https://ntrs.nasa.gov/search.jsp?R=19790007441 2018-03-25T04:23:40+00:00Z

D24

N79-15612

HUMAN/COMPUTER CONTROL OF UNDERSEA TELEOPERATORS+

T.B. Sheridan, W.L. Verplank and T.L. Brooks

Man-Machine Systems Laboratory
Department of Mechanical Engineering
Massachusetts Institute of Technology
Cambridge, Massachusetts 02139

Abstract

This paper discusses the potential of supervisory controlled teleoperators for accomplishment of manipulation and sensory tasks in deep ocean environments, and discusses one such system. Teleoperators and supervisory control are defined, the current problems of human divers are reviewed, and some assertions are made about why supervisory control has potential use to replace and extend human diver capabilities. The relative roles of man and computer and the variables involved in man-computer interaction are next discussed. Finally, a detailed description of a supervisory controlled teleoperator system, SUPERMAN, is presented.

1. Teleoperators and Supervisory Control

Many future undersea tasks may be accomplished by "teleoperators". We define teleoperators to be general purpose submersible work vehicles controlled remotely by human operators and with video and/or other sensors, power and propulsive actuators for mobility, with mechanical hands and arms for manipulation and possibly a computer for a limited degree of control autonomy. A manned submersible is not a teleoperator vehicle, but its attached manipulators are certainly teleoperators, requiring control through a viewing port or through closed-circuit video. Sometimes the term "teleoperator" is restricted to telemanipulator, excluding the system for remotely positioning and orienting a sensor, but for the sake of generality we include this important function.

This paper focuses on those aspects of undersea teleoperation which concern the human operator and the man-machine interface, and within this still relatively broad domain, it concentrates on the prospects for utilization of "supervisory control". Supervisory control is a hierarchical control scheme whereby a system (which could be a teleoperator, but could also be an aircraft, power plant, etc.) having sensors, actuators and a computer, and capable of autonomous decision-making and control over short periods and in restricted conditions, is remotely monitored and intermittently operated directly or reprogrammed by a person.

The distinction between direct human control of a teleoperator and supervisory control of a teleoperator is made graphically in Figure 1. In

---

+ This work was supported by Office of Naval Research (Contract N00014-77-C-0256) and by the MIT Office of Sea Grant (National Oceanographic and Atmospheric Administration).

342
PAGE INTENTIONALLY BLANK

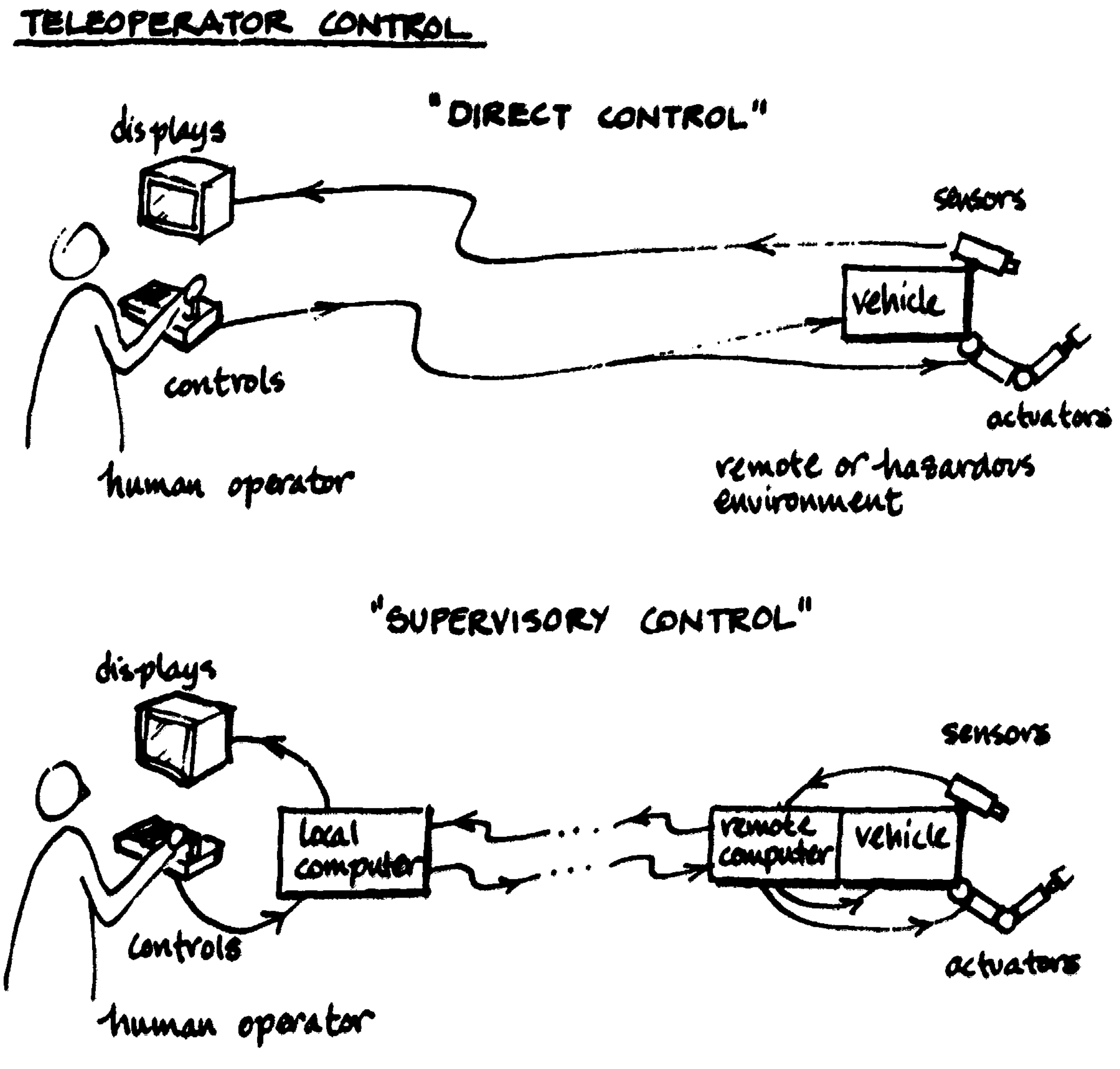

Figure 1. Definitions of Teleoperator and Supervisory Control

ORIGINAL PAGE IS
OF POOR QUALITY

Definitions

TELEOPERATOR

A vehicle having sensors and actuators for mobility and/or manipulation controlled by a human operator, and thus enabling him to extend himself to physically remote or hazardous environments.

SUPERVISORY CONTROL

A heirarchical control scheme whereby a device having sensors, actuators and a computer, and capable of automomous decision making and control over short periods and restricted conditions is remotely monitored and intermittently operated directly or reprogrammed by a person.

the upper figure the human directly controls, over either a wire or sonic communication link, the separate propulsive actuators of the vehicle, the actuators for the separate degrees of freedom of the manipulator, and the pan and tilt actuators of the video camera. The video picture is sent back directly to the operator. The "hand control" can be a master-slave positioning replica or a rate joystick.

In the lower figure a computer is added to the teleoperator, and for short periods and limited circumstances this teleoperator may function autonomously.

At the bottom are generic definitions of teleoperator and supervisory control. The upper drawing portrays the former without the latter. The lower drawing is the combination.

In supervisory control the teleoperator's (remote) computer communicates at high bit-rate with the teleoperator's sensors and actuators. But because of bandwidth constraints on the signal transmission link, or because of teleoperator sensing limitations, communication may be restricted to low-bit-rate with the human operator's (local) computer. For this reason, and also because of the intermittent nature of human monitoring and reprogramming of commands on a keyboard (and possibly joystick or other controls), the human supervisor's communication with the teleoperator tends to be at a slow rate, i.e., intermittent symbol strings or movement sequences on a master-controller with relatively many bits per instruction package. His communication with the local computer to refresh TV images or to edit or "dry run" his commands on a model before committing them to action may be constrained only by his own speed limitations.

The physical separation of local and remote computer is not necessary in aircraft, industrial plants or other systems where the operator is physically nearby, and where supervisory control is used for reasons other than physical remoteness and limited communication channel capacity between human operator and the object of control. In such situations supervisory control may be advantageous, nevertheless, to achieve faster or more accurate control, or to control simultaneously in more degrees-of-freedom than the operator can achieve by direct servo-control, or to relieve him of tedium. The latter reasons for supervisory control can apply to undersea vehicles when the human operator is not physically distant (as with manned submersibles) or to undersea teleoperators when a reliable high-bandwidth communication channel (wire or optical tether) is available.

2. Why Teleoperators Underseas? The Limits of Divers and Manned Submersibles.

The principal reasons for interest in using teleoperators for underseas tasks are dollar costs and safety.

Operations, including exploration, inspection, construction, maintenance, salvage and rescue, are having to be performed at increasing depths. At such depths - below, say 300m. (depending upon the paticular task) the time required for divers - mostly compression/decompression time - becomes excessive; factors having to do with depth per se, including life support equipment, become increasingly costly; personal safety is more and more difficult

ORIGINAL PAGE IS
OF POOR QUALITY

to maintain. These assertions are borne out by rather alarming mortality figures for commercial divers in the North Sea.

Water turbidity and other depth-related factors may require greater bottom-time, thus compounding the decompression-time factor. Under such conditions, a fixed-capability teleoperator, which sometimes is seen as too clumsy by comparison to a human diver at shallower depths, becomes much more attractive economically.

Happily, there is progress in the development of teleoperators, and they are becoming less clumsy. Inspection and manipulation tasks which simply could not be accomplished a few years ago are now achievable, due to steady progress in the design of video systems, mechanical valves and actuators, etc. For the immediate future, however, the primary technological factor which is changing the prospects for undersea teleoperation is the computer.

Circa 1970 divers seemed to have the edge on manned work-vehicles with manipulators in terms of maneuverability, manipulation, tactile sensing, and covertness. Because of smaller unmanned vehicles and eventually through unmanned untethered vehicles, however, the diver (especially the tethered diver) is losing his edge. Manipulation, sensing and cognition remain the primary advantages for the diver, but the computer is changing these also.

The comparison between teleoperators and manned submersibles is more clear-cut. The fact is that television cameras can now "see" with less light than the human eye, and new sonic imaging systems can see through densely turbid waters where neither human vision nor video can function. Spatial resolution or video can be made to approximate that of the eye by focusing. Present advantages of manned submersibles or teleoperators as work vehicles (neglecting for the moment personnel rescue) are: steropsis for close-up objects, and the ability of a human observer with a wide angle of view to keep track of the relative location of different objects. As the communication channel improves, to the point where the manipulator itself is the limiting factor, a man in a submersible can control manipulators or video pan-tilt controls just as well as a man on the surface. The major difference remaining between manned submersible and teleoperator are then cost and safety, as with the diver. The pressure vessel and life-support equipment make the manned submersible much more costly than the same vehicle without the pressure vessel and life-support equipment but with remote control instead. The factors of quality and reliability of communication and remote control then become the key factors.

3. Why Supervisory Control of Teleoperators Underseas? Some Assertions about the Problem.

a. Demands are increasingly stringent in terms of depth, sensory resolution, speed and accuracy of power of response for accomplishment of undersea tasks. Some of these tasks are always the same and are amenable to fixed automation, but many are different each time they occur and therefore cannot be done by fixed automation.

b. In terms of depth and skill human divers are reaching their limits, or when they go beyond these limits they do so at significant risk to life and cost in support equipment and personnel.

c. Teleoperators, i.e.,submersibles having video and other sensors, actuators for mobility and manipulation, and remotely controlled by human operators, offer much promise for extending man's flexible, adaptable, perceiving and control capabilities into remote and hazardous environments.

d. Present teleoperators are quite limited in sensory capability (e.g.,in turbid water), in manipulation capability (in speed and dexterity as compared to human hands), and in dealing with distortion in man-machine communication (misorientation of teleoperator to human body, time delays and noise).

e. Computers are rapidly getting smaller in size and power requirement and cheaper in cost for a given computing capability.

f. While accomplishment of one-of-a-kind undersea tasks by intelligent and completely autonomous robots may have appeal, we simply do not have available at this time such devices or the understanding to build such devices.

g. Undersea systems, like aerospace systems, demand conservative design because unreliability poses severe costs.

h. The most immediate and reliable approach would appear to be to add modest computer aiding and "artificial intelligence" to teleoperators, retaining human sensing, motor, memory and decision capability, at least for higher level planning, decision-making, and control.

i. Over a longer period of years, as computer control and artificial intelligence become more sophisticated, certain human functions in teleoperation may be replaced, but greater need and demand will be placed upon other human functions, and in these respects the need for improved man-computer interaction will increase, not diminish.

4. Relative Roles of Man and Computer, and Man-Computor Communication.

In analyzing the relationship between human operator and computor in teleoperation, it is useful to consider how human behavioral components, through two basic forms of communication are used in four human supervisory roles. Figure 2 summarizes the situation by arrows indicating causality between descriptors.

Commanding the computor is done by either typing strings of symbols or pushing dedicated buttons or switches (symbolic commands) or moving a joystick or replica controller, where there is a geometric isomorphism between control movement and its meaning (analogic commands). Observing can also be of symbolic displays (alphanumerics) or analogic displays (pictures or geometric diagrams). Imagining(internal mental visualization) may also be symbolic or analogic.

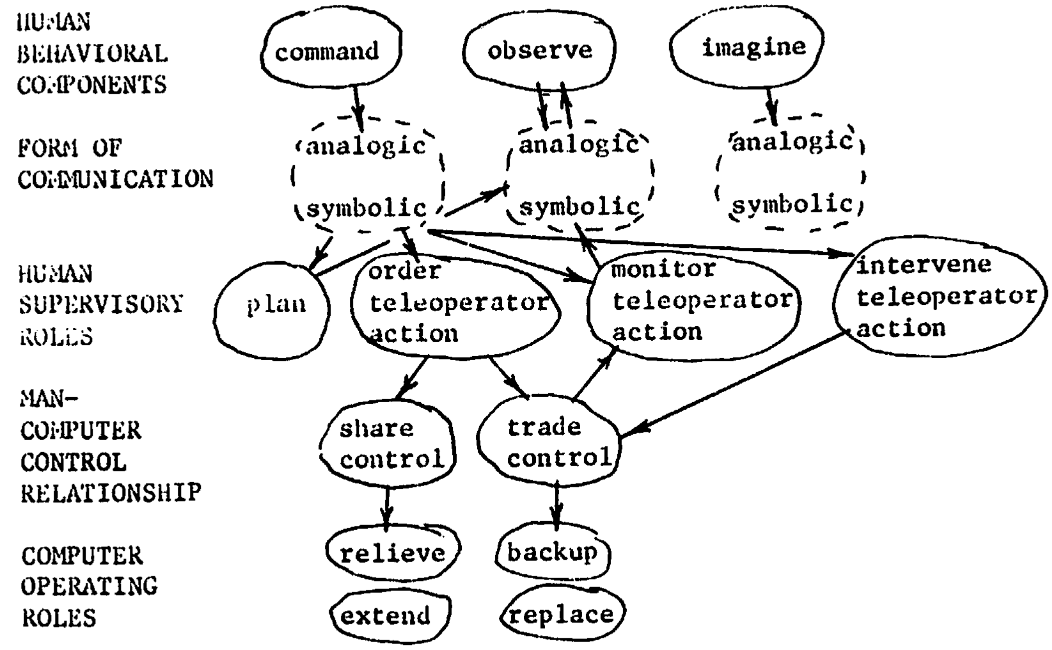

Figure 2. Relative Roles of Man and Computer

ORIGINAL PAGE IS
OF POOR QUALITY

Supervisory commands may be for purposes of planning information, such as referencing computer memory or testing a potential future action on a model; they may be for ordering teleoperator action; they may be for making computer adjustments while monitoring teleoperator action; or they may be for intervening in teleoperator action to assume direct manual control. Observation of displays is indicated in both the planning and the monitoring role of the operator.

As shown in the fourth row, the teleoperator may be ordered to act in two different ways. One is where the man shares control with the computer, i.e. the two work on the same task at the same time. The other is where the man trades control on all or some of the tasks, i.e. for those tasks he gives over complete control to the computer. Intervention means that the computer trades control back to the man. When control has been traded to the computer, part of the act of monitoring is to observe (analogic and symbolic) displays of its performance, as shown by the upward arrows.

The computer's alternative operating roles when sharing control with the human operator are relieve, i.e. do things which make his work easier, and extend, i.e. pushing his performance beyond where it would normally be. When control is traded to it the computer may backup the operator by being ready to take over in case he fails, or it may replace him altogether.

While this taxonomy of relationships at the present has no corresponding quantitative theory, it has been useful to the authors in thinking about what is desirable for man-computer control of teleoperators. In particular it has helped us think through the various forms of computer aiding which might be programmed into an experimental system. And it has clarified for us the potential of using a combination of general purpose typewriter and dedicated on special-purpose keyboard commands (symbolic)and force-reflecting master-slave and rate commands (analogic).

5. SUPERMAN: A System for Supervisory Manipulation

A brief description of a thesis by T.L. Brooks in progress at the Man-Machine Systems Lab at MIT is given on the following pages as an example of a supervisory manipulator system. This system is called SUPERMAN. Figure 3 shows the general relationships between the multiple inputs (keyboard, dedicated symbolic keys, and analog inputs), the computer states (STANDBY, DEFINE, EDIT, EXECUTE, AND TAKEOVER) and the control modes (RATE, MIXED MASTER/SLAVE AND RATE, MASTER/SLAVE, and COMPUTER control).

STANDBY State - When the computer is in this state, control resides with the main program and the operator. By pressing the proper button on the control console, the user can enter a particular manual control mode or another computer state (see Figure 4).

Manual Control Mode - A manual control mode is the method through which the user analogically interacts with the arms. A control mode is independent of the state. For example, the control mode might be MASTER/SLAVE while the state is EDIT. There are three kinds of modes:

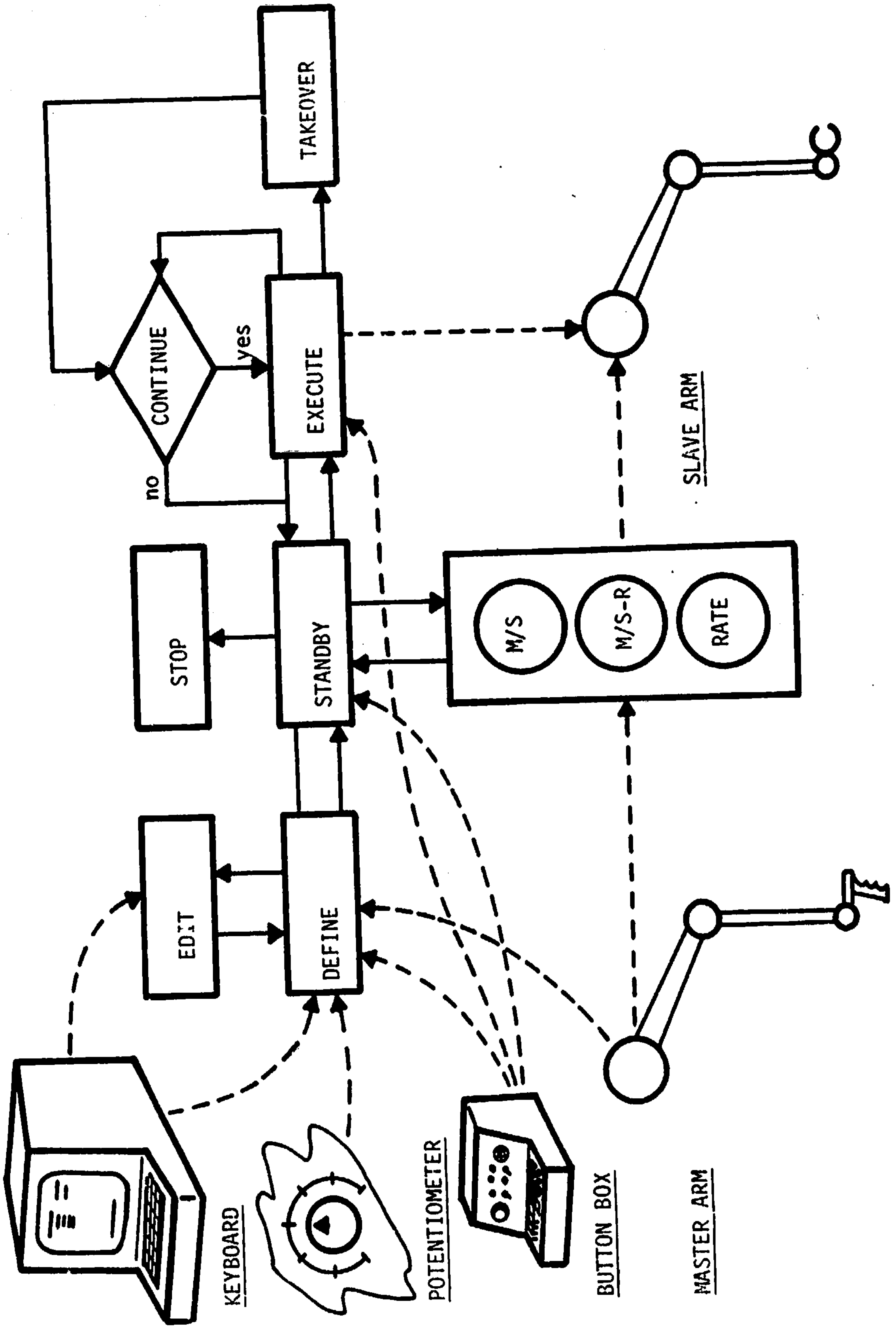

Figure 3. Block Diagram of SUPERMAN System

ORIGINAL PAGE IS
OF POOR QUALITY

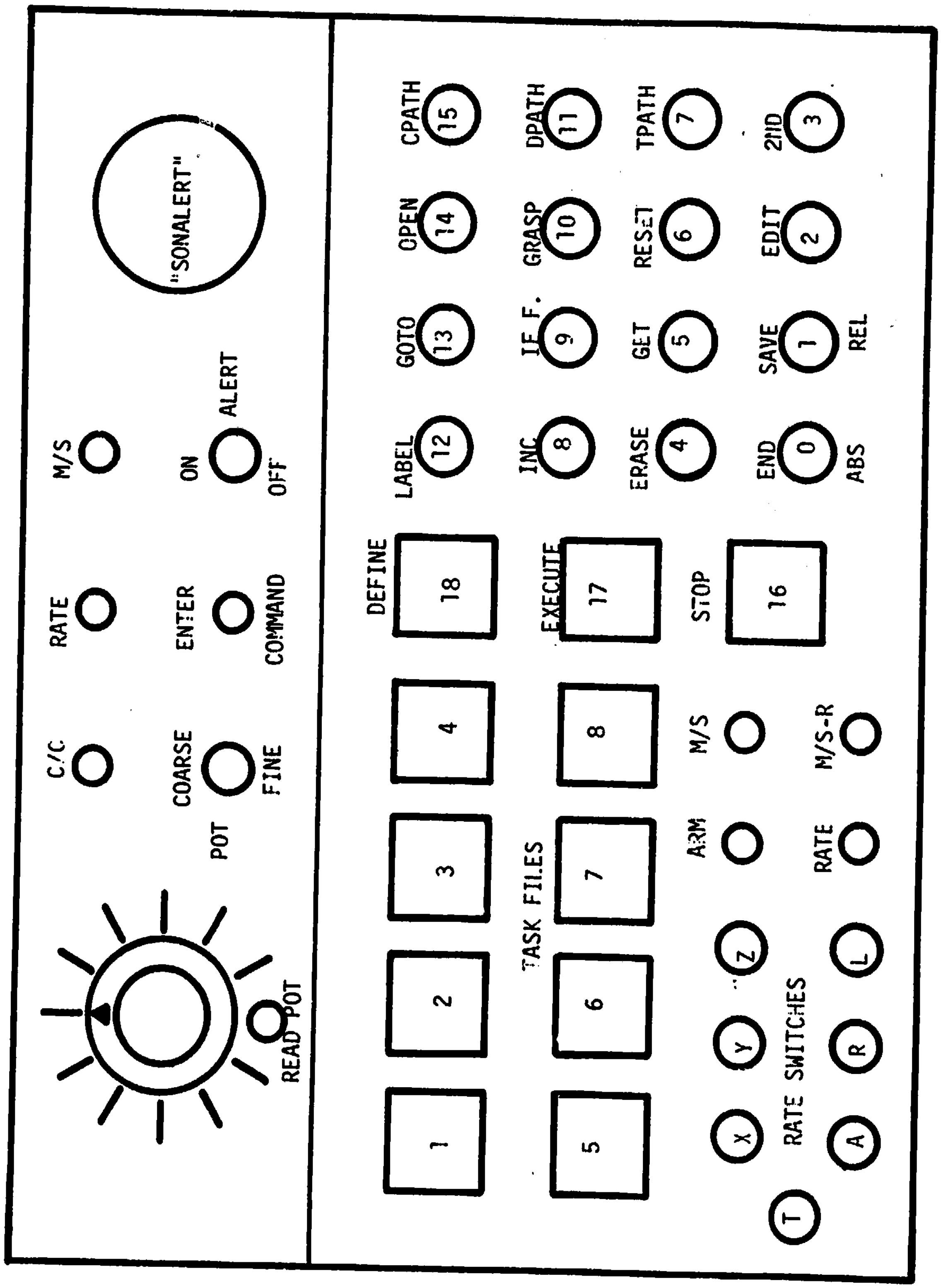

Figure 4. Special Control Console for SUPERMAN System

a) RATE - The individual degrees of freedom are controlled through rate commands by switches on the control console and a potentiometer for rate adjustment. Both rate and resolved-motion rate are available.

b) MIXED MASTER/SLAVE AND RATE - The master acts as a spring-loaded joystick in the X, Y and Z axes, giving rate commands to the X, Y and Z axes of the slave proportional to displacement of the master. (The rate of the slave arm is then reflected in the force feedback level which the operator feels in the master.) Both rate and resolved motion rate control are available. The remaining degrees of freedom, the left and right elevation, the azimuth and the end-effector are controlled in a master/slave mode.

c) MASTER/SLAVE - The slave arm is driven to duplicate in position the action of the master. Any force felt by the slave is reflected to the master giving the operator force feedback (i.e. proportional to position disparity between master and slave).

DEFINE - DEFINE is the primary state through which the operator enters a string of commands to be executed. Commands are entered by pressing specially dedicated buttons for each function. All of the buttons used in the DEFINE state have dual functions (see Figure 4 - dual function buttons are 0-15).

EXECUTE State - As the title implies, the string of commands is executed through this state. During the execution of the command register, if the operator desires to take control, there are two methods available. The operator can take immediate control: (1) by pulling on the appropriate control stick (i.e. the MASTER in the case of MASTER/SLAVE or MIXED MASTER/SLAVE AND RATE modes or the rate switches in the RATE mode), or (2) by pressing the STOP button (all action ceases after the STOP button has been pressed until the operator signals for continuation or return to STANDBY). The operator can execute a string of commands which have been saved as a task file by pressing one of the lighted TASK FILE buttons. The operator also has the option of executing the current command register by pressing the EXECUTE button. This allows the operator to define a string of commands and immediately execute them to determine if any modifications are necessary. After the operator is sure the command string performs the desired function correctly, that function can then be saved as a task file or a named file.

TAKEOVER State - TAKEOVER is a transition state between control modes, i.e. from computer control to the control mode in effect before the EXECUTE command. Special problems result during this state due to the mismatch between the master and the slave at the time of the takeover. The diamond in Figure 3 signifies that after the mismatch has been dissolved, the operator has the option of moving into the STANDBY state or continuing the EXECUTION state.

The detailed meanings of the DEFINE buttons 0-15 are given below:

BUTTON Number	Command
0	END Final command used to signal completion of DEFINE state.
1	SAVE Used to save the command register on the disk as either a task file or a named file. A task file can be recalled only by one of eight buttons in the STANDBY state, whereas a named file is saved under a user-designated title and can only be recalled by the same name through the GET button (6) in the DEFINE state.
2	EDIT The EDIT command allows the user to modify the command register. The following options are available through the keyboard after entering the EDIT state; a) CHANGE A LINE b) INSERT A LINE c) DUPLICATE A LINE d) DELETE A LINE e) LIST COMMAND REGISTER f) RETURN TO DEFINE
3	2ND Used to enter the second function of dual command keys. The first function of each key is printed in black letters above the button. The second function is written below the button in gold letters. To enter a second function command, press the 2ND key and then the desired second command.
4	ERASE LAST LINE [ERASE] Used to erase the last entry in the command register.
5	GET Used to retrieve a named command file from the disk. GET asks for the name of the command file to be recalled and then locates the file, reads it into the command register (and returns to DEFINE state).
6	RESET Used to initialize the necessary internal variables and the command register to zero.

7 THROUGH PATH [TPATH]

Records the present position of the arm for use in EXECUTE as a through point. (A through point is a position which the operator desires the arm to move through without stopping, i.e. non-zero velocity point.)

8 INCREMENT (DOF) (XXXX)

Makes an incremental motion in the desired degree of freedom by a selected value. The user enters the INCREMENT command, then the degree of freedom (DOF), adjusts the desired increment (XXXX) through the potentiometer and presses the READ POT VALUE button directly beneath the potentiometer.

9 IF (DOF) FORCE.GT.
EXECUTE NEXT COMMAND

If the force level in the desired degree of freedom (DOF) is greater than the level set by the operator (XXXX) the following command is executed. If the force level is less than the level set by the operator, the command immediately following the IF FORCE.GT. statement is skipped during execution. The user enters the IF FORCE.GT. command, then the desired degree of freedom, adjusts the force level through the potentiometer.

10 GRASP WITH FORCE (XXXX) [GRASP]

The user enters the GRASP command and adjusts the force level through the potentiometer.

11 DISCRETE PATH [DPATH]

Records the present position of the arm for use in EXECUTE as a stopping point. During execution, the slave arm is moved from its current position to the recorded position with zero final velocity.

12 LABEL (XXXX)

Labels a position in the command register which can be returned to through a GOTO command. The user presses the LABEL button and then the number (XXXX) of the desired label.

13 GOTO (XXXX)

GOTO is a conditional command which moves to label (XXXX), unless the operator signals during execution to change the branch to (YYYY) by pressing a different button. To enter the command the operator presses the GOTO button and then the number (XXXX) of the label to which GOTO should branch.

14	OPEN		Open jaws.
15	CONTINUOUS PATH	[CPATH]	Records the position of the master manipulator every 0.1 second for use in EXECUTE. A continuous path is achieved by interpolating between the recorded positions.
2ND - 0	ABSOLUTE		Informs the execution compiler that the command register is to be executed exactly as recorded (see RELATIVE). The user enters the absolute command by pressing the 2ND button (#4) and then the ABSOLUTE button (#0).
2ND - 1	RELATIVE		Informs the execution compiler that the positions in the command register are to keep the same relative displacement with respect to each other, but are to be transformed so that the first position following the RELATIVE command corresponds to the position of the slave at the time of execution. A RELATIVE command can be cancelled by an ABSOLUTE command, with the result that only the positions between the RELATIVE and ABSOLUTE commands are transformed. The user presses the 2ND button (#4) and then the RELATIVE button (#1) to enter the command in the register.
2ND - 2 through 2ND - 15	not assigned.		

As an example program consider a string of commands to take a nut off of a bolt and put it in a box. This program can be broken down into two major sections, one removes the nut and the other places it in the box. Since the user would prefer one nut removal program to be used for all nuts regardless of the orientation of the nut, a RELATIVE command should obviously be the first command in the register (the RELATIVE command and all of the following commands are briefly described under DEFINE). The entire command register for the nut removal program would be as follows. The following general formats will be followed throughout this example:

[BUTTON PUSH]
(POT READINGS)
"KEYBOARD COMMANDS"
COMPUTER REPLIES.

1	[RELATIVE]	
2	[LABEL] [1]	
3	[DPATH]	Place the slave on a nut and record that position by pressing the DPATH button.
4	[GRASP](200)	
5	[DPATH]	Turn the end effector 180° and record the position.
6	[INCREMENT] [Y](300)	Increment the slave by 300 counts in the direction that would pull the nut off.
7.	[IF FORCE.GT.][Y] (100)	If the force is greater than 100 in the Y direction, the nut is still on the bolt, therefore execute the next command.
8	[GOTO] [2]	
9	[GOTO] [3]	If the force had been less than 100 in the Y direction, the nut is free and this command would be executed.
10	[LABEL] [2]	
11	[INCREMENT] [Y] (-300)	Return the arm to position before incrementing in #6.
12	[OPEN]	Release the nut.
13	[GOTO] [1]	Return to LABEL 1 and continue turning the nut.
14	[LABEL] [3]	End of the first part of task - nut is off.
	[SAVE] "NUT-OFF"	Save command register as the named file "NUT-OFF" (typed in at the keyboard).

The second part of the task requires the manipulator to place the nut in a box. The entire command register for the program to put the nut in the box would be as follows:

1.	[ABSOLUTE]	The box would always be in the same place.
2	[TPATH]	Move the slave to a position just over and above the outside edge of the box and record this position by pressing the TPATH button.
3	[DPATH]	Move the slave to a position over the center of the box and record the position.
4	[OPEN]	
5	[TPATH]	Enter same position as in #2 by duplicating line 2.

[SAVE] "NUT-IN-BOX"

At this point the operator could call either program and execute it. The NUT-OFF program would simply take the nut off and return control to the operator as soon as the nut was free. But the present status of each file (i.e., a named file) requires that the operator type in each name to obtain the file to execute it. If the operator performs the following commands the file will be saved as a task file which is immediately executed at the touch of a button:

[GET] "NUT-OFF"

[GET] "NUT-IN-BOX"

The computer will reply by stringing the two files together as one file. Then enter:

[SAVE] "TASK-FILE"

and press the button which will retrieve the file (e.g., button #1). To remove a nut and put it in the box the operator simply presses the same button, the execution compiler transforms the first half of the register relative to the position of the slave at the instant the button is pressed and then executes the program. After the nut is removed and placed in the box the slave returns to the operator's position and the computer relinquishes control.