INTELLIGENT INTERFACE FOR REMOTE SUPERVISION AND CONTROL OF UNDERWATER MANIPULATION

Azad M. Madni, Yee-yeen Chu and Amos Freedy
Perceptronics, Incorporated
6271 Variel Avenue
Woodland Hills, CA 91406

Abstract

Underwater manipulation continues to pose challenges to both robotics and computer technologies. With recent advances in sensor and artificial intelligence technologies, "intelligent" robotics presents a novel and effective solution to coping with problems typically encountered in underwater manipulation tasks. To date, a number of manipulation systems with computer-based self-monitoring and planning capabilities have been demonstrated. While the principal contribution of these systems is to replace, augment, aid or improve operator performance in underwater tasks, overall system performance still depends on the effectiveness of the combination between the robotic system and the human counterpart. To this end, an "intelligent" interface (I^2) is proposed as a means to dynamically manage information, allocate tasks, and actively mediate between the operator and the robotic manipulator in a manner that optimizes their joint performance. The potential efficacy of the I^2 is discussed in relation to specific aspects of remote supervision in underwater manipulation tasks (e.g., effective data portrayal, plan communication, contingency handling, and task monitoring). An example of a link between AI and a mechanical underwater manipulator is shown via a prototype system, Task-Oriented Supervision Command System (TOSC). TOSC is a procedural net-based supervisory planning system that provides interactive programming of manipulation tasks in a high-level command language. A series of experimental studies were performed to evaluate TOSC features on a remotely-controlled six degree-of-freedom anthropomorphic manipulator system with an integrated control and display console. Laboratory evaluation of underwater manipulation and maintenance tasks at various levels of task complexity and manipulation difficulty was conducted. The results showed TOSC to be a highly effective and well accepted aid that both improved performance and reduced errors, especially in complex manipulation tasks under uncertain environmental conditions.

Introduction

Underwater manipulation is an ongoing research area that poses challenges to both robotics and computer technologies. Underwater manipulation devices have to perform in unstructured, uncertain and highly hostile situations resulting from the poor visibility, inaccessibility and unpredictability of the underwater environment. The manipulation task in underwater environments is significantly different from that in the industrial environment. In industrial settings tasks are routine with well-defined goals. In contrast, the underwater environment presents a situation where our sparse knowledge of hazards has to be used to introduce safety measures for dealing with largely unknown contingencies. As a result, in present underwater remote manipulation, most of the control tasks are manually performed. One method to improve the slow and often error-prone performance of remote manipulators is to augment or automate most of the operator's control functions. The current teleoperation approach allocates functions to a computer, as well as to the human operator. Such an allocation retains the favorable attributes of human intelligence and foresight, and combines these attributes with the obvious advantages of automatic, computer-controlled operation.

With recent advances in sensor and artificial intelligence technologies, "intelligent" robotics presents a novel and effective solution to coping with problems typically encountered in underwater manipulation tasks. To date, a number of manipulation systems with computer-based self-monitoring and planning capabilities have been demonstrated. These systems act as autonomous robots that respond effectively to their environment in the course of pursuing task-related goals preprogrammed by the operator. While the principal contribution of these systems is to replace, augment, aid or improve operator performance in underwater tasks, the human still plays an essential role. Overall system performance still depends on the effectiveness of the combination between the robotic system and the human counterpart. As we extend shared control of a manipulation system to the full range of capabilities afforded by the Artificial Intelligence computer elements, the question of man-machine communication becomes of primary importance. In any manipulation task, the operator must observe the actions of the manipulator, make judgments of the commands necessary to perform the task, and carry out those judgments in terms of manipulator control. However, the involvement of the operator in performing these tasks is strongly affected by the degree of assistance provided by the machine intelligence and by the role of the operator as defined by the task environment and the machine capabilities. These operator roles in automated underwater manipulation systems can be subsumed under the generic monitor, manager, and maintainer (M^3) functions [1].

As monitor, the human oversees the operation of a largely autonomous manipulator in terms of the health of various subsystems, exercising occasional override when task progress in his opinion is unsatisfactory. As manager, the human has a truly multi-faceted role. These include goal setting, decision-making, subsystem supervision, data organization, system control and contingency handling. Exactly how burdensome the manager role is depends on the data volume, complexity, time availability and available information management aids. In the maintainer role, the human is primarily concerned with (1) identifying and diagnosing malfunctions in various subsystems when breakdown occurs and performing both preventive and corrective maintenance, and (2) handling contingencies, where the human plays two key functions. The first is to anticipate the occurrence of abnormal situations. The second is to cope with malfunctions once they have occurred. In either case, the human has to understand, develop and/or implement containment measures to circumvent potential problems associated with the malfunction or unexpected situation. Dynamic replanning on the part of the human plays a major part under these circumstances.

The multi-faceted role of the human operator within the generic M^3 functions forms the basis for identifying critical man-machine interface problems in underwater manipulation. In order to tailor and communicate relevant status, sensory, and operational information to the human operator in each of his capacities (i.e., monitor, manager, maintainer), the interface must provide a display (1) giving him timely and cognitively-compatible information necessary for the manipulator operation, (2) providing him not only the information to discriminate the system and task

CH1972-9/83/0000-0106 $1.00 © 1983 IEEE

status, but also the information to identify specific functions introduced via the automation of the manipulation functions, and (3) providing him sufficient cues for reliable interpretation within the context of the task and system environment

To this end, what is needed is to provide "machine intelligence" in the interface to cope with environmental uncertainties and yet inevitable information deluge, to effectively manage information exchange between the operator and the systems, thereby maximizing the joint performance of the two. This pressing need provides the impetus for the development of an "intelligent interface" (I^2).

Intelligent Interface

As noted above, the purpose of the I^2 is to maximize operator-manipulator performance for a given degree of machine intelligence and a given level of operator capability. To this end, the design goals of the I^2 are to:

1. Compensate for the lack of commonality in the operator and the manipulator system world models.
2. Opportunistically select the communication media, language and information "chunking" for ease of assimilation on the part of the operator.
3. Simplify and enhance the operator's command and query facility.
4. Transform and sequence presented information to meet the expectations and fit within the cognitive constraints of the operator.
5. Simplify task-related knowledge base management and information retrieval.

The complexity of the I^2 design is to a large extent determined by how complex the task is and how "intelligent" the manipulator is. From a manager's viewpoint, the "smarter" the robotic manipulator and the more complex the decision environment, the richer is the repertoire of possible decision options the operator is faced with. In this situation, the intelligent interface can aid the operator by communicating the system status (i.e., current subgoal, selected option) when expected at the right level for the operator. From a maintainer's viewpoint, the more complex the automation, the greater the number of possible system malfunctions. This, in turn, implies greater demands on the comprehensibility (tractability) and explanatory capability of the interface, especially in clearly communicating ongoing activities, explaining possible malfunctions and providing guidance in performing recovery/checkout procedures. From a monitor's viewpoint, the more burdened the operator, the more critical are the various aspects of feedback information (e.g., when to summarize information, what format is appropriate, what modality, what order).

A generic architecture of an I^2 design is shown in Figure 1. On the human side of the ledger, high-level commands/queries are decomposed by the I^2 interpreter into primitive instructions directly executable by the machine during the situation assessment and planning/replanning cycle. On the system side of the ledger, information of various types (e.g., sensor data, status update, plans, contingencies, malfunction alarms, responses to operator queries) from various automated functions such as preview data collection, situation assessment, and planning are presented to the operator in the proper sequence at a suitable level of abstraction and via the appropriate modality.

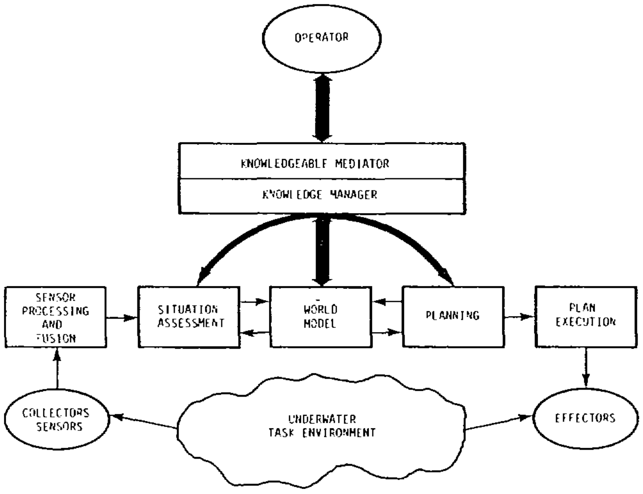

Figure 1. Design Architecture for an Intelligent Interface

The key functional features that can facilitate timely and cognitively-compatible information portrayal for the above process are:

1. Information Management Aids. In any of the M^3 roles, the operator is the recipient of various forms of information sensed and/or generated by the automated system. This information can range from various forms of situational/environmental inputs to internal status messages. This can create operator overload. To compound the problem, both the incoming and internally generated information may sometimes be redundant or irrelevant. In this set of circumstances, embedded models of the human operator could be used to selectively filter, prioritize [2], summarize and pace [3, 4] information presentation. The main purpose of these aids is to reduce operator burden by optimizing the relevance and presentation sequence of external and system generated information.

2. Tailored Status Presentation. Cognitive compatibility of information presentation can be achieved by: (1) employing multi-modality, multi-mode and multi-level information presentation, and (2) minimizing mismatch between the operator's mental model and the robotic system's world model.

Multi-modality feedback within the I^2 framework is geared to distributing feedback information among the various sensory channels (i.e., visual, auditory and tacticle modalities) such that no single modality is over-stressed and overall channel capacity is optimized. Multi-mode feedback refers to personalization of feedback, i.e., textual, graphic or voice communication based on operator preference, information type and situational demands. Multi-level feedback implies information at various levels of abstraction to: (1) cope with situational demands and time stress, and (2) adapt to users of various training and expertise levels.

A major purpose of the I^2 is to bridge the "gap" between the world model of the operator and that of the automated system. The degree to which each has a reasonable model of the other's goals and behavior will determine the degree to which the two can successfully cooperate in a task. At a minimum, the human carries a degree of understanding of the capabilities and limitations of the automated system. In addition, the human operator's mental model may typically contain a miscellany of lore and experiences including a self-updating history of the system. The world model of the robotic counterpart is designed by the computer software engineer with specific views on system functions, operations and overall behavior. In order to successfully cooperate on a task, both the system and its human counterpart should carry an implicit or explicit model of the expected behavior of the other.

3. High-Level Command and Query Language Facility. It is a generally accepted principle that users of computer systems should not be required to communicate with the system at a level of detail that requires significant understanding of the system. That is, the system should be transparent to the user. High-level task-specific command and query languages are typically constructed to allow the operator to communicate with the system at a level of abstraction convenient to the operator. Task representation models based on procedural nets [5] and extended Petri nets [6] have been successfully used to hierarchically decompose high-level command elements that can be executed by the robotic system.

4. Explanation Facility. Explanation capability is necessary for two main reasons: first, to respond to context-specific queries on the part of the operator; second, to keep the operator oriented within the overall task execution cycle, or to keep the operator in track with the strategy or direction of the automated system. The latter rationale is especially important if the problem-solving style employed by the system is significantly different from that of its human counterpart in the course of cooperative problem-solving.

5. Adaptive Task Allocation. Here an adaptive function allocation approach is proposed in which either the computer or the operator may perform a given function based on the availability of information and opportunity to perform the tasks [7]. Dynamic task allocation has several potential advantages. First, it ensures a more fault tolerant setup in the event of system malfunction or failure. Second, dynamically allocating tasks with sensitivity to operator workload allows for maintaining a more effective and consistent workload level. Third, a more effective utilization of system resources/operator capacities becomes possible with potential for improved operator task performance.

6. Task Performance Monitoring and Contingency Handling. Monitoring of task performance is important from two viewpoints: (1) to keep the operator apprised of where the system is in its task execution cycle, and (2) to be able to assess and localize system malfunctions. Contingency handling capability based on a task monitoring model [6] within the overall I^2 framework can allow the system to opportunistically communicate system malfunctions or contingencies at levels specific to the malfunction and along lines comprehensible to the operator.

7. Embedded Default/Reflex Responses. Incorporation of default (i.e., "hard-wired" or reflex) responses in the interface has two main advantages. First, search time can be minimized thus freeing the combined resources/capacities for more important functions. Second, the repeatability, and consequently, the reliability of such responses in similar situations allows the operator to develop a good feel for situations that "trigger" these responses. This is particularly useful in contingency handling or recovery-from-error situations.

A Task-Oriented Supervisory Command (TOSC) System

To illustrate the design and implementation of the I^2 concept in underwater manipulation, an example of a link between AI and a mechanical manipulator is discussed in this section. In this development, a hierarchical model based on the concept of procedural nets [8, 9] was developed to represent the planning process of manipulator action. The model was adapted to represent the man-machine communication process as a step in this planning process. In this application, the human operator develops the global task into a plan at some intermediate level of detail which he then communicates to the computer. The computer develops the plan further to the level of detail necessary for controlling the manipulator via position and force-sensor feedback. The language developed from this model is hierarchical and task-oriented allowing close cooperation between man and machine with a mixed initiative control allocation protocol.

Based on the communication model developed and the results of previous experiments, a shared man-computer communication and control language was developed that allowed the operator to specify manipulator movement in global, whole-task terms. This command language was designed to include the capability of allowing the operator to define a sequence of actions at multiple levels and recall the sequence with a single command string. Specifically, the command language consisted of two major elements: symbolic and analogic commands. The symbolic commands consist of two categories of commands: Primitive and Variable. The Primitive commands include the conceptual unitary motions. For the tasks used in our studies, the following Primitive commands were used:

FORWARD/BACKWARD
GRASP/RELEASE
ROTATE CLOCKWISE/COUNTERCLOCKWISE
MANUAL

The Variable category of commands are the user-defined commands which provide the capability to construct task specific commands. These Variable commands include:

POINT - moves the manipulator to a previously recorded position.
PATH - moves the manipulator along a previously recorded trajectory.
REVERSE PATH - moves the manipulator along a path from end to beginning.
CHAIN - moves the manipulator according to a recorded sequence of Primitive and Variable commands (including other CHAINS).

The CHAIN command permits the operator to aggregate several unitary manipulator motions together into a single complex motion, thus commanding tasks which are higher in the hierarchy of tasks. The command construct is straightforward and easy to use, as it can be described by the following rules:

MANIPULATION := {Command|Chain}
Chain := CHAIN ID + {Command|Chain}
Command := Symbolic|Analogic
Analogic := MANUAL + $\{X_1\ X_2\ X_3\ X_4\ X_5\ X_6\}$
Symbolic := SWITCH|FIXED|VARIABLE
Switch := SENSOR|RATE|CONTROLS
Fixed := ROTATES|FORWARD|BACKWARD|GRASP
Variable := Point|Path
Path := PATH ID + {Path|Point + Path}
Point := POINT ID + {Spatial position}

The rules read: "A manipulation consists of a group of single command or chain. A chain consists of chain declaration followed by a group of single command or chain. A single command is either symbolic or analogic," and so on. This SMC^3 language was implemented in Interdata 70 minicomputer assembly language. 20K memory words were used in the current system.

A prototype teleoperator command and control system has been developed to investigate the effects of computer aiding on the ability of trained operators to perform selected teleoperator tasks. The facility contains a seven degree-of-freedom, electronically controlled and hydraulically powered, servo manipulator, an Interdata minicomputer, and man-machine interface. The man-machine interface includes an integrated control and display console, a shared man-computer communication/control language for manipulation, and visual and force sensing devices.

Figure 2 shows a drawing of the teleoperator control/display station consisting of the dedicated command keyboard and two three-degree-of-freedom, rate control, self-centering joysticks. A 9" CRT display was used to provide alphanumeric messages concerning the condition of the system, the keyboard inputs, and the commands in the execution queue. A 2-view black and white TV viewing system was placed in front of the control panel. Two video cameras were placed approximately at a 90° orientation in the work space to provide two distinct views on the video monitors. This control console was placed in a room adjacent to the manipulator task room to provide the simulation of remote control and an indirect viewing situation.

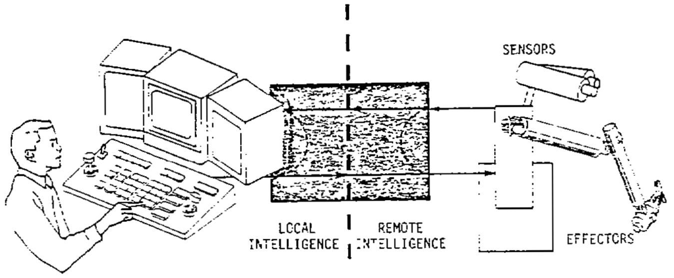

Figure 2. I^2 Control/Display Station for Remote Manipulation

A set of experiments was performed to study the effect of automation on three levels: lexical (control and programming), syntactic (command mix), and semantic (command organization):

1. Control and Programming is the effect of the basic teleoperation aiding functions. The two dimensions considered were (1) augmented (serial) computer control, and (2) traded (parallel) computer control. The levels of the first dimension were (a) individual joint rate-control (Direct), and (b) resolved-motion rate-control (RMC). The levels of the second dimension were (a) manual control, and (b) automatic motion control (AMC), where the operator could call on a predefined point and path via symbolic, push-button control.

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

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

Table 1 presents the comparative results of time performance improvement for various command/control modes based on execution time ratios. As can be seen, performance ratios vary significantly among control/command modes, task types and environmental conditions. The first noteworthy observation is that high-level chain commands with a complete set of language features showed the highest improvement in task execution, especially under degraded viewing conditions. However, as expected, the degree of improvement varies with task type.

Table 1. Performance Time Ratios for a Simulated Maintenance Task

COMMAND MODES	PERFORMANCE TIME RATIOS
RMC/Direct	0.91*
(AMC+RMC)/RMC	0.98
Variable/Fixed	0.95
Chained/Fixed (Normal Viewing)	0.44**
Chained/Fixed (Degraded	0.40**

$^{*}p < 0.05$

$^{**}p < 0.005$

Another issue of major concern is to provide proper feedback to the operator under different automation levels, i.e., the machine state and environment (workspace) feedback requirements. The formal requirements are derived from the fact that the symbolic command language alters the cognitive load and task requirements of the human operator. Much of the new requirements involve operator-system communication in multi-mode supervisory control where the operator (1) selects and confirms available computer-assisted functions, (2) determines which control mode is in force, and (3) monitors the progress of automated routines. The latter requirements are due to the dominant underwater environmental factors such as sediment, turbid water, failure of external lighting, or poor angle of view, which often result in poor visibility conditions. One suggested means to compensate for poor visibility is to selectively automate some of the manipulator motions or procedures. Such preprogrammed computer controls, once initiated, can proceed in an "orderly" fashion.

A second set of experiments was therefore performed to study the multi-mode visual feedback performance via an advanced display/control console equipped with 3D-stick-figure graphics, 2-view black-and-white TV viewing system, and 9" state-feedback monitor. The variables of interest included:

1. Machine state feedback level.
    a. Gross state feedback -- display of current command only.

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

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

3. Display type.
   a. Alphanumeric -- detailed display of commands and task status.
   b. Stick-figure 3D graphic.
   c. TV video -- single-view (1 view) or two-view TV.

The purpose of the experiment was to determine the appropriate level and type of pictorial display to be used in automated manipulation. The results, shown in Table 2, were grouped into two categories: the control/definition phase, where the operator was mostly in the control loop, and the monitoring/execution phase, where the operator was mostly in the supervision loop, with but occasional control activities. In general, performance was more sensitive to: feedback conditions in the control phase than in the monitoring phase, and to feedback level than to feedback format within the ranges of variation made during the studies. Details of this study are described in Chu, Madni, and Freedy [5].

Table 2. Performance Time Ratios for the Experimental Tasks

FEEDBACK MODE	PERFORMANCE TIME RATIOS	
	CONTROL/ DEFINITION TASKS	MONITORING/ EXECUTION TASKS
2-View/DEG 2-View	0.85*	0.94
(2-View + SF)/2-View	0.91	0.93
1-View/DEG 1-View	0.85*	---
3D Graphic/1-View	0.86	0.97
SF/1-View	---	1.00

$^{*}p < 0.05$

Overall, the greatest advantages of computer-aiding seem to occur in situations in which the input/feedback are degraded due to low fidelity of work-space representation or occluded environment. These advantages can be realized by improving man-to-machine communication and/or machine-to-man communication. Also, the extensive research in manual control demonstrates that the more able the subject is in predicting future course of event/trajectory, the more efficient is his control. The data collected in this study can well explain and generalize this predictive effect on performance. It appears that the TOSC language helped the subject to specify the general structure and organization of his knowledge about complex manipulation tasks in which plans for the future interact with the evolving state of the process and its control requirements. Better performances can then be attributed not only to control automation, but also to the efficient and confident planning processes embodied in the use of the command language.

Conclusions

This paper has presented an intelligent human-machine interface for effective management and control of underwater manipulator tasks. It has been suggested that the underwater manipulation poses great demands on the human operator primarily because of the poor visibility and the unpredictability of the environment. The intelligent interface has been suggested as a means for both enhancing the operator-manipulator performance and for reducing overall workload and effort the operator puts out. An illustrative example of an I^2 is presented in the form of a high-level operator-oriented command language. In experimental studies performed by the authors, this faculty was found to both increase overall manipulation performance and decrease operator errors.

References

[1] Lyman, J. and Madni, A.M. Operator Roles in Robotics, a paper submitted for publication in Robotics Age, 1983.

[2] Madni, A.M., Samet, M.G., and Freedy, A. A Trainable On-Line Model of the Human Operator in Information Acquisition Tasks, IEEE Trans. on Systems, Man, Cybernetics, Vol. SMC-12, No. 4, July-August 1982.

[3] Samet, M.G. Automated Information Selection and Pacing, in Proc. 1978 Int. Conf. Cybernetics and Society. Tokyo, Kyoto, Japan, Nov. 1978, New York, IEEE, pp. 1412-1420.

[4] Chu, Y. and Freedy, A. Computer-Aided Information Handling in Supervisory Control of Airborne Systems, IEEE Proc. Int. Conf. Cybernetics and Society, Seattle, WA, Oct. 1982, pp. 425429.

[5] Chu, Y., Madni, A.M. and Freedy, A. Communication with Remotely-Operated Robotic Manipulator via Task Oriented Supervisory Command Language, to appear in IEEE Trans. on Systems, May, and Cybernetics, 1983.

[6] Madni, A.M., Chu, Y., Purcell, D., and Brenner, M.A. The Use of Extended Petri Nets (EPN) in the Modeling and Analysis of Cooperative Maintenance Tasks, Draft Technical Report PDTR-1125-83-7, Perceptronics, Inc., Woodland Hills, CA, July 1983.

[7] Freedy, A., Davis, K.B., Steeb, R., Samet, M.G., and Gardiner, P.C. Adaptive Computer Aiding in Dynamic Decision Processes: Methodology, Evaluation, and Application, Technical Report PFTR-1016-76/8/30, Perceptronics, Inc., Woodland Hills, CA, August 1976.

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

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