Preface

This book grew out of a two day workshop that was held in May 2005 and was funded by the U.S. Defense Advanced Projects Research Agency (DARPA) and the U.S. National Science Foundation (NSF). The express purpose of this workshop was to gather together key contributors to the field of active sensing and sensor management to discuss the state-of-the-art in research, the main mathematical approaches to design and performance approximation, the problems solved and the problems remaining. At the end of the workshop the participants had generated an outline and agreed on writing assignments.

The intended audience for this book are graduate students, engineers and scientists in the fields of signal processing, control, and applied mathematics. Readers would benefit from a rudimentary background in statistical signal processing or stochastic control but the book is largely self contained. Appendices cover background material in information theory, Markov processes, and stopping times. A symbol index and a subject index are also included to facilitate the reader's navigation through the book.

Thus the book lies somewhere between a coherent textbook and a loose collection of papers typical of many recent edited collections on emerging topics in engineering. Like an edited collection, the chapters were written by some of the principal architects of recent advances in sensor management and active sensing. However, authors and editors attempted to adopt a common notation, cross reference other chapters, provide index terms, and adhere to an outline established at the NSF workshop. We hope the reader will find that the book has benefited from this extra planning and coordination.

Alfred Hero, David Castañón, Douglas Cochran, and Keith Kastella

Ann Arbor, Boston, Tempe, Ypsilanti
July 2007

Chapter 2

STOCHASTIC CONTROL THEORY FOR SENSOR MANAGEMENT

David A. Castañón

Boston University, Boston, MA, USA

Lawrence Carin

Duke University, Durham, NC, USA

1. Introduction

Consider the following scenario: a physician examining a patient for breast cancer feels a hard area during an initial examination; she subsequently sends the patient to obtain a mammogram of the area, and sends the mammogram to a pathologist. The pathologist notes the presence of two local areas with potentially cancerous growths, but is unable to localize the areas accurately; he sends the patient to another imaging center, where a full 3-D Computed Tomography (CT) image is obtained and returned to the pathologist. The new CT image identifies accurately the location of the potential areas, but does not provide enough information to identify the nature of the growth. The pathologist performs two biopsies to extract sample cells from the two areas and examine them under a microscope to complete the diagnosis.

In a different setting, consider a player at a blackjack table in Las Vegas, playing two simultaneous games against a dealer. The player sees all the cards in his hands for both games, but only sees one of the dealer's two cards. The player asks for an extra card in his first game; after seeing the card, he asks for a second one. He sees this card and decides to stop and switch to the second game. After examining his cards, he chooses to stop asking for cards and let the dealer draw.

In a third setting, a phased-array radar is searching for new aircraft, while trying to maintain accurate track and classification information on known aircraft in its field of regard. The radar schedules a sequence of short pulses aimed at areas where new objects may appear, interleaved with short pulses aimed at positions where known objects are moving to update their position and velocity information using recursive estimation. Occasionally, the radar also introduces longer high range resolution (HRR) imaging waveforms into the mix and focuses these on known objects to collect HRR images of scatterers on the moving platforms, thereby providing information for estimating the object type. The interested reader is referred to Chapters 4, 5, 7 and 10 for radar applications and to Chapter 11 for some history and perspectives on defense applications of sensor management.

The above examples share a common theme: in each example, decisions are made sequentially over time. Each decision generates observations that provide additional information. In each example, the outcome of selecting a decision is uncertain; each subsequent decision is selected based on the previous observations, toward the purpose of achieving an objective that depends on the sequence of decisions. In each example, uncertainty is present, and the ultimate outcome of the decisions is unknown. In sum, these are sequential decision problems under uncertainty, where the choice of decisions can be adapted to the information collected.

Sequential decision problems under uncertainty constitute an active area of research in fields such as control theory [19–21, 23–25, 154, 251], statistics [243, 55, 34, 35], operations research [111, 70, 192, 157, 198], computer science [41, 227, 121, 16] and economics [17, 202], with broad applications to problems in military surveillance, mathematical finance, robotics, and manufacturing, among others. This chapter presents an overview of mathematical framework and techniques for representation and solution of sequential decision problems under uncertainty, with a focus on their application for problems of dynamic sensor management, where actions are explicitly selected to acquire information about an underlying unknown process.

The classical model for dynamic decisions under uncertainty is illustrated in the control loop in Figure 2.1(a). In such systems, information collected by sensors is used to design activities that change how the underlying system evolves in time. The problems of interest in this chapter differ from this model in a substantial manner, as illustrated in Figure 2.1(b). In sensor management, actions are not selected to change the evolution of a dynamical system; instead, they are selected to improve the available information concerning the system state. Thus, the focus is on controlling the evolution of information rather than state dynamics.

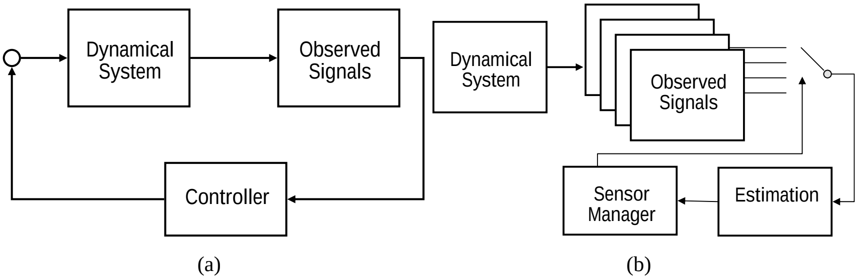


Figure 2.1. a) Feedback control loop b) Sensor management control

The foundation of sequential decision theory under uncertainty is a mathematical framework for representing the relationship between the underlying unknown quantities that may evolve in time, the relationship of observables to these unknowns, the objectives that measure the goals of the problem, and the effects that actions have on observables, the underlying unknown quantities, and the problem objectives. In this chapter, this representation is based on a formal probabilistic framework, with the following characteristics:

- Unknown quantities of interest are modeled as *states* of a dynamical system, modeled as a Markov process. A quick overview of Markov processes is included in Section 2 of the Appendix. At any time, the current state summarizes the statistical information needed to predict the evolution of future states.
- Observed quantities will also be modeled statistically, in terms of their relationship with the underlying state at the time observations are acquired.
- Actions may affect either the evolution of the state of the dynamical system, or the nature of the observation acquired. The latter will be more typical for sensor management. The choice of action may be constrained by the available information collected.
- Objectives will depend on the choice of actions and the specific dynamic trajectories of the state, with an additive structure over stages.

In the rest of this chapter, we differentiate between two classes of problems: *Markov Decision Problems* [193, 25, 198], where the observations provide enough information to determine exactly the current state of the system, and *Partially Observed Markov Decision Problems* [219, 218, 173, 248, 162], where the history of observations leaves residual uncertainty concerning the current state. We will overview the formulation for these classes of problems, selected key theoretical results, discuss algorithms for obtaining solutions. The chapter concludes with an example application to illustrate the methodology.

2. Markov Decision Problems

We restrict our discussion to decision problems that evolve using a discrete index, which we refer to as stages.

Definition 2.1 *A Markov Decision Process (MDP) consists of*

- *A discrete stage index* $k \in \{0, 1, \ldots, N\}, N \leq \infty$
- *A set of possible states* $\mathcal{S}_k$ *for each stage index* k
- *An initial value for the state* $s_0 \in \mathcal{S}_0$
- *A set of possible actions* $\mathcal{A}_k$
- *A family of action constraints* $\{\mathcal{A}_k(s) \subset \mathcal{A}_k\}$*, for* $s \in \mathcal{S}_k$*.*
- *A state transition probability kernel* $\mathcal{T}_k(ds'|s,a)$*, where* $\mathcal{T}_k(ds'|s,a) \equiv \mathrm{P}(s_{k+1} \in ds'|s_k = s, a_k = a)$*,*
- *A real-valued single stage reward function* $R_k(s,a)$

The spaces $\mathcal{S}_k$ and $\mathcal{A}_k$ are assumed to be metric spaces. The reward functions $R_k(s,a)$ and the transition kernels $\mathcal{T}_k(ds'|s,a)$ are assumed continuous functions of s, a. This is trivially satisfied when the spaces $\mathcal{S}_k, \mathcal{A}_k$ are discrete. Furthermore, we assume that, for each state, the admissible actions $\mathcal{A}_k(s)$ form a compact subset of $\mathcal{A}_k$. The resulting state evolves according to a Markov process given the actions $a_k, k = 0, \ldots, N-1$, so that the effects of an action a_k taken in state s_k depend only on the current value of that state and not on the prior history of the state.[1]

Observations of past decisions, plus past and current values of the state, are available to select the choice of next decision. Let $\mathcal{I}_k$ denote the information available at stage k, defined as:

$$\mathcal{I}_k = \{s_0, a_0, \ldots, s_{k-1}, a_{k-1}, s_k\} \tag{2.1}$$

[1] The reader is cautioned that this chapter does not adopt the upper/lower case notation to distinguish between random variables and their realizations, e.g., as in S_k and s_k, respectively. In this book, the upper/lower case convention is only used when its omission would risk confusing the reader.

A *policy* at stage k is a mapping $\gamma_k(\mathcal{I}_k)$, from the set of all possible information states $\{\mathcal{I}_k\}$ to $\mathcal{A}_k$. In this chapter, we define policies as deterministic mappings from available information into admissible actions. One can also define stochastic policies that map available information into probability kernels on $\mathcal{A}_k$, but such policies offer no better performance than deterministic policies for the MDP models discussed here. A policy is said to be *Markov* if the mapping γ_k depends only on the most recent value of the state, s_k. That is,

$$\gamma_k : \mathcal{S}_k \longrightarrow \mathcal{A}_k \tag{2.2}$$

An *admissible* policy is a sequence $\underline{\gamma} \equiv \{\gamma_0, \ldots, \gamma_{N-1}\}$ with the property that $\gamma_k(\mathcal{I}_k) \in \mathcal{A}_k(s_k)$, so that the selected decisions given past information satisfy the constraints imposed by the current state value. An admissible policy generates a random state trajectory $s_k, k = 0, \ldots, s_N$ and an action trajectory $a_k, k = 0, \ldots, N-1$. Associated with each trajectory is a total reward R which is assumed additive across stages:

$$R \equiv R_N(s_N) + \sum_{k=0}^{N-1} R_k(s_k, a_k) \tag{2.3}$$

This additive structure can be exploited to develop efficient algorithms for solving MDPs, as discussed later. Under appropriate conditions on the sets $\mathcal{S}_k, \mathcal{A}_k$, the transition kernels $\mathcal{T}_k(ds'|s,a)$ and the reward functions $R_k(s,a)$, the policy $\underline{\gamma}$ will generate sequences of well-defined random variables corresponding to state, action and reward trajectories. These conditions involve measurability assumptions, and are satisfied in most applications of interest. As discussed in [25], these conditions will be satisfied whenever the sets $\mathcal{S}_k, \mathcal{A}_k$ are countable and the reward functions $R_k(s,a)$ are bounded. Conditions for more general spaces are beyond the scope of this chapter; see [27, 24] for additional details.

Given an admissible policy Γ, the total reward R becomes a well-defined random variable with expectation $\mathbb{E}_\Gamma[R]$. The objective of the problem is to select the admissible policy that maximizes the expected total reward

$$\max \mathbb{E}_{\underline{\gamma}} \left[R_N(s_N) + \sum_{k=0}^{N-1} R_k(s_k, a_k) \right] \tag{2.4}$$

An important result in Markov Decision Processes is that, whenever an optimal admissible policy $\underline{\gamma}$ exists, there exist admissible Markov policies that achieve the same expected reward, and hence are also optimal. In the remainder of this section, we restrict our discussion to Markov policies $\gamma_k(s_k)$.

2.1 Dynamic Programming

The above formulation of a Markov decision problem has several important properties. First, the overall reward can be represented as an additive decomposition of individual rewards over stages. Second, the choice of admissible actions at each stage is not constrained by the states and actions that were generated at previous stages. Under these assumptions, Bellman's Principle of Optimality [19, 18, 25] applies:

DEFINITION 2.2 *Bellman's Principle of Optimality:*

Let $\underline{\gamma}^*$ be an optimal policy in a Markov Decision Problem. Assume that, when using $\underline{\gamma}^*$, the state s_k is reached with positive probability, where $k < N$. Consider the subproblem starting from state s_k at stage K, with the expected reward

$$\max_{\gamma_k,\cdots,\gamma_{N-1}} \mathbb{E}\left[R_N(s_N) + \sum_{i=k}^{N-1} R_i(s_i, a_i)|s_k\right] \tag{2.5}$$

The policy $\{\gamma_k^*, \ldots, \gamma_{N-1}^*\}$ is an optimal policy for this subproblem.

Bellman's Principle of Optimality leads to the dynamic programming algorithm, defined as follows. Consider the subproblem starting from a particular state s_k at stage k, and consider an admissible policy $\underline{\gamma}$. Define the value of policy Γ starting at state s_k, stage k as

$$V_{\underline{\gamma}}(s_k, k) = \mathbb{E}_{\underline{\gamma}}\left[R_N(s_N) + \sum_{i=k}^{N-1} R_i(s_i, \gamma_i(s_i))|s_k\right] \tag{2.6}$$

Define the optimal reward for the subproblem starting at stage K, state s_K as

$$V^*(s_k, k) = \max_{\underline{\gamma} \text{ admissible}} \mathbb{E}_{\underline{\gamma}}\left[R_N(s_N) + \sum_{i=k}^{N-1} R_i(s_i, a_i)|s_k\right] \tag{2.7}$$

The difficulty with (2.7) is that it represents a functional minimization over policies. The main result in dynamic programming is the Bellman equation, which provides a recursive solution for (2.7):

THEOREM 2.3 *For every initial state s_0, the optimal value $V^*(s_0, 0)$ is given by the backward recursion*

$$V^*(s, N) = R_N(s) \tag{2.8}$$

$$V^*(s,k) \quad = \quad \max_{a \in \mathcal{A}_k(s)} R_k(s,a) + \int_{s' \in \mathcal{S}_{k+1}} V^*(s', k+1) \mathcal{T}_k(ds'|a,s), \qquad k = 0, \ldots, N-1 \tag{2.9}$$

If there exist policies $\gamma_k(s)$ such that

$$\gamma_k(s) \in \operatorname*{argmax}_{a \in \mathcal{A}_k(s)} R(s,a) + \int_{s' \in \mathcal{S}_{k+1}} V^*(s', k+1) \mathcal{T}_k(ds'|a,s), k = 0, \ldots, N-1 \tag{2.10}$$

then the policy $\underline{\gamma}^* = \{\gamma_0^*, \ldots, \gamma_{N-1}^*\}$ is an optimal policy.

Bellman's equation decomposes the functional optimization problem over sequences of admissible policies to a sequence of optimizations over admissible actions for each state at each stage.

2.2 Stationary Problems

In many problems of interest, the Markov Decision Problem (MDP) description is stage invariant: The sets $\mathcal{A}_k(s), \mathcal{S}_k$, the reward functions $R_k(s,a)$ and the transition probability kernels $\mathcal{T}_k(ds'|s,a)$ are independent of the stage index. These problems are known as *stationary* MDPs; the number of stages N may be infinite, or may be a decision variable corresponding to choosing to stop the measurement process.

For stationary MDPs, define a *stationary* Markov policy $\Gamma = \{\gamma, \gamma, \ldots\}$, where the policy at any stage does not depend on the particular stage. We refer to stationary policy sequences in terms of the single stage policy γ, the policy that is used at every stage. Stationary MDP formulations often lead to optimal Markov policies which are also stationary, which allows for a simpler, time-invariant implementation of the optimal policy. There are three commonly used MDP formulations that lead to stationary policies with infinite horizon: discounted reward MDPs, total reward MDPs and average reward MDPs. The first two are commonly used models in sensor management; average reward models are seldom used because sensor management problems do not have statistics that are stage invariant over large numbers of stages. By choosing the discount factor or by rewarding stopping, one can approximately limit the horizon of the MDP problem to intervals where the statistics are stage invariant.

2.2.1 Infinite Horizon Discounted Problems.

Consider the case where the number of stages N is infinite. In order to keep the total reward finite, the reward R includes a nonnegative discount factor $\beta < 1$ for future rewards, as

$$R = \sum_{k=0}^{\infty} \beta^k R(s_k, a_k) \tag{2.11}$$

Assume that the rewards $R(s, a)$ are bounded, so that the total discounted cost R is finite. For these problems, Bellman's equation becomes

$$V^*(s) = \max_{a \in \mathcal{A}(s)} R(s,a) + \beta \int_{s' \in \mathcal{S}} V^*(s') \mathcal{T}(ds'|a, s) \tag{2.12}$$

Note that (2.12) does not involve a recursion over stages, unlike (2.9). The connection is given by the following relationship. Define $V^0(s)$ to be a bounded, measurable function of $s \in \mathcal{S}$. Define the sequence of functions $V^n(s)$ as

$$V^n(s) = \max_{a \in \mathcal{A}(s)} R(s,a) + \beta \int_{s' \in \mathcal{S}} V^{n-1}(s') \mathcal{T}(ds'|a, s) \tag{2.13}$$

Then, the sequence $V^n(s)$ converges to $V^*(s)$, the solution of Bellman's equation (2.12). Formally, let $\mathcal{S}$, $\mathcal{A}$ be complete metric spaces, and let $\mathcal{B}(\mathcal{S})$ denote the space of bounded, real-valued functions $f : \mathcal{S} \longrightarrow \mathbb{R}$ with the (essential) supremum norm $\| \cdot \|_\infty$. Define the dynamic programming operator $\mathbf{T} : \mathcal{B}(\mathcal{S}) \longrightarrow \mathcal{B}(\mathcal{S})$ as

$$\mathbf{T}f(s) = \max_{a \in \mathcal{A}(s)} R(s,a) + \beta \int_{s' \in \mathcal{S}} f(s') \mathcal{T}(ds'|a, s) \tag{2.14}$$

and the fixed policy operator

$$\mathbf{T}_\gamma f(s) = R(s, \gamma(s)) + \beta \int_{s' \in \mathcal{S}} f(s') \mathcal{T}(ds'|\gamma(s), s) \tag{2.15}$$

The following result characterizes the important property of the dynamic programming operator:

THEOREM 2.4 *Assume that $R(s, a)$ is bounded and $\beta < 1$. Then, the operator $\mathbf{T}$ is a contraction mapping with contraction coefficient β; i.e., for any functions $V, W \in \mathcal{B}(\mathcal{S})$,*

$$\| \mathbf{T}V - \mathbf{T}W \|_\infty \leq \beta \| V - W \|_\infty \tag{2.16}$$

The contraction mapping theorem [35] guarantees that the sequence $V^n(s)$ converges to a unique fixed point $V = \mathbf{T}(V)$ in $\mathcal{B}(\mathcal{S})$ as $n \to \infty$, where the existence is guaranteed by the completeness of the space $\mathcal{B}(\mathcal{S})$ from any initial estimate of V. This limit satisfies Bellman's equation (2.12). The main results in discounted dynamic programming are summarized below:

THEOREM 2.5 *Assume $R(s, a)$ is bounded and $\beta < 1$. Then,*

1 *For any bounded function $V \in \mathcal{B}(\mathcal{S})$,*

$$V^*(s) = \lim_{n\to\infty} (\mathbf{T}^n V)(s) \tag{2.17}$$

2 *The optimal value function V^* satisfies Bellman's equation (2.12). Furthermore, the solution to the Bellman equation is unique in $\mathcal{B}(\mathcal{S})$.*

3 *For every stationary policy γ and for any $V \in \mathcal{B}(\mathcal{S})$, denote by $\mathbf{T}_\gamma$ the dynamic programming operator when policy γ is the only admissible policy. The expected value achieved by policy γ for each state s, denoted as $V_\gamma(s)$, is the unique solution in $\mathcal{B}(\mathcal{S})$ of the equation.*

$$V_\gamma(s) = (\mathbf{T}_\gamma V_\gamma)(s) = \lim_{n\to\infty} (\mathbf{T}_\gamma^n V)(s) \tag{2.18}$$

4 *A stationary policy γ is optimal if and only if it achieves the maximum reward in the Bellman equation (2.12) for each $s \in \mathcal{S}$; i.e.,*

$$\mathbf{T}_\gamma V^* = \mathbf{T} V^* \tag{2.19}$$

The last property in Proposition 2.5 provides a verification theorem for establishing the optimality of a strategy given the optimal value function.

Although most sensor management applications will not have an infinite horizon, a discounted infinite horizon model is often used as an approximation to generate stationary policies. The choice of discount factor in these problems sets an "effective" horizon that can be controlled to reflect the number of stages in the problem of interest.

2.2.2 Undiscounted Total Reward Problems. Many sensor management applications are best formulated as total reward problems where the number of stages N can be infinite and no future discounting of value is used. For instance, consider a search problem where there is a finite number of areas to be searched, each of which may have an object present with a known probability in each area. When there is a cost associated with searching an area and a reward for finding objects in areas, an optimal strategy will search only the subset of areas where the expected reward of searching an area exceeds the expected cost of the search. In this problem, an admissible decision is to stop searching; however, the number of stages before the search is stopped depends on the stage evolution of the probabilities of areas containing objects, and is not specified *a priori*.

In undiscounted total reward problems, the accumulated reward may become unbounded. Hence, additional structure is needed in the rewards to guarantee that optimal strategies exist, and that the optimal value function remains finite. Following the exposition in [25], we specify two alternative assumptions for the undiscounted problems.

DEFINITION 2.6 *Assumption* **P***: The rewards per stage satisfy:*

$$R(s,a) \leq 0, \text{ for all } s \in \mathcal{S}, a \in \mathcal{A}(s) \tag{2.20}$$

DEFINITION 2.7 *Assumption* **N***: The rewards per stage satisfy:*

$$R(s,a) \geq 0, \text{ for all } s \in \mathcal{S}, a \in \mathcal{A}(s) \tag{2.21}$$

In discounted reward problems, the presence of a discount factor limits the effect of future rewards on the current choice of actions to a limited interval. In contrast, undiscounted problems must consider effects of long-term rewards. Thus, under Assumption **P**, the goal must be to bring the state quickly to a region where one can either terminate the problem or where the rewards approach 0. Under assumption **N**, the objective may be to avoid reaching a termination state for as long as possible.

As before, define the dynamic programming operator $\mathbf{T} : \mathcal{B}(\mathcal{S}) \longrightarrow \mathcal{B}(\mathcal{S})$ as

$$\mathbf{T}f(s) = \max_{a \in \mathcal{A}(s)} R(s,a) + \int_{s' \in \mathcal{S}} f(s')\mathcal{T}(ds'|a,s) \tag{2.22}$$

and the iteration operator for a stationary policy γ as $\mathbf{T}_\gamma$ as

$$\mathbf{T}_\gamma f(s) = R(s,\gamma(a)) + \int_{s' \in \mathcal{S}} f(s')\mathcal{T}(ds'|\gamma(a),s) \tag{2.23}$$

A key difference with the discounted rewards problem is that the operators $\mathbf{T}$, $\mathbf{T}_\gamma$ are no longer contractions. This raises issues as to the existence and uniqueness of optimal value functions and characterization of optimal strategies. The principal results for these problems are summarized below.

Under Assumption **P** or **N**, Bellman's equation holds, and becomes

$$V^*(s) = \max_{a \in \mathcal{A}(s)} R(s,a) + \int_{s' \in \mathcal{S}} V^*(s')\mathcal{T}(ds'|a,s) \tag{2.24}$$

Similarly, for any stationary policy γ, one has the property that

$$V_\gamma(s) = \mathbf{T}_\gamma V_\gamma(s) \tag{2.25}$$

Although Bellman's equation (2.24) holds, the solution may not be unique. However, the optimal value is either the largest solution (under Assumption **P**) or the smallest solution (under Assumption **N**). Specifically, under Assumption **P**, if V satisfies $V \le 0$ and $V \le \mathbf{T}V$, then $V \le V^*$. Similarly, under Assumption **N**, if V satisfies $V \ge 0$ and $V \ge \mathbf{T}V$, then $V \ge V^*$.

Characterization of optimal strategies differs for the two cases **P** and **N**. Under Assumption **P**, a stationary policy γ is optimal if and only if it achieves the maximum reward in Bellman's equation (2.24):

$$\mathbf{T}_\gamma V^*(s) = \mathbf{T}V^*(s) \text{ for all } s \in \mathcal{S} \tag{2.26}$$

However, the sufficiency clause is not true under Assumption **N**. A different characterization of optimality is needed: under Assumption **N**, a stationary policy is optimal if and only if

$$\mathbf{T}_\gamma V_\gamma(s) = \mathbf{T}V_{\gamma(s)} \text{ for all } s \in \mathcal{S} \tag{2.27}$$

Another important consequence of losing the contraction property is that there may be no iterative algorithms for computing the optimal value function from arbitrary initial estimates. Fortunately, when the iteration is started with the right initial condition, one can still get convergence. Specifically, let $V^0(s) = 0$ for all $s \in \mathcal{S}$, and define the iteration

$$V^n = \mathbf{T}V^{n-1},\ n = 1, 2, \ldots \tag{2.28}$$

Under Assumption **P**, the operator **T** generates a monotone sequence

$$V^0 \ge V^1 \ge \ldots \ge V^n \ge \cdots, \tag{2.29}$$

which has a limit (with values possibly $-\infty$) V^∞. Similarly, under Assumption **N**, the sequence generated by (2.28) yields

$$V^0 \le V^1 \le \ldots \le V^n \le \cdots \tag{2.30}$$

These limits become the optimal values under simple conditions, as indicated below:

THEOREM 2.8 *Under Assumption* **P**, *if* $V^\infty = \mathbf{T}V^\infty$ *for all* $s \in \mathcal{S}$, *and* $V^0 \ge V \ge V^*$, *then*

$$\lim_{n\to\infty} \mathbf{T}^n V^0 = V^* \tag{2.31}$$

Under Assumption **N**, *if* $V^0 \le V \le V^*$,

$$\lim_{n\to\infty} \mathbf{T}^n V = V^* \tag{2.32}$$

Conditions that guarantee that $V^\infty = \mathbf{T}V^\infty$ under Assumption **N** are compactness of the action sets $\mathcal{A}(s)$ for each s [25].

Important classes of sensor management problems that satisfy the assumptions in this subsection are optimal stopping problems, which are described in the Appendix, Section 3. In these problems, one must choose between a finite number of sensing actions, each of which has a cost that depends on the action and state $c(a, s)$, or stopping the sensing problem and receive a cost $s(a)$. For instance, the dynamic classification problems using multimodal sensing considered in [47, 48] fit this structure: there is a cost for using sensing time for each mode, and when the decision to stop sensing is made, the system undergoes a cost related to the probability of misclassification of each target. A similar formulation was used in [119] for underwater target classification using multiple views. This example is discussed in Section 5.

One can formulate these optimal stopping problems as maximization problems, redefining rewards $R(a, s) = -c(a, s)$, plus adding an additional termination state t to the state space $\mathcal{S}$, corresponding to the action of termination. Once the problem reaches a termination state, the only possible action is to continue in this state, incurring no further rewards. The resulting problem fits the model of Assumption **P** above.

2.3 Algorithms for MDPs

Bellman's equation (2.9) provides a recursive algorithm for computation of the optimal value function in finite horizon MDPs. For infinite horizon problems, Bellman's equation represents a functional equation for the optimal value function $V^*(s)$. The most common algorithm for computing $V^*(s)$ is known as *value iteration*. It consists of starting from a guess at the value function $V^0(s)$ and generating a sequence $V^n(s)$ using the iteration (2.19). For discounted reward problems, Theorem 2.5 establishes that this sequence converges to $V^*(s)$ from any initial condition. For total reward problems, the value iteration algorithm converges to $V^*(s)$ provided the initial condition $V^0(s)$ is chosen appropriately, as in Theorem 2.8.

In discounted reward problems, an optimal policy γ^* can be generated using (2.19) as

$$\gamma^*(s) \in \arg\max_{a \in \mathcal{A}} R(s, a) + \beta \int_{s' \in \mathcal{S}} V^*(s') \mathcal{T}(ds'|a, s). \tag{2.33}$$

For total reward problems under Assumption **P**, a similar characterization follows from (2.26). Assumption **P** is the most appropriate model for sensor management with stopping criteria, as illustrated in the example later in this chapter.

Another approach for the solution of discounted reward problems is *policy iteration*. In policy iteration, the algorithm starts with a stationary policy γ^0. Given a policy γ^n, Bellman's equation for a single policy is then solved to obtain the optimal value function $V_{\gamma^n}(s)$, as

$$V_{\gamma^n}(s) = R(s, \gamma^n(s)) + \beta \int_{s' \in \mathcal{S}} V_{\gamma^n}(s') \mathcal{T}(ds' | \gamma^n(s), s) \qquad (2.34)$$

A new policy γ^{n+1} is then generated as

$$\gamma^{n+1}(s) \in \arg\max_{a \in \mathcal{A}} R(s, a) + \beta \int_{s' \in \mathcal{S}} V_{\gamma^n}(s') \mathcal{T}(ds' | a, s) \qquad (2.35)$$

The policy iteration algorithm requires many fewer iterations than value iteration to converge. However, each iteration is more complex, as it requires the solution of a functional equation to obtain the value of a single policy. An approximate form of policy iteration is often used, where (2.34) is solved approximately using a few value iterations.

3. Partially Observed Markov Decision Problems

The primary assumption underlying the MDP theory discussed in the previous section is that the state of the system, s_k, is observed perfectly at each stage k. In many sensor management applications, the full state is not observed at each stage; instead, some statistics related to the underlying state are observed, which yield uncertain information about the state. These problems are known as Partially Observed Markov Decision Problems (POMDPs) as the observations yield only partial knowledge of the true state. POMDPs are also known as partially observable Markov decision processes.

DEFINITION 2.9 *A Partially Observed Markov Decision Process (POMDP) consists of*

- *A discrete stage index* $k \in \{0, 1, \ldots, N\}, N \leq \infty$
- *A finite set of possible states* $\mathcal{S}_k$ *for each stage index* k*, with cardinality* $|\mathcal{S}_k|$
- *An initial probability distribution* $\pi_0(s)$ *over the finite set* $\mathcal{S}_0$*, where* $\pi_0(s) \equiv \mathrm{P}(s_0 = s)$.

- *A finite set of possible actions $\mathcal{A}_k$ for each stage index k*
- *State transition probability matrices $\mathcal{T}_k(s'|s,a)$, where $\mathcal{T}_k(s'|s,a) \equiv \mathrm{P}(s_{k+1} = s'|s_k = s, a_k = a)$,*
- *A real-valued single stage reward function $R_k(s,a)$ and an overall objective J which is additive across stages*
- *A finite set of possible observations $\mathcal{Y}_k$ for each stage k*
- *An observation probability likelihood $\mathcal{Q}_k(y|s,a)$, where*

$$\mathcal{Q}_k(y|s,a) \equiv \mathrm{P}(y_k = y|s_k = s, a_k = a)$$

Unlike the MDP model, the sets $\mathcal{S}_k$, $\mathcal{A}_k$ and the observation sets $\mathcal{Y}_k$ are assume to be finite. The initial distribution $\pi_0(s)$ and the transition probability distributions $\mathcal{T}(s'|s,a)$ define a controlled Markov chain given a sequence of decisions $a_0, a_1, \ldots$. The observations y_k are assumed to be conditionally independent for different k, with distribution depending only on the current state s_k and the current action a_k. Note that the choice of action a_k affects the generation of observations; this model represents the sensor management problem, where choice of sensing actions determine what information is collected.

We assume that past decisions, plus past values of the observations, are available to select the choice of next decision. Let $\mathcal{I}_k$ denote the information available at stage k for selecting action a_k, defined as:

$$\mathcal{I}_k = \{a_0, y_0, \ldots, a_{k-1}, y_{k-1}\} \tag{2.36}$$

Note that this information does not include any observations of past states. Due to the finite assumption on the action and observation spaces, the number of possible information sets $\mathcal{I}_k$ is also finite. As in MDPs, a *policy* at stage k is a deterministic mapping $\gamma_k(\mathcal{I}_k)$, from the set of all possible information states $\{\mathcal{I}_k\}$ to $\mathcal{A}_k$. Such policies are *causal*, in that they select current actions based on past information only. Similarly, an *admissible policy* for POMDPs is a sequence $\Gamma \equiv \{\gamma_0, \ldots, \gamma_{N-1}\}$ with the property that $\gamma_k(\mathcal{I}_k) \in \mathcal{A}_k$.

An admissible policy generates a random state trajectory $s_k, k = 0, \ldots, s_N$, an observation trajectory $y_k, k = 0, \ldots, N-1$ and an action trajectory $a_k, k = 0, \ldots, N-1$. The causal sequence corresponds to the following: Given information $\mathcal{I}_k$, the policy generates an action $a_k = \gamma_k(I_k)$. Given this action a_k, a new observation y_k is collected, and the state transitions from s_k to s_{k+1}. The action a_k and the observation y_k are added to the information set $\mathcal{I}_k$ to generate $\mathcal{I}_{k+1}$. This causal chain is illustrated in figure 2.2. The policy $\underline{\gamma}$ will generate sequences of well-defined random variables corresponding to the state, action and reward trajectories.

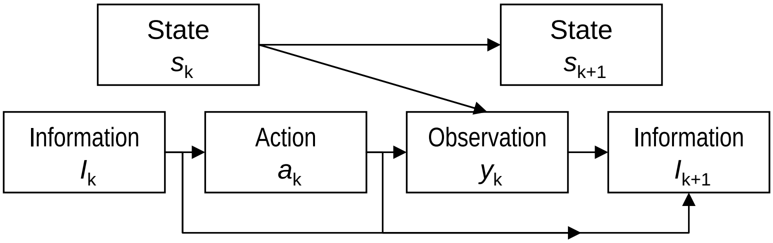


Figure 2.2. Illustration of sequential actions and measurements

Similar to the MDP formulation, there is a total reward R associated with these trajectories that is additive across stages:

$$R \equiv R_N(s_N) + \sum_{k=0}^{N-1} R_k(s_k, a_k) \tag{2.37}$$

Given an admissible policy Γ, the total reward R becomes a well-defined random variable with expectation $\mathbb{E}_\Gamma[R]$. The objective of POMDP problem is to select an admissible policy that maximizes the expected total reward

$$\max \mathbb{E}_{\underline{\gamma}} \left[R_N(s_N) + \sum_{k=0}^{N-1} R_k(s_k, a_k) \right] \tag{2.38}$$

3.1 MDP Representation of POMDPs

The POMDP problem can be converted to a standard MDP problem, where the underlying state corresponds to the information sets $\mathcal{I}_k$. Note that these sets take values in discrete sets, and evolve according to the simple evolution

$$\mathcal{I}_{k+1} = \mathcal{I}_k \cup \{a_k, y_k\} \tag{2.39}$$

To show the equivalent MDP problem, consider the objective of maximizing the expected reward. For any policy Γ, the smoothing property of conditional expectations yields

$$\mathbb{E}_{\underline{\gamma}} \left[R_N(s_N) + \sum_{k=0}^{N-1} R_k(s_k, a_k) \right] \tag{2.40}$$
$$= \mathbb{E}_{\underline{\gamma}} \left[\mathbb{E}_{\underline{\gamma}} \left[R_N(s_N) | \mathcal{I}_N \right] + \sum_{k=0}^{N-1} \mathbb{E}_{\underline{\gamma}} \left[R_k(s_k, a_k) | \mathcal{I}_k, a_k \right] \right]$$

Define the equivalent reward function $\hat{R}(\mathcal{I}_k, a_k)$ as

$$\hat{R}(\mathcal{I}_k, a_k) = \mathbb{E}_{\underline{\gamma}}\left[R_k(s_k, a_k)|\mathcal{I}_k, a_k\right] \tag{2.41}$$

Note that the conditional expectation in (2.41) does not depend on the specific strategies $\underline{\gamma}$, since all the past action values and past observation values are part of $\mathcal{I}_k$. To show equivalence to an MDP, we have to establish that (2.39) generates a Markov transition probability kernel that describes the random evolution of $\mathcal{I}_{k+1}$ given $\mathcal{I}_k$. This requires that

$$\mathrm{P}(\mathcal{I}_{k+1}|\mathcal{I}_k, a_k, \mathcal{I}_{k-1}, \ldots, \mathcal{I}_0) = \mathrm{P}(\mathcal{I}_{k+1}|\mathcal{I}_k, a_k) \tag{2.42}$$

To show this is true, note that the only conditionally random component of $\mathcal{I}_{k+1}$ given $\mathcal{I}_k, a_k$ is the observation y_k. Thus, one has to show that

$$\mathrm{P}(y_k|a_k, \mathcal{I}_k, \mathcal{I}_{k-1}, \ldots, \mathcal{I}_0) = \mathrm{P}(y_k|a_k, \mathcal{I}_k), \tag{2.43}$$

which follows because $\mathcal{I}_j \subset \mathcal{I}_k, j < k$.

The above argument establishes that a POMDP is equivalent to an MDP with states corresponding to the information sets $\mathcal{I}_k$, and Markov dynamics corresponding to (2.39), and Markov transition probabilities given by (2.42). Hence, all of the results discussed in the previous section can be applied to this problem. In particular, for N finite, there exists an optimal value function $V^*(\mathcal{I}_k, k)$ that satisfies Bellman's equation

$$V^*(\mathcal{I}_k, k) = \max_{a \in \mathcal{A}_k} \hat{R}(\mathcal{I}_k, a) + \mathbb{E}_{y_k}\left[V^*(\mathcal{I}_k \cup \{y_k, a\}, k+1)\right] \tag{2.44}$$

and the optimal strategies satisfy

$$\gamma_k^*(\mathcal{I}_k) \in \operatorname*{argmax}_{a \in \mathcal{A}_k} \hat{R}(\mathcal{I}_k, a) + \mathbb{E}_{y_k}\left[V^*(\mathcal{I}_k \cup \{y_k, a\}, k+1)\right] \tag{2.45}$$

However, it is difficult to extend this characterization to infinite horizon problems because the set of possible information sets $\mathcal{I}_k$ grows larger as k increases. An alternative parameterization is possible using the concept of sufficient statistics for control [225, 24, 25]. A *sufficient statistic* is a function $h_k(\mathcal{I}_k)$ such that the maximization in (2.45) is achieved by a function of $h_k(\mathcal{I}_k)$ instead of all of $\mathcal{I}_k$. To show that a statistic is sufficient for control, it is enough to show that the right hand side of (2.40) depends on $\mathcal{I}_k$ only through $h_k(\mathcal{I}_k)$. A sufficient statistic for stochastic control problems such as POMDPs is the posterior distribution $\pi_k(s) = \mathrm{P}(s_k = s|\mathcal{I}_k)$, the conditional probability that the state s_k takes the value s given past information $\mathcal{I}_k$.

Given a bounded function $g : \mathcal{S}_k \to \mathbb{R}$, we use the notation

$$< g, \pi_k >= \mathbb{E}\left[g(s_k)|\mathcal{I}_k\right] = \sum_{s \in \mathcal{S}_k} g(s)\pi_k(s). \tag{2.46}$$

Since $\mathcal{S}_k$ is a finite set, these bounded functions are real-valued $|\mathcal{S}_k|$ dimensional vectors, and $< \cdot >$ is an inner product. Define the function $r_k(a_k)(s_k) \equiv R(s_k, a_k)$. Then, (2.40) becomes

$$\mathbb{E}_{\underline{\gamma}}\left[R_N(s_N) + \sum_{k=0}^{N-1} R_k(s_k, a_k)\right] = \mathbb{E}_{\underline{\gamma}}\left[< R_N, \pi_N > + \sum_{k=0}^{N-1} < r(a_k), \pi_k >\right],$$

which establishes that the sequence $\pi_k, k = 0, \ldots, N$ is a sufficient statistic for POMDPs.

Given the independence assumptions of the POMDP model, one can compute a controlled Markov evolution for the sufficient statistic π_k as follows:

$$\begin{aligned}
\pi_k(s_k) &= \mathrm{P}(s_k|\mathcal{I}_k) = \mathrm{P}(s_k|\mathcal{I}_{k-1}, a_{k-1}, y_{k-1}) \\
&= \sum_{s_{k-1}} \mathrm{P}(s_k, s_{k-1}|\mathcal{I}_{k-1}, a_{k-1}, y_{k-1}) \\
&= \sum_{s_{k-1}} \mathrm{P}(s_k|s_{k-1}, \mathcal{I}_{k-1}, a_{k-1}, y_{k-1})\, \mathrm{P}(s_{k-1}|\mathcal{I}_{k-1}, a_{k-1}, y_{k-1}) \\
&= \sum_{s_{k-1}} \mathcal{T}(s_k|s_{k-1}, a_{k-1})\, \mathrm{P}(s_{k-1}|\mathcal{I}_{k-1}, a_{k-1}, y_{k-1}),
\end{aligned} \tag{2.47}$$

where the last equality follows from the Markov evolution of s_k given the actions a_k. We can further simplify the right hand side using Bayes' rule as

$$\begin{aligned}
\mathrm{P}(s_{k-1}|\mathcal{I}_{k-1}, a_{k-1}, y_{k-1}) &= \frac{\mathrm{P}(s_{k-1}, y_{k-1}|\mathcal{I}_{k-1}, a_{k-1})}{\mathrm{P}(y_{k-1}|\mathcal{I}_{k-1}, a_{k-1})} \\
&= \frac{\mathrm{P}(y_{k-1}|s_{k-1}, a_{k-1})\, \mathrm{P}(s_{k-1}|\mathcal{I}_{k-1})}{\mathrm{P}(y_{k-1}|\mathcal{I}_{k-1}, a_{k-1})} \\
&= \frac{\mathcal{Q}_{k-1}(y_{k-1}|s_{k-1}, a_{k-1})\pi_{k-1}(s_{k-1})}{\sum_{\sigma \in \mathcal{S}_{k-1}} \mathcal{Q}_{k-1}(y_{k-1}|\sigma, a_{k-1})\pi_{k-1}(\sigma)}
\end{aligned} \tag{2.48}$$

The resulting evolution is given by

$$\begin{aligned}
\pi_k(s) &= \sum_{s_{k-1} \in \mathcal{S}_{k-1}} \mathcal{T}_{k-1}(s|s_{k-1}, a_{k-1}) \frac{\mathcal{Q}_{k-1}(y_{k-1}|s_{k-1}, a_{k-1})\pi_{k-1}(s_{k-1})}{\sum_{\sigma \in \mathcal{S}_{k-1}} \mathcal{Q}_{k-1}(y_{k-1}|\sigma, a_{k-1})\pi_{k-1}(\sigma)} \\
&\equiv \hat{\mathcal{T}}_{k-1}(\pi_{k-1}, y_{k-1}, a_{k-1})
\end{aligned} \tag{2.49}$$

This evolution is a function of the causal dependencies depicted in Figure 2.2. A different order can be used (e.g. [173, 248, 162]) where the information set $\mathcal{I}_k$ includes the observation y_k, which is generated as depending on the states s_k and action a_{k-1}. The reason for the difference is that, in sensor management problems, decisions are chosen primarily to control the measurements obtained, and not the underlying state s_k. In contrast, standard POMDP formulations focus on using decisions to control the underlying state evolution.

Using sufficient statistics allows us to define an equivalent MDP with state π_k at stage k, and objectives (2.44). In the POMDP literature, these states are referred to as *information states* or *belief states* (See Appendix, Section 2). In terms of these information states, Bellman's equation (2.44) becomes

$$V^*(\pi, k) = \max_{a \in \mathcal{A}_k} < r_k(a), \pi > + \sum_{y \in \mathcal{Y}} V^*(\hat{\mathcal{T}}_k(\pi, y, a), k+1)\, \mathrm{P}(y|\mathcal{I}_k, a), \tag{2.50}$$

where

$$\mathrm{P}(y|\mathcal{I}_k, a) \equiv P_k(y|\pi_k, a) = \sum_{s' \in \mathcal{S}_k} \mathcal{Q}_k(y|s', a)\pi_k(s') \tag{2.51}$$

3.2 Dynamic Programming for POMDPs

The use of sufficient statistics allows for a constant-dimension representation of the underlying MDP state as the horizon increases. The information state now takes values in $\Pi_{|\mathcal{S}_k|}$, the simplex of probability distributions on $\mathcal{S}_k$. When $\mathcal{S}_k \equiv \mathcal{S}$ and $\mathcal{A}_k \equiv \mathcal{A}$ are constant in k, and the transition probabilities $\mathcal{T}_k$, measurement probabilities $\mathcal{Q}_k$ and rewards $R(s, a)$ do not depend on k, one can define stationary problems as in MDPs with infinite horizons, with or without discounting, and apply the MDP theory to the POMDP problem with the information state representation. The resulting discounted cost Bellman equation with discount factor β is

$$V^*(\pi) = \max_{a \in \mathcal{A}} < r(a), \pi > + \beta \sum_{y \in \mathcal{Y}} V^*(\hat{\mathcal{T}}_k(\pi, y, a))\, \mathrm{P}(y|\pi, a) \tag{2.52}$$

The equivalent MDP using information states has a special structure where the immediate reward associated with an action is a linear function of the state. Sondik [219, 218, 220] exploited this property to obtain a unique characterization for the solution of Bellman's equation (2.50). Sondik observed that

$$V^*(\pi, N) =< r_N, \pi > \tag{2.53}$$

is a linear function of π. Analyzing the recursion, this leads to the conjecture that, for $k < N$, there exists a set of real-valued vectors $\mathcal{H}_k$, of dimension $|\mathcal{S}_k|$, such that

$$V^*(\pi, k) = \max_{h \in \mathcal{H}_k} < h, \pi >, \tag{2.54}$$

which implies that $V^*(\pi, k)$ is a piecewise linear, convex function of π. This can be established by induction, as it is true for $k = N$. Assuming the inductive hypothesis that such a representation is valid at $k + 1$, and $\mathcal{H}_{k+1}$ is known, Bellman's equation yields

$$\begin{aligned} V^*(\pi, k) &= \max_{a \in \mathcal{A}_k} < r_k(a), \pi > + \sum_{y \in \mathcal{Y}_k} V^*(\hat{\mathcal{T}}_k(\pi, y, a), k+1) P_k(y|\pi, a) \\ &= \max_{a \in \mathcal{A}_k} < r_k(a), \pi > + \sum_{y \in \mathcal{Y}_k} \max_{h \in \mathcal{H}_{k+1}} < h, \hat{\mathcal{T}}_k(\pi, y, a) > P_k(y|\pi, a) \\ &= \max_{a \in \mathcal{A}_k} < r_k(a), \pi > + \sum_{y \in \mathcal{Y}_k} \max_{h \in \mathcal{H}_{k+1}} < h, \sum_{s' \in \mathcal{S}_k} \mathcal{T}_k(\cdot|s, a) \mathcal{Q}_k(y|s, a) \pi(s) > \end{aligned}$$

where the denominator in $\hat{\mathcal{T}}$ cancels the multiplication by $P_k(y|\pi, a)$. This can further be simplified as

$$V^*(\pi, k) = \max_{a \in \mathcal{A}_k} \sum_{y \in \mathcal{Y}_k} \max_{h \in \mathcal{H}_{k+1}} < \frac{r_k(a)}{|\mathcal{Y}_k|} + \sum_{\sigma \in \mathcal{S}_k} h(\sigma) \mathcal{T}_k(\sigma|\cdot, a) \mathcal{Q}_k(y|\cdot, a), \pi >, \tag{2.55}$$

where $|\mathcal{Y}_k|$ is the number of possible observation values. Note that the sum of a finite number of piecewise linear, convex functions of π is also a piecewise linear convex function, and the maximum of a finite number of piecewise linear convex functions is again a piecewise linear, convex function, which establishes the induction. Furthermore, a new set $\mathcal{H}_k$ containing the needed linear support functions can be computed as

$$\begin{aligned} \mathcal{H}_k = \{ h \in \mathcal{B}(\mathcal{S}_k) : h(s) = R_k(a, s) + \sum_{y \in \mathcal{Y}_k} \sum_{\sigma \in \mathcal{S}_{k+1}} h^y_{k+1}(\sigma) \cdot \\ \mathcal{T}_k(\sigma|s, a) \mathcal{Q}_k(y|s, a) \text{ for some } h^y_{k+1} \in \mathcal{H}_{k+1}, a \in \mathcal{A}_k \} \end{aligned} \tag{2.56}$$

The set $\mathcal{H}_k$ defined above contains far more linear support functions than are necessary to define $V^*(\pi, k)$ in (2.54); specifically, there are many functions $h \in \mathcal{H}_k$ for which there is no information state π such that $V^*(\pi, k) = < h, \pi >$. Thus, efficient algorithms for solution of POMDP problems focus on finding small subsets of $\mathcal{H}_k$ which are sufficient for defining $V^*(\pi, k)$. The details of these algorithms are beyond the scope of this overview chapter, and can be found in review articles such as [162, 45, 46, 159, 189, 173]. The key in all of these algorithms is that, given a specific information state π at stage k

and the set $\mathcal{H}_{k+1}$ of support functions, one can use (2.55) to construct the linear support function h for which $V^*(\pi, k) =< h, \pi >$. By judiciously choosing the information states π for which this is done, one can generate a minimal set of support functions for defining $V^*(\pi, k)$ recursively. In general, the number of linear support pieces still grows exponentially with the number of stages, limiting the application of numerical techniques to small horizon problems or problems with special structure.

4. Approximate Dynamic Programming

The dynamic programming algorithms described above often require extensive computation to obtain an optimal policy. For MDPs, one has to compute a value function indexed by the number of states, which could be uncountable. For POMDPs with finite state spaces, the value functions depend on the information states, which are probability vectors of dimension equal to the number of states. This has led to a number of *Approximate Dynamic Programming* (ADP) techniques [16, 28, 26] that are used to reduce the required computations. Such approximations form the basis for the results in Chapters 7 and 5. We discuss the nature of these approximations briefly for the case of discounted infinite horizon MDPs.

The foundation for most ADP techniques is the characterization of optimal strategies in (2.33)

$$\gamma^*(s) \in \arg\max_{a \in \mathcal{A}} R(s,a) + \beta \int_{s' \in \mathcal{A}} V^*(s') \mathcal{T}(ds'|a,s) \tag{2.57}$$

If the optimal value function $V^*(s)$ were available, one can compute the optimal action at the current state s by performing the above maximization. ADP techniques compute an approximation $\tilde{V}^*(s)$ to the optimal value function $V^*(s)$ and use (2.57) to generate decisions for each state.

There are three classes of ADP techniques commonly used in the literature. *Offline learning* techniques [16, 230] use simulation and exploratory strategies such as temporal difference learning to learn functional approximations $\tilde{V}^*(s)$ to the optimal value function. These are typically used for problems where the dynamical model is well-understood; a powerful application was demonstrated by Tesauro [230] in the context of backgammon. *Rollout* techniques [26, 207] use real-time simulation of suboptimal policies to evaluate an approximation to the future expected reward in (2.57); these are typically used when the problem instance is not known *a priori*, so that simulations are not readily implemented. *Problem Approximation* techniques [49, 48, 245, 205] use the exact value function computed for an approximate problem with special structure as surrogates for the optimal value function. These techniques exploit special classes of stochastic control problems that have simple solutions, such as the multi-armed bandit problems of Chapter 6 or one-step lookahead problems.

The effectiveness of ADP depends on the choice of technique and the problem structure. Specific ADP techniques tailored to sensor management problems are discussed in greater detail in Chapters 5 and 7.

5. Example

We conclude this chapter with an example consisting of selecting measurements to classify underwater elastic targets at unknown orientations using active acoustic sensors, described in greater detail in [119]. The scattered fields from each target depend on target type and the target-sensor orientation [200]. Typically, there are contiguous ranges of orientations for which the scattering physics is relatively stationary for each target type. Assuming that the targets of interest are rotationally symmetric, and the scattered fields are observed in a plane bisecting the axis of symmetry, the scattered fields at a fixed radial distance are characterized by a single orientation angle ϕ and the target type.

We model this problem as a POMDP. The underlying discrete states $\mathcal{S}_k$ consist of five target types and five discrete orientation bins from 0 to 90^o. Actions correspond to taking a measurement of the object from a given angular position in a fixed coordinate system. Assuming that the objects are moving, there is a Markov model for transitions from one relative orientation bin to another given measurements, since moving the sensor to a different angular location will change the relative orientation. Furthermore, there will be some random relative angular changes created by target motion. This is captured by a finite state Markov chain model that depends on the chosen action (measurement position).

A sensing action a at stage k corresponds to selecting a change in relative measurement angle $\Delta\Phi$ for movement of the sensor position from its previous position. As discussed above, there is a state transition probability $\mathcal{T}_k(s'|s,a)$ associated with this action. Furthermore, this action generates an observation y_k which is related to the underlying state: the true object type and the quantized relative observation angle. To continue the development of the POMDP formulation, one must describe the finite set of possible values $\mathcal{Y}$, and the observation probability likelihoods $\mathcal{Q}_k(y|s,a)$. For this application, we collected measured scattered fields for each target as a function of relative angle, with data sampled in 1° increments. Figure 2.3 shows a plot of the magnitude of the discrete Fourier transform of the measured scattered fields, for two of the five targets, as a function of relative sensing angle ϕ. The time-domain scattered fields from each target were processed using matching pursuits [200, 201, 169] to extract a set of feature vectors. The feature vectors were collected across all target-sensor orientations and target types, and vector quantization (VQ) was performed [96], leading to a finite number of possible observation values $\mathcal{Y}$. The error statistics of the vector quantization algorithm were used to generate the observation likelihoods $\mathcal{Q}(y|s,a)$, which were assumed to be stationary. For these experiments, the number of VQ codes (possible observations) was 25. The number of possible observation directions was discretized to 11 directions, at increments of $5°$, with a maximum displacement of $50°$.

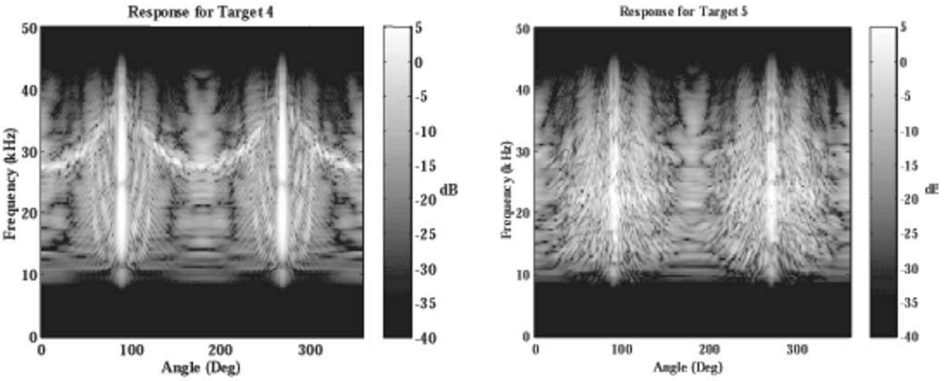


Figure 2.3. Scattered fields (magnitude) as a function of sensing angle.

After performing k observation actions, starting at stage 0, the information state π_k can be computed as in (2.49). To complete the POMDP description, one needs a description of the objective. Let C_{uv} denote the cost of declaring the object under interrogation to be target u, when in reality it is target v, where u and v are members of the five targets of interest. Given the collected information, one can choose to make a classification decision based on the current information, or to continue collecting information. If one chooses to make a classification decision, the selected label will be the one that minimizes the Bayes cost given the available information:

$$\text{Target class} = \arg\min_{u} \sum_{v=1}^{5} C_{uv} \sum_{s \in \mathcal{S}_v} \pi_k(s), \tag{2.58}$$

where $\mathcal{S}_v$ is the set of discrete states associated with target v. In terms of the maximization formulation discussed previously, a classification decision incurs a negative reward, corresponding to the expected Bayes cost. Note that this increases the set of potential control actions at any stage $\mathcal{A}_k$, as one can choose to either select a new measurement, or make one of 5 classification decisions. Making a classification decision places the state into an absorbing state, from which there are no further transitions or rewards associated with the state.

In addition to the classification costs, there are costs associated with any sensing action, which may depend on the cost of moving the relative angle displacement $\Delta\phi$. This cost will be independent of the underlying state s_k of the system, and is set to the value 1 in the results below, independent of the angle displacement. Using negative costs as rewards, this implies that, for sensing actions a, $R_k(s,a) \equiv -1$ for all $s \in \mathcal{S}$. For classification actions $a = u$, $R_k(s,a) \equiv -C_{u,v}$ if $s \in \mathcal{S}_v$. The classification costs C_{uv} have a uniform penalty for errors as $C_{uv} = C_c$ for all $u \neq v$, and $C_{uu} = -10$ (a reward of 10 is obtained upon correct classification); the error penalty will be varied in the experiments. In terms of overall reward R, the goal is to maximize the discounted infinite horizon reward

$$R = \mathbb{E}\left[\sum_{k=0}^{\infty} \beta^k R(s_k, a_k)\right], \tag{2.59}$$

where the discount factor β is chosen to be 0.99.

Denote by s_0 the special state in $\mathcal{S}$ which results from making a final classification decision. Note that $\pi_k(s_0) = 0$ until a classification decision is made, and $\pi_k(s_0) = 1$ for all k after a classification decision is made. Let $\mathcal{A}_m \subset \mathcal{A}$ be the admissible decisions to collect further measurements, and $\mathcal{A}_c \subset \mathcal{A}$ be the admissible decisions to make a classification.

Bellman's equation for this POMDP is given by (2.50), for all π such that $\pi(s_0) = 0$, as

$$V^*(\pi) = \max[\max_{a\in\mathcal{A}_m} < R(s,a), \pi >, -1 + \beta \max_{a\in\mathcal{A}_c} \sum_{y\in\mathcal{Y}} V^*(\hat{\mathcal{T}}(\pi, y, a))\, \mathrm{P}(y|\pi, a)]$$

This can be solved using the value iteration algorithm, starting from the piecewise linear approximation $V(\pi) = \max_{a\in\mathcal{A}_m} < R(s,a), \pi >$. After a finite number of iterations, the value function will be a piecewise linear, convex function (cf. (2.55)).

In the results below, the optimal value function was computed approximately using the point-based value iteration (PBVI) algorithm [189], which limits the number of linear support functions to those which support the value function at a given finite set of information states $\pi \in \Pi_{|\mathcal{S}|}$. This yields a lower bound on the optimal value function, which converges to the optimal value function as the discrete set of information states increases to fill the space of information states [162].

The POMDP formulation discussed above yields a non-myopic multi-stage sensor management policy, mapping belief states into actions that trade off immediate rewards for acquiring information to make better decisions. It is an adaptive stopping policy, in that the decision to make a final classification depends on the information state. However, its computational complexity can be significant unless the number of discrete linear support functions is restricted.

Alternative sensor management policies with reduced computation requirements can be developed using different POMDP formulations. For instance, one can consider an alternative adaptive policy, generated by a one-step lookahead POMDP: at each stage k, given an information state π_k that has not yet reached the classification state, select action a_k as

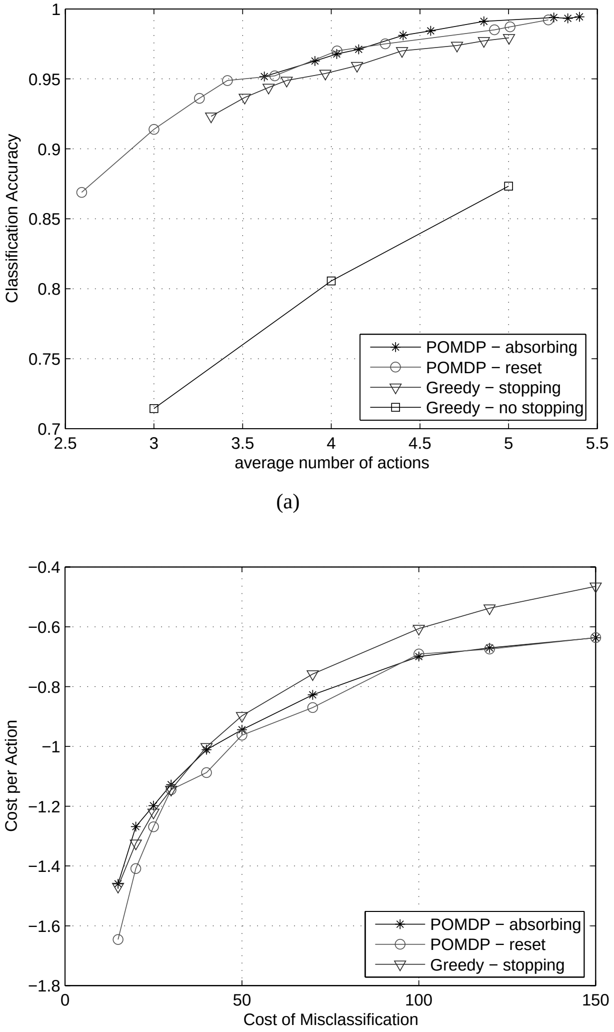
$$\begin{aligned} a_k(\pi_k) = &\arg\max[\max_{a\in\mathcal{A}_m} < R(s,a), \pi_k >, -1+ \\ &\beta \max_{a\in\mathcal{A}_c} \sum_{y\in\mathcal{Y}} \max_{a\in\mathcal{A}_m} < R(s,a), \hat{\mathcal{T}}(\pi_k, y, a)) > \mathrm{P}(y|\pi, a)] \end{aligned}$$

That is, decide the current choice of action by evaluating the benefit of making a classification decision at k, or taking one additional measurement and making a classification decision afterward. At every new stage k, this problem is solved to determine the current action a_k; this approach is known in control theory as *receding horizon control* or *model-predictive control*, and yields a low-complexity approximation to the longer horizon control problems.

Another approach to generating a sensor management policy is to use a finite horizon POMDP formulation that takes a fixed number of observations T before making a classification decision. The resulting policy does not stop adaptively, but instead performs a classification decision at stage T.

Figure 2.4(a) shows the classification accuracy achieved versus expected number of measurements taken for three algorithms: The POMDP algorithm with adaptive stopping, the one-step lookahead adaptive stopping algorithm, and the non-adaptive stopping algorithm with fixed number of views. Each point in the graph for the adaptive algorithms was generated by varying the cost of an erroneous classification C_c from 15 to 150. For the non-adaptive stopping algorithm, the number of measurement actions was varied. The results represent the average performance of the strategies using Monte Carlo simulations, averaging over possible initial conditions of the target type and orientation and measurement values.

Figure 2.4(a) highlights the advantages of adaptive stopping policies over the fixed stopping policy. Adaptive stopping exploits the availability of good measurements in determining whether additional measurements would be valuable. The results also highlight a small increase in performance when using the



(a)

(b)

Figure 2.4. a) Classification performance of the different sensor scheduling algorithms as a function of the average number of actions. b) Classification performance versus classification error cost for adaptive stopping policies.

full POMDP adaptive policy versus the one-step lookahead (single stage) policy, as expected.

Figure 2.4(b) presents classification performance for the two adaptive stopping algorithms versus the misclassification cost C_c. Note that the one-step lookahead algorithm underperforms the full horizon algorithm, as it underestimates the amount of information that can be acquired in the future (because it is one-step lookahead). As a consequence, the one-step lookahead algorithm tends to make classification decisions earlier, resulting in a loss of classification performance.

6. Conclusion

This chapter presented an overview of the models and algorithms used for sequential decision making under uncertainty, with a focus on sensor management models. Accumulated information is modeled as a Markov state that evolves in response to selection of sensor actions. Using a reward structure that is additive over time, we discussed the application of stochastic dynamic programming to characterize both the optimal rewards and optimal strategies in these classes of problems. We also presented alternative formulations of rewards, from finite horizon rewards to infinite horizon discounted and undiscounted rewards.

The models presented in this chapter form the foundation for sensor management applications that plan over temporal sequences of sensor actions and adapt to the information observed. Such models are exploited in Chapters 5, 6, 7, 9 and 10 for addressing specific sensor management applications. The next chapter develops information theoretic reward functions that are combined with the joint particle filtering methods of Chapter 4 to implement approximate POMDP sensor management algorithms discussed in Chapter 5.