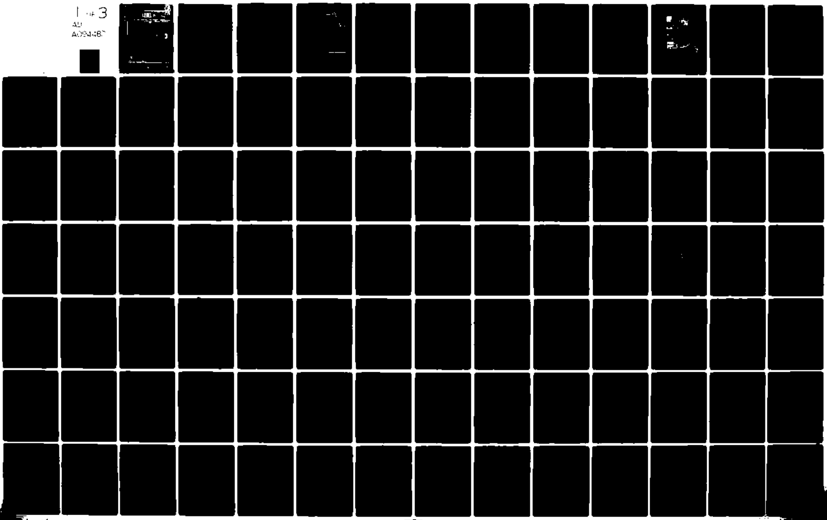
AD-A094 482 PERCEPTRONICS INC WOODLAND HILLS CALIF F/G 5/8
MAN-MACHINE COMMUNICATION IN REMOTE MANIPULATION: TASK-ORIENTED--ETC(U)
MAR 80 Y CHU, W H CROOKS, A FREEDY N00014-76-C-0603
JNCLASSIFIED PFTR-1034-80-3 NL

1 of 3
AD
A094482



PERCEPTRONICS

The views and conclusions [illegible]
are those of the author [illegible]
[illegible]
[illegible]
[illegible]

[illegible]
[illegible]

Final Technical Report PFTR-1034-80-3
Contract N00014-76-C-0603
NR 196-140
March, 1980

MAN-MACHINE COMMUNICATION IN REMOTE MANIPULATION: TASK-ORIENTED SUPERVISORY COMMAND LANGUAGE (TOSC)

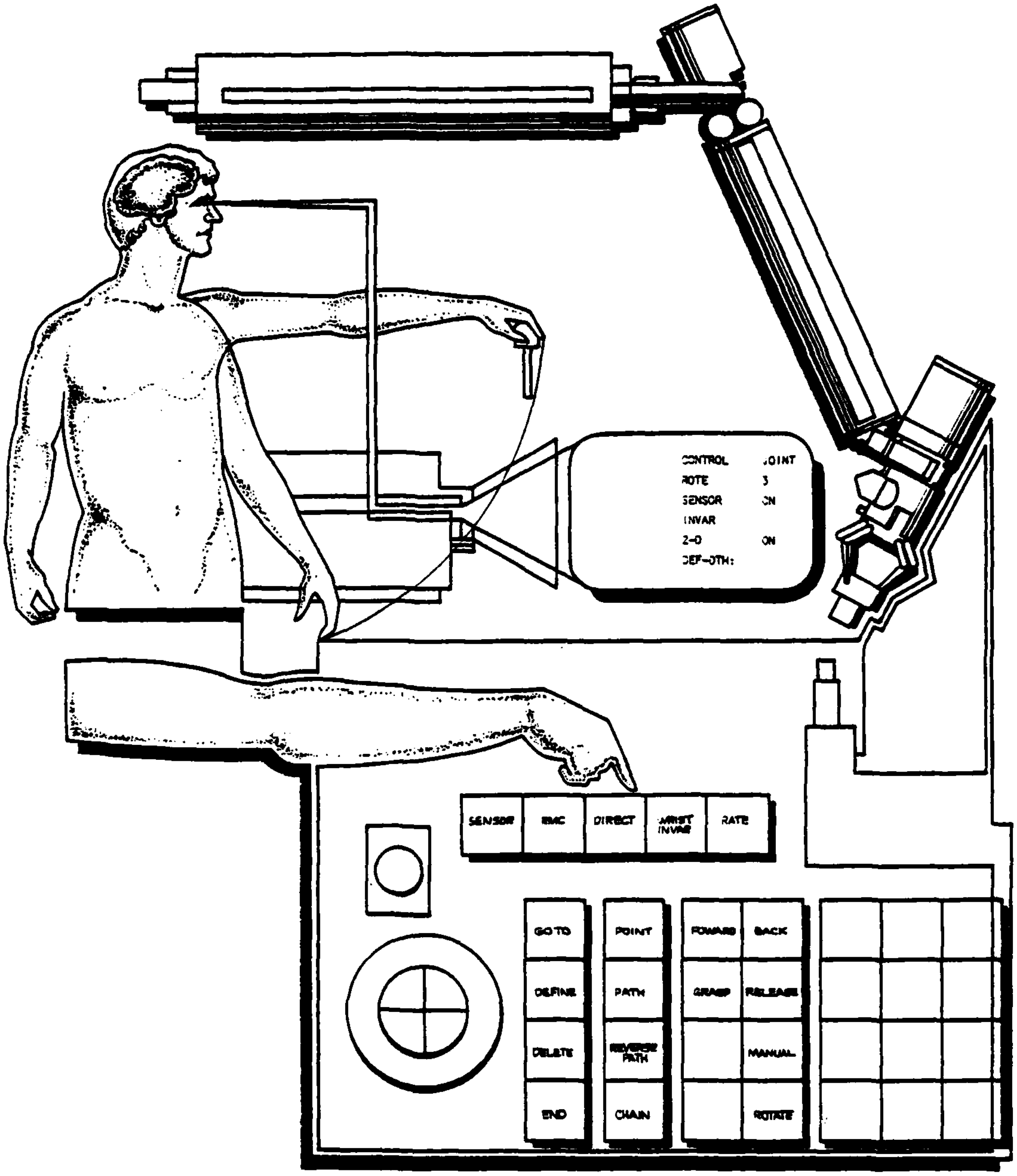
Yee-Yeen Chu
William H. Crooks
Amos Freedy

Prepared For:

Engineering Psychology Programs (code 455)
OFFICE OF NAVAL RESEARCH
Department of the Navy
Arlington, VA 22217

PERCEPTRONICS

6271 VARIEL AVENUE • WOODLAND HILLS • CALIFORNIA 91367 • PHONE (213) 884-7470



SHARED MAN COMPUTER CONTROL

ACKNOWLEDGEMENTS

In addition to the authors, contributions to this research effort were made by the following Perceptronics personnel: Dr. Efraim Shaket, Senior Scientist; Dr. Gershon Weltman, President and Chief Scientist; Mr. Johathan Klein, Research Associate; Mr. Yoram Alperovitch, Associate Software Engineer; Mr. Michael Rector, Associate Scientist, and Mr. Denis Purcell, Senior System Engineer.

The authors extend their appreciation to Mr. Gerald Malecki of the Office of Naval Research for his encouragement and counsel in the conduct of this research.

TABLE OF CONTENTS

TABLE OF CONTENTS (CONTINUED)

TABLE OF CONTENTS (CONTINUED)

1. INTRODUCTION

1.1 Summary

This report covers ~~the fourth year of a four-year program of~~ research and development directed toward the investigation and optimization of man-machine communication in computer-aided remote manipulation. The purpose of this program was to determine through analytical and experimental studies the relationships between primary man-machine communication factors and system performance, and to develop and demonstrate a communication design methodology to improve operator performance with remotely controlled systems.

Specific ~~program~~ objectives included the following:

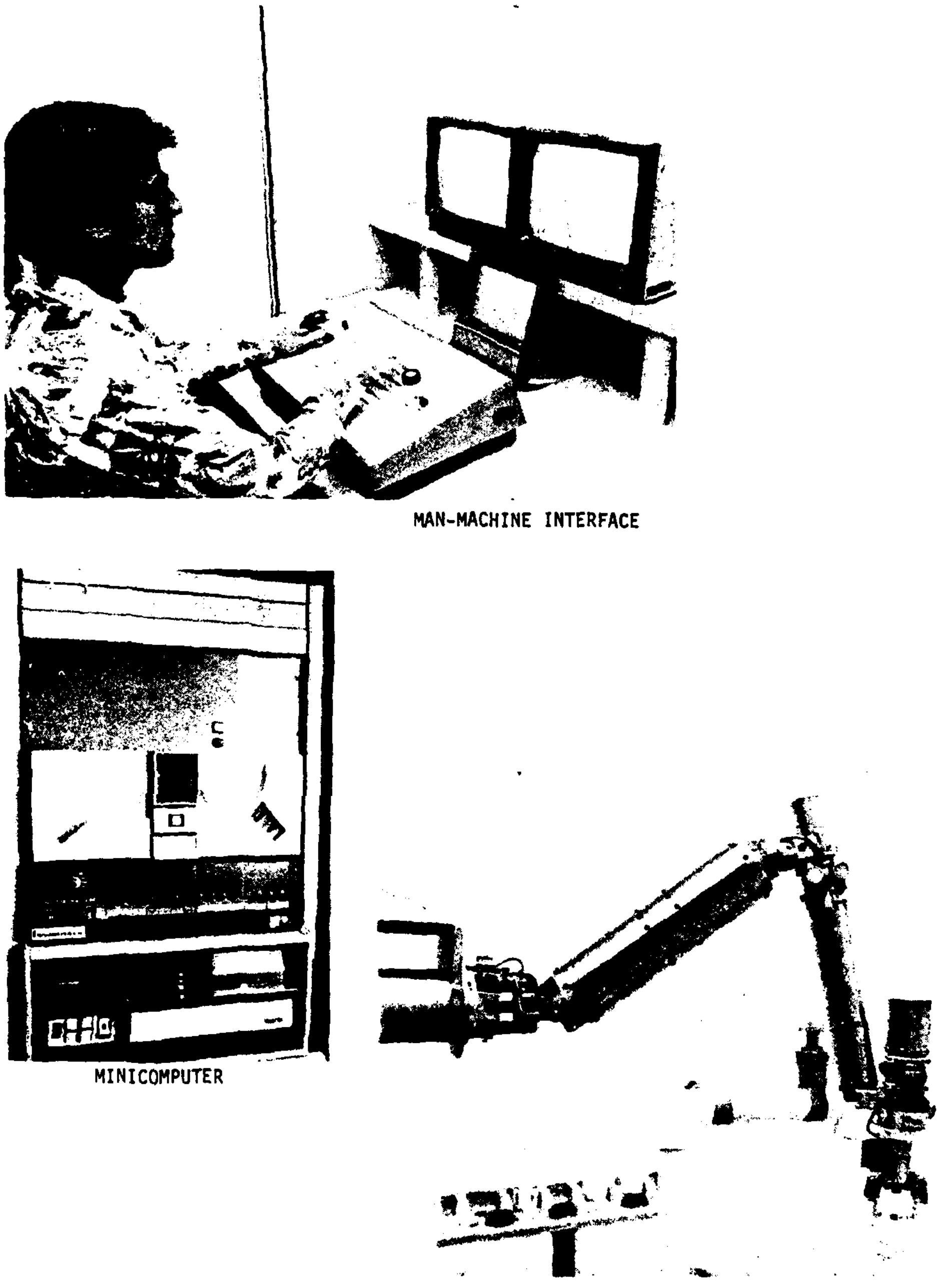
(1) To perform an analysis of communications requirements in computer-aided manipulation and closely related areas of adaptive and autonomous control,

(2) To establish an experimental system for study of task-oriented supervisory control of a remote manipulator, (Figure 1-1).

(3) To implement and evaluate communication systems, encompassing both language and interface, designed to permit natural and efficient control of a variety of remote manipulation tasks,

(4) To identify the primary factors influencing the success of shared man-computer control, and to establish quantitative relationships between these factors and the system performance measures, and

(5) To provide guidelines for the design of man-computer communication in subsequent autonomous, adaptive and remotely-manned systems.



MAN-MACHINE INTERFACE

MINICOMPUTER

SERVO-MANIPULATOR

FIGURE 1-1.
COMPUTER-AIDED REMOTE MANIPULATION
EXPERIMENTAL FACILITY

The initial year's work established a theoretical communications framework based on procedural nets, and examined experimentally the separate effects of several basic computer aiding techniques on the ability of trained operators to perform selected manipulation tasks. The experimental results indicated that computer aiding, in the form of real-time transformation from joint angle to resolved motion control (RMC) of the end point, significantly reduces the time required and the number of errors committed in performing most manipulation tasks. Computer aiding in the form of automatic motion control (AMC) to specifiable locations did not provide immediate performance improvement but showed potential usefulness for some well-specified manipulation tasks. The training results also suggested that computer aiding can be used to reduce the time required for personnel to become accomplished manipulator operators.

The second year's study emphasized the development of the procedural net model into a language model. This led to the development of a specific manipulation language and the corresponding control/display design concepts. The language structure, through the definition and execution of symbolic commands, provides the user with a flexible mechanism to define and use task oriented commands. The experimental study indicated that these pre-defined variable commands can significantly reduce task time to perform a repetitive task. These data also suggested that the benefit of variable commands increases with practice and that the organization of the commands helps to reduce the number of errors in a complex task.

The third year's study provided the extension and evaluation of the high level command structure and the needed feedback information to the operator regarding task state and system status. The experimental results indicated that the high level construct of a command can significantly improve task performance when tasks are discrete with low trajectory complexity, or when tasks are operated under degraded visibility. The study also suggested that a command-queue display with appropriate format and update rate can serve as both memory and task monitor aids.

The work reported here comprises the fourth year of the program and consolidates the results of current and previous studies in determining the relationship between man-machine communications factors and operator performance.

First presented, is a taxonomy of shared man-computer control in underwater manipulation developed to expand the domain of test results. Relative evaluations of the manned system performance level are discussed next. These evaluations are determined as functions of primary communications factors, including command structure, control mode, and feedback mode. The work ends with the presentation of a performance prediction model and a set of principles and guidelines, applicable to the design of man-machine interface, particularly the structure of multi-mode command inputs, apportionment of control functions between operator and computer, and methods for structuring feedback information to the operator.

1.2 Technical Approach

1.2.1 Computer-Aided Manipulation. Advances in computer-aided teleoperator control offer the potential for substantially improving the effectiveness of Navy underwater manipulator systems. Computer-aided control can be used, with reasonable cost, to improve system effectiveness by performing coordinate transformations to simplify simultaneous joint movements and by reducing operator task loading by allowing the operator to allocate certain task elements to machine automation. As we extend manual control of the manipulator system to the full range capability afforded by the computer element, the problems of man-machine communication become of great importance. An effective, closely coupled man-computer communication interface is needed because the task environment requires great flexibility and dexterity in planning and operation. The sparse evidence suggests that when this communication is awkward, computer-aided control of remote manipulation can be inferior to manual control. What is needed, then, is a systematic investigation of critical communication factors, and a new method for extracting and organizing task related information communicated between human operator and computer to implement efficient control systems.

As a major step toward the identification of critical communication factors, observations need to be made concerning the major dimensions of operator activities with a given task. In most manipulation tasks, the operator must observe the task environment, make judgments about the commands necessary to perform the task, and carry out the command execution while maintaining observation of the manipulator within the task environment. As the situation varies, the sequence and the complexity of the above operator activities may vary. In addition, the activities and, more fundamentally, the role of the operator could be changed with the introduction of computer aiding techniques. A typical example is the use of a set of sampled aiding techniques organized in a shared man-computer control framework. In such a case, which was examined in this program, the operator not only provides direct analog control of the manipulator's movements, but also he must (1) organize command sequences, (2) select any of a number of computer-assisted functions, (3) monitor the mode of operation and the progress of automated routines, and (4) be able to resume and dispatch execution functions. A wide range of communication modes, encompassing a wide spectrum of augmented and autonomous manipulation, is therefore required for effective use of computers in remote manipulation.

Accordingly, the approach in the present program was to focus on the general rules for constructing special-purpose communication languages for effective shared man-computer control. The use of remote manipulation represents a good example of a bounded (limited-bandwidth) communications area, one which is important in its own right to Navy operational goals. Language elements have been implemented at the man-computer interface. The relationship of variation in these elements to total system performance has provided the data upon which practical human factors design guidelines will be based.

1.2.2 Command Language Development. The p·evious year's work included the development of a communication model which forms the basis for the man-machine language design. A hierarchical model based on the concept of procedural net was developed to represent the planning process of manipulator action. The procedural net is a conceptual framework that models the process of plan development. The problem domain is described as a hierarchy of tasks and subtasks at various levels of abstraction. Each task node consists of a goal statement of what has to be accomplished by the task, an object on which the action is performed, and an action--the sequence of subtasks expressed at a lower level of abstraction. The specific sequence needed to accomplish a task is a function of both the state of the environment and what is requested at higher levels of the global task.

The procedural net model described above was adapted to represent the man-machine communication process as a step in the overall planning process. That is, the human operator develops the global task into a plan at some intermediate level of detail which he then communicates to the computer. The computer develops the plan further to the level of detail necessary for controlling the manipulator and proceeds to monitor the plan execution via position and force-sensor feedback. The language developed from this model is hierarchical, flexible, and task-oriented, allowing close cooperation between man and machine with a mixed initiative protocol of control allocation. Using this model, a task-oriented supervisory control (TOSC) language was designed which facilitates computer-aided manipulation. The main features of the language include the following:

(1) User-defined, hierarchical-structured commands.

(2) Command chains as concepts at various task levels.

(3) Sentence-structured keyboard command.

(4) Mixed-initiative control.

(5) Intermixing of analogic and symbolic, complex and simple commands.

(6) Machine state and command queue feedback organized around task hierarchy with the level of detail in correspondence with the operator's planning level.

1.2.3 Experimental Program. The objective of the overall experimental program was to evaluate techniques for improved man-machine communication in computer-aided remote manipulation. The major dimensions under investigation were:

(1) Command Language Structure. The manner by which the operator transmits commands to the remote element. The levels of this dimension are (1) Manual Commands, whereby the operator actuates analogic or symbolic control motions, (2) Variable Commands, in which the operator uses defined points, paths or the linkage of individual symbolic commands, (3) Chained (Automatic) Commands, in which the operator defines a multilevel sequence of commands which can then be performed automatically.

(2) Machine State Feedback. The information concerning the control mode and command queue established for the chained command mode of control. The levels of this dimension are (1) full delineation of the individual commands and their sequence, and (2) display of the currently operational command only.

(3) Visual Feedback. Visual observation of the manipulator in work space. The levels of this dimension are: (1) normal TV viewing, (2) degraded TV, and (3) stick-figure graphic display.

The experimental studies were conducted using the Perceptronics' facility for computer-controlled manipulation. This facility, developed and programmed in the previous years of the program, provides the necessary capabilities for testing language features, input/output protocol and feedback display levels. The manipulator itself is a dexterous, hydraulically-powered unit, combining quick response accuracy and high strength. The manipulator is commanded through a dedicated keyboard and joysticks, designed according to the command language guidelines, along with a CRT display for feedback of machine state information, two-view video display and a 3D stick-figure graphic to simulate remote operations.

1.3 Report Organization

The organization of this report is as follows: Chapter 2 describes the development of the taxonomy for shared man-computer control of teleoperation, emphasizing underwater manipulation. Chapter 3 presents the developed task-oriented supervisory command (TOSC) language and the system design. Chapter 4 summarizes the results of the experimental studies and the implications of the research findings. Chapter 5 generalizes the test results and discusses the issues in manned system evaluation and performance model development.

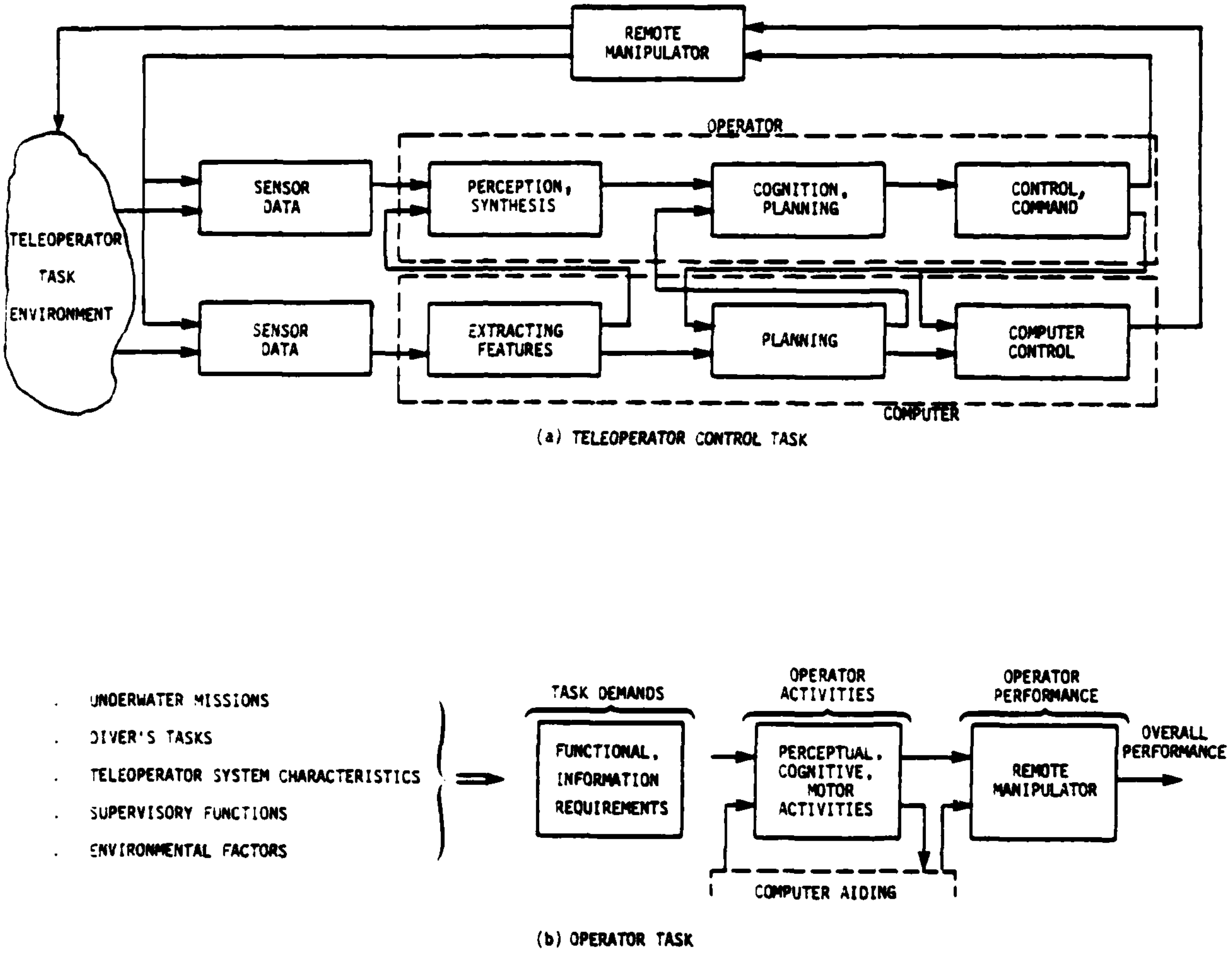
2. UNDERWATER MANIPULATION WITH SHARED MAN-COMPUTER CONTROL

2.1 Overview

This chapter presents the results of a taxonomical analysis of underwater manipulation that is suitable for a shared man-computer control approach. The taxonomy is descriptive in nature and identifies task dimensions suitable for characterizing the effects of man-machine communication factors over operator performance. It also provides a basis for generalizing the method and data achieved in our man-machine communication language design. Therefore, the emphasis here has not been to develop a rigorous classification scheme, but rather to provide a framework for evaluating and extending the experimental findings in terms of their application to various types of tasks. As such, the procedure of the study included a brief literature review, technical discussions with members in the related research areas, analyses of task attributes and requirements, and cross-examination of experimental findings. The topics discussed in the following sections represent those areas closely related to man-machine communications: low-level manipulation, environmental factors, machine characteristics (automation levels), and man-machine interaction factors.

2.2 Teleoperator Task Identification

A task can be viewed as the totality of the situation imposed on the operator or teleoperator. Here, teleoperator is referred to as "a general-purpose, dexterous, man-machine system that augments the operator by projecting his manipulatory capability across distance and through physical barriers" (Corliss and Johnsen, 1968). Figure 2-1a illustrates a functional representation of a general teleoperator system. Top-level functional descriptions of the subprocesses and the interactions between them are presented in the context of a remote manipulation environment. The major subprocesses of operator activities include:



(a) TELEOPERATOR CONTROL TASK

(b) OPERATOR TASK

FIGURE 2-1.
TELEOPERATOR TASK ANALYSIS

(1) Perception: including event sensing, which continously senses and manages data flow; and data synthesis, which updates situation estimates.

(2) Cognition: including problem recognition, which identifies conflicts or problems; and planning, which synthesizes and refines command/actions.

(3) Execution: including plan selection, which provides trade-offs between procedures; and command execution, which trades between activation and monitoring functions.

It appears that the operator's task, which is the main concern of this study, is determined by the teleoperator task demand, the environmental factors, and the specific teleoperator system given (i.e., the types of manipulator, tools, sensors, and computer-aiding available). Therefore, along with the discussions contained in the next few sections, a major effort will be devoted to map and build the operator task taxonomy out of the existing data sources of teleoperator task taxonomies. The "demand" side of task analysis, as described in Figure 2-1b, starts with the collection of underwater manipulation tasks based on the following sources:

(1) Navy underwater missions and commercial ocean operations.

(2) Diver's tasks.

(3) Underwater environmental factors.

(4) Teleoperator systems (including manipulators, tools, sensors).

(5) Supervisory control approaches (computer aiding).

(6) Planning and cognitive activities of everyday manipulation.

The "performance" side of task analysis will be expanded by a set of performance criteria. The detail of the performance study will be discussed in Chapter 5, while the following section concentrates on demand and activity analysis.

In the following sections the term "task" will refer to either teleoperator task or operator task, dependingon the context.

2.2.1 Methods of Specifying Teleoperator Control Tasks. Consider the teleoperator systems which are used to carry out some desired interaction with the underwater environment. Since it is impossible to abstractly model the complete physical environment, the abstract description of teleoperator-environments interaction is necessarily limited. Therefore, models that embody physical, geometrical, temporal and spatial characteristics are employed as components to describe an ongoing manipulation situation. Various approaches for task analysis were summarized by Sheridan and Verplank (1978), including task breakdown (e.g., Bien and McDonough, 1968), functional and information requirement analysis (e.g., Schneider, 1977), time-line or time-precedence analysis (e.g., Pesch, et al., 1970; Ocean System, Inc., 1977), and formal process descriptions (e.g., Whitney, 1969). These approaches are used when they are suitable for task analysis at different levels of detail. In our approach to the demand-side analysis, a teleoperator task is viewed as a black box translating information and functional requirements into operator activities. The requirements associated with the spectrum of underwater mission were derived through the following method. First, the scope of mission and functional operations to be performed by the state-of-the-art teleoperator were determined, and typical operations were selected to focus the task analysis on uncovering a spectrum of manipulation functions. This analysis, together with the review of documents generated in the past, provided a summary of current and projected underwater tasks.

2.2.2 Task Description at Operation Level. A useful list of tasks a manipulator can perform in underwater missions was given by Drenning, and is presented in Table 2-1. To be more specific about the manipulative capability required, two functional operations listed in the Table were considered for more detailed analysis: offshore production and

TABLE 2-1

TYPES OF TASKS A MANIPULATOR CAN PERFORM ON UNDERWATER MISSIONS

Salvage

Detach cables restraining objects to be salvaged
Clear debris away from objects to be salvaged
Prepare objects for lifting by attaching cables
Position objectives for salvage
Separate large objects
Excavate bottom sediment

Undersea Rescue

Aid in freeing entrapped submersibles
Aid in mating of rescue submersible to submarine

Service Habitats

Aid in heavy work operation
Aid in replenishment of supplies
Aid in placement and recovery of habitats

Offshore Oil/Gas Production Facilities Task

Assist during drill string landing
Prepare drill sites by removing debris
Replace blowout preventer rams
Make pipe connections
Replace and patch pipes
Recover objects dropped from drill platform
Inspect oil lines using hand held acoustical devices
Remove marine growth

Others

Place and retrieve acoustic markers
Place explosive devices
Clear and remove debris
Collect marine samples
Position transponders
Remove and replace defective equipment
Take bottom core samples
Collect mineral laden nodules

Source: Drenning, undated

salvage. This selection expands a task spectrum from a fundamental underwater mission requirement to an evolving commercial application. This task spectrum also will include a number of elementary operations in current and future Navy mission requirements. The following list given by Talkington (1978) provides an example:

> Search - to find lost items, locate work sites, and survey seafloor areas.
>
> Inspection - to classify detected targets, monitor continuing operations, define the integrity of structural components or pipelines, detect leakage of pollutants, and record the condition of objects, e.g., ships, aircraft and canned waste, on the seafloor.
>
> Recovery - to attach lifting devices, cut away moorings or clutter, and provide vertical and horizontal lifting forces to effect transport of objects from the seafloor to the surface.
>
> Assembly, Modification, or Repair - to conduct work on objects on the seafloor or within the volume, assemble parts and effect repairs, improvements, or alterations.

The general classification given above also covers most of the manipulator functions in offshore operations. Actually, most of the underwater tasks associated with production operation stem from the possible extension of operational depths of free swimmer/divers and the increased cost of diver support in deep water. As a result, the concept of modular offshore services is presently being considered (Ocean Industry, 1978) to permit modular manipulation activities, including inspection, maintenance, repair and construction of offshore structures. The possible manipulation-related tasks in these activities are given in Table 2-2. The typical tasks listed under offshore service are extracts from a tentative application list of Gray and Fridge (1978) in their comparison study of diver alternative work systems.

In addition to those tasks identified for the selected offshore service and maintenance operations, a summary of underwater tasks was put together through a review of various sources (Pesh, et al., 1976; Bertsche, et al., 1978; Battele, 1976; Bien and McDonough, 1968, etc.). The studies reviewed were conducted to identify current and projected design requirements for diver's or machine's manipulation tools and to apply the findings to the study of underwater vehicle and instrument design. While some of the studies provided further breakdowns of underwater tasks, it appeared throughout this review that the task identification effort in the offshore service represented the most general of possible underwater tasks. By correlating the tasks described in each study, a list of a general set of underwater manipulator activities was derived. These activities were based on the observation that a consensus of possible underwater tasks can be reached in spite of different application contexts. For instance, underwater inspection of offshore structures using an ultrasonic, nondestructive testing (NDT) gun will include gun pickup, positioning, initiation of scanning, defect locating and recording (Busby, 1979). Likewise, a position-support operation in an underwater salvage mission specifies salvage equipment pick-up, transportation, and placement which constitutes a similar set of underwater activities, as in the underwater inspection activity. Thus a generalized task spectrum can be generated. A sample of such a spectrum is listed below, divided into six categories:

TABLE 2-2.

TYPICAL TASKS OF MODULAR OFFSHORE SERVICE
(From Ocean Industry April and August, 1978)

INSPECTION, MAINTENANCE, REPAIR

Visual inspection
Cleaning, inspection of welds
Replacement of anodes
Minor repairs of platforms
Removal of debris
Check for scour
Inspection of pipe lines
Minor repairs of pipe lines
Connection of pipe lines
Tie-in of flowlines
Seabed survey
Seabed sampling
Fire-fighting
Pollution control
Rescue

UNDERWATER REPAIR, SUBSEA CONSTRUCTION AND PLACING OF HEAVY OBJECTS ON THE SEABED

Minor repairs of platforms
Connection of pipe lines
Placing of objects on the seabed
Pile driving
Installation of permanent moorings
Repair of objects on the seabed
Fire-fighting
Pollution control
Rescue

TYPICAL TASKS	TASK* DIFFICULTY	TYPICAL TASKS	TASK* DIFFICULTY
Inspect/observe	E	Make up kill line	B
Recover tools	E	Bolt, unbolt	B
Clean, brush, chip	E	Replace valves	A
Cut cables	D	Drill, tap	A
Jack, spread	D	Place shaped charge	A
Untangle lines	D	Precise alignment	A
Attach lines	D	Non-destr. testing	A
Connect hydr. lines	C	U/W welding	A
Opr. overrides	C	Replace modules	A
Open/close valves	C	Precise measurement	A
Stab overshots	B	Midwater observation	E

*Task Difficulty: A = Most difficult, E = Least difficult

(1) Search - survey/locate/observe. The search tasks include activities associated with the detection and location of target and lost objects, wrecks, and bottom features; and the determination of general condition of underwater objects and immediate environment.

(2) Activation - position/activate/attach. The activation tasks include such activities as positioning of tools and locating of detected areas around objects; instrument activation, active sensing; simple mounting and hook-up.

(3) Travel - pickup/transport/place. The travel tasks include activities associated with the recovery of small objects, simple pick-up tasks; moving of objects in a simple trajectory; and deployment of sensors and instruments.

(4) Clearance - scan/clear/excavate. The clearance tasks include such activities as NDT; debris clearance; trenching, tunneling, and dredging, etc.

(5) Assemble/Disassemble - precision measurement/alignment/attachment. The assembly tasks include a whole range of activities from the measurement of clearance of mating parts or mounting patch; open/close valve, bolt/unbolt; U/W welding, clamping, and so on.

(6) Structure Repair - structure monitoring/carrying/assemblying. The structure repair tasks include activities such as replacement of valves, modules, and other combinations of above-mentioned tasks.

It appears that the degree of difficulty increases as a specific task activity moves down this task spectrum. On the other hand, based on a time-history analysis of underwater diver tasks, the frequency of actually encountering a specific activity in underwater operation decreases as it moves down the task spectrum. And it is safe to say that, the easier the operation, the more frequently it is expected to be performed.

2.2.3 Task Description at Motion Level. A more detailed level of analysis of a manipulation task can be accomplished by task breakdown according to either motion function or information requirements. The formal approach is one of procedure decomposition into finer motion descriptions, such as the one derived and termed "behavioral elements" by Pesch, et al. (1970). This latter approach is one of a situation description along major spatial or physical dimensions for a specific motion activity. For a complete description of a task at this level, it is necessary to include both *sequence* information, which describes the order in which motions are performed, and *taxon* information, which describes the conditions under which a given motion is performed.

Considering the problem of selecting a manipulation system in terms of a specific operation, Pesch, Hill and Klepser (1970) dissected applied salvage missions, which included operations such as sample collection, valve manipulation, rigging chain, tapping, threading, drilling, and connect/disconnect, etc., into five basic behavioral elements: simple travel, complex travel, simple grasp, alignment and tool use. They suggested that relative system performance for specific tasks made up of these basic elements could be predicted based on the constituent behavioral elements. More comprehensive analyses of behavioral motion elements were later conducted by other researchers to provide design specification for special tools in underwater tasks. The following list and examples of motion elements and classifications was extracted from Bertsche, et al.

(1978), Battelle (1976), and Nevin and Whitney (1977) in their study of the underwater work system and manipulator-operated tools:

(1) Travel - simple trajectory/complex trajectory,
e.g., fetch, return, etc.

(2) Alignment - (a) low precision/high precision,
- (b) no relative motion/with relative motion,
e.g., position, grasp release, contact, etc.

(3) Accommodation - (a) low precision/high precision,
- (b) no relative motion/with relative motion
- (c) active force applied/passive compliance,
e.g., insert, rotate, depress, slide,
seat, etc.

(4) Tool use - (a) linear/rotary/special
- (b) steady/impact,
e.g., saw, drill, tapping, hammer, cut, winch,
etc.

Analysis at this level indicates that the major dimensions of taxon information can be summarized into three categories: (1) geometric factors, including distance-depth, orientation, proximity and allowance; (2) compliance factors, including touch/slippage, force torque, inertia impact, and compliance stiffness; and (3) temporal configuration, including end-effector dynamics (speed and acceleration), arm position and access, etc. Specification and quantification along these dimensions could provide more rigorous task classification than those existing in the literature when such a detail analysis is justified by its objective. For the purpose of this study, we chose to quantify only selected dimensions that were critical to man-machine communication. This minimum set of task dimensions will evolve with the further analyses of the following sections.

Another aspect of information requirements, the procedure information, also can be summarized into three categories: (1) motion sequence in serial operation, (2) motion disjunction/conjunction in parallel operation, and (3) motion repetition. Related to the procedure information are planning activities, which will be described in detail in the next section and next chapter. Suffice it to say here that a manipulation planning process, in general, includes not only the task breakdown discussed above, but also the identification (primitive motion or previously conceived motion procedures) and the reformulation and synthesis (experimenting, tradeoffs between taxon and procedure information). These last aspects of manipulation activities have been difficult to analyze, which was recognized as "the problem of properly dissecting the overall task." As indicated by Pasch, et al. (1970), most behavioral elements are not independent actions that can simply be added end to end to describe a given task, but rather are attempts to dissect a continuum of independent activities. Thus, the degree of success for this attempt is related to the degree of independence of a particular motion on the procedure information, a characteristic referred to later as the "discreteness" of a specific operation. Summarizing the discussions in this section, the following set of descriptors was considered as the major attributes characterizing a manipulation operation:

(1) Precision.
(2) Compliance/force.
(3) Motion degree-of-freedom.
(4) Discreteness.

The previous analysis, however, was primarily concerned with intrinsic characteristics of a manipulation task. In real operation, one has to be concerned with hazardous environments in underwater operations, in which environmental impacts on manipulation are more stringent than in controlled laboratory or industrial operations. Thus, tasks need to be analyzed also by the impact of the underwater environment, as will be discussed in the following section.

2.2.4 Environmental Factors in Underwater Manipulation. The purpose of environmental factors studies is to examine and identify the important dimensions that affect the condition and performance of underwater manipulation. The underwater environment imposes conditions of high hydrostatic pressure, dynamic forces from wave and currents, limited visibility, and low temperature. All of these factors interact to make an everchanging environment for underwater tasks. Our analysis indicated that among the physical factors of importance were turbidity, current, and depth (distance). These factors, in turn, influenced the visibility, stationarity, predictability and accessibility of the work-space, all critical for successful operator control.

The deep water environment can be described in terms of a number of physical properties, as summarized in the Underwater Handbook (Schilling, et al., 1976). These properties interact to create hostile conditions for man and material. With each additional fathom, these conditions become more severe for the working diver (Battelle, 1971). In remote manipulation, some hostile conditions may be alleviated that are related to immediate support and safety. However, other factors, due to depth, remain stringent, including those related to vision/sensing (e.g., illumination, texture, resolution, etc.), transmission, and resource constraints. These factors, compounded with time stress, uncertainties, and potential hazards concerning task operations, create a sense of "remoteness" - a psychological distance which increases with physical distance, and decreases with familitary.

The physical distance poses severe constraints on visibility, and thus accessibility, in workspace due to the fact that natural illumination, reflectance, and contrast all decrease with depth. Subjective familiarity is reduced when color and texture of the object become uniform and less recognizable. Besides, subjective estimates of range and orientation become difficult in deep water. After considering the effects of depth along the dimension of physical distance and subjective familiarity, two other especially important factors in underwater manipulation remain to be addressed--turbidity and current.

Turbidity of the water depends on the size and concentrations of suspended particles in the water. This characteristic varies from location to location and typically is the major factor in reducing visibility and predictability in most underwater environments. The reduction is due to forward scattering and omni-scattering of light. Effects of turbidity variations on perceptual performance and display requirements have been studied by a number of researchers (e.g., Kinney, et al., 1969; Brant, et al., 1972; Vaughan, et al., 1976, 1977, 1978). In general, the detection distance was shorter in turbid water than in clear water, and distance estimates were invariability greater in turbid water than in clear water. The effect on visibility is of particular importance since control of present-day underwater manipulators depends almost entirely on direct visual feedback (Busby, 1978; NOSC Ocean Technology Department, 1978). Experience in laboratory studies and field work has shown that such devices are virtually impossible to use when vision is degraded by high turbidity water, compounded by poor angle of view, light failure, etc. These conditions occur frequently in deep-sea operations, particularly during bottom work. In fact, visual observation of manipulation has been a critical factor in machine-to-man communication in almost all previous experiment work (Smith, et al., 1979).

Like turbidity, the underwater current and hydrodynamic forces depend also on location. Their main effects can be considered to be on the stationarity of operation and the predictability of control dynamics. While the later effect can be described as a motion/force disturbance of the control system, the former one is more complicated and often has to be considered within the context of moving platform dynamics. Here, the global consideration of operation stability to compensate for both external current and reaction forces becomes a major concern. For example, current flow in the manipulator workspace of up to several knots could impose rather severe operational constraints in terms of reduced admissible trajectory volume of the vehicle plus manipulator. Another problem of no less importance for operator control in this uncertain environment is the lack of orientation and motion references. The sense of orientation is a prerequisite to virtually all forms of control, and it is directly related to the estimate of position and dynamics.

2.3 Operator Activity Analysis

In studying the task analysis of simulated maintenance tasks in our previous experiments, it was readily apparent that much of the subject's planning activities were devoted to developing, initiating, and monitoring the subplan. While the subjects usually accepted the overall goal and plan as instructed, the process included the activities of mission definition, requirement analysis and planning task hierarchy. The partitioning of tasks into subtasks, and subtasks into lesser subtasks, etc., was left to the subjects, allowing concurrent lower level planning and execution (Sacerdoti, 1975; Weissman, 1976). The interface of the operator function in planning and the supervision of manipulation tasks is illustrated in Figure 2-2. In the case of a free-swimming (unmanned, remotely-controlled submersible) underwater manipulation, the typical supervisory control activities of the operator may include the following:

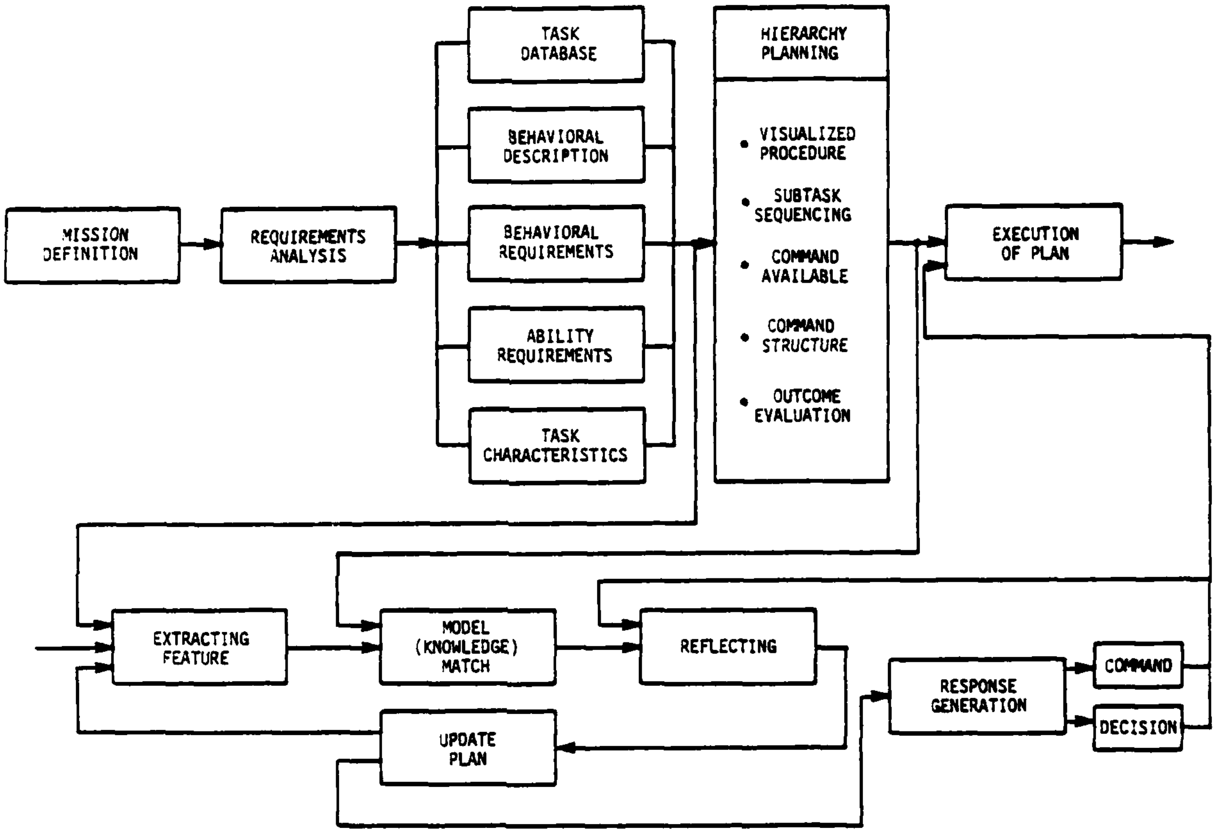


FIGURE 2-2.
OPERATOR FUNCTION IN PLANNING & SUPERVISION
OF MANIPULATION TASK

(1) Cognition of the environment.

(a) Object, obstacles, seafloor, turbidity, current, etc.
(b) Rate of change of environmental conditions.
(c) Search, detection, localization, identification, and estimation.

(2) Status apprisement.

(a) Navigation: three-dimensional position relative to object.
(b) End-effector maneuver: configuration, predicted trajectory.
(c) Command and control: traded or shared commands, control rates and modes (resolved motion or joint modes).
(d) Subsystems: capability, limitation, and reserves.

(3) Monitoring

(a) Subtask sequence and completion.
(b) Effector-object interaction.
(c) Geometric relation.
(d) Force-torque tolerances.
(f) Hardware failure and potential hazard.

(4) Execution functions.

(a) Change command and control modes.
(b) Change manipulator configuration.
(c) Begin next phase of subtask operation.
(d) Establish backup mode.
(e) Communicate information.

(5) Planning procedure and error recovery.

  (a) Command definition.
  (b) Sensor deployment.
  (c) Subplan editing, geometry editing.
  (d) Inspection, manipulation and testing.

The set of supervisory control activities can be grossly classified into global supervision and interactive control. The former emphasizes situation assessment and the latter emphasizes action implementation.

2.3.1 Supervisory Function Hierarchy. It is considered that in the supervisory loop, the operator's activity performance can be classified into synthesized categories of overall system performance. As Emery (1969) has recommended, the principal test of category inclusion will be payoff relevance--only categories that discriminate among actions and that result in actions with differing utilities or performance will be included. Utilities and performance in this context refer to the following dimensions:

(1) Achievement of task goal and operational function.

(2) Time period and expected frequency.

(3) Task operational requirement.

  (a) Time.
  (b) Accuracy.
  (c) Tolerance.
  (d) Force.
  (e) Dexterity.

(4) Task conditions, as summarized in Figure 2-3, including the following dimensions:

(a) Intrinsic manipulation characteristics, including those related to low-level motion constraints (precision, compliance, motion degrees-of-freedom, etc.), motion decomposition (discreteness and criticality of motion elements) and motion sequence (abstractness, structuredness, simultaneity, repetitiveness and variability, etc.).

(b) System features, including those related to effector (accuracy, speed, capacity and articulateness, etc.) and those related to feedback display (resolution, frame content, time delay, frame rate, and display-motion compatibility).

(c) Environmental conditions, including those related to work-space situation (visibility, stability, predictability, and accessibility, etc.) and those related to ambient conditions (turbidity, current, depth, distance and familiarity, etc.).

Operator supervisory activities, after previous enumeration of task demands, information and functional requirements, can be summarized into the following three categories:

(1) Situation Apprisement.

(a) Navigation - 4-D position relative to motion plan.
(b) Guidance - current and predicted trajectory.

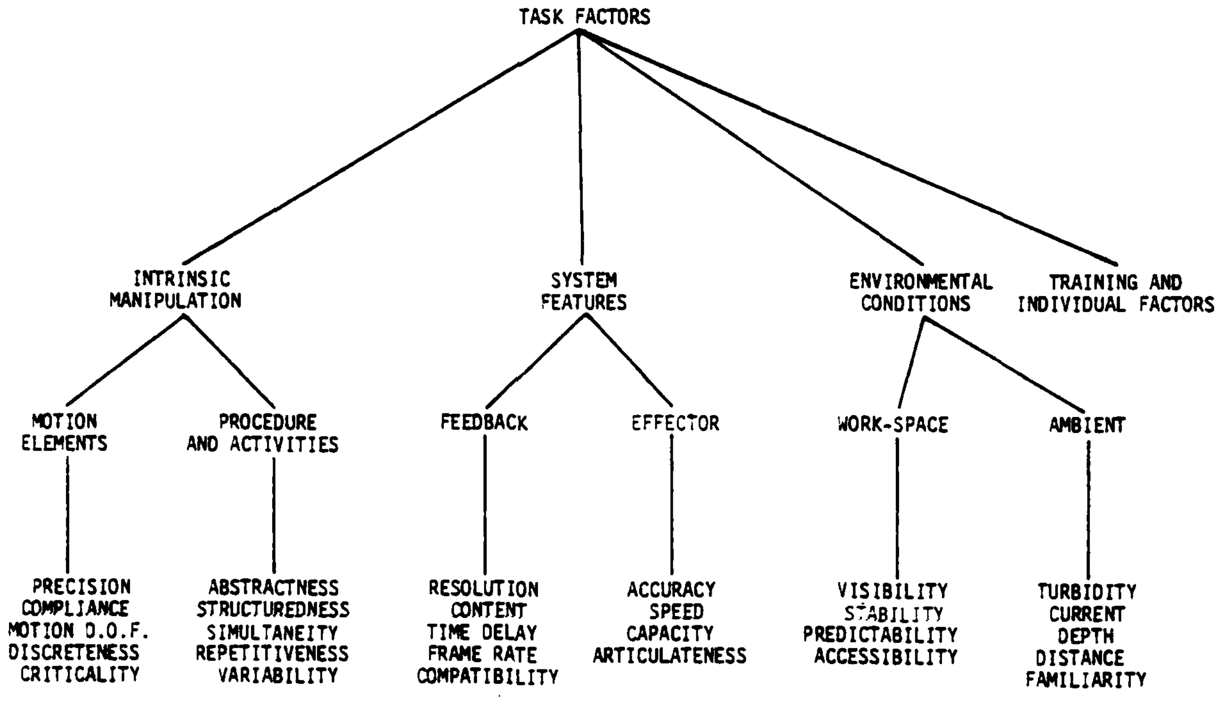


FIGURE 2-3.
MAJOR DESCRIPTORS OF TELEOPERATOR CONTROL TASKS

(c) Control mode - automatic modes and manual modes.

(d) Subsystem - configuration, capability and reserve.

(e) Environment - terrain, weather, traffic (threat).

(f) Clearance - obstacle, threat.

(2) Situation Monitoring.

(a) Subsystem performance tolerances.

(b) Geometric or force tolerances.

(c) Subplan sequencing/completion.

(d) Potential failures/hazards.

(3) Action for Execution.

(a) Change mission phase.

(b) Edit program.

(c) Change arm configuration.

(d) Reduce control accuracy errors.

(e) Communicate information.

The list consists of a spectrum of supervisory functions. The top level of this spectrum represents the demands for multiple channel processing by the operator, and the low level of this spectrum represents the demands for serial processing, with its associated limited capacity. It is contended in this study that the prediction of operator performance in complex tasks using a remote sensor-based system requires a knowledge of operator proficiency in the components of the task and an estimation of the information demands by each component.

Accordingly, the information components within specific information types relevant to the level of supervision hierarchy need to be classified. A tentative set of classifications is given as follows:

(1) Status/warning.
(2) Quantity.
(3) Comparison.
(4) Time sequence.
(5) Prediction.
(6) Instruction/alternatives.
(7) Feedback.

A useful analysis may lead to the development of a communication matrix relating information components to display a format such as:

(1) Alarm.
(2) Alphanumeric (text or tabular).
(3) Symbolic/schematic.
(4) Digital.
(5) Dial.
(6) Pictorial.
(7) Perspective.
(8) 3-Dimensional.

2.3.2 Teleoperator Control Task and Information Requirements. Since vision is the fundamental feedback system for most underwater manipulation systems, we started our analysis with visual function requirements. The general visual functions for manipulation tasks were studied and are listed as follows:

(1) Acuity - degree of detail that can be discriminated or affected by fixation position and also by illumination and contrast.
(2) Size estimation - judgment of absolute and relative size.
(3) Shape discrimination - discerning difference in form.
(4) Brightness discrimination - discerning changes of gray level.

(5) Recognition of pattern - recognizing image pattern in different orientations.

(6) Distance estimation - estimating distance between offset targets.

(7) Stereoacuity - size of detail in depth perception.

(8) Movement resolution - perceiving movement.

(9) Rate resolution - perceiving rate of movement.

(10) Color Discrimination - discriminating hue, saturation and brightness.

(11) Frame of reference - resolving operator's coordinate reference to both platform (manipulator system) and work object system.

(12) Zoom reference - resolving dual requirements of both acuity and range.

Further analysis and review of relevant documents will provide a useful crosslist between operator visual functions and visual cue requirements. As a summary of this section, Figure 2-4 presents a list of factors affecting the quality of visual (scene) information. Some of the factors were investigated in a simplified environmental feedback experiment, which is to be discussed in Chapters 4 and 5.

A prerequisite of this analysis was the study of visual information requirements. A matrix of operator function and relevant visual cues has been developed which correlates the following two sets of parameters (Figure 2-5):

(1) Visual cues.

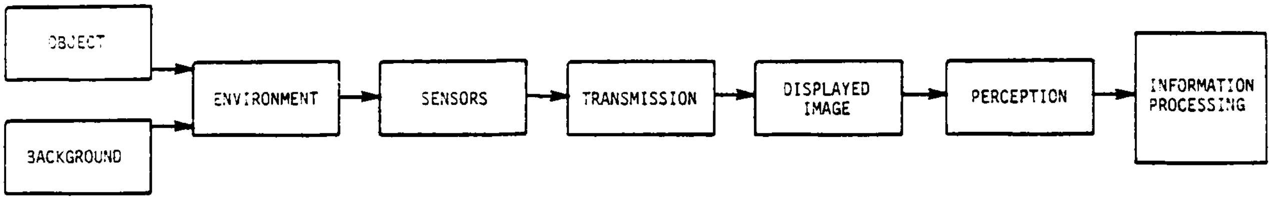
(a) Internal reference.

(b) External reference.

(c) Perspective.

(d) Texture.

(e) Motion parallax.



OBJECT/BACKGROUND

Size
Reflectance
Illumination
Contrast
Color
Texture Pattern
Motion Duration
Angularity
Markings

ENVIRONMENT

Depth
Ocean/Harbor
Turbidity
Absorption
Scattering
Backscatter

SENSOR/TRANSMISSION

Signal/Noise
Band Width
Format
Filtering
Spatial Resolution
Transducer Resolution
Frame Rate
Time Delay
Motion Resolution
Detection & View Angle
Frame of Reference

DISPLAY

Image Resolution
Contrast
Color
Brightness
Frame Rate
Depth of View
Monitor Size
No. of Monitor
Display Cue Augmentation

PERCEPTION

Visual Acuity
Size Estimation
Shape Discrimination
Brightness
Recognition of Pattern
Estimation of Distance
Convergence
Retina Parallax
Motion Parallax
Accommodation
Perspective
Size of Known Object
Light and Shadow
Interposition
Haziness of Distance
Detection of Motion
Rate Estimation
Contrast Sharpness
Color & Surface Texture
Frame of Reference

FIGURE 2-4.
VISUAL INFORMATION FACTORS

	VISUAL CUES									
	SPATIAL RELATIONS	PERSPECTIVE	EXTERNAL REFERENCE	INTERNAL REFERENCE	MOTION PARALLAX	INTERPOSITION	SIZE	SHAPE	LIGHT AND SHADE	TEXTURE
DISTANCE TO OBJECT	X	X	X	X	X		X		X	
LOCATION TO OBJECT	X	X	X	X	X		X	X	X	
AVOIDANCE OF OBSTACLE	X		X		X	X			X	
DIRECTION OF GRIPPER MOTION	X		X		X				X	
CLOSURE RATE	X	X	X		X	X			X	
ALIGNMENT OF GRIPPER	X	X	X					X	X	
GRASP ARRANGEMENT			X					X	X	
CARRYING EXTENDED LOAD	X	X	X	X	X	X	X		X	
UNLOADING	X	X	X	X	X	X			X	
LOCATION OF SPATIAL WAY-POINT	X	X	X	X		X	X		X	X
LOCATION OF GRIPPER TO SPATIAL WAY-POINT	X	X	X	X	X	X	X	X	X	X

FIGURE 2-5.
MATRIX OF OPERATOR FUNCTION AND RELEVANT VISUAL CUES

(f) Size.

(g) Shape.

(h) Filled space.

(i) Interposition.

(j) Light and snade.

(k) Spatial relations.

(2) Operator elementary control functions.

(a) Location of capture point.

(b) Location of gripper relative to capture point.

(c) Closure rate.

(d) Direction of gripper motion.

(e) Alignment of gripper.

(f) Rearrange gripper grasp.

(g) Carrying extended load.

(h) Unload.

(i) Avoidance of side structure.

(j) Location of object.

(k) Distance to object.

(l) Closure rate.

(m) Angle of closure.

Various types of computer aiding techniques for teleoperator control were considered and are summarized in Table 2-3. Control aiding functions include (1) motion control - those help to achieve the desired effector position/posture, (2) motion programming - those help to achieve a series of motion control or trajectory, and (3) high level command - those help to achieve operational, task-specific goals. To support the control aiding functions, appropriate display of system/task status may be used. Considerations include (1) the scope of environmental state and detail

TABLE 2-3

TYPES OF COMPUTER-AIDING FOR TELEOPERATOR CONTROL/DISPLAY FUNCTIONS

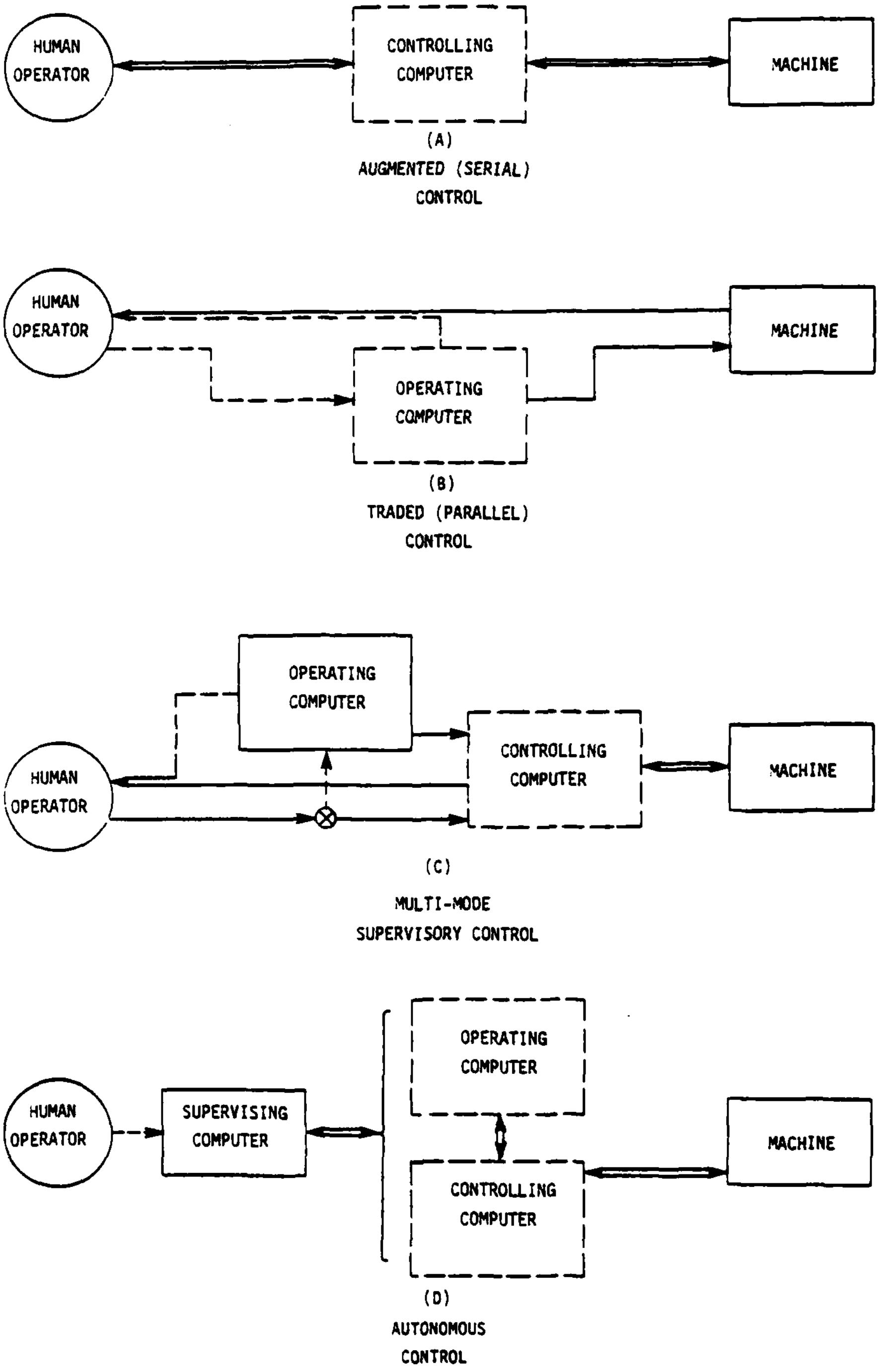
CATEGORY	SUBCATEGORY	EXAMPLES
1. Control		
Control Motion	Geometric/dynamic transformation	Resolved motion rate control. Resolved motion acceleration control
	Effector motion	Pre-canned low-level motions, Multi-finger dexterity
	Accommodation and compliance	Active sensor steering. Passive compliance
Programming Motion	Trajectory repetition	Spatial points. Spatial paths
	Trajectory interpolation	"Reverse problem" solutions
	Trajectory transformation	Relative (moving-reference) trajectories
Command mix	Symbolic vs. analogic	Function keys vs. joysticks
Command *Structure*	Abstraction	Multiple-level vs. single-level
	Organization	Linear sequential vs. hierarchical network
2. Display of System/Task Status		
Display scope and level	Environmental state	Spatial, force, tactile and proximity
	Machine and programming state	Control mode and status in parallel control
Display Format	Display codes	Alphanumeric, video TV, graphics
Display integration	Information flow	Control of Frame content and update rate
	Cue integration	Predictor, filtering and enhancement

level of machine and programming state, (2) the display code in use, and (3) display integration, such as control of information flow and cue integration. Many of these computer-aiding functions were included in the task-oriented supervisory control language (TOSCL) developed in this program. While the design and the evaluation of the language will be described in the next two chapters, the man-machine interaction aspects will be discussed in the next two sections.

2.4 Man-Machine Interaction

Recent advances in microelectronics and sensor and processing technology have lowered the cost of sensors, processors and communication to a level where use of unmanned distributed subsystems in the hostile environment, identifying, localizing, and manipulating moving objects, becomes feasible. In particular, the concept of having the operator in a remote site, overseeing the operations of automated inspection and manipulation platforms, holds great promise. The use of on-board intelligence in these systems can, in the foreseeable future, provide "shared man-computer control." This concept within the framework of this study implies that the operator in the commanding site communicates intermittently with the on-board computer, which in turn performs continuous control of the sensors, effects, and platform subsystems.

Four types of shared man-computer control of remote manipulation are perceived based on the functional relationship for the operator and computer in controlling the machine. They are (1) augmented (serial) man-computer control, (2) traded (parallel) man-computer control, (3) multimode supervisory control, and (4) autonomous supervisory control. In augmented control (Figure 2-6a), the major computer functions are interpreting and transforming control commands received from the operator. These computer functions include resolved motion control (such as resolved motion rate control) and steering calculation for active accommodation,



(A)
AUGMENTED (SERIAL)
CONTROL

(B)
TRADED (PARALLEL)
CONTROL

(C)
MULTI-MODE
SUPERVISORY CONTROL

(D)
AUTONOMOUS
CONTROL

FIGURE 2-6.
COMMUNICATIONS IN FOUR BASIC FUNCTIONS OF
COMPUTER-AIDED MANIPULATION

etc. Taxon information constitutes a major portion of operator computer communication. In traded control (Figure 2-6b), the major computer functions are sequencing and programming the operations of the machine based on the symbolic command given by the operator. (An example of traded control is the automatic motion control.) Procedural information constitutes a major portion of operator-computer communication. A hybrid of the above two-types of man-computer control is called the multi-mode supervisory control (Figure 2-6c), in which the computer's operating capabilities and controlling capabilities are aggregated. (Examples are the variable and chain commands.) Communication based on both taxon and procedural information needs to be specified between operator and computer. The most advanced type of shared man-computer control includes the supervisory and planning capabilities of the computer in performing manipulation tasks. The autonomous supervisory control (Figure 2-6d) permits the subsystems and vehicles to act as an autonomous robot for extended time periods. Jointly, the subsystem and the vehicles respond to their environment in the pursuit of task goals preprogrammed by the operator, and possibly updated in an asynchronous communication cycle.

Of the last two types of shared man-computer control, supervisory control is seeing increased use in a number of advanced development projects. Free-flying and free-swimming robot vehicles are currently being developed as microcomputer-controlled sensor platforms (Jet Propulsion Lab, 1979; Talkington, 1978). On-board intelligence has been demonstrated in preprogrammed maneuvers, and emergency/abort routines onboard these vehicles. The development of distributed sensor control in compliance with an artificial arm can be configured with the algorithms of distributed sensor signal interpretation and automatic generation of motion command (Corker, et al., 1980). Industrial processes can now be monitored and controlled by decentralized data acquisition and parallel processing. Such fast and intelligent computers can provide sound, well-evaluated decisions with potential improvements in reducing system risk, operator workload, and errors. On this frontier, the operator has to interact with an intelligent system capable of processing and routing information, executing control actions, and making choices in view of priority conflicts. The key underlying issues are the proper designation of the roles that the human and the computer are expected to play and suitable interface design for the specified roles. Related computer-aided interface functions include:

(1) Aggregate sensor data into a communicable format.
(2) Allocate control between the operator and the computer.
(3) Manage the information flow and the communication bandwidth of the link between the remote operational site and the control site.
(4) Generate a compatible display of system status and operational environment.

One major issue concerns man-computer functional allocation. One straightforward approach is to allocate a fixed portion of the set of the tasks to the computer with the remainder of the set being allocated to the human. Licklider (1960) has proposed that the human sets goals, formulates hypotheses, determines criteria, and evaluates results. On the other hand, the computer should perform routine work such as transforming data, simulating models, and implementing results for the human decision-maker. However, the division of tasks is not as clear-cut for processing and decision making tasks that include computerized decision subsystems.

In general, the rules of thumb suggested by many researchers are that man will handle the very-low-probability situations, and fill in the gaps in the problem solution or in the computer program; while the computer may serve as pattern or statistical inference, decision theory, or game theory machine, to perform elementary evaluation, diagnosis, and pattern recognition as a second role (e.g., Ramsey, et al., 1979; Vaughan, et al., 1972; Rouse, 1975; Steeb, et al., 1975; Johnsen, et al., 1978). On the other hand, many researchers, including Steeb, Weltman and Freedy (1976), and Chu and Rouse (1979), have suggested that a dynamic or adaptive allocation of responsibilities may be the best mode of human-computer interaction. With adaptive allocation, responsibility at any particular instant will go to the decision maker most able at that moment to perform the task. Such a scheme is adaptive in the sense that the allocation of responsibility depends on the state of the system as well as the states of the decision makers. Thus, changes in system or decision maker states results in changes in the allocation policy so as to optimize performance.

The policies espoused in these developments are particularly suitable for implementation in a hierarchical task structure of the procedural net, as will be described in detail in the next chapter. It is sufficient here to classify the existing schemes and method of computer-aided interface into four categories: control interface, programming interface, command interface, and communication automation. The schemes under various levels of development are listed in Figure 2-7, including those features incorporated in the task-oriented supervisory command (TOSC) language developed in this program: multi-mode control, shared multi-programmed, procedural network, and mix-initiative. These will also be discussed in the following chapters.

Once the task structure and the communication protocol between human and computer have been established and the status of decision makers and the system states has been determined, it then becomes possible to dynamically allocate functions. The three main reasons for adaptive allocation are:

(1) Efficient utilization of system resources, based on the theoretical analysis of system operation (e.g., Chu, 1978).

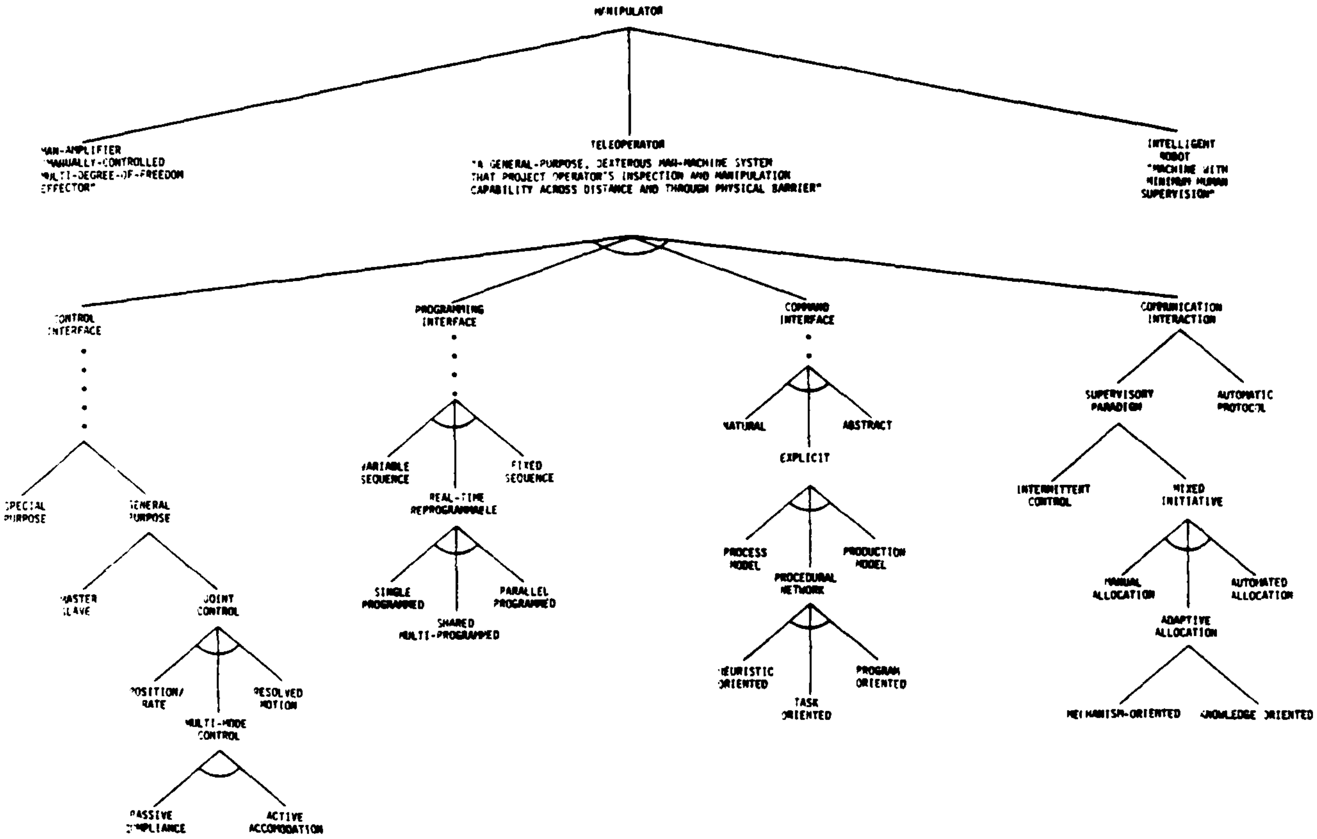


FIGURE 2-7.
TAXONOMY OF MAN-MACHINE INTERFACE IN TELEOPERATOR CONTROL

(2) Increased flexibility in coping with subsystem malfunctions. The possibility of the computer encountering either a hardware failure or an event whose decision making requirements exceed its abilities can never be overestimated. It would seem reasonable that the human should be allocated at least monitoring responsibility for all tasks at the top-most hierarchy level (e.g., Moray, 1976). On the other hand, if tasks are strictly allocated, the human would not know or attend to those operations under the computer's supervision until abnormal situations developed and placed still higher demands on the human to explore and control the subsystems.

(3) Effective role assignment to the human. The adaptive policy potentially assures the human a coherent role in that the considerations of operator load and style are taken into account (Steeb, et al., 1979).

The potential advantages of the supervisory control over a direct manual or a standalone automatic control include:

(1) Increased flexibility and dexterity in mission performance--achieved through the operator in the supervisory loop.

(2) Increased reliability and, for certain tasks, precision--achieved through system modularity, redundancy provided by supervisory loop, and extended subsystem capability.

(3) Less susceptable to failure and communication channel breakdown--due to confident autonomy in subsystems allowing intermittent communication with reduced bandwidth requirements.

(4) Great efficiency and lower communication cost when communication is extremely bandlimited or imposes a time delay--due to enhanced real-time response, parallelism in processing and allowed data/command compression in transmission.

In this context, the current program sought to identify and evaluate important man-machine communication factors, and can be considered as a first step toward the realization of those advantages in the use of supervisory control. For the convenience of evaluation, categories of communication factors were identified in the design process. They are listed in Figure 2-8, with areas covered in the studies shaded. Man-machine communication requirements within these categories were analyzed and evaluated and the results are reported in Chapter 3 and Chapter 4, respectively.

COMMUNICATION FACTORS CATEGORIES	MAN-TO-MACHINE COMMUNICATION	MACHINE-TO-MAN COMMUNICATION	TWO-WAY INTEGRATION	ENVIRONMENTAL FACTORS
LEXICAL LEVEL	CONTROL AND PROGRAMMING FUNCTIONS	DISPLAY CODE AND FORMAT	CONTROL-DISPLAY [illegible] COMPATIBILITY	UNCERTAINTY OVER OUTCOME TRAJECTORY
SYNTATIC LEVEL	COMMAND [illegible]	STATUS FEEDBACK [illegible]	FUNCTIONAL ALLOCATION AND PROTOCOL	UNCERTAINTY OVER PROCEDURE AND STRUCTURE
CONCEPTUAL/ SEMANTIC LEVEL	COMMAND CONSTRUCTS	DATA ORGANIZATION AND DISPLAY INTERPRETATION	PROBLEM DOMAIN PLANNING AND INFORMATION FLOW	UNCERTAINTY OVER TASK GOAL

FIGURE 2-8.
CATEGORIES OF COMMUNICATION FACTORS
(AREAS COVERED IN THE STUDY ARE SHADED)

3. TASK-ORIENTED SUPERVISORY CONTROL SYSTEM METHODOLOGY

3.1 Overview

As we extend shared control of a manipulator system to the full range of capabilities afforded by the computer element, the question of man-computer communication becomes of primary importance. Many of the previous studies (e.g., McGovern, 1974; Whitney, et al, 1977) advocated the shared man-computer control. Only sketchy attention, however, has been devoted to the analysis and design for effective man-machine communication in the task-oriented supervisory control paradigm. This chapter describes an attempt at Perceptronics to develop a model for the man-machine communication process, to derive principles for language design, and to implement a command language and associated interface. The effectiveness of this approach was experimentally tested and the results are to be discussed in the next chapter.

Previous work in manipulator command language has been largely concentrated in two approaches. In the "programming approach," exemplified by Ambler, et al (1973), Lozano-Perez (1977), Finkel, et al (1974), and Paul (1979), the manipulator is controlled by a program-like command language which requires prolonged programming and debugging for any specific task. In addiiton, the environment is presumable predetermined and tightly controlled, prohibiting the use of such a system in novel environments. In the "direct control" approach, exemplified by the master-slave system of Goertz (1954) and the submersible manipulators used by Pesch, et al (1971), continuous on-line attention of the human operator is required and advantage is not taken of the computer capability for control assistance.

The approach of this study falls in a third category called "shared man-computer control" by Freedy (1973) or "supervisory control" by Ferrell (1965). A procedural net planning concept of Sacerdoti (1975) was used as a model of the man-machine communication process on the conceptual level. This model suggests a language with the structure of defining new commands as "chains" of more primitive commands. On the syntactic level, previous work in man-machine dialogue (e.g., Foley and Wallace 1974) suggests that commands should have the form of separate sentences, each specifying a complete task, and that the communication process should have conceptual, visual, and tactile continuity. The language developed in this study follows the following principles: (1) communicate using complete concepts specific to the task at hand, (2) allow for on-line real-time adaptation to specific situations, and (3) allow communication at the level of detail most comfortable to the user. The language so designed is limited in that spatial points and trajectories have to be defined before they can be addressed, and in that the computer does not employ any model of the external world, as would be useful in correcting errors, automatic planning of motions, and updating subtask goals, etc.

Nevertheless, using this representation, a problem domain can be described as a hierarchy of procedural nets, each representing some task in the problem domain in terms of its goal, its component subtasks, and their relation to the environment. This information is represented as a combination of data structures and procedural information in the procedural net. Sacerdoti used this model as a formalism to represent complex manipulation tasks which is useful for planning and problem solving. This study hypothesizes that the adapted model is compatible with the human perception of a complex task, and can serve as a medium for communication of such a task between an operator and a manipulative mechanism. Using this model as a base we derived principles for language organization, command features and communication protocol implementation. The model, further, makes clear the role of the human operator in the supervisory control approach to manipulator control, and also clarifies the functions that should be performed by an assisting computer placed in the control and monitoring loop.

3.2 Background

3.2.1 General. The related areas of research to be addressed include human-computer interaction, human factors in interactive systems, command language in general, and manipulative command and feedback systems in particular. Relevant principles have been derived from the related literature to guide the design of the TOSC system and the exploration of the most effective command/display modes. These principles, described in the next section, are organized as follows:

(1) Preliminary principles of command language design.

- (a) Use task and concept oriented commands.
- (b) Use hierarchical command organization.
- (c) Allow for mixed initiative.
- (d) Provide concept definition capability.
- (e) Use constrained, standardized language.
- (f) Provide for tactile, visual and contextual continuity.
- (g) Any error can be undone.
- (h) Simplicity.

(2) Preliminary principles of feedback display design.

- (a) Provide feedback on system state.
- (b) Consistent structure of command and feedback.
- (c) Provide proportional effect and optimal scaling.
- (d) Provide spatial, movement, conceptual, and task compatibility.
- (e) Provide display integration of information channel.

3.2.2 Preliminary Principles of Command Language Design.

(1) *Use Task and Concept Oriented Commands.* This is related to the idea of making the system compatible with the user's concept of the problem domain. The human operator thinks in terms of tasks and complete concepts and these are the basic units in his language. Organizing the command language around such units provides a natural mental framework for the user, allowing him to think about the task at hand, rather than the mechanics of expressing his intentions in terms of manipulator orientation and joint motions. As Bennett (1972) has written in an excellent review on The User Interface in Interactive Systems, "Software designers have been justly criticized for providing tools that force users to behave in nonproductive modes. While designers have been correct in foreseeing new modes, either they did not anticipate new patterns accurately (poor design), or they did not effectively transfer the user's mind to the conceptual framework that guided the design (poor training). In any event, the impact on system performance of the user's concept of the tool is too important to be left to chance."

(2) *Use Hierarchical Task Organization.* As argued by Dijkstra (1972), and the structured programming discipline, such hierarchical, systematic structuring is essential for producing large, error-free programs. We believe that this principle carries over in other communication environments such as the language for real-time communication with a general-purpose manipulator. The language should have facilities to define a task in terms of its subtasks. The user will be able to express his intentions at a comfortable

level of plan detail. On one hand, giving commands in too much detail would cause the user to lose sight of the overall task while dealing with the details. On the other hand, giving a command at a gross level does not allow adaptation of the command to the peculiarities of the situation at hand. The communication should take place at the highest symbolic level comfortable to the user and he should have the facility to choose this level. There are, however, cases where it is necessary for an operator to manually control the manipulator motion, e.g., for fine adjustment, and a manual "back-up" facility should be incorporated into the hierarchical system.

(3) *Use Sentence Structure.* "Communication should be carried out in a terse 'Natural' language, avoiding the use of mnemonics. Abbreviation should be allowed wherever possible," (Kennedy, 1974). Foley and Wallace (1975) suggested that the commands in a language should be task-or-concept-oriented and should have sentence structure. That is: "An action language is *sentence structured* if, within a given phase or subdomain of discourse, each complete user-thought can be expressed in a continuous sentence of input device manipulations with standard patterns of beginning and termination. Upon termination, the machine returns to a state from which similar action sequences, other sentences, can begin." The structure is enhanced if the verb-noun format of natural language is utilized. Treu (1975) calls the essence of a command an "action primitive" and suggests a specific structure to each command: "(1) action verb; (2) action qualifier(s); (3) object(s) of action; (4) object qualifier(s)."

(4) *Allow for Mixed Initiative.* In the task of controlling a manipulator, some functions are performed better by the human operator and some are done better by the on-line computer. Allowing easy transfer of control from one to the other with initiative going to the best performer in each subtask should improve overall performance. In a hierarchically organized system, mixed initiative is relatively easy to incorporate.

(5) *Provide Concept Definition Capability.* In natural language we use pre-defined concepts and the perceptual capability of the listener to refer to the objects and actions in the environment. The current state-of-the-art in machine perception does not allow references to points or paths in space using objects in the environment. It is necessary to bring the manipulator to the specific point or to move through a path in order to define them for the computer. Using coordinates in a dynamic situation presents difficulty because of human distance estimation errors.

Additional principles derived from work in related areas of man/machine communication were also incorporated into the language and interface design. These related areas include: man/machine "conversation" about graphic data (Foley and Wallace, 1974); man-display mechanisms (Engel and Granda, 1975); and others in human factors engineering design.

(6) *Use Constrained Standardized Language.* As shown by Ferrell (1973), an artificial, constrained, and standardized language facilitates performance in manipulative tasks better than a free-format, English-like language. The advantage comes from

the fact that, although entire manipulation task goals are readily and perhaps most easily described by a person using ordinary English, the complex geometrical and temporary configurations of objects and motions are not readily formulated in such a way. A structured but flexible, task-oriented, artificial command language can enable the operator to work more efficiently as well as simplify the process of machine translation.

(7) *Provide for Tactile, Visual and Contextual Continuity.* Foley and Wallace (1974) also emphasize the importance of continuity in the operator sensory and comceptual interaction with the system. Tactile continuity refers to natural grouping and flow of motion required for the tactile input devices such as keyboards, joysticks, etc. Visual continuity refers to the arrangement of information, so that within a given sentence, (i.e., one conceptual command), the eye should focus on a single area on the control panel or move in a continuous manner throughout the expression of the sentence. Contextual continuity refers to providing immediately perceivable responses and giving standard feedback information in dedicated, fixed positions in the visual field.

(8) *Any Error Can Be Undone.* Human operators, especially under time pressure are error prone. A system accepting operator commands must have easy "error recovery capability." Engel and Granda (1975) see this as an important feature of any computer command language. Kennedy (1974) states: "Each entry should be short so that errors can be corrected simply and a reasonable tempo can be established." However,

"verbose error messages should be avoided for the sophisticated user...and he should have a facility for suppressing long error messages."

(9) *Simplicity.* To facilitate training and ease the memory burden on users the language must be simple. This precludes all kinds of fancy features that regular programming languages are replete with, such as loops, conditional statements and logical relations. Falkoff and Iverson (1973), while describing the design of APL, suggest additionally, simplicity and practicality as the general guidelines they used in the development of APL. In particular, "...Simplicity enters in four guises: Uniformity--rules are few and simple; generality--a small number of general functions provide as special cases a host of more specialized functions; familiarity--familiar symbols and usage are adopted whenever possible (this is also, in part, 'task oriented'); and brevity--economy of expression is sought."

Additional principles indicated by Kennedy are:

Control: "Control over all aspects of the system must appear to belong to the user."

Redundancy: "Redundancy in the dialogue should be avoided or reduced, especially as the user becomes more familiar with the system."

Adaptibility: "The system should adapt to the ability of the user."

Communication Rate: "The rate of exchange must be within the user's stress-free working range. Control of the rate should always appear to belong to the user."

A conscious attempt was made to incorporate as many principles as possible into the manipulator language design. A natural procedure is to go through a top-down, three-level consideration: conceptual/semantic level, syntactic level, and lexical level. The conceptual/semantic level refers to the fundamental approaches such as task-oriented or program oriented, sequential or hierarchical network. The syntactic level refers to the language structure, such as command sequence and command-operand order. At the lexical level, words of the action elements are related to the elementary hardware/software functions represented by an analogic or a fixed symbolic command word. The procedure is to be described in Section 3.4.

3.2.3 Preliminary Principles of Feedback Display Design.

(1) *Provide Feedback on System State.* In dealing with man-machine display interfaces, Engel and Granda (1975) emphasize the importance of providing continuous feedback to the user about the state of the computer system he is dealing with. The information alleviates the frustration generated in the operator when dealing with a complex black box.

(2) *Consistent Structure of Language and Feedback.* Language structure refers to the external formats of the commands in the language. Feedback structure refers here to the format of the information about the system state presented to the operator. Consistent structure reduces training time, error rate, and user memory requirements and is a

basic feature of a command language. It is related to Foley's and Wallace's (1975) idea of sentence structure.

(3) *Provide Proportional Effect and Optimal Scaling.* Proportional effect refers to a monotonic, analog visual form of geometric or sensor information. Optimal scaling refers to appropriate level of precision/resolution in displayed (task or environment) information (McCormick, 1970). High resolution sensor information is expensive to obtain in an operational environment, and much of the critical information for operator under shared man-computer control can be extracted from data of low frequency and reasonable resolution.

(4) *Provide Spatial, Movement, Conceptual, and Task Compatibility.* Spatial compatibility refers to the physical features and arrangement in space of the parts of the display (McCormick, 1970). Movement compatibility refers to the relationships among the directions of the movement of the displays, controls, and system responses (Chapanis, 1965). Conceptual compatibility is based on the association of user's intuitive understanding and the display code representation. Task compatibility refers to display information relevance and the format related to the desired perceptual task (Schutz, 1961). Whenever possible, related information should be presented in a common display to expedite visually encoding the data with the least effort (Roscoe, 1968). This relates to the aesthetic aspect of display integration, which refers to the coordination of the display components so as to minimize both interference and scanning effort. In general, compatibility is achieved

when "the ensemble of stimulus and response combinations comprising the task results in a high rate of information transfer" (Fitts and Seeger, 1953).

(5) *Provide Display Integration of Information Channels.* In multiple-task display integration, issues of consideration include minimized visual (attention) competition, conformed scan pattern, clarity of form, and disciplined prompting (Ashford, 1969). Priority of the display parts may be derived by appropriate features such as relative size, order, intensity, contrast color, and textures, etc. In multiple-source display integration, issues of consideration include the choice of effective feedback modality, resolution and format of the presentation, and the gating of the information. Bejczy (1977), in an exploratory study of displays for supervisory control of manipulators, proposed a design process of (a) selection of a proper type of display; (b) selection of a proper format for a given type of display so that the display presents all necessary information in a timely manner and in a form easily perceived by the operator; and (c) integration or integrated display of visual and non-visual sensor information.

In a shared man-computer control paradigm, it appears that complete feedback of system and task status includes information concerning: (1) machine status--manipulator and subsystem functions, (2) program state--commands defined and mode in execution, (3) task in progress--current activity and criteria, (4) environmental state--workspace and environmental data. A three-level, top-down organization of display feedback design is perceived: (1) display system level--sensory modality, display device and quality, (2) data structuring/processing level--frame content, frame rate and bandwidth, etc., and (3) coding presentation level--geometric primitives, symbol and cue considerations.

3.3 Man-Machine Communication Models

3.3.1 Background. A model is a representation of some part of the real world which is useful in understanding the important variables relevant to some problem, and the structural or quantitative relationships among such variables. Models considered are the ones used to describe observable events and relations among measurables of the communication processes. They may be descriptive if they intend to portray the external behavior or internal mechanism of the process. Or they may be normative when desirable, rather than actual, behavioral action is prescribed. Usually, a much more extensive knowledge about the modeled system is needed to provide a successful normative model. The objective here is a modest one, i.e., to develop a structure-descriptive model of a communication process which could bridge the conceptual level and syntactic level of process description.

For a shared man-computer manipulator control system, many models are needed to address different aspects of the system functions. In the lower level of manipulation, models are developed to describe the static dynamic behavior of the manipulator. The work of Paul (1971), who built a dynamic model of the manipulator at Stanford, is an example of a model using a rigid body model of each manipulator segment. The differential equations obtained were used to predict forces and trajectories of the manipulator parts under different loads. At a higher level, at the human-task interface, a model can be used to predict time and accuracy of motions. Fitts (1954), using information theory tecnniques, showed how a logarithmic measure of the ratio of distance of movement to the allowed error tolerance was a simple but useful predictor of move time.

This logarithmic measure became known as "Fitts' index of motion difficulty." Hill (1976) extended this analysis to movement with a manipulator and made extensive measures of time to completion with different manipulators and tasks with a varying number of degrees of freedom. He concluded that the Fitts index had to be substantially modified for tasks with differing constraints.

On the other end of the development are the machine intelligence models which have broadened the scope of communication problems to include the following: (1) specification of goal and subgoal, (2) cognition of the environment, (3) problem solving (on an abstract level), (4) planning of solution, (5) execution and monitoring of the plan, and (6) procedures for error recovery. All these problem areas have been continuously attacked by Artificial Intelligence researches. Although completely automated, general-purpose manipulation and problem solving are a long way from reality, these models provide useful directions in formulating communication models of advanced man-machine systems.

The most practical approach attempts to develop higher and higher command languages, ultimately reaching the point where the commands will specify complete tasks. For example, Lozano-Perez and Winston (1977) describe LAMA, a high level command language at M.I.T. It solves the problem of object collision avoidance by brute force--requiring extensive three-dimensional modeling of all objects in the work area. This system is, thus, an attempt at solving problems 1, 2, 3, and 4 above. Ambler, et al, (1973) describes a similar computer controlled assembly system. This system was unique in that it could recognize the parts it needs from a randomly dumped heap of parts. It would take the heap apart, put the parts in order, and proceed to assemble them by feel, according to a prepared assembly program. It can be instructed to perform a new task with different parts by spending an hour "showing" it the parts, and a day or two programming the assembly manipulations. Amber's system is, thus, a promising attempt at all the six problems previously stated, but at the current state it requires extensive control of the environment in terms of lighting and background and pre-knowledge of all parts in the heap.

Fike and Nilsson's (1970) STRIPS is an example of a symbolic problem solver that attempted to solve robot location problems in a simple environment. It worked well only on simple problems (in terms of the number of operators required in the solution), and Sacerdoti's (1975) ABSTRIPS achieved some success in pushing the complexity barrier by solving the problem in a hierarchy of abstract spaces. Fahlam (1974) describes a problem solving system for block manipulation which demonstrated high solving power for this idealized world of blocks. The key words, however, are idealized and abstract. These problem solvers cannot, at this state, handle the complexity of real world objects in terms of shape, position relations and function.

More recently, the focus of automatic problem solving/planning has been directed to the representation of knowledge for the particular problem domains (Sacerdoti, 1979) and to the distributed problem solving (Lesser and Corkill, 1979). Possible advantage may be obtained with the use of models and developments of distributed sensor, control, and decision in advanced manipulation (Lyman, et al, 1979; Corkill, 1980). Also, the representation mechanism developed in this area can be adapted to serve as a model for the man-machine communication environment. Thus, a more detailed description of the structure and function of one of the developments, Sacerdoti's procedural nets (1975), is warranted here.

The procedural nets (PN) is the formalism in which tasks are represented in the system. It is hierarchical data structure containing both declarative and procedural, domain-specific knowledge. The declarative part contains the preconditions necessary for an action to be applicable and the result or the effects of the action on the environment (as in STRIPS, Fikes and Nilsson, 1971). The procedural part indicates how this particular task can be achieved in terms of a more detailed, partially ordered, set of actions. When the system uses this representation to advise an apprentice, it develops (plans) a description of the task to be performed from the top level down. It starts with a general statement of the task and expands it hierarchically in successive levels of more detailed descriptions to the level of details that the apprentice can perform as single conceptual units. If he does not know how to do some subtask at the level of detail given, the computer can develop the node in the procedural net associated with this subtask into a sequence of more detailed subtasks. The declaration information in each node is used in the problem solving part of the process. The system uses it to develop a plan that will achieve the stated goal without internal conflicts, starting from the initial state of the world.

The essential aspects of procedural nets relevant to our problem are: the top-down hierarchical representation of tasks; and the fact that the conceptual units communicated between the system and the user are complete tasks at various levels of details. This is contrasted with other systems which dealt only with sets of primitive manipulator motions. Although this system deals with a computer supervisor and a human operator, while we are dealing with the inverted role-play of a human supervisor and a computer-controlled manipulator, the issues of planning, problem solving and communication are similar.

3.3.2 Adapted Procedure Nets Framework. The procedure nets model discussed above suggests a basic structure for the design of man-machine communication. In the NOAH system, the PN is used for task description and as a machine advisor for an apprentice. Thus, the PN are developed hierarchically, by the machine from the top down to the level of details that the apprentice can perform directly. This development is hierarchical in the sense that it can be developed to different levels of details for different parts of the task depending on the level of knowledge of the apprentice about the particular subtask at hand.

In a shared man-computer control paradigm, however, the role of operator and systems need to be redefined. The level at which the man-machine interface occurs determines what was termed the "communication mode" (Verplank, 1967). Figure 3-1 shows some examples of possible placements of the man-machine interface (Sheridan and Verplank, 1978), which is one of the main language design decisions. If it is placed at a very low level, as shown in Figure 3-1A, where the operator controls joysticks which cause link movements, then we have "direct control" and a computer is actually unnecessary. If the communication is done at a very high level, shown in Figure 3-1B, then we have "symbolic control," or, considering the fact that in this case a computer does a large part of the plan development and monitoring, this is sometimes called "automatic control." In controlling a general purpose manipulator, either control mode alone cannot be sufficient. As Verplank (1967) has shown, both analogic and symbolic commands are necessary for effective communication. Some tasks, those completely pre-specified, can better be called by a symbolic command, and others, those involving complex geometric motions that are not repeated, can better be specified by direct control. This is shown in Figure 3-1C. Figure 3-1D considers the situation when man sets up the goal at the top level, computer plots the strategy, and the actual execution is again given to the man. This control allocation is the one adopted in NOAH where the computer knows how to perform a set of tasks, the human apprentice sets the goal, and then requests from the computer the detailed plan (sequence of actions) that will accomplish

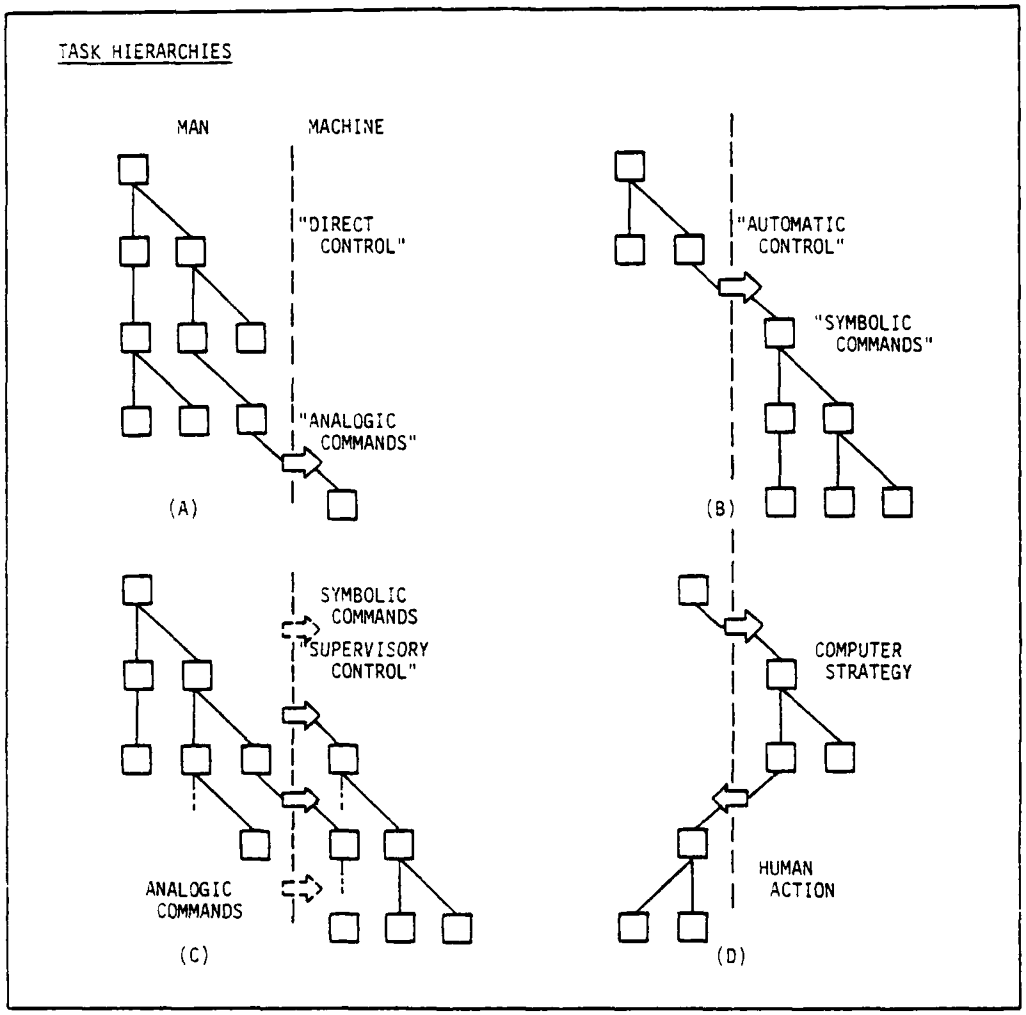


FIGURE 3-1.
ALTERNATIVE TASK HIERARCHIES SHOW THE NEED FOR ALTERNATIVE COMMUNICATION MODES (SHERIDAN AND VERPLANK, 1978).

the goal. He then goes ahead and executes the actions himself. This discussion shows the flexibility needed in the communication model in describing control allocation strategies.

The proposed task model treats an overall manipulation task as a network-like structure, with individual manipulator-environment interaction combined into subtasks, which are then combined into one overall task. Each node of the network structure consists of (1) a goal statement--what must be accomplished by the task, identified by a command or a named chain and a set of completion criteria, and (2) an action--the sequence of subtasks, expressed as lower level of the network structure, identified by the content of the command or chain. This information is represented as a planning procedure in each node of the network. The planning process is activated when a global task request is made at the node in question. The activation initiates the procedures within the selected node, evaluates the current state of the environment and then activates an appropriate sequence and control allocation of subtasks which, when accomplished one after another, by operator or computer or both, will satisfy the goal. These subtasks are then activated and the procedures within each respective nodes are, in turn, developed into a plan of finer levels of detail.

Figure 3-2 illustrates the representation of the task "Shut Valve." At the conceptual level, the task and command description is independent of valve type and valve location and the parameter that is needed at this level is an indication of the type of operation and which valve is operated on. One level lower in the task, "Approach-Grasp-Shut," the valve location and geometry is needed. At the level underneath it, a specific control mode and configuration has to be decided.

This representation of task depicts flexible role assigned between operator and machine. The operator conceives the task at a high level of abstrac-

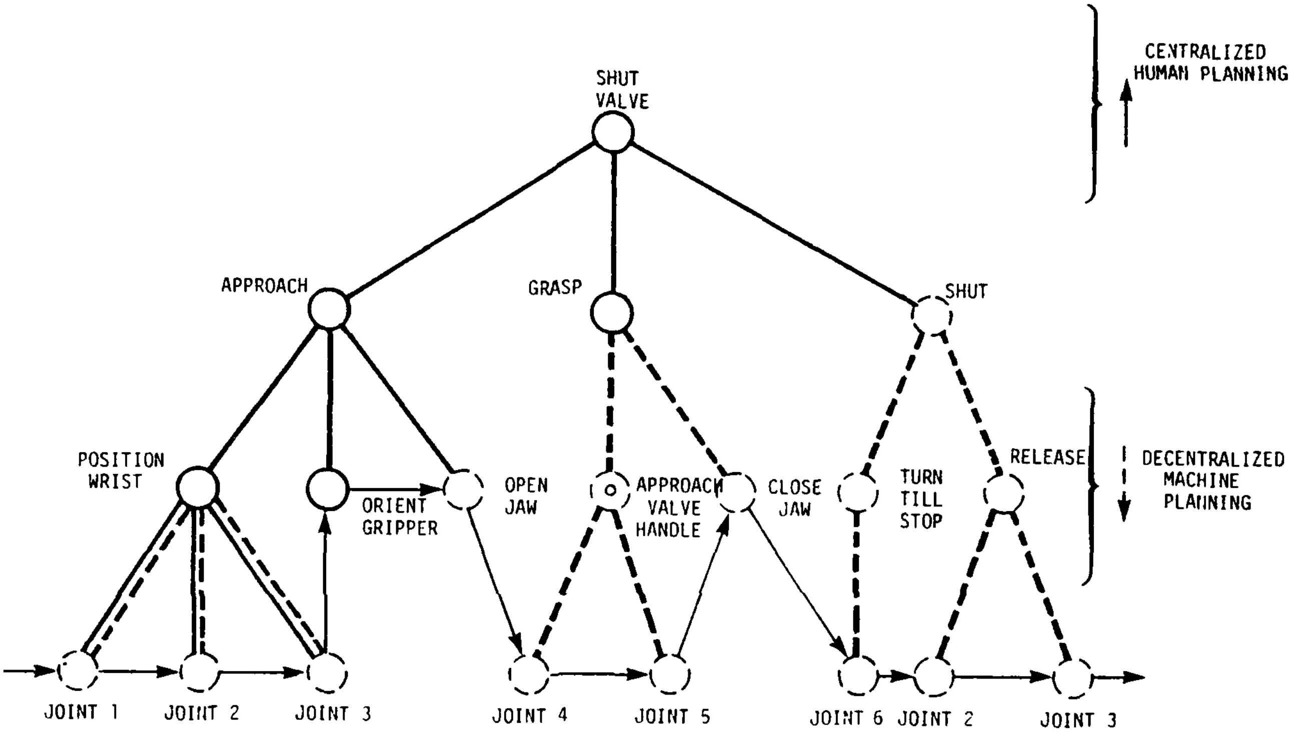


FIGURE 3-2.
PROCEDURAL NETS MODEL FOR "SHUT VALVE" TASK

tion and develops it to some intermediate level, where the machine (computer) can accept. He communicates this intermediate-level plan, which is shown in the figure as the tip-nodes of solid lines, to the computer. The computer, having previously stored the description of each subtask at that level, develops the plan to the level of primitive actions which, as represented in dotted nodes in Figure 3-2, can be communicated directly to the manipulator. The arrows connecting the tips of the procedural nets are the sequencing links between the manipulator primitive action, as observable in a task/environmental display.

The procedural net model suggests a basic structure for the communication language and feedback design. It is a task-oriented command and feedback with provisions to define task and to receive feedback information at suitable levels of complexity. Figure 3-3 illustrates the levels of detail information required for adequate indication of subtask state. The representation also provides multiple level-of-detail description of sub-goal achievement; and each node up the hierarchy requires broader field of view and lower resolution. This common structure of command and feedback not only greatly increases the ease of implementation but also provide gross correspondence between command range and spatial geometry hierarchy.

3.4 Syntactic Analysis of the Task-Oriented Supervisory Command Language

This section describes syntactic and lexical elements of the language along with the keyboard arrangement, feedback display, and the dynamics of the interactions. In Section 3.4.1 we discuss the two types of commands which make up the language: analog commands to specify detailed motions or points in space, and symbolic commands to direct the manipulator to do some primitive or structured task. Section 3.4.2 describes the keyboard and joystick interfaces through which the commands are activated and transmitted to the machine. The form and arrangement of the keyboard

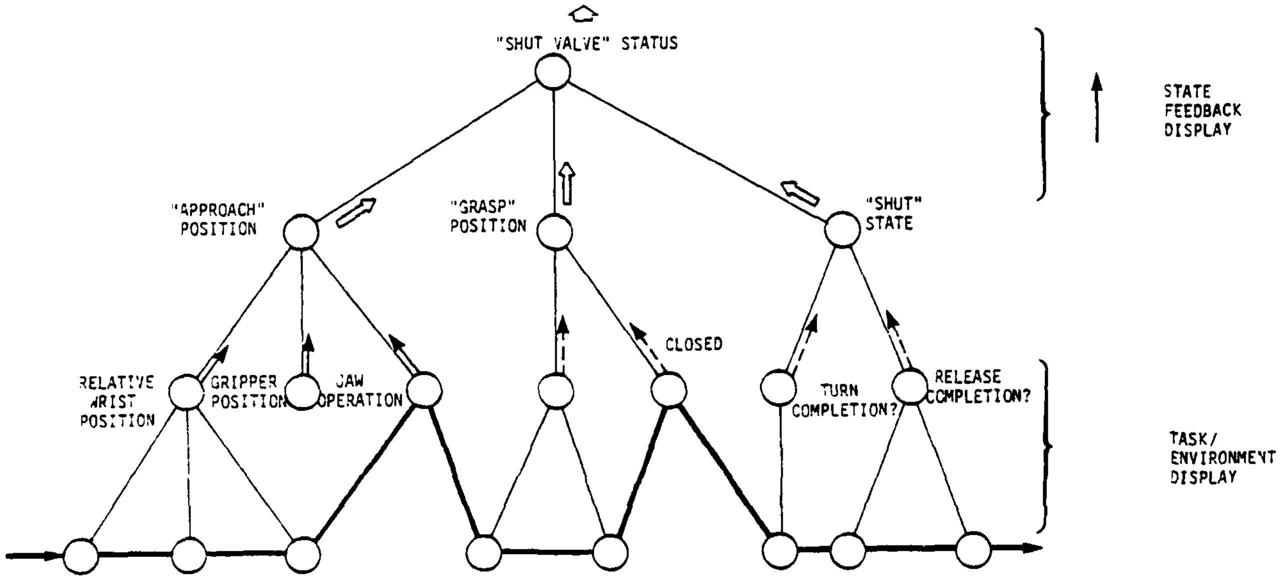
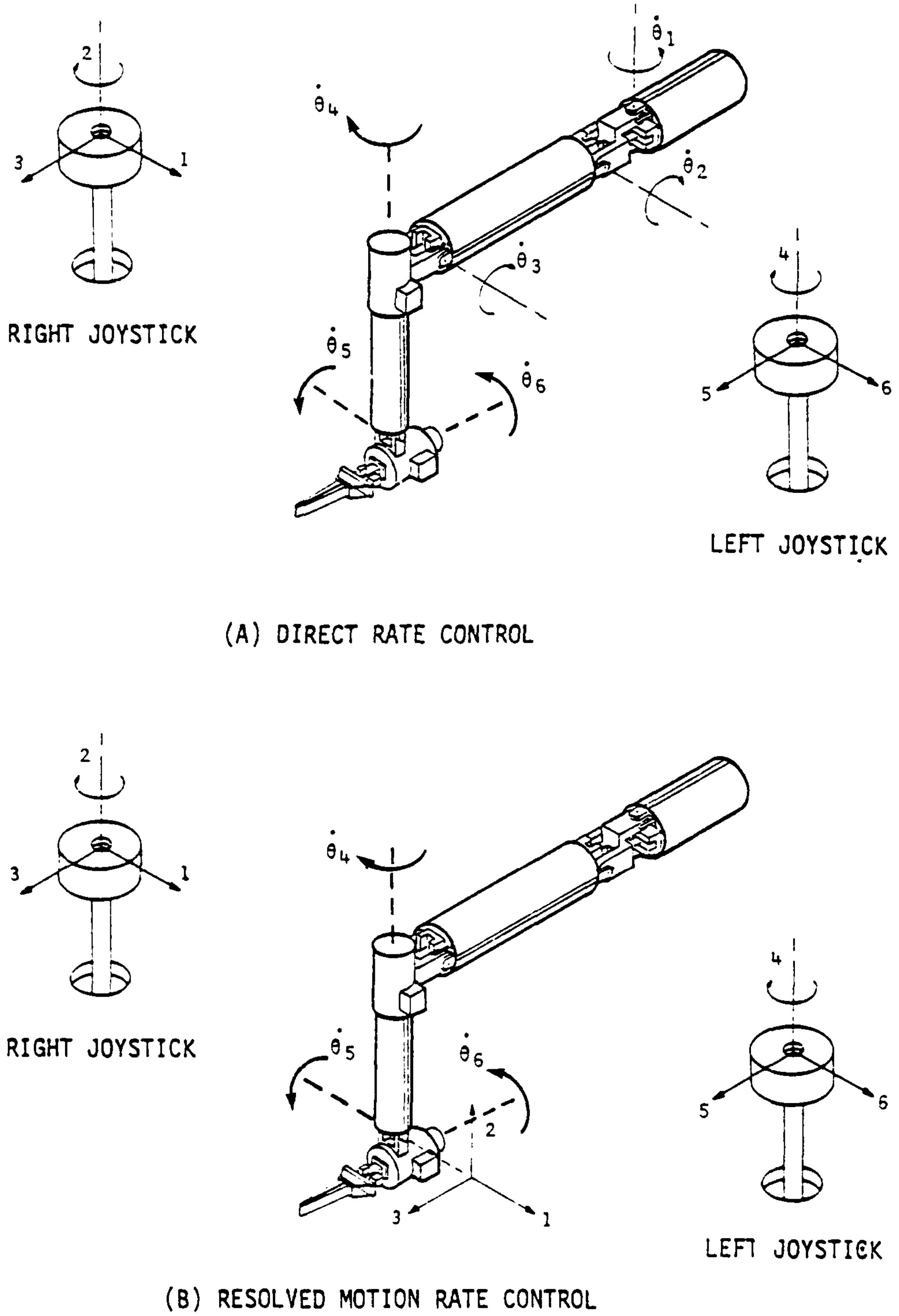


FIGURE 3-3.
INFORMATION FEEDBACK REQUIREMENT FOR "SHUT VALVE" TASK

are strongly related to the syntax and commands of the language. The keyboard arrangement is designed for visual and tactile continuity and for a physical realization of "sentence structure." Section 3.4.3 describes the language primitive concepts and lexical elements from points and paths to primitive motions and actions. Section 3.4.4 covers the hierarchical structuring capabilities of the language--namely chains. This capability allows on-line, real-time construction of automated subtask tailored by the user to the specific task at hand. Finally, Section 3.4.5 summarizes the command language features and characteristics.

3.4.1 Analog Versus Symbolic Commands. A basic feature of the manipulator command language is the capability of specifying by analog inputs points and paths in link space. The analog commands are complementary to the symbolic commands that comprise most of the command language. The symbolic commands are the subject of the rest of the sections in this chapter. The physical interfaces that provide these analog inputs are two three-degrees-of-freedom joysticks. The assignment of joysticks variables to arm motions have two modes that can be selected by the user, the "joints control" mode, and the "spatial control" (RMRC) mode.

In "joints control" each joystick variable is assigned to a specific manipulator link. The right joystick is assigned to the upper three links: shoulder rotate, shoulder raise and elbow flex. The left joystick is assigned to the lower three links: elbow rotate, wrist flex and wrist rotate. Figure 3-4A shows these control assignments. The operator's right hand is doing the major manipulator moves and the left hand is doing the final gripper orientation. Through previous experience it was found that the direct control mode is useful in subtasks where a particular manipulator configuration is sought (such as in stowing the manipulator) or a particular orientation of the gripper is needed.



(A) DIRECT RATE CONTROL

(B) RESOLVED MOTION RATE CONTROL

FIGURE 3-4.
JOYSTICK MOVEMENTS AND CORRESPONDING MANIPULATOR MOVEMENTS

In "spatial control" mode the variables of the right-hand joysticks are assigned to control the motions of the wrist in the x_1, x_2, x_3 cartesian space. The assignment is made to be as natural as possible: right-left (x_1) motions are controlled by the right-left joystick variable; forward-backward (x_3) are controlled by pushing the joystick away or toward the operator; and up-down (x_2) motions are controlled by rotating the right joystick. The allociations of the left joystick variables are not changed. Fibure 3-4B shows these control assignments of the two joysticks. Experimental results indicated that this control mode is useful for most general motions in the manipulator's work space--like moving around an obstacle on the way to the tool-box. Experiments done in the past further showed that both control modes were necessary. The command to shift control between the joint control and the spatial control modes is given by the following key sequence:

$\boxed{\text{SPATIAL JOINT CONTROL}}$ $\boxed{\text{DO}}$

One of these two control modes is always active, and the activation command switches between the two. The joysticks are rate controls and have a small nonactive zone around the spring loaded zero point. When they are moved out of this small area they cause motions as described above. Manipulator motions can be caused by either pure analog commands or by superimposed analog and symbolic commands. This "live joystick" feature allows real time small corrections to be done by the operator while the manipulator is controlled by the computer to perform some automatic task.

3.4.2 Keyboard Arrangement and Feedback Format

The close relation between the command language and the features of the keyboard and feedback-display-screen require a description of the keyboard arrangement and the feedback format. A special function keyboard was designed and built to optimize speed and smoothness of interaction between the operator and the system in sequence-structured commands. That is, all commands are intended to have a consistent form of a sentence. Each sentence corresponds to one conceptual task unit and is given in one sequence of key pushing from left to right. A typical command is the following:

$\boxed{\text{GO TO}} \quad \boxed{\text{POINT}} \quad \boxed{8} \quad \boxed{\text{DO}}$

which causes the manipulator to move to the pre-defined point 8. The syntax of all commands in general is as follows:

<verb> <noun> <label> <terminator>

The verb or the label or both may be omitted in some cases, as will be discussed in the next section. The keys, however, are organized in groups and the groups are placed in this order from left to right. Figure 3-5 shows the details of the key arrangement. Each group of keys is separated by physical location and a color coding.

The first column of keys from the left, (1) in Figure 3-5, is the verb group. They specify what action has to take place. The verbs used are:

GO TO, DEFINE, DELETE, and END

These keys are surrounded by a green color as indicated by horizontal line shading.

The second column of keys, (2) in Figure 3-5, is the noun group or the basic concepts of the language. They are:

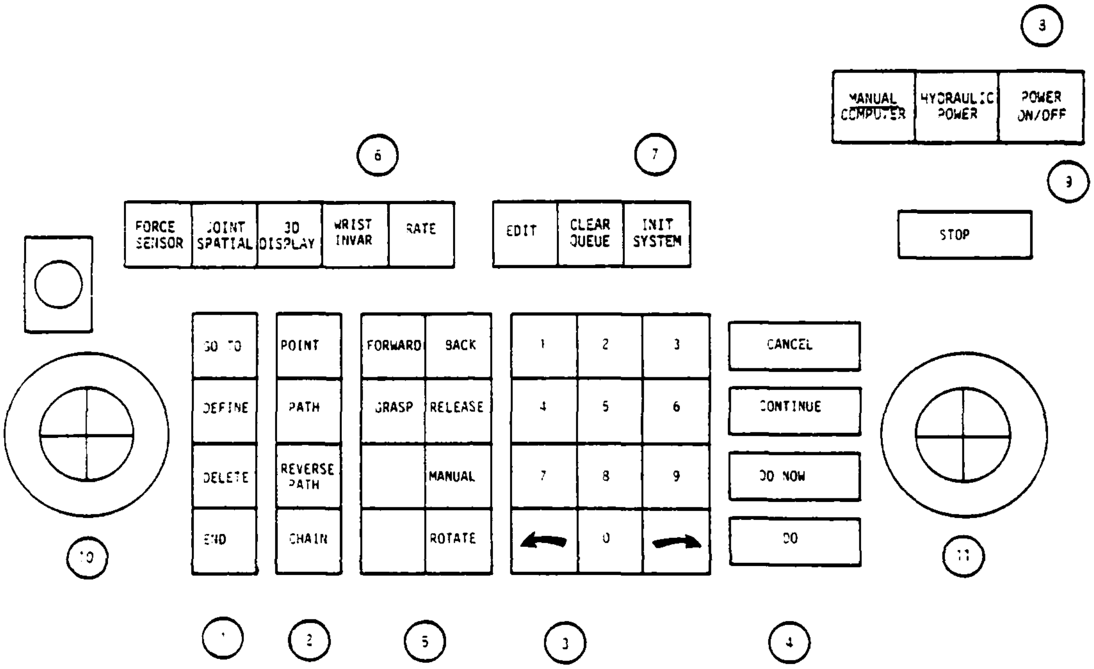


FIGURE 3-5.
DEDICATED KEYBOARD ARRANGEMENT

POINT, PATH, REVERSE-PATH, CHAIN

By separating the verbs from the nouns, 16 combinations are made available with only 8 keys. Furthermore, the verb-noun relation is natural for the operator.

The fourth group of keys from the left, (3) in Figure 3-5, is the label or parameter group. These keys are used to label or refer to a particular concept such as POINT, PATH or CHAIN. The system allows 10 points, 10 paths, and 10 chains to be defined and used. Alphanumeric labeling was not used for three reasons: one, to keep the keyboard small and comfortable to use--in terms of resting the hands on the two joysticks with a comfortable distance between them; two, to limit the number of points and paths etc. that can be defined so as not to overload the operator's memory; three, the number of keys to be pushed for a command would be increased if multi-number or multi-letter labels were allowed. Previous years' experience have shown that if the commands are too long the operator would probably resort to direct joint control of the manipulator and would not take advantage of symbolic commands.

The last column of keys from the left, (4) in Figure 3-5, are the terminators. Each command must be closed with a terminator; they are a signal to the computer to start an action. Up to that point, the user can cancel his command by pushing the "CANCEL" terminator (recovery from errors). The terminators available are:

DO: Is the regular command completion key and is used in all commands.

CANCEL: Indicates that an error was made in the command given and causes the computer to clear and ignore the command.

CONTINUE: Is a command to transfer from MANUAL mode to AUTO mode after that has been interrupted by a STOP command in the middle of a chain execution.

DO NOW: Indicates that the command entered has to be performed before all those pending in the execution queue. A STOP command has to be issued earlier to stop the execution of the chain in progress.

The third keypad from the left, marked (5), includes the command keys that do not relate to the nouns (or concepts) in the language. Most of them do not need a label, either. These commands are:

FORWARD, BACKWARD, GRASP, RELEASE, MANUAL, ROTATE

Only ROTATE uses parameter keys. These are ↶ and ↷ which are located in the label group, marked (3).

The other groups of keys above the main keypad have secondary importance; they cause mode changes and switch system states. They, too, are grouped together by their function.

The state variable keys are grouped together at the top left side of the main keypad, marked (6). They include:

SENSOR, SPATIAL/JOINT-CONTROL, FROZEN-WRIST, RATE

They are used to change the various parameters in the control state. Details will be given later. The syntax of these commands is the same as other commands. For example, to change the rate of manipulator motion to rate #3, the command is:

$\boxed{\text{RATE}} \quad \boxed{3} \quad \boxed{\text{DO}}$

Clearing, or reset, keys are grouped together at the top right, marked (7) in Figure 3-5. They are:

RESET, CLEAR, QUEUE, CLEAR

The STOP key is the overriding emergency key and is located separately for easy access near the right hand joystick, where the operator's hand usually rests. It is also used whenever a command must be given during AUTO mode. It causes a stop in motion and a transfer to MANUAL mode.

The two joysticks are located on either side of the main keypad, marked (10) in Figure 3-5.

Visual information about the state of the system and the progress in task execution is displayed on the CRT. The extensive feedback of information is important to keep the user informed about the processes in progress. To provide visual continuity the screen is divided into fixed fields and during all states of the communication process each contains its own type of information. Figure 3-6 shows the various display fields on the CRT.

USER INPUT FIELD: The commands entered by the user are immediately fed back to him, completely spelled out. This is in addition to the tactile and visual feedback provided by the keyboard itself. It is useful for command verification (before the terminator is used) and error identification.

ERROR MESSAGE FIELD: In case of a syntax error (the user enters an illegal sequence of keys), an error message will immediately appear in this field indicating the error and its type, and the command is ignored. The closeness of this field to the previous one on the screen again provides visual continuity.

SYSTEM STATE FIELD: The current state of the different systems options are continuously displayed in this field. These options and variables are the following:

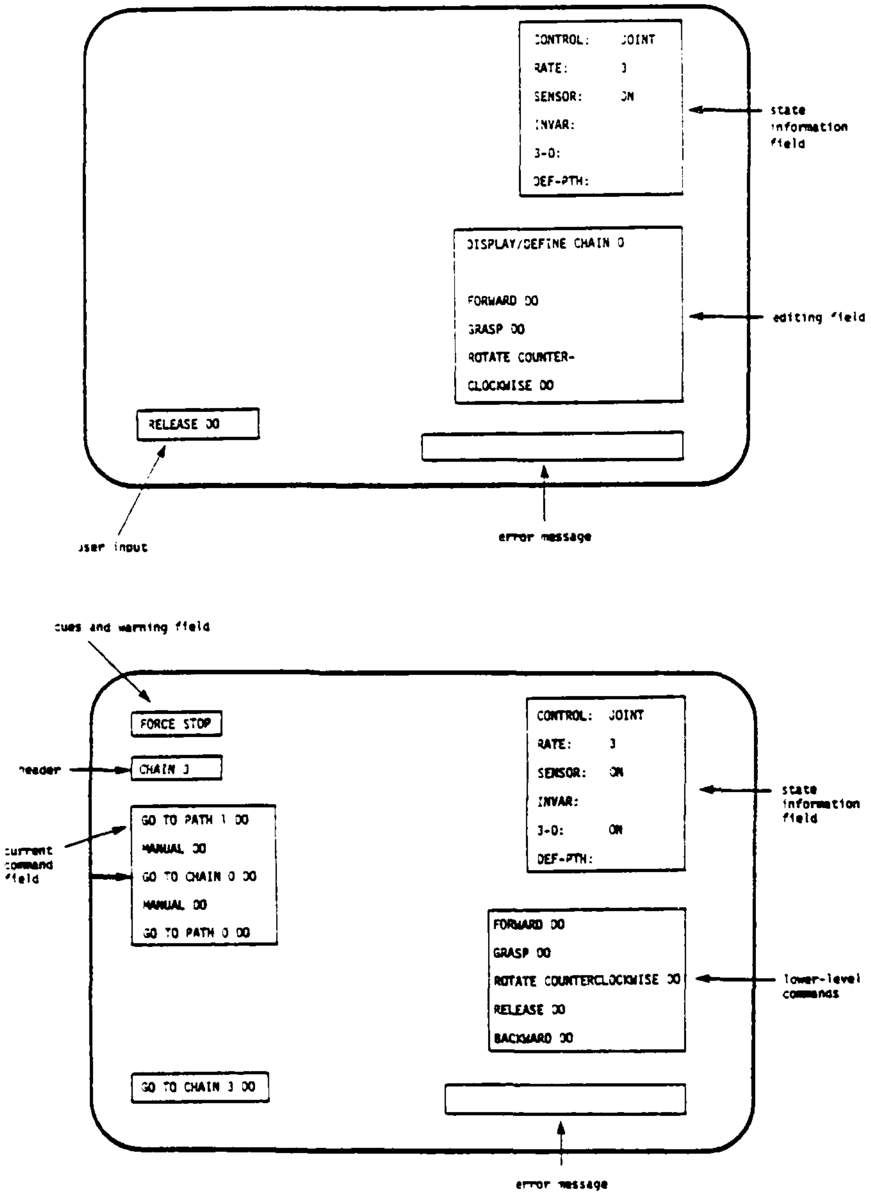


FIGURE 3-6.
STATE FEEDBACK FORMAT DURING CHAIN DEFINITION (TOP)
AND CHAIN EXECUTION (BOTTOM)

STATE: Whether AUTO, MANUAL or EDIT.
CONTROL: JOINT CONTROL or SPATIAL CONTROL.
RATE: Can be one of 1, 2, 3.
SENSOR: Whether the wrist sensor is ON or OFF.
3-D: Whether the 3-D display is ON or OFF.
WRIST: Whether FROZEN-WRIST or OFF.

CLUES AND WARNING FIELD: The system displays YOUR TURN, with accompanying tone, when control is transferred to the user in the middle of an automatic sequence. It is also used for emergency warning. For example, when the computer initiates an emergency stop to all motions, a STOP appears in this field.

CURRENT COMMAND AND HEADER FIELDS: Display the name and sequence of command in the chain now being executed. The specific current command is always pointed out in the list, and the pointer moves down as execution progresses.

CHAIN EDITING OR DETAILING FIELD: This field is used to show the scope of chain or command while being edited or executed under editing phase and execution phase, respectively. The various display activities and formats will be described in more detail in the following sections with corresponding command and machine states.

3.4.3 Language Primitives. The primitive commands are the conceptual unit tasks useful in undersea manipulations which are automated and can be called by the user in one symbolic command. The function performed by each command, the syntax of the command, the format of the user feedback on the CRT, and the available error messages, will be given for each command.

Points. Labeled points in the language correspond directly to points in link space or, in user terms, to a particular configuration of the manipulator. Points are defined by actually bringing the manipulator to the required configuration (including wrist position and gripper status--open or closed). A point is accessed by a GO TO command and the label can be erased or redefined in another configuration by using the DEFINE command again. The system can handle up to 10 points labeled from 0 to 9.

DEFINE POINT n

Syntax: DEFINE POINT <n> DO WHERE $0 \le n \le 9$

The current manipulator configuration is stored under the label n where $0 \le n \le 9$. If a point with the same label existed before, it is redefined to the current configuration. An example of the command is the following:

$\boxed{\text{DEFINE}}\quad\boxed{\text{POINT}}\quad\boxed{5}\quad\boxed{\text{DO}}$

Each box represents pushing one key.

GO TO POINT n

Syntax: GO TO POINT <n> DO $0 \le n \le 9$

The manipulator will move according to a computer generated sequence of small increments from its current position to the point labeled n. The path taken is an approximation of a straight line in link space. The command can be used to define spatial way-point and then to expedite the motion from subtask to subtask at high speed.

DELETE POINT n

Syntax: DELETE POINT <n> DO $0\leq n\leq 9$

The assigned value is deleted for the label n, and POINT n becomes undefined.

Display Related to Point. The following are examples of the messages that appear in the user message field as he presses the keys for the commands given above:

```
DEFINE    POINT    5    DO
GO TO     POINT    5    DO
DELETE    POINT    5    DO
```

Error Messages Related to Point. Error messages appear in the typing error field after the DO button is pressed. They identify the following errors:

If superfluous keys have been entered:
ERROR IN COMMAND

If no label (number) is given:
POINT NAME MISSING

If a GO TO POINT is made to an undefined point:
POINT n UNDEFINED.

Paths. Labeled paths in the language correspond to defined sequences of points stored in the computer. Paths are defined by setting starting time, going through the required path, and then terminating the path. The process of going through the required path can be manual or prestored, analogic or symbolic, with the computer taking all necessary interpolation between consecutive spatial-point specifications. All motions of the manipulator are stored, both those controlled manually and those controlled by the computer while executing a previously defined chain. The computer samples the manipulator's configuration about every half second and stores the values in sequence. Because of storage limitations, paths are limited to 1 minute in length. The values of the label can be in the range 0≤n≤9. A path is defined by using the DEFINE PATH and END PATH commands. It can be accessed in two ways, by a GO TO PATH command, and by a GO TO REVERSE-PATH. The provision for running both forward or backward on a path is useful to move from one area of the workspace to another and back. A path can be deleted by a DELETE PATH command, or changed by redefining it.

DEFINE PATH n, and END PATH

```
Syntax:  DEFINE PATH     <n>     DO     0≤n≤9
         END PATH                DO
```

The path of the manipulator motion is stored under the label n (0≤n≤9), by sampling it every half second. The sampling starts when the DEFINE PATH command is completed (with a DO); and ends when the END PATH command is completed, or when 60 seconds elapses since the start, whichever comes first. All the manipulator motions in this time interval are recorded, whether controlled manually or automatically by the computer. This provides the flexibility to construct complex (both static and dynamic) trajectory by calling a chain and some GO TO POINTS into a combined path.

GO TO REVERSE-PATH n

```
Syntax:  GO TO     REVERSE-PATH     <n>     DO     0≤n≤9
```

This command is identical to the GO TO PATH command, except the predefined path *n* is traversed from end to head. As indicated before, it is useful for example, for going around an obstacle to the tool box and back.

DELETE PATH n

Syntax: DELETE PATH <n> DO 0≤n≤9

The path with the label n is deleted from the list of available paths. A path can also be deleted by defining a new path with the same name.

Display Related to Path. The following corresponding messages will appear in the user message field when DO is depressed in the previous examples:

DEFINE PATH	6	DO
END PATH	6	DO
GO TO PATH	6	DO
GO TO REVERSE-PATH	6	DO
DELETE PATH	6	DO

Error Messages Related to Path. Error messages identify the following mistakes with a message in the error field:

If superfluous keys have been entered:
ERROR IN COMMAND

If no label number is given:
PATH NAME MISSING

If GO TO PATH is attempted to an undefined label:
PATH n UNDEFINED

Other Primitives. In addition to the POINT and PATH concepts, there are several commands that correspond to useful elementary/autonomous motions. These have been predefined as symbolic commands and are represented by the following commands:

ROTATE

```
Syntax:  ROTATE      <↶|↷>       DO
```

The manipulator wrist is rotated up to 180 degrees around the gripper long axis according to the direction indicated by the parameter, ↶ or ↷. The rotation is continued until the end of θ_6 travel is reached or a resisting torque greater than a threshold T_R is encountered. In this latter case, the motion is stopped and a warning is issued.

The appropriate display is fed back to the user in the user message field:

```
ROTATE     RIGHT     DO
ROTATE     LEFT      DO
```

GRASP, RELEASE

```
Syntax:  GRASP       DO
         RELEASE     DO
```

In the GRASP command the gripper is closed up completely or, if something is placed between the gripper prongs, it is closed until the object is grasped to a fixed threshold force. In the RELEASE command the opposite happens. The gripper is opened up to a maximum travel or until a force prevents it from further movement. These are examples of commands without the full *VERB-NOUN-PARAMETER* sentence structure. The feedback that are displayed to the user are:

```
GRASP      DO
RELEASE    DO
```

FORWARD, BACKWARD

Syntax:	FORWARD	DO
	BACKWARD	DO

In the FORWARD command, the gripper is moved forward along its axis of rotation to a distance equal to the length of the gripper's prongs. For a longer motion, the command is repeated. The command is useful in approaching an object for a grasp. According to the configuration of the manipulator, the computer calculates the link motions necessary to cause the gripper to move forward. In the BACKWARD command, the motion is reversed. The gripper is moved backward along its axis of rotation to a distance equal to the length of the gripper prongs. The direct feedback displayed to the user is the key sequence:

FORWARD	DO
BACKWARD	DO

MANUAL, CONTINUE

MANUAL is a special primitive command that is not useful by itself (it causes no manipulator motion) but is very useful within chains. It is included in the sequence of commands comprising a chain where the operator's intervention is expected. During chain execution, when this command is encountered, the automatic, computer-controlled motion is stopped and control is relinquished to the user. Thus, this command provides a means to transfer control from machine to man in the middle of an automatic sequence. The user pushes CONTINUE to give control back to the computer after he has done his part. The key sequence is:

MANUAL	DO	during chain definition
CONTINUE		after the operator has done his part in execution

Notice that CONTINUE is a terminator by itself and can be used in other cases where command has to be returned to a suspended automatic sequence. These cases include the expected events (e.g., inserted MANUAL), or unexpected events (e.g., a simple reflex or force stop).

3.4.4 Chains. Chains are the language structural element that corresponds to complete subtasks. They are sequences of specific primitive commands (or other chains) which were defined and labeled by the user as useful repeated subtasks. They provide the user with the capability to define a useful sequence only once and later use it repeatedly by calling it as a unit. The computer then performs that stored sequence automatically, faster and with less probability of errors. It also provides the capability to construct tasks by combining a sequence of simpler subtasks in hierarchical construction. For example, suppose a valve has to be closed and opened several times ; it might be expedient to define a chain for this task. Suppose, further, that PATH 4 was defined previously as a path to the particular valve and CHAIN 2 was defined to be a chain which approaches and rotates a valve once. Then the task of closing the valve completely can be performed by the following chain which we call CHAIN 9:

```
CHAIN 9
     GO TO       PATH        4
     MANUAL
     FORWARD
     GRASP
     GO TO       CHAIN       2
     GO TO       CHAIN       2
     RELEASE
     BACKWARD
     GO TO       REVERSE     PATH      4
```

PATH 4 is traversed forward in the beginning of CHAIN 9 and backward at its end. By using the MANUAL command the computer will momentarily relinquish control to the operator after it has brought the gripper into the vicinity of the valve. This allows the operator to make small adjustments to bring the gripper exactly in front of the valve. Thus, the adaptive interpolation is still left for the operator. After FORWARD and GRASP, CHAIN 2 is called twice to perform the actual rotation of the valve. This is an example of hierarchically using a chain within a chain. RELEASE opens the gripper and the last command brings the manipulator back to the initial position. A PATH is used in this example to bring the manipulator to and from the valve, making it possible for the manipulator to go around obstacles with which it might have collided, if GO TO POINT were used.

The commands used to handle chains are similar to those related to paths. Chains can be defined (and redefined), called for use, and deleted. The commands causing these actions largely resemble those for the path. In fact, chains can be considered a symbolic equivalent of paths, where paths contain stored analog sequences; chains contain a sequence of symbolic commands which may be primitive paths or even other chains.

DEFINE CHAIN, END CHAIN

```
Syntax: DEFINE     CHAIN     <n>     DO     0≤n≤9
        END        CHAIN     DO
```

The DEFINE CHAIN and END CHAIN commands are the de-limiters that indicate the start and end of the chain being defined. During the definition of the chain, no manipulator motion takes place--the system is in EDIT state.

In EDIT state, new chains are defined or old ones are updated. When defining a chain consisting of solely symbolic commands, the process of definition can be made either by going through the chain once and storing the sequence or, utilizing its symbolic nature, going out of execution mode and storing the required sequence without actual execution. The latter approach was used more frequently because with no manipulator motion the user can concentrate on the sequence of commands needed and can do it quicker--as fast as he can key in the commands.

The user enters the EDIT state by starting a chain definition using the DEFINE CHAIN COMMAND. If the chain already exists it will be redefined; otherwise, a new chain will be established. The user can then enter the sequence of the commands making up the chain in the order of their required execution. The commands entered are displayed for verification in the chain editing field. As the user enters his commands, one after the other, he can check the correctness of the entry which is first displayed in the user input field. If a mistake occurred, he can erase that line by pressing CANCEL instead of DO. Commands are added to the developing chain when DO is pressed. Current implementation provides no better correction capabilities outside the line correction. If the user wants to replace a command which he has already entered or to insert a new command in an earlier part of the chain, he has to redefine the complete chain. The user can then conclude the chain definition and exit from EDIT state by issuing END CHAIN command. This causes the CHAIN definition to be closed and the new chain to be stored under the appropriate label. No check is made by the computer as to the correctness of the chain. (Better editing facilities should probably be developed in the future.)

An additional feature that is provided is the display of a previously defined chain for viewing without modification. If the operator wants to observe the content of a chain or to remind himself of its function, he may issue DEFINE CHAIN, and the chain is displayed. Issuing END CHAIN brings him back to execution state.

GO TO CHAIN n

Syntax: GO TO CHAIN <n> DO $0 \leq n \leq 9$

Chains are activated by issuing this command. The activation of a chain is equivalent to a subroutine call where control is delegated to the computer within the duration of the chain's execution. Chain names are numbers in the range $0 \leq n \leq 9$, and there is no relation between a chain, a path, or a point with the same label. When a chain is called, the content of the chain appears on the screen. The chain primitives are executed one after another and the chain definition is scrolled on the screen so that the currently active primitive is shown at the top of the list. Figure 3-6 also presents an example of the display format during chain execution.

At any point in the chain execution, the user can add correcting commands to the computer generated motions by manipulating the joysticks to cause the desired additive motion. For example, if a chain command causes a motion toward the side of a valve, he can correct the motion while it is in progress by using the joysticks. Sometimes there is a need to make a fine alignment manually in the middle of an automatic chain execution. The user has the option to STOP the motion of the manipulator, perform the manual subtask, and then press CONTINUE to give control back to the computer. Automatic execution will commence from the point where the chain execution was stopped.

There are two commands that may appear in a chain and deserve special attention. One is the command which calls another chain such as: GO TO CHAIN 7. This allows chains to call chains to any depth. This is the mechanism which embodies the hierarchical structure of the chains. The second special command is MANUAL which, when encountered during execution, temporarily relinquishes control back to the user. Here we have the mixed initiative feature where the computer temporarily gives control to the user in the middle of an automatic sequence. Control is returned to the computer when the user issues the CONTINUE command.

DELETE CHAIN n

Syntax: DELETE CHAIN <n> DO $0 \leq n \leq 9$

The chain with the label n is deleted from the list of defined chains and becomes undefined.

3.4.5 Summary of TOSC Language Features. In comparison with Sacerdoti's procedural nets implementation, the task-oriented supervisory command language has the following features:

(1) Supervisory planning by operator instead of machine.
(2) No preprogramming or prestructuring required.
(3) On-line, real-time programming.
(4) Bottom-up aefinition of command by user.
(5) Added communication interface: dedicated command sets, sentence structure, and multi-mode feedback, etc.
(6) Added control input and dynamic queue of commands.

The command construction is straightforward and easy to use, as it can be described by the following simple, context-free rules:

MANIPULATION: = {Command|Chain}

Chain : = CHAIN ID + {Command|Chain}

Command : = Symbolic|Analogic

Analogic : = MANUAL + $[\dot{\theta}_1|\dot{\theta}_2|\dot{\theta}_3|\dot{\theta}_4|\dot{\theta}_5|\dot{\theta}_6]$

Symbolic : = Switch|Fixed|Variable

Switch : = Sensor|Joint|Rate|3-D|Wrist

```
Fixed          : = Rotates|Forward|Backward|Grasp|Release
Variable       : = Point|Path:Rev-Path
Path:Rev-Path: = PATH ID +  Path|Point + Path|Chain
Point          : = POINT ID + [Spatial position]
```

The rules are written using the following syntax: | is disjunction, {...} is composition including ordered conjunction and repetition, and [...] is composition including unordered conjunction and repetition. Therefore, these rules state: "A manipulation consists of a group of single commands and/or chains. A chain in the group is a solid chunk of motion/maneuvers directed by the computer and monitored by the operator. A chain consists of chain declaration, followed by a group of single command(s) or chain(s). A single command is either symbolic or analogic. An analogic command is a manual setting of analog control devices along major axes: Θ_1, Θ_2, Θ_3, ...,Θ_6, etc. A symbolic command is either a switch command, a fixed symbolic command, or a variable symbolic command. A switch command activates one of the selection switches including sensor, motion rate, motion coordinate, wrist invariant, and 3D-display controls. The fixed command may initiate one of the unitary motions, such as wrist rotates, forward, backward, grasp, and release. The variable command refers to the user-defined symbolic commands including spatial points and paths. A path command consists of path declaration followed by continuous trajectory, manually guided or internally assembled, using other paths, points, or chains. A point consists of point declaration followed by a marked spatial point."

4. EXPERIMENTAL STUDY

4.1 Overview

This chapter summarizes the results of a set of experiments performed in the program years to investigate the efficacy of various communication factors and the best mode of communication. Specific objectives included the following:

(1) Examine man-machine communication needs and evaluate the effects of system factors influencing man-machine communication.

(2) Investigate the relations between operator performance and computer aiding along three major dimensions of communication--control mode, command structure, and feedback display.

(3) Provide data for evaluating overall system effectiveness contributed by the level of automation provided in a shared man-computer control paradigm.

Proper evaluation and assessment of a complex man-machine system must include three aspects: (1) capability and quality (e.g., dexterity, capacity, precision, etc.) that can be achieved, (2) the corresponding resource consumption and loading for achieving a given performance (e.g., power, effort, and attention), and (3) system utilization and efficiency (e.g., throughput, frequency of use, allocated functions and probability of success, etc.). Most of the quantitative data obtained in the series of study are based on the performance time and error. Performance time data is widely used in the evaluation, due to the ease of measurement and its responsiveness to system variation. Under a different control of the experiment, performance time can reflect system capability (speed), resource consumption (processing time), or system utilization/efficiency (time occupancy, and time ratio). For reasons of economics and ability

AD-A094 482 PERCEPTRONICS INC WOODLAND HILLS CALIF F/G 5/8
MAN-MACHINE COMMUNICATION IN REMOTE MANIPULATION: TASK-ORIENTED--ETC(U)
MAR 80 Y CHU, W H CROOKS, A FREEDY N00014-76-C-0603
UNCLASSIFIED PFTR-1034-80-3 NL

2 OF 3
AD
A094482

to control key variables, the work discussed below was performed using the laboratory facility described in detail in Appendix B.

A convenient approach toward the communication mode assessment is based on performance improvement due to the added functions in the communication dimensions of interest. The selected functions of interest included the following: (1) augmented control represented by resolved-motion rate control (RMRC), (2) traded control represented by automatic motion control (AMC) using defined points and paths, (3) command mix of fixed and variable primitives, (4) command abstraction level represented by chain-in-chain structure, (5) display degradation due to environmental factors, (6) level of details in machine state feedback, and (7) display format. The evaluation approach was one of observing the incremental improvement in performance, which could lead toward the optimal design of man-machine communication with well-defined performance indices.

4.2 Evaluation of Man-to-Machine Communication

The principles and guidelines for language design described in the previous chapter were obtained through analytical studies, model construction, and principles suggested in related fields. A first step to validate these guidelines and principles is to perform experiments in the simulated laboratory environment (see Appendix B). The experiments were conducted following the design, implementation, and modification of the language features, which, in its final form, was described in Chapter 3. The experiments, using 6 to 12 paid subjects, emphasized the effect of automation (computer-aiding) on three levels: lexial (control and programming), syntactic (command mix), and semantic (command organization) levels:

(1) Control and Programming. Is the effect of the basic aiding functions which raised one level of abstraction in

communication. Two dimensions considered are (1) augmented (serial) control, and (2) traded (parallel) control. The levels of the first dimension were (a) individual joint rate-control, and (b) resolved-motion rate-control. The levels of the second dimension were (a) direct control, and (b) automatic motion control, where the operator could call on a predefined point in symbolic, push-button control.

(2) Command Mix. Is the effect of the lowest level of command structuring, i.e., the sequencing and grouping of the commands including point, path, and fixed primitives in sentence structure. This represented a mixing of symbolic/ analog commands and serial and parallel computer aiding. The levels of this dimension were (a) basic commands, where the operator could call on symbolic primitives in manual sequence, (b) variable commands, in which the language provided a facility to define in real time variable commands as groups of mixed symbolic or analogic primitives.

(3) Command Organization. Is the effect of the higher level of command structure, i.e., the chain-in-the-chain function of the command and appropriate manual check-point for higher level task oriented commands. The levels of this dimension were (a) fixed commands, same as the basic commands above, (b) chained commands, in which the language provided a facility to define multiple levels of chains of a lower level of commands.

Two versions of control/display station were used in the studies: direct viewing and indirect viewing. Figure 4-1 shows a drawing of the more complete, newer version of the control/display station which consists of the

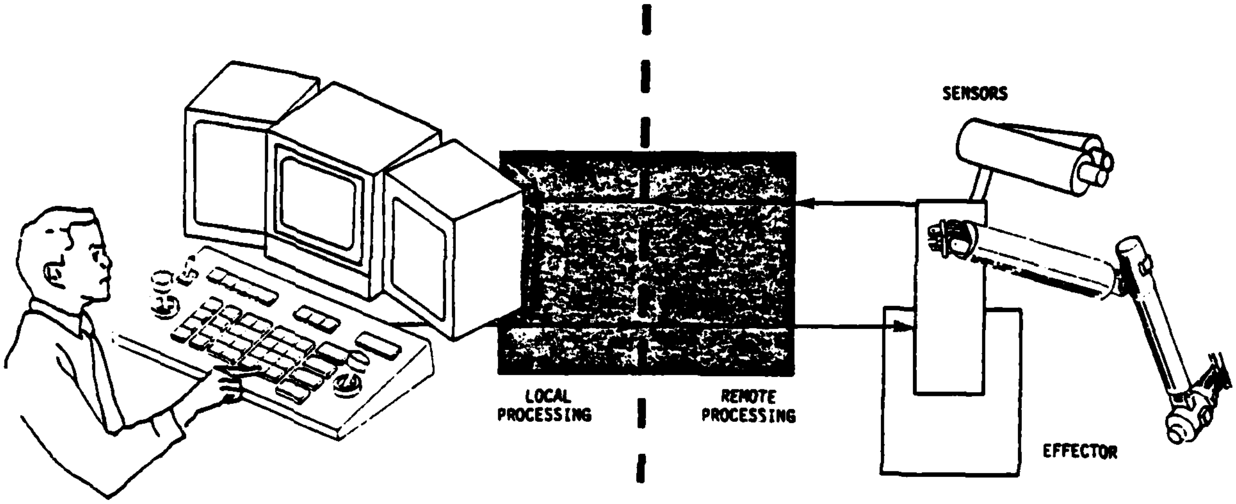


FIGURE 4-1.
TELEOPERATOR CONTROL/DISPLAY STATION

dedicated command keyboard and two three degree-of-freedom, rate control, self-centering joysticks. A 9" CRT display was used to provide alphanumeric messages concerning the condition of the system, the keyboard inputs, and the next command in the execution queue. A 2-view black-and-white TV viewing system added in later experiments was placed in front of the control panel to represent an indirect viewing situation.

The experimental studies reported in the following sections were divided into training and test phases. In the training phase the participants practiced basic manipulation and some constituent elements of the experimental tasks under specified conditions. In the test phase, the participants were required to apply the skills they developed during training to perform a simulated maintenance task using the pipe structures (Figure 4-2). The simulated task incorporated a representative sample of task elements that are frequently performed in underwater manipulation (e.g., orient and position the end-effector, gross travel, vertical and horizontal alignment, grasp and release objects, etc.), into a single integrated task with a defined goal. The participants were allowed to plan their strategy formally or informally before each experimental session.

Performance measures consisted of task performance time and error committed by participants in performing assigned tasks. When traded control was used, both task *definition* time and *execution* time were recorded to provide an estimate of the overall task *completion* time.* Since the overall maintenance task can be easily divided into subtask elements, the performance of the major subtask element, the valve turning task, was also evaluated and serves as a comparison in the following sections.

The initial intent of this evaluation was to provide an operational context for testing the features of the communication language developed analytically. As such, the experimental study was more a tool for the development of the language design guidelines than a rigorous demonstration

* Definition time is the time spent in constructing user-defined variable commands (i.e., when the operator is giving instructions solely to the computer). Execution time includes the time in activating commands, (i.e., passing instruction to manipulator through computer) and the time in monitoring and controlling of manipulator motion. Overall completion time is simply the interval from the time the task goal is given to the time the goal is reached.

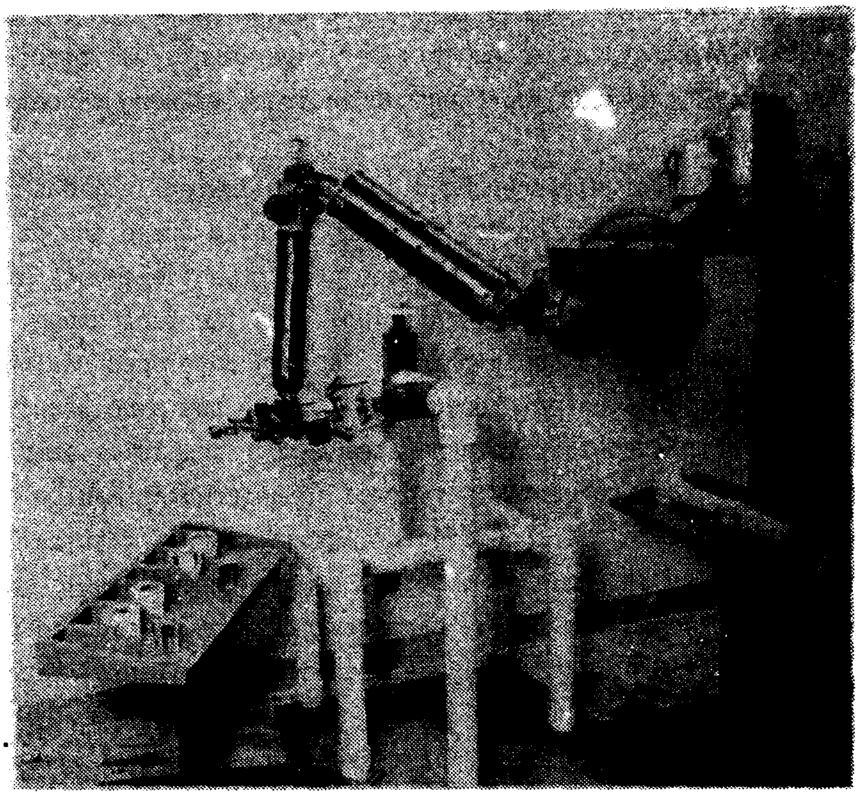


FIGURE 4-2.
VALVE TURNING TASK

of absolute system effectiveness. It would be difficult to compare the results in absolute terms with other experiments conducted on different manipulators, different tasks, and differently trained operators.

4.2.1 Control and Programming. The initial year's study (Berson, et al., 1977) evaluated separate effects of two basic aiding techniques: the resolved motion rate control (or spatial control) and the automatic motion control (i.e., go-to-point using gross control buttons). Subjects were required to perform a simulated maintenance task based on an initial setup of pipe structure. It contained two gate valves and arms, four sear rings, and a storage box for seals. The 2x2 design for control and programming evaluation included the following conditions:

(1) Direct control - unaided joint control.
(2) RMC - Control of manipulator wrist motion along X-Y-Z. Coordinate axis was provided by the right joystick.
(3) Direct plus automatic motion control (AMC), where the computer automatically moves the manipulator to preassigned locations.
(4) RMC plus AMC.

Figure 4-3 presents the task execution time under each of the four control modes. The overall comparison between direct control and RMC indicated that the participants performed significantly better under RMC than they did under direct control ($P < 0.025$). No significant main effect was obtained for AMC and the two-way interaction was not statistically significant. Analysis of the valve turning data indicated that while task execution times were less with AMC than without it, the difference was not statistically significant. The analysis indicated that the need to define extra spatial points for proper operation and the inefficiency in activating go-to-point function buttons reduced the advantage gained in the use

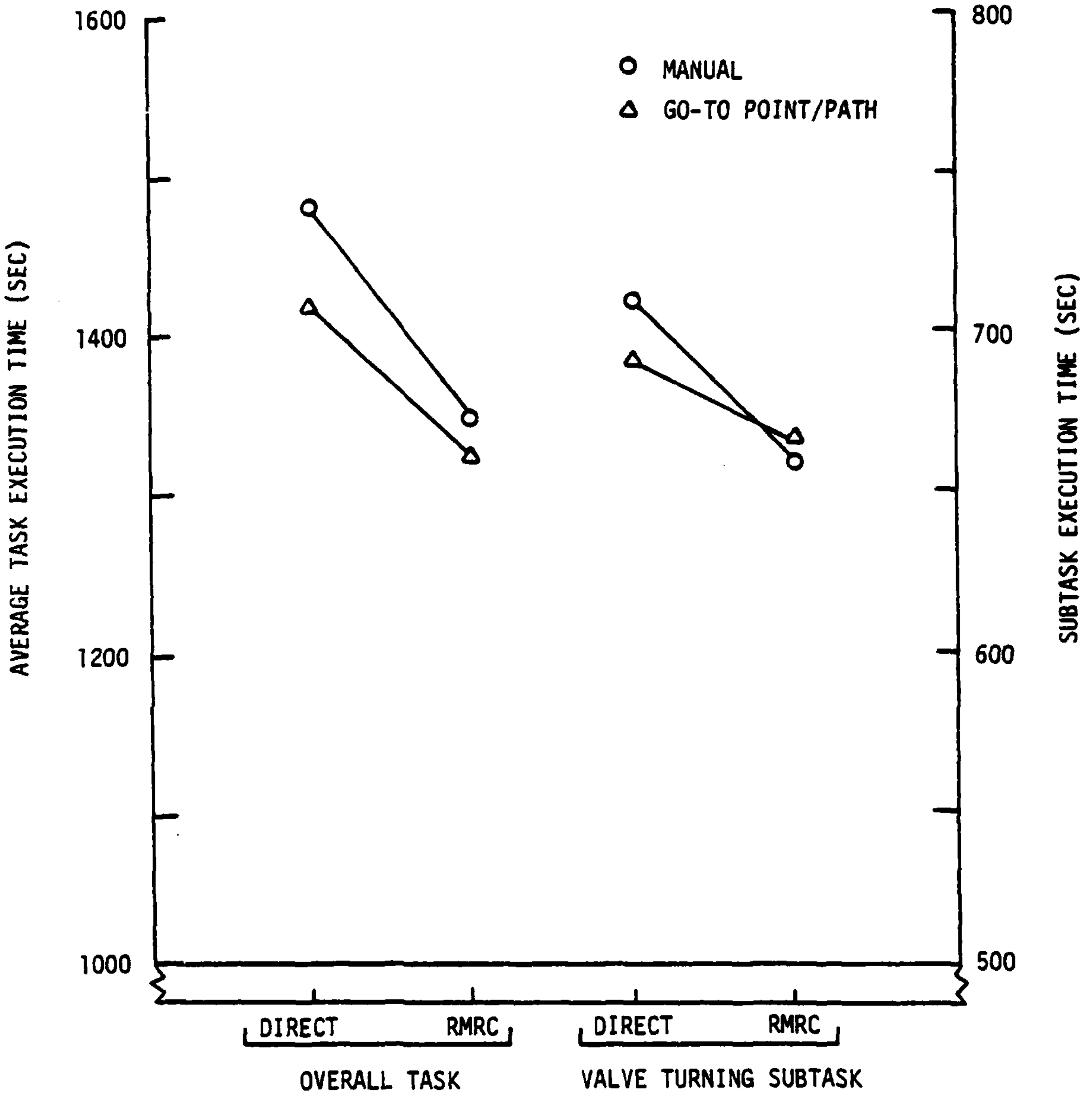


FIGURE 4-3.
AVERAGE TASK EXECUTION TIME AS A FUNCTION OF
BASIC CONTROL AND PROGRAMMING MODES

of AMC. The study concluded that augmented control, in the form of resolved motion of the wrist joint, almost universally reduces task performance times and error rates as compared with unaided manual control. This performance improvement is particularly high for tasks requiring fine, dexterous control and simultaneous multi-joint control. The automatic motion control could be useful for tasks with many repetitions or for tasks with few repetitions but with low definition and activation time.

4.2.2 Command Mix. The second year study (Crooks, et al., 1978) evaluated the potential benefits of mixed analogic and symbolic commands which could be fixed symbolic or user-defined. Concerning the structure of various means by which the operator transmits instructions to the remote manipulator, previous experience suggests the following command structure:

(1) When a variety of tasks are to be performed, a multi-mode supervisory control is needed for the operator to select from a variety of computer-aided control modes.

(2) Command structures should allow both analogic and symbolic primitives in serial or in parallel, and allow smooth transition among control modes.

(3) Selection of any control function should be as simple as possible, involving:

(a) A single command to request a control function.
(b) One command that initiates one unitary element.
(c) An interface that minimizes the time and effort to select the command.
(d) Automatic selection of control mode where feasible.

(4) When control functions must be performed in sequence, automatic functions should provide appropriate interpolation in actions or the interface should prompt the operator with the next required step.

The experimental study has selected to investigate the main effects of command mix in two levels: (1) Basic mode, which allowed the use in sequence of both analogic and low-level, unitary motion, symbolic primitives, and (2) Variable mode, which allowed in the command sequence the user-defined variable commands, including Point, Path, and Chain. A 3x2x2 repeated measures design was used to examine three task conditions, with two levels of command structure, over two trials. The three task conditions were Baseline, Increased Variability (with more non-repeatable subtasks) and Increased Complexity (with more repeatable subtasks). The experimental procedure allowed variable commands defined in Trial 1 to be used in Trial 2.

Figure 4-4 presents the task performance time as a function of command structure and task condition. The effects of task and trial number were significant ($P < 0.01$). This is mostly due to the definition time overhead in the first trial and the performance gain achieved in the second trial within variable command mode. There were slight improvements in overall task execution time using variable commands; the effect was not significant. This is reflected as the slopes of solid lines in Figure 4-4. Actually, the upward slopes of the dotted lines represent the definition time overhead, while the downward slopes of the solid lines represent the performance gain in execution time. The comparison between task and valve-turning subtask shows that the performance of the valve-turning subtask closely parallels the overall task performance except that the valve-turning subtask has very low definition overhead in general, and has high performance gain in high-complexity tasks, which has increased repeatable

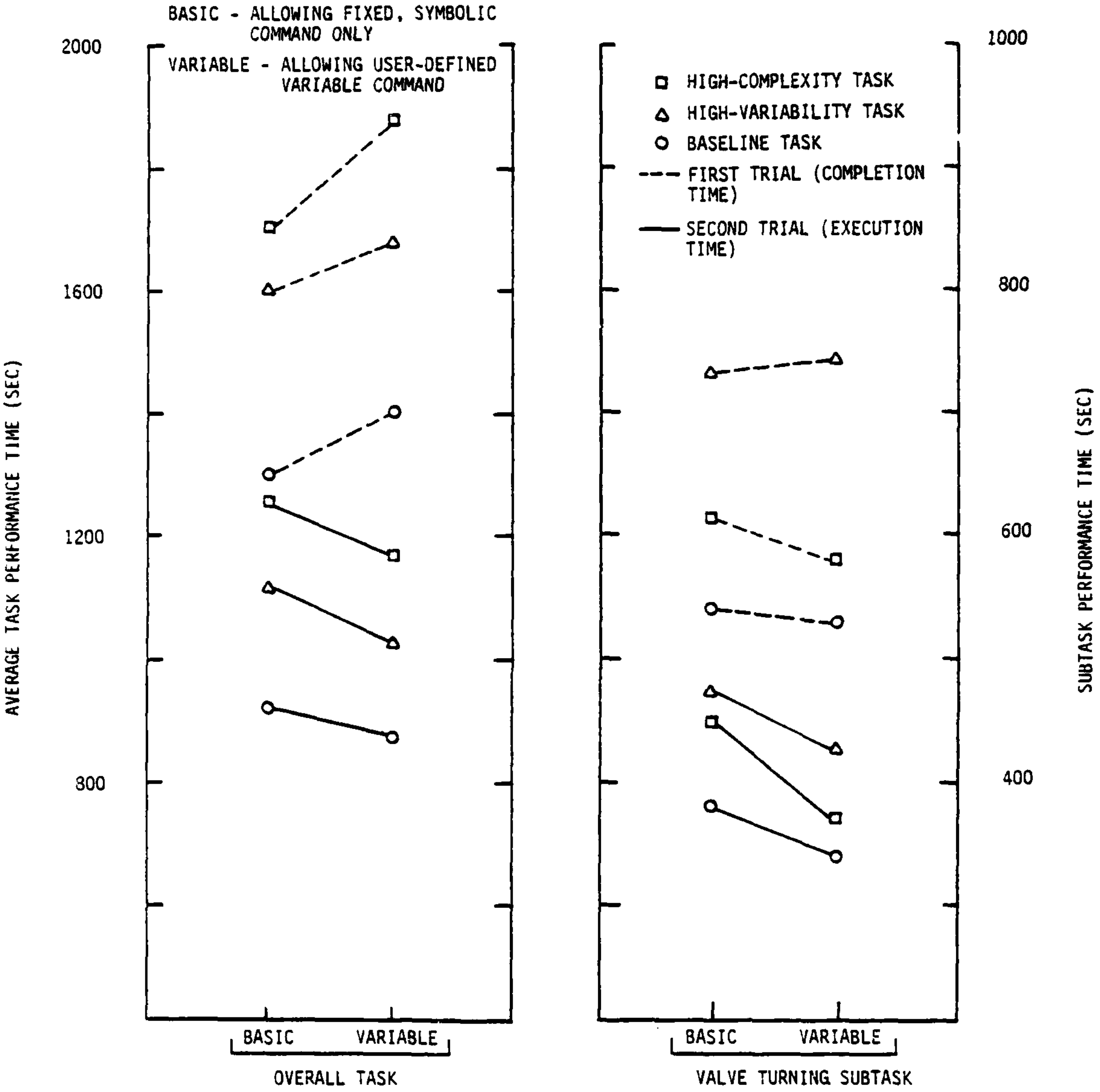


FIGURE 4-4.
AVERAGE TASK PERFORMANCE TIME AS A FUNCTION OF COMMAND STRUCTURE AND TASK STRUCTURE

task elements. This was thought to be due to the discrete and repetitive nature of the valve-turning subtask which, when using appropriate variable commands, can be easily defined by a group of symbolic commands and executed automatically without operator intervention. The results demonstrate the potential benefits to be gained by using variable commands in tasks which must be repeated one or more times. Better improvement can be expected in procedural repetitions than in trajectory repetitions, as the latter ones usually require higher definition time and demand more operator intervention than the former ones.

4.2.3 Command Organization. The third year study (Crooks, et al., 1979) evaluated the effect of higher level command organization, mainly the use of and the confidence associated with multiple-level chain-structure. Previous experience suggests the following command organization in a multimode supervisory control paradigm:

(1) The communication should take place at the highest symbolic level comfortable to the user. Giving commands in too much detail would cause the user to lose sight of the overall task while dealing with the details. One the other hand, giving a command at a gross level does not allow adaptation of the command to the peculiarities of the situation at hand.

(2) High level command construction should be achieved using hierarchical organization of user-defined variable commands. This structure could expedite modularity, parallel processing, and error isolation in both task definition and execution. A manual facility should be incorporated in the system that can be used easily in any level of hierarchy.

(3) By providing a manual request, computer prompting, real-time trajectory editing, and hierarchical feedback of machine

state, a mixed-initiative control allocation between operator and computer can be achieved.

(4) Composite constructs (chains in symbolic and paths in analogic) should be easy to use, involving:

(a) A definition capability.
(b) A labeling capability.
(c) A display for review capability.
(d) An editing capability.
(e) An activating procedure.
(f) A progress display capability.

The experimental study, conducted in an indirect viewing situation evaluated performance at two levels of command organization: (1) Fixed command, where no user variable command was used, and (2) Chained command, where command organization up to two-to-three levels of chaining was allowed. A 2x2 repeated measures design was used to examine the effects of command mode under two levels of visibility. The visibility effect was established using the contrast and brightness controls to maintain 59% and 17% contrast ratios for normal and degraded viewing conditions.

The average task performance time data are presented in Figure 4-5. The results of a within-subject analysis of variance on task *execution* time indicated that participants performed significantly better (faster and more consistently) using Chained commands than they did using Fixed commands ($F = 52.66$, $P < 0.001$). This represents a 51% reduction in execution time with the normal visibility and a 60% reduction with the degraded visibility ($t = 2.97$, $P < 0.05$). The visibility effect was found significant in the Fixed command mode in which the participant took an average of 17% longer time to complete the task with the degraded TV than with the

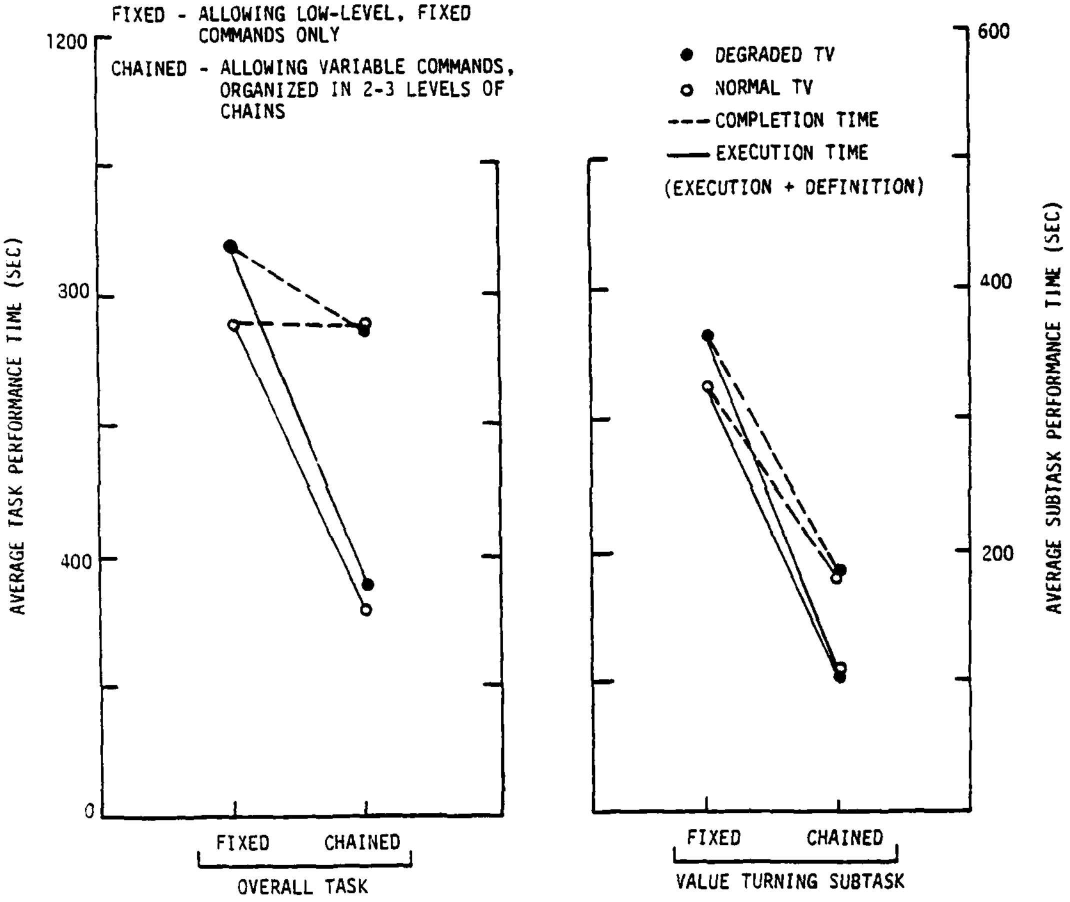


FIGURE 4-5.
AVERAGE TASK PERFORMANCE TIME
AS A FUNCTION OF COMMAND LEVEL

clear TV ($t = 2.61$, $P < 0.025$). The effect was much reduced in the Chained command mode, to the degree that the use of Chained commands had effectively maintained normal level of performance within the visibility ranges tested. In other words, the Chained command is potentially more effective in degraded than in normal visibility. In fact, a significant 18% reduction in task *completion* time ($t = 2.76$, $P < 0.025$) was obtained with the Chained command, the first evidence ever obtained for significant reduction of overall completion time with various traded control modes.

Separate task performance analyses were conducted for three major subtasks to provide a comprehensive description of the contribution of each subtask to total task performance. A compa-ison based on subtask characteristics, command utilization, and resulting performance was made. It showed that the valve turning subtask was the one receiving higher pay-off in the use of Chained commands, judged by its high task time reduction (49%), its high time benefit/overhead ratio (between 3.75 and 3.45 for the conditions tested), and its significant error reduction. This was due to a proper match of symbolic commands with a task of repetitive and structured nature, which may be further characterized by trajectory discreteness and simplicity, precision requirements, and environment uncertainty.

4.2.4 Summary of Man-to-Machine Communication. A comparison was made of time performance improvement for various command/control modes based on execution time or completion time ratios. As shown in Table 4-1, performance ratios vary significantly among control/command modes, task types and environmental conditions. The first observation is that the high-level chain commands, along with a complete set of language features, had the highest improvement in task execution, especially in degraded viewing situations. The degree of improvement varies among the type of tasks. The maintenance task is the most complex task, with the highest number of steps, with few (1 to 2) repetitions on spatial point/trajectory, and

TABLE 4-1.

PERFORMANCE TIME RATIOS FOR VALVE TURNING, RING PLACEMENT, AND MAINTENANCE TASKS

CONTROL MODES*	TEST PHASE	TIME RATIOS**		
		MAINTENANCE TASK	VALVE TURNING	RING/CAP PLACEMENT
RMC/DIRECT	1	0.91^{+}	0.99	0.78^{+}
(AMC+RMC)/RMC	1	0.98	0.93	(1.10)
VARIABLE/FIXED	2 2	0.95 (1.08)	0.88 (0.95)	0.56^{+} (1.20)
CHAINED/FIXED (NORMAL VIEWING)	3 3	0.44^{++} (1.02)	0.33^{++} (0.57)	0.53^{++} (1.16)
CHAINED/FIXED (DEGRADED VIEWING)	3 3	0.40^{++} (0.84)$^{++}$	0.30^{++} (0.51)$^{++}$	0.43^{++} (0.90)

*DIRECT - MANUAL RATE CONTROL
RMC - RESOLVED MOTION RATE CONTROL
AMC - RECORDED POINT/PATH MOTIONS
FIXED - BASIC PRIMITIVES PLUS RMC
VARIABLE - POINT/PATH PLUS FIXED
CHAINED - CHAIN-IN-CHAIN PLUS VARIABLE
(Note that although no formal comparison between Fixed Mode and RMC mode was made, it was observed that only small (up to 10%) performance time advantage was obtained for Fixed Mode over RMC Mode)

**BASED ON EXECUTION TIME (WITHOUT PARENTHESIS) OR COMPLETION TIME (WITH PARENTHESIS). Ratio of 1.0 = Equivalent performance for the tested conditions. For the definition of execution time and completion time, see Footnote on page 4-5.

$^{+}p < 0.05$

$^{++}p < 0.005$

with highest level of procedural structure. The valve turning task is the most discrete task with several low-level motion repetitions, and with the highest percentage of symbolic command usage. The ring/cap placement task required multi-joint trajectory and few (2-3) repetitions of trajectory. From system effectiveness point-of-view, RMC was the most effective in ring placement task where coordinated multi-joint control was required. The single Go-to-Point command was not particularly effective until it was incorporated in a structure used with Go-to-Path and other symbolic commands which existed in the variable command mode. With these, great time reduction (of 44%) in the execution of ring placement tasks was achieved, as the task required full trajectory repetition. However, the definition time overhead was so high that the total completion time was 20% higher than the fixed command.

Several factors were at work in performance improvement with the use of Chain command, compared to Fixed command. They are: high-level structure of task procedure, motion and procedure repetitions, use of Point and Path, and Point/Path repetitions. As shown, the overhead of Point/Path procedure definition was high in the maintenance task and ring placement task, and low in the valve turning task. Thus, the resulting execution and completion times were significantly lower in this command mode for the valve turning task, while the use of Chain command was marginally effective for the ring placement task.

To further clear up the working factor in Chained command modes, Table 4-2 summarizes performance time ratios for four levels of preprogramming over task procedure or task trajectory. As expected, the biggest gains can be achieved with both procedure and trajectory can be reliably preprogrammed (4-2D) followed with preprogrammed trajectory alone (4-2C). Significant gains can also be achieved with preprogrammed procedure alone in degraded visibility or with highly structured task (4-2B). The true

TABLE 4-2

PERFORMANCE TIME RATIOS:
CHAINED COMMANDS VS. FIXED COMMANDS

(A) NO PREPROGRAM

STRUCTUREDNESS VISIBILITY	LOW	MEDIUM	HIGH
CLEAR	1.16	1.02	.57**
DEGRADED	.90	.84*	.51**

(B) PREPROGRAMMED CHAINS

S V	LOW	MEDIUM	HIGH
CLEAR	.98	.85	.46**
DEGRADED	.76*	.71**	.41**

(C) PREPROGRAMMED POINT/PATHS

S V	LOW	MEDIUM	HIGH
CLEAR	.71*	.61*	.45**
DEGRADED	.57*	.53**	.40**

(D) PREPROGRAMMED POINT/PATH/CHAINS

S V	LOW	MEDIUM	HIGH
CLEAR	.53**	.44**	.33**
DEGRADED	.43**	.40**	.30**

* $P<0.05$

** $P<0.05$

achievement of Chain command over Fix command was with highly-structured tasks or tasks under degraded visibility (4-2A).

4.3 Evaluation of Machine-to-Man Communication

Two issues were of major concern in evaluating communication from machine-to-man: machine state and workspace feedback requirements. The symbolic command language alters the cognitive and task requirements of the manipulator operator. Most of the new requirements, as described in Chapter 2, involve communicating with the computer. Especially in multi-mode supervisory control, the operator not only provides direct analog control of the manipulator's movements, but also (1) selects and confirms available computer-assisted functions, (2) determines what control mode is currently operating, and (3) monitors the progress of automated routines, etc. The purpose of the first experiment was to determine the effect and the amount of machine-state feedback required when using the multi-mode control and high-level commands.

Another consideration is the effect of environmental factors in underwater manipulation, such as sediment, turbid water, failure of external lighting, or poor angle of view, which often results in poor visibility. In these cases, laboratory studies and practical experience have shown that performance is severely degraded--frequently to the point where it is impossible to proceed (Pesch et al, 1971; Estabrook, 1975). One suggested means to compensate for poor visibility, as demonstrated in previous sections and other researchers (e.g., Sheridan et al, 1979), is to automate some manipulator motions or procedures. Such preprogrammed computer controls can proceed "orderly" once they are initiated. One important question is how different are the display requirements for various task automation levels.

The intent of the evaluation studies to be discussed in the following sections was to provide an operational context for examining the feedback display requirements in the use of various command features. Similar to the studies discussed previously in Section 4.2, the objective of these studies was one toward performance improvement, rather than one toward optimal selection of communication modes. The question, then, is how to define and to establish the improved feedback with a given command mode. The variables of interest were chosen to represent the candidates for system modification, which might present potential performance bottlenecks. The performance bottlenecks refer to the situations when the performance measures of interest are expected to respond or improve substantially only when some of the modifiable parameters are varied within a certain region. As such, the studies discussed in this section were a diagnostic and relative evaluation in nature, and represented the required observation of possible machine-to-man communication bottlenecks in the current language design.

Six to eight subjects, trained in using various control modes, were used in the experiments. To provide the capability for multi-mode visual feedback, the operator's display/control console was equipped with a 3D-stick-figure graphics, a 2-view black and white TV viewing system, and a 9" state-feedback monitor (see Figure 4-6). The variables of interest included the following:

(1) Machine state feedback level.

(a) Gross state--display of current command only.
(b) Detailed state--display of queue list of commands and current command.

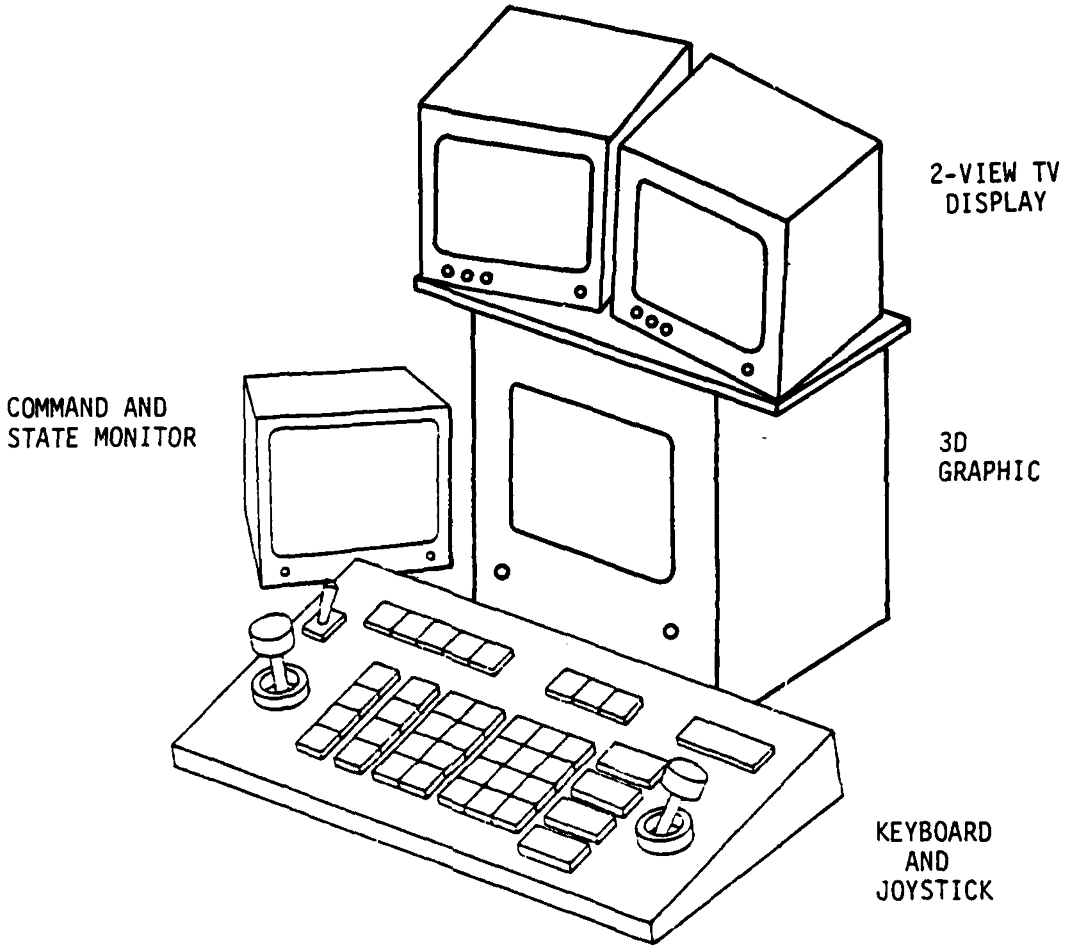


FIGURE 4-6.
DISPLAY CONSOLE IN ENVIRONMENTAL
FEEDBACK EXPERIMENT

(2) Visibility.

(a) Normal viewing--normal illumination and contrast.
(b) Degraded viewing--low illumination and contrast.

(3) Display type.

(a) Alphanumeric.
(b) Stick-figure graphic.
(c) TV video.

4.3.1 Machine State Feedback Level. An experiment was conducted to determine the appropriate scope and level of machine-state information presented to the operator (Crooks et al, 1979). In the earlier design, status of automatic functions and motion sequences were updated sequentially at a conceptual level parallel to the commands used. The gross level of feedback identified only the specific low-level command in definition or in execution. The detail level of feedback included the full queue of commands in definition or in execution. The subjects performed three sequences of five subtasks involving valve turning and ring placement operations. The results of the average task definition times and execution times as a function of feedback level and test session are shown in Figure 4-7. There were three trials in task execution and significant learning effects were found. Therefore, only the data of the third trial in the execution phase are presented here. The analysis showed that only the effects of section ($P < 0.01$) were significant. Although performance time seems to be reduced by detailed feedback in the first session of the definition phase and both sessions of the execution phase, the performance time differences between feedback conditions were not significant. This was due to the presence of the two-view, normal TV-feedback of task status, the effect of machine state feedback observed became

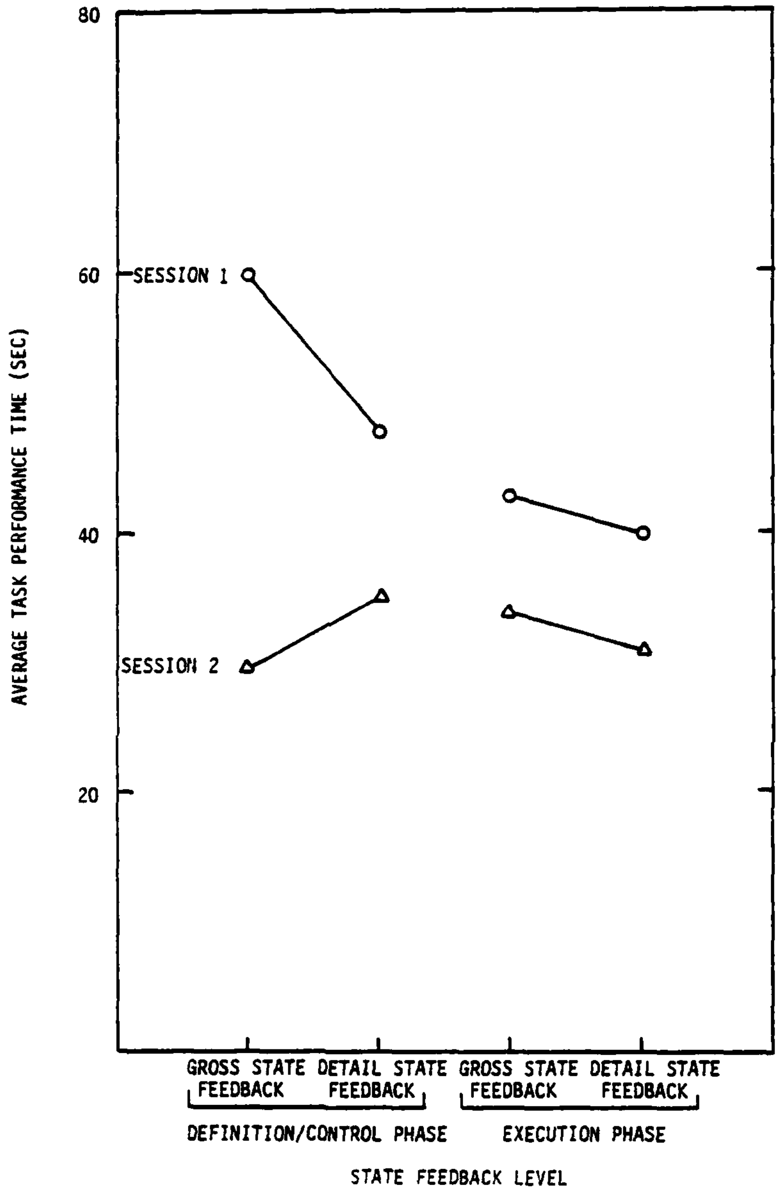


FIGURE 4-7.
AVERAGE TASK PERFORMANCE TIME AS A
FUNCTION OF STATE FEEDBACK LEVEL

secondary. Besides, subject's opinion indicated the difficulty in keeping up with the speed of updating low-level command sequence with the use of detailed state feedback. A later design using the display window and moving arrow received much better acceptance from the subject.

4.3.2 Machine State and Environmental Feedback Formats. Visual observation of a remote manipulator task has been a critical factor in machine-to-man communication in almost all previous experimental work. As we move toward automated manipulation, it is logical to ask whether the typical natural pictorial display (direct or TV viewing) can be replaced by a more abstract display of events at the manipulator workplace without degrading system performance. Three perspectives were considered:

(1) Availability and advances in graphic display technology provide an excellent potential for efficient man-computer communication. The degree of success of this approach (i.e., to what extent a typical natural pictorial display can be replaced by an abstract display of events at the manipulator workspace without seriously degrading system performance) should be tested against available technology.

(2) A typical computer-generated display would require high sensing and processing capabilities and large bandwidth to duplicate even a simple natural scene. With system limitations and cost under consideration, it is essential to evaluate the sufficiency of reduced-cue presentation.

(3) Display requirements may be different for various task automation levels. Proper methods for display synthesis remain to be developed commensurate with task requirements and operator capabilities. As a first step, observation can be

made in comparing the effect of display format on operator performance under different task situations.

This section describes an experiment conducted in the current year in comparing different types of visual display for automated, remote manipulation. The apparatus used, the procedures followed, and the data recorded are reported in the subsequent paragraphs.

Three-Dimensional Graphic System. A graphic representation of stick figure images of the manipulator and static task objects was given in a 3-D graphic terminal. The system accepted a series of processed position sensor information and provides a series of 2-D pictures in their true spatial relationship. The 3-D image was created by a mirror mechanism positioned in front of a fast CRT. The mirrors vibrated back and forth to produce a volumetric change in the viewed image. In the computer, the image was made up of a stack of X-Y (horizontal-vertical) planes. Each plane represented the information at a given depth in the Z (depth) direction. Z planes were stored in digital memory and outputed sequentially, in synchrony with the movement of the mirror unit. The net effect was a real 3-D image, in that each plane was inherently transparent. By changing the viewer's position with respect to the display, one could observe a parallax containing image.

For the convenience of the experimental study, the image map of objects and environment was prestored as static background. Manipulator position feedback signals were the only raw data that were processed to represent the workspace interaction. Manual calibration of the relative position between selected manipulator path and selected environmental objects was required before each experimental session to ensure accuracy in spatial correspondence. During manipulator operation, the feedback position was given by the six angles of the joints. Those analog signals were digitized by the modular interface and monitored by the computer. The XYZ coordinates of the end point were then computed and displayed with the lengths of the manipulator links given. In this hypothetical mode of operation, images of work space were created bearing the detail of about 1" in the operating space. This limitation was attributable to the size of the display volume, the resolution of the imagery system, and the resolution of the feedback signals. The analog position signals were limited to a resolution of about 0.5". Although the display space could be considered to be a 256x256x256 raster, the real-time processing and memory constraints limited the image complexity to a maximum of about 4000 points. The actual image space was about 5 inches by 4 inches by 2 inches. Real world objects appeared quite small in the display space. The static background consisted of the basic elements of the task object, work table, representative obstacles and receptacles. The display is shown in Figure 4-8.

Environmental and State Feedback Display. As a baseline study for indirect viewing of the work environment, a single-view, black and white TV viewing system was used. Two 12" (diagonal) video monitors were placed above the 3-D display and in front of the control panel. Only the one with front view was used in the experiment. No camera control (pan, tilt, or form) was provided, which prevented any changes in the camera view. One dimension of this experimental study was visibility simulation which represents combined attributes of TV monitors, camera and environmental factors. A simplified, two-visibility level, similar to those used last year (Crooks et al, 1979) was established through contrast and brightness control of the monitor. The normal viewing condition corresponded to a contrast ratio of 59% and the degraded one corresponded to a contrast ratio of 17%.

A 9" video monitor was placed beside the control panel and below the 3-D display. It was used to display alphanumeric feedback concerning communication with the automatic command program. The modified machine feedback display was used in the experiment. As shown in Figure 4-9, the list of

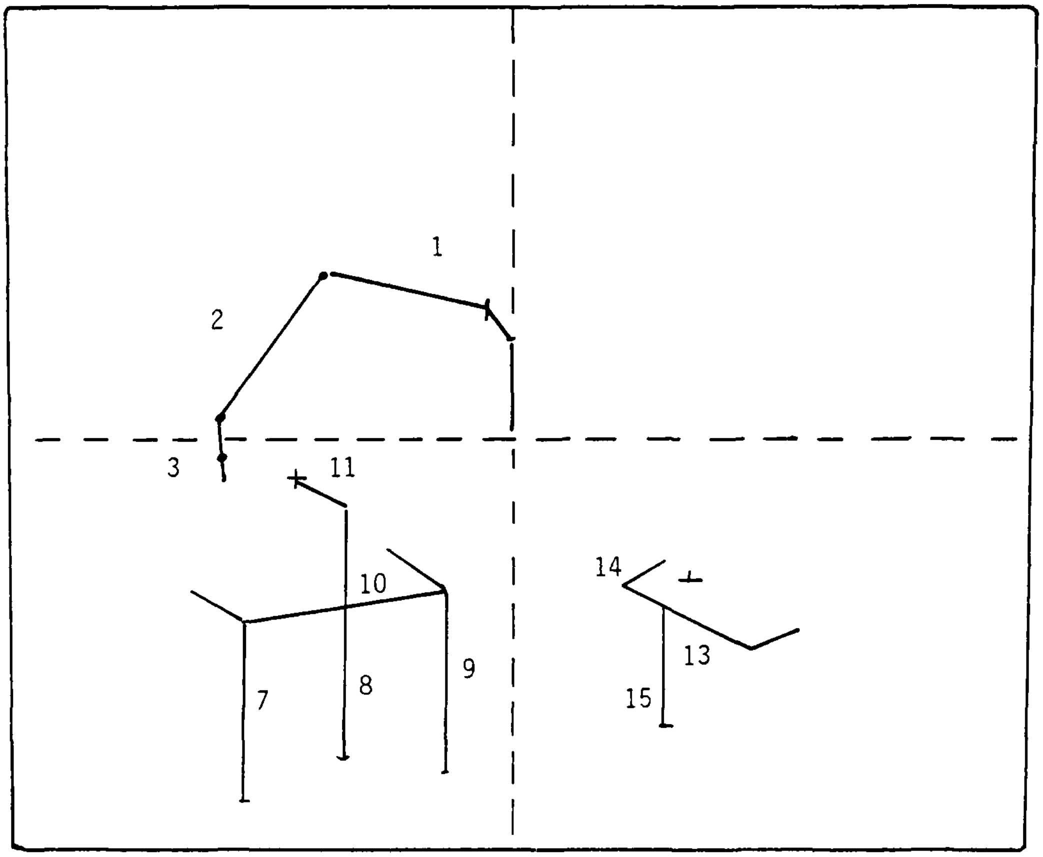
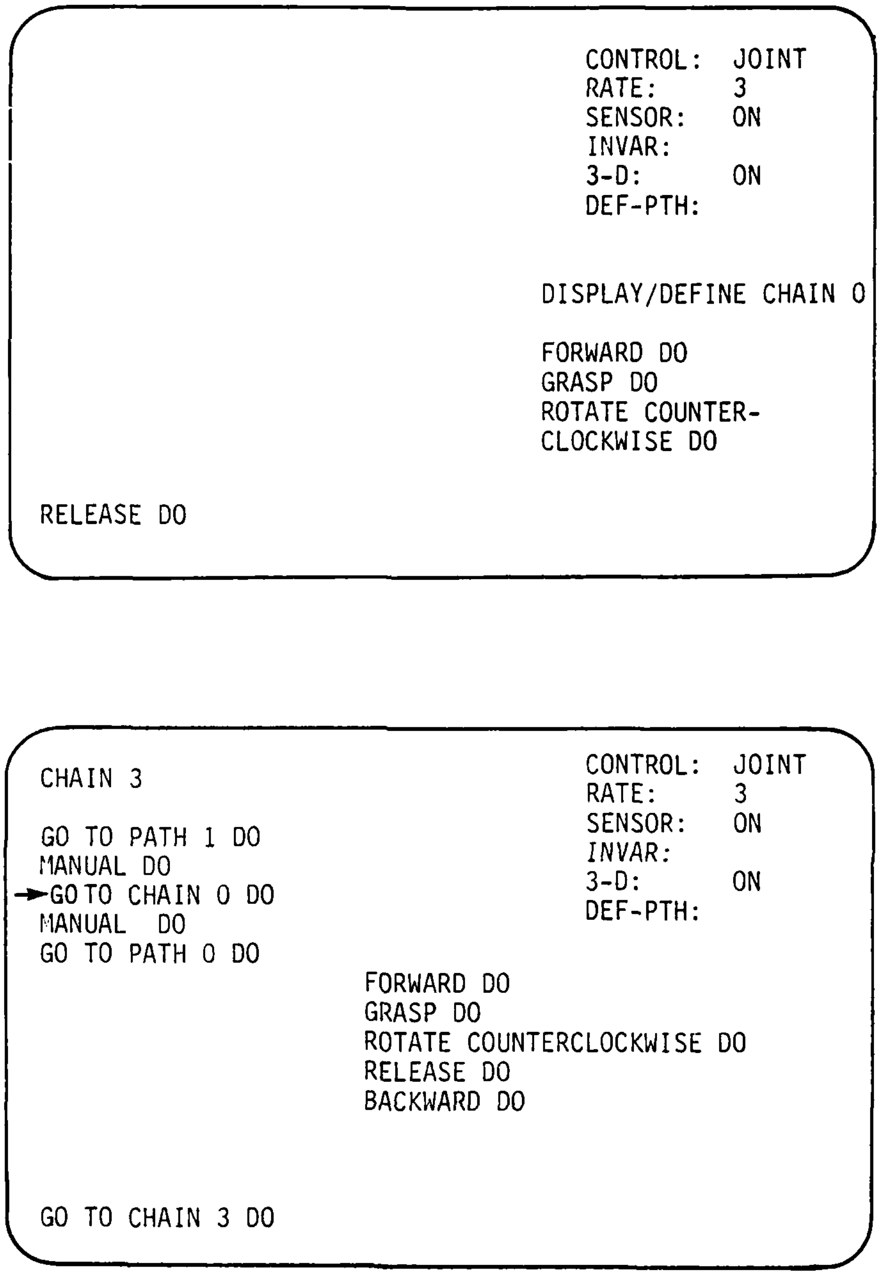


FIGURE 4-8.
STICK-FIGURE 3-D GRAPHIC DISPLAY OF MANIPULATOR AND WORKSPACE (VIEWED FROM RIGHT). LINES 1 THROUGH 3 REPRESENT MANIPULATOR CONFIGURATION; LINES 7 THROUGH 15 REPRESENT THE STATIC BACKGROUND



```
                                      CONTROL:  JOINT
                                      RATE:     3
                                      SENSOR:   ON
                                      INVAR:
                                      3-D:      ON
                                      DEF-PTH:


                                   DISPLAY/DEFINE CHAIN 0

                                   FORWARD DO
                                   GRASP DO
                                   ROTATE COUNTER-
                                   CLOCKWISE DO


RELEASE DO
```

```
                                      CONTROL:  JOINT
CHAIN 3                               RATE:     3
                                      SENSOR:   ON
GO TO PATH 1 DO                       INVAR:
MANUAL DO                             3-D:      ON
→GOTO CHAIN 0 DO                      DEF-PTH:
MANUAL  DO
GO TO PATH 0 DO
                      FORWARD DO
                      GRASP DO
                      ROTATE COUNTERCLOCKWISE DO
                      RELEASE DO
                      BACKWARD DO



GO TO CHAIN 3 DO
```

FIGURE 4-9.
MODIFIED MACHINE STATE FEEDBACK DISPLAY DURING CHAIN DEFINITION (TOP) AND DURING CHAIN EXECUTION (BOTTOM)

higher commands is posted in the left portion of the screen with slow update rate, while the list of lower level commands is displayed in the central portion of the screen with fast update rate. This two-level presentation appeared to provide easier perception of machine state than the previous design.

Experimental Variables. Two hypothetical tasks were used in the experiment. The first one was control-oriented and was used to evaluate the feedback format of environmental interaction. The second one was supervision-oriented and was used to evaluate the feedback format of machine state (in terms of task progress). This classification of feedback requirement was solely for the convenience of evaluation, and the physical presentation of the two types of feedback were not mutually exclusive. The levels of each dimension of interest were as follows:

(1) Environmental Feedback.

(a) Normal TV viewing - normal illumination and contrast.
(b) Degraded TV viewing - low illumination and contrast.
(c) 3-D graphical display - high illumination and contrast with low resolution.

(2) Machine State Feedback.

(a) Normal TV viewing.
(b) 3-D graphical display.
(c) Alphanumeric display of commands.

(3) Control Mode.

(a) Direct control - direct manual control of each manipulator joint.

(b) Resolved motion control - spatial X, Y, Z axis motion of the end-effector may be specified through joystick.

Performance measures consisted of task completion times and errors committed in performing assigned tasks. The time expended in the constituent elements of task operation, along with the errors including improper contact, dropping the object, and inappropriate command entry, was recorded. In all the experimental conditions, participants were instructed to perform the task as speedily as possible, allowing minor errors for each task performance.

Tasks and Procedures. Prior to the experimental sessions, four participants underwent several hours of orientation and practice. This unstructured practice included:

(1) Manual manipulation.
(2) Automatic command and control.
(3) Single-view TV in degraded condition.
(4) Brief viewing of 3-D display.

Two tasks were used in the experiment: the manual X-Z plane (horizontal-depth) positioning task, and the automatic monitoring of a simulated maintenance task. The manual positioning task performance evaluation followed the standard peg-in-hole task description. Namely, the task was divided into two phases: "reach" and "position." The "reach" phase was from start to within 3" of the receptacle. "Reach" time was assumed to be linearly related to distance and independent of final tolerance. "Position" time was assumed to be independent of distance and best modeled as a logarithmic function of tolerance. According to Fitts' index of difficulty in positioning task, $I_d = \ln\{2A/(B-C)\}$, where A is range distance and B-C is tolerance. Various amounts of tolerance were used to evaluate the level of task difficulty that can be supported by the limited resolution of the 3-D display. For 80% successful completion of the task, a tolerance of more than 1.5" was required with the use of current 3-D display.

The automatic monitoring task was a modified version of last year's simulated maintenance task (Crooks, et al, 1979). The task consisted of the following subtasks:

(1) Go to the top valve.
(2) Rotate the top valve clockwise.
(3) Go to the cap in the crossbar.
(4) Grasp the cap and remove it.
(5) Place the cap on the table.
(6) Pick up the cap from the table.
(7) Transport the cap to a position over the crossbar.
(8) Place the cap in the crossbar.
(9) Go to the top valve.
(10) Rotate the top valve counterclockwise.
(11) Return to the stow position.

To accomplish the automatic monitoring task, the participants were instructed to accurately define a two-level chain command with a sequence of manual checkpoints embedded. Then, to perform the task, the participants could simply activate the chain execution, monitor the subtask completion, and maintain continuous execution by depressing the "continue" key at the appropriate time. The chains utilize four points and two paths. The location of the points was as follows:

(1) Over the table, at a height corresponding to that of the cap.
(2) Between the table and top valve, of sufficient height over the latter to prevent collision with it when traveling to point 3.

(3) Over the top valve at a distance corresponding to that traversed by execution of a "backward do" command from the valve-grasp position, so that future execution of a "forward do" command will position the gripper to grasp the valve through execution of the "grasp do" command.

(4) Over the cap in the crossbar, at a distance corresponding to that traversed by execution of a "backward do" command from the cap grasp position, for reasons same as above.

The location of the paths was as follows:

(1) Traversing points 1-2-3, in order to provide arm movement from the table to the top valve.
(2) Traversing points 3-2-4, in order to provide arm movement from the top valve to the cap.

Once the task commenced, the participants were asked to perform the sequence as quickly as possible. The major efforts required of the participants were to recognize, and respond to the embedded manual checkpoints utilizing three separate types of feedback: alphanumeric messages, 3-D pictorial, and normal TV viewing.

Results. The average task compleiton time for the gross positioning task as functions of feedback type, control mode and trial number, are graphed in Figure 4-10. The results of a within-subject analysis of variance indicated that only the effect of control mode was significant ($F = 19.4$, $P < 0.01$). No significant main effect of display type was obtained and the two-way interaction (control x display) was not statistically significant. Comparing the display type, we found that participants were able to use the 3-D stick-figure display in performing the task within a reasonable error rate (<20%) and that they were faster using the 3-D display than using the degraded single-view TV display,

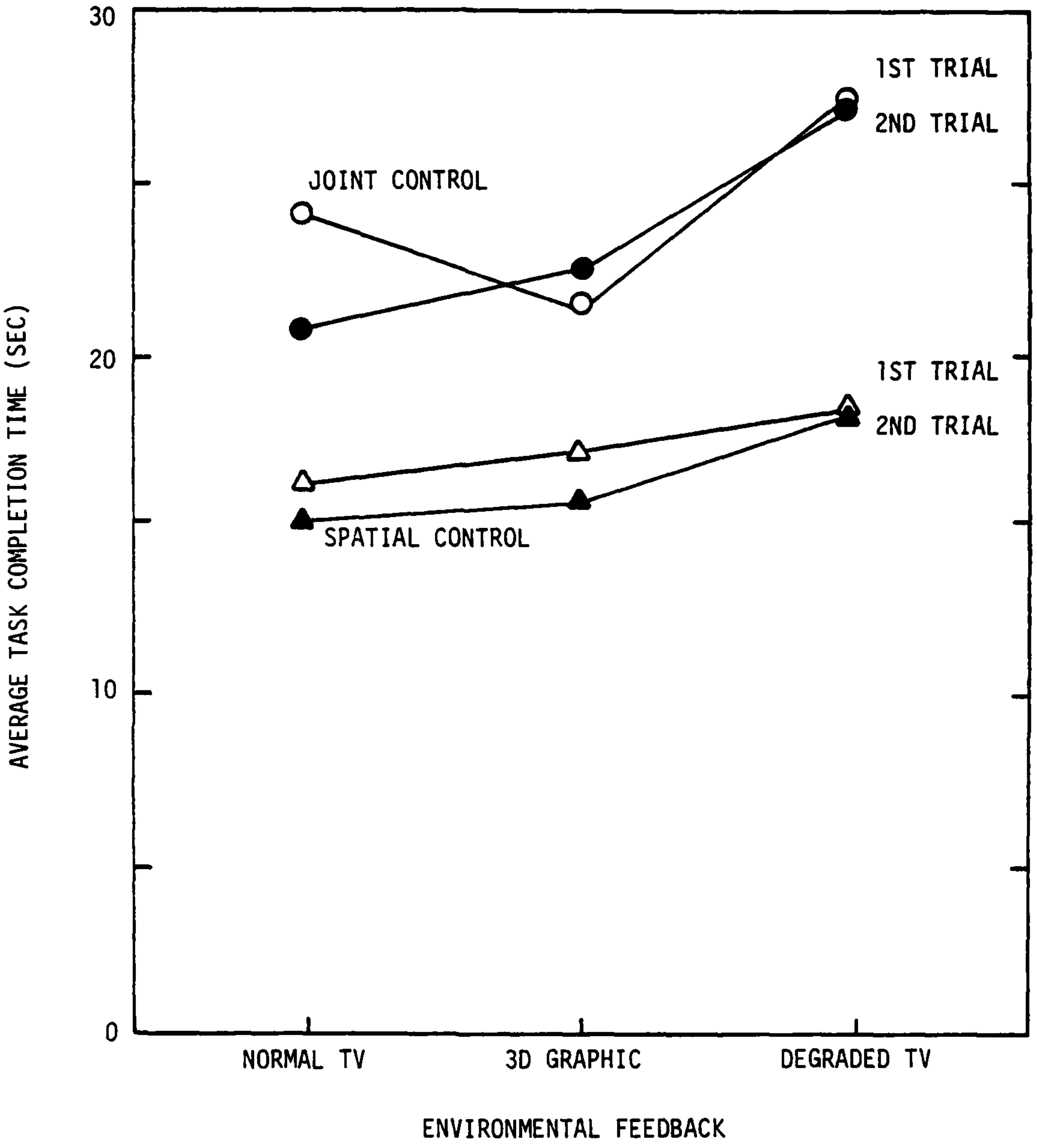


FIGURE 4-10.
AVERAGE TASK COMPLETION TIME AS A FUNCTION OF CONTROL MODE AND FEEDBACK TYPE FOR A GROSS POSITIONING TASK.

although they incurred a higher error rate in the former case than in the latter case. This observation suggested that while depth perception was important in this X-Z plane positioning task and the limited, but direct, depth perception provided by the 3-D stick-figure display was helpful, the major limitations in such a display remains to be its low resolution. Some other handicaps noted included inexact correspondance between image workspace and physical workspace and difficulty in recognizing details in physically small features. All these have limited the use of such a display to tasks with simple (one degree-of-freedom) and gross (high allowance and low interaction) motion, such as object transfer and gross alignment tasks demonstrated in the experiment. In these cases, the stick-figure display has the potential to achieve a better performance than can be achieved by a degraded TV-display.

The average task execution times in monitoring the simplified maintenance task as a function of display type and trial are graphed in Figure 4-11. The effect of display x trial interaction was significant (F = 11.57, P < 0.05). As shown in the figure, time performance of the TV display was superior to that of the graphic display in the first trial. The reverse was true, however, in the second trial. Since similar results were also obtained with the alphanumeric display with which the participants were well-aquainted, the interaction effects cannot be attributed solely to direct learning (in the use of rather abstract stick-figure display). Instead, user confidence (in the automated procedure) and the presentation of cue to motion completion were thought to be the major factors. The three types of display, TV, graphic, and alphanumeric provided the full-cue, reduced-cue for motion, and explicit message situations. Observations in the experiment suggested that the concept of a subjective confidence level may be used to describe the participant's monitoring processes as a continued observation of relevance cue until the confidence level was reached and action initiated. Overall, these

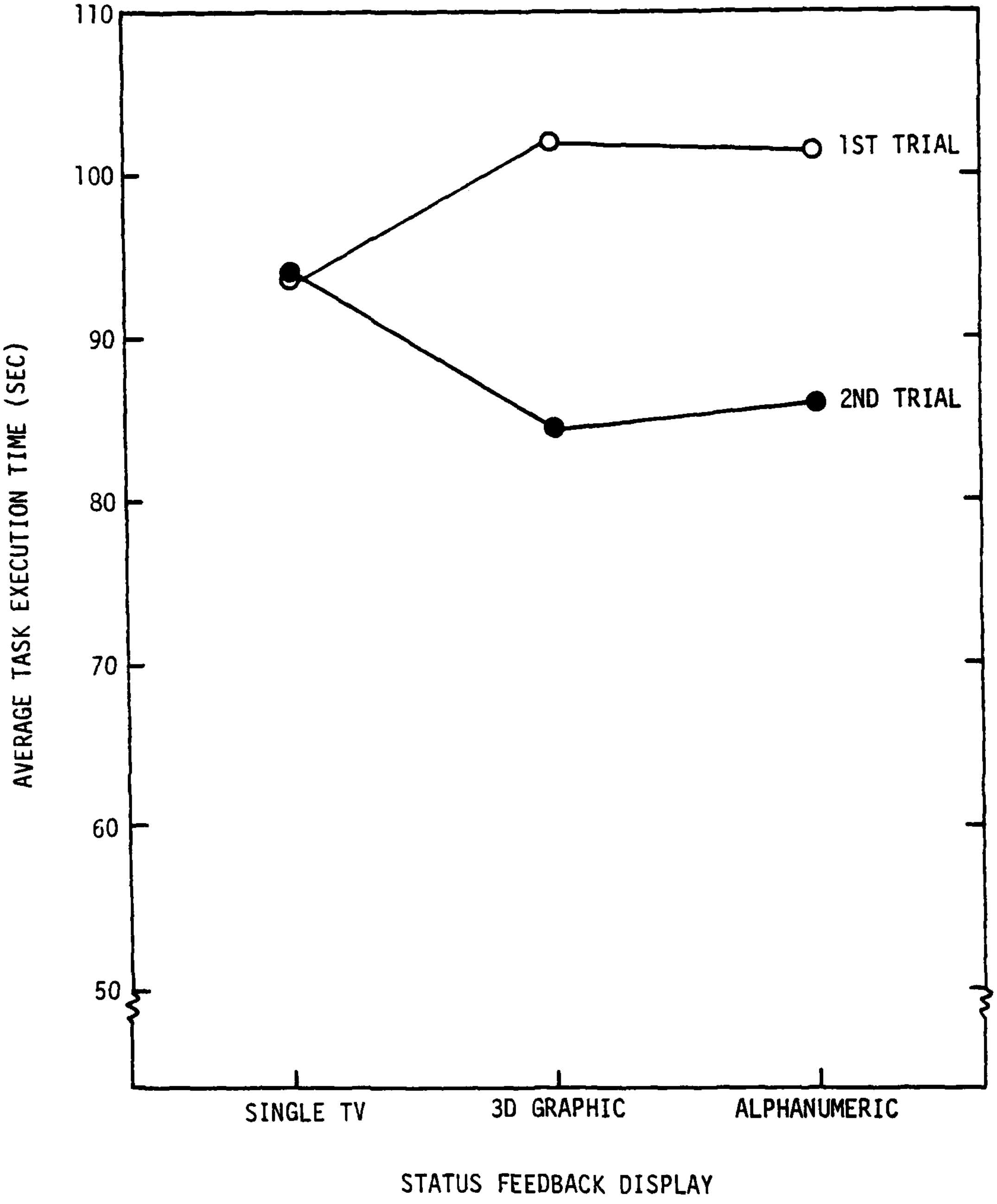


FIGURE 4-11.
AVERAGE TASK EXECUTION TIME FOR THE MONITORING OF A SIMPLIFIED MAINTENANCE TASK USING THREE SEPARATE DISPLAYS OF STATUS FEEDBACK

findings suggest that in monitoring the preplanned process, performance time can be reduced by using a reduced-cue presentation which, in effect, has lower noise and interference, provided that both cue sufficiency and user confidence have been obtained.

Comparisons were made of the effects of display format on operator performance with two very different types of tasks: The manual positioning task and the procedure monitoring task. The first one resembled tracking control and required close interaction between control and feedback. It required multiple cues in depth and motion perception. Considerable impairments in performance speed and accuracy could result from low resolution, insufficient cues or degraded environment. The second task resembled the detection/recognition task, except that it was simplified and could be accomplished by using motion cue alone. In such a case, performance was less affected by resolution and display factors.

4.3.3 Summary of Machine-to-Man Communication. Studies were made evaluating the effects of the feedback format and level (detail and visibility). In summary, comparisons were made of time performance ratios for various feedback conditions. The results, shown in Table 4-3, were grouped into two categories: the control/definition phase, where the participant's were mostly in the control loop, and the monitoring/execution phase, where the participants were mostly in the supervision loop, with occasional control activities. In general, performance was more sensitive to feedback conditions in the control phase than in the monitoring phase, and more sensitive to feedback level than to feedback format within the ranges of variation made during the studies. The values obtained were essentially averages across control and command modes and thus they did not reflect the variations along these dimensions, although they were expected to be relatively small. Therefore, care must be taken in generalizing this set of results to other systems with different command and control modes

TABLE 4-3.

PERFORMANCE TIME RATIOS FOR THE EXPERIMENTAL TASKS UNDER CONTROL/DEFINITION AND MONITORING/EXECUTION PHASES

FEEDBACK MODE*	TEST PHASE	TIME RATIOS**	
		CONTROL/DEFINITION TASKS	MONITORING/EXECUTION TASKS
2-VIEW/DEG 2-VIEW	3A	0.85+	0.94
(2-VIEW + SF)/2-VIEW	3B	0.91	0.93
1-VIEW/DEG 1-VIEW	4A	0.85+	----
3D GRAPHIC/1-VIEW	4A&B	0.86+	0.97
SF/1-VIEW	4B	----	1.00

*2-VIEW - TWO-VIEW TV UNDER NORMAL VISIBILITY

DEG 2-VIEW - 2-VIEW UNDER DEGRADED VISIBILITY

SF - DETAIL STATE FEEDBACK IN ALPHANUMERIC

1 - VIEW - ONE-VIEW TV UNDER NORMAL VISIBILITY

DEG 1-VIEW - ONE-VIEW TV WITH DEGRADED VISIBILITY

3D GRAPHIC - THREE-DIMENSIONAL STICK-FIGURE GRAPHIC

**BASED ON THE AVERAGE PART-TASK PERFORMANCE TIME OF THE SIMULATED MAINTENANCE TASK RATIO OF 1.0 = EQUIVALENT PERFORMANCE FOR THE TESTED CONDITIONS

+$P < 0.05$

to that of TOSC systems. Also, due to the use of tightly-controlled hypothetical tasks in the display evaluation, any attempt to extrapolating the quantative results should be guarded against possible uncontrolled task characteristics.

Some other qualitative observations in this comparison, however, are quite general. Overall, the greatest advantages of computer-aiding seem to occur where there is degraded input/feedback due to low fidelity of work-space representation or occluded environment. These advantages can be realized by improving man-to-machine communication and/or machine-to-man communication, as described in the previous sections. Also, a great number of research in manual control demonstrates that the more able the subject is in predicting future course of event/trajectory, the more efficient his control. Data gathered in this study, can well explain and generalize this predictive effect on performance. It appears that the command language helped the subjects to specify the general structure and organization of his knowledge about complex manipulation tasks in which plans for the future interact with state of the process and its control requirements. Better performances were then achieved not only due to control automation, but also due to the efficient and confident planning processes in the use of the command language.

5. DISCUSSIONS OF RESULTS AND RECOMMENDATIONS

5.1 Overview

The overall experimental program presented in the previous chapter was a series of studies to evaluate techniques and methods for improved man-machine communication in computer-aided manipulation. Among the factors that are significant in affecting system performance are the resolved motion control, the automatic motion control, command language structure, machine state feedback, task characteristics, environmental feedback and display. A summary and consolidation of the experimental data and analytic analyses provide the following conclusions and implications to be described in this chapter:

(1) The best mode of communication with the use of TOSC, that is, where and how the automation could be effectively applied to a remote manipulation system.

(2) The amount of automation in relation to the level of performance to be expected.

(3) The information requirements for the design of TOSC operational environment.

For the first two items, Section 5.2 describes the issues related to performance evaluation of man-machine communication. A simple time performance prediction model was derived; factors affecting a number of performance dimensions were identified; and the effectiveness of various aiding techniques was estimated. For the third item, Section 5.3 discusses the possible extension of TOSC design and future research needs, and Section 5.4 presents an example application.

5.2 Issues in the Design and Evaluation of Man-Machine Communication

The design and evaluation of man-machine communication should take into account the overall performance of the combined man-machine system.* Several issues need to be considered in such a process. First, there are a number of dimensions (or criteria) to the performance of man-machine communication, which are discussed in Section 5.2.1. Second, there are many factors affecting each performance dimension, including those related to man, those related to machine, and those related to both man and machine. Factors of major importance are considered in Section 5.2.2. Third, as the number of factors and the interaction among them increases with the complexity of man-machine systems, there is a need to have a systematic approach for the evaluation of the overall performance. A standard approach for performance analysis used in this study is further elaborated and discussed in Section 5.2.3. As an example of this approach, a simple descriptive model for performance improvement, is given in Section 5.2.4. Although the model, in an extremely simplified form, provides only limited performance prediction, it may serve as a benchmark indicator for early decisions in complex system design. Finally, some of the useful benchmarks for the design of a man-machine communication system are given.

5.2.1 Performance Dimensions of Man-Machine Communication. Like value, performance is a subjective concept. Depending on the purpose of performance evaluation, the set of relevant performance descriptors or indices varies. Among other purposes, the usual intent of man-machine communication evaluation is one of either two purposes:

*Within this chapter, the term 'performance' will be used explicitly to refer to how well the human does; whereas the term 'behavior' will refer to what the human actually has done. In a gross travel manipulation, for example, the human's performance is the speed and error resulted; whereas the human's behavior can be described as the time history of his command and control inputs.

(1) Performance monitoring and selection, in which the evaluator plans to include performance as a major criterion in the decision to employ a particular communication mode.

(2) Performance improvement, in which the evaluator plans to obtain actual data of an existing communication configuration to forecast the impact of changes or for the design of a new system.

The evaluation studies in this program belong to the second type, which includes a possibly broader set of performance indices. As stated earlier in Chapter 4, evaluation of overall performance for a complex man-machine system includes three main aspects: (1) quality in operation (i.e., capability and quality of a particular function realized), (2) demand/consumption level (i.e., the corresponding resource consumption for achieving a given operation level, such as power, attention and effort, etc.), and (3) productivity (utilitzation and efficiency factors such as throughput, allocated utilization, and probability of success). Although only the first aspect of performance was directly addressed in this study, indirect suggestions can be drawn related to the other two aspects. To bring these aspects into perspective, a list of important dimensions to the performance aspects is summarized in Table 5-1. While a rather restricted definition of performance is given in the Table in order to suggest easily quantifiable indices, many performance dimensions remain difficult to measure. Besides, a number of important performance dimensions have long eluded many researchers, such as the *ease of use,* the *effort required* to communicate, the *power* of a command set, and the level of *user acceptance.* Many of these dimensions reflect the psychological characteristics of human operators and their interaction with combined task and system features. While these characteristics and features will be discussed in the next two sections, the following paragraphs present a summary of the experimental findings along some of the performance dimensions listed in Table 5-1.

TABLE 5-1

MAIN CLASSES OF QUANTITATIVE INDICES OF
MAN-MACHINE COMMUNICATION SYSTEM PERFORMANCE

INDEX		GENERAL DEFINITION
Quality in Operation	Time duration (speed)	The length of time it takes the operator to accomplish a given set of tasks using a particular communication mode.
	Error (accuracy)	Number and type of errors the operator makes.
	Precision	A group of spatial and temporal measures.
	Functionality	The range and quality of tasks the operator can do in practice with a particular communication mode.
Demand/ Consumption	Processing time	Machine processing time (speed) required to maintain real-time supervisory control loop.
	Processing space	Memory size required for specified level of functionality.
	Pre-programming time	The time it takes the operator to pre-program the execution of a task.
	Learning	The time it takes a novice to learn the use of communication mode.
	Concentration	The level of intensity the operator has to attend to a task.
	Recall	The ease for the operator to recall how to use the communication mode on a task that has not been done for some time.
	Fatigue	The degree the operator gets tired when he uses the communication mode for extended periods.
Productivity	Production rate	The volume of unitary motions/activities produced in the unit time.
	Throughput rate	The volume of information transmitted in the unit time.
	Subsystem utilization	The fraction of time a specific subsystem is used.

Task-Time Duration. Task performance time data for various computer-aided functions have been discussed in Chapter 4. In summary, significantly shorter performance times were obtained with the addition of resolved motion control, variable symbolic commands, and high-level symbolic commands, respectively. The addition of feedback information as well as other types of control aiding has less impact on time performance compared to the command modes listed above. Nevertheless, timely and well-formatted feedback functions such as the alphanumeric state feedback and the 3-D graphic showed potential time improvement.

Performance Errors. Since performance time - accuracy tradeoffs were often required of the operator in manipulation tasks, performance errors need to be measured explicitly in the experiments. Performance data includes those of manipulative errors (e.g., contact obstacles, drop objects, etc.) and those of executional errors (e.g., improper command input, improper chain activation, etc.). Error data reported earlier (Berson, et al, 1977, Crooks, et al, 1978, Crooks, et al, 1979) has shown that significantly fewer errors were observed with the use of resolved motion control, variable symbolic command, high-level symbolic command, and alphanumeric state feedback. All the other tested aiding functions, including fixed symbolic command, force-sensor message and 3-D graphic, have shown potential for accuracy improvement.

Precision. This refers to the abilities of the teleoperator to meet a spectrum of spatial allowances or constraints. Observations in test sessions have shown that performance precision remained relatively at the same level with the RMC, and decreased by approximately 1/5" when any of the symbolic commands were used. Potential improvement of precision may be achieved by force sensor and 3-D graphics if better resolution of both devices can be achieved.

Functionality. This includes the capabilities related to the task attributes listed in Section 5.2.2, such as complexity, uncertainty and simultaneity. Some of the task domains that the aiding functions can be gainfully applied are:

(1) Procedural Comolexity - Variable command and high-level command.

(2) Task Uncertainty - Variable command and high-level command; also potentially, force sensor and 3-D graphic feedback.

(3) Simultaneity - Resolved motion control, variable and high-level commands.

Processing Time Requirements. The real-time processing requirements in support of TOSC features are discussed in Appendix B. Estimates of time required for each type of aiding function are summarized below (based on Interdata 70 minicomputer instruction execution time, in mili-seconds).

RMC - 30 ms
Symbolic Command - 10 ms
Machine State Feedback - 2 ms
Force Sensor/Display - 15 ms
3-D Graphics - 20 ms

Processing Memory/Core Requirement. The memory space requirement for the implemented aiding routines are listed below (in byte).

Resolved Motion Control - 5 K
Symbolic Command - 15 K
Machine State Feedback - 5 K
Force Sensor Routines - 5 K
3-D graphic - 20 K

Preprogramming Time. Preprogram is required for variable symbolic command and high-level symbolic command, and may incur significant overhead in total task completion time. In various experimental tasks, it ran from 0% to 64% of the total task time.

Production Rate. The amount of useful work produced per unit time is a function of the amount of command information transmitted (throughput rate), the amount of unitary motions produced by each command, and the effector motion rate. Given throughput and motion rates, it appears that high-level symbolic command is capable of generating higher production than low-level commands. The related notion of command language power will be discussed in a later section.

5.2.2 Critical Factors Affecting Task Performance. In general, from the operator's point of view, the task factors include both intrinsic task elements (such as degree of constraints--gross motion vs. fine motion) and environmental conditions (such as visibility and turbulence); the system factors include both communication system characteristics (such as symbolic vs. analogic commands and alphanumeric vs. graphic displays) and activities engaged (such as planning level). Within the scope of this research, only those tasks that are suitable to be performed by the manipulator were considered. Therefore, the set of identified *task* attributes affecting communication performance can be narrowed and described as follows (refer to Chapter 2 for details);

(1) Precision - position and force allowance, task status prediction.
(2) Speed - time-stress control, speed criticality in programming.
(3) Accuracy - procedural accuracy, execution accuracy.
(4) Complexity - number of constraints, variety of objects and tools, procedural complexity.

(5) Uncertainty - environmental disturbance, executional (structural or relational) uncertainty, goal specificity.

(6) Simultaneity - control degrees-of-freedom, number of processes supervised.

(7) Uniformity - dispersion of levels of difficulties resulting from above requirements.

On the other hand, the set of critical *system* attributes affecting communication performance was similarly identified:

(1) Control mode.

(a) Geometric/dynamic transformation.
(b) Hand motion.
(c) Accommodation and compliance.

(2) Programming mode.

(a) Trajectory repetition.
(b) Trajectory interpolation.
(c) Trajectory transformation.

(3) Command mix.

(a) Symbolic vs. analogic.
(b) Variable vs. fixed.

(4) Command structure.

(a) Multiple-level vs. single level.
(b) Linear sequential vs. hierarchical network.

(5) Display format.

    (a) Perceptual modality.
    (b) Display codes.

(6) Display scope and level.

    (a) Environmental state.
    (b) Machine and programming state.

(7) Display integration.

    (a) Frame content and update rate.
    (b) Predictive, filtering, and enhancement.

In addition, there is the third category of factors that is important in affecting communication performance, i.e., operator skills and training factors. The operator abilities requirements are such important factors in successful man-machine system design that suggestions have been made that the first rule in the design is to have operator abilities in mind (e.g., Schneiderman, 1980). In the context of this study, the operation of the TOSC system included the following:

(1) Spatial perception.

    (a) Closure flexibility.
    (b) Spatial relation.

(2) Manual dexterity.
(3) Control accuracy.

(a) Reaction time.
(b) End-effector steadiness.
(c) Speed of arm movement.
(d) Speed of wrist-gripper.
(e) Multi-degree-of-freedom coordination.
(f) Positioning.

(4) Scan-search.
(5) Assimilation/interpretation.
(6) Procedure visualization (abstraction).

(a) Identify.
(b) Decompose.
(c) Reformulate.

(7) Subtask sequencing.
(8) Command recall.
(9) Command organization.
(10) Subplan selection.
(11) Outcome evaluation and prediction.

Further analysis, such as the factor-analytic method, is needed in order to isolate and identify ability factors that constitute a necessary set of requirements for a wide range of communication modes. Ultimately, this classification scheme must make the "match" between specific abilities categories and performance requirements. To simplify these procedures, a gross categorization of operator characteristics concerning the use of the TOSC system has been derived and is presented as follows:

(1) Operator with programming skill?
(2) Operator with manual control skill?
(3) Operator familiar with remote environment?

(4) Operator familiar with general manipulation?

(5) Operator familiar with the use of automatic functions?

5.2.3 System Approaches of Man-Machine Communication Evaluation. As discussed in the last section, numerous aspects of performance, numerous classes of tasks, and numerous types of operators contribute to the overall man-machine communication evaluation. It therefore appears that there needs to be a systematic method for performance evaluation, for comparison purposes or for improvement purposes. Even though in most practical cases only very rough estimates can be obtained by any such methods, it is always wiser to consider the "benchmark" measures than to blindly search for an alternative design.

In comparison evaluation, a useful set of benchmark indices may prove sufficient in making initial design and selection decisions. One example is the index of information processing power associated with the instruction set of the central-processing computer, which was derived by Knight (1963) from opinions of experts. The index has been used in several extensive studies of technological progress in computer systems (Cenveny and Knight, 1973).

Another common approach in evaluation studies is the performance improvement approach. This evaluation approach was used in the experimental studies (see Chapter 4) as a means to lead toward the "optimal mode" of man-machine communication. As the studies evolved, it was evident that the task and system parameters that should be considered as candidates for modification were too many to be enumerated and their bounds were difficult to establish. Also, if performance was not too far from the optimum, the marginal improvement benefits might prove small, compared to the modification. Therefore, the improved performance, which is suboptimal in the absolute sense, may be truely optimal in terms of cost-effectiveness considerations.

In the improvement evaluation, a predictive model of performance dimensions under evaluation may prove invaluable. This improvement procedure is, by necessity, iterative and consists of two major phases: one in which hypotheses about the sources of possible improvements are tested, and one in which tailoring or addition of some features in communication is made. An essential role, is, therefore, played by performance analysis in improvement studies to project the hypotheses testing and to predict .e consequence of proposed modification.

As such, performance prediction models have long been sought to aid in man-machine communication system design and to better understand how system designs affect performance. Present modeling methods include both behavior models and performance models (Rouse, 1980). Since a behavior modeling approach would undoubtedly lead into many unnecessary complications and the performance predictions are all that are necessary for many design applications, the formal approach was adopted for further analysis.

It was concluded that no uniform approach to modeling the entire system in a simple way appears possible at this time. Thus, of necessity, the model to be presented is specific to a few dimensions of man-machine communication: the task execution time for skilled operators in performing error-free manipulation with various task-system characteristics. The model presented in the next section represents an initial attempt to quantify the performance evaluation for shared man-computer control systems in general, and teleoperator control systems in particular. Although there have been a number of research directions addressing issues of performance evaluation in teleoperator control systems, few have general, applicable, systematic evaluation in matching combined task-system characteristics on the one hand, and operator aiding and performance on the other. The rationale for such an approach has been closely described by Johannsen and Rouse (1979):

> "There are limits to context-free analytical modeling. First, there is the very important idea that human behavior mainly reflects the task environment. Thus, searching for a specific analytical model of general human behavior may only be fruitful to the extent that all task environments are common. Perhaps then, one should first search for commonality among environments rather than intrinsic human characteristics. In other words, a good model of the demands of the environment may allow a reasonable initial prediction of human performance."

5.2.4 A Simple Model for Predicting Improved Time Performance. The prediction problem to be addressed is as follows: *Given*

(1) A task (possibly including several levels of subtasks).

(2) The communication mode.

(3) Environmental conditions.

(4) The unit task times using direct manual control.

(5) Reduction parameters evaluated independent along the dimensions within items (1), (2), (3).

Predict the error-free task performance time that a skilled operator will take.

Performance Model Formulation. The basic assumptions of the model are:

(1) A complex task can be broken down, in series, to the lowest level of situation assessment-action execution pairs, which are quasi-independent behavioral elements.

(2) The situation assessment-action execution pairs may take the dichotomies of "reach & movement," "cognition & activation," or "acquire & execute."

(3) The relative execution time ratios for a given behavioral element are functions of communication mode, task intrinsic characteristics, and environmental factors only.

(4) Total task completion time is the sum of task definition time and task execution time which, in turn, is the sum of execution times of constituent behavioral elements.

Based on these assumptions, an additive model of task performance time is given in Figure 5-1 which represents a task with two levels of decomposition. Task completion time, T_C, and task execution time, T_E, are given for four types of communication models: direct control, augmented control, traded control, and high order command. These predictions are based on the empirical performance times for the complete set of behavioral elements, T_{ij}, and the mental decision/transition times associated with each behavioral element (ij), ε_{ij}. The time reductions in the use of augmented control are represented by γ factors (varying from 0.8 to 1.0 in the study) which are functions of intrinsic task characteristics and environmental factors (equation 2); whereas the time reductions in the use of both augmented and traded control are represented by an additional reduction factor α (which varies from 0.2 to 1.0) and is a function of combined task-system factors and environmental factors (equation 3). The model reflects the trend of reducing the number of operator decision/transition epochs (t's in Figure 5-1) and the trend of increasing definition overhead (d's in Figure 5-1) with the addition of traded control and high level commands.

Performance Model Validation. The model presented above is extremely simple, yet it seems effective in summarizing part of the experimental results. To determine how well the simple additive performance time model actually predicts, the experimental study related to command organization (Section 4.2.3) was re-examined and results were compared of model prediction and empirical data. The selection of the particular experiment for model validation was based on its broad spectrum of independent variables tested (across intrinsic task, environmental and system factors) and its complete measurement of dependent variables (to the degree that key-strokes

Estimated Overall task performance time* for:

1. DIRECT CONTROL (ANALOG PLUS SWITCH)

$$T_C = T_E = \sum_i \sum_j (t_{ij} + \varepsilon_{ij}) \qquad (1)$$

2. AUGMENTED/SERIAL CONTROL (WRIST-CENTERED RESOLVED-MOTION)

$$T_C = T_E = \sum_i \sum_j (t'_{ij} + \varepsilon_{ij}),\ t'_{ij} = t_{ij} \cdot \gamma_{ij}, \qquad (2)$$

$$\gamma_{ij} = \gamma_{ij}\ (a_1, a_2, \ldots)\ \varepsilon\ (0.8, 1.0)$$

3. TRADED/PARALLEL CONTROL (SYMBOLIC POINT/PATH/CHAIN)

$$T_E = \sum_i \{(\sum_j t''_{ij}) + \varepsilon_i\} \qquad t''_{ij} = t'_{ij} \cdot \alpha_{ij} \qquad (3)$$

$$T_C = T_E + \sum_i d_i \qquad \alpha_{ij} = \alpha_{ij}\ (b_1, b_2, \ldots)\ \varepsilon\ (0.2, 1.0)$$

4. HIGH-LEVEL COMMAND (CHAIN-IN-CHAIN)

$$T_E = (\sum_i \sum_j t''_{ij}) + \varepsilon_0 \qquad (4)$$

$$T_C = T_E + d_0 + \sum_i d_i$$

* T_C and T_E are the overall task completion and execution time estimates.

T_{ij}, T'_{ij} and t''_{ij} are the execution times for task element $_{ij}$ with direct, augmented, and traded control, respectively

ε_{ij}, ε_i, ε_0 are the decision and mental transition times.

γ_{ij} and α_{ij} are the time reduction factors for augmented and traded control, respectively.

d_i and d_0 are the command definition times.

a's and b's are the critical and task factors.

FIGURE 5-1.
AN ADDITIONAL MODEL OF TASK PERFORMANCE TIME

and microstructure movements could be reconstructed with the use of an automatic data collecting procedure).

Six trained subjects were used in the experimental study. The simulated maintenance task performed by the subjects was essentially the same as the one described in the Environmental Feedback experiment (Section 4.3.2). The task consisted of eleven subtasks, including:

(1) Go to the top valve.
(2) Rotate the top valve clockwise.
(3) Go to the cap in the crossbar.
(4) Grasp the cap and remove it.
(5) Place the cap on the table.
(6) Pick up the cap from the table.
(7) Transport the cap to a position over the crossbar.
(8) Place the cap in the crossbar.
(9) Go to the top valve.
(10) Rotate the top valve couterclockwise.
(11) Return to the stow position.

These subtasks were further decomposed into finer elements, including the following: approach, align (with valve handle), position (over cap), turn (Valve), stow, grasp/release, and insert. It was observed, during the experiment, that different degrees of time reduction were achieved for different task elements when the overall task weas repeated using automatic commands. This was due to differing system characteristics, task intrinsic and environmental characteristics, and the interactions among them. Further analysis of task-element performance suggested that the effects of the performance can be described in a classification tree of task dimensions, such as the one shown in Figure 5-2. Task attributes were organized in levels according to the order of importance. At the top level

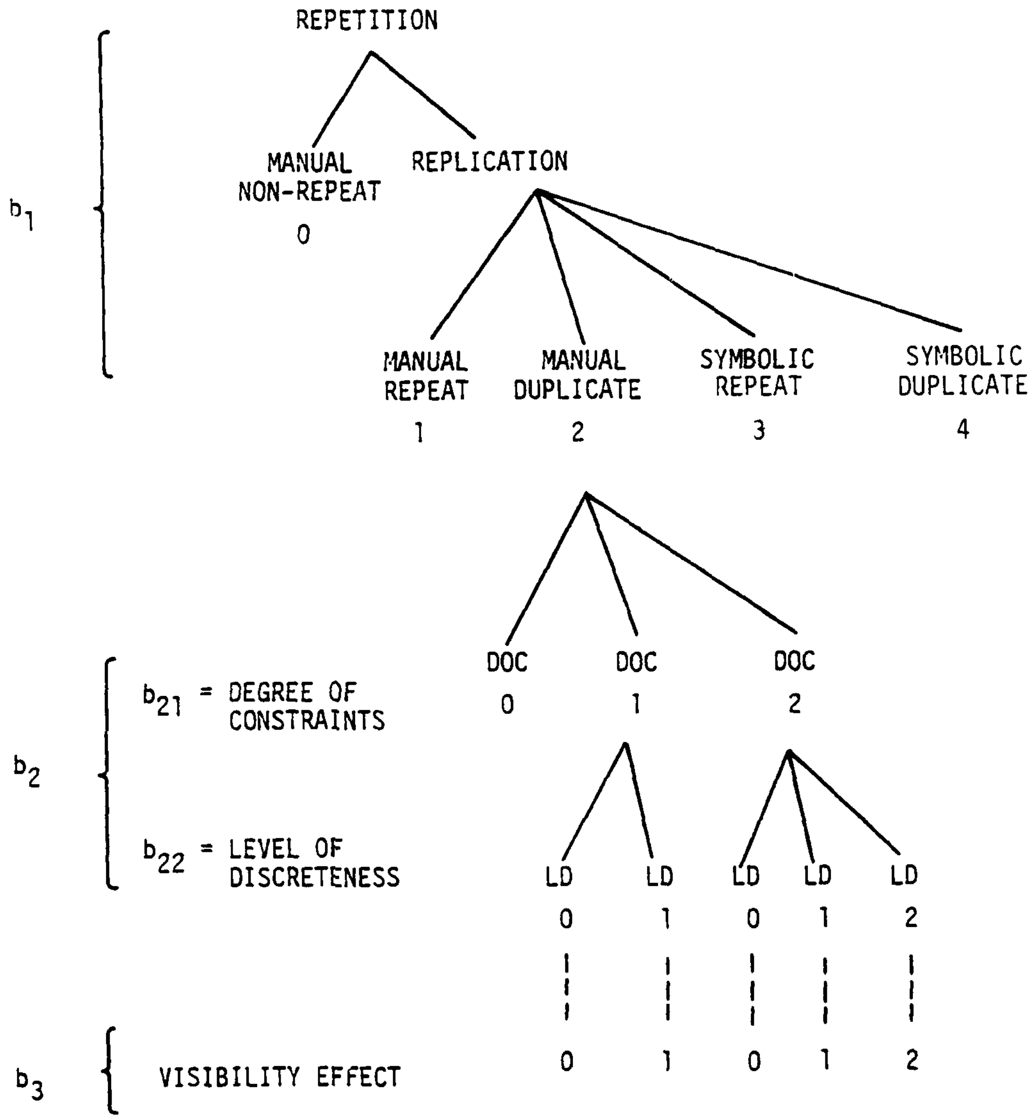


FIGURE 5-2.
CHARACTERISTICS OF MOTION ELEMENTS PERFORMED UNDER SHARED MAN-COMPUTER CONTROL

(b_1), procedural aspects of motion was manually-controlled and not repeated in any manner (i.e., $b_1 = 0$). Performance improvement was expected when motion could be replicated as (1) repeated (with relative reference in workspace) or duplicated (with absolute reference); and as (2) manual (with no symbolic abstraction) or symbolic (with symbolic abstraction). Hence, there were four combinations of motion replication: manual repeat ($b_1 =1$), manual duplicate ($b_1 = 2$), symbolic repeat ($b_1 = 3$) and symbolic duplicate ($b_1 = 4$). Here, the values assigned to b_1 represent the relative levels of performance improvement expected for each type of replication, which mainly reflects the system (command) contribution.

At the second level, constraint (taxon) aspects of motion elements were considered. It was considered that major improvements can be achieved when difficult multi-degree-of-constraint motion could be effectively discretized into single or low degree-of-constraint motions and duplicated with the use of computer assisted functions. The combined task-system effect of constraint discreteness (b_2) can be represented in a simple additive function of (1) degree of constraints, b_{21}, and (2) level of discreteness, b_{22}.

At the third level, environmental aspects of motion elements were considered. It was noted in the experiment that performance improvement was higher in degraded visibility than in normal visibility, especially with a task of high level of discreteness where requirements of visual observation were greatly reduced with the use of automatic commands. It was then assumed that visibility effect was differentiated by the level of discreteness only. Other factors may enter the analysis with further expansion of the tree in more detailed levels. To determine the ratios of time reduction, the tree of motion analysis can be folded back, with the relevant attributes identified. Table 5-2 presents the calculated values of time reduction factors for a set of selected motion elements under normal visibility (α's) and degraded visibility ($\bar{\alpha}$'s). These values were used next in the calculation of execution time.

TABLE 5-2

DETERMINATION OF TIME REDUCTION FACTORS FOR A SET OF SELECTED MOTION ELEMENTS

MOTION ELEMENTS	LEVEL OF ATTRIBUTES			TIME REDUCTION FACTORS*	
	b_1	b_2	b_3	α_{ij}	α'_{ij}
APPROACH	4	0	0	0.400	0.400
ALIGN (VALVE)	2	2	1	0.518	0.466
POSITION (CAP)	2	3	1	0.427	0.384
TURN (VALVE)	2	4	2	0.336	0.269
STOW	1	0	0	0.850	0.850
GRASP/RELEASE	0	0	0	1.	1.
INSERT	0	0	0	1.	1.

* The reduction factors α's are calculated according to the following multiplicative function of attribute levels: $\alpha_{ij} = (1 - \beta_1 b_1) \cdot (1 - \beta_2 b_2) \cdot (1 - \beta_3 b_3)$, where β_1, β_2, β_3 are parameters independently measured under a set of baseline conditions (i.e., those tip-nodes with "0" attribute level shown in Figure 5-2); $\beta_1 = 0.15$, $\beta_2 = 0.13$, $\beta_3 = 0.05$ were used.

Execution time was calculated for subtask performance under automatic command using the equation (3) in Figure 5-2. The motion element performance times, t_{ij}, were taken from results under fixed command. Mental decision/transition times, ε_{ij}, were estimated from the performarce data under the automatic command itself. Ideally, these variables should have been determined independently, if possible sources of data independent of experimental situations were available. Fortunately, since the magnitude of these variables is rather small (in the order of one to three seconds), the effect on overall prediction accuracy is small. The results are summarized in Table 5-3, which also gives the observed execution times from the experiment for comparison.

The predicted execution times are quite accurate. This can be seen in Figure 5-3, which plots the predicted versus observed data from Table 5-3 in logarithmic scale. The correlation coefficient was computed and found to vary from 0.79 to 0.92 for the six subjects.* The root mean square (RMS) error is 6.2% of the average predicted execution time, which is quite satisfactory for most practical applications.

The data comparison has provided evidence for the simple additive model in predicting execution time performance in the traded control mode. As the verification of the model in task definition and augmented control mode is relatively straightforward, it appears that the model would be adequate in providing simple prediction. Given the approach used, the time required for a trained operator to perform a simulated manipulation task can be predicted to within about 7% by a simple additive-multiplicative function of task elements and factors. Although the approach is narrow in requiring the detailed micro-attribute analyses and in being

* These values are somewhat inflated due to the method used to determine some of the parameters, as described in the preceding paragraph.

TABLE 5-3

CALCULATED AND OBSERVED TASK EXECUTION TIMES

TASKS	CALCULATED EXECUTION TIME (SEC)	OBSERVED[+] EXECUTION TIME (SEC)	PREDICTION ERROR
VALVE CLOSING (NORMAL)	102.54	104.33 ± 27.12	-1.7%
VALVE CLOSING (DEGRADED)	100.04	104.67 ± 16.47	-4.4%
CAP REPLACEMENT (NORMAL)	134.95	127.00 ± 19.22	6.3%
CAP REPLACEMENT (DEGRADED)	171.66	160.84 ± 25.43	6.7%
VALVE OPENING (NORMAL)	74.95	71.83 ± 7.48	4.3%
VALVE OPENING (DEGRADED)	72.12	69.50 ± 7.72	3.8%
STOW (NORMAL)	20.26	20.17 ± 7.96	0.4%
STOW (DEGRADED)	18.42	18.17 ± 7.99	1.4%

[+] POPULATION MEAN ± STANDARD ERROR FOR SAMPLES OF SIZE 6

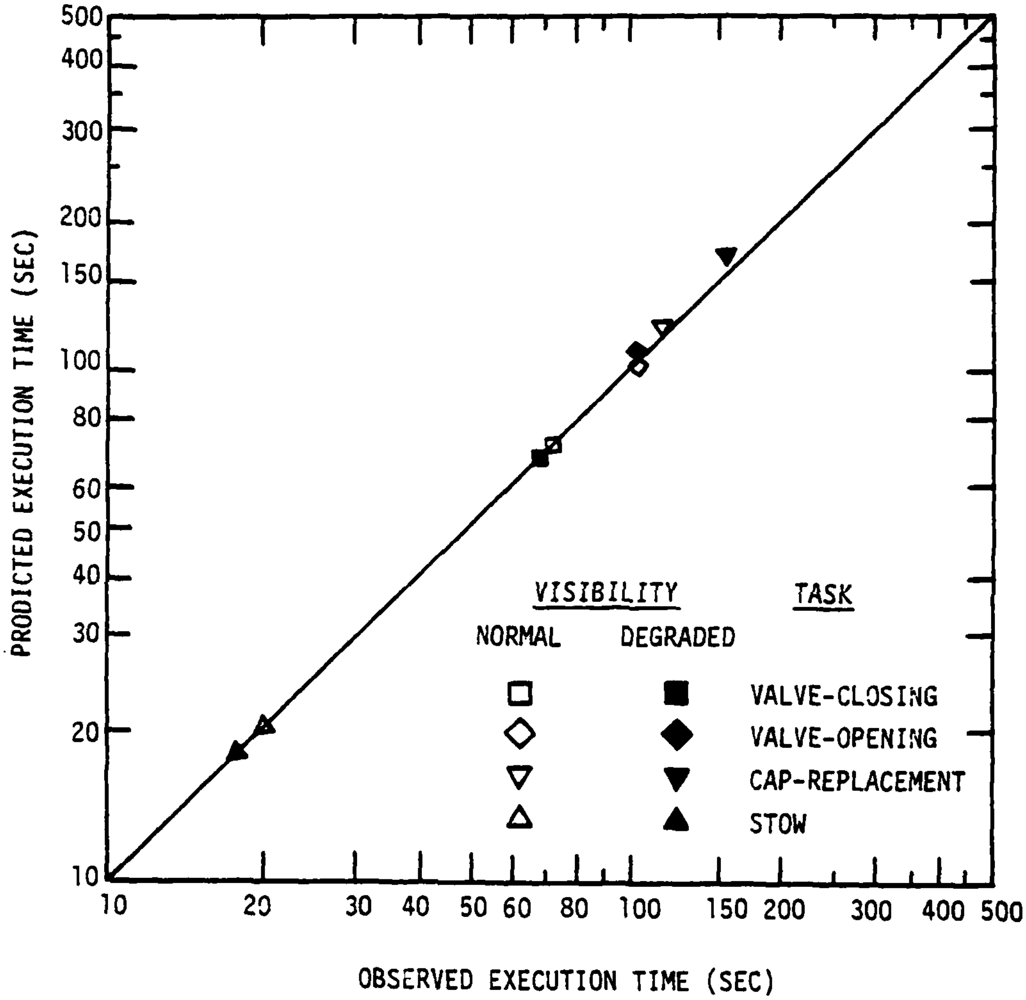


FIGURE 5-3.
PREDICTED VS. OBSERVED TASK EXECUTION TIMES

limited to error-free skilled manipulation, it is powerful in estimating performance improvement with only simple calculations.

Performance Index. Given the ability to predict task time performance, it is possible to construct or to define certain performance measures that are useful in comparing various features of command language. A first step is to collect a group of relevant performance measures that focus on comparing various command or computer-aiding functions. Those measures discussed in Sections 5.2.1 and 5.2.2 were evaluated against a set of aiding techniques tested in the program years, and the result is presented in Figure 5-4. Those of "functional" characteristics involved judgments, based on experience rather than evaluation. These functional characteristics will be discussed in the next section. Those that belong to performance characteristics were based on experimental data, observation, and extrapolation. As shown in the previous discussions, performance of aiding techniques may depend on task situations, and thus may not be generally effective or only occasionally effective, as indicated in each performance dimension in Figure 5-4.

A second step is to assemble a combined index that, for selected tasks, differentiates system performance. One central concept concerns the power of command language, which is considered to be the equivalent total output volume (in terms of productivity or capacity) per unit operator input. The definition needs to be more specific to provide a quantifiable measure. For example, Knight (1963) defined information processing power of a computer instruction set as:

$$P = \frac{[(L-7)\ Mk_1]^{k_2}}{k_3(\bar{t}_x + \bar{t}_{ion})}$$

FUNCTIONAL CHARACTERISTICS

Key:

H ≤ 60 ms
L ≤ 20 ms
VL ≤ 5 ms
B ≥ 16 K Core
S ≤ 8 K Core
H = High
L = Low

M = Medium
T = Training
P = Preprogram

	PROCESSING TIME REQUIREMENTS *	PROCESSING SPACE REQUIREMENTS	ADDITIONAL COSTS IN HW/SW INTERFACE	OPERATOR INTERFACE COMPLEXITY	PROBLEM FLEXIBILITY SITUATION GENERALITY	INCORPORATE UNCERTAINTY	EXTRA EFFORT REQUIRED
Resolved Motion Control	H	S-M	-	L	L	M	T
Fixed Symbolic Command	L	S	1-3K	L	L	M	T
+ Variable Symbolic Command	L	S-M	1-3K	H	H	L	T,P
+ High - Level Command Structure	L	S-M	1-3K	H	H	L	T,P
Program State Feedback	VL	S	<1K	L	H	M	-
Sensor (Force) Monitor	L-M	S	3-10K	L	H	H	-
Symbolic Sensor Graphic	M-H	B	2-10K	L	M	H	T

PERFORMANCE CHARACTERISTICS

Key:

G = Generally Effective

0 = Occasionally Effective

	PRECISION IMPROVEMENT	SPEED IMPROVEMENT	ACCURACY IMPROVEMENT	PROCEDURAL COMPLEXITY REDUCTION	UNCERTAINTY REDUCTION	SIMULTANEITY REDUCTION	INCREASED FLEXIBILITY
Resolved Motion Control	-	G	G	-	-	G	0
Fixed Symbolic Command	0	G	0	-	0	-	-
+ Variable Symbolic Command	G	G	G	-	-	G	G
+ High - Level Command Structure	-	G	G	G	-	G	0
Program State Feedback	-	-	G	-	0	-	-
Sensor (Force) Monitor	-	0	G	-	G	0	-
Symbolic Sensor Graphic	G	0	G	0	G	-	0

* Using servoarm manipulator direct joystick control system (see Appendix b) as a baseline system

FIGURE 5-4.
CHARACTERISTICS OF SAMPLED AIDING TECHNIQUES

where L is the instruction length in bits, M is the memory size in words, $\bar{t}_x$ is the mean instruction execution time, $\bar{t}_{ion}$ is the mean non-overlapped I/O time, k_1 is a coefficient whose value depends on word length, and k_2, k_3 are constants. This definition of instruction power is useful, but is hardware-oriented.

Based on the analysis in performance model validation in the last section, one candidate measure of command language power in manipulation control can be defined as the ratio of the volume of motion elements (i.e., the total number of tip nodes expanded by a multi-level tree structure such as one shown in Figure 5-2) to command definition time (the sum of all d's in Figure 5-1), or

$$P = \frac{(k_1\ k_2\ N)^L}{\sum_{j=1}^{L} c_j \cdot (d_0 + \varepsilon_0)}$$

where L is the level of command abstraction; N is the total level of discreteness; k_1 and k_2 are the level and degree (%) of command replication and repetition, respectively; $(d_0 + \varepsilon_0)$ is the top-level command definition time; c_j are parameters based on command mixes specification and vary within the range of 1 to 2, with full use of augmented control to full use of traded control. The power of two command sets can then be compared with the use of benchmark tasks using the above formula. As the power index increases exponentially with L while it only proportionally decreases with L as indicated in the relation described above, it appears that, in most cases (with sufficiently large N), the high level abstraction is a definite plus.

5.3 System Extention and Future Research Needs

5.3.1 Extension of TOSC Language. From previous analysis, it becomes immediately apparent that current system performance could be improved with the features described in the following added to the system. Fortunately, current implementation allows immediate extension of the following features due to modularly-structured software design:

(1) Increased number of low-level automated functions (as fixed primitives).

  (a) Localized sensor-based steering.
  (b) Force compliance.

(2) Improved replica control.

  (a) Trajectory calculation-computational overhead.
  (b) Relative geometry (point/path).

(3) Flexible command organization and chaining.

  (a) Conditioned construction of chain.
  (b) Control looping.

(4) Expandability -- allowing for:

  (a) Additional physical constraints.
  (b) Additional force/contact constraints.
  (c) Increased automatic/learning functions.
  (d) Increased database.

(5) Data compactness.

   (a) Formal grammatical rules.
   (b) Formal state definition -- command, sensors, control mode, effector.
   (c) Relative displacement, orientation, and distance to objects.

5.3.2 Future Research Needs. This section summarizes what we believe to be primary research needs in machine-to-man communication in advanced teleoperation:

(1) How can spatial information best be presented; conversely, will graphic geometric feedback results in improved task performance comapred to video feedback of limited quality and bandwidth? Performance must be compared in the speed and accuracy achievable, the complexity of tasks that can be accomplished, the unburdening provided, and the communication bandwidth required.

(2) In what conditions will geometric feedback with predictive/ directive information results in improved performance compared to full bandwidth video?

(3) What level of detail and what form of spatial presentation is essential for various levels of control automation? The evaluation may use a set of representative underwater manipulation tasks with varying degrees of complexity, different viewing orientations and scope, and varying response times.

(4) What are the best performance measures for the man-machine information system and its several components? Is there a suitable overall criteria measure? Consideration includes the following measures:

Task Performance

(a) Speed - time to complete.
(b) Accuracy - contact errors or RMS accuracy.

Task Difficulty

(a) Complexity - number of task elements.
(b) Dynamics - movement of elements.
(c) Orientation - presence of reference cues.
(d) Intermittency - blanking of communications.
(e) Turbidity - degradation of communications by noise (S/N).

Unburdening

(a) Secondary task capability.
(b) Percent of control by machine.
(c) Difficulty of task (capability of man-machine system).

System Performance

(a) Bandwidth required.
(b) Response time.
(c) Training time.

(5) What are the relationships between mode of presentation, context, content, and task performance measures?

(6) A better empirical database to specify how performance degrades as a correspondence between sensor (range, resolution), communication (noise, band-limited, transmission rate), and local presentati^n (resolution, delay) degrades.

6. REFERENCES

Adolfson, J. and Berghage, T. Perception and Performance Underwater. New York: John Wiley and Sons, 1974.

Ambler, A.P., Barrow, H.G., Brown, C.M., Burstell, R.M., and Popplestone, R.J. A Versatile Computer-Controlled Assembly System. Third International Joint Conference on Artificial Intelligence, Stanford, 1973:298-307.

Ashford, F. The Aesthetics of Engineering Design. Business Books, Ltd., London, 1969.

Battelle. U.S. Navy Diving Gas Manual, Washington, DC, U.S. Navy Supervisor of Diving (1971) (Navships 0994-003-7010).

Battelle. Underwater Vehicle Work Systems and Tool Suits. Columbus, Ohio, 1976.

Bejczy, A.K, and Paine, G. Displays for Supervisory Control of Manipulators. Proceedings of Thirteenth Annual Conference on Manual Control, MIT, Cambridge, MA, June 1977.

Bennett, J. L. The User Interface in Interactive Systems. In Annual Review of Information Science and Technology, C. A. Cuadra (Ed.), Vol. 7, Washington, DC: American Society for Information Science, 1972:159-196.

Berson, B.L., Crooks, W.H., Shaket, E., and Weltman, G. Man-Machine Communication in Computer-Aided Manipulation. Perceptronics, Inc. (Woodland Hills, CA) Technical Report PATR-1034-77-3/1, March 1977.

Bertsche, W.R., Logan, K.P., Pesch, A.J. and Wernli, R.L. Evaluation of the Design and Undersea Work Capacity of the Work Systems Package. NOSC TR 214, U.S. Navy, April 1978.

Bertsekas, D.P. and Rhodes, I.B. On the Minimax Reachability of Target Sets and Target Tubes, Automatica, 1971, 7:233-24.7.

Bien, A. and McDonough, P.J. Naval Application of Man-In-The-Sea Concepts. SRI NWRC 7000-212, 1968.

Bryant, S.B. and Funk, C.J. Performance study of present and near-future diving viewing systems. Naval Undersea Center, San Diego, CA, July 1972 (NUC-TP-302).

Busby, R.F. Review of Manned Submersibles Design, Operation Safety and Instrumentation. Report, R. Frank Busby Associates, Arlington, VA, July 1978.

Busby, R.F. Underwater Inspection/Testing/Monitoring of Offshore Structures. Ocean Engineering, 1979 6(4):355-491.

Carbonell, J.R. Artificial Intelligence and Large Interactive Man Computer Systems. 1971 IEEE System, Man, and Cybernetics Society Annual Symposium Record, October 1971, Anaheim, CA, pp. 167-172.

Card, S.K., Moran, T.P., and Newell, A. The keystroke-level model for user performance time with interactive systems. Report SSL-79-1, Xerox Palo Alto Research Center, Palo Alto, CA, March 1979.

Cevveny, R.P. and Knight, K.E. Performance of minicomputers. Proceedings of 2nd Texas Conference on Computer Systems, November 1973, 28.1-28.7.

Chapanis, A. and Lockhead, G. A test of the effectiveness of sensor lines showing linkages between displays and controls. Human Factors, 1965, 7(3): 219-229.

Chu, Y. Adaptive Allocation of Decision Making Responsibility Between Human and Computer in Multi-Task Situations, Ph.D. Thesis, University of Illinois (Urbana, IL), 1978.

Chu, Y. and Rouse, W.B. Adaptive Allocation of Decision Making Responsibility between Human and Computer in Multi-task Situations. IEEE Trans. Syst. Man Cybern., December 1979, SMC-9(12):769-78.

Corker, K., Mishkin, A. and Lyman, J. Achievement of a sense of operator presence in remote manipulation. Report Biotechnology Laboratory, UCLA, 1980.

Corkill, D.D. An organizational approach to planning in distributed problem solving systems. COINS Technical Report 80-13, Computer and Information Science, University of Massachusetts, Amherst, MA, May 1980.

Corliss, W.R. and Johnsen, E.G. Teleoperator Controls, NASA SP-5070, 1968.

Crooks, W.H., Shaket, E., and Alperovitch, Y. Man-Machine Communication in Computer-Aided Remote Manipulation. Perceptronics, Inc. (Woodland Hills, CA) March 1978.

Crooks, W.H., Shaket, E., Chu, Y., and Alperovitch, Y. Man-Machine Communication in Remote Manipulation. Technical Report, Perceptronics, Inc. (Woodland hills, CA) March 1979.

Crooks, W.H., Steeb, R., and Weltman, G. Design of a Video System Providing Optimal Visual Information for Controlling Payload and Experiment Operations with Television. Design Report PTR-1021-74-10/28, Perceptronics, Inc., (Woodland Hills, CA) October 1974.

Dijkstra, E.W. Structured Programming. Chapter 1, Academic Press, 1972.

Drenning, R.P. Manipulator/Deep Ocean Tool Work System. Westinghouse Ocean Research and Engineering Center, Annapolis, MD, undated paper.

Engel, S.E. and Granda, R.E. Guidelines for Man/Display Interfaces. Poughkeepsie Laboratory, IBM Technical Report TR002720, December 1975.

Estabrook, N.B. An Underwater Work Systems Package, Proc. 2nd Conf. Remotely Manned Systems (RMS), Los Angeles, CA, June 9-11, 1975.

Fahlman, S.E. A Planning System for Robot Construction Tasks. Artificial Intelligence, 1974, 5:1-50.

Falkoff, A. and Iverson, K.E. The Design of APL. IBM Journal of Research and Development, July 1973.

Ferrari, D. Computer Systems Performance Evaluation. Englewood Cliffs, NJ: Prentice-Hall, 1978.

Ferrell, W.R. Command Language for Supervisory Control of Remote Manipulation. In Remotely Manned Ssytems, E. Heer (Ed.), California Institute of Technology, 1973.

Ferrell, W.R. Remote Manipulation with Transmission Delay. IEEE Trans. Human Factors in Electronics, 1965, HFE-6:24-32.

Ferrell, W.R. and Sheridan, T.B. Supervisory Control of Remote Manipulation. IEEE Spectrum, October 1967:81-88.

Fikes, R.E. and Nilsson, N.J. STRIPS: A New Approach to the Application of Theorem Proving to Problem Solving. Artificial Intelligence, 1971, 2:189-208.

Finkel, R., Taylor, R., Bolles, R., Paul, R., and Feldman, J. AL, A Programming System for Automation. Stanford Artificial Intelligence Laboratory (Stanford, CA) Memo AIM-243 (STAN-CS-74-456), November 1974.

Fitts, P.M. and Seeger, C.M. S-R compatibility: spatial characteristics of stimulus and response codes. J. of Experimental Psych. 48:199-210, 1953.

Foley, J.D. and Wallace, V.L. The Art of Natural Graphic Man-Machine Conversation. Proceedings of the IEEE, June 1975, 62(4).

Fleishman, E. A. Toward a Taxonomy of Human Performance. American Psychologist, December 1975:1127-49.

Freedy, A. Learning Control in Remote Manipulator and Robot Systems in Learning Systems, K. S. Fu (Ed.) ASME publication, 1973.

Grady, W.E. and Fridge, D.S. How to Select Diving Systems in Off-shore Applications, Ocean Industry, April 1978:51-56.

Goertz, R.C. Mechanical Master-Slave Manipulator. Nucleonics, November 1954, 12:45-46.

Heckman, P., and McCracken, H. An Untethered, Unmanned Submersible, Ocean 79 Proceedings, San Diego, CA, September 1979.

Hill, J.W. Study to Design and Develop Remote Manipulator Systems. Two measures of performance in a peg-in-hole manipulator task with force feedback. Proceedings: Thirteenth Annual Conference on Manual Control, M.I.T., 1977.

Hill, J.W. and Sword, A.J. Study to Design and Develop Improved Remote Manipulator Systems. NASA CR 2238, Stanford Research Institute, April, 1973.

Jagacinski, R.J., Repperger, D.W., Moran, M.S., Ward, S.L. and Glass, B. Fitt's Law and the Microstructure of Rapid Discrete Movements. Journal of Experimental Psychology: Human Perception and Performance, 1980, 6(2):309-320.

Jet Propulsion Lab. Machine Intelligence and Robotics: Report of the NASA Study Group. Jet Propulsion Laboratory 730-51, California Institute of Technology, September 1979.

Johannsen, G. and Rouse, W.B. Prospects of a mathematical theory of human behavior in complex man-machine system tasks. Proceedings of the Fourteenth Annual Conference on Manual Control, University of Southern California, April 1978.

Kennedy, T.C.S. The Design of Interactive Procedures for Man-Machine Communication. International Journal of Man-Machine Studies. 1974, 6:309-334.

Kinney, J.S., et al. Effects of Turbidity on Judgements of Distance Underwater. Perceptual and Motor Skills, 1969, 28:331-333.

Knight, K.E. A study of technological innovation: The evolution of digital computers. Ph.D. Thesis, Carnegie Institute of Technology, Pittsburgh, PA, November 1963.

Lesser, V.R. and Corkill, D.D. Functionally Accurate Cooperative Distributed Systems. Proceedings of IEEE 1979 Conference on Cybernetics and Society, Denver, CO, October 1979, pp. 346-53.

Licklider, J.C.R. Man-computer symbiosis. IEEE Trans. on Human Factors in Electron. HFE-1(1):4-11, March 1960.

Lozamo-Perez, T. and Winston, P.H. LAMA: A Language for Automatic Mechanical Assembly. (Boston, MA) Fifth International Joint Conference on Artificial Intelligence. August 1977:710-716.

Martin, J. Design of man-computer dialogues. Englewood Cliffs, NJ: Prentice-Hall, 1973.

McGovern, D.E. An Investigation of supervisory control of manipulation. Mechanism and Machine Theory. 1977, 12.

McGovern, D.E. Factors Affecting Control Allocation for Augmented Remote Manipulation. Ph.D. Dissertation, Stanford University, 1974.

Miller, L.A. and Thomas, J.C. Jr. Behavioral Issues in the Use of Interactive Systems. International Journal of Man Machine Systems, 1977, 9:509-536.

McCormick, E.J. Human Factors Engineering. McGraw-Hill, 1970.

Minsky, M. Automation and Artificial Intelligence in Science, Technology, and Modern Navy. Thirtieth Anniversary 1946-1976. E. I. Salkovitz (Ed.) Office of Naval Research, Department of the Navy, Arlington, VA.

Moray, N. Attention, Control, and Sampling Behavior. In T.B. Sheridan and G. Johannson (Eds.) Monitoring Behavior and Supervisory Control, New York: Plenum, 1976.

Nevin, J.L. and Whitney, D.C. Computer Controlled Assembly. Scientific American, January 1978.

Nilsson, N.J. Principles of Artificial Intelligence. Palo Alto, CA: Tioga Publishing, 1980.

Ocean Industry. New Concept: Modular Offshore Services. August 1978: 56-57.

Ocean Systems, Inc. What do those expensive divers really do? Offshore Services Magazine, September 1977:78-81.

Ocean Technology Department, NOSC. Ocean Technology--A Digest. NOSC Technical Document 149, March 1978.

Paul, R. Evaluation of Manipulator Control Programming Languages. IEEE Eighteenth Conference on Decision and Control, December 1979, Fort Lauderdale, FL.

Paul, R. Trajectory Control of a Computer Arm. IJCAI, 1971:385-390.

Paul, R.L. Performance Measurement and Evaluation of General Purpose Manipulators. Proceedings of Second International Joint Conference on Artificial Intelligence, London, 1971:385-390.

Pesch, A.J. and Bertsche, W.R. Performance measurement for undersea systems. In T. B. Sheridan (Ed.) Performance Evaluation of Programmable Robots and Manipulators, N.B.S. SP-459, 1976.

Pesch, A.J., Klepser, et al. Capabilities of Operators as Divers: Submersible Manipulator Controllers in Undersea Tasks. Groton, CT, General Dynamics Corporation, AD-716532, June 1970.

Pesch, A.J., Hill, R.G., and Allen, F.L. At-Sea Operator Performance of Small Submersible Manipulators. General Dynamics Electronic Boat Division Technical Report No. U-413-71-031, 1971.

Ramsey, H.R. and Atwood, M.E. Human Factors in Computer Systems: A Review of Literature, Technical Report SAI-79-111-DEN, Science Applications, Inc. Englewood, CO, September 1979.

Ranadive, V. Video Resolution, Frame Rate and Gray Scale Tradeoffs Under Limited Bandwidth for Undersea Teleoperation. Technical Report, Man-machine Systems Laboratory, MIT, September 1979.

Roscoe, S.N. Airborne displays for flight and navigation. Human Factors 10(4):321-332, 1968.

Rouse, W.B. System Engineering Models of Human-Machine Interaction. New York, New York: Elsevier North Holland, 1980.

Rouse, W.B. Design of Man-computer Interface for On-line Interactive Systems. Proceedings of the IEEE, June 1975, 63(6):847-857.

Rechnitzer, A.B. and Sutter, W. Naval Applications of Remote Manipulation, in E. Heer (Ed.) Remotely Manned Systems, California Institute of Technology, 1973.

Sacerdoti, E.D. Problem Solving Tactics. Technical Note 189, SRI International, Menlo Park, CA, July 1979.

Sacerdoti, E.D. Planning in a Hierarchy of Abstraction Spaces. Third International Joint Conference on AI, 1973:412-422.

Sacerdoti, E.D. A Structure for Plans and Behavior. Stanford Research Institute Artificial Intelligence Center, Technical Note 109, 1975.

Schneider, M.H. Task Analysis for Undersea Manipulators. M.S. Thesis, MIT, 1977.

Schutz, H.G. An evaluation of formats for graphic trend displays--Experiment II. Human Factors, 1961, 3(2):99-107.

Schweppe, F.C. Uncertain Dynamic Systems. Englewood Cliffs, NY: Prentice-Hall, 1973.

Seireg, A. and Kassem, E. Design considerations for a computer controlled platform to support working divers. Ocean '77 Proceedings of Third Annual Combined Conference of IEEE and MTS, 1977.

Sheridan, T.B., Brooks, T.L., Takahashi, M., and Ranadive, V. How to Talk to a Robot. Proceedings of IEEE International Conference on Cybernetics and Society in Denver, CO, October 1979:33-35.

Sheridan, T.B. Evaluation of Tools and Tasks: Reflections on the Problem of Specifying Robot/Manipulative Performance. In T.B. Sheridan (Ed.) Performance Evaluations of Programmable Robots and Manipualtors, NBS SP459, October 1976.

Sheridan, T.B. and Verplank, W.L. Human and Computer Control of Undersea Teleoperators. Man-Machine System Laboratory, MIT, 1978.

Shilling, C.W., Werts, M.F. and Schandelmeier, N.R. (Eds.) The Underwater Handbook. New York: Plenum Press, 1976.

Schneiderman, B. Software Psychology. Cambridge, MA: Winthrop Publisher, 1980.

Steeb, R., Artof, M., Crooks, W.H., and Weltman, G. Man/machine Interaction in Adaptive Computer-aided Control: Final Report. Perceptronics, Inc., (Woodland Hills, CA) NR 196-118, December 1975.

Steeb, R., Chu, Y., Clark, C., Alperovitch, Y. and Freedy, A. Adaptive Estimation of Information Values in Continuous Decision Making and Control of Advanced Aircraft. Perceptronics, Inc. (Woodland Hills, CA) Technical Repot PATR-1037-79-6, June 1979.

Steeb, R., Weltman, G. and Freedy, A. Man-Machine Interaction in Adaptive Computer Aided Control. Human Factors Guidelines. Perceptronics, Inc. (Woodland Hills, CA) Technical Report PATR-1008-76-1/31, January 1976.

Talkington, H.R. Remotely Manned Undersea Work Systems at Naval Ocean Systems Center. Naval Ocean Systems Center, San Diego, CA, 1978.

Tierney, J. P. and Walrod, R.A. Development of an electronic control system for remote controlled underwater vehicles. Ocean '77 Proceedings of Third Annual Combined Conference of IEEE and MTS, 1977.

Treu, S. Interactive Command Language Design Based on Required Mental Work. International Journal of Man-Machine Studies. 1975, 7:135-149.

Vaughan, W.S. An Analysis of Environmental and Perceptual Determinants of Display Legibility Underwater. Oceanautics, Inc. Landover, MD, April 1976.

Vaughan, W.S., Glass, R.A. and Williams, J. Legibility of Self-Luminous Display Variations Viewed Through Artificially Turbid Waters. Oceanautics, Inc., Annapolis, MD, August 1977.

Vaughan, W.S., Jr., Glass, R.A., and Williams, J. Peripheral detection and identification of self-luminous display variations in 'Ocean' and 'Harbor' viewing environments. Annapolis, MD: Oceanautics, Inc., November 1978.

Vaughan, W.S. and Mavor, A.S. Behavioral characteristics of men in the performance of some decision-making task components. Ergonomics, 1972, 15(3): 267-277.

Verplank, W. L. Symbolic and Analogic Command Hardware for Computer-Aided Manipulation. M.S. Thesis, MIT, 1967.

Verplank, W.L. Is There an Optimal Workload in Manual Control? Ph.D. Thesis, MIT, August 1977.

Verplank, W.L. Display Aids for Remote Control of Untethered Undersea Vehicles,. Proceedings of Fourth Annual Combined Conference of IEEE and MTS, September 1978.

Vertut, J. Experience and remarks on manipulator evaluation. In T. B. Sheridan (ED.) Performance Evaluation of Programmable Robots and Manipulators. N.B.S. SP-459, 1976.

Weissman, S.J. On a Computer System for Planning and Execution in Incompletely Specified Environments. Ph.D. Dissertation, Unviersity of Illinois at Urbana-Champaign, 1976.

Whitney, D.E. (a). State space models of remote manipulation tasks. IEEE Transactions on Automatic Control, 1969, AC-14(6).

Whitney, D.E. (b). Resolved motion rate control of manipulators and human prostheses. IEEE Transactions on Man Machine Systems, 1969, MMS-10(2).

Whitney, D.E., Watson, P.C., Drake, S.H. and Simunovic, S.N. Robot and Manipulator Control by Exteroceptive Sensors. Proceedings of Joint Automatic Control Conference, June 1977:155-163.

APPENDIX A

SYSTEM APPLICATION GUIDELINES

SYSTEM APPLICATION GUIDELINES

1. Overview

The program research has resulted in a number of analytic and experimental findings. This section is a compilation of those findings arranged by topical area. The intent is to provide a set of guidelines to assist designers of upcoming systems and to provide a framework within which subsequent findings can be inserted.

2. Application Areas

We have found that hierarchical, procedural network-based, command language design methodology has primary application in systems with the following characteristics:

(1) Remote or decentralized communication and control.
(2) Bandwidth-limited or multiple-information sources.
(3) Supervisory control of systems with extensive autonomy.
(4) Local and remote sites computational capability.
(5) Overload operator (with complex and time-extended visual and pre-decision tasks).

Among the types of systems sharing these characteristics are remote controlled vehicle or subsystem platforms, central command of distributed intelligent systems, teleconferencing and interactive teleprocessing systems.

The area showing the greatest immediate application in current Navy systems is that of advanced teleoperator control in a free-swimming submersible or a remote control vehicle. These systems already have a substantial amount of automation, high information and control task load, remote communication and distributed processing, and critical man-machine interaction requirements.

3. Design of Multi-Mode Communication

The study suggests the following design principles and steps that should be used in multi-mode, supervisory control communication language design.

(1) Analysis of task.

(a) Identify a complete set of behavioral elements of tasks.
(b) Identify repeated subtasks and discrete subtasks.
(c) Classify subtasks along major dimensions of taxon information requirements: precision, speed, accuracy, complexity uncertainty and simultaneity.
(d) Classify overall tasks along major dimensions of procedural requirements: abstraction, parallelism, control, uncertainty, and uniformity.

(2) Analog primitive determination.

(a) Determine the required degree-of-freedom and number of analog inputs needed concurrently.
(b) Determine the need for indexing or multiple mode analog input.
(c) Determine the analog input devices and their assignment.

(3) Symbolic primitives determination.

(a) Define fixed primitives to correspond to repeated and discrete subtasks in 1b.

(b) Determine parameters requirements for such fixed primitives according to taxon requirements in 1c.

(c) Define variable primitives screened according to taxon requirements in 1c.

(4) Composite organization.

(a) Determine analogic elements for composite organization (e.g., Paths) with facilities to define, label, modify, and display them.

(b) Determine symbolic elements for composite organization (e.g., Chains) with corresponding facilities to define, label, modify and display them.

(c) Determine scope of taxon information within each composite element.

(5) Command syntax structure.

(a) Define atomic level of sentence structure. Use constrained, standardized language used in everyday operation.

(b) Define autonomous, computer-controlled primitives and special primitives. Minimize the number of verbs and nouns.

(6) Command interface design.

(a) Define as integrated symbolic and analogic interface to correspond physically to the syntax structure in 5.

(b) Provide conceptual continuity through tactile, visual and contextual continuity.

(c) Provide immediate perceivable response and feedback.

(7) Interaction protocol design.

(a) Provide facilities for smooth transitions between control modes and command activation levels; including a definiion capability, a labeling capability, a display for review capability and editing capability, an activating procedure, and a progress display capability.

(b) Initiate delegation of the appropriate performer between human or computer.

(c) Allow for queueing of input commands and natural transfer between definition mode and execution mode.

(8) Feedback format.

(a) Use consistent structure and level-of-detail between command and feedback.

(b) Provide integrated display of taxon and procedure information including machine and program states, message in task progress, and sensor-event reports.

4. System Performance Analysis

4.1 Overall System Performance. The overall performance measure of a complex manipulation task is normally gauged using a combination of indices such as speed, accuracy, error rates, and capacity. For simplicity in exploratory studies, it is possible to use independent measures of speed and error rate as figure of merits by assigning accuracy and capacity requirements into discrete categories and allowing subjective trade-offs between speed and error rates. Much of the work on which these guidelines are based used such a method. The aiding actually provided to the human operator can be estimated in two ways:

(1) Increased system performance. The difference in overall system performance with and without aiding gives a measure of the system improvement. However, this performance gain does not reflect the unburdening provided by aiding.

(2) Unburdening. The unburdening of the operator may be measured by subjective estimation or by a change in secondary task performance with and without aiding. The formal approach is used more often than the latter in the study of manipulator control, mainly due to the problems on stability and interference of primary task performance in the use of a standard secondary task. Other alternatives, but with less sensitive measures of unburdening, are the reduction of operator control and execution time, decrease in number of degree-of-freedom to be controlled simultaneously, and the reduction in procedural variability and information complexity.

4.2 Performance of Computer-Aiding Techniques

The effectiveness of various aiding techniques toward successful manipulation can be evaluated by improved performance measures mentioned above, checks of operator acceptance, reduced system consumption, and increased capability. As the first two issues are our major concern, additional measures of performance improvement should be considered. In the augmented (serial) control, the additional control quality measures determined from estimated variations from average performance may be used. These include time, error rate, and accuracy measures, etc. In the programming (parallel) control, the additional performance measures should include programming overhead, and quality in programmed repetition (replicability). In a high-level command mode, important factors are command definition overhead and perceived command complexity. In a communicaion level, additional measures may include transition (mode switching or responsibility transfer) overhead, percent control allocation and frequency of conflict in allocation decision, etc. In general, these additional measures can be translated as functional characteristics of various aiding techniques, as follows:

(1) Predicted performance--prescribed mean and variations in standard performance.

(2) Resource requirements--processing time, space and hardware/software interface.

(3) Interface complexity--perceived complexity in command structure.

(4) Replicability--probability of successful repetition.

(5) Information (taxon and procedure) requirements.

(6) Extra effort in preprogramming--preprogramming and definition overhead.

(7) Extra effort in interaction--switching and transfer costs.

4.3 Subjective Measures

The attitude of operators toward their interaction with the computer aiding system can be examined in structured form by means of rating scales and in free form through voluntary comments and experimenter observations. Rating scale judgments should be taken after an experimental sequence is completed, rather than interrupting the continuity of the task with interim ratings. Scales that have been found to be useful are:

(1) Competition/Cooperation--To what extent was the interaction characterized by conflict?

(2) Aiding--How much unburdening did the machine provide about the dimensions of control simultaneity, complexity, uncertainty, and dexterity?

(3) Relative Effectiveness--What are your estimates of the machine's and your own quality of performance?

(4) Automation Predictability--What proportion of the procedural and spatial preprogramming was expected?

(5) Satisfaction--How satisfied were you with your own and with the machine's performance?

5. Task-Oriented Supervisory Control System Evaluation

5.1 Overview

The model used to describe the role of a man-machine command language in a supervisory controlled manipulator is an adaptation of the concept of procedural net, which was developed by Sacerdoti (1973 and 1975) to represent plans of manipulative actions. Using this representation, a problem domain is described as a hierarchy of procedural nets, each representing some subtask in the problem domain in terms of its goal, its functional elements, and their relation to the environment. This information is organized as a combination of data structures and procedural information in the procedural net. In this study we have adapted this model, believed compatible with the human perception of a complex task, to be a medium for communication of such tasks between an operator and the manipulative mechanism. Using this model as a base, principles for language organization, features, and communication protocol were derived. The model further makes clear the role of the human operator in the supervisory control loop and the functions that should be performed by an assisting computer placed in the control loop. This section presents a list of factors and issues discussed in the previous chapters related to the design and evaluation of a shared man-computer communication system.

5.2 Task-Oriented Supervisory Control Language (TOSC) Methodology

A. Preliminary Design Principles

(1) Use task and concept oriented commands.

(a) Sentence structure.
(b) Special purpose keyboard and joysticks.
(c) "Chains" as concepts at varous task levels.
(d) Concept definition capability.

(2) Use hierarchical task organization.

(a) Allow for communication at highest symbolic level.
(b) Allow for suitable level of plan detail.
(c) Allow for manual backup and mixed initiative.

(3) Allow for mixed initiative.

(a) Allow for smooth control transitions between man and machine.
(b) Initiate delegation to the better performer between operator or computer.
(c) Provide feedback of system and task status.

(4) Provide concept definition capability.

(a) Constrained standardized language.
(b) Autonomous computer controlled primitives and special primitives.
(c) "Chains" as conceptual units corresponding to task elements.

(5) Smooth control transition.

(a) Machine relinquishment of control to operator during chain execution.
(b) Manual override in automatic mode.
(c) Computer monitor and warning preventing impact and reflex.
(d) Real-time manual modification to computer controlled action.

(6) Provide feedback of system and task status.

(a) Machine status - manipulator and subsystem.
(b) Program state - command definition and execution.
(c) Task in progress - current activity and sensor control.
(d) Environment state - work space and environmental sensor data.

(7) Provide tactile, visual and contextual continuity.

(a) Natural grouping and flow of motion for operating tactile input devices.
(b) Information arrangement providing continuous eye movement throughout sentence expression.
(c) Immediate perceivable responses and feedback to reinforce the effect of action sequence.

(8) Consistent structure of language and feedback.

(9) Provide suitable levels of error recovery, redundancy and simplicity in dialog.

B. TOSCL Features (Compared with Sacerdoti's Procedural Net Model)

(1) Supervisory planning by operator instead of machine.
(2) No preprogramming or pre-structuring.
(3) On-line real-time programming.
(4) Bottom-up definition by user.
(5) Added communication aspects: sentence structure, dedicated keyboard, and multi-mode feedback, etc.
(6) Added analog input, dynamic queue.
(7) Eliminate parallel (concurrent) machine activities.
(8) Eliminate explicit goal statements for each node.
(9) No criticism and checking in formal procedure.

C. TOSCL Current Implementation Characteristics

(1) Analogic and symbolic primitives.
(2) Specific versus general commands.
(3) Direct versus resolved motion control.
(4) Sentence versus programming structure of command.
(5) Hierarchical versus event sequential structures of procedure.
(6) Task-oriented versus program-oriented communication.

5.3 Command Language Characteristics

A. General Command Structures

(1) Abstraction.

(a) Low-level versus high-level.
(b) General versus specific.

(2) Procedure.

(a) Sequential.
(b) Parallel.
(c) Hierarchical concurrent.

(3) Flow of control.

(a) Interactive sequential.
(b) Transitive sequential.

(4) Database.

(a) Relational.
(b) Hierarchical network.

(5) Command composition.

(a) Atomic.
(b) Grammatic.

B. <u>General Command Primitives</u>

(1) Analogic.

(a) Position versus rate.
(b) Replica versus switch.
(c) Direct versus resolved motion.

(2) Fixed symbolic.

(a) Spatial points.
(b) Paths.
(c) Low-level motions.
(d) Automatic functions.

(3) Variable symbolic.

(a) Command building.
(b) Protocol control.
(c) Index and quantifier.

(4) Expandability - allowing for--

(a) Additional physical contraints.
(b) Additional force/contact constraints.
(c) Increased definition learning functions.
(d) Increased database.

(5) Data compactness.

(a) Formal grammatical rules.
(b) Formal state definition - command, sensors, control mode, effector.
(c) Relative displacement, orientation, and distance to objects.

(6) Information processing power of a command set, P, is given by:

$$P = \frac{(k_1 \; k_2 \; N)^{L}}{k_3 \; (\bar{t}_e + \bar{t}_d)} \; .$$

Where:

L : highest level of command abstraction.

N : number of fixed primitive commands.

$\overline{t}_e$: execution (processing) time of instruction.

$\overline{t}_d$: non-overlap definition time.

k_1: coefficient depends on degree of repetitiveness.

k_2: coefficient depends on percent of user-defined command.

k_3: parameter based on command mixes specification.

6. Application Considerations in the Free-Swimming Teleoperator Control

6.1 Overview

Much of the shared man-computer communication and control methodology developed in this program may be demonstrated in the context of tethered or free-swimming teleoperator operations. The possibility of mapping the developed methodology into a prototype interface design incorporated with the RCV platform has recently been investigated. The study presented in this section offers an opportunity to examine the issues and applicability of various command and communication design techniques derived earlier.

The test-bed platform under consideration is the free-swimming submersible or remote controlled vehicles developed in Navy laboratories during the past few years. A primary example is a recently commenced development of unmanned free-swimming vehicles being developed at Naval Ocean System Center. The submersible, as currently configured, has a modular construction which allows expansion to accommodate additional payloads and new sensor systems. The vehicle is microprocessor-controlled and is capable of following preprogrammed instructions. It can be programmed via a computer console and an umbilical cable which is disconnected after the initial programming phase. Additional control and data sensors will be added (Heckman and McCracken, 1979). These include an acoustic control link, an acoustic slow-scan television link, and a fiber optic communication link. The end result will be a system which is relatively inexpensive, free from cable drag and cable-handling problems and one which shouid automously perform rudimentary tasks without direct operator control (Talkington, 1978).

As several studies of underwater tasks analysis (e.g., Bien and McDonough, 1971; Busby, 1979) have concluded, most operations require rudimentary manipulation capability, even in simple inspection and survey tasks. And indeed, this rudimentary manipulation capability accomplishes a major portion of typical manipulation tasks. Therefore, of converging importance is the requirement for any undersea teleoperator to be capable of performing a low amount of manipulation such as grasp, carry and remove, etc. These attempts to provide an effective platform to achieve manipulation function have quickly revealed the need for a coordinated manipulator/ platform control and effective operator-machine communication. This study reviews the problem area and examines general design requirements for man-machine communication in platform/manipulator control.

6.2 Background

In the next decade we will need a better technology for offshore drilling, undersea exploration, undersea mining, aquaculture and rescue. As suggested by Minsky in a review of automation and artificial intelligence related to science, technology, and the modern Navy (1976), the advances in teleoperator control systems might be the key to these developments. So far, the most frequently encountered tasks, such as underwater inspection, testing and monitoring, have largely been performed by the diver. The increasingly demanding condition and more complex structures today have produced the need to develop diver alternative work systems. This need is due to rapidly increased dollar costs and risks involved in modern operations in inspection/testing/monitoring of offshore structures. As such, two alternatives had been proposed by Busby (1979) after an intensive survey of underwater inspection/testing/monitoring of offshore structures: the manned submersible and the remote controlled vehicle. He suggested that:

> "Both systems are capable of being used to produce high quality visual and photographic inspections, and both can bring some form of cleaning device (wire brush, chipping hammers) to the inspection site....
>
> The role of remotely controlled vehicles in underwater inspection/testing is less well-defined than their manned couterparts. As inspection/photographic-documentation vehicles, they appear to be excellent.... In open waters they have been used quite successfully as pipeline inspection vehicles, but around and within steel structures they have--in addition to other problems--experienced difficulties with cable entanglement and location."

Although the relative value between manned and remotely manned submersibles remain arguable, the most important tradeoff seems to be the operator's presence, his interpretive ability to see, and the costs of building and operating manned and remotely manned vehicles. When technology advances are taken into account, Talkington's (1976) argument may seem convincing:

> "It suggests that remotely operated systems are better suited for the performance of most undersea projects for at least six reasons: relative economy of development in time and equipment costs, unlimited endurance on site by virtue of the cable link to the surface, surface control and coordination of project efforts, ability to perform in hazardous areas without endangering personnel, ability to change or modify all system com,onents to meet individual tasks range needs without affecting system safety or certification status, and ease of changing crews without disrupting the mission."

In support of the above concept, several Navy laboratories and ocean industries have been involved in the development of remotely manned, tethered and untethered vehicles (e.g., submersible cable-actuated teleoperator, free-swimming vehicle and SNOOPY at NOSC, RCV-150 at Hydro Products, Inc. and deep ocean work systems at Ametek). These are new generation microcomputer-based, remote controlled vehicles which show great promise for future operations.

General issues are addressed in this study with the recognition that the ultimate success of free-swimming inspection/manipulation will depend on viewing and maneuverability on a stable platform plus adequate feedback of the machine state and environment. The problem areas resulted from a moving platform which was not considered in the laboratory experiments is summarized in the following platform-related design considerations:

(1) It should be capable of operating on a non-stationary submersible and transporting the tools and sensor devices to the work location.

(2) It should be able to compensate for the "dead weight" of the manipulator and the support system by buoyancy forces.

(3) It should stably maintain its position and orientation and correct any error resulting from the dynamic or static loading imposed by the end effector.

(4) It should provide the ability of the operator to keep "oriented" relative to the environment that the platform finds itself.

(5) It should be of modest size and of adequate reach capabilities (high flexibility and maneuverability).

(6) It should provide support for sufficient man-machine interface for vehicle navigation and control coordination.

(7) It should provide support for sufficient man-machine interface for viewing range and display.

(8) It should provide support for efficient sensor data processing and presentation.

These issues of moving platform dynamics can be largely resolved through proper design of the combined vehicle and mechanical manipulator system. As suggested oy Rechnitzer and Sutter (1973), the platform should provide (1) stability compensation; (2) reaction forces; (3) viewing; (4) access to the work; (5) power; and (6) payload. For example, current flow in manipulator workspace could impose rather severe operational constraints in terms of reduced admissible trajectory volume. New concepts in end-effector compliance and local sensor control may present a solution to this problem. Another problem of no less importance for operator supervisory control in this uncertain environment is the lack of orientation and motion references. The sense of orientation is a prerequisite to virtually all forms of control, and it is directly related to the estimate of position and dynamics. Additional reference aids may be beneficial such as the ones proposed and tested in Perceptronics' previous study for extraterrestrial manipulator video systems. The program was intended to provide optimal visual information for controlling payload and experiment operations (Crooks, et al, 1974). The reference aids included a computer-generated artificial horizon, display of vehicle-manipulator-environment geometry for orientation reference, and use of a graduate reticle for distance and dynamics estimation.

The second major problem area concerns the combined requirements of information feedback of the machine state and visual observation of the environment. When one considers moving the laboratory facility into real-world remote underwater manipulation, it is evident from the analyses in earlier sections and Chapter 2 that a great deal more effort should be first extended to circumvent the problems of environmental uncertainty (e.g., poor visibility) and limited communication capabilities (e.g., low-bandwidth).

One approach to circumvent the poor visibility, suggested by the results obtained in our earlier experimental study (see Section 4.2.3), is to employ a supervisory control mode with limited force feedback. This control mode permits the manipulator to act as an autonomous robot for short time periods, responding to its environment in the pursuit of subtask goals previously programmed in by the operator. The improved performance, in terms of speed and errors, showed that this supervisory control could partially compensate for low visibility and predictability in a structured or repetitive task (Crooks, et al, 1979). In our view, however, neither this semi-autonomous control nor the use of force reflection (primitive force monitoring or detail vector display) alone offers the entire answer to the problem of visibility. With force feedback, for example, the operator tends to lose the object when the manipulator is moved away from it. Much time is consumed in relocating already identified parts, surface, receptacles, etc., and identification by feel is never as sure as by feel and vision together. Likewise, it is unlikely that any single exteroceptive sensor (touch or proximity) under current development will adequately replace the complex geometric information now required for the operator in supervision of manipulation. It appears that the operator must be given visual feedback by video means or by computer-generated display of multi-mode sensor feedback.

Above all, the near-term technological bottleneck for efficient remote teleoperation seems to remain in the limited transmission bandwidth. Current studies along this line at MIT address the issue of tradeoffs between frame rate, resolution and grey-scale in performing manipulation tasks while viewing through a limited-bandwidth TV channel (Sheridan, et al, 1979). The underlying aiding concept here is to allow the computer to adjust visual continuity by frame rate and resolution according to task situations. Another approach of potential payoffs is to use the "computational plenty" in local control site capability to provide greater flexibility in command planning and resulting lower bandwidth requirements for display of information. One display idea, bearing a gross similarity to that of Verplank's predictor display for underwater vehicle control with delay and low frame rate (Verplank, 1978), is tc interpolate between snap-shot samples of the transmitted sensory image of the manipulator workspace. Obviously, the simulation of manipulator motion by itself would be far more challenging than that of vehicle control, due to the complexity in both control dynamics and environment interaction in teleoperator control. Nevertheless, it appears that the proper use of moving predictor symbols representing relative geometric interaction between end effector and environment could enhance performance and reduce bandwidth requirements in gross manipulation.

It follows that the main concern is the proper formatting and presentation of a variety of spatial information. Would alphanumeric, schematic, symbolic or pictorial formats alone be sufficient? If not, would combinational methods using super-position be more efficient than plain layout of information? These questions can only be answered in a more specific task context. Related to this concern, results from one of our previous experiments indicated that the alphanumeric representation of machine state received only limited acceptance from experiment subjects. The provision of state feedback in alphanumeric form resulted in no performance gain in the augmented control mode. In a traded control mode, when the computer took over, frequently performance was found to show significant gains (Crooks, et al, 1979) after subjects received sufficient practice. In addition, it was observed during the experimental study that the subjects preferred to obtain machine state information through inference and partial cues from TV viewing to the separate state feedback messages.

The above observation led to the initial evaluation of Perceptronics' 3-D stick-figure display. The results demonstrated that a considerable amount of information can be conveyed via manipulator configurations, even in a stick-figure format (see Section 4.3.2), as it has been predicted by Herman (1979). Two important categories (physical space and machine state space) of such information are:

(1) The physical state of the manipulator, such as carrying, free slewing, hitting obstacles, etc.

(2) The machine (execution) state and suggested task state such as valve turn, and waiting for manual alignment, etc.

In summary, the second major problem area addressed above represents a set of man-machine communication barriers which have so far blocked effective supervisory control of underwater teleoperation. These barriers, as shown in Figure A-1, include the factors of low visibility, moving platform, local supervisory control/state feedback, intermittent viewing/ interposed objected, and interaction with touch/force/proximity sensing.

AD-A094 482 PERCEPTRONICS INC WOODLAND HILLS CALIF F/G 5/8
MAN-MACHINE COMMUNICATION IN REMOTE MANIPULATION: TASK-ORIENTED--ETC(U)
MAR 80 Y CHU, W H CROOKS, A FREEDY N00014-76-C-0603
UNCLASSIFIED PFTR-1034-80-3 NL

3 of 3
AD
A094482

END
DATE
FILMED
2 81
DTIC

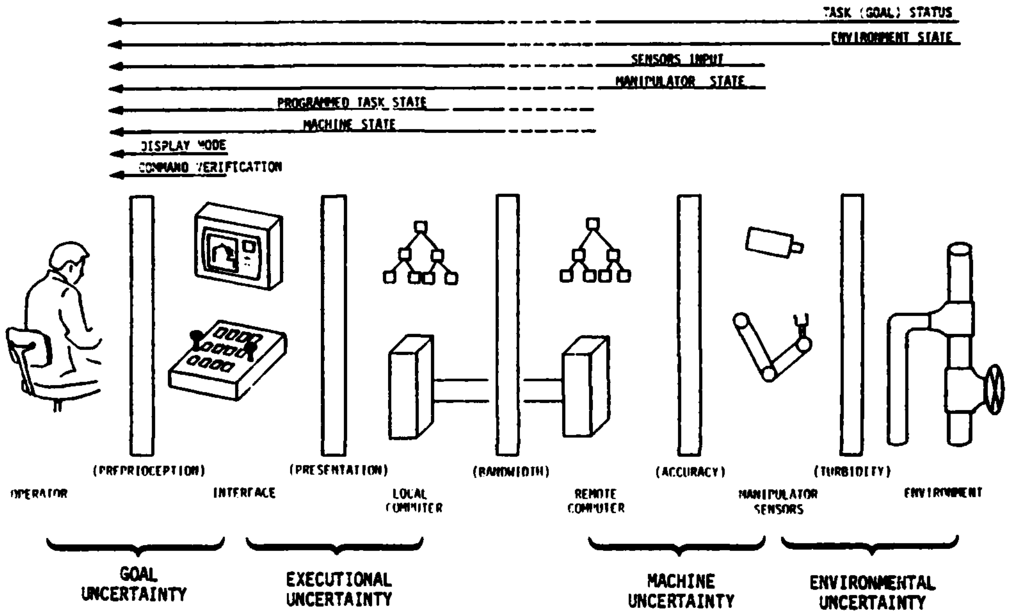


FIGURE A-1.
FEEDBACK TYPES AND BARRIERS IN TELEOPERATOR SYSTEMS

APPENDIX B

COMPUTER-AIDED MANIPULATION FACILITY

COMPUTER-AIDED MANIPULATION FACILITY

1. Overview

Perceptronics' computer-aided manipulator facility includes a hydraulic servo manipulator, a minicomputer, and a man-machine interface (Figure B-1). An operator controls the manipulator through the joysticks and pushbuttons of the control console; he observes the manipulator activities through the two-view TV displays. The operator's inputs are processed by the minicomputer; the minicomputer, in turn, controls the servo manipulator and responds to the manipulator's position-sensing potentiometers. Data communications between the minicomputer and the control console or manipulator electronics occur via the programmable interface. A review of the command language design is given in Chapter 3. The individual components of the manipulator, interface, and software are described in the following sections.

2. Servoarm Manipulator

The servoarm manipulator, shown in Figure B-2 is electronically-controlled and hydraulically-powered. The manipulator has six rotating joints (each with a full 180° movement range) plus gripper closure. The arm motions and joint numbers are (in anthropomorphic notation):

(1) Shoulder rotation.
(2) Shoulder elevation.
(3) Elbow flexion.
(4) Forearm rotation.
(5) Wrist flexion.
(6) Gripper rotation.

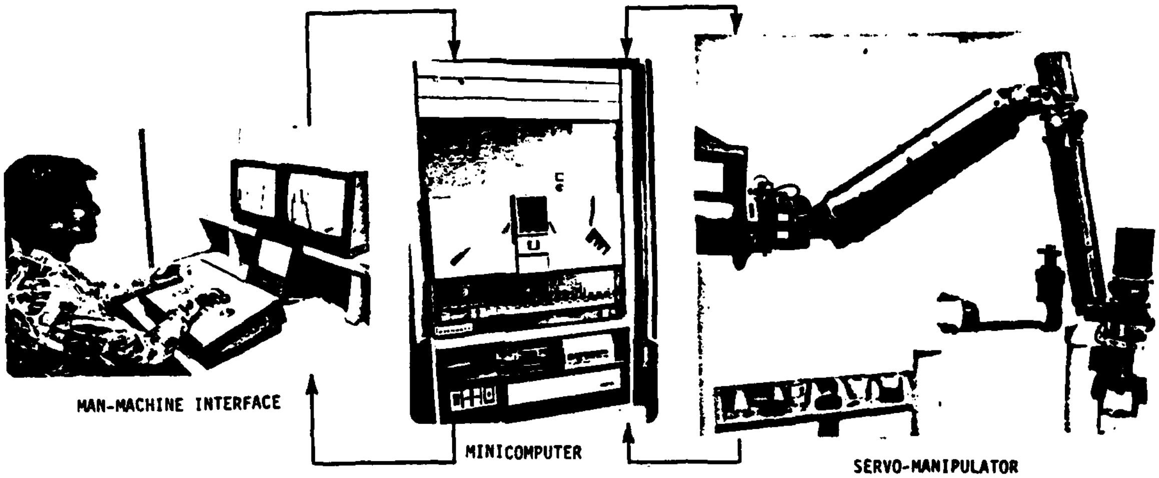


FIGURE B-1.
COMPUTER AIDED MANIPULATOR FACILITY

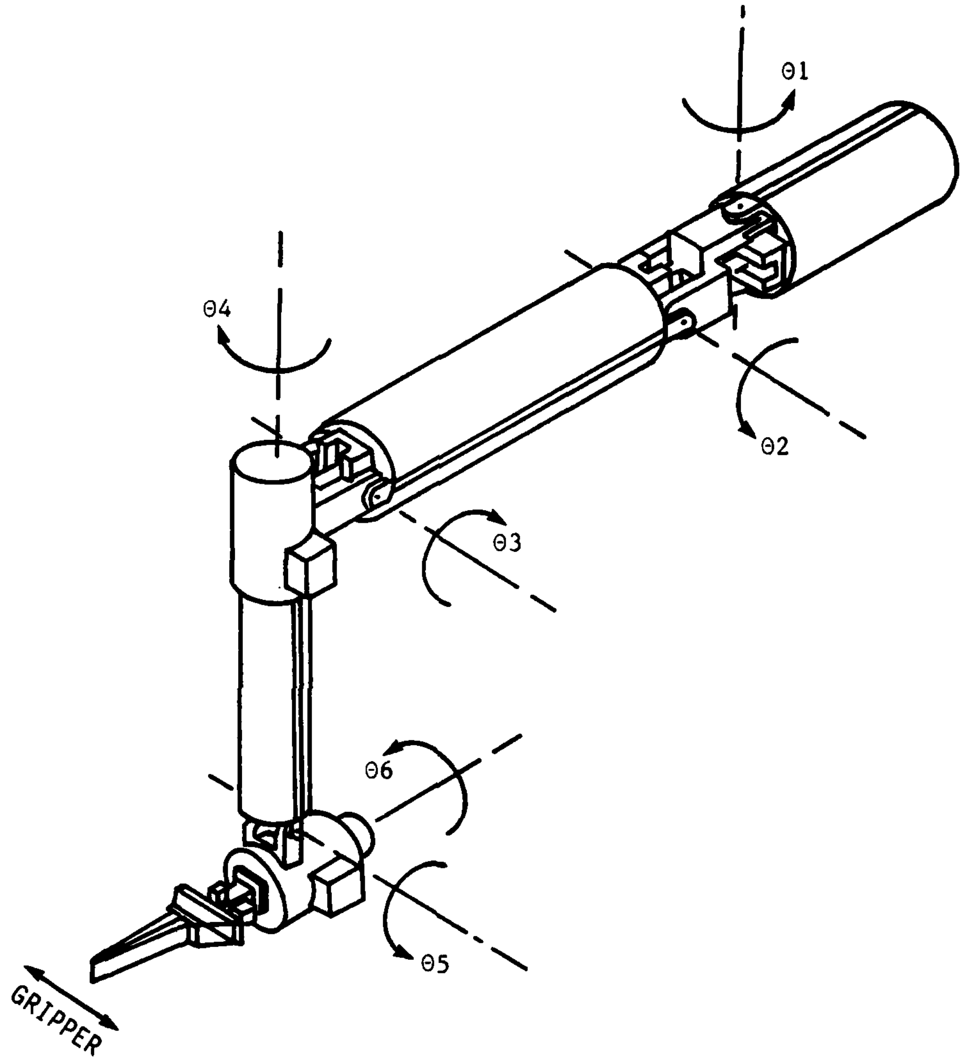


FIGURE B-2.
SERVO MANIPULATOR WITH MOTIONS
OF THE SIX ROTARY JOINTS

These motions provide the six degrees of freedom necessary to position and orient an object in the work space. With the computer in direct control of the manipulator, computer programs can provide a direct manual control mode as well as control modes ranging from computer-assisted functions (e.g., resolved motion control) to fully automated performance of routine tasks. In addition, with fully integrated computer software, an operator can change naturally between control modes to perform a variety of remote manipulation tasks.

3. Integrated Control Console

3.1 Control Modes

A library of manipulator control routines is integrated into a single computer program, which permits the operator to select any of several control routines in any desired sequence. The available mode options include:

(1) Direct Control. This mode gives the operator direct manual control of each manipulator joint angle. Each degree-of-freedom of the joystick controllers is associated with a specific manipulator joint.

(2) Resolved Motion Control (RMC). This control mode allows the operator to move the wrist along task or world coordinates. Each degree-of-freedom of the joystick is associated with movement of the manipulator end-effector along a specific X, Y, or Z axis of the work space. The operator specifies the speed and direction of motion of the manipulator wrist, and the computer calculates the required angle of each joint and outputs these as commands to the individual joints of

the manipulator. RMC frees the operator from the responsibility of determining which combination of speed and motions will produce the required trajectory.

(3) Automatic Commands. Under Automatic Motion Control (AMC), the computer assumes control and moves the arm from its current location to any preassigned location. Under Fixed Commands, the computer performs single unitary motions through the operator's keyboard commands. Under Variable command, the computer records and moves the arm automatically to previous recorded configurations (extended AMC); accepts task procedure and supervises automatic execution of complex tasks, with possible intervention and modification by the operator. Under Chained command, the computer decodes task procedures and performs the functions under the Variable Command mode.

(4) Speed Adjustment. This option allows the operator to select the maximum rate of manipulator motion. The joysticks are rate controllers with greater stick deflection, providing faster manipulator movement. The speed adjustment routine allows the operator to select one of the three rates over which the joysticks function.

(5) Other System Features. This includes enable/disable of wrist invariance function, enable/disable of force sensor messages, enable/disable of 3D display and other commands: definition, activation, edition, and alphanumeric feedback features, etc.

3.2 Console Configuration

The man-machine interface consists of a control console by which an operator can manually control arm motions, select computer assistance control functions, and observe control status.

The operator uses both joysticks and the pushbuttons to control the manipulator. The joysticks are used for manual control, and the pushbuttons are used for control mode selection. Using the pushbuttons and joysticks, the operator can smoothly take the manipulator through a sequence of tasks, selecting control modes, rates, and manual operations that are most appropriate for each subtask. For example, he may use Automatic Command for gross movement from the stowed position to the target area. He can then change immediately to RMC for fine movements and for alignment and insertion. While in the RMC mode, the operator can move the arm to the "drop" point, and then record the latter point to facilitate repetition of the task.

One alphanumeric CRT display and two TV monitors are used to provide system state feedback and two-view TV viewing of the manipulator work space. The details were discussed in Chapter 4. Also included in the display capabilities is a computer-generated three-dimensional, stick-figure display of the work environment, which is described in the next section.

3.3 3-Dimensional Display

Figure B-3 is a photograph of the Perceptronics 3-D display. The three-dimensional image is created by a patented mirror mechanism positioned in front of a fast Cathode Ray Tube (CRT). The mirrors vibrate forward and back to produce a volumetric image in the viewing area. The viewer looks directly at the CRT screen. This means that the image can be as detailed

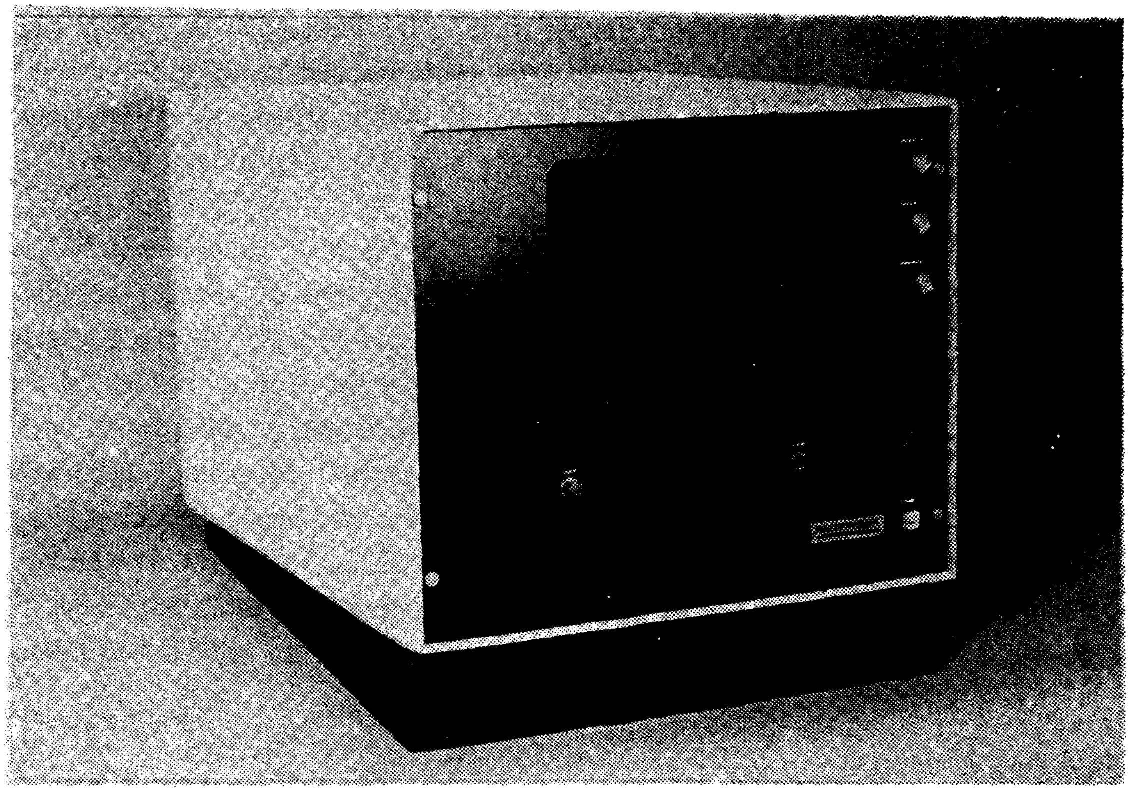


FIGURE B-3.
3-D DISPLAY TERMINAL

and as bright as the CRT itself, although actual resolution is determined by digital memory size and data throughput rates. As presently configured, the image space is about 3½ inches deep. The image can be viewed without special glasses or optics, and can be photographed by single-lens or stereo cameras. Parallax is present, so that by changing viewing position, the user can look around the image for a better view of any side.

In the computer, the image is made up of a stack of two-dimensional X-Y planes, where X and Y correspond to the dimensions of the CRT screen. Each plane represents the information at a given depth in the Z direction, and is termed a Z-plane. Z-planes are stored in digital memory, and are output sequentially in synchrony with the movement of the mirror unit. Synchronizing signals are provided by a photocell detector. Generation of a complete image requires output of all Z-planes. By outputting the complete stack of planes at 30 cps, a rate above the critical flicker rate of the human eye, a continuous three-dimensional image is achieved. The image is inherently transparent, allowing views into solids and through plane surfaces.

During operation with the manipulator, the computer calculates the image coordinates of the arm and generates a "stick figure" image of the arm on the 3-D display. As the arm moves within the work space, the displayed arm figure moves in a corresponding manner within the 3-D image volume. Present configuration of the 3-D display used in the experimental study does not allow direct user interaction with the actual image space (such as on-line measurement and rotation, etc.) and no real-time capabilities of environmental mapping are provided. What remains as an untested operation, therefore, is the force sensor attached to the manipulator wrist, in which an image of objects in the work space can be created by recording and displaying those points in space where the end-effector touched the object. Once the outlines of the object have been established by contact mapping, the operator can position the manipulator in the image space to bring the end effector into contact with the work object, and can initiate automatic control routines, thus allowing task performance in visually degraded environments.

4. Computer Control System

4.1 Control Processor

The supporting processor for the manipulator system is an Interdata Model 70 minicomputer. This machine has a memory cycle time of 1 μs and basic instruction execution times averaging between 1 and 3 μs. As presently configured, the processor system includes 48 kilo bytes of core memory, a high speed paper tape reader/punch, a line printer, a CRT alphanumeric terminal, a selector channel, a disc memory and a re-settable precision interval clock. The disc drive, CRT, and other peripherals are used to support program development work and are not part of the real time manipulator control system.

4.2 Processor Interface

Data transfer between the computer and manipulator servo-electronics and between the computer and the control console is performed by a Perceptronics +I/O Programmable Interface. Besides providing all analog-to-digital (A/D), digital-to-analog (D/A), and digital-to-digital (D/D) conversions among the system components, this interface allows the outputs of all controlled devices to be treated by the processor as if they were the product of only one device, thus simplifying the software arrangements at the processor.

The +I/O Interface contains a number of functional modules arranged along a transfer buss by which commands, data and status signals are communicated.

These modules perform such individual functions as standardizing communication with the processor, sequencing data transfers across the buss, and performing D/A conversion and output.

In addition to an interface module between the processor and interface, the +I/O includes an A/D module that is used to interface with the servo position potentiometers and control console joysticks. Thirty-two individually addressable input channels are provided. An eight-channel D/A module is used for converting and sending position commands (voltages) to the control inputs of the servos. Finally, a D/D module is used to provide the 16 input and output channels for the button and lamp arrays of the control console.

4.3 Software

The software system includes three main modules shown in Figure B-4. Module 1 is the main control process. It contains the central loop which executes most of the software functions. All the programs of primitive and non-primitive functions, joystick control and state command programs are included in this module. Module 2 is the teletype process module. It contains the programs which read the keyboard codes, then analyzes and interprets them. Module 3 is the I/O process module. It contains programs which "read" and "write" and send characters to the CRT.

The following sections summarize the status and functional structure of each module. More detailed documentation can be found in earlier reports (Crooks, Shaket, Alperovitch, 1978; and Crooks, Shaket, Chu and Alperovitch, 1979).

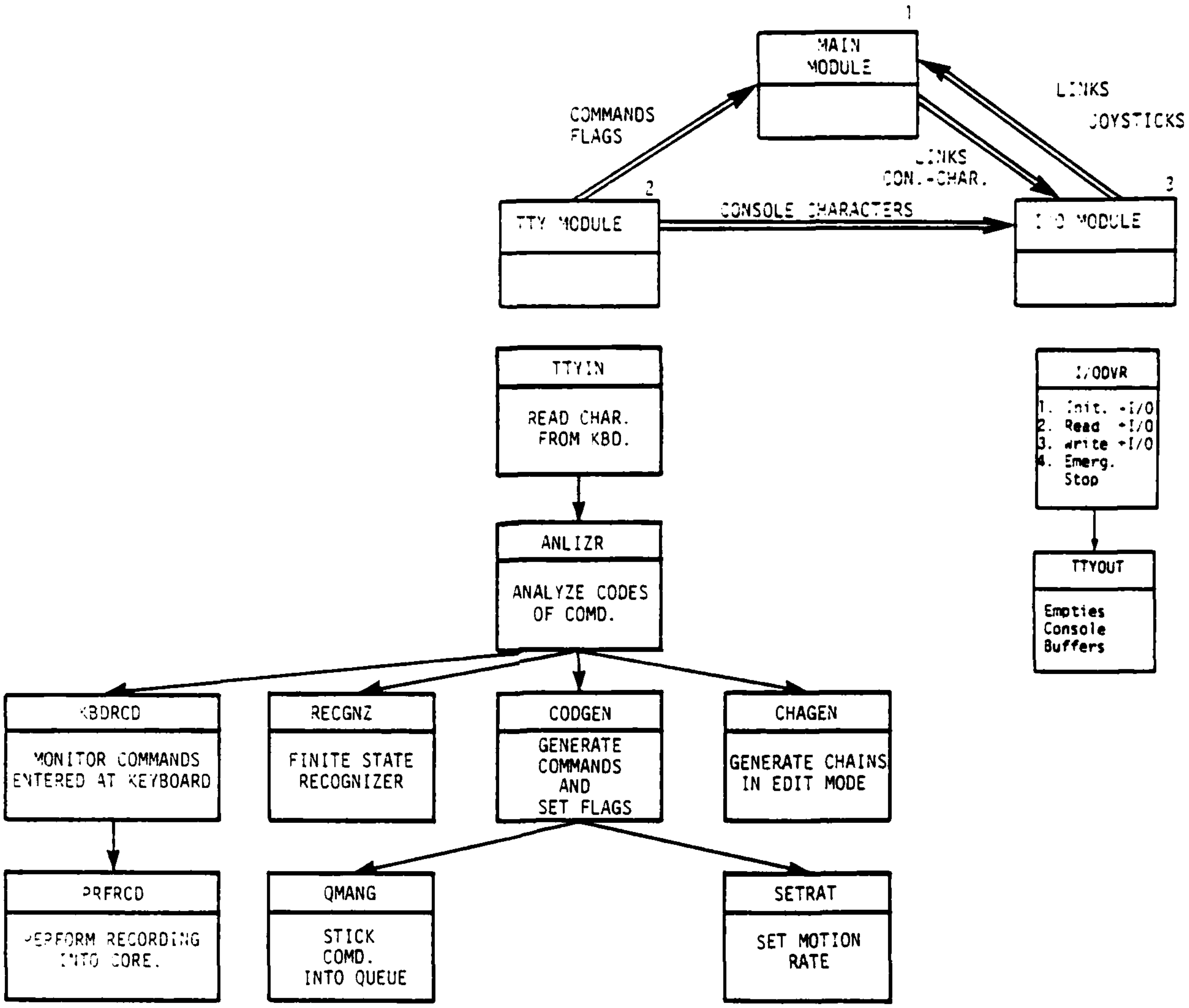


FIGURE B-4.
DATA CONNECTION MODULES

4.3.1 Main Control Process Functional Structure.

MNIP78 - Including the main control process function as shown in Figure B-5. This is the highest level program of the system. It invokes a sequence of routines which are initialized in the system. It then senses the switches of the computer's panel; and if all of them are reset, then a system calibrating program is invoked followed by termination. If at least one of the switches is set, the EXPER program is invoked and an experimental session can be conducted. When the experiment is concluded, a summary program is invoked and the data which was monitored during the session is processed and printed as a report.

EXPER - This is the central program of the system by which most of the other programs are invoked. Basically, the program is a one "DO WHILE" loop executed every 100 ms. The while condition is the same as in MNIP78. Whenever all the switches are reset the program terminates; otherwise it loops on. The main cycle is a long sequence of programs which is invoked in certain contingencies. Generally, each program has a flag which indicates whether it should be invoked or not. A short description for each program is given below.

INITAB - Initializes screen and core where points, chains and paths are stored.

READSP - Sets the feedback display format according to the selected mode of machine state feedback.

MONITOR CURRENT PRIMITIVE - Includes several automatic command monitoring programs. When a primitive command code is in the execution area, it invokes the PRIMM, which monitors primitives execution. If primitive execution is finished, INTRP is invoked and it fetches a new primitive into the execution area.

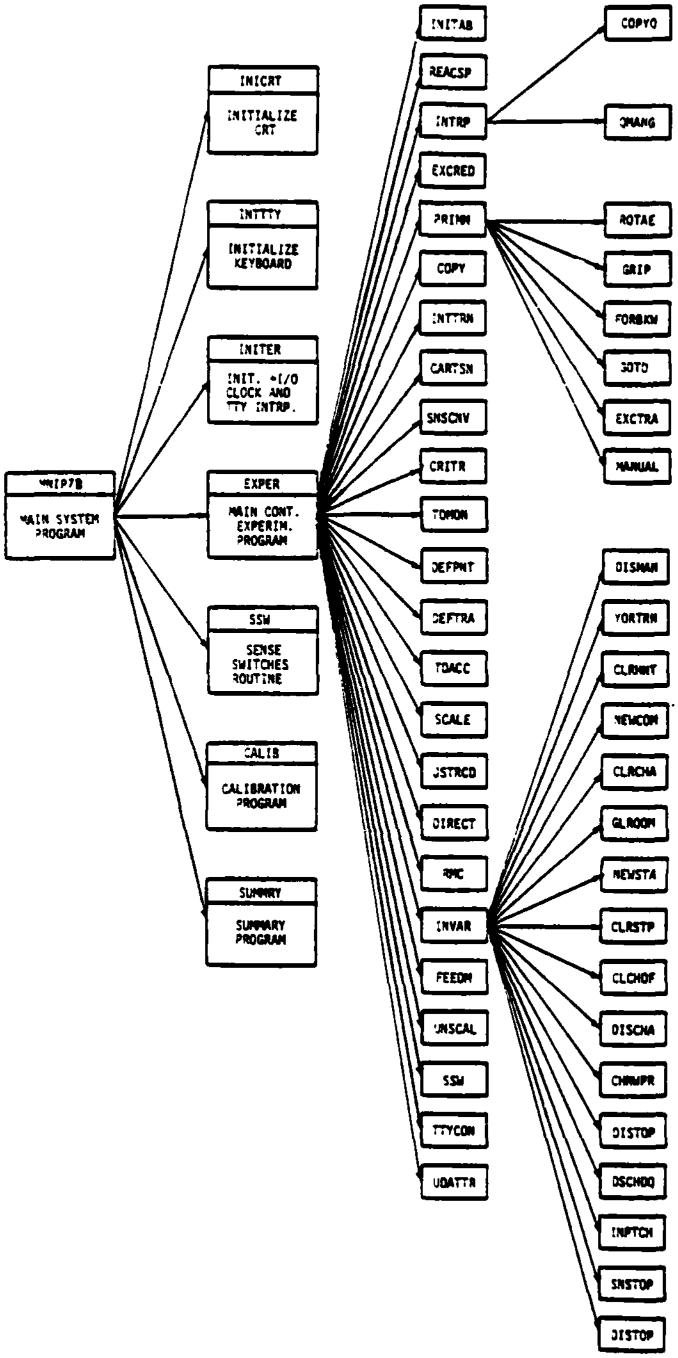


FIGURE B-5.
MAIN MODULE HIERARCHICAL TIER STRUCTURE

QMAL - Manages the queues of commands for execution.

COPYQ - Copies the invisible queue into the visible queue.

ROTATE - Performs end gripper rotation.

GRIP - Closes and opens the gripper.

FORBKW - Moves the gripper forward or backward preserving constant orientation.

GOTO - Performs goto point. A special feature is added to the "goto" function. Whenever "goto point" executes and the operator manipulates the joysticks in the same time, in order to change the final manipulator's position, the final goal point is adjusted accordingly, and the function terminates when manipulator reaches the adjusted goal point.

EXCTRA - Executes trajectory forward and backward.

MANUAL - Stops primitives execution to enable manual intervention with joysticks.

EXCRED - Is invoked to monitor commands which are submitted to execution. It is a part of the real-time monitoring system.

COPY - Copies input floating values into output floating values.

INTTRN & UDATTR & CARTSN - Transformation routines which are transforming values in link space into cartesian space.

SNSCVV - Sensors conversion routine. It processes the raw input data coming from the sensor and checks for threshold excess values.

CRITER - Sensors (stress and torque) criteria routine which protects the manipulator from damage by generating indication of warning and stop conditions.

TDMON - 3-D monitoring routine.

DEFPNT - Define point routine.

DEFTRA - Define trajectory routine. It is invoked whenever the trajectory flag is on and specific synchronization conditions are prevailing.

TDACC - Three dimensional display activation routine. It should be only invoked when the 3-D is hooked up. All the following programs are invoked only by the main cycle.

SCALE - Scales input integer values which are read from the +I/O into radius floating values.

JSTRCD - Is the second real-time monitoring program, which monitors joystick activity whenever it is operated.

Now, continuing down the main cycle loop, we see:

DIRECT - Joysticks direct control program which changes manipulator links according to values received from joysticks.

RMC - Joysticks resolved motion control program which enables manipulation in cartesian space.

INVAR - Grippers constant orientation preserving routine.

UNSCALE - Current floating point values are unscaled into integer values which are written into the I/O.

SSW - Sense switches routine.

FEEDM - CRT feedback display generating routine, including routines to display command inputs, commands in queue, sensor and system status.

NEWDQ - Displays on screen a visible queue.

DSCHDQ - Displays on screen a chain in execution.

INPTDQ - Increments pointer of displayed chain.

ERASDQ - Erases from screen displayed queue.

INPTCH - Increments pointer of displayed chain.

ERCHDQ - Erases displayed chain from screen.

SNSWRN - Displays sensor warning message.

SNSTOP - Displays sensor stop message.

4.3.2 TTY Process Functional Structure

TTYIN - Handles the codes received from the keyboard.

ANLIZR - Performs monitoring of syntactical errors and cancelations.

KBDRCD - Automatically monitors every command which is entered on the keyboard.

PRFRCD - Performs actual monitoring of command into core prefixed by value of the clock at time of invocation.

RECGNZ - Recognizes valid sequences of codes using finite state recognizer.

CHAGEN - Generates chain under the edit mode.

CODGEN - Generates codes and sets flags when valid commands are recognized.

4.3.3 I/O Process Functional Structure

I/ODVR - It is the Main program of the I/O process.

READ - Read manipulators link values into core.

COMPAR - Emergency compare provided to stop the arm whenever the difference between the input and output values exceed a certain threshold.

WRITE - Writes new values into the manipulator.

TTYOUT - Outputs characters to CRT from high and low priority buffers.

4.3.4 3-D Display Utility Programs. A set of programs which was used to perform basic graphics operations on the 3-D display is specified here. The group of programs is compatible with Fortran V and/or Fortran IV programs as well as with assembly language programs, such that default values are supplied in the absence of explicit parameters, where possible. The basic operations available are the displaying of single points and arbitrary lines (represented as a series of points) in a three dimensional graphics work space. Supporting the above will be a small number of utility functions.

A number of contiguous memory locations may be allocated as the display buffer. The buffer is considered to be partitioned into smaller contiguous areas, each of which correspond to one Z-frame or cross-section of the display space.

Although each program is accessible from Fortran, all coding is in Interdata 70 assembler language to provide byte and bit manipulation, special I/O capability, and for ease of internal communication between programs in the package.

Initialize Display - D3INIT

This routine is called to initialize a buffer. It sets all entries to zero, sets the mode for that buffer and if variable mode, sets the intensity for that buffer.

Starting Display - D3TURN

This utility routine is used to start or stop the display. It should not be used until a display buffer has been created.

Obtaining 3-D Status - D3STAT

This routine is provided to enable the user to obtain two binary inputs available from the 3-D through a sense-status command, and to tell whether or not the selector channel is busy outputting to the display. The actual status is sensed, while the selector channel is turned off in the program

D3TURN. Thus, D3STAT cannot be called until D3TURN has previously been called. After D3TURN has been called, however, the binary inputs are available to the user through D3STAT (i.e., on an asynchronous basis with respect to the operation of the 3-D and selector channel). The information as to whether or not the selector channel is busy allows the user to utilize the "backswing" of the 3-D mirror for buffer modifications if desired.

Defining Points - D3PONT

This routine inserts points into the display buffer specified by the "D3BUFF" routine. It does not cause those points to be displayed unless the specified buffer also happens to have been selected by the "D3TURN" routine.

Deleting Points - D3DELP

This allows one to delete points created by "D3PONT."

Defining Lines - D3LINE

This utility creates lines in the display space consisting of several points.

Delete Lines - D3DELL

As with D3DELP, this routine removes lines previously created with D3LINE. In fact, it is used essentially the same as D3DELP.

Point Imaging Routine - PPOINT

This routine allows the user to specify a location in the manipulator work space in floating Cartesian coordinates within the range of that work space (i.e., $-36.0 \leq x \leq 36.0$, $0.0 \leq y \leq 36.0$, $-32.0, \leq z \leq 32.0$). A point image is placed in the corresponding location in the 3-D work space through a call to D3PONT.

Manipulator-Arm Imaging Routine - ARM

On the first call to this routine, the new arm image is placed in the buffer with D3LINE. Thereafter, D3DELL is first used to delete the old arm image before the new image is constructed.

4.3.5 Real-Time System Constraints. Average execution of the main control functions (excluding 3-D interface) takes about 60 ms and the loop is executed every 100 ms. As the technique of intermediate coordinate frames is used, spatial transformation calculations required approximately 30 milliseconds of computer time as compared with approximately 500 milliseconds for direct transformations. The Perceptronics computer system is programmed to output updated arm angle commands at 100 millisecond intervals. During the 100 millisecond interval, the computer reads the current arm position, joystick controller positions, and command pushbutton status, as well as calculates spatial transformations and performs overhead functions. With this 100 millisecond command interval, the manipulator end-effector can be moved under RMC mode in excess of 20 inches per second with no noticeable jerkiness and no overshoot.

The clock interrupts are generated every 4 ms and CPU control is transferred to TRAP1 until execution of the I/O module is finished and control is returned to the main module. Execution of an I/O interrupt takes about 1 ms. The I/O module is regarded as a critical section, so that any other interrupts are disabled.

With the use of 3-D graphics, the space-time trade-offs of the system resources become crucial. The rule-of-thumb for synchronizing 3-D interface is that each 3-D Z-frame is equal in length to the length of the display buffer minus 8 divided by 256. With the selector channel to the 3-D operating at the rate of one output per 5 microseconds, and the 3-D display operating at the rate of one mirror cycle every 33 milliseconds, the optimum buffer size would be about 6152 bytes in constant mode and 9224 bytes in variable-intensity mode (multiples of 256*2 bytes + 8, utilizing 15.36 milliseconds of the 16.15 milliseconds available on one forward sweep of the 3-D mirror). Each Z-frame would then last long enough for 12 points to be output.

The TTY module is invoked by the keyboard generated interrupts. Control is transferred to TRAP2 and the TTY module is executed while any other interrupts are disabled. An average execution time takes about 1 ms. In order to respond to coming interrupts with minimal delay, the main process must not disable CPU interrupts for long periods of time. This requires an even distribution of critical sections along the main process control flow so that an I/O or TTY interrupt will not get lost.

We can visualize the concurrent processes in the system in the following way. There is a basic cycle every 100 ms. For about 60 ms, the CPU works on the main control loop. Clock interrupts are coming every 4 ms., so that control is diverted to the I/O process. One TTY interrupt can occur during the cycle, and its time consumption is negligible. Overall in about 10-15 ms. of 100 ms. cycle, the CPU is free for other applications, such as interactive graphics or sensor processing.

5. Operational Procedures

5.1 Starting Sequence

At the Manipulator's Site:

(1) Turn on the hydraulic pump.
(2) Open the hydraulic valve on the manipulator.
(3) Turn on the video cameras.

At the Experimenting Site:

(1) Turn on the button for power, hydraulic and computer on the upper right hand side of the operator's keyboard.
(2) Turn on the two video displays.
(3) Turn on the feedback control display.

At the Computer Site:

(1) Turn on the +I/O.
(2) Turn on the main disc switch (colored red).
(3) Open the disc's case and put in the ONR79 + 3D disc.
(4) Make sure that both "write protect" are ON.
(5) Close the disc's case and press the RUN/STOP button.
(6) Turn on the ADDS terminal.
(7) Wait until the READY button is lit before proceeding with the rest of the sequence.
(8) Bring up the disc operating system, DOS, by the following sequence:

    (a) Set rotary switch to ADR/MRD.
    (b) Set panel switches to HX '02D0'

(c) Reset RUN and SGL switches and press EXC switch.

(d) Set rotary switch to PSW.

(e) Press INIT, RUN and EXEC switches.

(f) If DOS doesn't show on screen--repeat Step a through e--if it still doesn't work--get help.

(9) Activate the system load module file by typing: AC ONR79, 1C7.

(10) Load and start by typing: LO 1
ST 2C00

(11) Turn on power switch on 3-D.

(12) Turn off the disc drive by pressing the ON/OFF switch.

(13) Start experimentation.

5.2 Stopping Sequence

At the Computer:

(1) Reset all panel switches--so that the system stops.

(2) Make sure the disks are not running or else turn them off and wait for the SAFE light before proceeding on.

(3) Turn off the disc's case using the big red switch.

(4) Turn off computer, terminal and +I/O.

At the Manipulator Site:

(1) Turn off hydraulic valve and pump.

(2) Turn off the video cameras.

At the Experimenting Site

(1) Turn off the operator's keyboard.
(2) Turn off 3-D, video displays, and control display.

5.3 Report Generation Operating Sequence

This sequence should be executed immediately after experimental session sequence.

Starting Sequence

(1) Reset all panel switches so that system stops.
(2) Perform the usual stopping sequence at the manipulator and experimenting sites as shown in 5.2.
(3) Open the disc's case and put MOSHE disc instead of the ONR79 + 3D.
(4) Remove the "write protect" from the upper disc.
(5) Turn-on the disc driver.
(6) Turn-on the line printer.
(7) Wait until the READY button is lit and then proceed.
(8) Activate the system control file by typing AC RUNMON, 5C7.
(9) Transfer control to it by typing: TR 5.
(10) When report printer terminates perform the stopping sequence.

Stopping Sequence

(1) Turn off the discs by pressing the ON/OFF button.
(2) Turn off the computer, terminal and line printer.
(3) Wait until the SAFE light is on.
(4) Turn off the main disc switch.

APPENDIX C

LIST OF REPORTS AND PUBLICATIONS

Reports

Berson, B.L., Crooks, W.H., Shaket, E., and Weltman, G. Man-Machine Communication in Computer-Aided Manipulation. Perceptronics, Inc. (Woodland Hills, CA) Technical Report PATR-1034-77-3/1, March 1977.

Crooks, W.H., Shaket, E., and Alperovitch, Y. Man-Machine Communication in Computer-Aided Remote Manipulation. Perceptronics, Inc. (Woodland Hills, CA) Technical Report PATR-1034-78/3, March 1978.

Alperovitch, Y. ONR 77 Software Documentation. Perceptronics, Inc. (Woodland Hills, CA) Software Documentation PSD-1034-78-3/1, March 1978.

Crooks, W.H., Shaket, E., Chu, Y. and Alperovitch, Y. Man-Machine Communication in Computer-Aided Remote Manipulation. Perceptronics, Inc. (Woodland Hills, CA) Technical Report PATR-1034-79/3, March 1979.

Chu, Y., Crooks, W.H., and Shaket, E. Man-Machine Communication in Computer-Aided Remote Manipulation. Perceptronics, Inc. (Woodland Hills, CA) Technical Report PATR-1034-80/3, March 1980.

Conference Papers

Shaket, E. and Freedy, A. A Model of Man/Machine Communication in Computer Aided Manipulation. IEEE Conference on Cybernetics and Society, Washington, DC, September 19-21, 1977.

Freedy, A. and Shaket, E. A Model, Language Design and Experiments for Man/Machine Communication in Computer Aided Manipulation. Third CISM IFToMM Symposium on Theory and Practice of Robots and Manipulators, Udine, Italy, September 12-15, 1978.

Shaket, E., Freedy, A., and Crooks, W.H. Evaluation of a Command Language for Supervisory Control in Computer Aided Manipulation. International Conference on Cybernetics and Society 1978, Tokyo, Japan, November 3-5, 1978 and Kyoto, Japan, November 7, 1978.

Chu, Y. and Crooks, W.H. Man-machine Communication in Computer Aided Teleoperator Control Systems. 24th Annual Meeting of the Human Factors Society, Los Angeles, CA, October 1980.

DISTRIBUIION LIST

CDR Paul R. Chatelier
Office of the Deputy Under Secretary
of Defense
OUSDRE (E&LS)
Pentagon, Room 3D129
Washington, D.C. 20301

Director
Engineering Psychology Programs
Code 455
Office of Naval Research
800 North Quincy Street
Arlington, VA 22217 (5 cys)

Director
Undersea Technology
Code 220
Office of Naval Research
800 North Quincy Street
Arlington, VA 22217

Director
Information Systems Program
Code 437
Office of Naval Research
800 North Quincy Street
Arlington, VA 22217

Special Assistant for Marine
Corps Matters
Code 100M
Office of Naval Research
800 North Quincy Street
Arlington, VA 22217

Commanding Officer
ONR Western Regional Office
ATTN: Mr. R. Lawson
1030 East Green Street
Pasadena, CA 91106

Commanding Officer
ONR Western Regional Office
ATTN: Dr. E. Gloye
1030 East Green Street
Pasadena, CA 91106

Director
Naval Research Laboratory
Technical Information Division
Code 2627
Washington, D.C. 20375 (6 cys)

Dr. Robert G. Smith
Office of the Chief of Naval
Operations, OP987H
Personnel Logistics Plans
Washington, D.C. 20350

CDR G. Worthington
Office of the Chief of Naval
Operations, OP-372G
Washington, D.C. 20350

Dr. W. Mehuron
Office of the Chief of Naval
Operations, OP 987
Washington, D.C. 20350

Dr. Andreas B. Rechnitzer
Office of the Chief of Naval
Operations, OP 952F
Naval Oceanography Division
Washington, D.C. 20350

Dr. Jerry C. Lamb
Submarine Sonar Department
Code 325
Naval Underwater Systems Center
New London, CT 06320

Human Factors Department
Code N215
Naval Training Equipment Center
Orlando, FL 32813

Mr. Milon Essoglou
Naval Facilities Engineering Command
R&D Plans and Programs
Code 03T
Hoffman Building II
Alexandria, VA 22332

CDR Robert Biersner
Naval Medical R&D Command
Code 44
Naval Medical Center
Bethesda, MD 20014

Dr. Arthur Bachrach
Behavioral Sciences Department
Naval Medical Research Institute
Bethesda, MD 20014

CDR Thomas Berghage
Naval Health Research Center
San Diego, CA 92152

Dr. George Moeller
Human Factors Engineering Branch
Submarine Medical Research Lab
Naval Submarine Base
Groton, CT 06340

Dr. James McGrath, Code 311
Navy Personnel Research and
Development Center
San Diego, CA 92152

Mr. J. Williams
Department of Environmental
Sciences
U.S. Naval Academy
Annapolis, MD 21402

Dr. Gary Poock
Operations Research Department
Naval Postgraduate School
Monterey, CA 93940

Mr. H. Talkington
Ocean Engineering Department
Naval Ocean Systems Center
San Diego, CA 92152

Mr. Paul Heckman
Naval Ocean Systems Center
San Diego, CA 92152

Mr. Warren Lewis
Human Engineering Branch
Code 8231
Naval Ocean Systems Center
San Diego, CA 92152

Dr. Ross L. Pepper
Naval Ocean Systems Center
Hawaii Laboratory
P.O. Box 997
Kailua, HI 96734

Dr. A. L. Slafkosky
Scientific Advisor
Commandant of the Marine Corps
Code RD-1
Washington, D.C. 20380

Mr. Arnold Rubinstein
Naval Material Command
NAVMAT 08D22
Washington, D.C. 20360

Mr. Phillip Andrews
Naval Sea Systems Command
NAVSEA 0341
Washington, D.C. 20362

Mr. John Quirk
Naval Coastal Systems Laboratory
Code 712
Panama City, FL 32401

Mr. Merlin Malehorn
Office of the Chief of Naval
Operations (OP-115)
Washington, D.C. 20350

Director, Organizations and
Systems Research Laboratory
U.S. Army Research Institute
5001 Eisenhower Avenue
Alexandria, VA 22333

Technical Director
U.S. Army Human Engineering Labs
Aberdeen Proving Ground, MD 21005

U.S. Air Force Office of Scientific
Research
Life Sciences Directorate, NL
Bolling Air Force Base
Washington, D.C. 20332

Dr. Donald A. Topmiller
Chief, Systems Engineering Branch
Human Engineering Division
USAF AMRL/HES
Wright-Patterson AFB, OH 45433

Defense Technical Information Center
Cameron Station, Bldg. 5
Alexandria, VA 22314 (12 cys)

Dr. M. Montemerlo
Human Factors & Simulation
Technology, RTE-6
NASA HQS
Washington, D.C. 20546

Dr. T. B. Sheridan
Department of Mechanical Engineering
Massachusetts Institute of Technology
Cambridge, MA 02139

Dr. Harry Snyder
Department of Industrial Engineering
Virginia Polytechnic Institute and
State University
Blacksburg, VA 24061

Dr. W. S. Vaughan
Oceanautics, Inc.
422 6th Street
Annapolis, MD 21403

Dr. Robert Williges
Human Factors Laboratory
Virginia Polytechnical Institute
and State University
130 Whittemore Hall
Blacksburg, VA 24061

Dr. Alphonse Chapanis
Department of Psychology
The Johns Hopkins University
Charles and 34th Streets
Baltimore, MD 21218

Journal Supplement Abstract Service
American Psychological Association
1200 17th Street, N.W.
Washington, D.C. 20036 (3 cys)

Dr. Christopher Wickens
University of Illinois
Department of Psychology
Urbana, IL 61801

Mr. Glenn Spalding
Naval Material Command,
  MAT 07
Washington, DC 20360

Mr. Jim Katayama
Naval Ocean Systems Center
Hawaii Laboratory
P. O. Box 997
Kailua, Hawaii 96734

Dr. John K. Dixon
Computer Scientist
Naval Research Laboratory
Washington, DC 20375

MAJ Jack Thorpe
Cybernetics Technology Office
Advanced Research Projects Agency
1400 Wilson Blvd.
Arlington, VA 22209

Dr. A. K. Bejczy
Jet Propulsion Laboratory
California Institute of Technology
Pasadena, CA 91103

Dr. John Lyman
Biotechnology Laboratory
School of Engineering and Applied Sciences
University of California
Los Angeles, CA

Mr. Clifford Winget
Woods Hole Oceanographic Institute
Woods Hole, MD 02543

Dr. William Verplank
Xerox Business Systems
Systems Development Department
701 S. Aviation Blvd.
El Segundo, CA 90245

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER	2. GOVT ACCESSION NO. AD-A094482	3. RECIPIENT'S CATALOG NUMBER 9
4. TITLE (and Subtitle) 6 Man-Machine Communication in Remote Manipulation: Task-Oriented Supervisory Command Language(TOSC).		5. TYPE OF REPORT & PERIOD COVERED Final Technical Rep[redacted]t. 2 Feb [redacted]79-[redacted] 1 Feb [redacted]80
		6. PERFORMING ORG. REPORT NUMBER 14 PFTR-1034-80-3 ✓
7. AUTHOR(s) 10 Yee-Yeen/Chu William H./Crooks Amos/Freedy		8. CONTRACT OR GRANT NUMBER(s) 15 N00014-76-C-0603 ✓
9. PERFORMING ORGANIZATION NAME AND ADDRESS Perceptronics, Inc. ✓ 6271 Variel Avenue Woodland Hills, CA 91367		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS NR 196-140
11. CONTROLLING OFFICE NAME AND ADDRESS Office of Naval Research (Code 455) 800 N. Quincy Street Arlington, VA 22217		12. REPORT DATE 11 Mar[redacted]80
		13. NUMBER OF PAGES 232
14. MONITORING AGENCY NAME & ADDRESS(if different from Controlling Office) 12 225		15. SECURITY CLASS. (of this report) Unclassified
		15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

Unclassified - Distribution of this document is unlimited

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

Command Language	Man-Machine Communication
Computer-Aided Control	Remote Manipulator
Control Display	Supervisory Control
Hierarchical Planning	Task-Oriented Supervisory Control
Man-Computer Interface Design	Teleoperators

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

Computer-aided control of a general purpose teleoperator system provides a prime example of a new type of man-machine interaction in which the human operator must supervise and control a complex and often adaptive computerized system. Earlier studies showed that the communication between the operator and the computer-controlled system is an important determinant of the overall system performance. This report represents the results of a set of analytical and experimental studies focusing on the effectiveness of command and feedback

390285 Jw

20. ABSTRACT (CONTINUED)

modes for improved man-machine communication in supervisory control loop, where the operator may plan the tasks, command goal-directed actions, monitor task performance, and control when appropriate. The experimental results indicated that high-level command structure of computer-aiding functions can significantly improve task performance, especially in degraded environment. The study also suggested that feedback display of task status involving both machine state and environmental interaction is essential for successful manipulation.

This report describes (1) general teleoperator task functions, especially those required in underwater manipulation, (2) the theoretical analysis of man-machine communication based on the procedural nets model, and the development of the task-oriented supervisory command (TOSC) language, (3) a series of experimental investigations on the effect of command language features and feedback display on the ability of trained operators to perform selected remote manipulation tasks, (4) methods to evaluate overall teleoperator performance and a model for predicting performance as a function of command mode, intrinsic task characteristics, and environmental feedback, (5) a man-machine communication design methodology and guidelines for applications.

MED

8