AMES GRANT
IN-53-CR
48066
P-35

Acquisition and Production of Skilled Behavior in Dynamic Decision-Making Tasks:

Semiannual Status Report*

for

NASA Research Grant NAG2-656

Principal Investigator:

Alex Kirlik
School of Industrial and Systems Engineering
Georgia Institute of Technology
Atlanta, GA 30332-0205

(NASA-CR-188962) ACQUISITION AND PRODUCTION OF SKILLED BEHAVIOR IN DYNAMIC DECISION-MAKING TASKS: MODELING STRATEGIC BEHAVIOR IN HUMAN-AUTOMATION INTERACTION: WHY AND AID CAN (AND SHOULD) GO   N92-13576

Unclas
G3/53 0048066

*As per agreement, this semiannual status report consists of a Center for Human-Machine Systems Research Technical Report: Number 91-5.

Modeling Strategic Behavior in Human-Automation Interaction:

Why and "Aid" Can (and Should) Go Unused

Alex Kirlik

Center for Human-Machine Systems Research
School of Industrial and Systems Engineering
Georgia Institute of Technology
Atlanta, GA 30332-0205

Report 91-5

September 1991

This research was supported by NASA Ames grant NAG2-656, Robert J. Shively, technical monitor.

Abstract

Advances in computer and control technology offer the opportunity for *task-offload* aiding in human-machine systems. A task-offload aid (e.g., an autopilot, an "intelligent" assistant) can be selectively engaged by the human operator to dynamically delegate tasks to an automated system. Successful design and performance prediction in such systems requires knowledge of the factors influencing the strategy the operator develops and uses for managing interaction with the task-offloa aid. We present a model showing how such strategies can be predicted as a function of three task context properties (frequency and duration of secondary tasks, costs of delaying secondary tasks) and three aid design properties (aid engagement and disengagement times, aid performance relative t human performance). Sensitivity analysis indicates how each of these contextual and design factors affect the optimal aid usage strategy and attainable system performance. The model is applied to understanding human-automation interaction in laboratory experiments on human supervisory contro behavior. The laboratory task allowed subjects freedom to determine strategies for using an autopilo in a dynamic, multi-task environment. Modeling results suggested that many subjects may indeed have been acting appropriately by not using the autopilot in the way its designers intended. Althoug autopilot function was technically sound, this aid was not designed with due regard to the overall task context in which it was placed. These results demonstrate the need for additional research on how people may strategically manage their own resources, as well as those provided by automation, in an effort to keep workload and performance at acceptable levels.

Introduction

It has long been recognized that measures of individual task performance are insufficient to predic overall human performance in multi-task human-machine systems. Many human performance limitations concern, not difficulties associated with performing any single task, but rather constraints on a person's ability to meet multiple, possibly concurrent task demands. As a result, a great deal of research has been directed toward understanding time-sharing (e.g., Wickens, 1984; 1987; Navon and Gopher, 1979) and serial task selection and switching (e.g., Senders, 1964; Moray 1986). Quite often, psychological constructs (e.g., workload, resources) have been proposed to describe and measure the performance limitations which come into play when a person manually attempts to do many things at the same time.

Due in part to these limitations on human performance, aids have been introduced into a variety of human-machine systems to allow the operator to selectively offload tasks to automation. While some of these aids have reduced workload and improved performance, in other cases automation has not been so successful. A number of factors must be taken into consideration when attempting to predict the effects of introducing aids into the operational environment. One issue of great importance is the strategy the operator develops for managing interaction with an aiding device.[1] Human supervisory controllers have the capability, and often the freedom, to strategically manage their interaction with automation in an effort to keep both workload and system performance at acceptable levels. As a result, knowledge of the factors influencing strategy selection is required to predict the system-level effects of introducing operator aids. Unfortunately, there is a shortage of modeling techniques capable of predicting strategic behavior in human-automation interaction. However, unless the psychological constraints on the design of automation are at least as well articulated as are the technological constraints and opportunities, the real potential for technologically-driven, rather than user-centered design, is only made more likely.

The present research is an attempt to identify features of both aid design and task context that influence the strategy an operator will develop for interacting with one particular class of automated systems. The current focus is on the design and use of *task-offload* aiding. Task-offload aids (e.g., an autopilot in an aircraft, cruise control in an automobile) can be selectively engaged by the operator to dynamically offload specified tasks to an aiding device. In complex technological systems, the

[1] Or, the strategy developed and mandated by an organization. The techniques presented in this paper for specifying appropriate aid usage strategies are relevant to both the individual controller as well as to an organization concerned with developing procedures for human-automation interaction.

rationale behind such aiding may be to achieve the potential economic benefits associated with aid performance, to reduce the need for time-sharing and task switching, or to free the operator's resources to pursue long range planning or decision-making. In such systems the operator has responsibility for engaging, supervising, and disengaging the aiding system. In addition, the operator typically retains the ability for direct manual control when deemed appropriate. Thus, unlik passive automation (e.g., automatic transmission in an automobile) task-offload aiding requires the operator to develop and implement a strategy which specifies mode of control (manual vs. automatic) based on an assessment of task demands and performance objectives.

One set of factors which determine whether task-offload aiding will be beneficial are the design features of the aid itself. Of these design features, perhaps those recieving most attention from the engineering community concern the technical performance characteristics of the aiding device. This focus is appropriate, for no matter how much care is given to the human factors of automation design, the fact remains that high levels of technical performance and reliability are necesary attributes of any automated system considered for the operational environment. However, aid design features other than performance also may critically influence the overall contribution of automation to system effectiveness. For example, the time (and most likely the effort) required to program, engage, and disengage a task-offload aid are likely to influence the strategy developed by the operato to govern aid use. A computer, for example, can perform a given multiplication millions times faster than can a person, but the time and effort required to write a program to multiply 28 x 34 mitigates any speed advantages relative to performing the multiplication by hand. If an operator perceives that the potential benefits of an aid are similarly outweighed by engagement and disengagement burdens, the aid may go unused. Thus, dimensions of aid design concerning both performance and the control interface are likely to influence strategy development and resulting system performance.

However, it can be expected that aid design dimensions only partially determine how operators will come to interact with task-offload automation. An additional set of issues, sometimes given insufficient attention, concern the overall task context in which the aid is deployed. In this paper, task context is taken to mean the frequency, duration, and criticality of "secondary" tasks which divert the operator's attention from the "primary" control task. By primary task, we mean a demand for ongoing or high frequency control activity which the operator may choose to meet with either manual or automatic control. By secondary task, we mean an intermittent or low frequency demand for typically discrete control activity that would normally require a diversion of operator attention from the manually performed primary task. Features of the both task context and aid design most likely interact to produce different strategies for best using an aiding system. For example, if secondary task duration is long with respect to aid programming and engagement time, one might expect that it would be best to engage an aid to perform the primary task before turning attention to the secondary task. On the other hand, if programming time and engagement time is long, or delays in initiating the secondary task have high cost, an operator might deem it best to let an aid go unused and to find some other strategy for coping with multi-task demands.

In order to gain insight into these issues, a modeling approach was developed to predict strategy development and estimate system performance as a function of both aid design and task context. Three aid design parameters (aid engagement and disengagement times, aid performance relative to human performance) and three task context parameters (frequency and duration of secondary tasks, costs of delaying secondary tasks) are represented in the model. When applied to a particular task environment, the model identifies an optimal policy for aid use and also estimates maximum attainable system performance with the optimal policy. Sensitivity analysis can be used to measure the effect of varying design and context parameters on the resulting optimal policy and attainable system performance. One use of the model would be the partial specification of aid design parameters given a task context. Thus, if the designer would like to ensure that an aid could be used effectively as a task-offload device, minimal performance and setup time parameters for aid design can be estimated. This modeling approach could also be used to assess the feasibility of introducing an existing aid into a new task environment, or to determine effective strategies for using newly introduced aiding systems.

The modeling approach is presented and described within the context of a laboratory study which allowed subjects freedom to determine strategies for using an autopilot in a multi-task environment. The results of this study motivated the present analysis as it was found that subjects did not use the autopilot in the manner in which its designers intended and predicted. Nevertheless, modeling and sensitivity analysis demonstrated that many subjects may indeed have been acting appropriately by using (and in some cases *not* using) the aid in unexpected ways. This aiding system was apparently not designed with due regard to critical features of the overall multi-task context. It is hoped that the present study contributes to aid designs which are not only technically sound, but also sensitive to features of the operational contexts in which they are deployed.

Experimental Task

The approach for modeling human-automation interaction was motivated by experiments on human performance in a laboratory simulation of a light helicopter supervisory control task. The task required either one-person or two-person crews to pilot a "scout" helicopter through a simulated 100 square-mile partially forested world to discover cargo and engage enemy craft during each 30 minut session. The experimental apparatus, configured for a two-person crew, is shown in Figure 1. Bo the map display (on the left) and the pilot's display (on the right) provided information useful for piloting the scout. The map display showed the entire world in a top-down format. The pilot's display showed a 100-degree pie-shaped visual angle of the terrain within 2000 feet in front of the scout. Both manual (joystick) and autopilot control of the scout helicopter were available.

The autopilot was activated by first entering a waypoint by positioning a cursor on the map display with the control stick shown in the center of Figure 1. Pressing a pushbutton then activated pathfinding mechanism that automatically avoided trees and guided the scout to the indicated waypoint. Multiple successive waypoints for the scout could also be programmed. The autopilot was available to the crews at all times but was inferior to maximally attainable manual control performance. The maximum speed of the the scout under autopilot control was 75% of the maximum speed under manual control. This speed restriction was partially due to computations necessary to perform on-line pathfinding through the forested world. In addition, the scout consumed fuel at a higher rate under autopilot control than under manual control. It was generally tc the crews' advantage to maintain scout movement and to conserve fuel. Thus, whether manual or automatic control was most effective was determined in part by the skill of the pilot to manually control the scout. It must be pointed out that the scout helicopter had inherently stable dynamics to make the piloting task managable for subjects without extensive training. Thus, a diversion of attention from the piloting task resulted only in the scout coasting to a stop at constant altitude.

Insert Figure 1 about here

Crews also had supervisory control over four "friendly" helicopters which were used to assist the scout in meeting mission goals. Friendly craft were commanded by constructing strings of action commands via a text editing terminal specially configured for the experiment, shown at the left side of Figure 1. Most of the text editing sessions dealt with low priority commands to the friendly craft, such as instructions to load cargo or search regions of the world. A minority of the editing sessions were of high priority due to unexpected emergency conditions that required immediate crew attention In these situations, delays in entering an appropriate command increased the probability of losing the use of a friendly craft for the remainder of the current mission.

In the one-person crew condition, one subject was responsible for controlling both the scout and the friendly craft. In the two-person condition, one subject, called the pilot, controlled the scout, while the second subject, called the navigator, controlled the four friendly craft. Crews were instructed to gain as many points as possible during each mission by engaging enemy craft and by finding, loading, and unloading cargo at home base. Experiments were performed to compare one-versus two-person crew performance to provide data to support psychological modeling. Five one-person and five two-person crews were used. Subjects were male young adults taken from a university population. For a more complete task description and a comparison of one- and two-person crew performance see (Kirlik, Plamondon, Lytton, Miller and Jagacinski, 1991), and for a process model of crew behavior see (Kirlik, Miller and Jagacinski, 1991). The present paper considers only aspects of the task and crew behavior related to autopilot use.

Strategies for Autopilot Use

The present analysis considers only data from the five one-person crews. In the two-person condition the pilot was dedicated to scout control and therefore did not need to divert attention from the piloting task to edit commands for the friendly craft. The one-person crew data are especially interesting because these crews often had to deal with simultanous demands for both scout and friendly craft control activity. The experimenters expected the autopilot to play an important role in allowing one-person crews to effectively cope with these simultaneous task demands. Specifically, i was expected that one-person crews would use the strategy of engaging the autopilot before text editing sessions and resuming manual control after editing. This strategy would allow the crews to exploit the superiority of manual control over automatic control (in terms of speed and fuel usage efficiencies) when there were no demands for text editing, but to also achieve the benefits of autopilo control when attention was diverted from manual control by the presence of a secondary task.

Contrary to expectation, none of the five crews used this strategy to select control modes. Two crews used the autopilot almost exclusively, one crew used manual control almost exclusively, and two crews used both manual and automatic modes to a substantial degree. Although these two crews alternated control modes, the selection of control mode was independent of demands for text editing friendly craft commands. These crews were no more likely to use automatic control when editing than when not editing. The rationale behind mode switching for these two crews cannot be clearly determined with the available data. However, informal observations suggested that manual control was more consistently used when tight navigation constraints were present. The resolution of the cursor on the map display did not always allow subjects to specify a waypoint near enough to the location of home base or a piece of cargo so that loading and unloading could be effectively performed. Other factors may have also contributed to these crews' mode selection strategies.

The most striking result was that no crew used the autopilot as a task-offload aid in the way its designers intended. As is the case with many human-machine interaction problems, in retrospect it is not difficult to hypothesize factors contributing to this finding. Most importantly, the time to program and engage the autopilot was relatively long with respect to the duration of the majority of text editing sessions. In addition, many editing sessions were of high criticality and could not be safely delayed while the autopilot was being engaged. Berg and Sheridan (1985) also found that subjects were reluctant to use an autopilot in an aircraft piloting simulation due to the fact that the time and effort necessary to engage the aiding system were not worth the benefits received.

Factors Affecting Strategy Selection

One modeling goal was to gain insight into why the autopilot in the laboratory task was not useful as a task-offload aid. The approach used was to describe the task environment in such a way that optimal policies for using the autopilot could be determined as a function of various design and task context parameters. The first step was to identify a number of aid design and task context factors that presumably influenced the strategies developed by crews for using the autopilot. Six factors, discussed below, appeared to influence control mode selection strategies in the present task. In other task environments it is quite possible that a different set of factors affect strategy development. Although the current analysis considers only design and context features relevant to the present experimental task, the modeling approach described below could be extended in various ways to capture other influences on strategic behavior.

The first three factors hypothesized to influence strategy selection in the laboratory task were autopilot design features. The first was the ability of the autopilot to control the scout relative to the crew's ability to successfully control the scout manually. Presumably, the autopilot would have been especially attractive for crews who could not control the scout as effectively as could the autopilot. A second factor was the autopilot engagement time. Between 5 and 10 seconds were required to specify a waypoint with the map cursor and activate the autopilot via pushbuttons. A third potential factor influencing autopilot usage was disengagement time. However, since the autoplilot could be disgengaged in approximately one second, this factor was not expected to strongly contribute towards understanding why crews did not use the autopilot as a task-offload aid.

The second three factors were features of the overall task context in which the autopilot was placed. The first was the duration of secondary tasks that required a diversion of attention from the piloting task. Editing sessions lasted between 5 and 15 seconds. Thus, the autopilot engagement time was between 33% and 200% of the total time in which the autopilot would be in use under the strategy of using the autopilot only during editing sessions. This relationship between secondary task duration and aid engagement time almost surely had a strong influence on crews' strategies for using (or not using) the autopilot. A second task context factor which may have affected mode selection was the high cost associated with delaying editing sessions. Delaying a critical editing session by the time spent engaging the autopilot increased the probability of losing the use of a friendly craft for the remainder of the experimental session. A final task context factor considered was the time between demands for secondary task activity. If the intervals between secondary tasks are short relative to the duration of secondary tasks, the attractiveness of the autopilot control could be expected to increase. Even though secondary task duration may be brief, if secondary task demands occur at a high frequency it may be appropriate to use purely automatic control. This strategy would eliminate the need for frequent autopilot engagement and disengagement, while maintaining scout motion during the many periods during which manual control would not be possible. The model described below was developed to investigate the role of the six factors identified above in influencing appropriate control mode strategy and resulting system performance.

Markov Decision Process Modeling

The semi-automated, multi-task human-machine system described above was modeled as a continuous time Markov Decision Process (MDP). A continuous time MDP consists of a set of states, state transition rates, a set of actions available in each state, and a payoff or reward rate associated with occupancy time in each state. Although the general MDP formulation also allows payoffs to be associated with the execution of actions and state transitions, the model described below associates rewards only with state occupancy durations. Use of a policy[2] (an action to be selected in each state) induces a Markov chain associated with the MDP. An optimal policy is

[2] In this paper, the term *policy* is used to refer to the mathematical entity describing the actions to be taken in each state. The term *strategy* is reserved to refer to the psychological construct describing how strategic behavior is mediated. The distinction is maintained to keep separate statements about the model and statements about the human operator. For example, although a given policy may be optimal in the model, the corresponding strategy may not be optimal due to model simplifications, measurement approximations, etc.

defined to be that set of actions, one per state, that when taken maximize the reward acheived over the lifetime of the process. Here we assume an infinite planning horizon. While system operating time is surely finite, the assumption of an infinite horizon only has the effect of defining a steady state policy which is unaffected by mission termination.

The goal of modeling the human-machine system as an MDP was to understand how the six aid design and task context features combine to determine optimal policies for using the autopilot. Associated with each policy is a strategy that could be potentially used by the human crews. The MDP model used here has four states and the identical set of two actions associated with each state. The actions are: 1) Select autopilot control; and 2) Select manual control. The states are: 1) Manua control -- No editing required; 2) Autopilot control -- No editing required; 3) Manual control -- Editing required; and 4) Autopilot control -- Editing required. Note that taking Action 1 in States 2 or 4, and taking Action 2 in States 1 or 3 are null actions which only serve to maintain the current control mode. A graphical representation of the MDP model is shown in Figure 2.

Note that the state space is the cross product of the set of two exogenous task states (editing required, no editing required) and the set of the results of the two actions (select manual control, select autopilot control). The inclusion of action related information into the state space was requirec to satisfy the Markov assumption that the current action decision can be based only upon the current state and not on the previous action or state. Since the decision to select autopilot or manual control almost surely depends on the current control mode, control mode must be included in the state space for the Markov assumption to be satisfied. A two-state model (Editing required, No editing requirec could be used under the condition that the action decision could depend on the previous action taken, thereby violating the Markov assumption. Achieving the analytical benefits associated with formulating this model as an MDP necessitated a larger state space.

Insert Figure 2 about here

The behavior of this MDP model with an associated decision policy is as follows. At each state transition, an action is selected according to the decision policy. The action selected determines the mean of the exponentially distributed state occupancy time and the next state which will be entered at the end of that time. In addition, the action specifies the payoff which is earned per unit time in the current state. When the transition to the new state occurs, this process is repeated based on the actior to be selected for the new state as given by the decision policy.

With the state space and action set as defined above, the only model components left to be specified are the state transition rates and payoffs as a function of the action selected in each state. The state transition diagram in Figure 2 and the transition rate and payoff information in Table 1 summarize this information. As mentioned above, the implementation of a policy in conjunction with an MDP induces a Markov chain. A description of a few possible chains will now be given to help explain the information given in Table 1 and the state transition diagram.

Insert Table 1 about here

Assume the process begins in State 1 (Manual control -- No editing required). If Action 1 (Select autopilot control) is selected, the process will continue to occupy State 1 for a mean duration of *TEngage* seconds (the mean time to engage the autopilot). An earning rate of zero in the table indicates that no payoff is earned during this interval. That is, until the autopilot takes control, the scout does not begin to move and search the simulated world, thereby contributing nothing to mission success during this interval. After a mean time of *TEngage* seconds, the process will transition to State 2 (Autopilot control -- No editing required). The process will occupy this state for a mean duration of *TNonEdit* seconds (the mean time between demands for editing). Assuming autopilot control is selected once again in this state (i.e., autopilot control is maintained), earnings of 1 unit per second are accrued due to the performance of the autopilot. After a mean time of *TNonEdit* seconds, the process will transition to State 4 (Autopilot control -- Editing required). Assuming a selection of autopilot control again, earnings of 1 unit per second continue to accrue during the editing session of mean length *TEdit* seconds due to the performance of the autopilot while the operator is editing. After a mean of *TEdit* seconds, the process will transition back to State 2 (Autopilot control -- No editing required).

Now, assume instead that Action 2 (Select manual control) was selected from the initial state (State 1: Manual control -- No editing required). For a mean duration of *TNonEdit* seconds, earnings of *M* units per second will accrue due to the manual control exerted by the operator. If *M* is greater than 1, the operator's performance at piloting the scout is superior to the performance of the autopilot; if *M* is less than 1, the operator's performance is inferior to the autopilot. After a mean of *TNonEdit* seconds, the process transitions to State 3 (Manual control -- Editing required). If manual control is selected (maintained), no earnings are accrued for the mean *TEdit* seconds in which the operator is editing and attention is diverted from the manual control task. If the operator had decided to switch to autopilot control when the editing session was required, a penalty of *P* units per seconc would be paid for the mean *TEngage* seconds delay before editing could begin. After a mean *TEngage* seconds, the process would transition from State 3 to State 4 (Autopilot control – Editing required). At this point, the process would once again begin to accrue the 1 unit per second payoff while the autopilot was operating and the operator was editing.

The three task environment factors (*TEdit*: mean editing session duration, *TNonEdit*: mean time between editing sessions, and *P*: the penalty for delaying an editing session) and the three autopilot design factors (*TEngage*: time to engage the autopilot, *TDisengage*: time to disengage the autopilot, and *M*: the relation of human control performance to autopilot control performance) combine to produce an optimal policy for control mode selection. Knowledge of the way that various values of these task and design parameters interact to produce different optimal policies for control mode selection and attainable levels of system performance would allow the partial specification of autopilot design criteria as a function of task context. The following section describes a sensitivity analysis of optimal policy and performance as a function of various levels of these factors.

Sensitivity Analysis Approach

A policy improvement algorithm (Howard, 1960) was used to identify the optimal decision policy and maximum level of expected performance as a function of the six model parameters mentioned above. A description of the algorithm used to solve the present problem and is provided in Appendix 1. The purpose of the sensitivity analysis was to identify how the optimal policy for mode selection and attainable system performance changes as a function of various levels of the six context and aid design parameters.

To keep the scope of the analysis manageable, three of the six parameters were chosen as the focus of the sensitivity analysis. These were: 1) *P* = the penalty for delaying text editing sessions; 2) *M* = manual control performance as a percentage of autopilot control performance; and 3) *TEngage* = the time to program and engage the autopilot. Thus, the three remaining factors were held constant throughout the analysis. These were: 1) *TEdit* = mean editing session duration (10 seconds); 2) *TNonEdit* = mean time between editing sessions (40 seconds); and 3) *TDisengage* = mean time required to disengage the autopilot (1 second).

The choice of the factors to be varied was partially motivated by the initial hypotheses about why crews did not use the autopilot as a task-offload aiding device. The primary hypotheses concerned the crews' manual control abilities, the relatively long autopilot engagement time, and the penalties associated with delaying editing sessions. Thus, the sensitivity analysis was designed to focus on how these three factors interact to produce alternate optimal policies. The values of the three fixed factors were selected to achieve reasonable agreement with the properties of the laboratory task environment.

The value of *M* (human manual control performance as a percent of autopilot control performance was varied between 50% and 250%. Lower levels of *M* indicate manual control performance worse than autopilot control, *M* = 100% indicates equality of performance, and high values of *M* indicate manual control performance superior to the autopilot. The theoretically highest value of *M* obtainable in the present experiment is near 200%. This would be earned by an operator capable of flying at full speed, as opposed to the autopilot ceiling of 75% with twice the fuel usage of manual control.

The value of *TEngage* (the mean time required to engage the autopilot) was varied from 0.5 seconds to 10.0 seconds. An approximate figure for the mean time to engage the autopilot in the laboratory task is 8 seconds. Values of *TEngage* much lower than this figure were used in the sensitivity analysis to identify by how much engagement time must have been reduced in order to have increased the attractiveness of the strategy of switching to autopilot control during editing sessions.

The value of *P* (the per second penalty for delaying editing sessions) was varied between 0.0 and 10.0. The unit of measurement of *P* is equivalent to the units on the autopilot and manual control earning rates. Thus, *P* = 5.0 indicates that the cost per second of delaying an editing session is five times as great as the reward per second of autopilot control of the helicopter. As could be expected, *P* could not be precisely measured in the laboratory task. However, it did seem clear that there was a sense in which the costs of delaying editing sessions did trade off with the costs of scout control. In this paper, though, *P* is perhaps best thought of as an ordinal measure of editing session criticality. Issues related to the measurement problems underlying the present modeling approach are discussed in the final section of this paper.

Modeling Results

The results of the sensitivity analysis are graphically presented in Figures 3, 4 , 5 and 6. Figures 3, 5 and 6 depict two-way policy sensitivity analyses as a function of autopilot setup time and manual control performance. Each of these two-way analyses was performed at a different level of *P*, the per second penalty of delaying editing sessions by the time required to engage the autopilot. Figure 4 shows the system performance that would result from using the optimal policies shown in Figure 3.

The results corresponding to $P = 0.0$ (no penalty for delaying editing sessions) appear in Figures 3 and 4. A P value of zero would represent editing sessions which are entirely self paced and no costs are associated with delays. An example of such a session would be the preparation of a action command which would not be executed by the friendly craft until some later time. Figure 3 shows the optimal mode selection policy as a function of autopilot setup time (horizontal axis) and manual control payoff as a percent of autopilot control payoff (vertical axis). Four mode control policies were identified by solving the MDP via the method discussed in the appendix: 1) Always use the autopilot; 2) Always use manual control; 3) Use manual control from session initiation until the first editing session is required, then switch to the autopilot for the rest of the session; and 4) Use the autopilot when editing is required and use manual control otherwise. Recall from the experimental results that no subjects used strategies corresponding to Policies 3 or 4.

Insert Figures 3 and 4 about here

Policy 1 (always use the autopilot) is always optimal if autopilot control is superior or equal to manual control regardless of the autopilot setup delay or editing session criticality. This should come as no surprise for there is no advantage to using manual control in this situation. Figure 4 indicates that this policy will earn an average of 1 unit per second at steady state.

Policy 4 (use the autopilot when editing, manual control otherwise) is optimal if manual control is superior to autopilot control and the penalty for editor delay is small. When manual control is superior to autopilot control and editor delay is large, Policy 2 (always use manual mode) is optimal. Over intermediate levels of autopilot engagement time, engagement time and manual control performance interact to determine the optimal policy. Qualitatively, Policy 2 is optimal for moderate engagement times (4-6 seconds) only if manual control performance is excellent (greater than 180% of autopilot performance). For these same moderate engagement times, Policy 4 is optimal if manual control is in the intermediate range (between 110% to 170% of autopilot control performance).

Thus, high levels of manual control performance and long autopilot programming and engagement times combine to make pure manual control an attractive strategy. It should be remembered that under this policy no earnings accrue while the operator is using the text editor. The cost associated with long engagement times in this case can be found by reference to the Figure 4. If manual control performance is 140% as great as automatic control performance, the optimal policy depends upon autopilot setup time. If setup time is one second, the optimal policy is to engage the autopilot when editing is required and use manual control otherwise. This policy has a steady state earning rate of 1.3 units per second. If autopilot engagement time is greater than seven seconds, the optimal policy is to always use manual control. The steady state earning rate of this policy is 1.1 units per second. Therefore, the 6 second increase in autopilot engagement time results in an approximate 15% decrease in overall system performance.

Higher criticalities for editing session delays (Figures 5 and 6) serve to decrease the size of the parameter space in which Strategy 4 is optimal. In the highest criticality case ($P = 5$, Figure 6), the only case in which a switching strategy is optimal is if the autopilot setup time is one second or less. At this level of editing session criticality, virtually the only two optimal policies are pure autopilot control and pure manual control.

Insert Figures 5 and 6 about here

The results of this sensitivity analysis suggest that the potential benefits of a given task-offload ai can only be predicted with knowledge of the overall task context in which the aid is deployed. High levels of secondary task criticality were typical of the supervisory control task upon which this analysis is based. The results of the sensitivity analysis at high levels of editing criticality indicate that three crews, by using strictly autopilot or strictly manual control, may indeed have been acting appropriately given the poor system design with which they were confronted. In addition, recall that two other crews used both automatic and manual control, but mode selection was independent of secondary task demands. Instead, it was observed that these two crews may have intermittently engaged manual control because they could more successfully perform target acquisition by manually piloting the scout than they could by using the autopilot. The results of the sensitivity analysis are consistent with this observation. During normal piloting activity, crews may have recognized that autopilot control was superior to their own manual control abilities, especially given the need to edit friendly craft commands. Given the autopilot engagement time and the criticality of secondary tasks in the present experiments, pure autopilot control would be most appropriate under such conditions. During more demanding target acquisition, on the other hand, crews may have recognized that they could more efficiently navigate to targets using manual control than they could by attempting to specify an autopilot waypoint with the low-resolution map cursor. Under such conditions, (manual control performance greater than autopilot performance), the sensitivity analysis results indicate that pure manual control is most appropriate.

Discussion

By showing how aid design and task context factors combined in complex ways to influence appropriate strategies for human-automation interaction, the modeling and sensitivity analysis provided valuable insights into why none of the five crews used the autopilot as a task-offload aid in the manner intended and expected. It is hoped that the research reported here motivates those involved with introducing automation into human-machine systems to give explicit consideration to understanding how operators may strategically manage their interaction with aiding systems in an effort to keep workload and performance at acceptable levels. Perhaps most importantly, the present research emphasizes the need to appreciate the critical role played by the overall task context in which aids are deployed. High levels of technical performance and reliability are surely necessary attributes of any automated system considered for the operational environment. These properties are far from sufficient, however, since it is only through the operator's strategy for managing automation that the potential benefits of aiding systems are realized in system performance.

In closing, it is of interest to discuss a number of methodological issues pertinent to both evaluating the present research and to extending it to operational environments. Of perhaps greatest importance are issues concerning the appropriateness of the style of modeling presented (normative), and the measurement problems that would be confronted if this modeling approach were to be applied to a more complex task environment than the one studied here.

One criticism sometimes levied against normative modeling, and often rightly so, is that we require knowledge of what operators *will* do, rather than what they *should* do. However, in many situations, and certainly those in which descriptive models are lacking, an understanding of normative behavior can be an important first step toward the prediction of actual behavior. In many human-machine systems, operators gain skill over an extended period of time through an ongoing process of productive adaptation to environmental and goal structures. This long term adaptation process, rather than any presumed ability for optimality-seeking decision-making, is the mechanism that often brings skilled behavior in line with task goals. Note that the present model was specifically *not* advanced as a process account of strategy development or use. Rather, the aim was to identify what behaviors would be exhibited if operators had become productively adapted to their task environments. Knowledge of limitations in human adaptivity, and thereby knowledge of expected behavioral deviations from optimality, is surely necessary for the development of descriptive models capable of accurate performance prediction. However, it is difficult to make the argument that aids should be designed in such a way that they do *not* assist the effectively adapted operator, as was the case with the aid design in the present experiment. Normative modeling helped to understand why unexpected behavior was observed. A more effective autopilot design would almost surely have resulted if modeling and sensitivity analysis were conducted prior to system design.

A second, and related, methodological issue central to the appropriateness of the current modeling approach concerns the measurement problems associated with parameter identification. The style of modeling used here required numerical estimation of a number of factors that were difficult to quantify. The penalty for delaying secondary tasks is one such example. It could be expected that such measurement problems would only multiply if the present modeling approach was applied to an existing operational context, such as an aircraft flight deck or air traffic control. One potential criticism of the present approach is that it may require quantification of subjective factors such as task criticality and payoff rates for various control activities. However, this criticism only takes force if alternative, qualitative methods are available which do justice to the apparent richness of subjective assessment but still provide explicit and defensible techniques for predicting how a complex set of factors will combine to influence behavior. Such tools are currently in short supply. As a result, these predictions are typically left to either designers' intuitions or to costly, high-fidelity simulation.

It is our observation that the problems involved in describing and measuring environmental complexity contribute as much, and in some cases more, to our inability to predict skilled behavior in complex systems as does our lack of knowledge of the psychological mechanisms involved. Even if we had the ideal models of the relevant psychological processes, this knowledge could not be applied to performance prediction without equally good models of the environmental structures to which skilled behavior must be sensitive. The research reported here described a method for environmental description that allowed predictions of one type of behavior to be made in a relatively straightforward fashion. Other techniques for describing task environments are surely necessary to capture other influences on behavior. Successful performance prediction requires techniques for environmental modeling that are just as rich, precise, and formal as the techniques used to model the human operator. Until the time when such environmental models are available, the most economical representation of operational environments will continue to be high-fidelity simulations, or perhaps even the operational environments themselves. We suggest that the problem of measuring and describing environmental complexity deserves increased attention by those involved with applying psychological principles to the design of complex work environments.

Acknowledgments

This research was supported by NASA Ames Grant NAG2-656 to Georgia Insitute of Technology, Robert J. Shively, technical monitor, and by NASA Ames Grant NAG2-195 to Ohio State University, E. James Hartzell, technical monitor. The author thanks R. A. Miller, Richard J. Jagacinski, Brian Plamondon, and Lynn Lytton for their contributions to the research program upon with this work is based. The author also thanks the organizers and attendees of the NASA Ames Workshop on Strategic Behavior and Workload Management held in March 1991 at Big Sur, California, for stimulating discussions on many of the issues presented in this paper.

References

1. Berg, S. L. and Sheridan, T. B. (1985). The impact of physical and mental tasks on pilot mental workload. *Proceedings of the Twenty-First Annual Conference on Manual Control,* Columbus, Ohio.

2. Cinlar, E. (1975). *Introduction to Stochastic Processes.* Englewood Cliffs, NJ: Prentice-Hall.

3. Heymann, D. P., and Sobel, M. J. (1984). *Stochastic Models in Operations Research: Volume II: Stochastic Optimization.* New York: McGraw-Hill.

4. Howard, R. A. (1960). *Dynamic Programming and Markov Processes.* New York: John Wiley & Sons. (Esp. Chapter 8)

5. Kirlik, A., Plamondon, B. D., Lytton, L., Jagacinski, R. J., and Miller, R. A. (1991). Supervisory control in a dynamic and uncertain environment II: Comparisons of one and two-person crews. Manuscript submitted for publication.

6. Kirlik, A., Miller, R. A., and Jagacinski, R. J. (1991). Supervisory control in a dynamic and uncertain environment II: A process model of skilled human-environment interaction. Manuscript submitted for publication.

7. Moray, N. (1986). Monitoring behavior and supervisory control. In K. Boff, L. Kaufman, and J. P. Thomas, Eds., *Handbook of Perception and Human Performance Volume 2.* New York: John Wiley and Sons.

8. Navon, D. and Gopher, D. (1979). On the economy of the human processing system. *Psychological Review, 86,* (pp. 254-255).

9. Senders, J. (1964). The human operator as a monitor and controller of multidegree of freedom systems. *IEEE Transactions on Human Factors in Electronics,* HFE-5. (pp. 2-6).

10. Wickens, C. D. (1984). Multiple resources in attention. In R. Parasuraman and R. Davies, (Eds.), *Varieties of Attention.* New York: Academic Press.

11. Wickens, C. D. (1987). Attention. In P. Hancock, (Ed.), *Human Factors Psychology.* Amsterdam: North Holland Publishing Company.

Figure Captions

Figure 1. Experimental apparatus configured for a two-person crew. For the one-person crew the pilot's workstation and the map display were moved toward the editor and the button panel in front of the map display was removed.

Figure 2. State transition diagrams for the MDP model. The network at the top indicates the state transitions caused by selecting autopilot control upon entering each state. The lower network shows the state transitions caused by selecting manual control upon entering each state. Transition rates shown by the parameters on each arc represent the mean time before the indicated state transition occurs.

Figure 3. Sensitivity analysis results for $P = 0$. P is the per second penalty for delaying editing sessions, *TEngage* is mean autopilot engagement time and M is manual control performance as a percent of autopilot performance. The entry in each cell shows the optimal policy for managing control modes given the associated parameter values.

Figure 4. Sensitivity analysis results for $P = 0$ showing the levels of system performance that would result from using the optimal policies shown in Figure 3.

Figure 5. Sensitivity analysis results for $P = 2$. See Figure 3 caption for explanation.

Figure 6. Sensitivity analysis results for $P = 5$. See Figure 3 caption for explanation.

Appendix 1

The method used to formulate and solve the Markov Decision Process presented in this paper is provided below. The policy improvement algorithm of Howard (1960) is described. In addition, a number of terms and concepts from Cinlar (1975) and are used.

Specifying the Markov Decision Process

The Markov Decision Process (MDP) is defined with a set of states, S_i, $(i = 1, N)$ and a set of actions (or decisions) available in each state. In our example, the states are defined as follows:

S_1 = Manual control mode, no editing required
S_2 = Automatic control mode, no editing required
S_3 = Manual control mode, editing required
S_4 = Automatic control mode, editing required

and the decisions available in each state S_i are denoted d_i where:

$d_i = 1$ if autopilot control is selected in S_i, and
$d_i = 2$ if manual control is selected in S_i

We then define a decision policy D to be the vector (d_1, d_2, d_3, d_4) indicating the action to be taken in each state. For example, $D = (1,1,2,2)$ is the policy of selecting autopilot control in States 1 and 2 and manual control in States 3 and 4. For notational convenience, let us also write this decision policy as $D_{d1,d2,d3,d4}$. Thus, the example policy above becomes $D_{1,1,2,2}$.

Associated with each decision policy are a state tranisition matrix Q, a transition rate matrix A, and an earnings or reward vector r. The state transition matrix Q indicates the probability that the process will move to S_j at the next state transition given the process is in S_i under the indicated decision policy. For example, the diagram at the top of Figure 2 indicates the state transitions assoicated with the policy $D_{1,1,1,1}$ (selecting autopilot control in each state). As indicated in the diagram, the state transition matrix associated with this policy is:

$$Q_{1,1,1,1} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 \end{bmatrix}$$

Thus, under policy $D_{1,1,1,1}$, State 2 always follows State 1, State 4 always follows State 2, State 4 always follows State 3, and State 2 always follows State 4.

Use of a given policy also determines the transition rate matrix A, which specifies the rates with which the system moves from state to state under the policy. Table 1 provides the transition rate information with which we construct the transition matrix for a given policy. Using the same policy as before, $D_{1,1,1,1}$, for example, the transition matrix A is constructed from the the rows in Table 1 associated with the selection of automatic mode in each state. The entries of the A matrix are simply the *inverses* of the mean state occupancy times given in the table (see below). (Note: For simplicity of presentation Table 1 uses zeros to represent transition rates that can be considered to have infinite values because these transitions cannot occur - inverses of these values result in zero entries in the A matrix as seen in the example below). For example, letting *TEngage* = 10s, *TNonEdit* = 40s, and *TEdit* = 8s, the transition matrix under this policy is:

$$A_{1,1,1,1} = \begin{bmatrix} \frac{-1}{10} & \frac{1}{10} & 0 & 0 \\ 0 & \frac{-1}{40} & 0 & \frac{1}{40} \\ 0 & 0 & \frac{-1}{10} & \frac{1}{10} \\ 0 & \frac{1}{8} & 0 & \frac{-1}{8} \end{bmatrix}$$

One can interpret the off-diagonal entries, a_{ij}, in the A matrix as follows. In a brief time interval dt a process currently in State i will move to State j with probability $a_{ij}\,dt$ $(i \neq j)$. Thus, the times between transitions from State i and State j are exponentially distributed random variables with parameter a_{ij}. The expected value of these times is given by $\frac{1}{a_{ij}}$ from the properties of the exponential distribution. In addition, we see that the times of state transitions define a Poisson process. The diagnonal elements a_{jj} are defined as:

$$a_{jj} = \sum_{i \neq j}^{N} a_{ji}$$

Thus, each of the rows of A sum to zero, as is the case with our example presented above. The transition rate matrix A is also termed the generator matrix associated with the Markov process (Cinlar, p. 256).

Finally, an earning or reward vector is also associated with each decision policy. The model used in this paper only associated rewards with state occupancy durations, although a formulation allowing rewards to also be associated with state transitions is also available (e.g., see Howard, pp. 104-106). For the current problem, the rightmost column in Table 1 indicates the reward rates (earnings per unit time) associated with taking each action in each state. Using the same policy as before, the relevant information is once again composed of the rows in the table associated with the selection of automatic mode in each state. Assuming *P* (the per second penalty for delaying editing sessions) is 5, the reward vector is:

$$r_{1,1,1,1} = \begin{bmatrix} 0 \\ 1 \\ -5 \\ 1 \end{bmatrix}$$

Definition of the states, action sets available per state, state transition and transition rate matrices for each policy, and the reward vectors completely specifies the Markov Decision Process.

Recurrent Chains and Ergodic Policies

We require a few additional concepts before describing the method used to solve for the optimal decision policy.

A *recurrent chain* of a Markov process is a set of states connected by possible transitions such that the process moves from state to state within this set, but never moves outside this set (Howard, p. 13). A Markov process may have only one recurrent chain and is thus called a *single-chain* process. In this case there is some non-zero probability that the process will occupy any of the N states as time approaches infinity. In other cases, however, after sufficient time the process may make transitions only within certain subsets of states. Once it has entered one of these subsets, or recurrent chains, it is forever trapped there. If there are more than one such subsets of states, the process is a *multi-chain* Markov process. For example, the Markov process defined by using policy $D_{1,1,1,1}$ in our example is a single-chain process. As can be seen by the top network in Figure 2, over time this process will visit each of the four states with non-zero probability.

Consider, however, policy $D_{2,1,2,1}$. Under this policy, manual mode is selected (maintained) in each state in which manual mode is already active, and automatic mode is selected (maintained) in each state in which automatic mode is already active. This policy gives rise to two recurrent chains. If the process starts in States 1 or 3 it will forever continue to occupy one of these two states, and if the process starts in States 2 or 4 it will only make transitions between these two states. Thus, depending upon the policy selected, the present model has the capacity to exhibit both single-chain and multi-chain behavior. It is important to recognize the potential for multi-chain behavior because multi-chain decision processes require slightly more complex solution methods than do single-chain processes, as will be seen below.

An *ergodic policy* is one that defines a single recurrent chain. Policy $D_{1,1,1,1}$ is therefore an ergodic policy whereas policy $D_{2,1,2,1}$ is not. The solution method presented below can be used for processes which have both ergodic and non-ergodic policies. A slightly simpler solution method capable of handling processes with only ergodic policies can be found in Howard (pp. 109-110).

Solving for an Optimal Policy

The goal is to find the policy D^* which maximizes the reward received over the lifetime of the process. For the present problem, there are only $2^4 = 16$ different policies, so a fully enumerative solution method would not be expensive. However, problems with large state and action sets or the need for extensive sensitivity analysis may require more efficient solution methods. Below we describe the policy improvement algorithm. For a derivation of this algorithm and a proof of its convergence properties see Howard (Chapter 8).

Define $v_i(t)$ to be the expected total reward earned by a given policy over a time t given the process starts in State i. Also let g_i be the *gain* of a given policy, which indicates the average reward earned by the policy per unit time. For large t,

$$v_i(t) = tg_i + v_i$$

That is, the expected total reward earned in time t, given an initial State i, is composed of two separable contributions. The first, tg_i, represents the average per-unit-time earnings of the policy, while the second, v_i, represents earnings due specifically to starting the process in State i. The reason the gain term is indexed by initial state is that the different recurrent chains may have different average reward rates. Within a given recurrent chain, however, the values of all the g_i's for the states within the chain will be equal, and thus independent of initial state.

The evaluation of a given policy is accomplished by solving the following pair of systems of equations:

$$\sum_{j=1}^{N} a_{ij}g_j = 0$$

$$g_i = r_i + \sum_{j=1}^{N} a_{ij}v_j$$

for $i = 1$ to N, and by setting the value of one v_i in each recurrent chain to zero. The variables a_{ij} and r_i are as defined above. This calculation is called the policy evaluation step of the solution process. Solving for each of the g_i's and v_j's in these equations determines the expected total reward of the policy specified by the a_{ij}'s and the r_i's. One of the v_j's is set to zero in each recurrent chain so that this underdetermined system of linear equations can be solved. This step is appropriate since we are only concerned with *relative* values of the v_j's.

To begin the algorithm, we simply choose any policy to serve as an intial guess and solve the equations above to determine the value of this policy. Once values for the g_i's and v_j's are found for this intial policy, we can then enter the policy improvement step of the solution process. To do so, for each state i, find the decision or action k that maximizes:

$$\sum_{j=1}^{N} a_{ij}^{k} g_j$$

and make this action the new decision in State i. If the above quantity results in ties, the new action must be selected on the basis of the relative values, v_j's, rather than on the basis of the gains alone. In such a case, find the decision or action k in State i that maximizes:

$$r_i^k + \sum_{j=1}^{N} a_{ij}^{k} v_j$$

and make this action the new decision in State i. If the new set of actions selected according to this process are identical with the actions of the previous policy, the algorithm terminates and the final set of actions is the optimal policy. If any action was changed during the policy improvement step, re-enter the policy evaluation step again and recalculate the v_i's and g_j's. The process then cycles until convergence is found.

Checking the Distributional Assumptions

The approach described above assumes that state occupancy times are random variables that can be approximately described by the exponential distribution. If this assumption is unreasonable for a given application, extensions to the present approach are available that allow occupancy times to take on arbitrary distributions (e.g., see Heyman and Sobel, 1984). In most cases, though, these extensions result in the need for more complex analytical and computational solution methods. In many cases, though, occupancy times can be approximated by the exponential distribution, and in cases where this is not appropriate simulation may well be the most efficient solution technique.

Given one would like to apply the approach presented in this paper, a technique is needed to determine the degree to which the exponential assumption holds. Cinlar's Theorem 5.21 (p. 266) can be used to generate a very simple method for this purpose.

Assume one is observing a human operator performing a sequence of tasks, and the goal is to determine whether the task durations can be approximated with the exponential distribution. Generate a series of sampling times, T_0, T_1, T_3, ... T_N in such a way that the lengths of the intervals T_1 - T_0, T_2 - T_1, ..., T_N - T_{N-1} are exponentially distributed random variables with a fixed parameter. This process can be performed quite simply with the aid of a pseudo-random number generator. Observe the operator until time T_N, and record the number of occurences of the operator performing each of the different tasks at each time T_i. Afterward calculate the fraction of occurences the operator was observed performing for each task. In addition, record the percentage of time the operator was performing each of the tasks during the entire sampling interval. If the fraction of occurences of the operator performing a given task when observed at the sampling times is approximately equal to the percentage of time the operator was performing that task over the duration of the entire sampling interval, then the assumption of exponentially distributed task durations is probably reasonable.

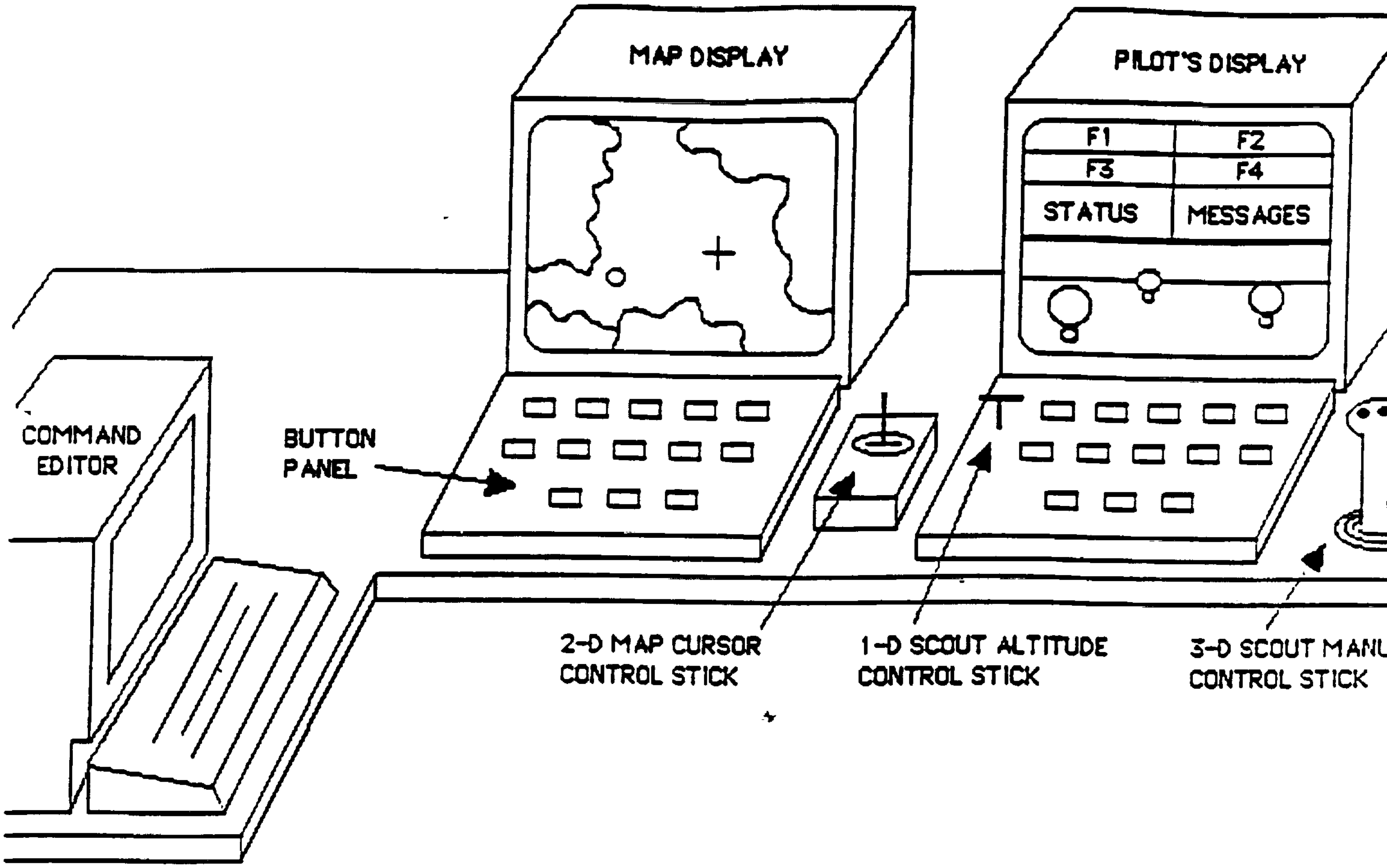

MAP DISPLAY
PILOT'S DISPLAY
F1
F2
F3
F4
STATUS
MESSAGES
COMMAND EDITOR
BUTTON PANEL
2-D MAP CURSOR CONTROL STICK
1-D SCOUT ALTITUDE CONTROL STICK
3-D SCOUT MANU
CONTROL STICK

Figure 1

Transition Rates for Choosing Action 1 in Each State
(Select Automatic Control Mode)

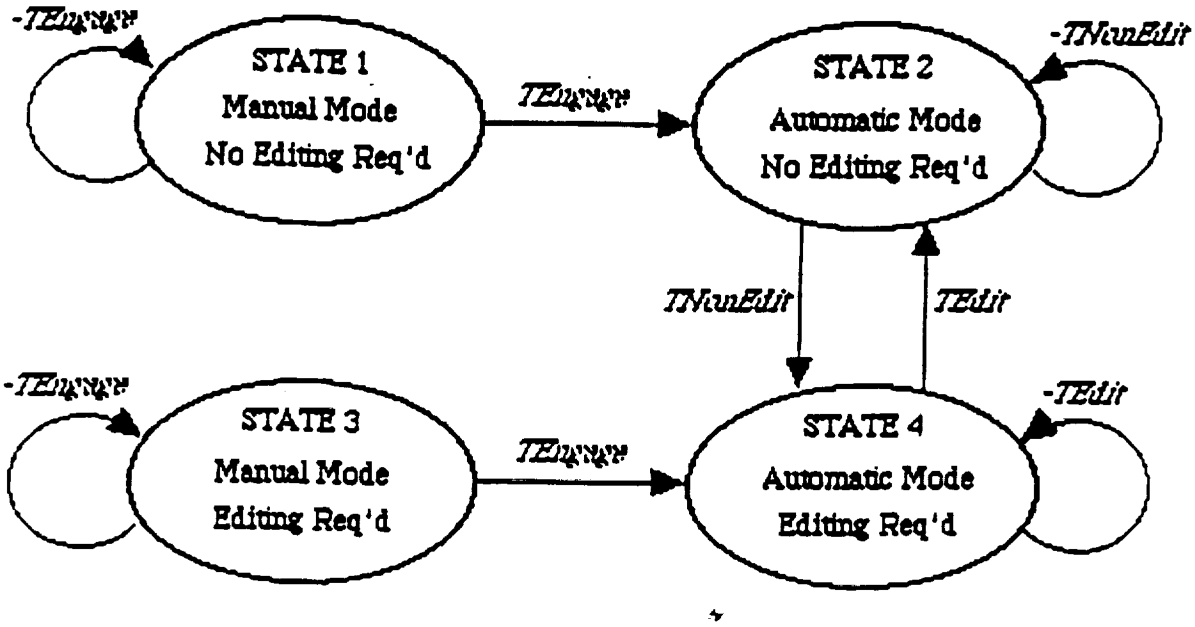
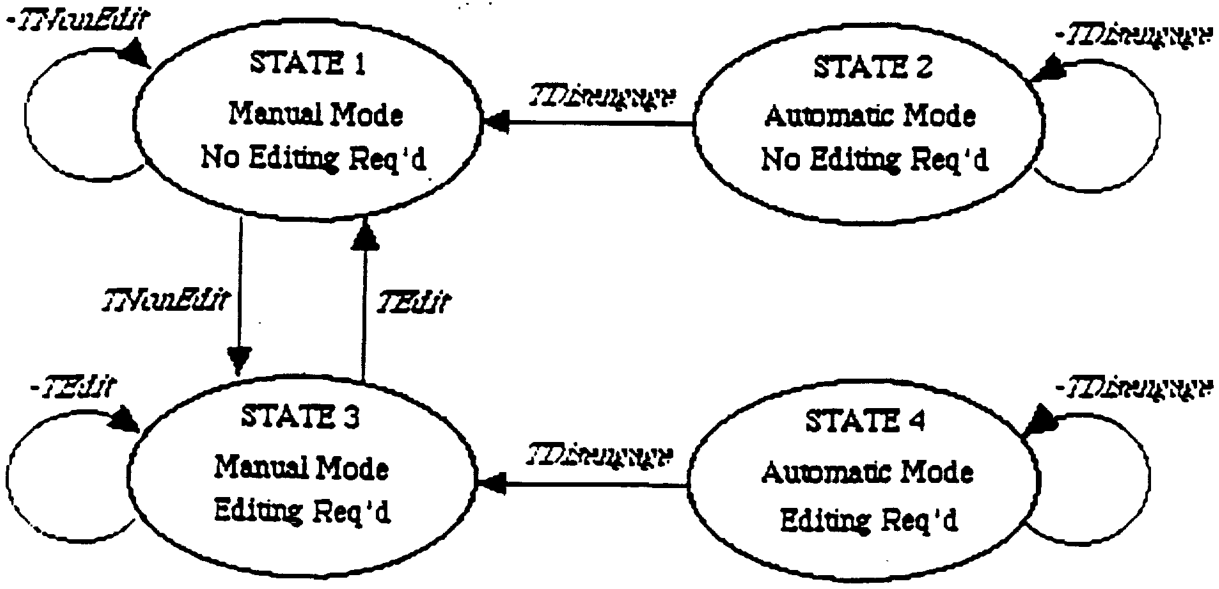


Figure 2

Transition Rates for Choosing Action 2 in Each State
(Select Manual Control Mode)



Parameter	Definition
TEngage	Mean time to engage automatic control mode
TDisengage	Mean time to disengage automatic control mode
TEdit	Mean duration of editing sessions
TNonEdit	Mean time between editing sessions

TABLE 1

STATE TRANSITION AND EARNING RATES AS A FUNCTION OF ACTION SELECTED IN EACH STATE

i	State Name	Action Selected	Transition Rate to State *j*				Earning Rate
			j =1	*j* =2	*j* =3	*j* =4	
1	Manual Mode No Editing Req'd	1. Select Auto	-*TEngage*	*TEngage*	0	0	0
		2. Select Manual	-*TNonEdit*	0	*TNonEdit*	0	*M*
2	Automatic Mode No Editing Req'd	1. Select Auto	0	-*TNonEdit*	0	*TNonEdit*	1
		2. Select Manual	*TDisengage*	-*TDisengage*	0	0	0
3	Manual Mode Editing Req'd	1. Select Auto	0	0	-*TEngage*	*TEngage*	*P*
		2. Select Manual	*TEdit*	0	-*TEdit*	0	0
4	Automatic Mode Editing Req'd	1. Select Auto	0	*TEdit*	0	-*TEdit*	1
		2. Select Manual	0	0	*TDisengage*	-*TDisengage*	0

Parameter	Definition
TEngage	Mean time to engage automatic control mode
TDisengage	Mean time to disengage automatic control mode
TEdit	Mean duration of editing sessions
TNonEdit	Mean time between editing sessions
M	Ratio of Manual to Automatic Performance
P	Cost/Sec of Delaying an Editing Session

***P* = 0**

***TEngage* (secs)**

M		1		2		3		4		5		6		7		8		9		10
50%	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
60%	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
70%	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
80%	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
90%	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
100%	1	1	1	1	1	3	1	1	1	1	1	1	1	1	1	3	1	1	1	1
110%	4	4	4	4	4	4	3	3	3	3	3	3	3	3	3	3	3	3	3	3
120%	4	4	4	4	4	4	4	4	4	4	4	4	4	3	3	3	3	3	3	3
130%	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	2	2	2
140%	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	2	2	2	2	2
150%	4	4	4	4	4	4	4	4	4	4	4	4	4	4	2	2	2	2	2	2
160%	4	4	4	4	4	4	4	4	4	4	4	4	4	2	2	2	2	2	2	2
170%	4	4	4	4	4	4	4	4	4	4	4	4	2	2	2	2	2	2	2	2
180%	4	4	4	4	4	4	4	4	4	4	4	2	2	2	2	2	2	2	2	2
190%	4	4	4	4	4	4	4	4	4	4	4	2	2	2	2	2	2	2	2	2
200%	4	4	4	4	4	4	4	4	4	4	2	2	2	2	2	2	2	2	2	2

Policy	Definition
1	Always use autopilot control
2	Always use manual control
3	Use manual before first editing, then autopilot
4	Use autopilot when editing, otherwise manual

Figure 3

***P* = 0**

***TEngage* (secs)**

			1		2		3		4		5		6		7		8		9		10
M	50%	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0
	60%	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0
	70%	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0
	80%	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0
	90%	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0
	100%	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0
	110%	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0
	120%	1.1	1.1	1.1	1.1	1.1	1.1	1.1	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0
	130%	1.2	1.2	1.2	1.2	1.2	1.1	1.1	1.1	1.1	1.1	1.1	1.1	1.1	1.1	1.1	1.1	1.0	1.0	1.0	1.0
	140%	1.3	1.3	1.3	1.2	1.2	1.2	1.2	1.2	1.2	1.2	1.2	1.2	1.1	1.1	1.1	1.1	1.1	1.1	1.1	1.1
	150%	1.4	1.3	1.3	1.3	1.3	1.3	1.3	1.3	1.3	1.3	1.2	1.2	1.2	1.2	1.2	1.2	1.2	1.2	1.2	1.2
	160%	1.4	1.4	1.4	1.4	1.4	1.4	1.4	1.3	1.3	1.3	1.3	1.3	1.3	1.3	1.3	1.3	1.3	1.3	1.3	1.3
	170%	1.5	1.5	1.5	1.5	1.5	1.4	1.4	1.4	1.4	1.4	1.4	1.4	1.4	1.4	1.4	1.4	1.4	1.4	1.4	1.4
	180%	1.6	1.6	1.6	1.6	1.5	1.5	1.5	1.5	1.5	1.5	1.5	1.5	1.4	1.4	1.4	1.4	1.4	1.4	1.4	1.4
	190%	1.7	1.7	1.6	1.6	1.6	1.6	1.6	1.6	1.6	1.5	1.5	1.5	1.5	1.5	1.5	1.5	1.5	1.5	1.5	1.5
	200%	1.7	1.7	1.7	1.7	1.7	1.7	1.7	1.7	1.6	1.6	1.6	1.6	1.6	1.6	1.6	1.6	1.6	1.6	1.6	1.6

Figure 4

P = 2

TEngage (secs)

M		1		2		3		4		5		6		7		8		9		10
50%	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
60%	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
70%	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
80%	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
90%	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
100%	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
110%	4	4	3	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
120%	4	4	4	4	3	3	3	3	1	1	1	1	1	1	1	1	1	1	1	1
130%	4	4	4	4	4	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2
140%	4	4	4	4	4	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2
150%	4	4	4	4	4	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2
160%	4	4	4	4	4	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2
170%	4	4	4	4	4	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2
180%	4	4	4	4	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2
190%	4	4	4	4	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2
200%	4	4	4	4	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2

Policy	Definition
1	Always use autopilot control
2	Always use manual control
3	Use manual before first editing, then autopilot
4	Use autopilot when editing, otherwise manual

Figure 5

P = 5

TEngage (secs)

M		1		2		3		4		5		6		7		8		9		10
50%	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
60%	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
70%	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
80%	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
90%	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
100%	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
110%	3	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
120%	4	4	3	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
130%	4	4	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2
140%	4	4	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2
150%	4	4	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2
160%	4	4	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2
170%	4	4	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2
180%	4	4	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2
190%	4	4	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2
200%	4	4	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2

Policy	Definition
1	Always use autopilot control
2	Always use manual control
3	Use manual before first editing, then autopilot
4	Use autopilot when editing, otherwise manual

Figure 6