems.

1. Joint-Space Controllers

The simplest form of controllers available on servoed robots use a control system that is totally based in joint-space coordinates. These controllers are typically the teach-record-playback type. A robot with such a controller is programmed by moving the joints using individual joint control buttons until the robot is in a configuration that places the end effector in the required location and orientation for a desired trajectory point in the program. This position is recorded by reading in and storing the values from the joint position indicators on the robot A sequence of these points is programmed and recorded in this manner. To have the robot repeat this sequence, the controller is put into execution mode which simply causes the recorded joint values to be played back in the correct order and timing to the servo system. As the set of joint values that define each location are input to the servos, the joints move to these specified positions. When the joints have moved to within a predefined delta value of their commanded positions, the next set of joint values is sent to the servos. Different sequences of such joint coordinates can be recalled from memory and played through the execution mode of the controller causing the robot to perform different prerecorded tasks. However, as mentioned above, a stored program taught on one robot cannot be transferred to another robot and have the same task trajectories executed. Nor is there an interface to the robot's servo system that allows the user to input a separately generated set of joint position values. (This would be desirable, for example, if the user provided a separate coordinate transformation processor to calculate joint position values from sets of Cartesian coordinates.)

Joint-space controllers provide the necessary logic and algorithms to allow tasks to be programmed and played back, but limit the user's capabilities to only programmed tasks through which the robot must be

physically moved in order to record the joint position values.

2. Coordinate-Transformation Controllers

Several robot controllers are equipped with sufficient computational capability to calculate the transformation from some defined coordinate system to the joint coordinates of the particular robot. These controllers allow the user to program straight-line motions, motions about a tool point axis, and other trajectories which reference some relatively robot-independent coordinate system. Programming becomes a simpler and easier task since the operator can move the robot into position using a joystick, or other means, which command motions of the end effector directly. Motions up, down, sideways,and rotations about a tool tip are all commanded in a straightforward and simple fashion. The necessary joint coordination is provided through the coordinate transformation algorithms. Again, the controller provides programming and execution capabilities, but still does not provide a user interface into either the coordinate transformation calculations or the joint-position inputs to the servos. Rather, these are stand-alone controller-robot systems that are used to teach their own programs and execute them.

There are two major differences between these controllers and the joint-space controllers mentioned previously. First, coordinate-transformation controllers ease programming since coordinated joint motions are commanded in workplace or tool coordinates rather than the operator having to coordinate various joints through individual joint motions. Second, these controllers also provide a higher degree of trajectory control during execution. That is, even though the programming still consists of recording only the end points of the trajectories, the executed trajectories can be straight lines rather than the arc type motions about the rotary joints that are seen with the joint space controllers. This straight line motion is due to the real-time computation of the coordinate transformations. The end points are stored in the form of some workspace coordinates such as Cartesian, cylindrical or spherical. Intermediate trajectory points are calculated during execution and transformed to the corresponding robot joint coordinate values. This results in controlled-trajectory motion in the workspace coordinate system.

While this type of controller provides enhanced capabilities over the joint-space controllers, it still

limits the user to only that set of manufacturer supplied features.

3. Higher-Level Programmable Controllers

In general, the differences between the joint-space controllers and the coordinate-transformation controllers has been the inclusion of extensive computing capabilities. As the manufacturers increase the amount of real-time processing in their controllers, additional features can be incorporated. These could include trajectory optimization; trajectory smoothing through intermediate points; user-specified acceleration and deceleration profiles; trajectory modification based on sensory feedback data; off-line programming of trajectories without requiring the robot to move through the points in a teach mode; real-time branching to alternate routines to cope with varying situations and error conditions; self diagnosis; etc.

It is the addition of the computer to the industrial robot manipulator that is resulting in a truly general purpose programmable automation device with extensive features and capabilities. Indeed, the computer has created so many possible additional capabilities, that no one controller would be able to provide all of them. This is why it is important for interfaces to be provided to these controllers so that the full advantages of computer control can be realized. If the controller does not provide a certain function, such as trajectory modification using a particular sensor's feedback, and there is no interface into the system to accept this type of functional input from a user-supplied computer system, then the robot will not be able to incorporate this capability. With the proper interfaces into the manufacturer's controller, however, the robot's potential capabilities can be fully realized.

C. Advantages of Robot-Independent Interface

The following is a description of some of the advantages that would be provided by a robot-independent control interface.

1. Interchangeability

A robot-independent control interface would provide the capability of interchangabity. For example, one robot could be physically removed from a work area (for repairs for instance), another (not necessarily from the same manufacturer) moved into its place, and the same program executed without reteaching it with the

new robot. If several robots were to perform the same task in identical work areas, all of them could execute copies of the same program. This would be possible since the task program would specify the end points of the various motions in a robot-independent workspace coordinate system which would describe the desired positions and orientations of the end effector. These positions and orientations are a function of the task and the workspace geographical layout, not the joint configuration of the robot. The tailoring of these programs to a particular robot would occur within the manufacturer-supplied controller. The robot-independent trajectory coordinates could be fed in through the proposed interface, transformed into joint values for that robot, and executed.

If the output of the reverse transformation (namely from the joint space coordinates of the robot back to the same robot-independent coordinate system such as Cartesian coordinates) were made available through this interface, then a task program could be taught on one robot, the robot-independent form of the trajectory points stored external to the controller, and executed on any other suitable robot.

2. Optimized Control

This type of interface would also provide the user the option of generating specialized trajectories tailored to the particular needs of the task, thus allowing trajectory control capabilities not available from the manufacturer. The following are examples of capabilities that would become possible: special acceleration-deceleration profiles; passing through intermediate path points according to some desired smoothing algorithm; displaying controlled motion through some particular coordinate reference frame; moving about an end effector that is offset from the end plate of the robot in some unusual manner; programming tasks through a type of joystick input specialized to some optimal user coordinate reference system; varying tolerance specification on how close the robot must come to a programmed point before executing the next point; using specialized approach and departure paths; etc.

This interface being discussed supplies the controller with the executable trajectory points in a real-time manner. It essentially allows complete external control over the motions of the robot on an instant-by-instant basis where the type of control is only limited by the user's algorithms processed in the user-supplied external computing system.

3. Sensory Feedback Interaction

Goal-directed behavior refers to the capability of a robot to carry out a programmed task in spite of small perturbations in its environment that would normally lead to incorrect behavior in a simple teach-record-playback type of robot. An example of this is the acquisition of a part on a conveyor that has some arbitrary rotation and placement on the belt. In order for the robot to acquire this part, certain information must be made available to the control system to produce the corrective trajectory modifications necessary to cope with the situation and to pick up the part. This requires sensors to measure the conditions, computing systems to process the data into a proper form, and a control system to make trajectory modifications in real-time in order to accomodate these conditions. If the manufacturer's controller does not perform these functions or does not have the necessary type of interfaces to allow sensor input or external control of the robot, then the robot will not be able to exhibit this real-time sensory interactive behavior.

If an interface into the robot controller exists such that trajectory points can be input from an external source and executed by the controller in real-time, then the user has the option, by way of an external processor and this interface, of using any sensory feedback data to control the robot's motions. The user must provide the sensor, the sensory processing, and the control system algorithms for trajectory generation based on the sensory data. The output of this user-supplied system is a set of trajectory points in some defined robot-independent coordinate reference frame. These values pass into the controller, are transformed into the joint space of the robot, and sent to the servo system to execute the proper motions on the robot.

Thus, this type of interface would provide the option of having the robot interact in real-time with any sensor the user might desire to use in a particular application.

4. Off-line Programming

In general, there are two methods of programming a robot task. With one, the robot is "taught" the task by leading it through all the necessary locations, recording each to be played back in sequence during execution.

The other method, called off-line programming, creates

a task description without the necessity of using the robot's actual motion through those locations. This programming is accomplished in much the same manner as a computer program is written. A procedural description is generated indicating the flow of control with various branching conditions. The actual location values can be entered in a number of possible ways. These trajectory location values are stored in some robot-independent form (for example, coordinate values in a Cartesian reference frame). They may have been entered by typing the values into a terminal; having some program generate them, such as might be done with the locations in a palletized array; accessing an already existent data base (such as might be done with the locations of holes to be drilled in a wing skin, the positions of which are already stored in some CAD/CAM (Computer Aided Design/Computer Aided Manufacturing) system); leading a robot physically through the task and having the robot-independent form of the coordinates calculated from the reverse coordinate transformation of the joint position values. The important concept here is that since the trajectory locations are stored in terms of a robot-independent set of coordinate values, they can come from a number of sources and can be used to control execution of the programmed task on any robot with sufficient physical and performance capabilities.

Since these values are not dependent on a specific robot, they can be generated and manipulated "off-line," without the robot, and used in an integrated computer-aided-manufacturing environment where the execution of these programmed trajectories can be accomplished by any number of robots. The user is thus given the capability to do real-time scheduling, where jobs can be routed to alternate workstations and the robot-independent trajectories down-loaded to whichever robot is available in that workstation.

Clearly, the robot-independent control interface greatly aids in the implementation of a flexible, integrated system. It also forms a very necessary part of an effective "off-line" programming capability.

To fully realize the benefits of programming tasks by a procedural description of what is desired to be accomplished, it is necessary to be able to describe the tasks independent of which particular robot may be used. This is only possible if the programs can reference task positions and orientations in a form that is dependent on the required action, but not dependent on a particular robot's joint configuration. If this type

of robot independence is not available, every robot in every workstation would have to be led through every possible set of actions that might be performed at that workstation by that robot, and the joint values of every robot of every workstation for every one of these positions will have to be recorded. This programming effort is enormous, cumbersome, time consuming, ineffective, unstructured in its approach, tedious to maintain, and leads to unreliable and less than optimum use of the robots.

To reach a truly flexible integrated computer-aided-manufacturing system, robots will have to become general purpose, interchangeable modules with a common, well-defined interface into the rest of the system. Robot-independent task descriptions (trajectory points) will be passed through this interface to be transformed by the robot's controller computer into the appropriate joint values needed to execute the tasks.

D. Users of Real-Time Control Interfaces

The general consensus of the working group was that at the present time there would be a small number of potential users of this optional robot control interface if it were available. Most users would want to employ the robot in stand-alone applications where only the capabilities supplied by the manufacturer would be needed, and programming each and every task by the teach-record-playback technique would not be considered a problem.

Even though it was felt that the number of potential users was small, it was noted that they would be mostly the large manufacturers and therefore represent a sizeable number of installations. Several of these major manufacturing companies have already set up their own internal research efforts in this area as part of an overall program of upgrading and automating their manufacturing process. The computer is the underlying component of this effort in robotics, which is just one aspect of the overall goal of the integrated computer-aided-manufacturing system. The research effort to produce an easily programmable, flexible, automated manufacturing cell is directed towards this end. The robot is, of course, an important element in this system, but system integration requires well defined interfaces. For these research efforts to take full advantage of the sophisticated control and sensory interactive behavior possible with advanced computing systems, the robot cannot be a stand-alone device without any method for the user to interface to it to enhance and tailor its capabilities to the user's particular tasks. This type of robot-independent interface specification is one of the necessary first steps

required in the definition of a set of appropriate interfaces between robots and external computing systems providing sensory data, control, access to other data bases, and overall integration of the system.

In general, users of this type of interface would be those involved in research and development of integrated systems using robots. This also includes third party vendors who would develop and install robot systems in companies that did not have their own company based research effort in this area, as well as vendors that would supply different components such as a vision sensor module. This type of interface would considerably ease the task of integrating user-defined capabilities and sensory interactive behavior into a turn-key system. Vendors could develop their own retrofit packages providing users the ability to purchase the components required for the particular application, and essentially plug them together to produce the desired system. A user could, for instance, buy a robot from one manufacturer and a vision system from another and connect them to allow the robot to perform part acquisition of misoriented parts.

Thus, while the number of users who might make use of this interface directly is currently small, the number of applications could be considerable.

E. Control Level of the Interface

Possible interfaces into each of three different levels of control were discussed. (See Fig 4.) These were the servo level, the coordinate transformation level, and the trajectory generation level. The coordinate transformation level was thought to be optimal for this interface at the present time since it provided a fairly simple and straight forward control specification while being the lowest level that this specification could still be robot-independent. The following is a brief description of the advantages and disadvantages of interfacing to the three previously mentioned levels of control.

An interface into the joint servo system would allow user control of the robot but would require the user to provide the coordinate transformation routines. This interface would also be robot-dependent since the joint values for a given position and orientation of the robot are unique to each robot.

It was felt that a more likely level for the interface was at the control level above this, namely the level at which the robot trajectory points are transformed from some workspace oriented coordinates into the joint values for the individual actuators of the robot. Most manufacturers will

eventually supply this capability on their controllers for the ease in programming and trajectory control it affords. The interface into these transformations is rather simple and well-defined and provides the user the option of complete real-time trajectory control of the robot. At the same time, it offers the advantage of robot-independent specification of robot motions, with all the benefits previously described.

A number of the participants expressed an interest in an interface at a high level in the control system. Only trajectory end-points would be entered through this interface, while the robot's controller would provide the trajectory generation, acceleration and deceleration, path control, etc. The consensus of the group was that this would probably be a desirable interface, but at the present time it would be more difficult to define due to different ways trajectories can be generated, the various possible formats used by different manufacturers, and the rather unclear view of how to specify sensor interaction at this level. However, it was felt that as experience is gained with the more sophisticated control systems becoming available, an interface at this level should definitely be reconsidered in the future.

Again, the result of this area of discussion was that the input at the coordinate transformation level seems the most likely place for a common interface since it provides the lowest level of input to the robot that is robot-independent while affording the capacity for real-time control. It does, however, require the user to provide all the programming, sensory interaction, and trajectory-execution calculations on user supplied computing system that is totally separate from the robot controller.

IV. SCOPE OF THE INTERFACE

The specification of this interface has been partitioned into two parts. One deals with the communication aspect of the interface, the other with the control information.

A. Communication Interface

1. Hardware

There was general agreement that a hardware communications interface should be some standard computer-type interface. There is certainly no need to duplicate the efforts that are presently being carried out in the development of reliable communications hardware. Of the various hardware configurations available

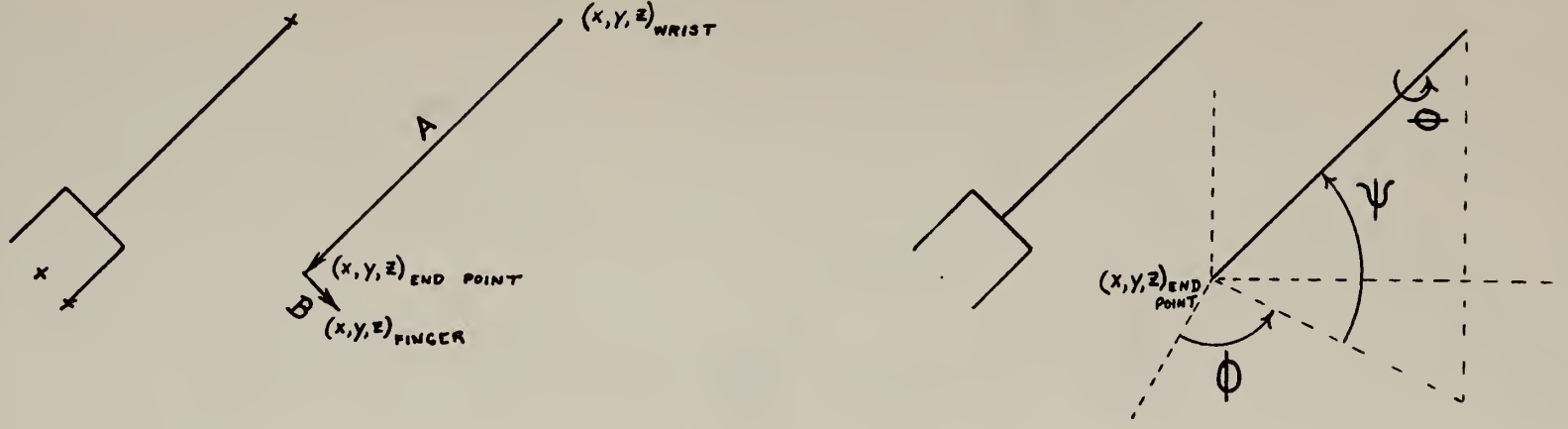


FIG 5 This figure illustrates how the position and orientation of the end effector can be specified in a robot-independent manner. Two examples are shown. The first uses 3 sets of coordinates in a Cartesian reference frame based in the work place. The end point (x,y,z) specifies the location of the tip of the end effector; this point and the wrist (x,y,z) define a vector A pointing along the axis of the end effector. This vector totally defines the position and orientation of the end effector except for the rotate about its axis. The end point (x,y,z) and the finger (x,y,z) are used to define a vector B that specifies this rotation of the end effector about its axis.

The second example shows the same end effector in the same position and orientation but specified in a spherical coordinate system. An x,y,z location defines the position of the end point in the workspace. The orientation of the end effector is specified by the three rotational angles shown.

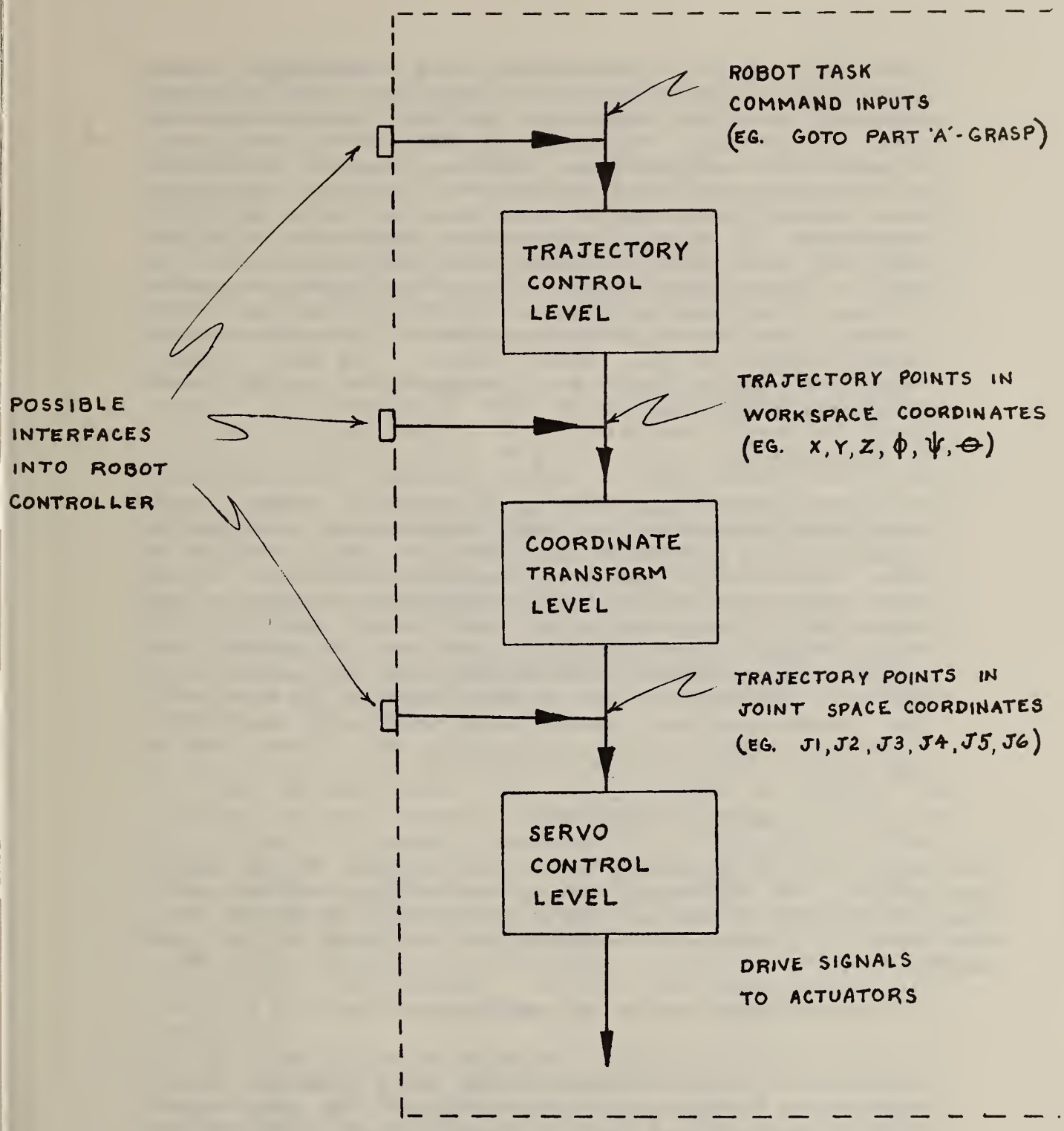


FIG 4 Schematic modularization of Robot Control System depicting possible interface locations for external control signals.

(parallel, serial, and network), the serial type seems most applicable since it requires few lines, can communicate over long distances, and has sufficient data rate capability. The amount of data to be transferred through this interface is relatively small. A suggested data rate was 19.2 kilobaud. This seems applicable since specification of a robot location in an arbitrary coordinate system is assumed to require nine values, each having 16 bits of precision. An update rate of 100 Hz was hypothesized yielding 1600 bits per second of data. This was multiplied by 10 to allow for other control information and communication protocol overhead giving 16000 bits per second (baud). The next highest "standard" data rate is 19.2 kilobaud, so it was chosen as a desirable value.

2. Protocol

It was suggested that the communications links should take advantage of the new LSI (large scale integration) communication circuitry that offers a large amount of user transparent protocol and error detection. Small chip sets have sophisticated protocol routines that essentially guarantee error-free communications by doing extensive error checking on the incoming data stream and allowing rebroadcast from the sender if an error is detected. Much of this protocol is handled in the communications system with relatively small overhead requirements on the processors at either end of the link.

B. Control Information Interface

This interface is specified by the information to be passed through it. The following describes the type of information that was thought to be necessary and whether it was to be in the form of a standard interface or specified by the robot manufacturer.

1. Information Content and Structure

It was decided that the information should be a set of coordinate values sufficient to fully describe the six degrees of freedom of an end effector on the robot. (For example, this could be done by specifying the position and orientation of a mounting face plate on the end of the robot arm.) These coordinate values would be in some robot-independent coordinate system, such as Cartesian or spherical. (See Fig 5.) It was also agreed that the particular coordinate reference frame to be used should be specified by the robot manufacturer. As long as the robot locations are specified in some

robot-independent coordinate system, it is not important which coordinate system is used. Any user with enough expertise to take advantage of this interface would have sufficient understanding of the control system to be able to easily generate a post-processor type of software module (see Fig 6) to transform the location values in some user-defined coordinate system into the coordinate system specified by the manufacturer. The user would only need to write one of these post-processors for each robot manufacturer, not one for each robot. All the robots of one manufacturer type would use the same post-processor.

The location values would represent absolute coordinates, as opposed to delta offset values from the last set sent to the robot. This interface essentially provides a switch (see Fig 7) that chooses either the manufacturer's control system or the user-supplied control system to input the robot-independent form of the trajectory coordinates into the coordinate transformation level in the robot controller. This interface would be a two-way link in that the coordinates specifying the next desired position and orientation would pass through to the robot contoller and the coordinates describing the actual position and orientation would come back from the controller's reverse transformation, having been calculated from the present joint position values of the robot.

Even though these coordinates specify position, velocity and acceleration can be generated by varying the spacial distance between specified locations. (See Fig 8.) Given a constant time interval, velocity is determined by the length of the commanded motion at each new update. For example, if the update rate is 30 Hz and the coordinate values input to the interface specify a 1.0 cm motion during each update cycle, then the resulting velocity is 30 cm/sec (1.0 cm/cycle * 30 cycles/sec = 30 cm/sec). If the coordinate values call for a 2.0 cm move each time, then the velocity would be 60 cm/sec.

If the length of the motion specified by the coordinate values is different for each cycle, then acceleration or deceleration will result. For example, if the robot started from a stationary position and the coordinate inputs called for motions, each 0.1 cm larger than the previous motions (that is, the first output would call for a 0.1 cm motion, the second for a 0.2 cm motion, the third for a 0.3 cm motion, etc.), then the robot would undergo an acceleration of 90 cm/sec/sec. This value is arrived at in the following manner During

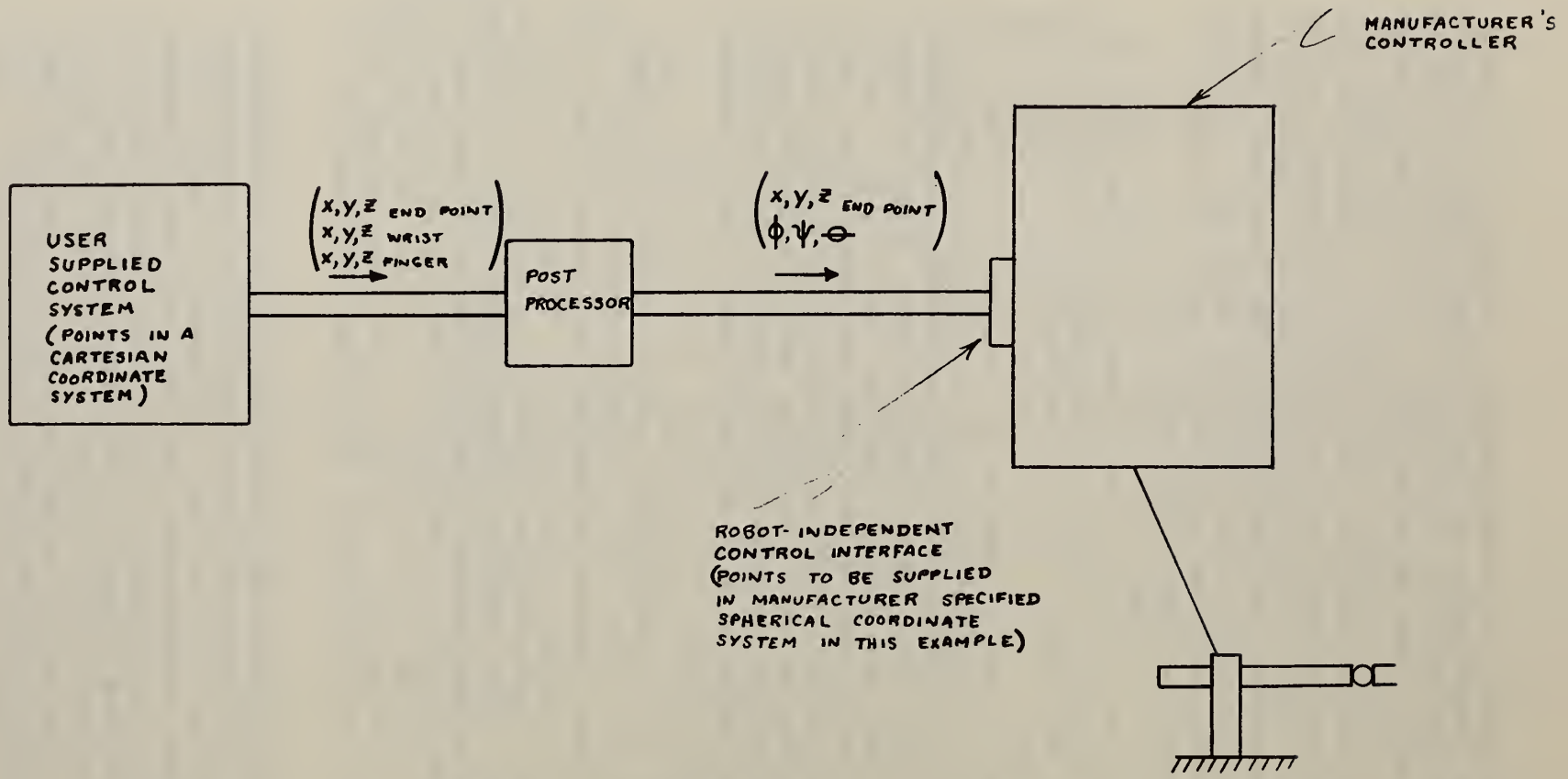


FIG 6 A block diagram illustrates the logical position of the post-processor to transform from the user defined coordinates into the manfacturer's specified coordinate system for the robot-independent control interface. This example shows the user's trajectory points defined by three sets of Cartesian coordinates as in Fig 4. The user supplied post-processor performs a transformation from this coordinate system into the spherical coordinate system specified by the robot manfacturer for input into the controller interface. Within the manfacturer's controller, these values are transformed by the robot-specific coordinate transformation algorithms into the joint space coordinate values for the particular robot as shown in Fig 6.

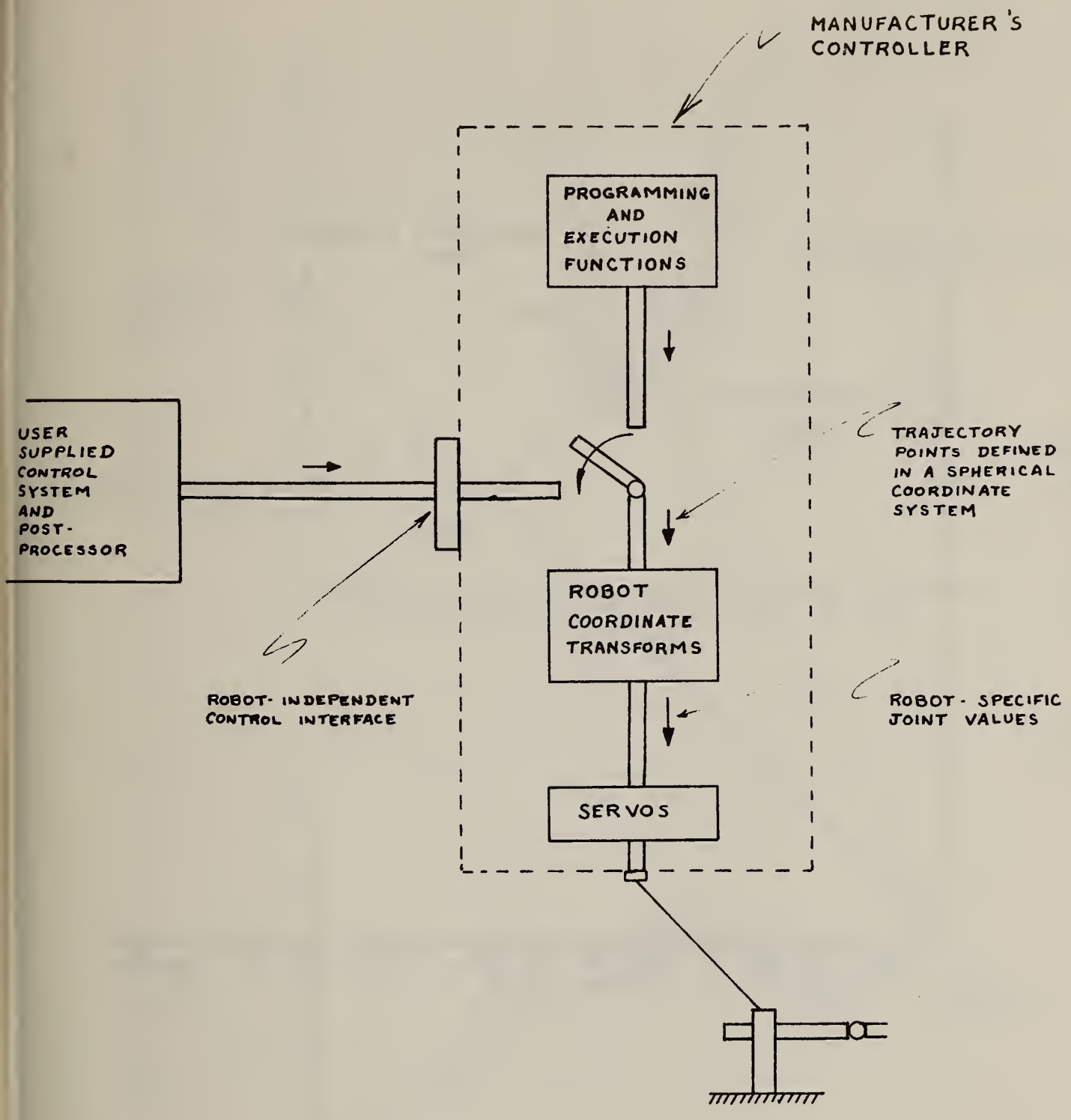


FIG 7 This figure schematically depicts the switching within the robot controller to accept trajectory point input either from the manfacturer's supplied teach-execute programs or from the user's supplied control system through the manufacturer's specified robot-independent interface. The user has post-processed the trajectory points into the manufacturer's defined coordinate system (in this example, a spherical coordinate system) which then feeds into the internal coordinate transformation algorithms to calculate the corresponding joint values for the particular robot.

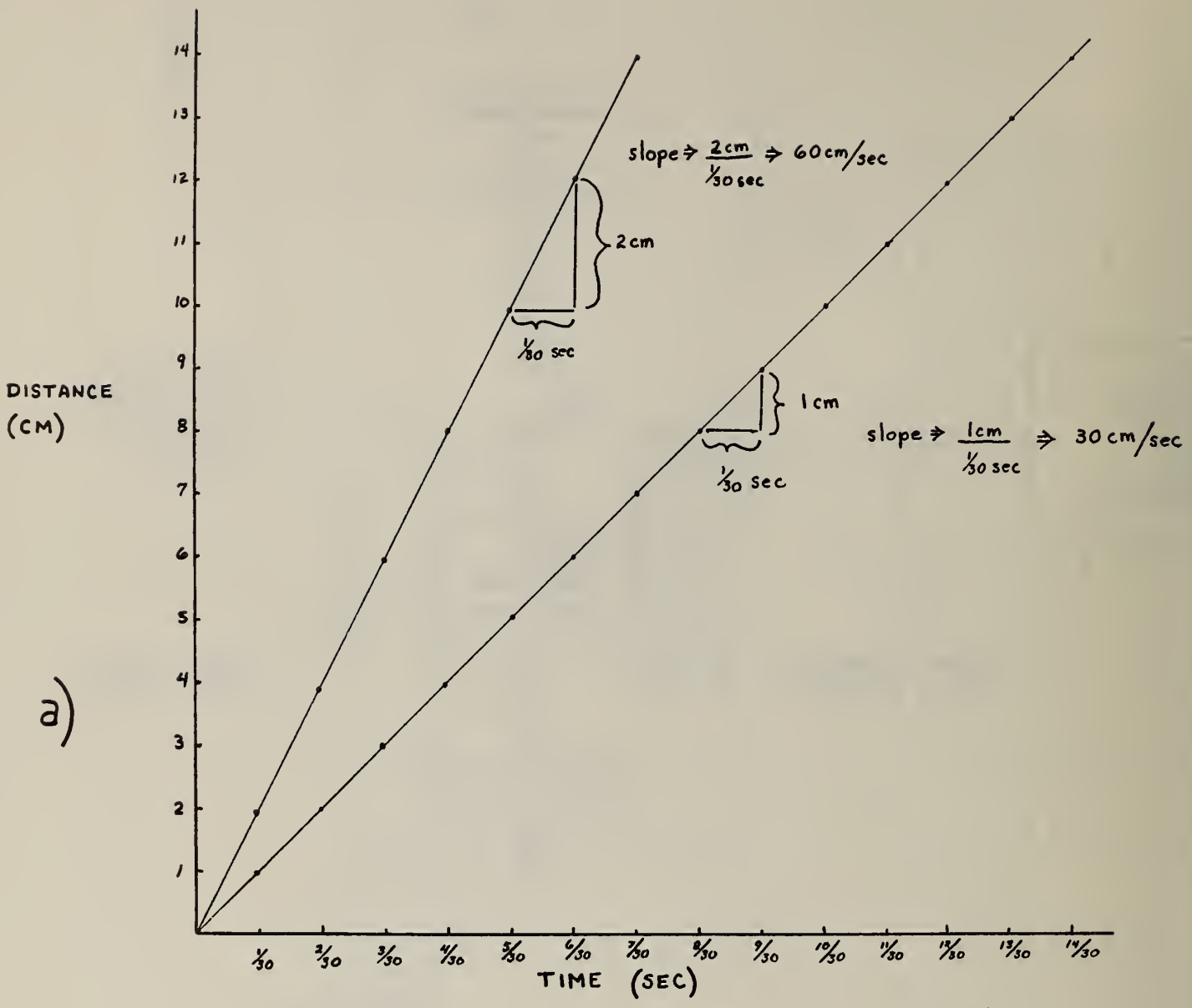


FIG 8a Plot illustrating calculation of two example velocities (30 cm/sec and 60 cm/sec) obtained by specifying position values at constant time intervals (1/30 sec).

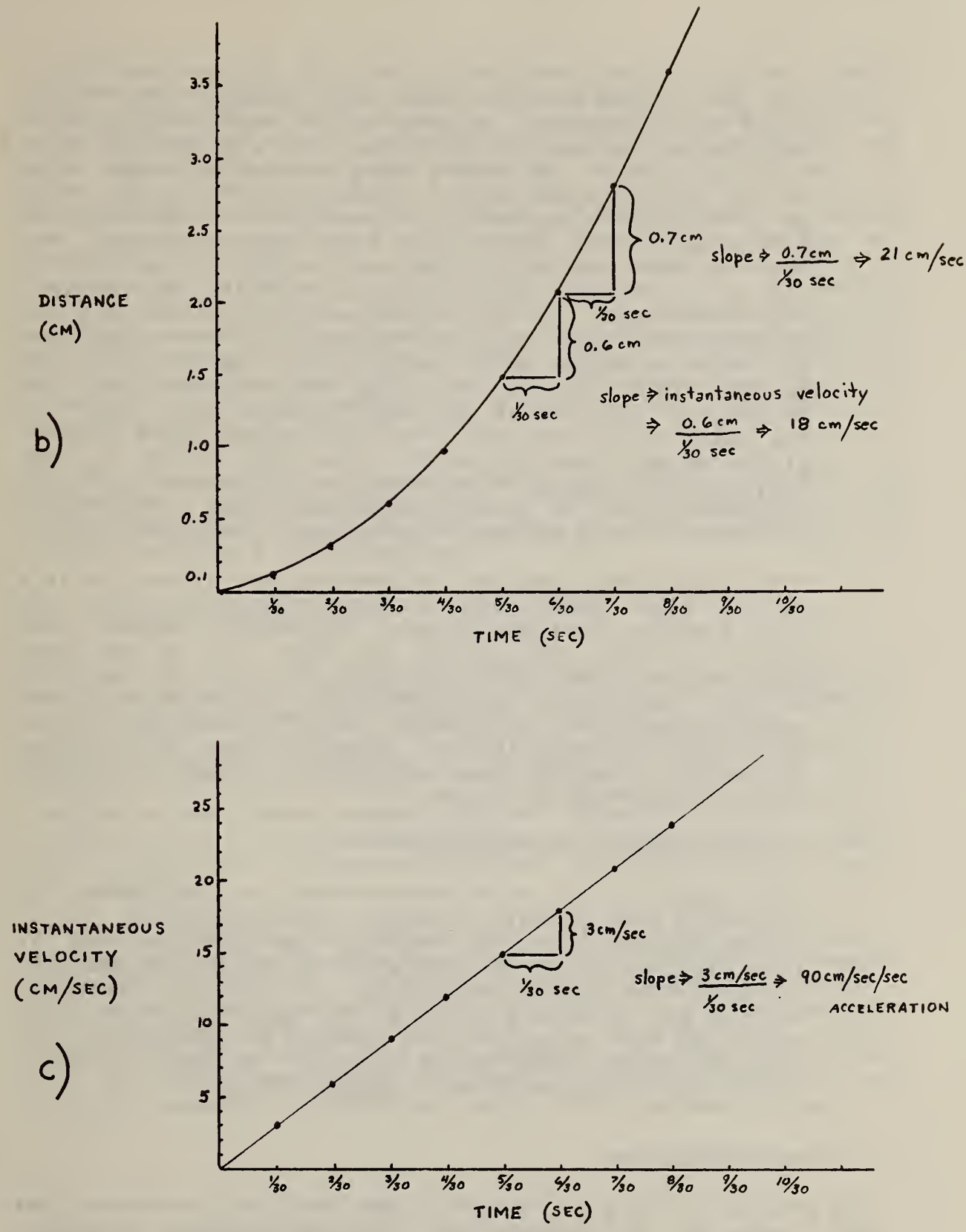


FIG 8b and 8c Example showing computation of instantaneous velocities (b) obtained from steadily increasing distance moves in constant time intervals (1/30 sec) and the resulting acceleration (c).

the last update period of the first second, the robot would be commanded to move 3.0 cm (30 updates, each lengthening the motion by 0.1 cm, for a total of 30 * 0.1 cm = 3.0 cm) during that particular update time period for an instantaneous velocity of 90 cm/sec. (3.0 cm in 1/30 of a second is an instantaneous velocity of 3.0 cm/cycle * 30 cycles/sec = 90 cm/sec.) During the last update period of the next second, it would move 6.0 cm for a velocity at that instant of 180 cm/sec. The difference in instantaneous velocities from one second to the next would be 90 cm/sec. (The instantaneous velocity would be 90 cm/sec at the end of the first second; 180 cm/sec at the end of the second second; 270 cm/sec at the end of the third second, etc.; that is, the velocity is increasing by 90 cm/sec each second giving an acceleration of 90 cm/sec/sec.) Thus, by controlling the length of the motions called for by each successive set of coordinate values, both velocity and acceleration can be specified through positional data.

This positional data will be sent to the interface at a fairly high update rate in order to ensure smooth and stable motion from the robot. The actual update frequency of the data will be left up to the robot manufacturer since this will be a function of the speed of the coordinate transformation algorithms in the robot controller. In general, these updates will be in the range of 25 to 100 Hz. Whatever rate is specified by the robot manufacturer will be a constant for operation of that type of robot.

2. Data Format and Precision

Coordinate values can be represented in a number of different data formats (integer, floating-point, etc.). It was decided that the format of the data should be specified by the robot manufacturer, as should the precision of the data. The values of these coordinates must be at least as precise as the joint position indicators on the robot, or else some of the precision capability of the robot would be lost. In general, these are in the range of 16-bit integer values.

V. STANDARDIZATION

The general feeling of the group was that this interface was not ready to be standardized. Standardization requires a total specification of all the data, in structure, format, precision, communication protocol, etc. It was felt that this would be too restrictive on both manufacturers and users at this time since the total requirements on this

interface are not that well understood yet. It was also agreed that there was a much better chance to get an interface of this type implemented by the manufacturers if they were allowed to specify the interface to be as compatible as possible with their present systems. As long as an interface is fully specified, and the coordinates are in a robot-independent form, the purpose of this interface would be met. As mentioned previously, the post-processing required to transform any user-supplied coordinates into a manfacturer's-defined coordinate system is a fairly straightforward task. The availablity of inexpensive processing capability has somewhat relieved the requirement of a single standard interface format, since, as long as the information is completely defined, it can be processed by either a user's or manufacturer's controller into the required form for the robot system.

It is not improbable, however, that a standard should be formulated at some later time, after enough experience has been gained concerning this interface to fully specify all of the necessary requirements.

VI. TIMING OF EFFORT

It was felt this was an appropriate time to consider setting up this particular interface, but too early for a standardization effort. Several robot manufacturers presently offer sufficient capabilities in their controllers that this interface could be introduced as an optional input to their robots. As other robots are introduced with at least the real-time coordinate transformation capabilities in their controllers, it is hoped that these workshop efforts will help to encourage the manufacturers' inclusion of this type of interface.

There is presently a considerable interest and the beginnings of in-depth programs within industry and government aimed at integrated computer-aided-manufacturing systems. The objective is necessarily dependent on a structured systems of modular elements connected by well-defined interfaces. These efforts will undoubtedly help to emphasize the need and the requirements for these interfaces. Now is the time to consider them and to begin their implementation.

COMPLEX SENSOR INTERFACE

I. GROUP MEMBERS

Albus, James S.*	-	National Bureau of Standards
Dexheimer, John*	-	General Electric
Baird, Henry S.	-	RCA
Becker, Dick	-	Cheeseborough-Ponds, Inc.
Berkowitz, Sid	-	Naval Ship R&D Center
Bowerman, Ed	-	GTE Laboratories
Brower, William S.	-	Control Data
Caughman, Don	-	Lockheed -Georgia Company
Dawson, Brian	-	Cincinnati-Milacron
Hacker, Patrick	-	Boeing
Handwerg, R. J.	-	General Dynamics
Kasvand, Tony	-	Nat'l Research Council Ontario, Canada
Mei, Lori	-	SME
Makhlin, A. G.	-	Westinghouse
McClure, Ward	-	Texas Instruments
Nagel, Roger N.	-	National Bureau of Standards
Potter, Louis	-	RCA
Seltzer, Donald	-	Charles S. Draper
Spaulding, Charles	-	Unimation
Vranish, John	-	Naval Surface Weapons
Walker, Jerry	-	Nordson

* Co- Chairperson

II. DEFINITION

A complex sensor is one which has some sort of preprocessor associated with it that performs at least an analog-to-digital conversion on the sensory data, and usually performs scaling, filtering, formatting, analysis, and coordinate transformations as well. The preprocessor is typically realized in the form of a microprocessor. A complex sensor communicates with the robot control system by means of digital signals.

III. NEEDS AND SCOPE

The need for complex sensors is obvious. Present day industrial robots are deaf, dumb, blind, and have little or no sense of force or touch. In order for a robot to interact with an even slightly unstructured environment, it must have the ability to sense errors and correct trajectories based on sensory measurements. This typically requires computation by the sensory data channel to detect features, recognize patterns, and compare observed input with internal expectations. Usually a coordinate transformation is

necessary before the sensory data can be used by the control system for modifying the behavioral actions of the robot. Certainly, if the data is visual, a great deal of computation is required before the control system can use it to make decisions.

The use of sensory data for controlling the robot's actions implies not only computation, but speed of response. The robot cannot exhibit high-speed, closed-loop performance (as it must if it is to be cost competitive) unless the response time of the feedback loop is short. Decisions cannot be made and errors cannot be corrected until the feedback information arrives at the control center. High-performance, closed-loop stability is incompatible with loop delays of more than a few milliseconds.

Sophisticated use of sensors and efficient processing of sensory data also requires input to the complex sensor from the control system. Sensory computations need to have information as to what action is being performed and what response is expected from the environment. In the manufacturing environment, this is particularly important because so much is, or can be, known about the environment. The vision system can be told almost exactly what to expect. It can be furnished with a model of the parts with which it must deal. It can even be told where they are (within some tolerance). The vision system of one robot can be given a picture of what a particular part looked like to the vision system of another robot that just finished handling that same part. Thus, it is important that the communication be two-way between the control system and the complex sensors. In fact, in many cases, the volume of data may be highest in the downward direction, from the control system to the sensory processing module.

There are, of course, many kinds of complex sensors. Vision is the most complex, the one for which the data rates are highest and the processing most complicated, but force, touch, and various proximity sensors can be extremely sophisticated as well. There are many types of data features, 3-dimensional models, and other types of information that must flow freely and rapidly back and forth across the sensory interface. Finally, there can be many sensors, or many different kinds of sensors, on the same robot or associated with the same group of robots.

These being the needs, the task is then to devise the most efficient, simple, reliable, and inexpensive interface standard that meets these needs. It is, of course, important that any standard for complex sensory interfaces not impede the development of any sensor technology because of unnecessary limitations of the interface itself.

IV. THE WORKSHOP DISCUSSIONS

The complex sensory interface subgroup focused its discussions on the configuration shown in Figure 1. There are, however, other possible physical configurations than the bus structure of Figure 1. There is the star configuration of Figure 2 where each complex sensor communicates with the robot controller through a separate port. This simplifies the protocol, since there is only one sensor on each port, but it requires a large number of ports. In practice this is often difficult because of the large number of pin connections required. One way around this difficulty is shown in Figure 3, where the robot controller communicates through a single port to a multiplexer which provides the required number of ports. A variation on the bus configuration is the ring structure of Figure 4, which has the advantage of redundancy. The ring can be broken at any point, and the ring becomes a bus.

While some complex sensors may consist of nothing more than an A/D converter and data communications port, it was assumed that a great majority of complex sensors would contain a microprocessor with a program and a data storage area. Thus, it was agreed that the interface should provide means by which process-to-process communication could take place.

Figure 1 indicates the various types of information that need to flow in both directions through the complex sensory interface. Information flowing from the complex sensors to the robot controller consists of highly processed information such as part position and orientation; part classification or identification; inspection data such as part dimensions, surface finish, verification of the presence of holes, etc.; force or touch vectors, or even matrices indicating the robot motions called for by sensory measurements.

Information flowing from the robot controller to the complex sensors is of the form of commands to make certain measurements or to execute certain sensory processing algorithms; state variables indicating what action is being executed by the control system or by other systems such as conveyors, robot carts, machine tools, etc; or even expected data or parameters such as images, maps, or features which may have been derived from teaching or from data bases containing part characteristics and process plans. The downward link from robot controller to complex sensor processor may also be used to download programs.

This implies that the downlink will often be required to convey large quantities of data. In most cases the downloading of large programs or data bases such as images will not be done during program execution, but will take place

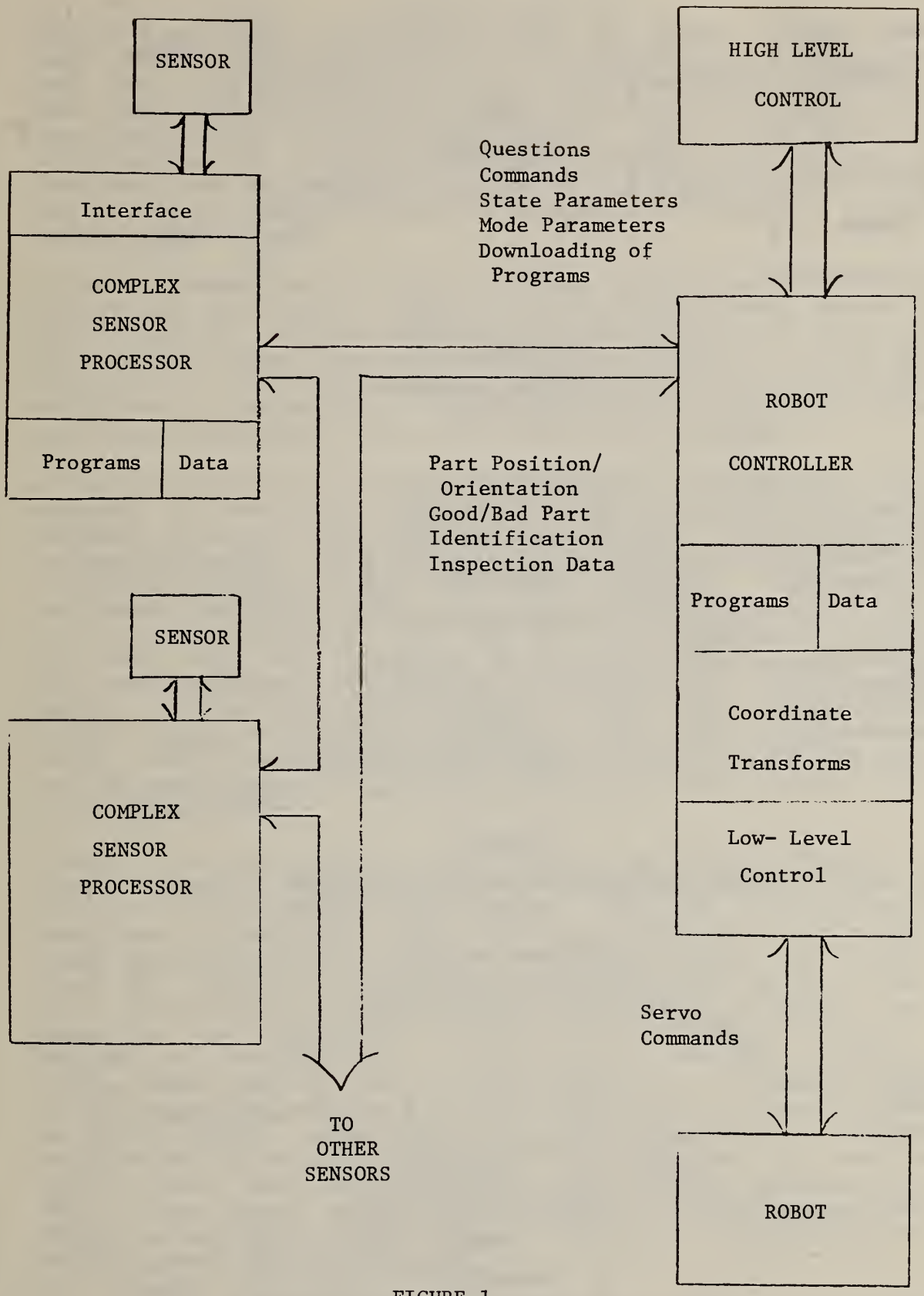


FIGURE 1

Basic Configuration of a Complex Sensor System

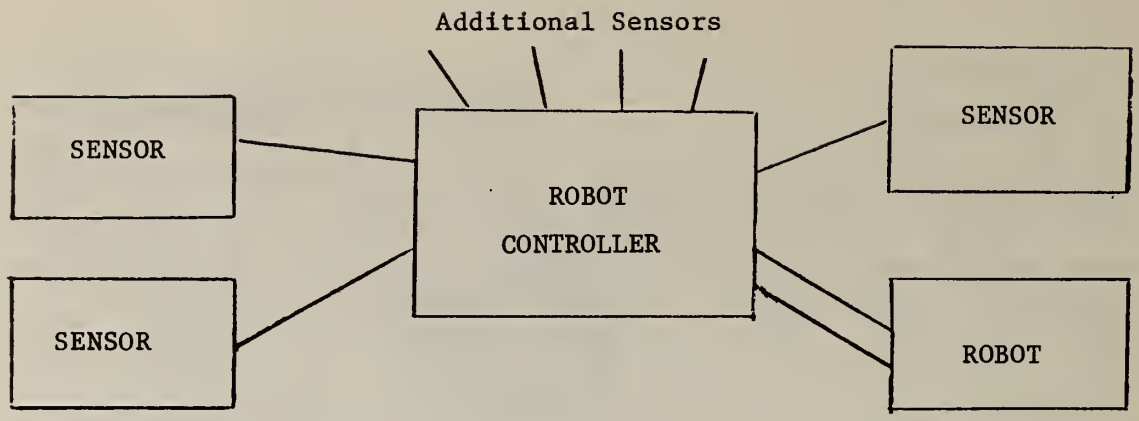


FIGURE 2

Star Configuration

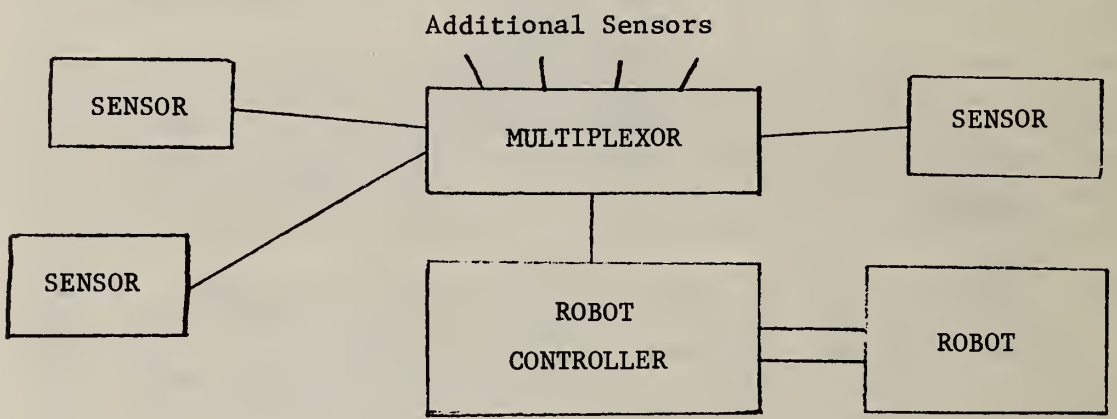


FIGURE 3

Star Configuration using a Multiplexor

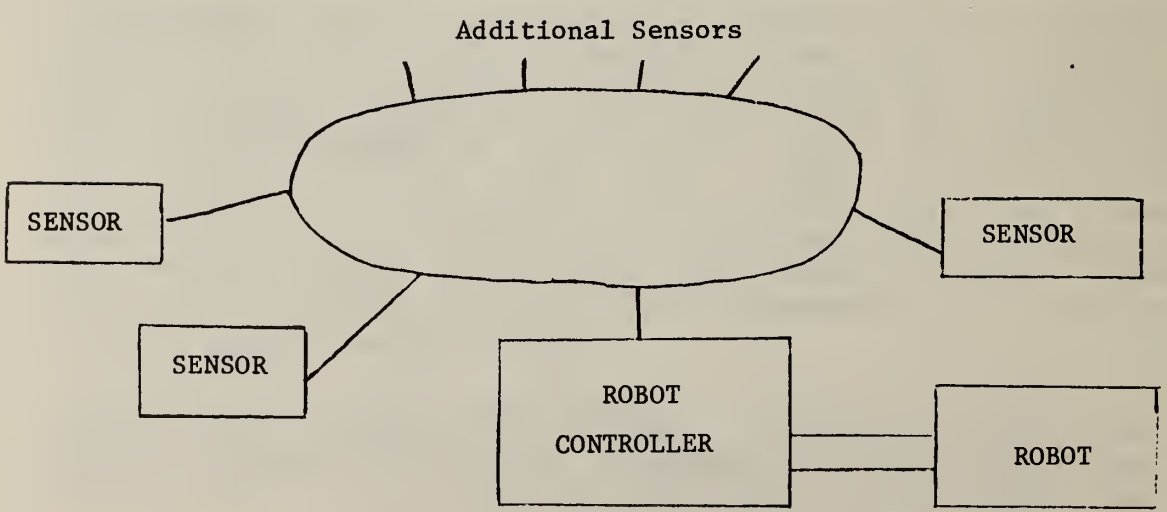


FIGURE 4

Ring Configuration

before the robot's operational task is begun. Thus, the data rate need not be sufficient to carry out large block transfers in a few milliseconds. Nevertheless, it is possible that the downloading of programs or data may need to be done frequently, (for example, between each part that is presented to the robot), so the data transfer may need to be done in a few seconds, as opposed to a few minutes. If this is the case, this requirement will dictate the speed of the communications link.

It was assumed that the robot control system would act as a master and the complex sensors and their processors would be slave devices. It was generally agreed that it is best if the robot controller polls the complex sensors, but it was understood that using interrupts may be required by some applications. Therefore, the communications should not rule out the possibility of an interrupt driven system.

It is crucial that sensory information be accessable by the program in the robot controller in an efficient and timely manner. It was noted that the time delays that could be tolerated could vary depending on the type of data and the use that is made of it in the control system. Information required for tight servo loops must be available within ten to fifty milliseconds. This is the turnaround-time, which includes the time required to send the command requesting the data plus the time required to collect and process the data, plus the time required to transmit the results of the processing to the robot controller. The data transmission rates need to be high enough to support this entire sequence for as many sensors as are required to respond within the loop delay period.

Control programs must not only be able to test and branch on complex sensory data but must be able to use sensory data as arguments in functions and routines. This is an extremely important requirement which does not presently exist on any commercially available robot. It requires that the interface at the robot controller have some means by which complex sensory data can be inserted into memory locations accessible as arguments by the robot control software. Without this feature it is cumbersome, if not impossible, to cause the robot to move in the direction of a sensed force or to track and acquire a visual feature on a part. Some existing robot controllers allow the robot to be commanded to move an incremental amount in x, y, z position, but not in pitch, yaw, and roll rotation. At present, however, even this is not easily accomplished through a convenient interface between the complex sensor and the data locations used by the robot control program.

It is equally important that commands and expected data from

CURRENT STANDARDS FOR

COMPUTER-COMPUTER COMMUNICATIONS

1. RS -232C

2. RS -449 (A high- speed version of RS -232C)

3. IEEE 583 This runs 132 parallel lines at 20 - 800 kHz, with a length of up to 100 meters

4. IEEE 583 (CAMAC) This runs 2 or 4 serial lines at 20 - 800 kHz, with a length of up to 2 kilometers

5. IEEE 488 This runs 24 byte-parallel lines at 1000 kHz, with a length of up to 20 meters

6. MILSTD 1553A This runs 2 serial lines at 100 kHz, with a length of up to 100 meters

7. MILSTD 1553B (A 1000 kHz version of MILSTD 1553A)

FIGURE 5

the robot control system be available to the programs in the complex sensor processor. This implies that the complex sensor have a comparable interface by which information and commands from the robot controller can be easily inserted into the data locations used by the sensory processing software.

The types of data, types of communication links, and various standard information protocols that have been used for similar kinds of networking in the past were discussed. It was felt that no new communications systems designs were needed, at least not initially. Presently used methods for computer-to-computer communication were felt to be adequate. The table in Figure 5 lists a number of present communications standards that were felt to be potential candidates for a complex sensor interface.

As a result of the fact that complex sensors will often be required to work in a shop environment where electrical noise interference is a serious and ever-present threat, it was felt that an adequate communication protocol must include some form of error detection and correction procedures. It was also felt that fiber optics would ultimately be the communications medium of choice. In the meantime, the use of a single coaxial cable was felt to be a desirable option. However, none of the standard systems in Figure 5 work with a single signal.

V. RECOMMENDATIONS

It was concluded that it is too early to recommend any specific standards for complex sensory interfaces. At present there are only a few commercially available robots with complex sensor interfaces of any kind, and these interfaces are limited in utility and cumbersome to use. Thus, there is very little as yet to standardize. Furthermore, there are too few persons with sufficient experience in complex sensors to make reasonable recommendations as to what such standards should be. However, this will undoubtedly change in the near future. Many research labs and robot users are pursuing the use of complex sensory data in robot control systems. Thus, an experience base is rapidly being developed.

Clearly it is not too early to begin discussing the requirements and suggesting promising approaches. This was the purpose of the workshop. It was recommended that these preliminary discussions be followed up with additional workshops and that contacts be established with the appropriate standards organizations so that complex sensory interface standards can be developed as soon as that would be practical. This is an area where interface standards

might be a powerful force in creating a market for complex sensors. An interface standard would allow manufacturers to develop a single product that could be used on a wide variety of robot types.

FUTURE GUIDELINES TOWARDS INTERFACES

I. GROUP MEMBERS

Nagel, Roger N.*	-	National Bureau of Standards
VanderBrug, Gordon*	-	Automatix
Albus, James S.	-	National Bureau of Standards
Ames, Carl V.	-	General Dynamics
Baird, Henry S.	-	RCA
Barbera, Anthony J.	-	National Bureau of Standards
Brower, William S.	-	Control Data
Colleen, Hans	-	ASEA
Dunne, Maurice	-	Unimation
Ennis, Jerry	-	McDonnell Douglas
Mei, Lori	-	SME
Makhlin, A. G.	-	Westinghouse
Mayer, Gordon	-	Wright-Patterson
Newlin, Burt	-	Department of Defense
Plumley, Wally	-	Lockheed-Georgia
Sheehan, Joe	-	Naval Ship R&D Center
Smith, Bradford M.	-	National Bureau of Standards
Snowdon, Al	-	Eaton Corp.
VanderBrug, Gordon	-	Automatix
Walker, Jerry	-	Nordson
Wiley, Jack	-	John Deere & Co. (NBS Planning Office)

* Co-Chairperson

II. INTRODUCTION

The workshop sessions on Future Guidelines towards Interfaces were specifically charged with discussing language, system integration, and database considerations in the future of robotics. It was a ground rule of the discussion group that the time was too early for standards efforts by the very nature of the topic areas. However, the directions and trends of the future of robotics were discussed with an eye towards the need for future standards and comments on standards were reflected throughout the discussions. It became clear early in the discussion that it was difficult to separate the robot of the future from the factory of the future and discussion often shifted to the role of automation in manufacturing cells and factories wherein robots played a significant role. It was conjectured that the role of the robot in the factory of the future was not as an individual component but as a part of the system. The session leaders made a strong attempt to focus the discussion around robots in the factory of the future and stand-alone robot systems of the future.

The discussions that took place at the two sessions covered not only the three areas of language, systems integration, and databases but also languages for the factory, systems debugging, and robot requirements in the future. In the write-up which follows, discussions are summarized for each of the six topics.

III. SYNOPSIS OF GROUP DISCUSSIONS

A. Robotic Languages

The group felt that the current method of teaching robots via a teach box would be replaced in the future by off-line programmming languages, "off-line" in this context meaning without the robot carrying out the commands in real-time, but not without a terminal and an interactive language processor. Language can broadly be categorized as the user interface to the control system of the robot. The general perception of the group was that robotic languages as found today can be put into two broad categories: explicit languages and implicit languages. An explicit language is one in which one talks about the robot joints and positions. Examples of explicit languages include VAL, EMILY, SIGLA, and WAVE. Implicit languages are those in which one describes the tasks to be performed rather the motions through which the robot will pass. Examples of implicit languages are AL, ROBOT APT, AUTO PASS, RAPT, and MAL. There was a general discussion about the fact that the two categories defined above were rather broad. In actuality, several levels exist within each of the categories, and there could be some disagreement about the categorization of a particular robot programming language. Because of the fact that most of the languages described above are not broadly available and that the group did not have significant experience with several of them, it was decided that it would be more important to talk about the desirable attributes of languages rather than go through an exhaustive analysis of the languages themselves. The group postulated the following as desirable attributes for robot languages.

> A language for programming a robot in an off-line mode would ideally be robot-independent. It should be a standard, and there might need to be several languages with efforts similiar to that being done for ROBOT APT at the current time. The language must allow for a hierarchy of control and exist at several different levels of complexity. The language should have provision for modifying the behavior of the robot due to the input of real-time sensory data. The language should have good interfaces to the various manufacturing and design databases found in a modern manufacturing environment. Programming of the robot in this language

should be engineered so that the user need not be a computer scientist but rather someone involved in the manufacturing process. The language must have provisions for system verification or debugging of the robot program as created at a terminal. Examples of this would be a graphic simulation of the robot program. Robot programs should not be made robot specific until the program is ready to be executed. That is, it should be symbolic until the time of execution. Post-processors for particularized robots should be created in order to allow the language to be independent but available for a variety of robots.

The group concluded that the several efforts on robot languages should continue and that others would most probably surface. At the current time it is far too early to begin a standardization process on any one particular language. However, as experience is gained in the use of these languages, some of them will naturally come forward as standards of the industry.

B. Systems Integration

The group felt that robots will naturally be imbedded into the factory of the future. The key concepts with respect to the integration of robots into the factory of the future are the accessing by robot programs of the databases found in computer-aided design systems and the passing of control information that needs to be used by robot programs in workstations and higher-order factory constructs. In both of these interface areas it was felt that there is not yet enough industrial experience to define standard interfaces. However, key issues were identified with respect to robots accessing databases in the manufacturing and design arena.

It was pointed out that it will be necessary for designers to indicate the view, pick-up points, and other parameters about a part being designed that are relevant to the robot manipulating that part. This requires the robot to have access to the descriptive geometry and part description information created by the part designer. With respect to control information, the simplest method of passing it to a robot from a higher-order construct would be in robot-independent position coordinates, using standardized terms and commands that could have meaning to multiple vendors' robots. In addition to control information, the robot will have to interact with other sensors in the automated factory. Because parts of these topics were covered in the complex sensor sessions, the group did not linger on this topic.

C. Databases

As was reported above, databases and the interface of robot programs to databases are important concepts if we are to integrate robots into higher-order factory systems. In particular, it was felt that the database interface was critical to creating a systems hierarchy for a manufacturing cell. Potential database interfaces were identified in several areas.

1. A manufacturing database.
2. Inventory of parts and equipment.
3. Shop floor database.
4. Scheduling.
5. NC databases.
6. Process planning databases.
7. Design databases.

The group's conclusion with respect to databases was that the concepts required in databases had impact on the ability to integrate robots into higher-order systems and on the design of robot programming languages. The group concluded that, as the use of robots becomes more complex and robots take on higher-order tasks in the factory manufacturing environment, they will by necessity need access to several of the databases outlined above in order to carry out their tasks.

D. Languages for the Factory

The group discussed the language for control of an automated manufacturing operation and spent a good deal of time trying to define the levels in a factory. There was excellent agreement on the fact that a factory was organized into a set of hierarchical components. There was significant disagreement on the identification of each of these levels.

Figure 1 shows the levels as perceived by the group in the ICAM terminology. The process is the lowest level in the ICAM terminology and represents the carrying out of a particular task. Above the process is a workstation which coordinates the efforts of several processes. Above the workstation is a cell, above the center, and above that, a factory. Typical commands for the bottom three levels of this hierarchy have been depicted in the figure.

As the group attempted to state commands for the center and the factory, it became clear that there was some disagreement as to the exact definitions of these levels. The group was, however, able to label several processes which go on across some of the higher-order levels, and they have been included in Figure 1. The general conclusion was that a

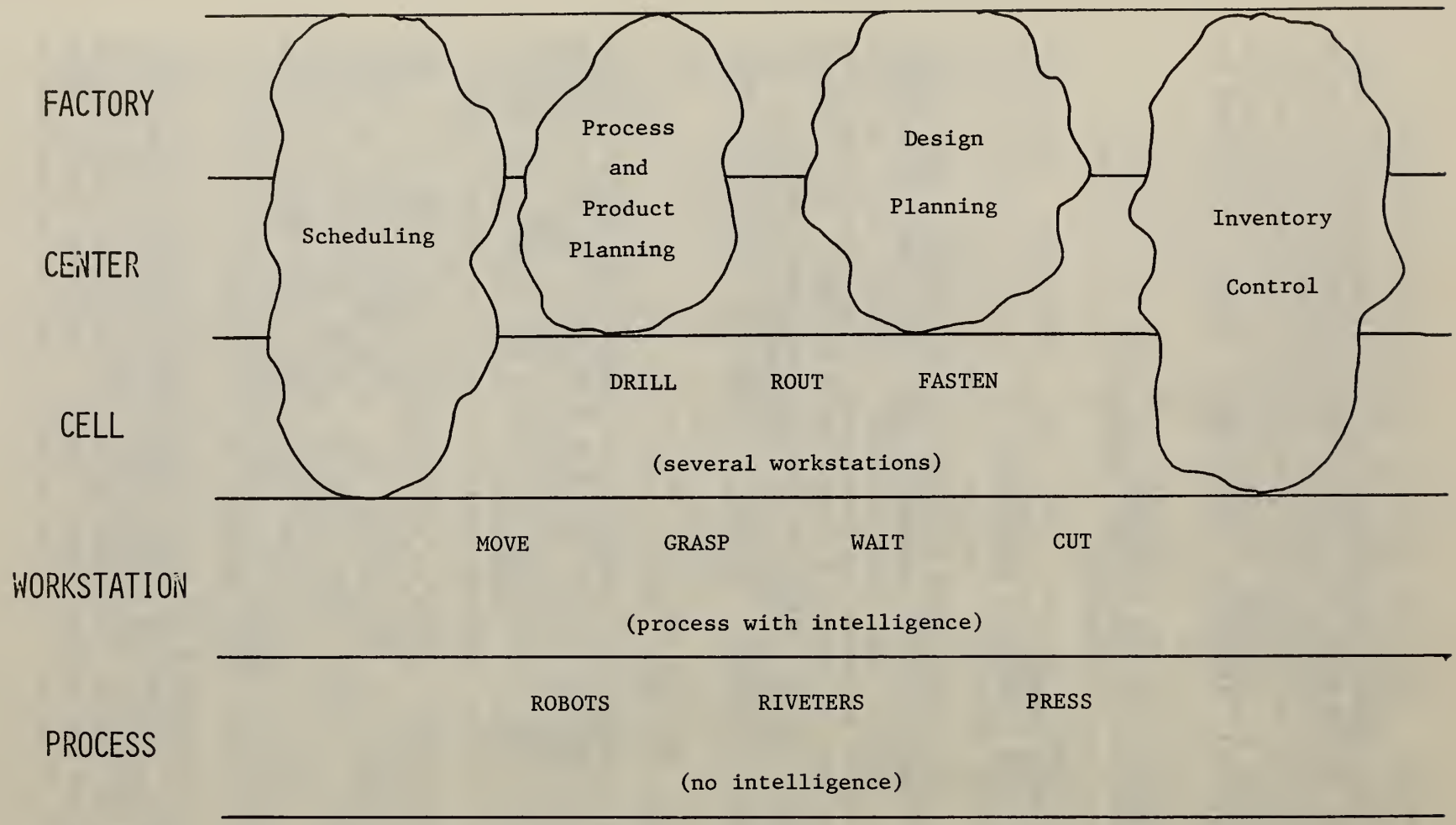


FIGURE 1

ICAM Automated Factory Levels

factory would be organized into a hierarchy similar to that used in the ICAM terminology.

As part of the discussion on hierarchical organizations, the group examined the hierarchy for the NBS robot. The NBS hierarchy has a complex task at the highest level, breaking down into simple tasks, trajectory calculations, coordinate transformations, and finally servo commands. It was felt that the NBS experience with this hierarchy substantiated the concepts expected to be used in a modern manufacturing cell. The group felt that, as factory plans are organized in the hierarchical nature and as they begin to be implemented, there would be better agreement on the terminology and commands which exist at the different levels of the factory. General principles that the group agreed upon were that the organization would be hierarchical; that it will use the task decomposition approach; that all the levels exist at each moment in time with different cycle times for up-dating; that each level in the hierarchy has its own commands or input language which could be carried out in time-sequencing commands and inputs into lower levels; and that each level would get input from above in the nature of commands and from below in the nature of feedback.

E. Systems Debugging

The topic of debugging constantly arose in the sessions. In particular, the availability of debugging tools in the future use of robotics was considered to be extremely important. The need for debugging, it was felt, underscores the essential nature of establishing well-defined interfaces. The use of graphics at the manufacturing cell level was considered to be an important ingredient for debugging off-line programming, in order not to tie up resources and for safety considerations (for both people and equipment). The use of graphics will require an interface standard to allow graphic products to work for more than one robot and more than one robot programming language. The group felt that, beyond emphasizing the concept of building in debugging tools for off-line robot programming languages, there was not yet enough experience to specify distinct or direct algorithms for the debugging.

F. Robots of the Future

The general consensus of the group was that robot systems would become more and more complex. They would use off-line, higher-order control languages as well as employ vision and other complex sensors. They would have robot-independent programming languages. Robot hardware would use standard interfaces to the language modules and include post-processors as well as defining robot-independent

position commands. Robot software would allow for the integration of the robot into complex factory systems or for a robot with a complex array of sensors to be used as stand-alone equipment.

IV. CONCLUSIONS

In summary, the Future Guidelines working group concluded that robots would be more extensively used in the factory of the future. Off-line programming languages for robots and manufacturing cells would develop in a hierarchical fashion. Advanced programming languages will need to be robot-independent and should provide extensive debugging techniques in an off-line mode. Interfaces will need to be defined for robots, databases, control systems, and complex sensory devices. In addition, hardware interfaces need to be defined to allow for smart end effectors and a variety of hardware connections of the robot to its environment. The general conclusion of the group, as expected, was that it is currently too early to start a standardization effort in these areas. However, the topics should be reviewed at a future conference, in order to measure progress and stimulate interest.

GENERAL RECOMMENDATIONS AND CONCLUSIONS

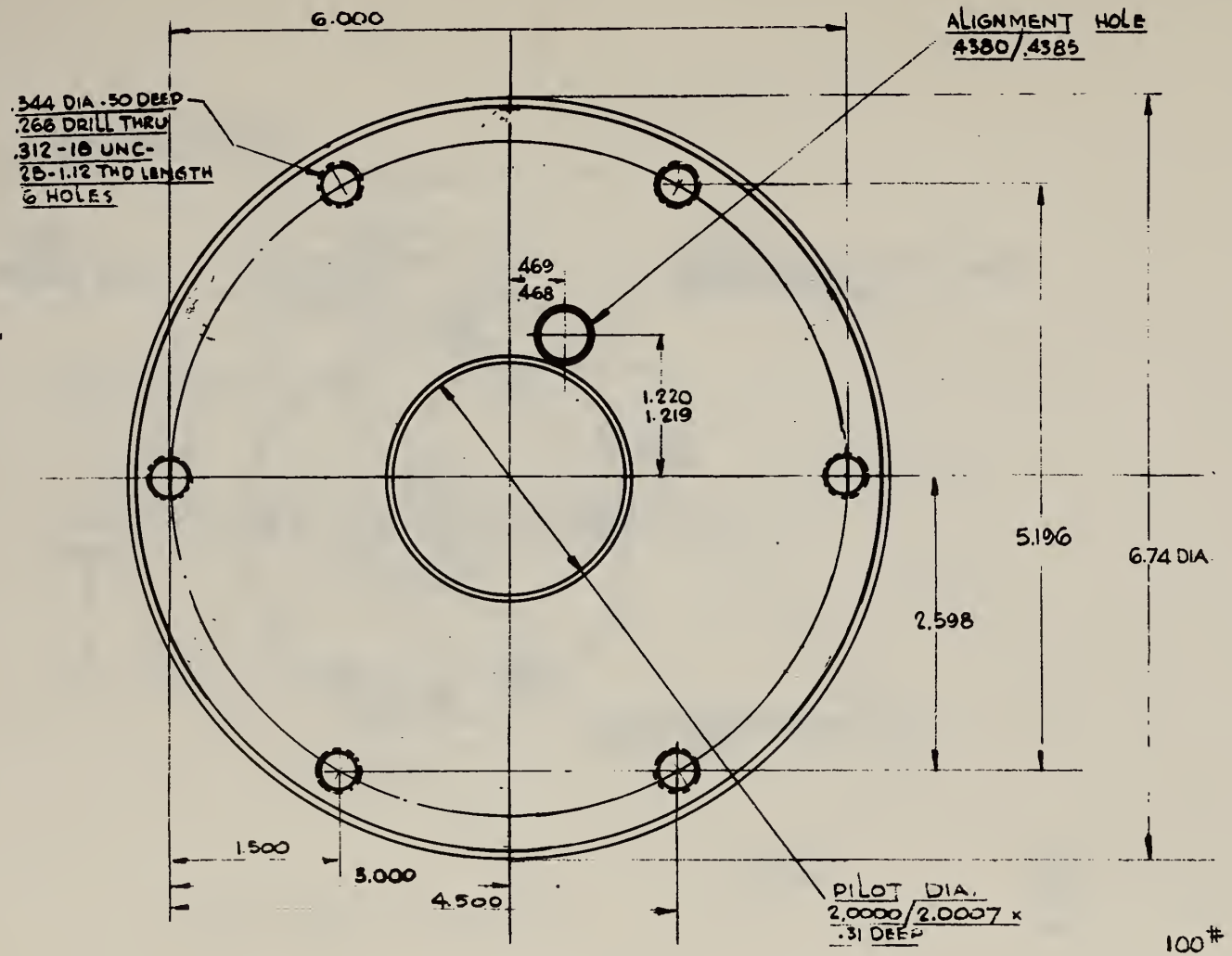
A group session was held on the last day of the workshop for general discussions and conclusions by the full group on the various interfaces. Each session chairperson presented a summary of their group's recommendations and conclusions. The starting up of a standards effort was recommended for both the Simple Sensor and Wrist Interfaces. It was felt that both had progressed to a point where standards could be implemented without impeding any developing technology.

The Common Robot Control and Complex Sensor Interfaces chairpersons reported on guidelines towards their respective standards. It was felt that the Common Robot Control Interface was ready to be implemented, but not ready to be standardized. Standardization would require a full set of specifications as to data type and rate, protocol, etc. to be agreed upon, and was felt to be too restrictive on users and manufacturers at this time. The Complex Sensor Interface chairperson reported that at present, complex sensors were not in wide use, and as such, also not ready to be standardized. It was felt that further discussions should be held at a later date. Both chairpersons recommended a waiting period for technology and experience to develop in these areas before actual implementation of a standard.

The Future Guidelines towards Interfaces chairperson, having a rather broad area of topics ranging from databases to off-line programming, outlined a list of desirable attributes for each of the topic areas. As most areas are still developing (or are virtually non-existent at present), it was recommended that each topic area be examined again in another standards workshop to be held in mid-1981.

APPENDIX I

The following drawings show the wrist mounting surface for typical commerical robots on the market today.



ORIGINAL DATE OF DRAWING			
REVISIONS			
NO.	E. C. N.	CHANGE	DATE
1			
2			
3			
4			

PIECE NO.	NOMENCLATURE	NO. REQ'D

NATIONAL BUREAU OF STANDARDS
WASHINGTON, D.C. 20234

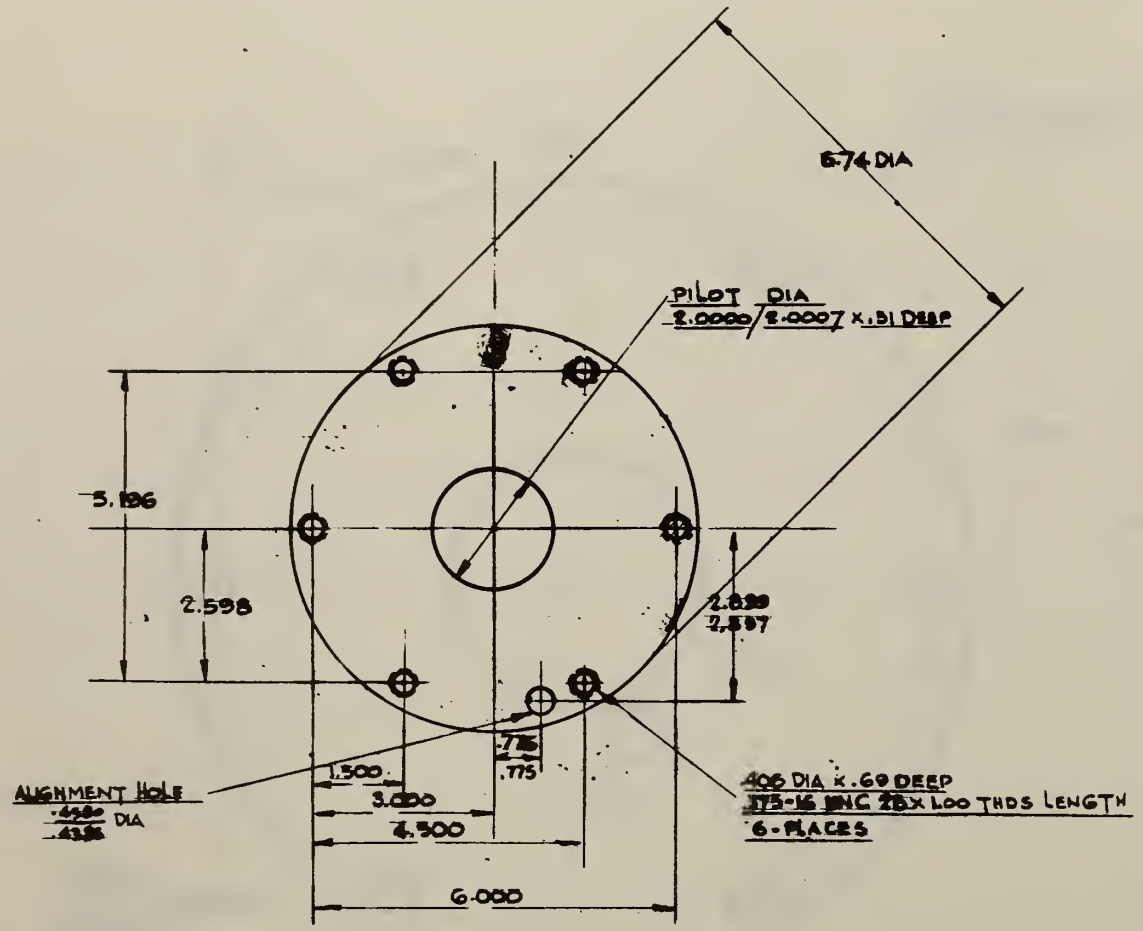
T³ ROBOT HAND MOUNTING PLATE

FOR CINCINNATI MILACRON

MODEL	TYPE	SCALE FULL
DIMENSIONS IN INCHES (Unless otherwise specified)	DRAFTSMAN 11-19-80 C.L.L.	CHECKER
TOLERANCES (Unless otherwise specified)	PROJECT ENGR.	PROJECT ENGR.
DECIMALS ±.005 FRACTIONS ±.015 ANGLES ±½°	SUBMITTED BY CHIEF, DSC.	
DO NOT SCALE THIS PRINT	EXAMINED BY CHIEF ENGINEER	
DIV. DSC. 738 .04	THIS PRINT ISSUED	APPROVED BY CHIEF, DIV.

738-12353-B

☆ U.S. GOVERNMENT PRINTING OFFICE GPO # 299-842



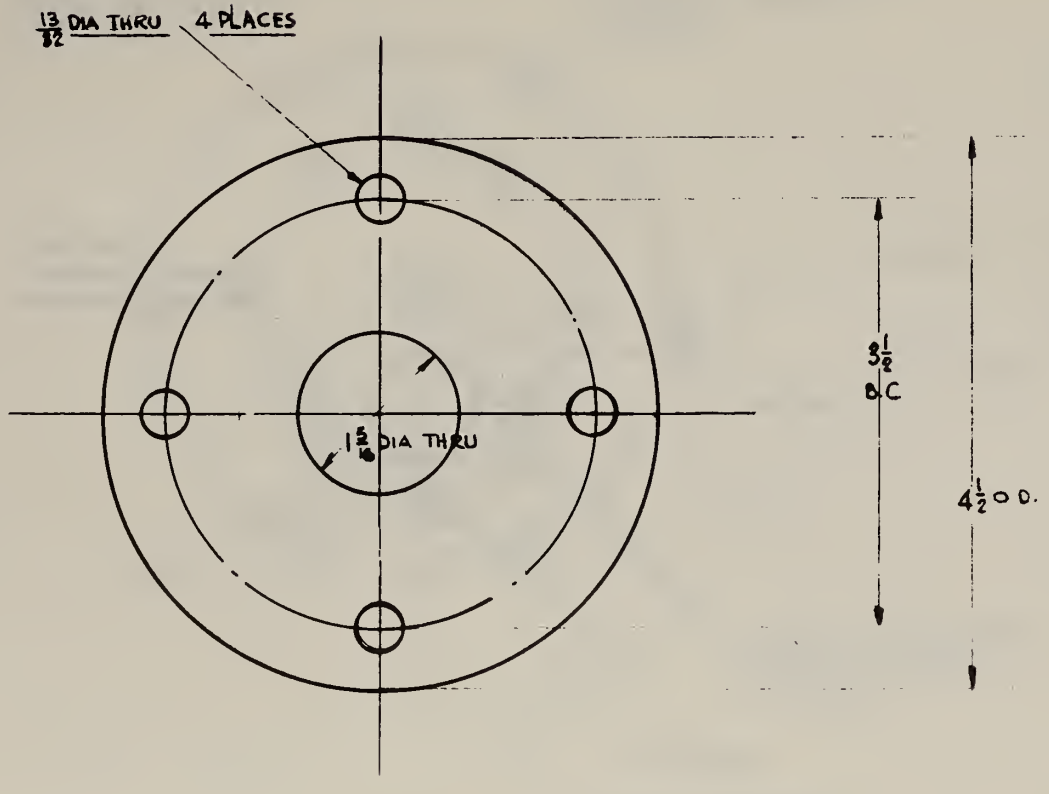
ORIGINAL DATE OF DRAWING			
REVISIONS			
NO.	E. C. N.	CHANGE	DATE
1			
2			
3			
4			

PIECE NO.	NOMENCLATURE	SIZE
NATIONAL BUREAU OF STANDARDS, WASHINGTON, D.C. 20234		
HT^3 ROBOT HAND MOUNTING PLATE		
FOR CINCINNATI MILACRON		

MODEL	TYPE	SCALE 1/2
DIMENSIONS IN INCHES (Unless otherwise specified)	DRAFTSMAN C.J.L. 11-25-80	CHECKED
TOLERANCES (Unless otherwise specified)	PROJECT ENGR.	PROJECT ENGR.
DECIMALS ±.005 FRACTIONS ±.015 ANGLES ±1°	SUBMITTED BY GROUP, SEC.	
DO NOT SCALE THIS PRINT	DRAWING BY GROUP ENGINEER	
DIV. SEC. 735 04	THIS PRINT ISSUED	APPROVED BY GROUP, DIV.
		738-12352-B

☆ U.S. GOVERNMENT PRINTING OFFICE: GPO 299-842

8½ x 11 OUTLINE



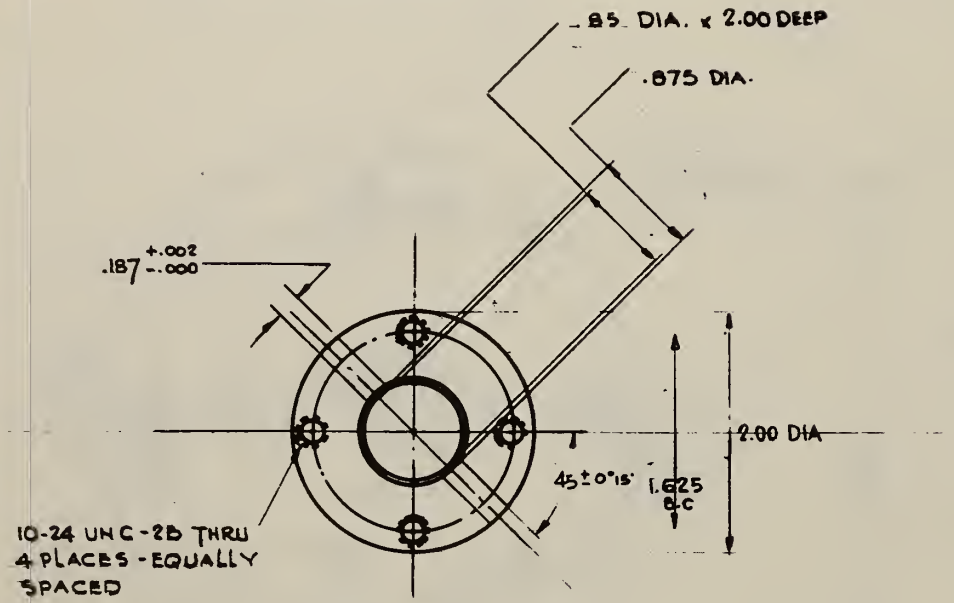
3/8 THICK HARD ROLLED STEEL

50*
75*
100*

ORIGINAL DATE OF DRAWING			
REVISIONS			
NO.	E. C. N.	CHANGE	DATE
1			
2			
3			
4			

PIECE NO.	NOMENCLATURE	
NATIONAL BUREAU OF STANDARDS WASHINGTON, D. C.		
GRIPPER MOUNTING PIVOT PLATE		
FOR PRAB CONVEYORS, INC		
MODEL	TYPE	SCALE FULL
DIMENSIONS IN INCHES (Unless otherwise specified) TOLERANCES (Unless otherwise specified) DECIMALS ±.005 FRACTIONS ±.010 ANGLES ±½° DO NOT SCALE THIS PRINT	DRAFTSMAN 11-18-55 C.Lh	MATERIAL
	PROJECT ENGR.	PROJECT ENGR.
	SUBMITTED BY CHIEF, DIV.	
	DRAWING BY CHIEF ENGINEER	
DIV. OR SEC. 738.04	THIS PRINT ISSUED	APPROVED BY CHIEF, DIV.
		738-12357-B

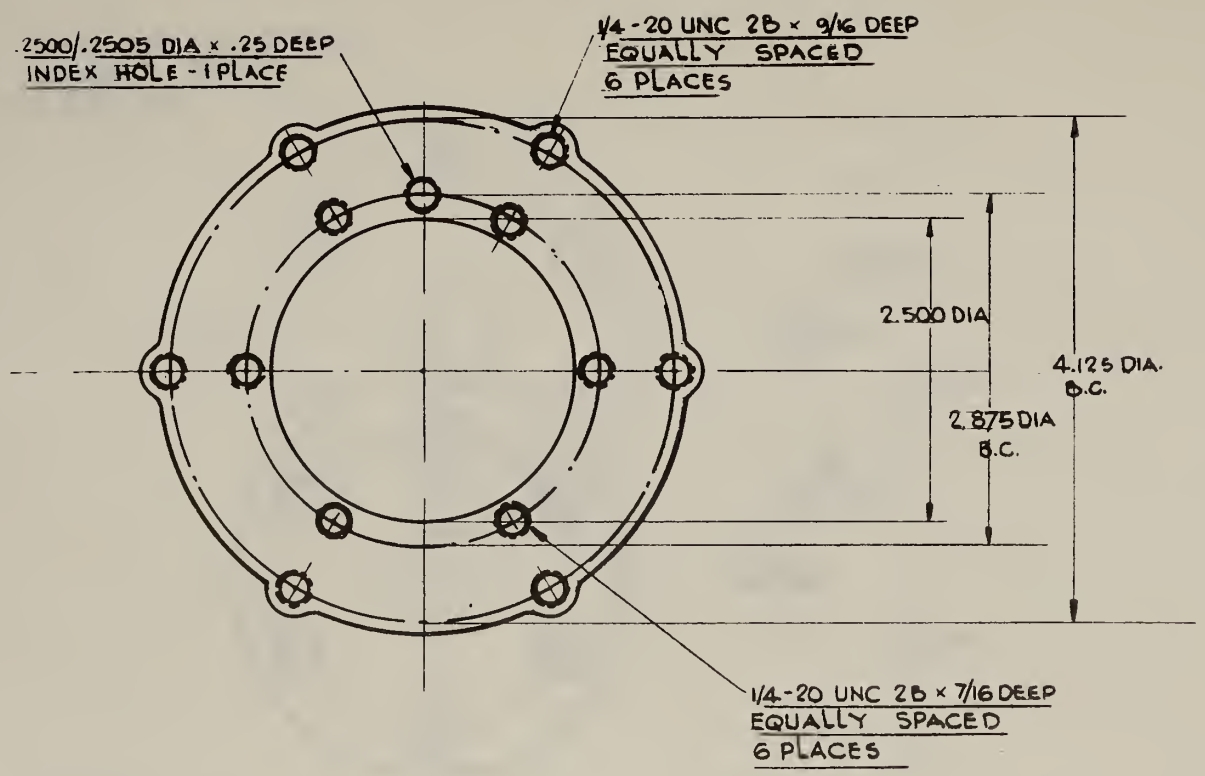
☆ U.S. GOVERNMENT PRINTING OFFICE : 0P0 799-842



ORIGINAL DATE OF DRAWING			
REVISIONS			
NO.	E. C. N.	CHANGE	DATE
1			
2			
3			
4			

PIECE NO.	NOMENCLATURE	NO. REQ'D
NATIONAL BUREAU OF STANDARDS WASHINGTON, D.C. 20234		
WRIST - MOUNTING FLANGE		
FOR PUMA		
MODEL	TYPE	SCALE Full
DIMENSIONS IN INCHES (Unless otherwise specified)	DRAFTSMAN 11-18-80 C.H.	CHECKED
TOLERANCES (Unless otherwise specified)	PROJECT ENGR.	PROJECT ENGR.
DECIMALS ±.000 FRACTIONS ±.000 ANGLES ±½°	SUBMITTED BY	CHIEF, SEC.
DO NOT SCALE THIS PRINT	EXAMINED BY	CHIEF ENGINEER
GOV. SEC. 738 .04	THIS PRINT ISSUED	APPROVED BY CHIEF, DIV.
		738-12363-B

☆ U.S. GOVERNMENT PRINTING OFFICE: 0PO# 299-842



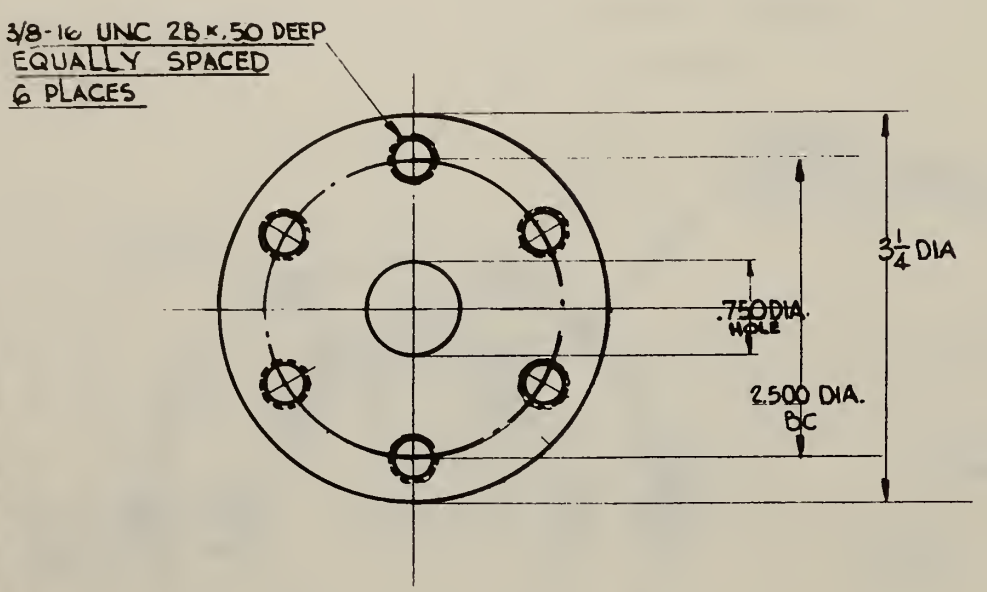
50#

ORIGINAL DATE OF DRAWING				
REVISIONS				
NO.	E. C. N.		CHANGE	DATE
1				
2				
3				
4				

PIECE NO.	NOMENCLATURE	
NATIONAL BUREAU OF STANDARDS, WASHINGTON, D.C. 20234		
MOUNTING PLATE - WRIST 5° ST'D.		
FOR UNIMATION INC 2000 SERIES		
MODEL	TYPE	SCALE FULL
DIMENSIONS IN INCHES (Unless otherwise specified)	DRAFTSMAN 11-28-80 - C.J.h	CHECKED
TOLERANCES (Unless otherwise specified)	PROJECT ENGR.	PROJECT ENGR.
DECIMALS ±.005 FRACTIONS ±.015 ANGLES ±½°	DESIGNED BY	GROUP, SEC.
DO NOT SCALE THIS PRINT	DRAWN BY	GROUP ENGINEER
DIV. SEC. 738 .04	THIS PRINT ISSUED	APPROVED BY GROUP, DIV.
		738-12364-B

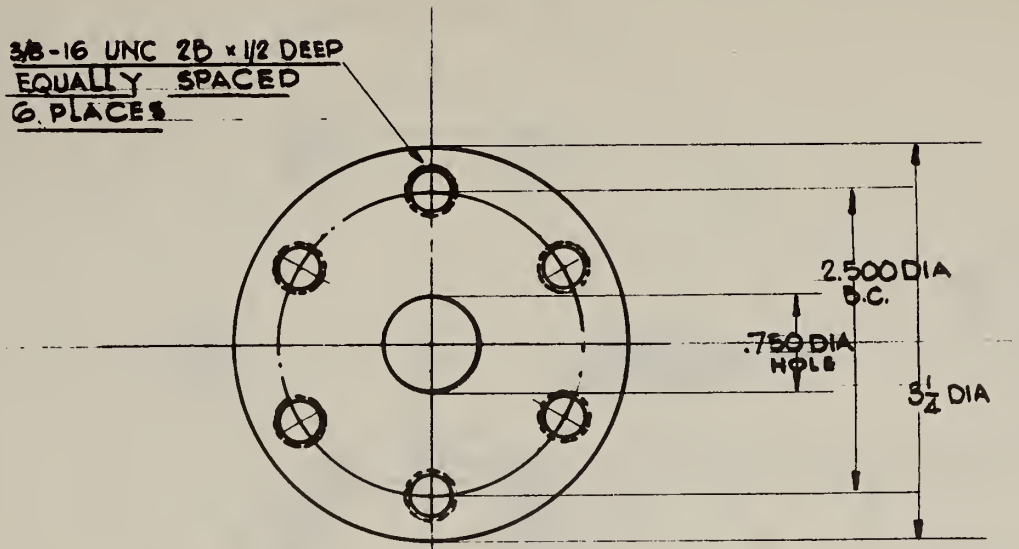
☆ U.S. GOVERNMENT PRINTING OFFICE GPO # 299-842

ORIGINAL DATE OF DRAWING			
REVISIONS			
NO	E C N.	CHANGE	DATE
1			
2			
3			
4			



PIECE NO.	NOMENCLATURE	NO. REQ'D
NATIONAL BUREAU OF STANDARDS WASHINGTON, D.C. 20235		
MOUNTING PLATE-WRIST 6° HVY. DUTY		
FOR UNIMATION INC.- 2000 SERIES		
MODEL	TYPE	SCALE FULL
DIMENSIONS IN INCHES (Unless otherwise specified)	DRAFTSMAN D-24-80 CJ/s	CHECKED
TOLERANCES (Unless otherwise specified)	PROJECT ENGR.	PROJECT ENGR.
DECIMALS ±.000 FRACTIONS ±.015 ANGLES ±1°	QUANTITIES BY	CHIEF, SEC.
DO NOT SCALE THIS PRINT	EXAMINED BY	CHIEF ENGINEER
DIV. SEC. 738 .04	THIS PRINT ISSUED	APPROVED BY CHIEF, DIV.
		738-12362-B

☆ U.S. GOVERNMENT PRINTING OFFICE: GPO # 299-842

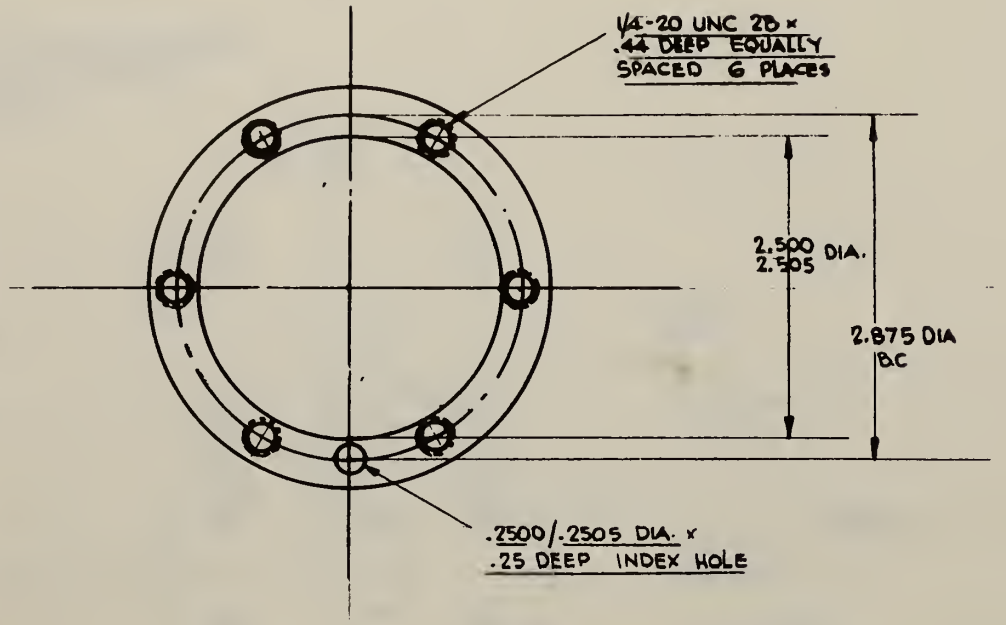


55#

ORIGINAL DATE OF DRAWING			
REVISIONS			
NO.	E. C. N.	CHANGE	DATE
1			
2			
3			
4			

PIECE NO.	NOMENCLATURE	
NATIONAL BUREAU OF STANDARDS WASHINGTON, D.C. 20234		
MOUNTING PLATE-WRIST 6° ST'D		
FOR UNIMATION INC. 2000 SERIES		
MODEL	TYPE	SCALE full
DIMENSIONS IN INCHES (Unless otherwise specified) TOLERANCES (Unless otherwise specified) DECIMALS ±.005 FRACTIONS ±.015 ANGLES ±½°	DRAFTSMAN CJ 11-25-80 PROJECT ENGR.	CHECKED PROJECT ENGR.
DO NOT SCALE THIS PRINT	SUBMITTED BY CHIEF, DIV.	
	EXAMINED BY CHIEF ENGINEER	
DIV. SEC. 738 .04	THIS PRINT ISSUED	APPROVED BY CHIEF, DIV.
		738-12361-B

☆ U.S. GOVERNMENT PRINTING OFFICE: GPO# 299-842



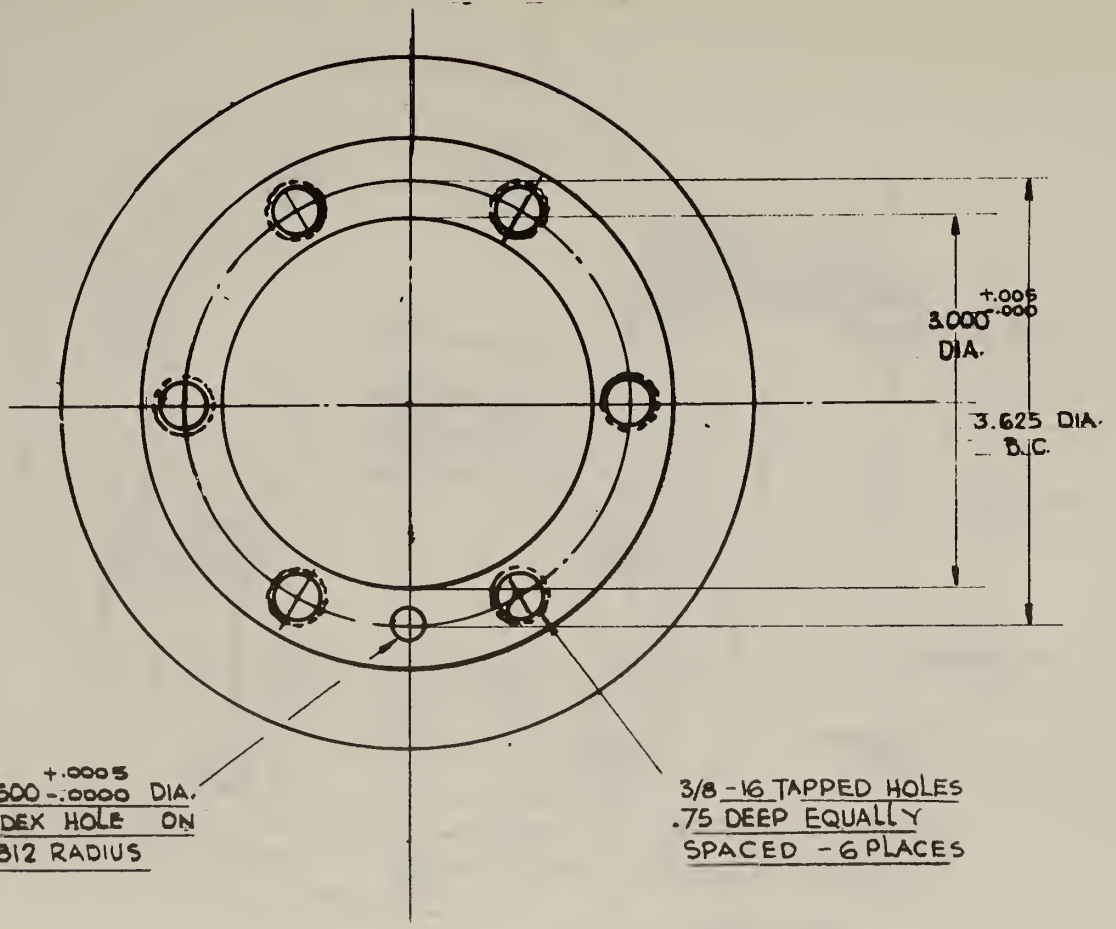
100*

ORIGINAL DATE OF DRAWING			
REVISIONS			
NO.	E. C. N.	CHANGE	DATE
1			
2			
3			
4			

PIECE NO.	NOMENCLATURE	NO. REQ'D
NATIONAL BUREAU OF STANDARDS WASHINGTON, D.C. 20234		
MOUNTING PLATE-WRIST 5° HVY DUTY		
FOR UNIMATION INC. 2000 SERIES		

MODEL	TYPE	SCALE Full
DIMENSIONS IN INCHES (Unless otherwise specified) TOLERANCES (Unless otherwise specified) DECIMALS ±.000 FRACTIONS ±.015 ANGLES ±15' DO NOT SCALE THIS PRINT	DRAFTSMAN 4-15-80 ejh PROJECT ENGR.	MATERIAL PROJECT ENGR.
	SUBMITTED BY GROUP, SEC.	
	DRAWING BY CHIEF ENGINEER	
DIV. SEC. 738 .04	THIS PRINT ISSUED	APPROVED BY GROUP, DIV.
		738-12358-B

★ U.S. GOVERNMENT PRINTING OFFICE: 1979 299-042

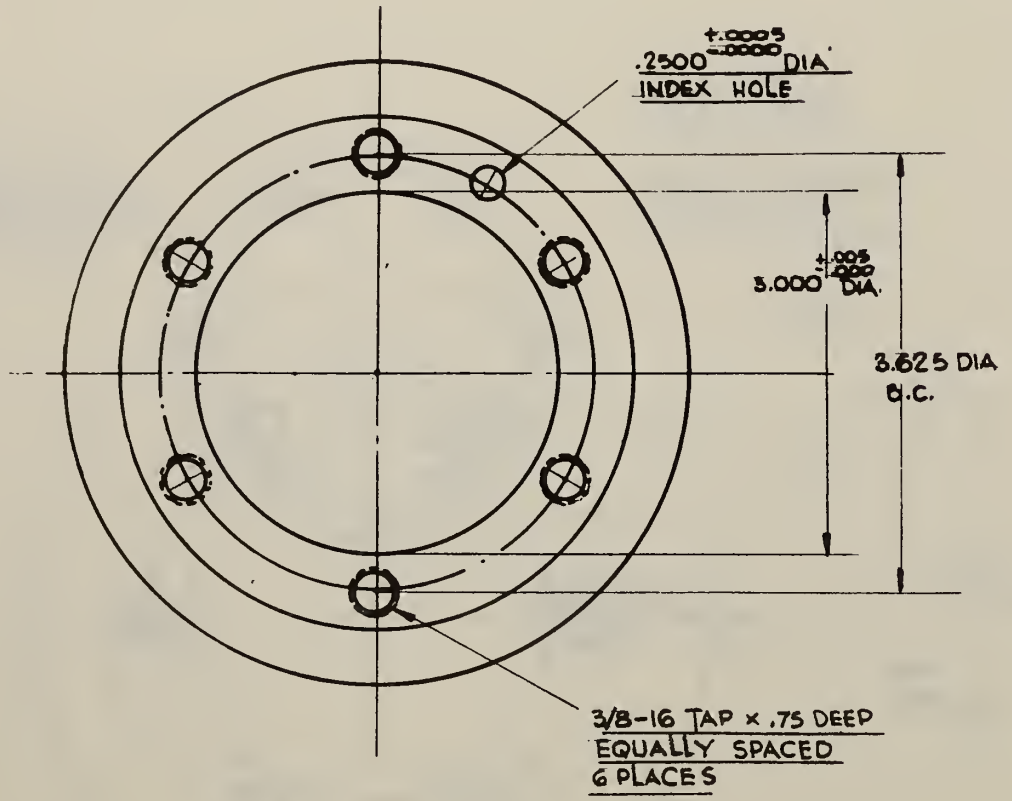


ORIGINAL DATE OF DRAWING			
REVISIONS			
NO.	E C N.	CHANGE	DATE
1			
2			
3			
4			

PIECE NO.	NOMENCLATURE	NO. REQ'D
NATIONAL BUREAU OF STANDARDS WASHINGTON, D.C.		
MOUNTING PLATE		
FOR UNIMATION INC. 4006 B SERIES		

MODEL	TYPE	SCALE FULL
DIMENSIONS IN INCHES (Unless otherwise specified)	DRAFTSMAN T2-1-80-C.L.h.	REVISION
TOLERANCES (Unless otherwise specified)	PROJECT ENGR.	PROJECT ENGR.
DECIMALS ±.005 FRACTIONS ±.010 ANGLES ±½°	SUBMITTED BY GROUP, SEC.	
DO NOT SCALE THIS PRINT	EXAMINED BY GROUP ENGINEER	
DEV. SEC. 738 .04	THIS PRINT ISSUED	APPROVED BY GROUP, DIV.
		738-12360-B

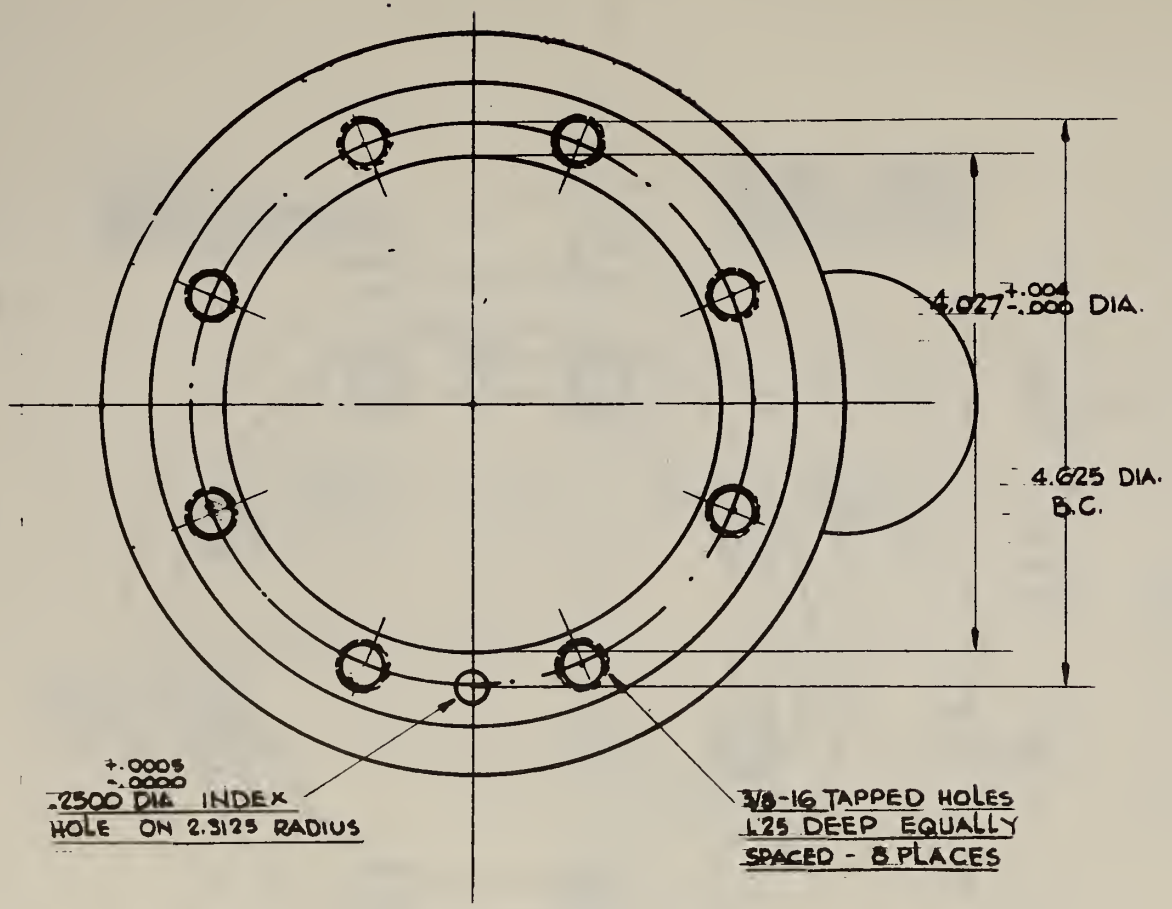
☆ U.S. GOVERNMENT PRINTING OFFICE: GPO# 799-842



ORIGINAL DATE OF DRAWING			
REVISIONS			
NO.	E. C. N.	CHANGE	DATE
1			
2			
3			
4			

PIECE NO.	NOMENCLATURE	
NATIONAL BUREAU OF STANDARDS WASHINGTON, D.C. 20234		
MOUNTING PLATE		
FOR UNIMATION INC. FOR 4000B.C.		
MODEL	TYPE	SCALE FULL
DIMENSIONS IN INCHES (Unless otherwise specified)	DRAFTSMAN CJ b	CHECKED
TOLERANCES (Unless otherwise specified)	PROJECT ENGR.	PROJECT ENGR.
DECIMALS ±.000 FRACTIONS ±.010 ANGLES ±½°	SUBMITTED BY GROUP, SEC.	
DO NOT SCALE THIS PRINT	EXAMINED BY GROUP ENGINEER	
GRP. SEC. 738 .04	THIS PRINT ISSUED	APPROVED BY GROUP, DIV.
		738-12366B

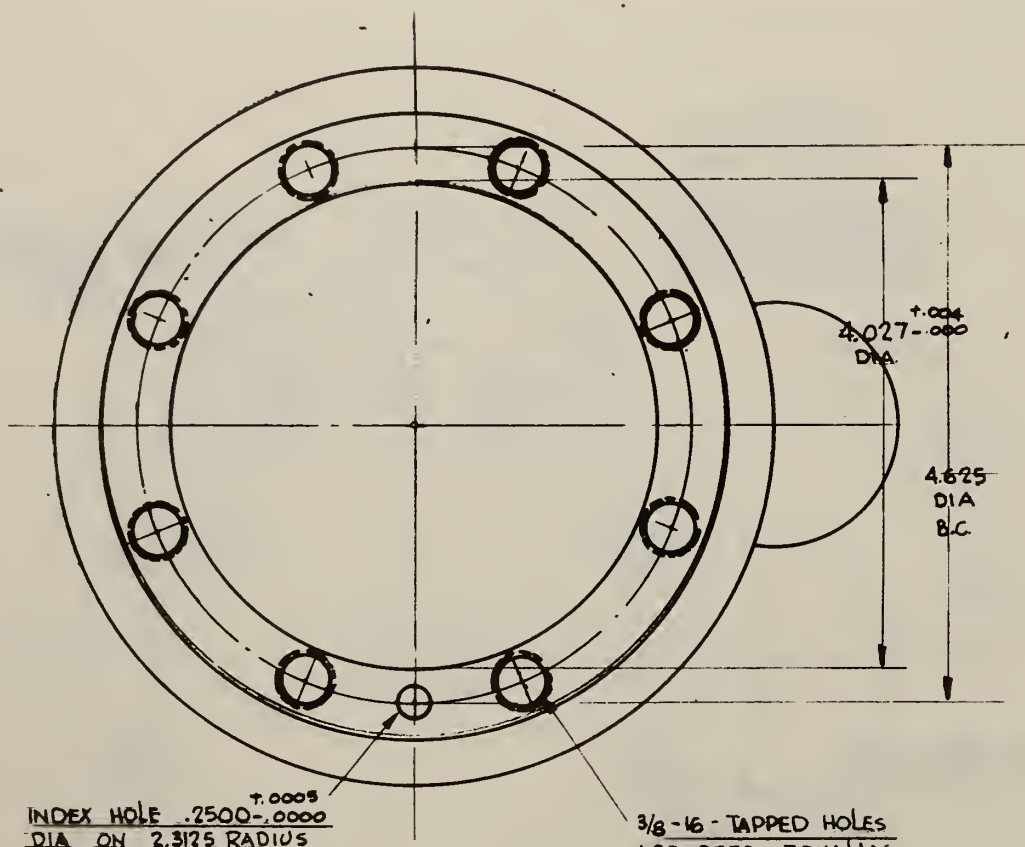
☆ U.S. GOVERNMENT PRINTING OFFICE : GPO# 299-842



ORIGINAL DATE OF DRAWING			
REVISIONS			
NO.	E. C. N.	CHANGE	DATE
1			
2			
3			
4			

PIECE NO.	NOMENCLATURE	
NATIONAL BUREAU OF STANDARDS WASHINGTON, D. C. 20234		
MOUNTING PLATE		
FOR UNIMATION INC.- 4005 D & 4005 E		
MODEL	TYPE	SCALE FULL
DIMENSIONS IN INCHES (Unless otherwise specified) TOLERANCES (Unless otherwise specified) DECIMALS ±.005 FRACTIONS ±1/64 ANGLES ±½° DO NOT SCALE THIS PRINT	DRAFTSMAN 12-4-85 C.J.L. PROJECT ENGR.	CHECKED PROJECT ENGR.
	SUBMITTED BY CHIEF, SEC.	
	EXAMINED BY CHIEF SECTION	
DEV. SEC. 738.04	THIS PRINT ISSUED	APPROVED BY CHIEF, DIV.
		738-12365-B

★ U.S. GOVERNMENT PRINTING OFFICE: 1970 O-399-842



ORIGINAL DATE OF DRAWING			
REVISIONS			
NO.	E. C. N.	CHANGE	DATE
1			
2			
3			
4			

PIECE NO.	NOMENCLATURE	NO. REQ'D

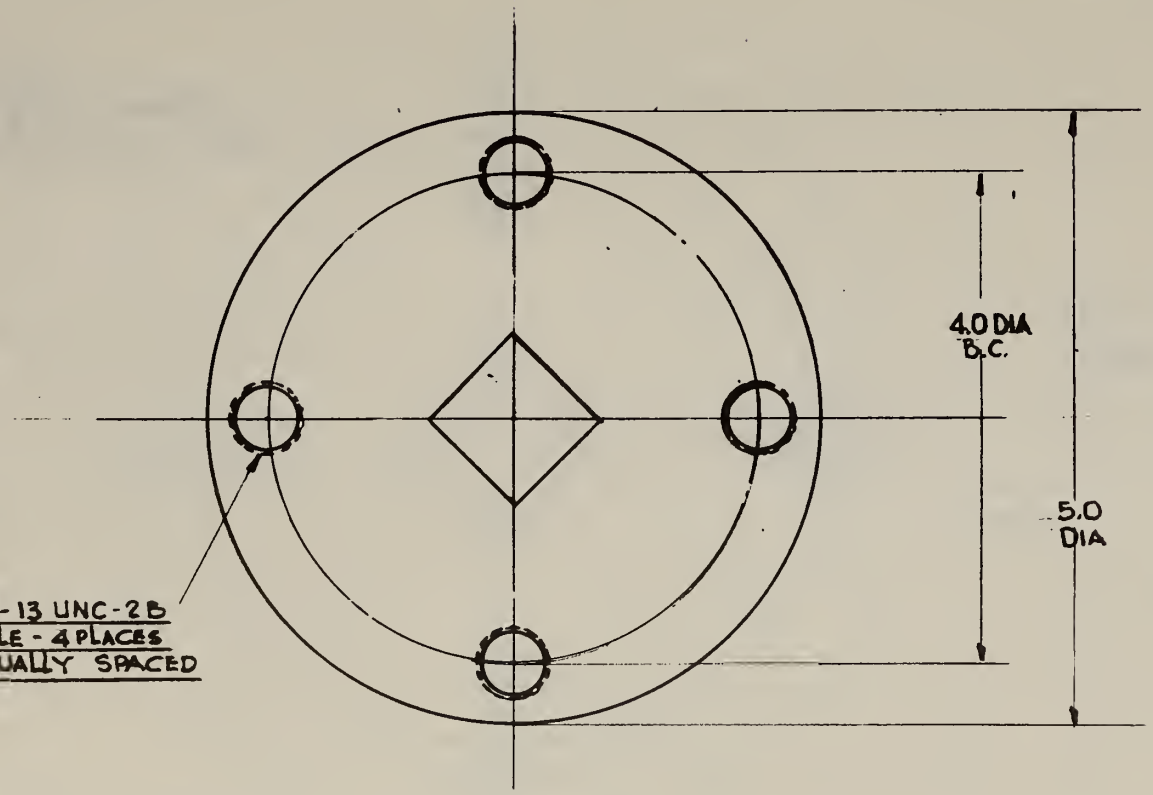
NATIONAL BUREAU OF STANDARDS
WASHINGTON, D. C.

MOUNTING PLATE

FOR UNIMATION INC-FOR-400 & 400 SJ

MODEL	TYPE	SCALE FULL
DIMENSIONS IN INCHES (Unless otherwise specified)	DRAFTSMAN 12-1-80 C.J.b	CHECKED
TOLERANCES (Unless otherwise specified)	PRELIMINARY CHECK.	PRELIMINARY CHECK.
DECIMALS ±.005 FRACTIONS ±.015 ANGLES ±15'	SUBMITTED BY GROUP, SEC.	
DO NOT SCALE THIS PRINT	EXAMINED BY GROUP CHIEF	
DIV. SEC. 738 .04	THIS PRINT ISSUED	APPROVED BY GROUP, DIV.
		738-12359-B

☆ U.S. GOVERNMENT PRINTING OFFICE: GPO# 299-042

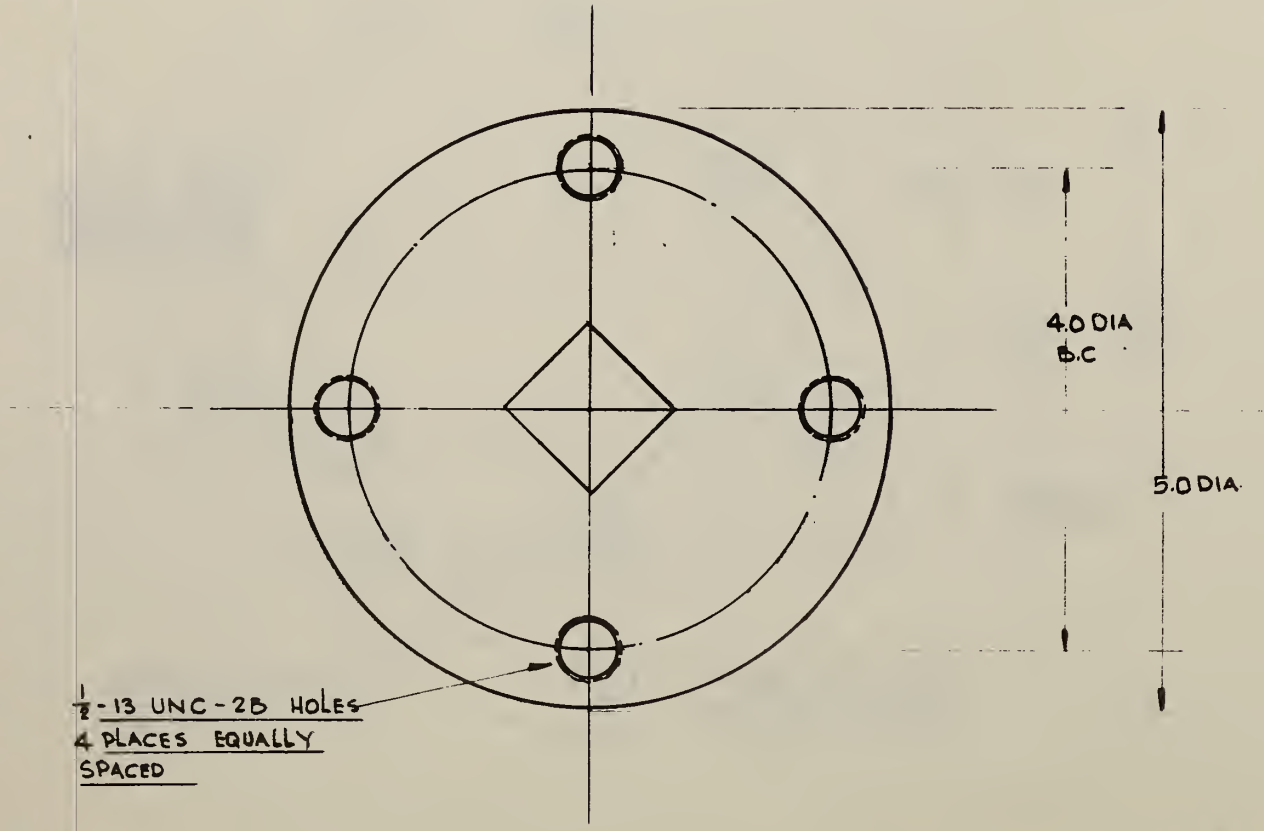


250*

ORIGINAL DATE OF DRAWING			
REVISIONS			
NO.	E. C. N.	CHANGE	DATE
1			
2			
3			
4			

PIECE NO.	NOMENCLATURE	NO. REQ'D.
NATIONAL BUREAU OF STANDARDS WASHINGTON, D.C. 20234		
MODEL "FAHYS" TOOLING MTG. PLATES		
FOR A.M.F. VERSATRAN		
MODEL	TYPE	SCALE FULL
DIMENSIONS IN INCHES (Unless otherwise specified)	DRAFTSMAN K-18-80-C.J.H.	CHECKED
TOLERANCES (Unless otherwise specified)	PROJECT ENGR.	PRODUCT ENGR.
DECIMALS ±.005 FRACTIONS ±.010 ANGLES ±½°	SUBMITTED BY CHIEF, DIV.	
DO NOT SCALE THIS PRINT	ENGINEER BY CHIEF ENGINEER	
DEV. SEC. 738 .04	THIS PRINT ISSUED	APPROVED BY CHIEF, DIV.
		738-12356-B

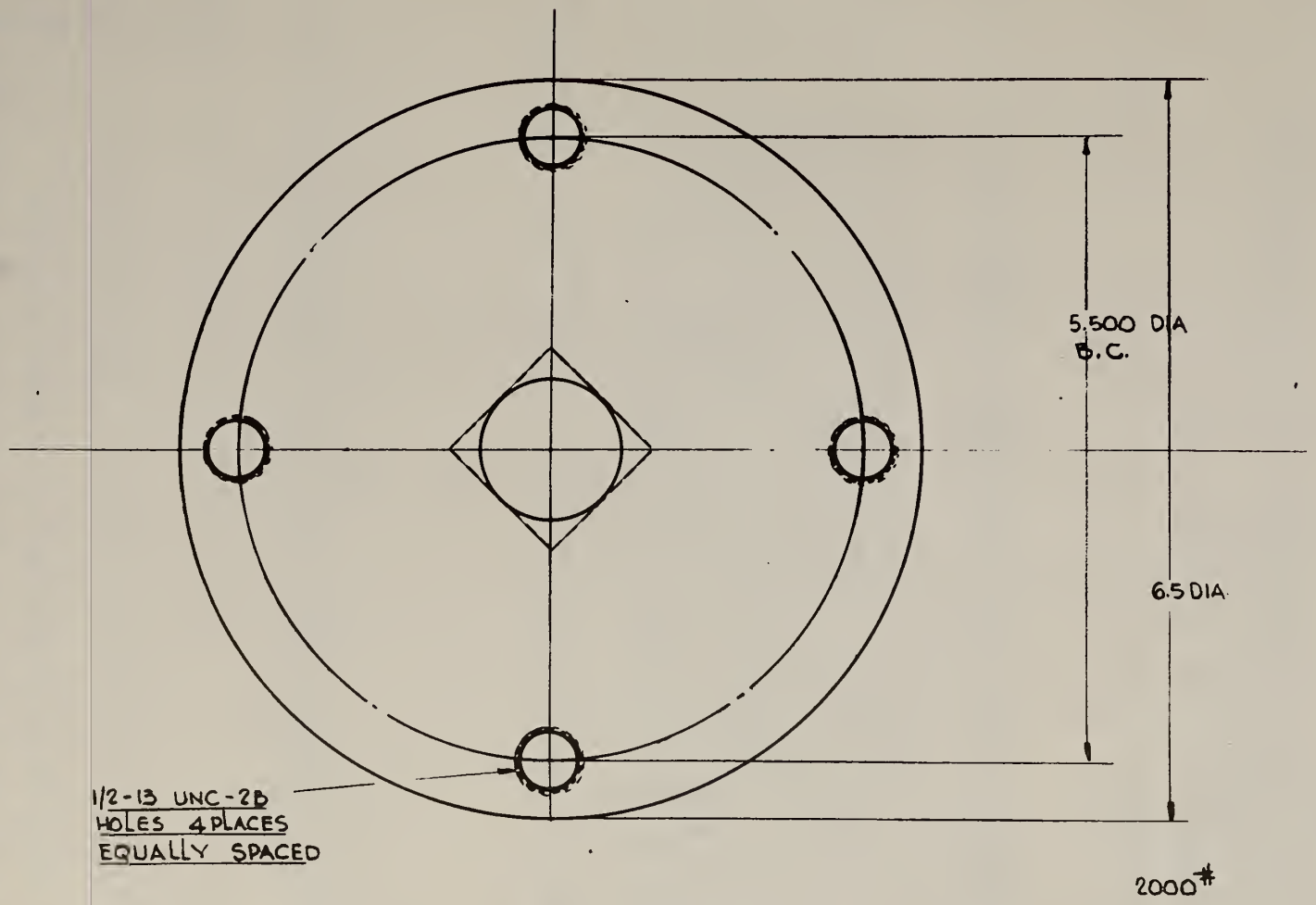
* U.S. GOVERNMENT PRINTING OFFICE: 0P06 299-842



ORIGINAL DATE OF DRAWING			
REVISIONS			
NO.	E. C. N.	CHANGE	DATE
1			
2			
3			
4			

PIECE NO.	NOMENCLATURE	NO. REQ'D
NATIONAL BUREAU OF STANDARDS WASHINGTON, D.C. 20234		
MODEL "FBHVS" TOOLING MTG. FLANGE		
FOR AMF VERSATRAN		
MATERIAL	TYPE	SCALE FULL
DIMENSIONS IN INCHES (Unless otherwise specified) TOLERANCES (Unless otherwise specified) DECIMALS ±.005 FRACTIONS ±1/64 ANGLES ±½°	DRAFTSMAN 11-15-60 C.J.L PROJECT ENGR.	CHECKED PROJECT ENGR.
DO NOT SCALE THIS PRINT	SUBMITTED BY CHIEF, SEC.	
	EXAMINED BY CHIEF ENGINEER	
DIV. SEC. 738 .04	THIS PRINT ISSUED	APPROVED BY CHIEF, DIV.
		738-12355-B

☆ U.S. GOVERNMENT PRINTING OFFICE: GPO # 299-842



ORIGINAL DATE OF DRAWING			
REVISIONS			
NO.	E. C. N.	CHANGE	DATE
1			
2			
3			
4			

PIECE NO.	NOMENCLATURE	
NATIONAL BUREAU OF STANDARDS WASHINGTON, D.C. 20234		
MODEL "FCHVS" TOOLING MTR FLANGE		
FOR A.M.F. VERSATRAN		
MODEL	TYPE	SCALE FULL
DIMENSIONS IN INCHES (Unless otherwise specified) TOLERANCES (Unless otherwise specified) DECIMALS ±.005 FRACTIONS ±.015 ANGLES ±½°	DRAFTSMAN 11-19-80 C.J.h. PROJECT ENGR.	ENGINEER PROJECT ENGR.
DO NOT SCALE THIS PRINT	SUBMITTED BY CHIEF, SEC.	
	EXAMINED BY CHIEF ENGINEER	
DIV. SEC. 738 .04	THIS PRINT ISSUED	APPROVED BY CHIEF, DIV.
		738-12354-B

☆ U.S. GOVERNMENT PRINTING OFFICE: GPO# 299-842

NBS-114A (REV. 2-8C)

U.S. DEPT. OF COMM. BIBLIOGRAPHIC DATA SHEET (See instructions)	1. PUBLICATION OR REPORT NO. NBSIR 80-2152	2. Performing Organ. Report No.	3. Publication Date

4. TITLE AND SUBTITLE

NBS/ICAM WORKSHOP ON ROBOT INTERFACES

June 4-6, 1980

5. AUTHOR(S) EDITORS:

Thomas Wheatley, James S. Albus and Roger N. Nagel

6. PERFORMING ORGANIZATION (If joint or other than NBS, see instructions)

NATIONAL BUREAU OF STANDARDS
DEPARTMENT OF COMMERCE
WASHINGTON, D.C. 20234

7. Contract/Grant No.

MIPR SY1455-78-00003

8. Type of Report & Period Covered

Final Report

9. SPONSORING ORGANIZATION NAME AND COMPLETE ADDRESS (Street, City, State, ZIP)

The National Bureau of Standards and
the Air Force ICAM Project, Air Force Systems Command
Wright-Patterson Air Force Base, Ohio 45433

10. SUPPLEMENTARY NOTES

☐ Document describes a computer program; SF-185, FIPS Software Summary, is attached.

11. ABSTRACT (A 200-word or less factual summary of most significant information. If document includes a significant bibliography or literature survey, mention it here)

This paper will describe the proceedings of a workshop on robot interfaces held at NBS from June 4-6, 1980. Five possible areas for standardization of interfaces were discussed: (1) Simple Sensors, the interface between simple peripheral devices and a robot control system; (2) the Wrist Interface, between the robot wrist and the end effector; (3) the Common Robot Control Interface, between the robot controller and the coordinate transformation module; (4) the Complex Sensory Interface, between the robot controller and vision, complex touch, and other sensors; and (5) Future Guidelines, covering data base, offline programming, and system integration interfaces. The goal was to define the areas ready for current standards, and those for which guidelines for future standards would be useful.

This workshop was jointly sponsored by NBS and the Air Force ICAM project.

12. KEY WORDS (Six to twelve entries; alphabetical order; capitalize only proper names; and separate key words by semicolons)

Complex Sensors; off-line programming; robot control; robot interfaces; robot standards; robot wrist; simple sensors

13. AVAILABILITY

☒ Unlimited

☐ For Official Distribution. Do Not Release to NTIS

☐ Order From Superintendent of Documents, U.S. Government Printing Office, Washington, D.C. 20402.

☐ Order From National Technical Information Service (NTIS), Springfield, VA. 22161

14. NO. OF PRINTED PAGES

15. Price