HUMAN CONTROL OF REMOTE COMPUTER-mNIPULATOFS

Thomas B. Sheridan
William R. Ferrell
Massachusetts Institute of Technology
Cambridge, Massachusetts

Summary

Thi3 paper describes a man-computer-maninulator system for doing a variety of exploration and assembly tasks for space, undersea, hazardous terrestial environments, warehousing and medical applications. Emphasis is placed upon the allocation of functions to man and machine and the nature of their on-line interaction. The human operator through a combination of analogic commands (hand movements to specify direction, magnitude, time relative to his view of the task) and symbolic commands (typewritten alphanumeric characters), sets subgoals, subroutines, and stopping conditions in terms of the manipulator's position and touch sensors. The computer interprets the human commands, reads data from the manipulator's own sensors, performs geometric transformations and executes optimal or heuristic procedures to drive the manipulator actuators. Laboratory experiments with such a system are described, and problems of organizing command languages, designing touch sensors and manipulators are detailed.

Introduction and Problem Statement

Remote manipulation UP to the present time has been carried out mainly in nuclear "hot cells" and to a lesser extent undersea. Generally in these applications a direct mechanical or electromechanical linkage continuouslv slaves the movements of the mechanical hand to those of a handle guided by a human operator a relatively few feet away. The only extra-terrestial application has been the manipulator on the Surveyor spacecraft which was commanded in very simple discrete movements to dig the surface of the moon and to position experimental apparatus. General purpose manipulators have recently been employed in production tasks, wherein the manipulator repeats a stored motion sequence "taught" it by a human operator who moves its degrees of freedom through the desired path to record the important points along it. None of these devices makes use of artificial intelligence or even environmental sensing for automatic control. An extensive historical review of developments in remote manipulation has recentlv been written by Johnsen and Corliss.1

Beginning with the experiments of Ernst several developments of autonomous robots have begun at MIT, Stanford University, and Stanford Research Institute[3,4,5] which have potential application to an unmanned Mars mission. Completely autonomous devices will surely be practical for a wide range of recognition and control tasks eventually. It is the authors' contention, however, that for the immediate future it will be economically expedient to send a manipulator or vehicle having a low level of artificial intelligence but supervised by a human operator on earth. It would transmit to him relatively simple sense data and other information for his interpretation and for use as the basis of further instructions. At lunar and nearer distances the telemetry required is entirely within the present state-of-the-art.

More generally it is believed that non-routine mechanical handling or assembly tasks too remote or too hazardous or scaled too large or too small for direct manipulation by man can best be controlled by a mixture of human and artificial intelligence. The problem is how man and computer should interact. This paper describes laboratory simulation experiments involving human subjects together with computers in real-time control of two, three and seven degree-of-freedom mechanical manipulator devices. In these experiments the computer continuously recognizes simple patterns and effects control trajectories to achieve relativelv near subgoals, whereas the man intermittentlv observes system states and sets subgoals. The paper also illustrates by example the ways in which a human operator and a computer can efficiently complement one another in planning and executing task strategies.

Loop Delays from Signal Transmission, Coding and Process Dynamics

A continuous control system usually becomes unstable when loop delavs exceed one half cvcle at any signal frequency for which loop oain exceeds unity. In control from earth the speed of light poses loop delays (round trip) of 1/3 second to synchronous Earth satellites, 2.6 seconds to the moon and at best several minutes to Mars. To this is added the delay for signal coding and decoding which may be nearly five seconds even for high priority signals in complex state-of-the-art vehicles like Apollo. Low resolution pictures from Mars will take several minutes, high resolution pictures an hour or more. Process lags of inertia and viscosity can add further delay to remote mechanical operations.

Laboratory experiments in human control of simple manipulators (two-dimensional translation and grasp) have shown that with only visual feedback human subjects avoid the slow and erratic movements that delay tends to induce by consistently adopting a move-and-wait strategy, whereby limited open loop moves (without feedback) are punctuated by waits of one loop delay time in order to gain feedback and confirmation or reorientation.[6] Task completion time can be accurately predicted from completion times and measures of open-loop performance taken when therr is no delay. Unfortunately, the move-and-wait strategy takes time and completion time increases linearly with delay. Other experiments with loop delays in tasks having continuous force feedback (one-dimensional contact pushing tasks without vision) indicate that high feedback gain can cause instability and that searching movements must be slow to limit impact forces.

These results suggested that when tine delays are necessarily present the human operator should not control in a continuous closed loop but should serve instead as a supervisor or intermittent monitor and setter of subgoals for an automatic or computer-control loop.[8] The dynamic movement of the manipulator would then be under control of a local computer eliminating stability problems. Moreover, the completion time would become far less sensitive to delay since the operator would command relatively long segments of the task, reducinq the number of waits for correct feedback.

Such a system is illustrated in Fio. 1. A laboratory simulation has been implemented consisting of an American Machine and Foundry Model 8 master-slave manipulator driven bv stepping motors through an augmented PDP-8 computer.

Man-Machine Console Interface

The human supervisor may issue his commands as strings of alphanumeric characters through a conventional teletype as in Fig. 2a (such commands we call "symbolic commands"). He may also indicate direction, magnitude, and time through moving a multi-degree-of-freedom joystick which may even have articulations corresponding to those of the manipulator (these we call "analogic commands"). Experiments by Verplank[9] with a device shown in Fig. 2b suggest it is quicker and certainly more natural for the human operator to demonstrate to the computer certain movements or positions by the analogic mode than symbolicallv. This is true even if the analogic control device has only on-off switches for each articulation so that the operator must himself adjust his arm and body to achieve the position correspondence. Conceivably the analogic control could also take the form of a scaled physical model of the manipulator in a model of the task environment. The human operator*s goals could then be communicated to the computer bv his manual rearrangement of the model.

We have experimented with sensinn and display of both visual and tactile feedback. As one would expect, ordinary television feedback is ouite satisfactory provided the human onerator can, at will, call either for a close-up or a comprehensive view. The close-up is needed for details of grasp or position of tool relative to object, while the comprehensive plan or profile view shows the manipulator base or absolute reference frame, the mechanical hand and the manipulated objects all in relation to one another.

Even the crudest of tactile sensors such as a simple electrical contact is useful for automatically stopping the arm motion if an object is touched. The manipulator arm has sufficient elasticity to render collisions with fixed objects harmless, and transient shocks are quickly damped out. We have also developed a prototype touch sensor having on its surface a deformable mirror which distorts a regular grid pattern. The human operator observing the reflected image through closed circuit television can infer the point by point pattern of normal forces on the gripping surface of the manipulator hand. This device is shown in Fig. 3.

Command Language

In order to develop, on-line, the commands appropriate to a particular task from a small set of primitive commands, a simple compiler called MANTRAN has been written by Barber.[10] It accepts tvped commands to the manipulator to move at a specified rate to a specified position or in a specified direction until certain specified conditions occur, allowing the operator to combine these into "programs" that can be called by name. Such programs may have within them various searching or grasping or emergency conditional subroutines. A simple but useful mode is to be able to assign a name to a given manipulator configuration (simultaneous set of relative positions or angles at each articulation) and at some arbitrary time later to simply type the configuration name and have the manipulator automatically return there. This renuires, of course, that the computer maintain a state vector for the current manipulator configuration with which to compare the stored state vector corresponding to the named configuration. Alternatively, the system can be commanded to respond to the analogic controller until certain positions are obtained or until the hand contacts an object, at which time the program branches out of the analooic mode. An example of MANTRAN is shown in Fig. 4.

Functional Organization of the Computer-Manipulator System

It is convenient to represent the breakdown of functions in the computer manipulator system as in Tig. 5. Whereas Fig. 1 illustrates the physical location of major components, Fio. 5 indicates the various kinds of logical processing necessary for supervisorv control. Most but not all of these functions are implemented in the laboratory system described above.

Beginnino at the left, the ill-defined function of evaluating and setting subgoals is performed by the human operator. He interacts with the symbolic (keyset) and analogic (local physical model) controllers to convey his intentions. A computer element called the command interpreter converts these human responses into a unirrue concatenation of commands, or if certain criteria are not met, returns to the operator via the display with a querv or a reminder.

The operator, and in turn the interpreter, *can g*ive four kinds of commands to an executive controller! 1) he can instruct the exteroceptors (recentors which sense the external environment) to be positioned, oriented and focussed in a certain wav and to tell him what thev sensei 2) he can request that certain "what would happen if___" experiments be conducted within an abstract representation of the manipulator and task internal to the computer; 3) he can request the execution of certain moves—simple ones like move manipulator angle A plus-two degrees, or complex ones like search region R for part P and assemble it with 9; 4) he can rerruest that certain transformations available within the computer svstem be applied to data and that the results then be displayed to him.

The block labelled "task model" is itself worthy of far more discussion than can be qiven it in the present paper emphasizing the nature of the human operator's supervisory control, and its internal organization is discussed much more fully in a separate paper by Whitney submitted for this conference.[11] In summary, however, one can say that if the positions in six-dearees-of-freedom of all rigid elements of both manipulator and environment are represented by a sinqle vector, then any interactive movements of the manipulator with the environment can be represented by a series of vectors or by a trajectory in vector space. In theory, then, optimal control strategies are those for finding minimal cost trajectories from initial state to goal state in the state space.12

While a finely reticulated state space for a system having many motor states, many sensor states, and many environment (manipulated object) states is clearly far too large to be useful for overall control, the state space model provides a formal norm or baseline to which other techniques can be compared. State space models of a tractable size can be used to perform simple manipulations, or to perform parts of more complex ones. In anv case an important practical problem concerns the definition of what are rigid elements, what are costs, and what discrete resolution is appropriate to represent the real world in the state snace mode1.

A simple example of interaction of the human operator with the task model is described in a subsequent section of the paper.

It is important to note that the executive controller oversees manv "automatic" feedback loops and interactions which are not under direct control by the human supervisor. As indicated by dotted lines and letters in Fig. 5, the executive controller mediates direct or straight throuqh remiests for (a) sensory analysis of the environment or for (b) task model experiments. Imperative move commands automatically call for, in sequence, (c) search and identification of objects in the environment, (d) task model experiments, and finallv (e) manipulator movements.

The executive controller mediates (f) the feedback from the exteroceotor actuators (actually the interoceptors of the exteroceptor subsystem, analogous to the position sensors of the muscles of the eyel) and the (g) signals from the task model to control where to look next. Similarly (h) the feedback control of the experiments of the task model or internal representation of the manipulator as well as (i) the feedback control of the actual manipulator are handled by the executive controller.

The task model is updated by (j) the sensed state of the environment and (k) the sensed state of the manipulator. Knowledge of (1) the sensed state of the environment and (m) the results of the task model experiments are automaticallv used in optimal or sub-optimal control of the manipulator. A significant part of this latter function is the decision as to what individual angular changes of actuator movements within the manipulator will achieve ths final desired position. The reverse transformation from component angles to gross configuration is easy trigonometry but the required transformation from joirt or hand position to component angles can reauire complicated matrix inversions, or approximation methods, or worse yet (in the case of redundant degree-of-freedom manipulators) , be undefined. This specific problem has been the subject of several previous papers.[13,14]

Finallv the operator can ask for a varietv of information (n) to be displayed to him. The orqanization of this aspect of the system is probablv very important but, since as yet no experiments have been done emphasizing it, it is left indeterminate.

Note that the executive controller, the task model, the command console interpreter and also the display boxes must all be implemented larcely by digital logic. Which computer functions belonq to the "local" computer and which belong to the "remote" computer are unsolved engineering problems and will surely depend upon, among other thinqs, the specific task context, the amount of telemetry processinq reouired for signals of the necessarv precision and the like.

Human Intervention in Computer Control Procedures

Because of the economic limits on multi-dimensional state space or other formal models of whole manipulation tasks referred to above, a primary goal of our research is identification of the ways human operators and computers can interact in planninq and executing manipulation tasks. This will be presented bv example.

Fig. 6 represents a manipulation task in a reticulated 10 by 10 unit space. The manipulator jaws J and J' and the blocks A and B are each one unit on a side and all moves of blocks and jaws must be an integer number of units in extent. The qoal is to move the jaws so as to get block A to square A'. Neither blocks nor jaws can occupy a crosshatched square (wall). The jaws move horizontallv as one, and vertically J* moves relative to J over four spaces from closed to wide open (they are illustrated one space open)• The jaws can grasp a block or push it; in order to grasp a block the jaws must first straddle it with one unit space on each side.

Table 1 qives an arbitrarv set of primitive functions and tests which can be concatenated in various ways to perform tasks such as the one in our example. The functions and tests are organized into three subroutines GRASP, MOVE and PUSH. Within each subroutine a series of tests is madei for a yes answer the arrow to the right calls for an action and a return to the previous test or a jump to a different subroutine; for a no answer the downward arrow calls for the next indicated test or action. In moving a block from one position to another a qeneral rule is first to try to move bv grasping the block, and if there is no room for that to trv to move by pushing the block.

The task of getting A to A» is still too difficult for the subroutines of Table 1 to handle, and reauires the human operator to intervene at least often enough to break the task into three elemental move B to B*, return jaws to left side of wall, move A to A'; by intervening a bit more often a considerable saving can be made in use of subroutines,

A completely different procedure is for the human operator to provide the defining information for a state space, as referred to earlier in this paper and as described formally in a companion paper submitted for this conference by Whitney .11'[12] Given three objects (two blocks and a set of jaws) which can move to any position on a 10 x 10 grid, a vector space of size 1o2.io2.io2 is required. Since J' can assume one of four states relative to J we need a total space of $4\text{-}10^6$ states. We assume that an initial and terminal state (that is, initial and terminal nositions of jaws and blocks in the 10 x 10 grid) having been specified, an algorithm exists for getting from one to the other in optimal or at least satisfactory fashion, details of which are given by Whitney.

If the operator intervened and called for a series of three initial to terminal state trajectories (B to B,, return jaws to left, A to A^1), then the state space required for each subtask would be considerably smaller than that required for the whole task. For this example B to B' would reouire at most consideration of all jaw states and all B states but no A states, thereby reducing the state space a factor of 10^2. Return of the jaws would require no A or B states, netting a reduction by 10^4. A to A' would net a reduction by 10^2. Clearly, the sum of these state spaces, even if all of the 10 x 10 physical grid is allowed for each task element, is much smaller ($4\text{-}10^4$ ♦ $4\text{-}10^2$ + 4. 10^4 ■ 8.10^4) than the 4.10^6 state space rerruired for the whole task. If, in addition to judgino that certain objects can be eliminated from certain subtasks the human operator also judges that certain regions of the 10 x 10 grid can be eliminated from consideration (for instance in returning jaws from the B' neighborhood to the left side of the wall) then further reduction can certainlv be made.

To make our point about the computational saving which accompanies intelligent intervention by the human operator permit us to digress from our example of Fig. 6. Suppose (Fig. 7) that for a 10 x 10 grid two objects X and Y must interact. At the outset suppose we (the human operator) know that A and B must interact in a oeneral wav within the bounded area 1, then must pass together from area one to area two while located within the 1-2 overlap area, interact within area 2, similarlv pass to 3, interact in 3, pass to 4 and interact in 4. It would not be necessary to consider a state plane of size (10.10) for object A combined with (10«10) possibilities for B, or 10^4. This is because A and B are never simultaneouslv in different zones except when together in the overlap areas. Thus a state space of (5.5) • (5.5) accomodating all A-B combinations in zone 1 overlapping a state space (6.5) • (6.5) in zone 2 similarly overlapping a (6.6) • (6.6) zone 3 state space which in turn overlaps a (5.5) • (5.5) zone 4 state space provides for an optimal trajectory to be selected from among all possibilities. The total state snace is 3446, less than 35% of the 10^4 state space. In general, a task involving n objects, each of which can assume any one of S states, requires a state space of S^n. Breaking the task down into m subtasks in the ith of which only ri of the objects can move and each is limited to o i states renuires a state space ofs

$$\sum_{i=1}^{m} O_i^{r_i}$$

This will, except in unusual circumstances, represent a savinqs in both computational time and storage, since ri=n and 0i=S,

It is our purpose here to compare the savinqs in calculation and storaqe; i.e., the reduction in the total number of states, resulting from specification and delimitation of <u>either</u> heuristic procedures or formal state spaces. We presume that a human operator is the one who intervenes, sets appropriate qoals and calls appropriate heuristic routines, or that he bounds the state space—at least we do not understand how a hiqher level computer proqram can do it.

Returning to our first example, Table 2 shows various elements of the task of Fig. 6. Task elements 1 through 11 are possible parts of the whole task (move A to A^1) and are not mutually exclusive. Task element 1 (move B to B') incomorates the rule "try movinq by graspinq before trying Pushing." This procedure would automatically call subroutines in the order 12 12 3 and utilize 19 moves. For comparison, if the operator saw that pushing should be tried first, only heuristic 3 would be used and remiire 10 moves. The same objects are reauired for the state space procedure in either case and we allow (conservatively) a $4.10^{2}\cdot10^2$ space. In either case the state space procedure can be expected to calculate that a 10 move nush is the best.

Task elements 3 and 4 are simpler elements, so simple that a human operator miqht be expected to call for the actual point to point move path (desiqnated m) instead of calling a subroutine. With either of these the grid space required could be qreatlv restricted.

Task element 5 could be included as part of 6, 7 or 8 thouqh here it is given separately. Element 6 actually illustrates a cascade of four subtasks (move A to A_1, then to A2, then to A3, then to A'). The subgoals are achieved at the slash marks in the subroutine sequence. This series of heuristics would result if a human operator initially thought a shortest path was A to A1, then to A2, then directly to A*. But after gettina to (or committing the program to) A2 he would realize (or have to have pointed out to him) that with A at A2 he has not left room for regrasping A and will have to set a new subgoal A3 to which A can be pushed and then grasped or pushed to A*. If the human operator had thought ahead he would have set A_4 and A_5 as subgoals, and as indicated a shorter string or heuristics would be necessary. Again, by state space procedure, the same (but a verv larae) state space would qet from A to A' automatically.

As a further example of how the operator can delimit the state space, suppose for task element 8 he recognized that from A_1 or A_4 to A2 or A5 the jaws would have to be closed. Then the state space for this portion of the task need consider only block A over a 7.2 grid and the jaws J over a 7.1 qrid. Let the jaws and A operate open or closed-jawed over the grid B wide and ten high at the left for the A to A_1 or A4 subtask and a over a grid five wide and ten high at right for the last part. This then renuires a state space (4»3*10*3-10) + (7.2.7*1) + (4.5.10.5.10) - 13698 in size. This compares with the 40000 state space otherwise needed. The situation is similar to that of Fig. 7.

The task elements 10, 11, 12 and 13 provide a finer breakdown of the A to A' subtask.

Below the double line in Table 2 are shown several ways of accomplishing the whole task; in each case are shown the total strinq or heuristic subroutines and implied moves using that procedure; the total state space is also indicated with the implied moves for that method.

Indices of efficiency for heuristic or state space techniques such as these can be formed by making a ratio of the number of moves, number of routines called, or size of state space for a qiven procedure, to the corresponding number for a normative heuristic or brute force procedures such as the first whole task procedure given. Note that the last and most efficient procedure reauires the human operator to observe that block B may be pushed out of the way by block A without need to modify the A to A' procedure. Much more dramatic efficiencies can be demonstrated for human intervention in the state space techniques when the nhvsical space is more finely reticulated due to the

$$\sum_{i=1}^{m} o_i^{r_i}$$

rule stated above. Heuristic techniques are by nature context dependent and not much generalization can be made.

Conclusion

A laboratory simulation and some empirical evidence have been described which demonstrate the advantages of havinq human operators operate as supervisory controllers of remote computer-manipulators, where the man sets subqoals and sets procedural constraints and the machine does the rest automatically. Many practical advantages of such man-machine interactive systems are foreseen to speed UD remote or monotonous tasks, avoid the costs and risks of supporting man in hazardous environments, and provide a true flexibilitv of exploration and manipulation not near to achievement by completely autonomous machines.

References

1. Johnaen, E.G. and Corlins, W.R., Teleoperators and Human Augmentation, NASA, SP-5047, Dec. 1967,

2. Ernst, H.S., MHi A Computer Operated Mechanical Handi M.I.T. ScD Thesis, 1961.

3. Application of Intelligent Automata to Reconnaiasance, (C.A. Rosen and N.J. Nilason, Eds) SRI report dated Nov. 1966.

4. M. Green, S. Wahlstrom, G. Forsen (Same title), Tech. Report No. RADC TR-66-822, Rome Air Development Center, Griffiss Air Force Base, N.Y., 1967.

5. C. Rosen, N. Nilsson, (same title), RADC TR-67-657, Jan. 1968.

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

7. Terrell, W.R., "Delayed Force Feedback," Human Factors, PP. 449-455, Oct. 1966.

8. Terrell, W.P. and Sheridan, T.B,, "Sunervisorv Control of Remote Manipulation," IEEE Spectrum, pn, 81-88, Oct. 1967.

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

10. Barber, D.J., MANTRANx A Symbolic Language for Supervisory Control of a Remote Manipulator, SM Thesis, M.I.T., 1967.

11. Whitney, D.E., "State Space Models of Remote Manipulation Tasks," paper submitted for International Joint Conference on Artificial Intelligence, Washington, D.C., May 1969.

12. Whitney, D.E., State Space Models of Remote Manipulation Tasks, Ph.D. Thesis, M.I.T., Jan. 1968.

13. Merqler, H.W. and Hammond, P.W., "A Path Optimization Scheme for a Numericallv Controlled Remote Manipulator" 6th Annual Symposium of the IEEE Human Factors in Electronics Group, May 1965.

14. Whitney, D.E., "Methods of Rate and Position Control of Remote Manipulators and Human Prostheses," paper submitted to IEEE Trans. on Man-Machine Systems.

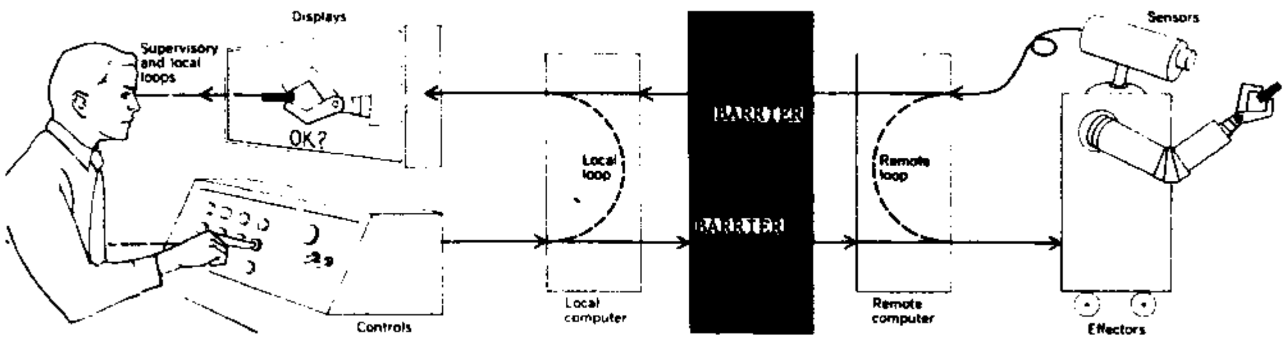
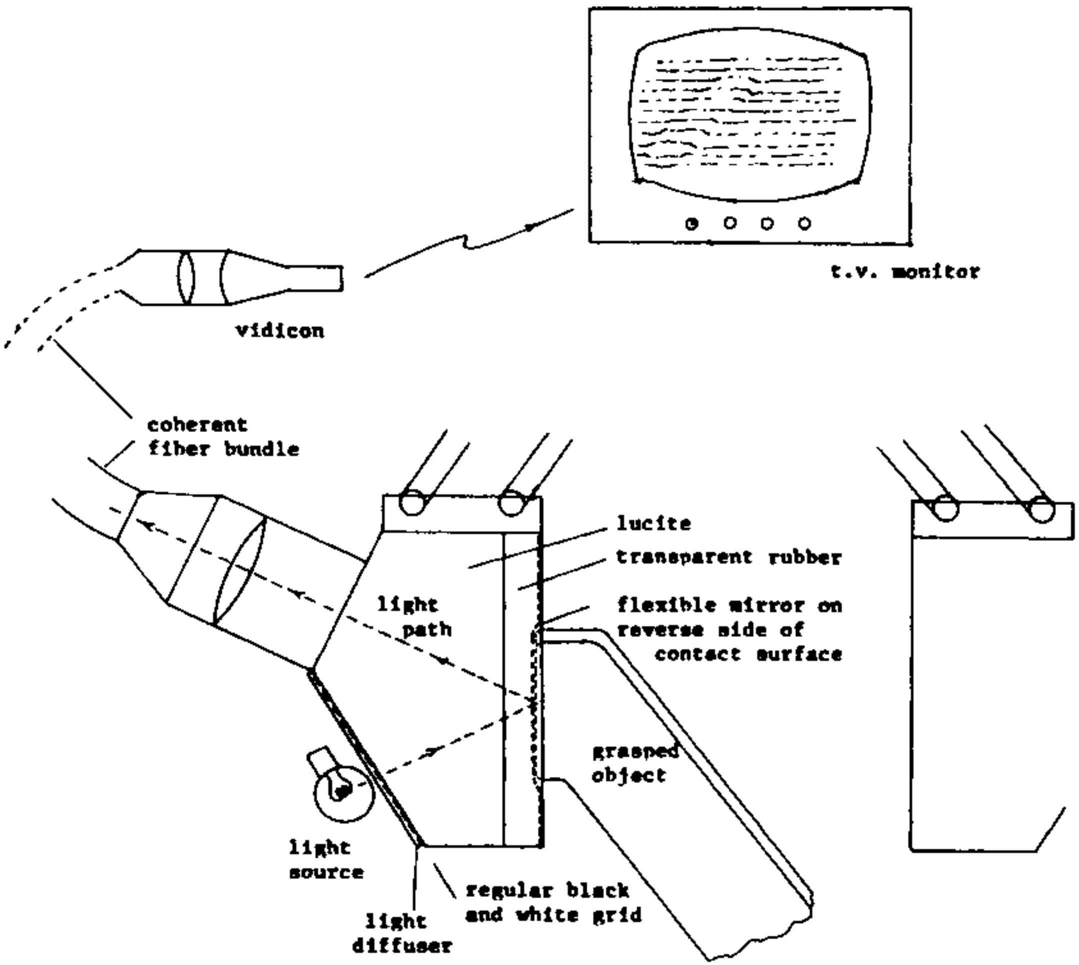


Fig. 1. Schematic Diagram of Supervisory Control of Remote Manipulation.

(a)

(b)

Fig. 2. Symbolic and Analogic Controllers. Above (a) human subject controlling mechanical hand to pick up coffee cup through use of symbolic commands, Below (b) in each of seven degrees of freedom (which correspond to manipulator) operator can operate a three positional (plus, zero, minus) switch. Knobs on the shoulder piece are for switching computer modes (increment, rates, etc.)*

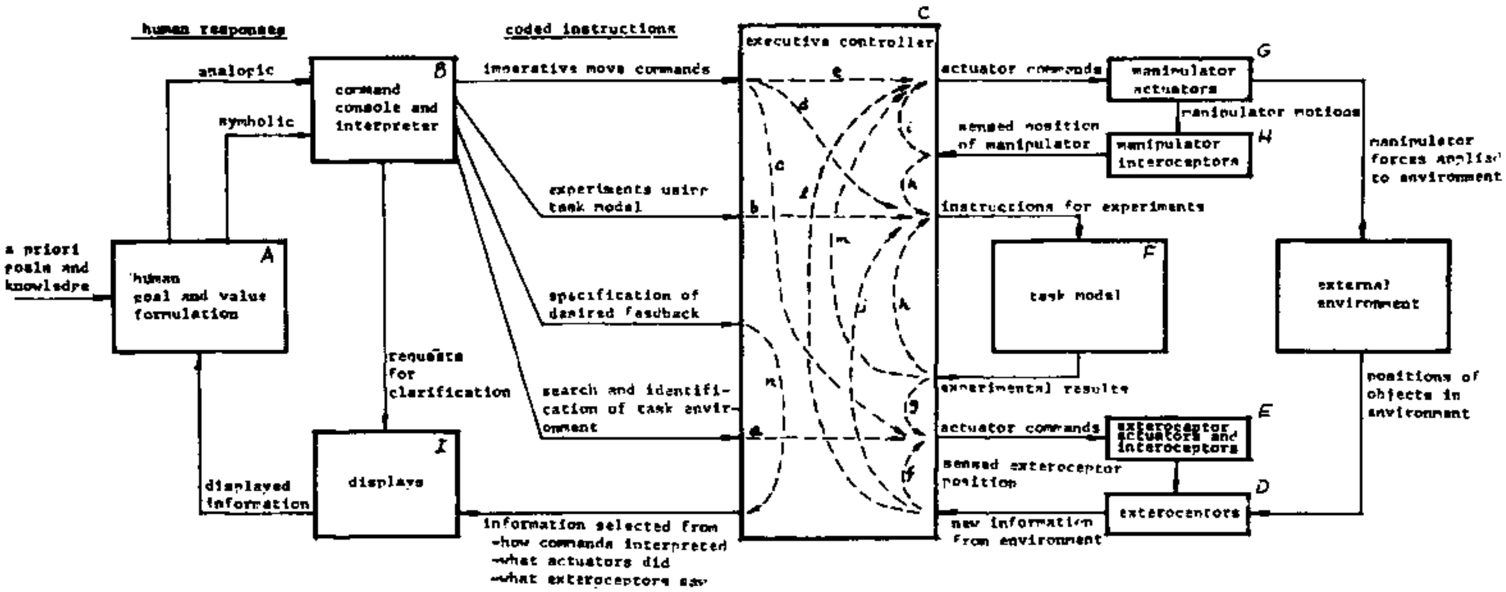


, Fig. 3. Touch Sensor. Grasped object deform flexible mirror which causes distorted grid pattern to occur as human operators visual (not tactile!) display.

```
ST 1  MOVE FORWARD 13 AND LEFT 54.
      UNTIL A)  TOUCH LEFT
         OR B)  TOUCH FRONT.
IF MOVE CONDITION SATISFIED,  GO 2
          IF A)  DO 3
          IF B)  DO 4.

ST 2  MOVE FORWARD 13 AND RIGHT 54)
      UNTIL A)  TOUCH RIGHT
         OR B)  TOUCH FRONT.
IF MOVE CONDITION SATISFIED,  GO 1
          IF A)  GO 5
          IF B)  DO 4.

ST 3  MOVE BACK 43 AND LEFT 24.

ST 4  TYPE "DONE" AND STOP.

ST 5  MOVE BACK 41 AND RIGHT 24.

ST 6  DO 4 ?
```

Fig. 4. Example of MANTRAN. At each step the computer types what is underlined and the human operator responds to set subgoals or procedures.

Figure 5 Functional Taxonomy of Supervisory Control of Computer/Manipulator



a. direct instructions to exteroceptor actuators

b. direct instructions to perform experiments on task model

c. instructions to exteroceptor actuators as part of move strategy

d. instructions to perform task model experiments as part of move strategy

e. direct instructions to manipulator actuators

f. modification of exteroceptor positions as a function of what exteroceptors sensed

g. modification of exteroceptor positions as a function of task model experiments

h. execution of task model experiments as a function of previous experiment results

i. manipulator position feedback control

j. updating of task model as a function of new information on the environment

k. updating of task model as a function of where manipulator is

l. manipulator control as a function of new information on the environment

m. manipulator control as a function of task model experiment results

n. information selection for display

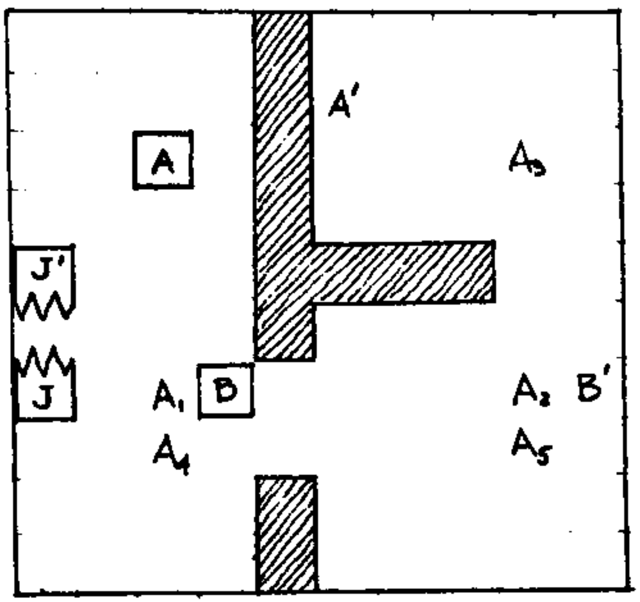


Fig. 6. Manipulation Task in a 10 x 10 Grid: Control jaws J and J' to move A to A' without going through wall.

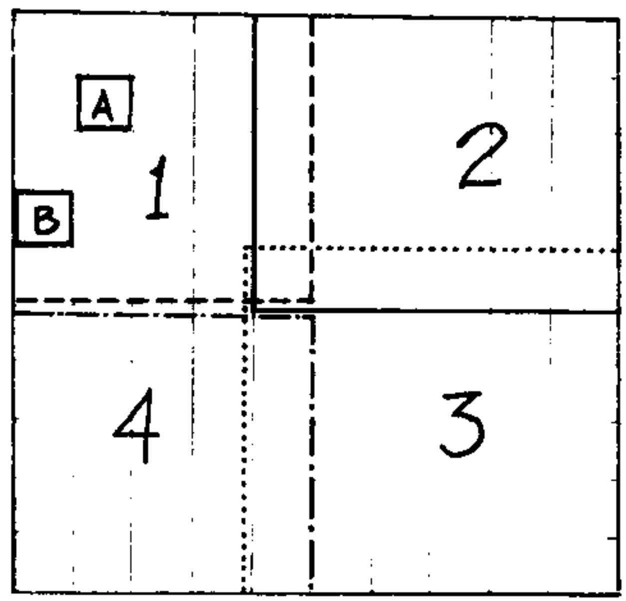


Fig. 7. Example of State Space Reduction. Physical space (a), a 10 x 10 grid. Task requires that A and B after each of interactions in zone 1, 2, and 3 transition to 2, 3 and 4 respectively while being simultaneously in overlap region. This reduces to four state spaces with (5·5), (5·6) and (6·6) and (5·5) respectively.

Table 1. Heuristic Subroutines for Fig. 6.

1. GRASP

a) are jaws grasping block → go to MOVE
↓
b) are jaws straddling block → grasp
↓
c) are jaws lined up to slide to straddle block → straddle
↓
go to MOVE

2. MOVE

a) examine PATH to subgoal
↓
b) is there wide enough free path → take it (done--go to SUPERVISOR for greater subgoal)
↓
c) can jaws be closed more → close
↓
d) is there moveable obstacle in otherwise free path → put marker on execute list, reenter GRASP with new arguments)
↓
e) is there block in jaws → go to PUSH
↓
(stuck--go back to SUPERVISOR for lesser subgoal)

3. PUSH

a) examine PATH to subgoal
↓
b) is there no pushing path exclusive of moveable obstacle → (stuck)
↓
c) are jaws in pushing postion → push (done, go to SUPERVISOR for greater subgoal)
↓
d) is there wide enough free path to move to pushing position → move
↓
e) is there a moveable obstacle in path to get to pushing position → put marker and arguments on execute list, reenter PUSH with new arguments

(stuck--go back to SUPERVISOR for lesser subgoal)

	TASK ELEMENTS	ALTERNATIVE PROCEDURES: HEURISTICS — Routines Used (Refer to Table 1)	HEURISTICS — Number of Moves	STATE SPACE — Approximate of Size State Space	STATE SPACE — Number of Moves
1.	Move B to B'	12123	19	40000	10
2.	Push B to B'	3	10	40000	10
3.	Line up jaws to push B thru hole	m	4	20	4
4.	With jaws lined up push B to B'	m	6	100	6
5.	Return jaws to left side	23	6	40	6
6.	Move A to A' (thru A_1, A_2 and A_3)	1212/12123/12123/1212	45	40000	45
7.	Move A to A' (thru A_2 and A_4)	1212/12123/1212	41	40000	41
8.	Move A to A' with constrained state space	-	-	13660	41
9.	Move A to A_4	1212	14	3000	14
10.	Move jaws to push A position	m	6	50	6
11.	With jaws lined up push A to A_5	m	5	10	6
12.	Move A from A_5 to A'	1212	15	4000	15
	Whole task (1, 5, 7)(one state space)	12123/23/1212 12123/12123/1212	66	4000000	57
	Whole task (1, 5, 7)(two state spaces)	same	66	80000	57
	Whole task (2, 5, 7)(two state spaces)	3/23/1212 12123/12123/1212	57	42000	57
	Whole task (3, 4, 5, 9, 10, 11, 12)(seven state spaces)	m/m/23/1212/m/ m/1212	57	7200	57
	Whole task (9, 10, 11, 12)(four state spaces where push one block with the other)	1212/m/m/1212	41	7100	41

Table 2. Various Task Elements and How They Combine To Accomplish the Task of Fig. 6.