NASA

Technical Memorandum 82108

Decentralized Control of Markovian Decision Processes: Existence of σ- Admissible Policies

Arnold Greenland

AUGUST 1980

National Aeronautics and
Space Administration

Goddard Space Flight Center
Greenbelt, Maryland 20771

DECENTRALIZED CONTROL OF MARKOVIAN DECISION PROCESSES: EXISTENCE OF σ-ADMISSIBLE POLICIES[1]

Arnold Greenland[2]
Mission and Data Operations Directorate
NASA Goddard Space Flight Center
Greenbelt, Maryland 20771

August 1980

[1]Work sponsored by 1980 NASA-ASEE Summer Faculty Fellowship Program

[2]Permanent Address Department of Mathematical Sciences
George Mason University
Fairfax, VA 22030

CONTENTS

DECENTRALIZED CONTROL OF MARKOVIAN DECISION PROCESSES: EXISTENCE OF σ-ADMISSIBLE POLICIES

1. INTRODUCTION

Classical control theory deals with the problem of controlling the performance of a dynamical system. In their earliest form, control models were completely deterministic in the sense that there was no probalistic or random nature to them. The first stochastic control model simply added a probabilistic disturbance term to the previously deterministic model, yet it did so with tremendous utility.

A further development in the control of dynamical systems took place with the addition of more substantive stochastic components into the model. For example, one could consider controlling an industrial production model by monitoring demand for the product, a probabilistic process, and controlling the process by a combination of factors (say, by adjusting size of work force, amount of inventory, etc.). This model is an example of what has been come to be called a Markovian decision process. An elementary explication of this theory can be found in Derman [7].

In each of the previous models, it was assumed that control was overseen by one central decisionmaker with access to all available information. Many examples exist of cases where (1) there are multiple decisionmakers and (2) no single decisionmaker has all of the information. It is this decentralized control problem that is the subject of our study. For added motivation we will end this section with several distinct examples of decentralized stochastic control models.

(a) Satellite Communications and Control

Consider a network of satellites each with finite buffers onboard to store data. The main problem to be considered is how to store data until access to a broadcast channel is available and the buffer can be dumped. Assume data arrives at each node (i.e., satellite) by some known random process. The goal is to initiate onboard control of this data management system without instantaneously

knowing the state of other nodes in the network, in particular, to avoid two satellites starting to use the same channel simultaneously (resulting in transmission collision and data loss). The best control strategy is chosen by using a realistic cost function designed to include such variables as cost due to data loss, storage cost, etc.

For an interesting explication and partial solution to this problem, see Schoute [23].

(b) Combat Command Structure

A very different sort of model is that of the control (or management) of a battle by field officers who cannot maintain perfect communication with their superiors (and certainly the "heat" of a battle field is not conducive to good information flow).

The question is: can one produce a strategy which minimizes battle losses and maximizes the chance of reaching an objective, but which functions in an environment of decentralized decision-making and information flow?

(c) Management of Marketing Force

For an example which has application to the business world, consider a sales force out "in the field." They have only bare communication (possibly they agree to call in to a central person twice daily) yet one wishes to produce a strategy of resource allocation which maximizes profits and minimizes duplication of effort. The random occurrences are such things as product demand, weather conditions, etc. Each salesman must make some decisions but without benefit of full information. Possibly a set of "guidelines" for decisionmaking could produce more overall reward than each decisionmaker maximizing only his own reward.

2. MOTIVATION FOR CURRENT RESEARCH

In this section, we will lay out the particular model suggesting this research. Certainly one of the driving forces behind this work is a NASA-sponsored study on Artificial Intelligence and Robotics (usually referred to as the Sagan Report). One recommendation of that study was that NASA move

toward employing more "state-of-the-art" technology in these areas as well as taking a leading position in research and development of such technologies. Within that context, the following model was communicated to us by Mr. R. L. Larsen.

Assume first of all that the hardware and software exists for onboard satellite computer processing. By this we mean the capacity to perform a wide range of functions such as preprocessing of telemetry data (with the idea in mind of filtering out bad or useless data), orbit and attitude determination, and corrective maneuvering. One could foresee some of the scheduling and planning functions now occurring manually on the Earth being transferred for automatic control to an onboard network of communicating computers.

The problem then surfaces: how does one control the real-time flow of processing throughout this network of computers. The object is to minimize loss of information (because of buffer overflow) yet at the same time reduce costs of data transfer and other communication costs. Again, because of time delay and other constraints, it may not be possible for each satellite control device to have full information about states and decisions taken at other satellites.

The mathematical model reduces to a network of queues, where computer jobs or input data (in some standard units) are the objects lining up on the queues. The controlling device is to change the rate of data input or output by doing one of several "actions." For example, one could transfer processing to a neighboring node in the network, one could transfer back to an essentially infinite queue on the Earth, one could activate reserve computing power (say on a space station), or one could reprioritize the processing of jobs in a way which allows faster service. Each of those actions is essentially simply altering the arrival and service parameters of the queues at each node. These decisions are viewed as being made locally at each node and only with partial information concerning the entire network. For example, a particular node may only have vital information concerning itself and, say, its two closest neighboring satellites; however, it must decide how to control its processor based on that incomplete knowledge. The details of what one means by a controlling device for such a decentralized system and ways to obtain such devices are the basic problems to be addressed in this document.

3. MATHEMATICAL PRELIMINARIES

We present in this section a short discussion of Markovian decision models. An excellent elementary introduction is that due to Derman [7], while more sophisticated treatments will be found in Hinderer [9] and Schäl [22].

A Markovian decision process is a stochastic process modeling the time evolution of a dynamic system which is "controlled" by sequences of decisions (actions) periodically or at regular time intervals. The process may evolve either in a discrete or continuous time frame: however, for reasons of simplicity and as we shall later see with no loss in generality, we will consider only discrete time processes in this report.

The model is described by a 5-tuple (S, A, D, p, r) with the following interpretations:

(1) S is the set of states of the system. In general, S may be taken to be a standard Borel space: but for the purposes of this report, we assume S is at most countable.

(2) A is the set of possible actions. Again, there are general assumptions that can be made concerning A, but we assume A is a set satisfying a decomposition that will be described in the next paragraphs.

(3) D is a function from "histories" of the process into subsets of actions. By a history at time t, h_t, we mean a member of the set: $\bigcup_{x \in S} (\{x\} \times A_x)^{t+1}$. Thus, a history, h_t, looks like:

$(x_0, a_0, x_1, a_1 x_2, a_2, \ldots, x_t, a_t)$ where

x_0 = state of time 0

a_0 = action taken at time 0

x_1 = state of time 1

.
.
.

a_t = action taken at time t.

An augmented history simply adds the state at time t + 1 onto the history at time t. We will denote it by the symbol, (h_t, x_{t+1}).

The function D associates to each augmented history, (h_t, x_{t+1}), the set of actions available to the decisionmaker when the process evolved exactly as that described in the augmented history.

For the purpose of this study, we will have the following assumption and decomposition:

(a) $D(h_t, x_{t+1})$ is finite for all (h_t, x_{t+1})

(b) $A = U\{D(h_t, x_{t+1}) | (h_t, x_{t+1})$ is an augmented history$\}$

(4) The symbol p represents the (Markov) transition probability. For each $a \in A$, $x, y \in S$, we interpret the symbol $p_{xy}(a)$ to be the probability of changing from state x to state y under action a. Of course, the following regularity conditions must hold:

(a) $p_{xy}(a)$ is defined only when it makes sense (i.e., a must belong to $D(h_t, x)$ for some history, h_t.

(b) $\sum_{y \in S} p_{xy}(a) = 1$ for all $x \in S$ and $a \in D(h_t, x)$ for some h_t.

This transition probability represents the uncontrollable probabilistic aspect of the model.

(5) The last element of the model is the function, r, the reward function. Formally r is a function from state/action pairs into the real numbers. r(x,a) is interpreted to mean the reward (negative reward is "cost") associated with choosing action a when in state x. Of course, r is defined for all $x \in S$ and $a \in D(h_t, x)$ for some h_t.

The basic problem associated with this model is to "control" the time evolution in such a way as to "maximize" reward. There are still two key concepts yet undefined.

Let H_t be the set of all histories up to time t. By a policy, π, we mean a sequence, $(\pi_0, \pi_1, \pi_2, \ldots)$ where $\pi_t : H_t \times S \to \mathcal{P}(A)$ and $\mathcal{P}(A)$ is the set of probability distributions on the action space.

$\pi_t(\cdot | h_t, x)$ is the probability distribution on A when the system has experienced an augmented history, (h_t, x). Since we are assuming $D(h_t, x)$ is a finite set, $\pi_t(\cdot | h_t, x)$ can be interpreted as a discrete probability density on $D(h_t, x)$. Let the symbol, Δ, denote the set of all such policies.

If for each t the probability is concentrated on one state, i.e., $\pi_t(y|h_t, x) = 1$ for some state $y \in S$, then we call π a deterministic policy.

It is the policy π that in fact "controls" the time evolution of the process. It is desirable to find deterministic policies because otherwise the decisionmaker is faced with the unpleasant option of having to perform some random procedure (say, flip a theoretical coin) to decide which action to take when in some state.

As a means of picking an optimal policy, we will use the reward function defined above. Further, we want to pick a policy that is good throughout the evolution of the process (or at least to some large finite time frame). It is here that there is significant divergence in the analysis. For the purpose of this study, we will restrict ourselves to "discounted" reward functions, but we acknowledge that other reward functions are of interest and deserve similar investigation.

Let α be a fixed discount factor, $0 \leq \alpha < 1$. Let π be a fixed policy. The theory of stochastic processes insures that to each policy and to each initial distribution on the state at time 0, there is a stochastic process generated. We denote the stochastic process by (X_t, A_t) where X_t is the state at time t and A_t is the action taken at time t. We will use the notation P_π and E_π for the probability and expectation operators under the policy π.

Define $V_\alpha(\pi, x) = E_\pi\left\{\sum_{t=0}^{\infty} \alpha^t r(X_t, A_t) | X_0 = x\right\}$. This is the discounted reward function given the process starts in state x. Since we will be assuming α is constant throughout, that symbol will be suppressed in the notation.

Next define, $V(x) = \sup\{V(\pi, x) | \pi \in \Delta\}$. If there exists a policy π^* such that for all $x \in S$, $V(\pi^*, x) = V(x)$, then we say π^* is an optimal policy.

The basic results in this area are (1) theorems providing (under appropriate assumptions) existence of optimal (possibly deterministic) policies and (2) theorems outlining methods of computing those policies. We refer the reader to Derman [7] for a documented account of such results:

however, one could not proceed without at least mentioning the "principle of optimality" of Bellman because of its motivational value. One can usually prove (in centralized control models) that the function V(x) satisfies the following equation:

$$V(x) = \sup_{a}\left\{r(x, a) + \alpha \sum_{y} p_{xy}(a)\, V(y)\right\}$$

The interpretation is important: the equation says that the expected discounted reward starting from state x is the same as the sum of the one step cost of being in state x under the "best action," a, plus the weighted and discounted cost of starting from another state, y, one time unit later.

This equation is useful in deriving algorithms for computing optimal strategies.

4. DECENTRALIZED MARKOVIAN DECISION MODELS

In this section we will build upon the structure discussed in Section 3 to apply it to a decentralized control situation. There is no claim of full generality in the model to be described; in fact, we have limited the scope initially for ease of analysis.

Consider a network with N nodes, and assume that the state of the system at each node can be described by a non-negative integer. A useful example to have in mind is that at each node there is a queue where an integer will describe the number of people in line or in service at that queue. The state of the entire network is thus given by a vector $(n_1, n_2, \ldots, n_N)$ where n_i is the state of node i. Let A_i be the set of actions available to a controller at node i. We can fit this situation into the context of the previous section by defining:

$S = S_1 \times \ldots \times S_N$ and $A = A_1 \times \ldots \times A_N$

where S_i is the set of states available at node i.

The key ingredient that must be added now is that of an information structure. Several authors have dealt with this problem already. With no pretense of being complete we mention two sources (in very different disciplines) dealing with this issue. See, for example, Marschak and Radner [19] or Witsenhausen [30] for more details of work in this area.

In this paper an information structure, σ, will be a finite sequence of projection functions

$$\sigma = (\sigma_1, \ldots, \sigma_N)$$

where each σ_i maps the set $S_1 \times \ldots \times S_N$ onto some subset of the S_i's, say, $S_{i_1} \times \ldots \times S_{i_k}$.

For example, suppose that $\sigma_1(x_1, x_2, \ldots, x_N) = (x_1, x_3, x_N)$. We interpret the function, σ_1, as saying that controller 1 has full information about nodes 1, 3 and N.

To simplify notation we introduce the use of σ_i as a superscipt. Its use is meant to simply apply the appropriate projection operation whenever it is needed. It is best to illustrate by example.

If σ_1 is as above, by $(x_1, \ldots, x_N)^{\sigma_1}$ we mean the vector, (x_1, x_3, x_N). Similarly σ_1 can be "applied" to actions as follows:

$$(a_1, \ldots, a_N)^{\sigma_1} = (a_1, a_3, a_N).$$

In fact, we want to also use the notation freely with such complicated objects as histories. If h_t is a history in the decentralized model, then h_t looks like: $h_t = (x_1^0, \ldots, x_N^0), (a_1^0, \ldots, a_N^0), (x_1^1, \ldots, x_N^1), (a_1^1, \ldots, a_N^1), \ldots)$ where x_i^t and a_i^t are the state and action at time t at node i.

By $(h_t)^{\sigma_1}$ we mean: $((x_1^0, x_3^0, x_N^0), (a_1^0, a_3^0, a_N^0), \ldots)$. In other words, the superscript σ_i indicates that all information available to controller i is extracted from the "whole system" state or action vectors.

Our current goal is to define what we mean by a policy which is compatible with the information structure. It is obvious that the N decisionmakers can only "go on" the data available to them. This notation helps express this idea. We add that this model does not really handle time delays in information which were noted in Section 2 as being important. A notation more useful to those features is that of Witsenhausen [30], but we will forego those complications in the context of this document.

To complete the Markov decision model, we assume the existence of a transition probability p, a decision structure D and a one-step cost function r as described in Section 3.

By a σ-admissible policy, π, we mean a policy such that whenever $(h_t, x)^{\sigma_i} = (h'_t, x')^{\sigma_i}$ we have

$$P_\pi(A_t^{\sigma_i} = a^{\sigma_i} | h_t, x) = P_\pi(A_t^{\sigma_i} = a^{\sigma_i} | h'_t, x').$$

The above definition simply quantifies what we mean by decentralized control: i.e., whenever a controller, i, has a certain information configuration, he will always act in the same way even if the history or state at other nodes is different. The set of all such σ-admissible policies is denoted Δ^σ.

The mathematical problems are similar to those in the standard control problem. For example, does there exist a σ-admissible optimal policy? If so, is it deterministic? Does the discounted reward function satisfy a functional equation similar to the "Principle of Optimality?" Finally, are there reasonable and implementable algorithms that can be used to produce the optimal policies?

In Section 5 we will answer the first question, and in Section 6 we will suggest some approaches that could be useful in answering the remaining questions.

5. EXISTENCE OF σ-ADMISSIBLE POLICIES

We begin this section by defining the topology on the space of policies Δ. We say $\pi^{(n)}$, a sequence of policies in Δ, converges to π if and only if for each $t = 0, 1, 2, \ldots$; $x \in S$; $h_t \in H_t$ and; $a \in D(h_t, x)$ $\lim_{n \to \infty} \pi^{(n)}(a|h_t, x) = \pi(a|h_t, x)$. This is exactly the standard product topology in the compact space

$$\prod_{\substack{(h_t, x) \\ \text{and} \\ t = 0, 1, \ldots}} [0, 1]^{D(h_t, x)}$$

In fact, Δ is simply a closed subset of the above compact space, i.e., those elements such that for each t, (h_t, x), the function $\pi : D(h_t, x) \to x$ satisfies

$$\sum_a \pi(a|h_t, x) = 1.$$

Thus Δ is a compact set.

It is well known, see Derman [5], that under the assumptions of this report, $V_\pi(x)$ is a continuous function on the space Δ. Thus, since Δ is compact, V_π attains extreme values in the set Δ.

We use this fact in the following:

Theorem 1. There exist σ-admissible policies.

Proof. First we will clarify the meaning of σ-admissible optimality. π^* is such a policy if for all $x \in S$

$$V_{\pi^*} = \sup\{V_\pi(x) | \pi \in \Delta^\sigma\}$$

We will be able to apply the standard argument for continuous functions on compact sets if we can show that Δ^σ is itself compact. In fact, we need only show that as a closed subset of Δ.

To that end let $\{\pi^{(n)}\}$ be a sequence of policies in Δ^σ and assume $\pi^{(n)} \to \pi$ in Δ. We want to show that $\pi \in \Delta^\sigma$. Let t be a fixed time parameter and assume that for $i \in \{1, \ldots, N\}$

$$(h_t, x)^{\sigma_i} = (h'_t, x')^{\sigma_i}$$

Since $\pi^{(n)} \in \Delta^\sigma$ for each n, we have

$$P_{\pi^{(n)}}(A_{t+1}^{\sigma_i} = a_{\sigma_i} | h_t, x) = P_{\pi^{(n)}}(A_{t+1}^{\sigma_i} \; i = a\sigma_i | h'_t, x')$$

Since $\pi^{(n)} \to \pi$, then

$$P_{\pi^{(n)}}(A_{t+1}^{\sigma_i} = a_{\sigma_i} | h_t, x) \to P_\pi(A_{t+1}^{\sigma_i} = a_{\sigma_i} | h_t, x)$$

and

$$P_{\pi^{(n)}}(A_{t+1}^{\sigma_i} = a_{\sigma_i} | h'_t, x') \to P(A^{\sigma^i} = a_{\sigma_i} | h'_t, x')$$

Thus it follows that

$$P_\pi(A_{t+1}^{\sigma_i} = a_{\sigma_i} | h_t, x) = P_\pi(A_t^{\sigma_i} = a_\sigma | h'_t, x')$$

Therefore, Δ^σ is closed. The conclusion of the theorem follows using exactly the same argument as that used by Derman [5].

6. OPEN QUESTIONS

The previous section settles the question of whether it is worth looking for σ-admissible optimal solutions to the decentralized control problem. There are several important and unanswered questions in this area.

First, it is important to know whether there are circumstances under which there always exist deterministic σ-admissible optimal policies. A deterministic policy is given simply by a sequence of functions, $\pi = (f_1, f_2, \ldots)$ where each f_i maps states into actions (i.e., the action which has possibility 1 of occurring). In addition, the defining condition means that whenever $(h_t, x)_i^{\sigma_i} = (h_t', x')^{\sigma_i}$ then $[f(x)]^{\sigma_i} = [f(x')]^{\sigma_i}$. It is our view at this point in time that some stringent conditions will have to be imposed in order to force the existence of deterministic policies. Likely candidates for such conditions are (1) either some heirarchical structure where information is known by "superiors" in the network about all nodes "under" them, or (2) an ordering on actions which would allow the use of monotone policy analysis. See, for example, Serfozo [26] for methods in the latter direction.

The second major area of work is with finding implementable algorithms to produce the optimal policies. Historically, with the centralized control problem, the discounted reward function, V, satisfied a functional equation which allowed the use of dynamic programming techniques. Thus far we have been unable to obtain such results.

A more hopeful approach may be one similar to that outline by Ho [10] in a recent paper. The idea is to pick or guess a starting optimal policy. One uses the general policy, leaving out one node at a time, say node i, and obtain a "revised" solution at node i (with other nodes fixed). After all nodes have a revised solution, one compares the reward function under the revised solution to the original. The algorithm concludes when no better reward function is obtained.

Of course there are no theorems yet obtained in this regard, but we feel it is a method worth investigating.

7. REEXAMINATION OF MOTIVATING EXAMPLE

We close this report with a few comments concerning the relationship between the problem outlined in Section 2 and that discussed subsequently. The first point to make is that the control problem outlined in Section 2 is clearly a continuous time model. Certainly arrivals to the various nodes in the network can happen at any instant in time. However, the mathematics described is for a discrete time problem.

The point is that the continuous time problem has the discrete problem imbedded inside (at the jump points in the process); and, more importantly, a solution of one of the problems (i.e., optimizing the reward function) will induce a solution to the other. For an excellent explication of this phenomenon see Serfozo [25].

A second point to be made is that the mathematics described in this document does not (on the surface) embrace the problem of time-delayed information structures. In fact, with a notation more adapted to that situation, say the notation in Witsenhauser [30], one can include the time-delayed case also. It is expected that subsequent work will follow that direction.

8. ACKNOWLEDGMENTS

I wish to thank Mr. R. L. Larsen for introducing us to this most facinating area and for providing many hours of useful dialogue; and to N. U. Prabhu, R. F. Serfozo and A. K. Agrawalla for fruitful discussions of the problem. Finally, I acknowledge the support of the NASA-ASEE Summer Faculty Fellowship Program toward completion of this report.

BIBLIOGRAPHY

[1] Blackwell, D., "Discrete Dynamic Programming" Ann. Math. Stat. 33 p. 719-726 (1962).

[2] Blackwell, D., "Discounted Dynamic Programming" Ann. Math. Stat. 36 p. 226-235 (1965).

[3] Crabill, T. B., D. Gross, M. J. Magazine, "A Classified Bibliography of Research on Optimal Design and Control of Queues" Ops. Res. 25 p. 219-232 (1977).

[4] Denardo, E. V., "Contraction Mappings in the Theory Underlying Dynamic Programming" SIAM Review 9 p. 165-177 (1967).

[5] Derman, C., "Markovian Sequential Control Processes Denumerable State Space" J. of Math. Anal. and Appls. 10 p. 295-302 (1965).

[6] Derman, C., "Denumerable State Markovian Decision Processes-Average Cost Criterion" Ann. Math. Stat. 37 p. 1545-1553 (1966).

[7] Derman, Cyrus, Finite State Markovian Decision Processes, Academic Press (1970).

[8] Federgruen, A. and H. C. Tijms, "The Optimality Equation in Average Cost Denumerable State Semi-Markov Decision Problems, Recurrency Conditions and Algorithms" J. Appl. Prob. 15 p. 356-373 (1978).

[9] Hinderer, K., Foundations of Non-Stationary Dynamic Programming with Discrete Time Parameter Lec. Notes in Operations Research and Mathematical Sciences, No. 33 Springer-Verlag, N.Y.

[10] Ho, Y. C., "Team Decision Theory and Information Structures" Proc. IEEE 68 p. 644-654 (1980).

[11] Ho, Y. C. and K. C. Chu, "Team Decision Theory and Information Structures in Optimal Control Problems – Parts I and II" IEEE Trans. on Automatic Control AC-17 p. 15-28 (1972).

[12] Ho, Y. C. and K. C. Chu, "Information Structure in Dynamic Multi-Person Control Problems" Automatics 10 p. 341-351 (1974).

[13] Ho, Y. C., M. P. Kastner, E. Wong, "Teams, Signaling and Information Theory" IEEE Trans. Auto. Control AC-23 p. 305-312 (1978).

[14] Kakumanu, P. "Continuously Discounted Markov Decision Model with Countable State and Action Space" Annals, of Math. Stat. 42 p. 919-926 (1971).

[15] Kakumanu, P. "Continuous Time Markovian Decision Processes Average Return Criterion" J. of Math. Anal. Appl. 52 p. 173-188 (1975).

[16] Kakumanu, P. "Relation Between Continuous and Discrete Time Markovian Decision Problems" Naval Research Logistics Quarterly 24 p. 431-439 (1977).

[17] Lippman, S. A. "Applying a New Device in the Optimization of Exponential Queueing Systems" Ops. Res. 23 p. 687-710 (1975).

[18] Marschak, J., "Elements For A Theory of Teams" Management Science 1 p. 127-137 (1955).

[19] Marschak, J. and R. Radner The Economic Theory of Teams Yale Univ. Press, New Haven, CT (1971).

[20] Miller, B. L., "Finite State Continuous Time Markov Decision Processes with a Finite Planning Horizon Siam J. of Control 6 p. 266-280 (1968).

[21] Radner, R., "Team Decision Problems" Ann. Math. Stat. 33 p. 857-881 (1962).

[22] Schal, M., "Conditions for Optimality in Dynamic Programming and for the Limit of n-Stage Optimal Policies to be Optimal" Z. Wahrscheinlichkeitstheorie verw. Gebiete 32 p. 179-196 (1975).

[23] Schoute, F. C., "Decentralized Control in Packet Switched Satellite Communication" IEEE Trans. Auto. Control AC-23 p. 362-371 (1978).

[24] Serfozo, R. F., "Monotone Optimal Policies for Markov Decision Processes" p. 202-215 in Mathematical Programming Study 6 North-Holland Pub. Co. (1976).

[25] Serfozo, R. F., "An Equivalence Between Continuous and Discrete Time Markov Decision Processes" Operations Research 27 p. 616-620 (1979).

[26] Serfozo, R. F., "Optimal Control of Random Walks, Birth and Death Processes and Queues" to appear, Advances in Applied Probability.

[27] Sobel, M. J., "Optimal Operation of Queues" in Math. Models of Queueing Lec. Notes in Econ. and Math. Systems Vol. 98 (Springer) p. 231-261 (1974).

[28] Stidham, S. J. and N. U. Prabhu, "Optimal Control of Queueing Systems" in Mathematical Methods in Queueing Ed. A. B. Clarke, Lecture Notes in Economics and Math. Systems, Vol. 98 Springer, N.Y. p. 263-294 (1974).

[29] Strauch, R. E., "Negative Dynamic Programming" Ann. of Math. Stat. 37 p. 871-890 (1966).

[30] Witsenhausen, H. S., "Separation of Estimation and Control for Discrete Time Systems" Proc. IEEE 29 p. 1557-1566 (1971).