

N 73-10143

39. A Computer Assisted Teleoperator Control Station With Tactile Feedback*

J. W. HILL AND J. C. BLISS

Stanford Research Institute

This report describes a computer-assisted teleoperator control system for making comparative performance evaluations. A local and remote control station, each with decision-making capability, communicate with each other through a simulated time delay. Supervisory control at three increasingly automatic levels is possible. The highest level of programmed control is facilitated through the ARM language which was developed to permit easily readable program manuscripts to be written and assembled into programs of motions by novice programmers. Experimental results show the advantage of this form of supervisory control with both direct and delayed (3 sec) manipulation tasks. In addition, two systems to measure and reproduce force distributions have been designed. One system reproduces contact on the external surfaces of the remote hand using 21 airjet simulators. Another system reproduces the shape of the contact between object and jaws using 288 piezoelectric (bimorph) stimulators. In the course of this work the Rancho Arm was upgraded through mechanical strengthening and the addition of a proportional control system.

INTRODUCTION

Evaluation of touch-sensing and feedback systems for a mechanical hand require their use in a wide variety of tasks and control situations. Since tactile feedback does not interfere with manipulation as force feedback does in the time-delay situation, one of the variables in these evaluations should be the amount of time delay. Operations with a time delay, however, are so slow and tedious that manipulations without some form of supervisory control are not representative of the state of the art. To fulfill these requirements for a flexible control system, a computer program was written that simulates different control conditions, time delays, and levels of supervisory control. The advantage of this simulation is the way parameters and control options can be included or modified through program (software) rather than equipment (hardware) changes.

* For a more complete presentation refer to Bliss, Hill, and Wilbur (ref. 1) and Hill and Bliss (ref. 2). This investigation was conducted under NASA contract NAS2-5409.

The arm-control program on our LINC-8 computer simulates both a local and a remote control station, each with decision-making capability. The two stations communicate with each other through a communications time delay that may range from milliseconds, seconds, even to minutes. Supervisory control at three increasingly automatic levels is possible using this simulation. The first level is purely manual: the operator moves the arm using the control brace. The second level is automatic: the operator requests that remote sensors be tested or remote actions be carried out by typing a two-character instruction. The third level is programmed control: the operator types in the name of a previously written program containing the list of instructions he wishes to be carried out. This simulation is described under the heading Arm-Control System, and an example of a program to unscrew a nut from a bolt follows. An evaluation experiment comparing manual operation with program-assisted manual operation is also given.

Touch feedback systems for both a manual and automatic operation are described in this report. One system consists of 21 force sensors

distributed on the outside surfaces of the mechanical hand. These sensors are connected one-to-one with airjet tactile stimulators mounted over corresponding areas of the operator's hand. The second system consists of two 6×24 arrays of sensors distributed on the faces of the tongs. These sensors are connected one-to-one with piezoelectric (bimorph) stimulator arrays positioned on the fleshy pads of the operator's index finger and thumb. These sensors and stimulators provide force distribution feedback instead of force feedback in the conventional sense; thus it is possible to use these touch-reproducing systems in the time-delay situation without interfering with manipulations as force feedback does.

ARM-CONTROL SYSTEM

A control system for carrying out tasks at a distant location with a mechanical arm consists of the following three basic elements:

- (1) A control station where the operator controls the motion of the arm by transmitting commands and by supervising the resulting action using various displays and feedback.
- (2) A remote-station that accepts the commands and uses them along with information from environment sensors to control the arm.
- (3) A communications link that limits information flow. The limitations may take the form of a time delay, a bandwidth limitation, a signal-to-noise ratio, maximum video frame rate, and so forth.

There are obviously many combinations of design options for each of these three basic elements, depending on the task to be carried out. Hot-cell manipulations are usually carried out with simple control and remote stations. Frequently the remote arm is servoed directly to a joystick. This requires that the human operator interpret a TV display, access the situation, and provide manual inputs to carry out the appropriate action. The human operator supplies all the intelligence in this case.

In communication systems with a time delay, such as those involved in exploration of the moon or the planets, direct control by a human operator becomes a very slow and laborious process. To overcome this problem, both control stations and remote stations of varying complexity have

been proposed. For example, the control station may include predictor displays, or the remote station may have computing power enabling it to utilize joint positions, force sensors, range finders, or TV cameras to aid the operator by carrying out semi-automatic or "supervisory" operation. Carrying this idea to its logical conclusion, artificial intelligence laboratories are studying computer stations capable of completely automatic control. In this case the operator communicates to the remote station only with sentence-like commands.

Our goal is to design a control scheme that optimizes performance in carrying out remote tasks by combining the best attributes of man and computer. Therefore, man's ability to interpret scenes, estimate distances, and project motion with a multicoordinate control brace, is combined with the computer's ability to save and accurately duplicate arm positions, remember sequences of motions, carry out tests based on arm position, and interpret touch sensors. Background material on such supervision control is given by Johnson and Corliss (ref. 3, pp. 69-74), and Corliss and Johnson (ref. 4, pp. 117-124).

In general, man interprets and directs overall or organizational aspects of a task and the computer-directed arm attends to detailed aspects. The following hypothetical example of picking up a block illustrates this point. First, the operator recognizes the block in the picture transmitted by a TV monitor. He directs the arm to its position using a control brace. Control is then transferred to a computer subroutine which, utilizing feedback from the touch sensors, directs the hand to grasp the block and center it in the jaws. Next, as the operator directs the arm to go to a previously stored position, another subroutine adjusts the jaw pressure so that the block will not be dropped.

In order to study manual control supplemented by such semi-automatic operations in a time-delay environment, we programmed a small computer (LINC-8 with 4K memory) to simulate the entire control system shown in figure 1. The program actually has separate subroutines for the control station and the remote station that interact with each other only through a subroutine that simulates the time delay. Real-time interactions between the two control sta-

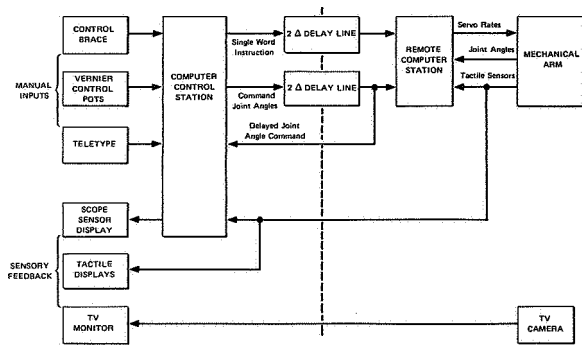


FIGURE 1.—Block diagram of the simulated man-machine control system.

tions are maintained by dividing time into 1/60-sec intervals. During each interval the inputs (if any) to the control station are serviced and a new command is formed for insertion into the delay line. Next, the command is inserted into the front of the delay line and the delayed command removed from the end. Finally, the delayed command is used by the remote-station subroutine to calculate new servo-rate outputs and to sample sensors or joint angles. Thus, the LINC-8 is programmed to share time between the two stations. Sampling and storing manual input signals at 60 Hz is adequate to preserve human accuracy and smoothness of motion.

Control Station

The arm-control station as it currently stands is shown schematically in the left half of figure 1. It consists of several manual inputs, several visual and tactile displays, and a computer program to permit the operator to select and transfer among inputs, displays, and programs of motion to accomplish a given task.

Manual inputs.—The manual inputs are illustrated in figure 2. The Rancho anthropomorphic control brace measures the joint angles of the operator's arm with a set of seven potentiometers. These joint angles can also be controlled with the individual potentiometers mounted on a panel. Manipulations in tasks requiring either a long time to complete or precise positioning are generally best carried out with knob control. Control can also be entered directly from a teletype using the format

#6←45

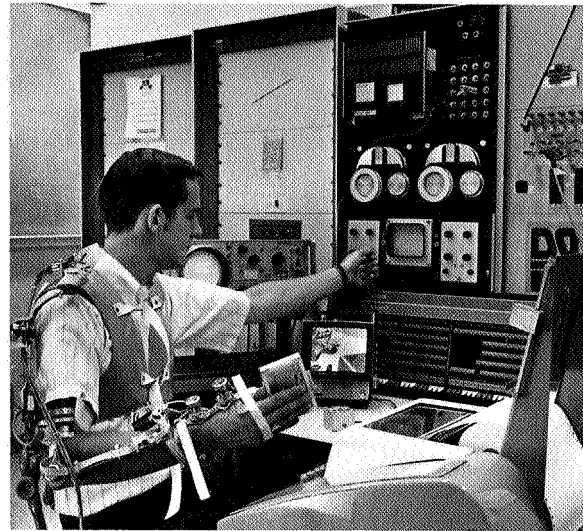


FIGURE 2.—Arm control station. (From left to right are (1) the master brace with single 4 X12 tactile display on index finger, (2) computer-driven display with sensing pad information, (3) TV view of hand grasping block, (4) operator adjusting vernier potentiometer on computer panel, and (5) teletype terminal.

where # and ← are prompts from the teletype, 6 is the joint number, and 45 is the joint angle in degrees. Teletype control has been used largely for testing and debugging manipulation programs. A desirable control input would be a miniature (scaled down) control brace light enough to maintain the position it is put in and small enough to be manipulated with the operator's fingers.

Sensory feedback.—Primary visual feedback is supplied by a broadcast-quality, closed-circuit television system. In addition, a computer-driven scope display presents the state of some of the touch-sensor information. One version of this display is shown in figure 3. Here information from three preliminary touch sensors and two simulated jaw sensors are presented in perspective.

Tactile feedback from the arm to the operator is provided by a set of touch sensors on the hand. The touch sensors fall into two groups: a pair of touch-sensing pads on the gripping surface of the manipulator tongs, and a number of individual force sensors covering the outer surface of the tongs and wrist. Since these sensors and the anthropomorphic tactile stimulators represent a

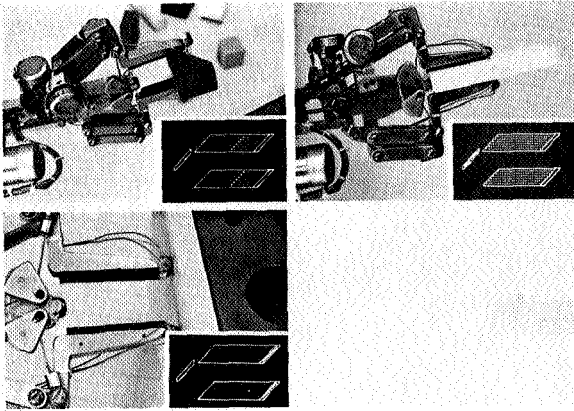


FIGURE 3.—Gripping and touching situations and corresponding displays on the computer scope. (The display program uses simulated data since the sensor multiplexer has not yet been constructed.)

major part of this project they are separately described later.

Control-station operation.—The control-station program is organized to provide three increasingly automatic levels of control. This control structure is described in the command tree shown in figure 4. By moving from branch to branch, typing single letters or numbers, the operator selects manual or program control to carry out the desired task.

Direct manual control is achieved by typing either *K* (for knobs), *B* (for brace), or *T* (for teletype), followed by either *A* (for absolute), or *R* (for relative). Absolute control causes joint positions to be read directly from these devices and to be transmitted to the remote station. Relative control specifies that joint commands from the control source take up where the previous joint commands left off. Thus, after a transfer from brace to knobs (or vice versa), the new control source picks up where the old one left off, and there is no transient motion artifact.

Decision-response control achieved by typing *D* permits two character instructions to be transmitted to the remote-control station. The first character of the pair selects a remote test to be carried out and the second selects a remote action to be carried out if the test is passed. For example, test *T* is "thumbsensor closed," and action *O* is "open jaws." Therefore, the instruction that appears as the sequence "*(T,O)*" means "if thumb sensor closed then open jaws." The in-

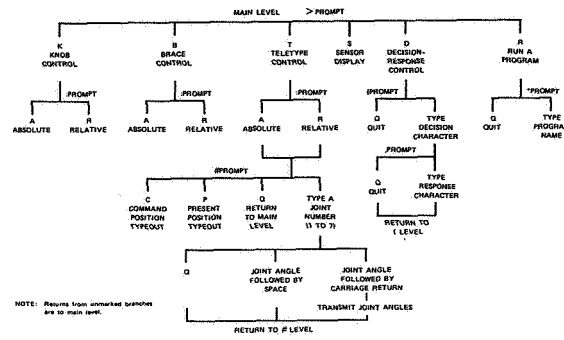


FIGURE 4.—Command tree of the control station.

struction "*(,O)*" requires no tests and causes the jaws to open. All of the *six* tests and all of the *seventeen* actions possible with the subroutines built into the remote computer are thus executable by typing in pairs of characters under decision-response control. Since any test can be used with any action, there are a total of $6 \times 17 = 102$ possible instruction that may be transmitted using this mode. A detailed listing of the different tests and actions is given later in figure 8(a).

Program control achieved by typing *R* and then the name of the desired program causes a program of tests and actions to be loaded into the remote computer's 256-word memory. Such programs are then run under decision-response control. Positions of objects are input to the program using the save of *\$* command in the form "*(,20\$)*." This command causes the seven joint angles to the arm to be stored beginning at location 20 in the program. When the operator is ready to enter the program, he types a go or *G* command as "*(,100G)*." This command causes further instructions to be taken from the list in the program starting at location 100. A usage example involving a program to unscrew a nut from a bolt is given later.

Remote Station

The remote station as it currently stands in the communication system is shown schematically in the right half of figure 1. It consists of a modified Rancho Arm* with a number of specially

* Model 8A, a 7-degree-of-freedom anthropomorphic manipulator manufactured by R. & D. Electronics, Downey, California.

built touch sensors, a TV camera, and a computer program with several subroutines for automatic operations. The physical layout is shown in figure 5. While the arm and its control system are described in the rest of this section, the touch sensors are separately described.

Mechanical arm.—During this study, the Rancho Arm was upgraded to reduce the amount of play in the joints and to increase the range of motion. In total, all joints but one have been refurbished to some degree, two members have been completely replaced, and two joints completely rebuilt. These changes were deemed necessary, based on our previous experience with the arm, in order to carry out meaningful manipulation experiments with it.

An initial study of the sources of play or looseness in the Rancho Arm revealed that poor design in the three worm-gear-driven joints was the major trouble. Replacing the bearings with commercial roller bearings and incorporating simple backlash adjusters greatly improved smoothness of performance. To lighten the arm, cable drivers for jaw closure and wrist prehension/extension were lengthened in order to mount the motors on the main pedestal. To increase the range of hand motion, thus increasing the number of meaningful tasks that could be carried out, wrist flexion/extension range was increased from 90° to 180°, and wrist rotation (supination/pronation) was increased from 90° to 360°. To

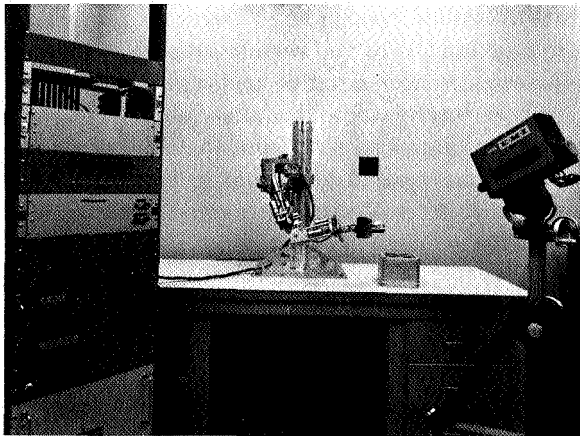


FIGURE 5.—Remote station. (From left to right are (1) the computer interface containing proportional control amplifiers, sensor amplifiers, and power supplies; (2) the modified Rancho Arm; and (3) the TV camera.

reduce the play between the tongs, the prehension linkages were rebuilt, replacing the machine-screw bearings with tightly fitting pin bearings.

Proportional arm controller.—Because of the many difficulties experienced with the original relay-operated bang-bang control system, a new proportional-control system was designed and built. This system has shown the following advantages over the original system:

(1) The time for a given movement can be halved by driving the motors harder than the original system but still retaining stability.

(2) Smaller movements are permissible.

(3) The smooth acceleration and deceleration reduces the mechanical coupling between joints and the vibrations at the beginning and termination of movements.

(4) Proportional control allows computer programs to govern rates of motion as well as position.

The proportional power amplifiers use a pulse-width-modulation drive to keep the power dissipated in the drive transistors low and also limit the drive current to prevent the motors from burning out. Since torque is proportional to motor current, this current limiting also provides a linear and easily adjustable analog to a mechanical clutch. Stiction effects are reduced by incorporating a "negative" dead band or high-grain region into the amplifier's otherwise linear characteristic. Stiction was additionally reduced by lowering the modulation frequency to the point where a slight vibration or mechanical dither is produced by the motors when the error signals are small.

Structure of the remote station.—The block diagram of the remote station is given in figure 6. The delayed instruction (two computer words) and the delayed joint commands (seven computer words) from the control station are the only inputs. The auto-manual switch is under program control and can be either closed to accept manual inputs from the control station or opened to allow commands generated by programs to move the arm. Arm control is quite conventional, with actual joint positions subtracted from the command joint positions, and the difference multiplied by the joint gains and output to the servo-motors as angular rates.

For program control, the following simulated

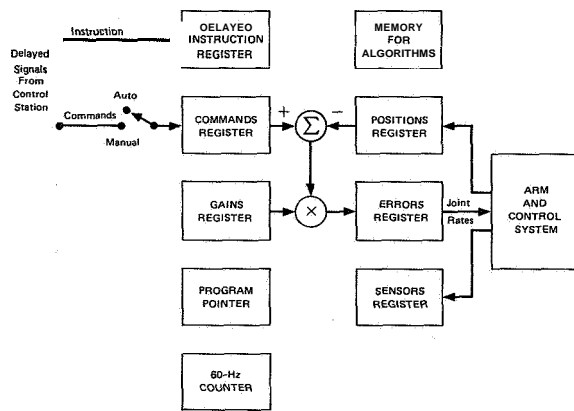


FIGURE 6.—Components of the remote station.

features have been added to the remote station:

(1) The counter, which is decreased by one count every $1/60$ sec. There is an instruction that sets the counter to any desired value and a test instruction that detects when it has reached zero. Time delays (or waits) in programs can be achieved by setting the counter to the desired value and waiting for it to become zero.

(2) The 256-word memory, which can be loaded with a list of instructions by a single command from the remote-control station.

(3) The program pointer, containing the address of (or pointing to) the next instruction to be executed. The address may be that of the delayed instruction register or may be any of the instructions in the 256-word memory. There are instructions that back up, skip, or specifically set the program pointer.

With the use of these registers and the memory in combination with the registers in the control system and the sensor information, meaningful tasks can be carried out.

Automatic Subroutines

To experiment with semi-automatic operation of the arm, subroutines that execute simple, remote tests and actions have been built into the remote-control station's computer program. These subroutines are the building blocks used by both single-instruction commands from the control teletype and multi-instruction algorithms executed by the remote computer. The test subroutines are based on both the arm-joint posi-

tions and the status of the touch sensors. The action subroutines change the contents of one of the arm-control registers to produce a desired movement or rate of movement, or to tell the program where to obtain the next instruction under program control.

A single instruction transmitted from the control station requests that a specific test be executed and that a specific command be carried out if the test was passed. The first half of the instruction word is used to select one of 64 possible tests by means of a look-up table. If the test is passed, the second half of the instruction is similarly used to select one of 64 possible actions. Even though only 6 tests and 17 actions have been implemented, a rich variety of operations is already possible. An example of a single instruction is as follows:

If "fingertip Sensor closed" then "open jaws."

For the computer, this instruction is the octal number 5157, where the "fingertip sensor closed" test is specified by 51 and the "open jaws" command is specified by 57. A list of the tests and actions is given later in figure 8.

One advantage of this system is the ability to converse with the arm in a language more natural than the machine language normally used to program small computers. Another main advantage in the time-delay situation is that the entire subroutines of machine-language instructions for the given test or action need not be transmitted to the remote computer. These subroutines are already built into the remote computer. Only a single instruction need be transmitted.

These instructions also allow short sequences of operations to be sent from the control station at one time instead of having to be sent one by one, waiting for a return message after each one. Thus the sequence of commands

- (1) If "fingertip sensor closed"
- (2) Then "stop arm"
- (3) And "open jaws"

allows a particular job to be done with one cycle of transmissions through the time delay that would ordinarily take three cycles. Additionally, with long time delays, this sequence of commands specifies a task that would require great caution if performed completely under manual control.

Just touching an object in such a time-delay situation is difficult without producing some overshoot that may knock the object away or without having to use a stop-and-wait strategy with successive motions of decreasing amplitude that may take a great deal of time.

The set of tests and commands is also intended to be used in longer lists (actual programs of movement), each with perhaps 10 to 100 instructions. These programs can provide such simple features as position memory or path memory, or can perform such complicated automatic tasks as unscrewing a nut from a bolt. A single command specifying that successive commands be taken from a list of commands can be a very powerful and flexible method for producing computer-assisted manipulations.

The structure of the remote-computer program executing these special instructions is given in figure 7. Combined with the short test and action subroutines and the control registers this subroutine is all that is required to implement the ARM language. The subroutine is entered every 1/60 sec after a new instruction and command position become available from the delay line, but before the servo subroutine (computing errors from arm positions) is carried out.

ALGORITHMIC LANGUAGE FOR REMOTE MANIPULATION (ARM)

If requests for the automatic operations described in the preceding section are taken from a list, the list can be considered a program of motions (an algorithm) to carry out a manipulation task. The effective utilization of such programs, however, requires a means of writing them in an easy-to-use language and a means for assembling (or generating) a list of arm operations from the statements in the language. Using a sufficiently powerful computer program, a list of such operations or instructions could be generated from any reasonable language. With a small computer system, care must be exercised when designing a programming language so that the assembly of programs is possible. Under this constraint, we simultaneously developed the separate concepts of the ARM language, the assembler, the instruction set, and the procedure for carrying out instructions.

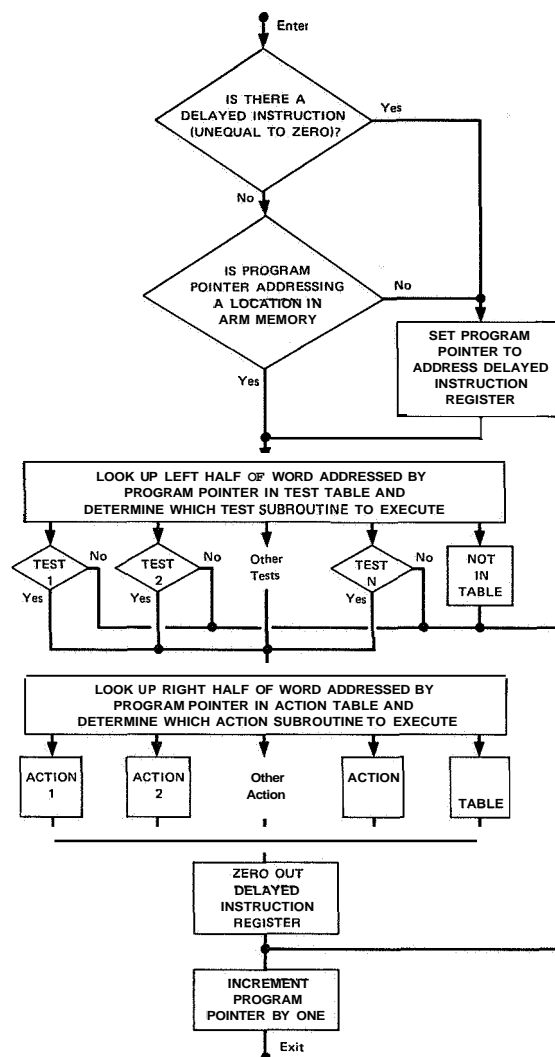


FIGURE 7.—Flow chart of the remote station subroutine. (This subroutine carries out single instructions and programs of instructions. It can be seen that if instructions are being taken from a program in arm memory, an instruction sent from the control station will cause the program to be stopped and the instruction to be carried out.)

ARM differs from the MHI or THI language developed by Ernst (ref. 5) and from MAN-TRAN reported by Sheridan (ref. 6), in that manual inputs from the operator can be used in addition to teletype inputs. Thus, the operator can move his control brace and request that the arm move to "this" position or move "this" joint "this" much. Each "this" in the preceding sentence is a manually specified quantity that is

difficult to verbalize, much less to quantify as a joint vector for typewriter input. ARM should not be considered an entity itself, but a means of carrying out automatically programmed motions within the scope of the teleoperator control system previously described.

An example of a program (written in ARM), an algorithm to unscrew a nut from a bolt, is given in figure 8. The program consists of the following three parts:

(1) Definitions of the symbols for the assembler (fig. 8(a)) to use in converting names to instructions.

(2) An algorithm for the task (fig. 8(b)), giving the order of individual motions and tests.

(3) Labeled storage space for arm positions and gains used in various parts of the algorithm (fig. 8(c)).

This entire manuscript is given to an assembler for conversion to a list of numbers (instructions) for execution by the remote computer program.* In this case the compiling is quite

* Note the meaning of some special symbols:

/ A comment.

→ Assign a value to a name.

; The line terminator.

: A label.

```

\BASIC DEFINITIONS FOR THE ASSEMBLER
\DEFINITIONS OF THE TESTS
    THUMB=6400    \THUMB TIP SENSOR CONTACT
    FINGER=5100  \FINGERTIP SENSOR CONTACT
    B0TH=4200    \BOTH THUMB & FINGER CONTACT
    WEBB=6100    \WEBB OF JAW SENSOR CONTACT
    DONE=4400    \ALL JOINT ERRORS < 17
    WAIT=6500    \CLOCK NOT YET = 0

\DEFINITIONS OF SINGLE WORD ACTIONS
    STOP=63      \LOAD PRESENT POSITION IN COMMAND VECTOR
    OPEN=57      \OPEN JAW
    CLOSE=43     \CLOSE JAW
    AUTO=50      \TAKE POSITION COMMANDS FROM PROGRAM
    MANUAL=62    \TAKE POSITION COMMANDS FROM CONTROL STATION
    REPEAT=42    \SUBTRACT 1 FROM PROGRAM COUNTER
    SKIP=53      \ADD 1 TO PROGRAM COUNTER
    SKIP2=22     \ADD 2 TO PROGRAM COUNTER
    BELL=18      \RING TELETYPE BELL
    DDT=44       \STOP PROGRAM & LOAD DEBUGGING PACKAGE

\DEFINITIONS OF TWO WORD ACTIONS WITH PARAMETER IN SECOND WORD
    G3=47        \SET PROGRAM COUNTER WITH PARAMETER
    CLOCK=67     \SET 60 HZ CLOCK COUNTER WITH PARAM.

\DEFINITIONS OF TWO WORD ACTIONS WITH VECTOR ADDRESS IN SECOND WORD
    INCREMENT=13 \ADD POSITION VECTOR TO COMMAND VECTOR
    DECREMENT=15 \SUBTRACT POS. VECTOR FROM CMD. VECTOR
    MOVE=55      \LOAD CMD. VECTOR FROM POS. VECTOR
    SAVE=4       \STORE CMD. VECTOR IN POS. VECTOR
    GAINS=17     \LOAD GAIN VECTOR FROM GIVEN VECTOR

\SOME FILL-WORDS, ONLY TO MAKE PROGRAMS READ EASIER
    BY=0
    IF=0
    THEN=0
    SET=0
    IN=0
    TO=0
  
```

(a) BASIC DEFINITIONS OF THE TESTS AND ACTIONS

```

\PROGRAM TO UNSCREW A NUT FROM A BOLT
\LABELS---INSTRUCTIONS-----COMMENTS
BEGIN
    AUTO
    CLOSE
    \SWITCH TO PROGRAM CONTROL
    \MAKE SURE JAWS CLOSE HARD

LOOP:  DECREMENT BY: HALFTURN
    SET CLOCK: 140
    IF WAIT THEN REPEAT
    \LEFT 180 DEGREE WRIST ROTATION

    SAVE IN: NUT
    \SAVE POSITION OF NUT

    SET GAINS: TESTGAIN
    MOVE TO: TEST
    CLBE
    SET CLOCK: 200
    IF WAIT THEN REPEAT
    \MOVE TO TEST POSITION

    SET GAINS: D0NEGAIN
    IF DONE THM SKIP2
    GO TO: FINISH
    \IS NUT FREE?

    SET GAINS: RETGAIN
    MOVE TO: NUT
    SET CLOCK: 200
    IF WAIT THEN REPEAT
    \RETURN TO NUT

    SET GAINS: FUGAIN
    OPEN
    SET CLOCK: 200
    IF WAIT THEN REPEAT
    \OPEN JAW

    INCRMMT BY: HALFTURN
    SET CLOCK: 140
    IF WAIT THEN REPEAT
    \RIGHT 180 DEGREE WRIST ROTATION

    CLBE
    SET CLOCK: 200
    IF WAIT THEN REPEAT
    \CLOSE JAW
    GO TO: LOOP

FINISH: SET GAINS: FULLGAIN
    CLBE
    MOVE TO: FINAL
    SET CLOCK: 300
    IF WAIT THEN REPEAT
    \MOVE TO FINAL POSITION

    OPEN
    SET CLOCK: 200
    IF WAIT THEN REPEAT
    \DROP NUT

    BRL
    GO TO: 1000
    \RING BELL=ALL DONE
    \END PROGRAM
  
```

(b) THE ALGORITHM

FIGURE 8.—Continued.

```

\POSITION AND GAIN VECTORS
\THE 1 NUMBERS CORRESPOND TO THE SEVEN JOINTS
\BEGINNING AT SHOULDER AND ENDING AT TONGS
HALFTURN: 01 01 01 01 -7601 01 01  \RIGHT HALF TURN
FINAL:    01 01 01 01 01 01 01  \FINAL POSITION
NUT:      01 01 01 01 01 01 01  \TEMPORARY POSITION
TEST:     01 01 01 01 01 01 01  \TEST POSITION
TESTGAIN: 20001 20001 20001 2000  \LOWER FORCE
          01 01 2000
RETGAIN:  31771 31711 31111 3171  \RETURN GAIN
          01 01 3111
D0NEGAIN: 01 31111 01 37711 01 01 0  \TEST TWO JOINTS
FULLGAIN: 37711 31711 37111 3711  \FULL GAINS
          31771 31111 3777

END
  
```

(c) THE STORAGE LOCATIONS

FIGURE 8.—Concluded.

straightforward: Values for the various symbols on each line are simply added together to form the instruction. This simple assembler, however, is quite powerful, as indicated by the easy-to-understand phrases of the algorithm shown in the program.

The unscrewing program is designed to receive both typed commands entered from the control-station teletype and manual commands entered from the control brace or knobs. Thus, the pro-

FIGURE 8.—Program to unscrew a nut from a bolt. (All of the numbers in this program are octal.)

gram of figure 8 is not completely self-explanatory. For the unscrewing task the operator carries out the following operations:

(1) Load the unscrewing program (a total of 116 instructions) into the remote computer using the run command "R*UNSCREW."

(2) Sets up Absolute control by the Brace by typing "B:A."

(3) Moves the arm to the final position where the nut is to be placed and stores the position vector in the "FINAL" storage area of the program under decision control using the command "(,104\$)." (The octal number 104 is the location of FINAL; hopefully a more refined version of the control-station program will allow name input).

(4) Moves the arm to the test position located 1 to 2 in. directly behind the nut and similarly saves the position vector in the "TEST" storage area using the command "(,122\$)."

(5) Moves the arm to the nut and grasps it using the brace and then starts the program using the command "(,G)."

Instructions taken from a list then direct the arm to unscrew the nut by half-turns, attempting to pull the nut to the test position after each half-turn is completed. If the test position is not reached, the arm knows that the nut is still attached and goes back for another half-turn. When the test position is reached, the arm moves to the final position, opens its jaws, and drops the nut into the container.

AN EVALUATION OF A COMPUTER-ASSISTED MANIPULATION TASK

To determine what advantage a program of motions would have in reducing the time required for a task, a task was designed around the unscrewing program previously described. The task was to unscrew a hex nut from a 16 mm (5/8-in.) machine bolt and drop it into a small 5-by-8 cm receptacle located approximately 40 cm away. The two different experimental conditions were as follows:

(1) The communications time delay was either zero or 2.6 sec (the round-trip moon-time delay).

(2) Control was either pure manual or computer-assisted manual. Using knob control for

all manual inputs and direct viewing, all four combinations of these two conditions were performed twice by one subject (JH). Knob instead of brace control was used because of the inability to hold the control brace sufficiently stationary for several minutes in the longer tasks and the inability of the human wrist to rotate through 360° for efficient unscrewing.

The computer-assisted manual-control conditions followed the modus operandi for running the unscrewing algorithm detailed in the preceding section: The "final" and "test" positions were found manually and entered into the program and then control was handed over to the computer algorithm, which finished the task. The program was changed only to allow the nut to be removed with full turns by doubling the wrist increment "Halfturn."

The manual removal of the nut followed a slightly different course. No test position was required in this case, so the task began by unscrewing the nut turn-by-turn for the first three of four revolutions. Since the nut would come off in five or six turns, the strategy changed at this point and an attempt was made to pull the nut off before releasing it. If it did not come off, the jaws were opened, the grasping position found again, and another revolution undertaken. When the nut came off, it was manually positioned over the receptacle and dropped.

The results of these tests are given in figure 9. In both time-delay situations, manual-computer control was faster than manual control. The difference is greatest in the time-delay condition where a time reduction of greater than 5 to 1 was achieved by employing the computer-assisted operation.

TOUCH-FEEDBACK SYSTEMS

Two touch-feedback systems for the teleoperator control system have been constructed. Each system consists of a set of sensors mounted on the mechanical arm and a corresponding set of tactile stimulators mounted on the control brace. All of the sensors utilize conducting rubber that is deformed to complete an electrical circuit upon physical contact. Individual sensors activate corresponding stimulators in a binary fashion: a stimulator is either full on or full off. The follow-

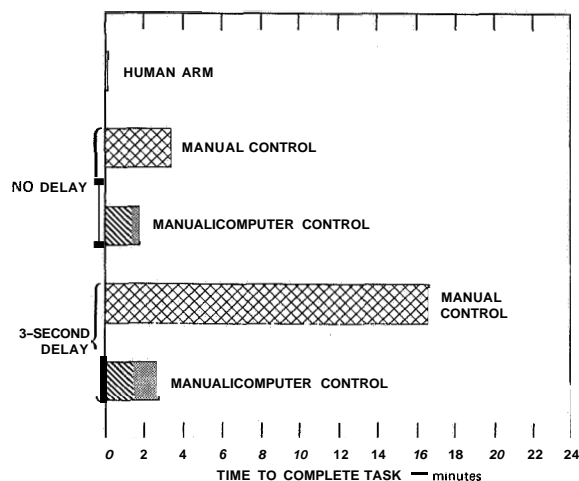


FIGURE 9.—Comparison between manual and computer-assisted manual control. (The time taken to perform the task directly with a human hand is shown only for comparison. Computer-assisted times are broken into two parts: the time required by the program which is the same in both delay conditions, and the time required by the manual portion.)

ing sections describe the construction of the sensors and the two touch feedback systems.

Sensor Design

Individual sensors for the hand and wrist areas of the manipulator must have the following properties, which are seldom found together in commercial microswitches:

- Respond to forces over a specified (often large) area.
- Respond to small forces over a range of directions.
- Be small in size.

Prototype sensors having some of these characteristics have been developed to transduce mechanical forces to electrical signals. The sensors, shown in figure 10, all use conducting rubber in some form and have the ability to provide both discrete (binary) and continuous force information. Descriptions of these sensors follow:

(1) The omnidirectional force sensor shown in figure 10(a) consists of a bridge of conducting rubber cast into an insulating mounting block. Sensitivity depends on both the thickness of the

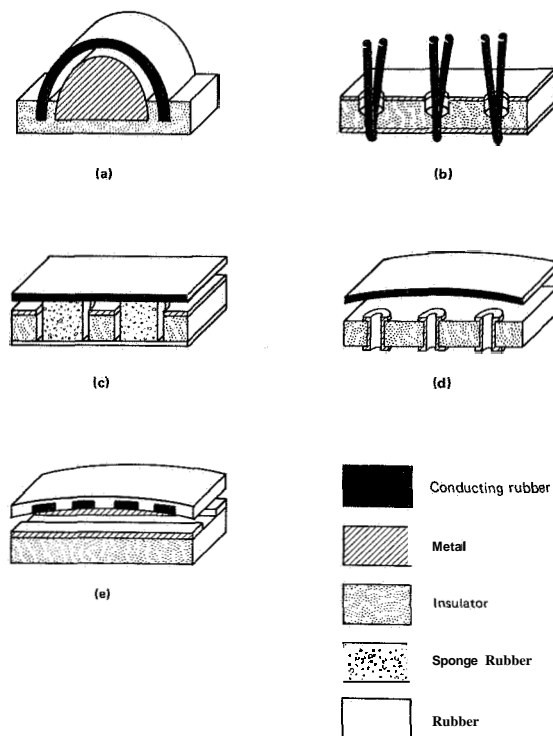


FIGURE 10.—Five types of sensor construction using conducting rubber.

conducting rubber and the spacing between the rubber and metal rod.

(2) The whisker sensor shown in figure 10(b) consists of thin conducting rubber strips pulled through holes in two-sided printed circuit boards. This sensor has high sensitivity because of the mechanical advantage of the whiskers and has a wide dynamic range (over 1000 to 1) if several parallel whiskers (10) are employed in the same sensor.

(3) The surface sensor shown in figure 10(c) consists of a conductive rubber sheet held by sponge rubber columns above a sheet of single-sided printed circuit board. If islands of copper foil are made by etching the circuit board, then the contact force can be localized to one or more islands.

(4) The force-distribution sensor shown in figure 10(d) consists of a sheet of conducting rubber arched over an insulating board studded with microeyelets. The shape of the force pattern is measured by measuring the pattern of contact

resistance between the eyelets and the conducting rubber sheet.

(5) The force distribution sensor shown in figure 10(e) consists of rows of conducting rubber imbedded in insulating rubber above perpendicular rows of foil on an insulating board.

Hand Contact-Sensing System

The purpose of this system is to reproduce to the operator the contact between the mechanical hand and the object being touched or manipulated. This system consists of a number of conducting rubber sensors mounted on the outside surfaces of the mechanical hand, as shown in figure 11. The outside of the tongs is completely

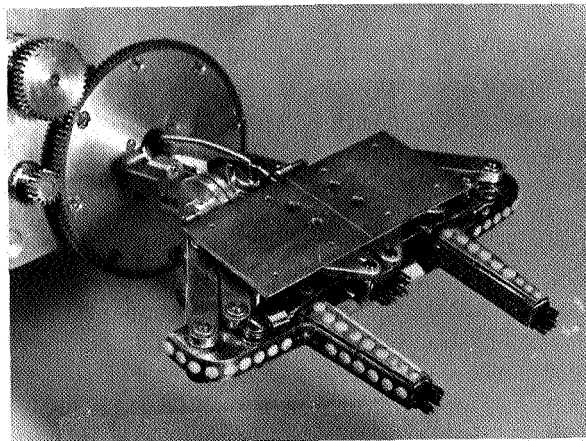
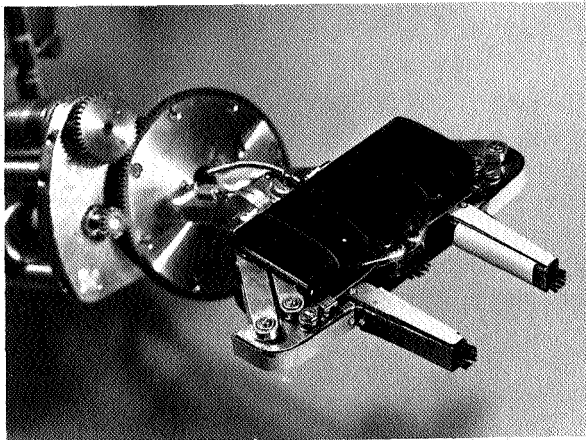


FIGURE 11.—Construction of touch sensors. (Upper photo shows printed circuit board bonded to the tongs and wrist. Lower photo shows finished assembly with conducting rubber covers in place.)

covered with these sensors, as are other extreme or protruding parts of the upper hand. The sensors are arranged so that any contact with the hand and a flat surface is sensed and that any contact with the tongs is sensed. The locations of these sensors on the remote hand are given in table 1. Each of the sensors is connected via

TABLE 1.—Contact Sensors on Mechanical Hand

| Location | Number | Type, fig. 10 |
|---------------------------|--------|------------------|
| Tips of tongs | 2 | b |
| Top of tongs, distal | 2 | c |
| Top of tongs, proximal | 2 | e |
| Bottom of tongs, distal | 2 | c |
| Bottom of tongs, proximal | 2 | c |
| Back of tongs, distal | 2 | e |
| Back of tongs, proximal | 2 | c |
| Web of jaw | 1 | b |
| Knuckles | 2 | c |
| Top of wrist | 2 | d |
| Bottom of wrist | 2 | d |

amplifying and gating circuits to an air-jet tactile stimulator. The air jets are positioned on the control brace to produce a touch sensation on a portion of the operator's hand corresponding to the location of the sensor. Each jet produces an area of pulsating pressure on the skin approximately $3/16$ in. in diameter. The arrangement of air-jet stimulators on the control brace is shown in figures 12 and 13. The construction of the air-jet stimulators is described by Bliss and Crane (ref. 7, appendix B).

Jaw Shape-Sensing System

The purpose of this system is to reproduce to the operator the shape and location of the object held in the remote jaws. Two sensing pads using the design shown in figure 10(d) are built into the tongs of the mechanical hand as shown in figure 11. Each of these two opposing pads consists of 144 individual contacts in a 6×24 rectangular pattern. Two corresponding 6×24 rectangular arrays of bimorphs contacting the index finger and thumb are built into the control brace as shown in figure 12.

The connections between the sensor and stimulator are one-to-one: If one contact is closed, the

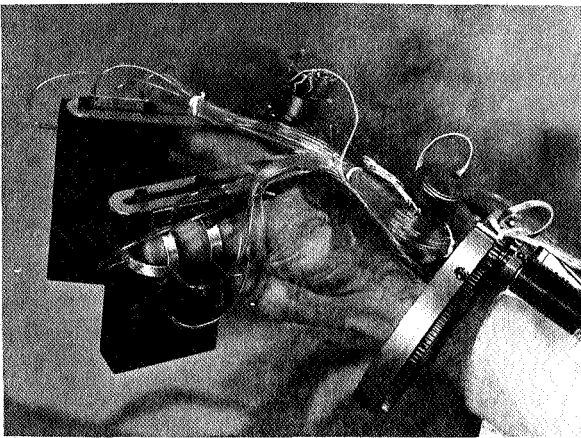
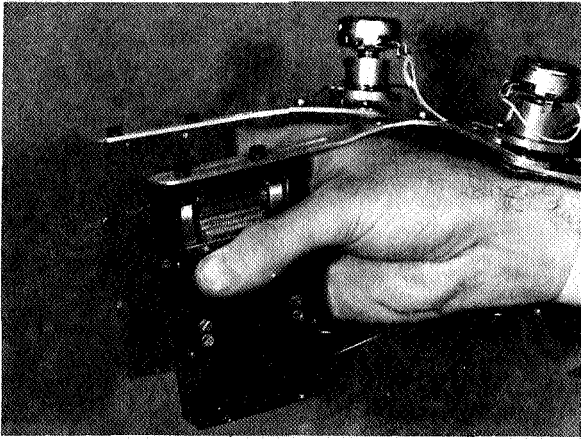


FIGURE 12.—Tactile simulators on the hand controller. (The upper photograph shows the array of 6X24 piezoelectric vibrators positioned under the thumb. The lower photo shows the same hand controller with seven air jets in place on each finger.)

bimorph in the corresponding location vibrates. Bimorphs produce a vibration of the skin restricted to an area about 1 mm in diameter. Thus the pattern of contact closures is reproduced as a pattern of vibration enabling the operator to feel on his own thumb and index finger the shape and location of the object held in the remote jaws. A complete description of nearly identical bimorph arrays used in shape recognition and reading experiments is given by Bliss (ref. 8) and Bliss, Katcher, Rogers, and Shepard (ref. 9).

This shape-sensing system represents a considerable improvement over our previous jaw shape-sensing system (Bliss, Hill, and Wilber, ref. 1). The new sensors are exactly the size of

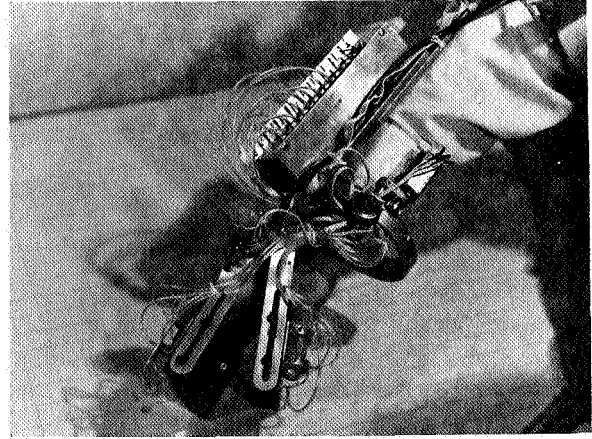


FIGURE 13.—Overall view of the tactile stimulators on the control brace. (Air valves in the box on forearm individually activate air jets on the finger and thumb.)

the jaws (10 by 50 mm) and are only about 3 mm thick, whereas the previous sensors were twice as wide and twice as thick. In spite of their clumsiness, the previous sensors were necessary to carry out obscured manipulations and they greatly reduced the occurrence of drops and fumbles in pickup and extraction tests.

ACKNOWLEDGMENTS

The authors are grateful to those who have substantially contributed to this project. Professor James L. Adams of Stanford University assisted in diagnosing problems with the Rancho Arm and suggesting improvements for it, as well as a philosophy for designing a teleoperator control system, and evaluation experiments to be carried out with the system. Mr. Anthony J. Sword of Stanford University helped set up the initial control-system hardware, suggested several of the features in the control-station simulation, wrote the first two versions of the control-system computer program, and designed and constructed the first touch sensors. His work was supported under grant NASA-NgR-05-020-345 with Stanford University. Mr. John P. Gill of Stanford Research Institute contributed to the design and construction of the piezoelectric bimorph tactile stimulators and built the majority of both touch-sensing systems.

REFERENCES

1. BLISS, J. C.; HILL, J. W.; AND WILBER, B. M.: Tactile Perception Studies Related to Teleoperator Systems. NASA-CR-1775, 1971.
2. HILL, J. W.; AND BLISS, J. C.: Tactile Perception Studies Related to Teleoperator Systems. Final Rept. 2 on Contract NAS2-5409, SRI Project 7948, Stanford Research Institute, 1971.
3. JOHNSON, E. G.; AND CORLISS, W. R.: Teleoperators and Human Augmentation. NASA-SP-5047, 1967.
4. CORLESS, W. R.; AND JOHNSON, E. G.: Teleoperator Controls. NASA-SP-5070, 1969.
5. ERNST, H. A.: A Computer Operated Mechanical Hand. Ph.D. Thesis, Massachusetts Institute of Technology, 1961.
6. SHERIDAN, T. B.: Advancements in Teleoperator Systems. NASA-SP-5081, 1970.
7. BLISS, J. C.; AND CRANE, H. D.: Experiments in Tactile Perception. Final Rept. on Contract NAS2-1679, SRI Project 4656, Stanford Research Institute, 1965.
8. BLISS, J. C.: A Relatively High-Resolution Reading Aid for the Blind. IEEE Trans. on Man-Machine Systems, vol. MMS-10, 1969, pp. 1-9.
9. BLISS, J. C.; KATCHER, M. H.; ROGERS, C. H.; AND SHEPARD, R. P.: Optical-to-Tactile Image Conversion for the Blind. IEEE Trans. on Man-Machine Systems, vol. MMS-11, 1970, pp. 58-64.