N79-17494

ABSTRACT

Supervisory Dynamic Decision-Making in Multi-Task Monitoring and Control

by
M. K. Tulga and T. B. Sheridan
Massachusetts Institute of Technology

As computers become smaller, cheaper and more sophisticated, the tasks of the human pilot and ground controller are changing radically from that of being a continuous controller in one or a few control loops to that of being a monitor of many separate tasks, or a supervisory-coordinator of semi-automated subsystems. Human operators of nuclear and other process control plants are undergoing a similar change in role. Quantitative models by which to describe and predict behavior are lacking, however, and therefore need to be developed. This paper describes preliminary research in one such modeling effort.

In the experimental paradigm, a number of "tasks" are represented simultaneously on a computer display as rectangles of varying heights (representing relative "value density" of given tasks) and widths (representing task "durations"). Tasks appear at random times and places and move at fixed velocity toward a "deadline" at the right-hand margin. The subject's objective is to "attend" to one task at a time (hold a cursor, by means of a graphics tablet, successively to different rectangles) and thus cause that task's width to collapse at a uniform rate, hopefully to disappear before the deadline is reached and the opportunity time therefore lost. The reward earned is the aggregate of reduction in areas of all tasks. "Tasks" may be clustered in groups with additional delays imposed for switching the cursor from one group to another. The subject obviously tries to concentrate on the most rewarding tasks, but may lose time by changing the cursor to unimportant tasks or by shifting to a different task group. The experiment is implemented on an Imlac graphics terminal coupled to an Interdata display, and provides a wide variety of parameter combinations.

An optimal decision control model has been developed, which is based primarily on a dynamic programming algorithm which looks at all the available task possibilities, charts an optimal trajectory, and commits itself to do the first step (i.e., follow the optimal trajectory during the next time period), and then iterates the calculation. A Bayesian estimator has also been included which estimates the tasks which might occur in the immediate future and provides this information to the dynamic programming routine.

Preliminary trials comparing the human subject's performance to that of the optimal model show a great similarity, but indicate that the human skips certain movements which require quick change in strategy. We are planning to rerun the model under a variety of parameter combinations (principally, rate of discounting future data and/or preview span, and response time delay) to solve the inverse optimal problem. That is, for what parameters do the human data and the optimal model match?

Further, we will attempt to show how the experimental paradigm (and model) correspond to future flight management tasks.

MODELING HUMAN DECISION MAKING BEHAVIOR IN SUPERVISORY CONTROL +

M. K. Tulga
Department of Mechanical Engineering
Massachusetts Institute of Technology
Cambridge, Mass. 02139 U.S.A.

T. B. Sheridan
Department of Mechanical Engineering
Massachusetts Institute of Technology
Cambridge, Mass. 02139 U.S.A.

As computers become smaller, cheaper and more sophisticated, the tasks of the human pilot and ground controller are changing radically from that of being a continuous controller in one or a few control loops to that of being a monitor of many separate tasks, or a supervisory-coordinator of semi-automated subsystems. Human operators of nuclear and other process control plants are undergoing a similar change in role. Quantitative models by which to describe and predict behavior are lacking, however, and therefore need to be developed. This paper describes preliminary research in one such modeling effort.

In the experimental paradigm, a number of "tasks" are represented simultaneously on a computer display as rectangles of varying heights (representing relative "value density" of given tasks) and widths (representing task "durations"). Tasks appear at random times and places and move at fixed velocity toward a "deadline" at the right-hand margin. The subject's objective is to "attend" to one task at a time (hold a cursor, by means of a graphics tablet, successively to different rectangles) and thus cause that task's width to collapse at a uniform rate, hopefully to disappear before the deadline is reached and the opportunity time therefore lost. The reward earned is the aggregate of reduction in areas of all tasks. "Tasks" may be clustered in groups with additional delays imposed for switching the cursor from one group to another. The subject obviously tries to concentrate on the most rewarding tasks, but may lose time by changing the cursor to unimportant tasks or by shifting to a different task group. The

+ The authors gratefully acknowledge support from the National Aeronautics and Space Administration, Grant No. NSG-2118.

49

experiment is implemented on an Imlac graphics terminal coupled to an Interdata display, and provides wide variety of parameter combinations.

An optimal decision control model has been developed, which is based primarily on a dynamic programming algorithm which looks at all the available task possibilities, charts an optimal trajectory, and commits itself to do the first step (i.e., follow the optimal trajectory during the next time period), and then iterates the calculation. A Bayesian estimator has also been included which estimates the tasks which might occur in the immediate future and provides this information to the dynamic programming routine.

Preliminary trials comparing the human subject's performance to that of the optimal model show a great similarity, but indicate that the human skips certain movements which require quick change in strategy. We are running the model under a variety of parameter combinations (principally, rate of discounting future gain and/or preview time, and response time delay) to solve the inverse optimal problem. That is, for what parameters do the human data and the optimal model match?

INTRODUCTION

With the advent of computer technology which resulted in increased computing power and speed, and in its ever increasing availability due to decreasing costs and size (Ref. 1), the role of the human operator in manned systems is changing from that of a continuous "in-line" controller to that of a monitor/supervisor, (Ref. 2). He is therefore asked to supervise multiple dynamic processes, each of which is controlled continuously by a hierarchy of "intelligent" machines. Higher-level automation is already a commonplace practice in space- and aircraft technology (Ref. 3), and is now gaining acceptance in industrial plants as well.

PARADIGM

We can model the situation of supervisory control as the decision-maker (DM) monitoring and controlling a system of processes as shown in Figure 1.

Note that in Figure 1 the number of processes that the DM can supervise is limited mainly because of the limitations in information transmission to the DM through the displays.

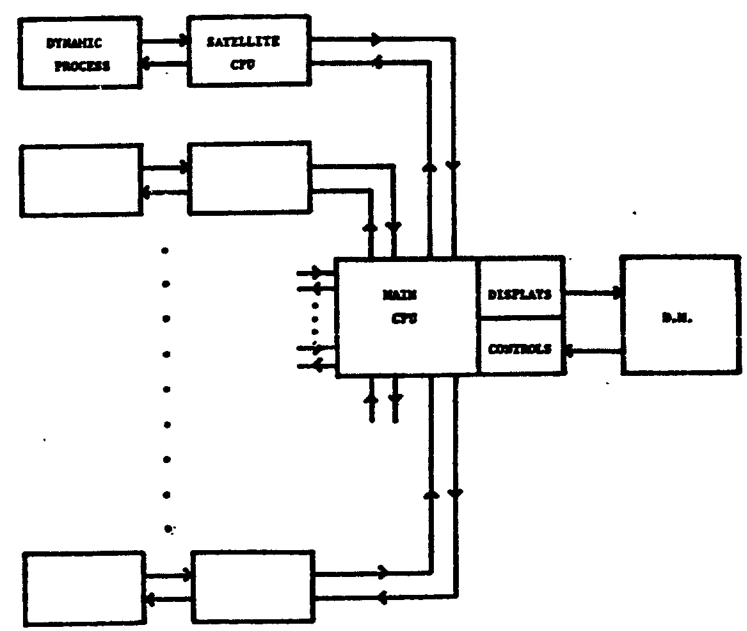

Figure 1. Hierarchial Control of Large Scale Systems with the Human Operator as the Supervisor of a Distributed Computer Network.

In the above figure we can treat the messages sent to the main CPU from a particular satellite CPU as a random noise process with an exponential auto-correlation. The longer the time constants of the auto-correlations, the less frequently the processes need higher-level attention, and the more the number of processes the DM can monitor and control.

We can group the similar processes in 'N' different ensembles. When a task is created by the satellite CPU of a particular process it will queue for the action of the DM in the particular ensemble of the process. Each queue may be characterized with a different mean interarrival time between the tasks; furthermore there may be "transition time" losses $\tau_{i,n}$ for the DM when he transfers his action from the i.th queue to the n.th one. (Ref. 4)

In each queue different tasks (i,j), $i = 1, 2, \ldots, N$
$j = 1, 2, \ldots, M_N$

-- where 'i' is the index number of the ensemble to which the task 'j' belongs -- may be created throughout the operation of the system and are worthy of the DM's attention and action.

Each task (i,j) will be characterized with a finite number of variables to indicate:

1) How far away the task is from the 'deadline' for successful action on it -- the 'POSITION' of the task.
2) With what 'SPEED' the task is moving to this deadline.
3) The 'DURATION' of the task.
4) The 'PRODUCTIVITY' of the DM for that task (or group of tasks).
5) The 'VALUE DENSITY' of the task to indicate the benefits accrued per unit time the DM acts on it. Value can then be earned either as the time integral of value density of tasks acted upon, or alternatively, the time integral of the value densities of only such tasks that are successfully completed.

At this point we can observe that the first two factors together specify the amount of time the DM has - TIME AVAILABLE - to act on the task. Similarly the third and fourth variables combine to give the amount of time the DM has to spend - SERVICE TIME - to successfully complete the task.

Note further that among the first and second, and among the third and fourth variables, there is (or may be) an extra degree of freedom. Thus we can let either 'initial position' or 'speed' be constant and also can let either 'initial duration' or 'productivity' related to the particular task be constant.

In Figure 2 we show these task variables explicitly for task (1, 1) which is queueing for the attention and/or action of the DM, along with other tasks present, namely (1, 4), (2, 1), (2, 3), (2, 5), and (3, 2) and (3, 4).

From the above discussion we can infer that the 'state' of the 'system' that the DM is supervising is the vector:

$$X = (x_{1,1,1}, x_{1,1,2}, x_{1,1,3}, x_{1,1,4}, x_{1,1,5}, x_{1,4,1}, \ldots, x_{i,j,k}, \ldots, x_{N,M_N,5})$$

where i, j, k represent the queue, the task in the particular queue, and the variables of this task at a particular time, respectively. Note that the first five terms in the above vector define the position, speed, duration, productivity, and value density associated with the first task in the first queue (1, 1) at the particular time.

In Figure 3 we show part of the state space of a task (which itself is a part of the state-space of the supervised system) to illustrate some of the above-mentioned ideas. Note that the two arrowed curves represent two possible

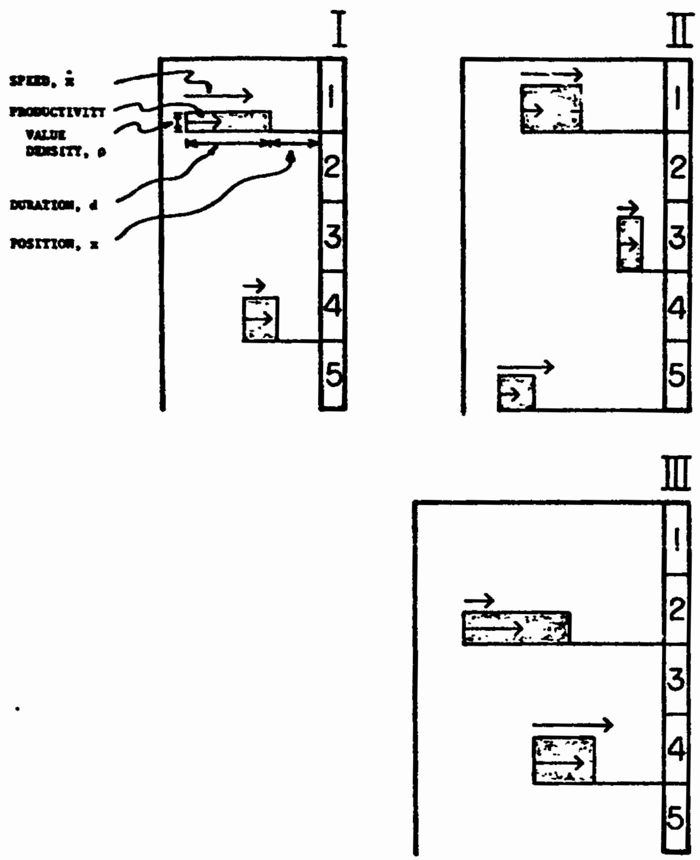

Figure 2. Paradigm of Dynamic Task Demands with Multiple Queues. (Experimental Computer Display)

state trajectories. The successful completion of a task requires that the duration associated with the task go to zero before the position variable reaches zero. The ellipse in the figure represents the boundary of a volume within which is the initial task state at the creation, with 99% likelihood.

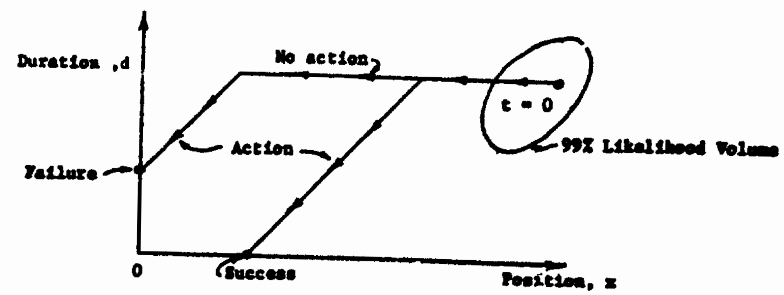

Figure 3. Task State-Space with Two State Trajectories.

We can now model the system as tasks generated according to a probability distribution, and which queue for the DM's action. In particular, we will assume that the tasks are created independently according to a real-time Poisson process with Gaussian distribution of task parameters:

$$P\left[m_{i,j+1}(t+\tau), \underline{X}_{0i} \,|\, m_{i,j}(t)\right] = P(\lambda_i, \tau, \underline{X}_{0i})$$

Note that the parameters of this probability distribution may be different for each queue. The most important parameters in this context are:

1) The mean interarrival times between successive tasks i.e., $1/\lambda_i$
2) The window-length of the queue (W_i), as an indication of the maximum expected position for the tasks on the queue.
3) The mean values and variances of the position, speed, duration, productivity and value density of the created tasks at their initial appearance (and if they do change, their system dynamics).

We are thus modeling the input to the system as an exponentially correlated disturbance. The auto-correlation and the power-spectral density of this process are shown in Figure 4. Note that this input can be considered as the output of a first-order shaping filter with characteristic Φ_{xx} to which white-noise is the input.

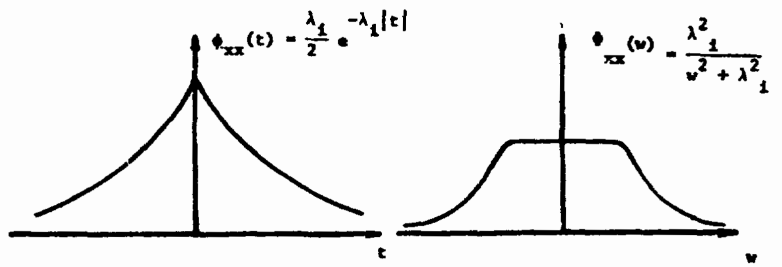

Figure 4. Autocorrelation and Power Spectral Density of Input

The responsibility of the DM is to choose among the alternatives. He will act on -- and therefore control -- the task that he chooses. He usually can act only on one task at a time, although he can time-multiplex his control action by switching from one task to another. This idea is very close to the 'Bang-Bang' control (Ref. 5), where the control action is on the boundary of a feasible control space, with the added hard-constraint that there be only one task that is being acted upon.

OPTIMIZATION

In choosing his control - i.e., which task to act upon - we can model the DM as an optimal controller who maximizes his expected returns over a planning horizon. Dynamic Programming (Ref. 6) is the most promising technique to use in the above mentioned Supervisory Control Paradigm.

In particular, the DM will act to maximize his expected total returns over a finite planning horizon, and perhaps with a non-zero discount function f(t) : --which can for example be $\exp(-\beta t)$--

$$\max E\left[\int_0^T R(t)\ f(t)\ dt\right]$$

where $R(t) = \sum_{ij} R_{ij}(t)$

in which the summation is over all the tasks that the DM expects to act over his planning horizon. $R_{ij}(t)$ is the return he gets for acting (or completing) the task (i,j) during (or at) time t.

For the case in which the DM gets credit continuously while acting on a task, the $R_{ij}(t)$ will be as shown in Figure 5.

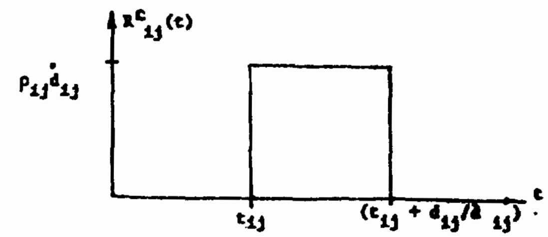

Figure 5. D.M.'s Return Function for "Continuous Credit While Acting".

In this figure, t_{ij}, ρ_{ij}, d_{ij}, $\dot{d}_{ij}$ represent the time at which the DM plans to start acting on the task, the value density of the task, the duration of the task, and the productivity of the DM for the task, respectively.

If however, the DM is going to get credit only after successfully completing a task, then $R_{ij}(t)$ will be as shown in Figure 6.

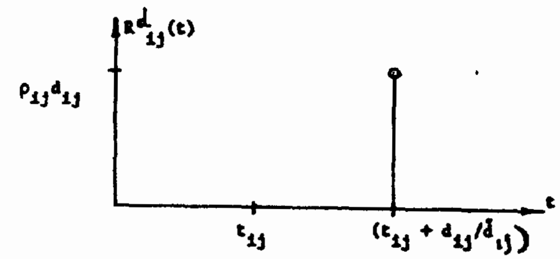

Figure 6. D.M.'s Return Function for "Credit Only When Finished".

The DM will, in effect, at each decision point, choose a number of tasks -- the number being a function of the planning horizon and the mean interarrival times-- in a linked list that he intends to act upon to maximize his expected returns, and then he will actually act upon the first task in this list.

It is probable and acceptable that he might have to give up on acting on some tasks when their 'available times' are small - due to their high speed and/or due to their proximity to the deadline - or when they have comparatively low value densities, especially in competition with other simultaneously available tasks which are preferred in these respects. Another important parameter, of course, is the transfer time between the queues. He has to consider the fact that he will end up getting no credit for a period of time when he transfers his control from the i.th queue to the n.th one. Note that when this Transition-Time matrix is a null matrix, i.e., when there are no transfer time losses and when the DM is continuously being awarded for the task he is serving, then he will do instantaneous maximization, as this will satisfy the maximization of expected returns over the long term too.

The parameters T and β which are the planning horizon and the discount rate will directly affect each other. There also is a physical limit on the planning horizon: it can (or should) not be greater than the time left to the end of the experiment.

$$T < (T)_f - t$$

From the form of the optimization equation above we can see that the discount term might be thought of as inversely proportional to the planning horizon, since the latter is the limit of the integral within which is the exponential decay term of the former. However, there are no physical limits on the discount parameter, other than it being positive semi-definite.

ESTIMATION

Up to now we have talked about the human decision maker as an optimizer. However in many cases estimation of the structure and the state of the system that he is controlling will be an integral part of his optimal dynamic decision making (Ref. 7).

In the Supervisory Control Paradigm, estimation of the state will especially be important when the speeds of the tasks are high, when the transfer time from the present queue to another one is significant, and/or when the DM is not allowed to observe another queue while acting on a task in the present one.

We have noted in the previous section that the DM has a finite planning horizon. We also note that some part of this time-horizon will be (or may be) displayed to the DM, since he will have access to a finite 'window' through which he can observe the tasks that are in the vicinity of the deadline. However in some cases this might not suffice to cover the whole planning horizon. In particular we note that the effective contribution of the DM's observation of a particular queue through the displays to his planning horizon will be:

$$L_i / V_i$$

where L_i and V_i represent the mean position and mean speed of the tasks at their initial appearance (creation) in queue i, respectively. He therefore may still have to predict ahead of what is available to him by in effect extending L_i/V_i to T, where T is defined as the planning horizon of the DM.

We note again that it is entirely possible for the DM to use the effects of only those tasks that are observable by him, in making a dynamic decision as to which task to act upon. A possible exception will be the case of not being able to observe the other queues when acting on the present one.

When making estimates about the non-observable tasks, however, the DM will use his own estimated values of the system variables - like mean position L_i, and mean speed V_i - which are presented to him as random variables, and maybe even τ_{ip} , which, although is presented as a deterministic quantity, might be perceived as a random variable by the DM, perhaps due to his own noise creation. We might add that if the DM is 'trained' in interacting with (controlling) the system, then his subjective probability distributions about the state of the system will not be too much different from the actual ones (Ref. 8).

We can model this estimation process as one in which the DM uses his *a priori* subjective probability distributions before making judgments at each instant, and then updating these subjective distributions according to what he has observed. This estimator then becomes a recursive filter, i.e., there is no need to store past measurements for the purpose of computing present estimates. We therefore are going to model this updating as one in which the DM will 'learn' about the Gaussian (and Poisson) properties of the queues while he is controlling the system (Ref. 9) - unless he was already trained about them — in a maximum likelihood/minimum variance Bayesian way. The DM will then use these *a posteriori* subjective probabilities in making his next decision to maximize his expected gains. It is interesting to note here the parallelism between this approach and the Kalman Filter used in Linear-Quadratic-Gaussian control theory, in which it is assumed that the physical properties of the Gaussian random processes are known exactly, and then the future states are estimated based upon the present states and the Gaussian properties (Ref. 10).

Taking the estimation into account, we are now ready to present a block diagram of our modeling effort:

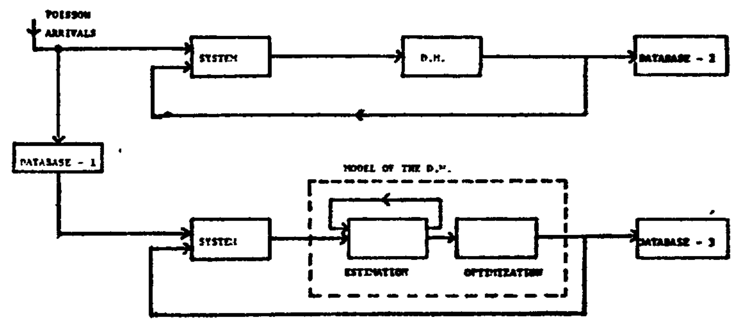

Figure 7. Block Diagram of the Modeling of the Decision-Maker.

EXPERIMENTS & Preliminary Results

Due to the speed and programming restrictions of our Interdata-70 computer/Imlac-display pair we currently have 2 programs for the experimental set-up. The first program is the interactive one. It essentially creates the tasks and displays them to the DM as in the form shown in Figure 2. The DM is able to choose one of these by pressing a pen on the data-tablet. At each iteration the choice of the DM is recorded on the disk. Also recorded in another database are the Poisson arrivals of the tasks and their parameters.

After the experiment we use these two databases to print out the time history of the DM's actions under the given circumstances at that instant. Figure 8 shows a particular subject's control action history of a 3-queue, 5 task per queue (statistically max.) system. Note that the two numbers at the top of each dotted sequence indicate the queue and the task number, respectively. Also at the appearance of each task, two numbers are written in the figure to indicate to the reader the service time the task requires and the time span the task would be available on the screen. The numbers after the stars indicate the relative value densities of the tasks, relative to the average of the mean values of this parameter in each queue.

The A's indicate action by the DM on the particular task at the given time -- written on the left hand side of the Figure. If the task was successfully completed 'SSSS' is written at the time of completion. If, however, the task was not successfully completed before it disappeared from the screen, then 'FFFF' is printed at the time of disappearance.

The second program uses the databases created during the experiments with the subjects, and does estimation and optimization at given points in the parameter space, as shown in Figure 7.

In Figure 9, we show the action trajectory of the optimal model for the partial credit mode as shown in Figure 5. The first thing that attracts attention while comparing Figures 8 and 9 is that the model responds more quickly to new tasks than does the human. To compensate for this we can adjust the model to simulate the human "Response-Time" r_{in} which adds to the τ_{in} 'Transfer Time' loss between queues i and n. This 'response-time' loss is due to decision time loss, neuro-muscular lags, and the time losses for physical hand movement on the data-tablet. In our preliminary experiments we have noticed that the effects of this response-time loss on the human's performance to that of the optimal model (i.e., operating at zero response-time loss) decreases rapidly as the 'available-times' of the tasks are increased.

Figure 10 is a comparison of value gained by the 'Optimal Control' of Figure 9 (operating at $r = \emptyset$, $\beta = \emptyset$, T = time until the end of the experiment) with the control output of a subject as shown in Figure 8. Figure 11 shows the same data for a second subject.

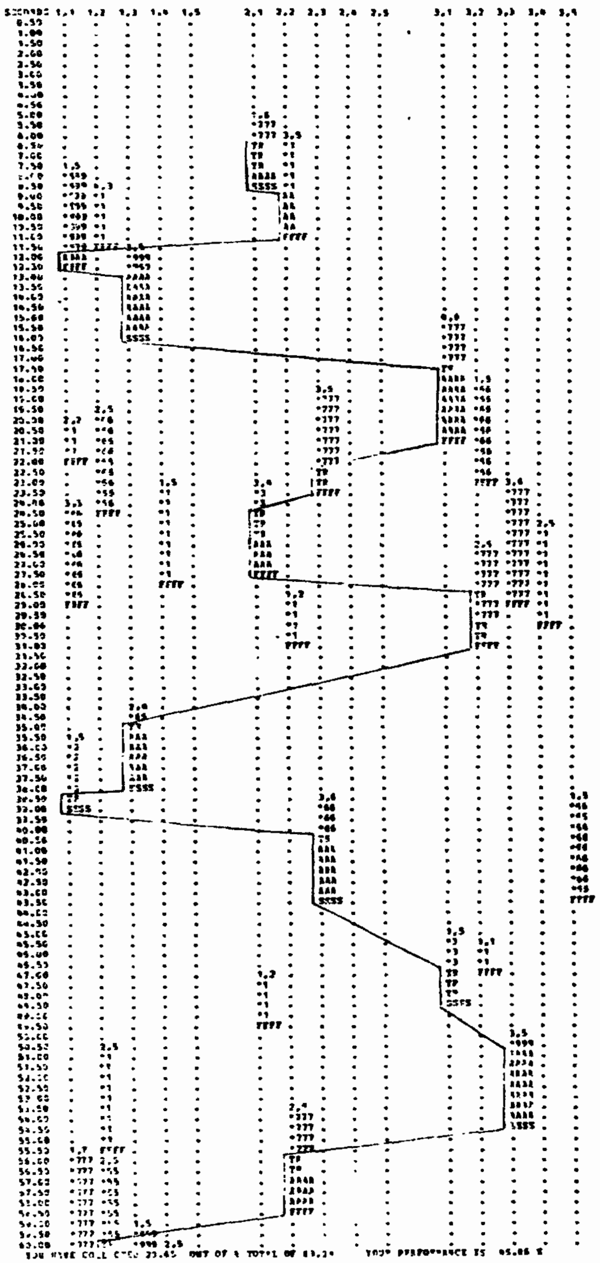

Figure 8. Record of D.M.'s Action Trajectory in Real Time

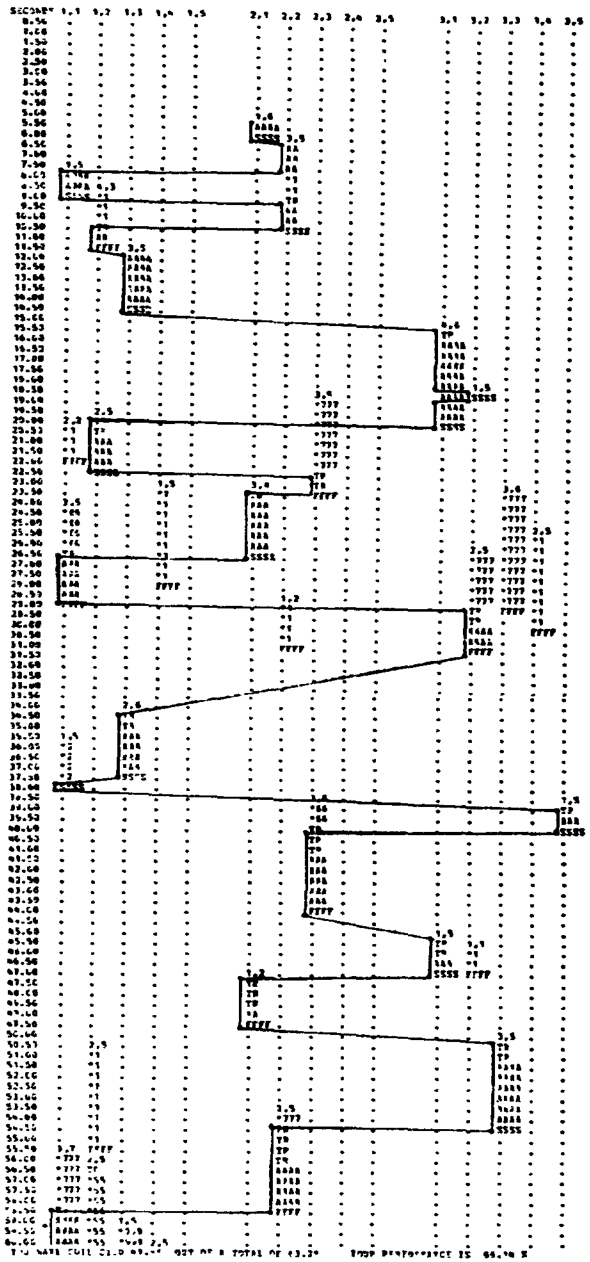

Figure 9, Record of Model's Action Trajectory in Real Time

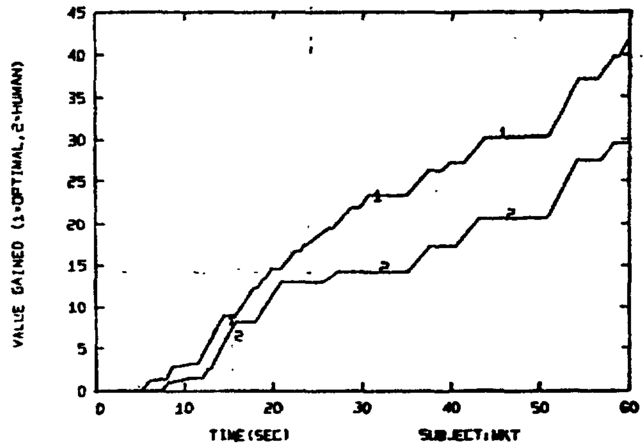

Figure 10. Time Histories of Value Gained by the DM and the Model of Figure 9. (Subject 7).

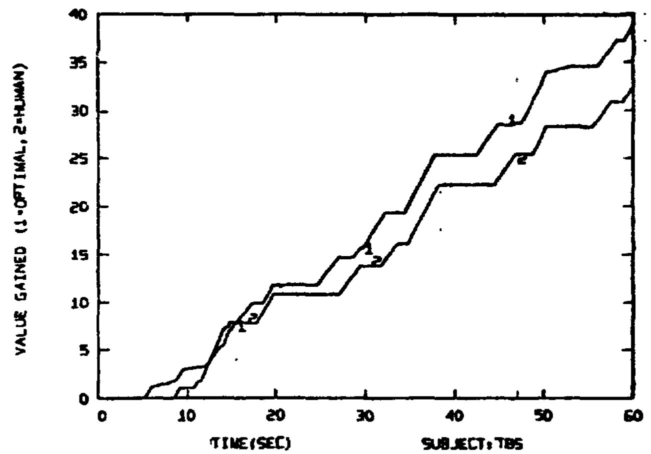

Figure 11. Time Histories of Value Gained by the DM and the Model of Figure 9. (Subject II).

By using a discount function f(t) for future returns

$$f(t) = 1 - \beta(t), \quad 0 < t < \frac{1}{\beta}$$

in the objective function, we can search the parameter β space and find the best β that can be employed to get maximum returns under given task conditions like interarrival times, transfer time delays, mean values and variances of task states, etc. While searching this β-space we also compare the time histories of cumulative value gained by the human vs. the model - as in Figures 10 and 11 - and can identify the discount parameter for the human as the β which minimizes the least squares difference between these two curves. As shown in Figures 12 and 13 the parameter β that best identifies the human is not necessarily the β that maximizes the total value gained at the end of the experiment.

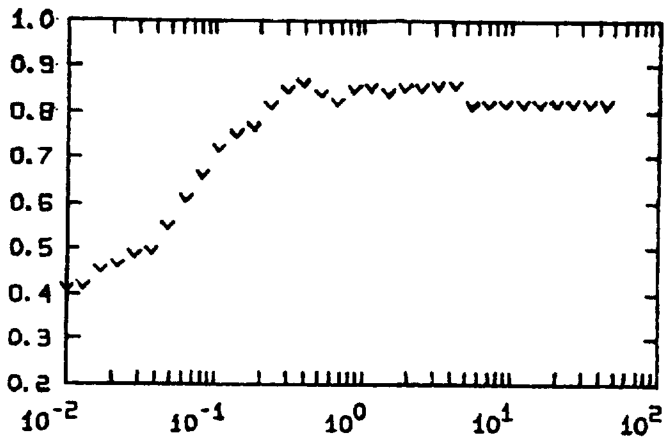

Figure 12. Value Gained by the Model as a Percentage of Total Value Offered with Varying β-discount Rate.

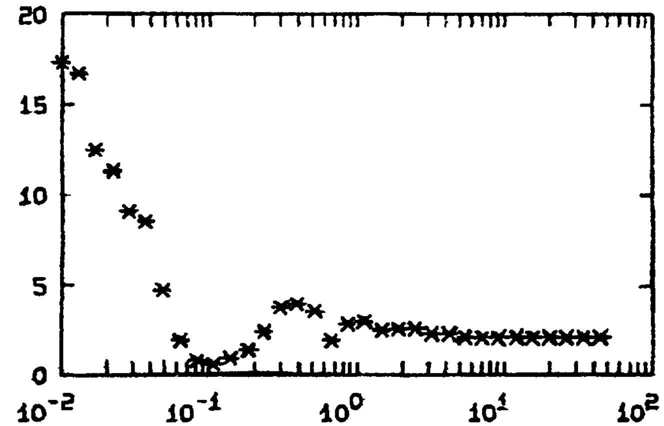

Figure 13. Least-Squares Difference Between the Model and the Human as a Function of β-discount Rate.

We are doing the identification under different criteria as well. These are:

1) minimization of the sum of the squares of the differences in instantaneous or incremental values simultaneously gained by the human and by the model.

2) maximizing the fraction of total experimental time the model and the human are simultaneously serving the same tasks.

3) maximizing the number of tasks which eventually are served by both the model and the human independent of when they are served or for how long.

REFERENCES

1. "The Smart Machine Revolution", Business Week, No. 2439, July 5, 1976.

2. J. C. R. Licklider, "Man-Computer Symbiosis", I.R.E. Trans. on Human Factors in Electronics, V. HFE-1, No. 1, March 1960.

3. E. Palmer, "Pilots Manual for the 4D Area Navigation and the Autopilot Systems in the Flight Management Project Cockpit Simulator", N.A.S.A. Report.

4. T. B. Sheridan, "Optimum Allocation of Personal Presence", I.E.E.E Trans. on Systems Science and Cybernetics, V. SSC-6, No. 3, July 1970.

5. A. E. Bryson and Y-C. Ho., "Applied Optimal Control", John Wiley & Sons, (1975 - revised printing).

6. A. Rapoport, "Dynamic Programming Models for Multistage Decision-Making Tasks", J. of Math. Psychology, V. 4, pp. 48-71, 1967.

7. H. S. Witsenhausen, "Separation of Estimation and Control for Discrete Time Systems", Proceed of the I.E.E.E., V. 59, No. 11, November 1971.

8. K. S. Fu, "Learning Control Systems - Review and Outlook", I.E.E.E. Trans. on Automatic Control, V. AC-15, No. 2, April 1970.

9. D. G. Keehn, "A Note on Learning for Gaussian Properties", I.E.E.E. Trans. on Information Theory, V. IT-11, No. 1, January 1965.

10. A. Gelb, ed., "Applied Optimal Estimation", M.I.T. Press, (1974).