3

Dynamic Programming

Once the performance measure for a system has been chosen, the next task is to determine a control function that minimizes this criterion. Two methods of accomplishing the minimization are the minimum principle of Pontryagin [P-1], and the method of dynamic programming developed by R. E. Bellman [B-1, B-2, B-3]. The variational approach of Pontryagin (Chapter 5) leads to a nonlinear two-point boundary-value problem that must be solved (Chapter 6) to obtain an optimal control. In this chapter we shall consider the method of dynamic programming and see that it leads to a functional equation that is amenable to solution by use of a digital computer.

3.1 THE OPTIMAL CONTROL LAW

In Chapter 1 we defined an optimal control of the form

$$\mathbf{u}^*(t) = \mathbf{f}(\mathbf{x}(t), t) \tag{3.1-1}$$

as being a *closed-loop* or *feedback* optimal control. The functional relationship $\mathbf{f}$ is called the *optimal control law*, or the *optimal policy*. Notice that the optimal control law specifies how to generate the control value at time t from the state value at time t. The presence of t as an argument of $\mathbf{f}$ indicates that the optimal control law may be time-varying.

In the method of dynamic programming, an optimal policy is found by employing the intuitively appealing concept called the principle of optimality.

3.2 THE PRINCIPLE OF OPTIMALITY†

The *optimal* path for a multistage decision process is shown in Fig. 3-1(a). Suppose that the first decision (made at a) results in segment a-b with cost

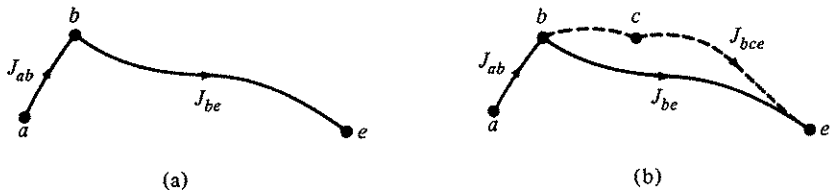

Figure 3-1 (a) Optimal path from a to e. (b) Two possible optimal paths from b to e

J_{ab} and that the remaining decisions yield segment b-e at a cost of J_{be}. The *minimum* cost J_{ae}^* from a to e is therefore

$$J_{ae}^* = J_{ab} + J_{be}. \tag{3.2-1}$$

ASSERTION: If a-b-e is the optimal path from a to e, then b-e is the optimal path from b to e.

Proof by contradiction: Suppose b-c-e in Fig. 3-1(b) is the optimal path from b to e; then

$$J_{bce} < J_{be}, \tag{3.2-2}$$

and

$$J_{ab} + J_{bce} < J_{ab} + J_{be} = J_{ae}^* \tag{3.2-3}$$

but (3.2-3) can be satisfied only by violating the condition that a-b-e is the optimal path from a to e. Thus the assertion is proved.

Bellman [B-1] has called the above property of an optimal policy the principle of optimality:

> An optimal policy has the property that whatever the initial state and initial decision are, the remaining decisions must constitute an optimal policy with regard to the state resulting from the first decision.

† Sections 3.2 through 3.6 follow the presentation given in [K-4].

3.3 APPLICATION OF THE PRINCIPLE OF OPTIMALITY TO DECISION-MAKING

The following example illustrates the procedure for making a single optimal decision with the aid of the principle of optimality.

Consider a process whose current state is b. The paths resulting from all allowable decisions at b are shown in Fig. 3-2(a). The optimal paths from c, d, and e to the terminal point f are shown in Fig. 3-2(b). The principle of

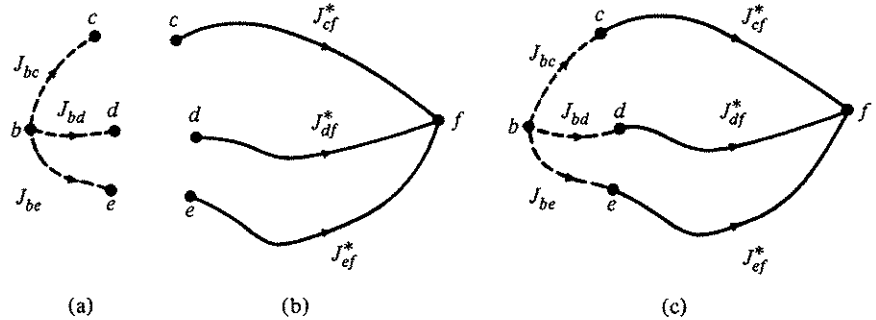

Figure 3-2 (a) Paths resulting from all allowable decisions at b. (b) Optimal paths from c, d, e to f. (c) Candidates for optimal paths from b to f

optimality implies that if b-c is the initial segment of the optimal path from b to f, then c-f is the terminal segment of this optimal path. The same reasoning applied to initial segments b-d and b-e indicates that the paths in Fig. 3-2(c) are the only candidates for the optimal trajectory from b to f. The optimal trajectory that starts at b is found by comparing

$$\begin{aligned} C_{bcf}^* &= J_{bc} + J_{cf}^* \\ C_{bdf}^* &= J_{bd} + J_{df}^* \\ C_{bef}^* &= J_{be} + J_{ef}^*. \end{aligned} \tag{3.3-1}$$

The minimum of these costs must be the one associated with the optimal decision at point b.

Dynamic programming is a computational technique which extends the above decision-making concept to *sequences* of decisions which together define an optimal policy and trajectory. The optimal routing problem in the next section illustrates the procedure.

3.4 DYNAMIC PROGRAMMING APPLIED TO A ROUTING PROBLEM

A motorist wishes to know how to minimize the cost of reaching some destination h from his current location. He can only travel (one-way as indicated) on the streets shown on his map (Fig. 3-3), and at the intersection-to-intersection costs given.

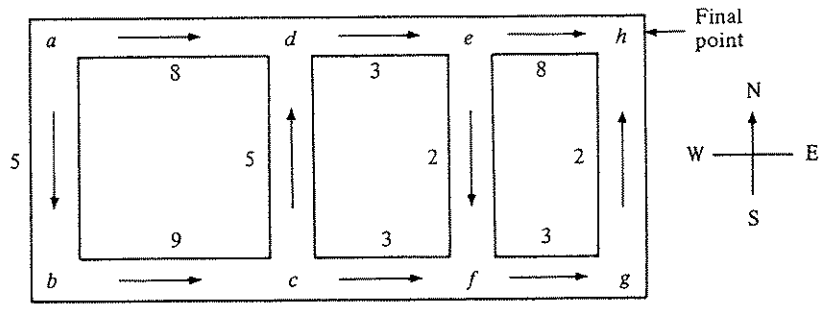

Figure 3-3 The road map

Instead of trying all allowable paths leading from each intersection to h and selecting the one with lowest cost (an exhaustive search), consider the application of the principle of optimality. In this problem, "state" refers to the intersection and a "decision" is the choice of heading (control) elected by the driver when he leaves an intersection.

Suppose the motorist is at c; from there he can go only to d or f, and then on to h. Let J_{cd} denote the cost of moving from c to d and J_{cf} the cost from c to f. Assume that the motorist already knows the minimum costs, J^*_{dh} and J^*_{fh}, to reach the final destination h from d and f. (In this example, $J^*_{dh} = 10$ and $J^*_{fh} = 5$.) Then the minimum cost J^*_{ch} to reach h from c is the smaller of

$$C^*_{cdh} = J_{cd} + J^*_{dh} = \text{minimum cost to reach } h \text{ from } c \text{ via } d \tag{3.4-1}$$

and

$$C^*_{cfh} = J_{cf} + J^*_{fh} = \text{minimum cost to reach } h \text{ from } c \text{ via } f. \tag{3.4-2}$$

Thus,

$$\begin{aligned} J^*_{ch} &= \min\{C^*_{cdh}, C^*_{cfh}\} \\ &= \min\{15, 8\} \\ &= 8 \end{aligned} \tag{3.4-3}$$

and the optimal decision at c is to go to f.

How does the motorist know the values for J^*_{dh} and J^*_{fh}? These quantities must have been calculated previously by working backward from h. For example, $J^*_{gh} = 2$—there is only one path from g to h. J^*_{gh} is then used to find J^*_{fh} from

$$\begin{aligned} J^*_{fh} &= J_{fg} + J^*_{gh} \\ &= 3 + 2 \\ &= 5. \end{aligned} \tag{3.4-4}$$

Then

$$J^*_{ch} = \min\{J_{eh}, [J_{ef} + J^*_{fh}]\} \tag{3.4-5}$$

and so on. The general approach should now be evident. It remains to formalize the computational algorithm. In this connection it will be convenient to introduce the following notation:

- α is the current state (intersection).
- u_i is an allowable decision (control) elected at the state α. In this example i can assume one or more of the values 1, 2, 3, 4, corresponding to the headings N, E, S, W.
- x_i is the state (intersection) adjacent to α which is reached by application of u_i at α.
- h is the final state.
- $J_{\alpha x_i}$ is the cost to move from α to x_i.
- $J^*_{x_i h}$ is the *minimum cost* to reach the final state h from x_i.
- $C^*_{\alpha x_i h}$ is the minimum cost to go from α to h *via* x_i.
- $J^*_{\alpha h}$ is the minimum cost to go from α to h (by any allowable path).
- $u^*(\alpha)$ is the optimal decision (control) at α.

When this notation is used, the principle of optimality implies that

$$C^*_{\alpha x_i h} = J_{\alpha x_i} + J^*_{x_i h} \tag{3.4-6}$$

and, as before, the optimal decision at α, $u^*(\alpha)$, is the decision that leads to

$$J^*_{\alpha h} = \min\{C^*_{\alpha x_1 h}, C^*_{\alpha x_2 h}, \ldots, C^*_{\alpha x_i h}, \ldots\}. \tag{3.4-7}$$

These two equations define the algorithm called dynamic programming. To illustrate the procedure, the automobile routing problem has been "solved" in Table 3-1, where only the consequences of lawful decisions are included. Notice particularly that the intersections nearest the destination h are considered first, and that the optimal trajectories (routes) are built up from h *backwards* toward the earlier states (intersections). This is necessary in order that $J^*_{x_i h}$ be known *prior* to the calculation of $C^*_{\alpha x_i h}$ $(= J_{\alpha x_i} + J^*_{x_i h})$.

Once the table has been completed, the optimal path from *any* intersection to h can be obtained by entering the table at the appropriate intersection and reading off the optimal heading at each successive intersection along the trajectory. For example, if the motorist starts at b, the table tells him to head east. Heading east, he arrives at c, where the table indicates he should again move east. Continuing the process, we find the optimal path from b to h to be *b-c-f-g-h* and the minimum cost to be 17.

The information in the table also allows the motorist to adjust his route if he is forced to deviate from the optimal path. Suppose he started at b and reached c only to find the road to f closed for repairs; he is forced to move to d. After doing so, he enters the table and finds that from d the optimal path to h is *d-e-f-g-h*.

Notice that a motorist at intersection a who heads south instead of east is being misled by the prospect of a short-term gain. His overall cost will be higher, even if he thereafter follows the optimal route.

Table 3-1 CALCULATION OF OPTIMAL HEADINGS BY DYNAMIC PROGRAMMING

Current intersection	*Heading*	*Next intersection*	*Minimum cost from* α *to* h *via* x_t	*Minimum cost to reach* h *from* α	*Optimal heading at* α
α	u_t	x_t	$J_{\alpha x_t} + J^*_{x_t h} = C^*_{\alpha x_t h}$	$J^*_{\alpha h}$	$u^*(\alpha)$
g	N	h	$2 + 0 = 2$	2	N
f	E	g	$3 + 2 = 5$	5	E
e	E	h	$8 + 0 = 8$		
	S	f	$2 + 5 = 7$	7	S
d	E	e	$3 + 7 = 10$	10	E
c	N	d	$5 + 10 = 15$		
	E	f	$3 + 5 = 8$	8	E
b	E	c	$9 + 8 = 17$	17	E
a	E	d	$8 + 10 = 18$	18	E
	S	b	$5 + 17 = 22$		

3.5 AN OPTIMAL CONTROL SYSTEM

Consider a system described by the first-order differential equation

$$\frac{d}{dt}[x(t)] = ax(t) + bu(t), \tag{3.5-1}$$

where $x(t)$ and $u(t)$ are the state and control variables, respectively, and a and b are constants. The admissible values of the state and control variables are constrained by

$$0.0 \leq x(t) \leq 1.5$$

and

$$-1.0 \leq u(t) \leq 1.0, \tag{3.5-2}$$

and the performance measure (cost) to be minimized is

$$J = x^2(T) + \lambda \int_0^T u^2(t)\, dt, \tag{3.5-3}$$

where T is the specified final time, and λ is a weighting factor included to permit adjustment of the relative importance of the two terms in J. $x(T)$ and $u(t)$ are squared because positive and negative values of these quantities are of equal importance. This performance measure reflects the desire to drive the final state $x(T)$ close to zero without excessive expenditure of control effort.

Before the numerical procedure of dynamic programming can be applied, the system differential equation must be approximated by a difference equation, and the integral in the performance measure must be approximated by a summation. This can be done most conveniently by dividing the time interval $0 \leq t \leq T$ into N equal increments, Δt. Then from (3.5-1)

$$\frac{x(t + \Delta t) - x(t)}{\Delta t} \approx ax(t) + bu(t), \tag{3.5-4}$$

or

$$x(t + \Delta t) = [1 + a\,\Delta t]x(t) + b\,\Delta t\, u(t). \tag{3.5-5}$$

It will be assumed that Δt is small enough so that the control signal can be approximated by a piecewise-constant function that changes only at the instants $t = 0, \Delta t, \ldots, (N-1)\,\Delta t$; thus, for $t = k\,\Delta t$,

$$x([k+1]\,\Delta t) = [1 + a\,\Delta t]x(k\,\Delta t) + b\,\Delta t\, u(k\,\Delta t); \qquad k = 0, 1, \ldots, N-1. \tag{3.5-6}$$

$x(k\,\Delta t)$ is referred to as the kth value of x and is denoted by $x(k)$. The system difference equation can then be written

$$x(k+1) = [1 + a\,\Delta t]x(k) + b\,\Delta t\, u(k). \tag{3.5-7}$$

In a similar way the performance measure becomes

$$J = x^2(N\,\Delta t) + \lambda\left[\int_0^{\Delta t} u^2(0)\,dt + \int_{\Delta t}^{2\,\Delta t} u^2(\Delta t)\,dt + \cdots + \int_{(N-1)\,\Delta t}^{N\,\Delta t} u^2([N-1]\,\Delta t)\,dt\right], \tag{3.5-8}$$

or,

$$\begin{aligned} J &= x^2(N) + \lambda\,\Delta t[u^2(0) + u^2(1) + \cdots + u^2(N-1)] \\ &= x^2(N) + \lambda\,\Delta t \sum_{k=0}^{N-1} u^2(k). \end{aligned} \tag{3.5-9}$$

Now the method of dynamic programming can be applied as in the automobile routing problem. For numerical simplicity let $a = 0$, $b = 1$, $\lambda = 2$, $T = 2$, $\Delta t = 1$, in which case $N = 2$; i.e., this is a two-stage process described by the difference equation

$$x(k+1) = x(k) + u(k); \qquad k = 0, 1 \tag{3.5-10}$$

where $u(0)$ and $u(1)$ are to be selected to minimize the performance measure (cost)

$$J = x^2(2) + 2u^2(0) + 2u^2(1) \tag{3.5-11}$$

subject to the constraints

$$\begin{aligned} &0.0 \leq x(k) \leq 1.5; \qquad k = 0, 1, 2 \\ \text{and} \qquad & \\ &-1.0 \leq u(k) \leq 1.0; \qquad k = 0, 1. \end{aligned} \tag{3.5-12}$$

The first step in the computational procedure is to find the optimal policy for the last stage of operation. This is essentially a matter of trying *all* of the allowable control values at *each* of the allowable state values. The optimal control for each state value is the one which yields the trajectory having the minimum cost. To limit the required number of calculations, and thereby make the computational procedure feasible, the allowable state and control values must be quantized. In this problem it will be assumed that the quantized values are $x(k) = 0.0, 0.5, 1.0, 1.5$, and $u(k) = -1.0, -0.5, 0.0, 0.5, 1.0$.

Using these values, we find that the computational procedure for determining the optimal policy over the last stage is

1. Put $k = 1$, select one of the quantized values of $x(1)$, try all quantized values of $u(1)$, and calculate the resulting trajectories. The optimal control for this state value is the one which yields the minimum cost.
2. Repeat the procedure in step 1 for the remaining quantized levels of $x(1)$.

The resulting calculations are shown in Table 3-2, where calculations leading to a violation of the constraints have been omitted. Notice that the cost J_{12} of going from state $x(1)$ to state $x(2)$ is dependent on the *value* of the state $x(1)$ and on the *value* of the control applied, $u(1)$; hence the notation $J_{12}(x(1), u(1))$. Similarly, the minimum cost $J_{12}^*(x(1))$ and the optimal control $u^*(x(1), 1)$ applied at $k = 1$ are dependent on the value of the state $x(1)$.†

Now consider the next-to-last stage of the process by putting $k = 0$. At each quantized value of the state $x(0)$ all quantized values of the control $u(0)$ are tried. The trajectory from $x(0)$ to $x(1)$ is computed for each trial, together with the cost J_{01}. Then, knowing the value of $x(1)$ at the end of each such trajectory, we may follow the optimal trajectory over the last stage with the aid of the data available in Table 3-2. In mathematical terms this means that

$$C_{02}^*(x(0), u(0)) = J_{01}(x(0), u(0)) + J_{12}^*(x(1)), \tag{3.5-13}$$

and thus the cost of the optimal trajectory is

$$J_{02}^*(x(0)) = \min_{u(0)} [J_{01}(x(0), u(0)) + J_{12}^*(x(1))], \tag{3.5-14}$$

where

$C_{02}^*(x(0), u(0))$ is the minimum cost of operation over the last two stages for one quantized value of $x(0)$ *given a particular trial quantized value of* $u(0)$.

$J_{01}(x(0), u(0))$ is the cost of operation in the interval $k = 0$ to $k = 1$ for specified quantized values of $x(0)$ and $u(0)$.

$J_{12}^*(x(1))$ is the cost of the optimal last-stage trajectory which is a function of the state $x(1)$.

$J_{02}^*(x(0))$ is the minimum cost of operation over the last two stages for a specified quantized value of $x(0)$.

Notice that (3.5-13) and (3.5-14) are analogous to (3.4-6) and (3.4-7) in the automobile routing problem.

Finally, (3.5-13) and (3.5-14) are mechanized in Table 3-3 to complete the dynamic programming algorithm.

The information contained in Tables 3-2 and 3-3 may now be used to determine the optimal trajectory from any allowable quantized value of $x(0)$

† Rather than adhere to the form of (3.1-1) for the optimal control law, $\mathbf{u}^*(k) = \mathbf{f}(\mathbf{x}(k), k)$, we will shorten the notation by writing simply $\mathbf{u}^*(\mathbf{x}(k), k)$.

Table 3-2 COSTS OF OPERATION OVER THE LAST STAGE

Current state $x(1)$	*Control* $u(1)$	*Next state* $x(2) = x(1) + u(1)$	*Cost* $x^2(2) + 2u^2(1) = J_{12}(x(1), u(1))$	*Minimum cost* $J^*_{12}(x(1))$	*Optimal control applied at* $k = 1$ $u^*(x(1), 1)$
1.5	0.0	1.5	$(1.5)^2 + 2(0.0)^2 = 2.25$		
	−0.5	1.0	$(1.0)^2 + 2(-0.5)^2 = 1.50$	$J^*_{12}(1.5) = 1.50$	$u^*(1.5, 1) = -0.5$
	−1.0	0.5	$(0.5)^2 + 2(-1.0)^2 = 2.25$		
1.0	0.5	1.5	$(1.5)^2 + 2(0.5)^2 = 2.75$		
	0.0	1.0	$(1.0)^2 + 2(0.0)^2 = 1.00$		
	−0.5	0.5	$(0.5)^2 + 2(-0.5)^2 = 0.75$	$J^*_{12}(1.0) = 0.75$	$u^*(1.0, 1) = -0.5$
	−1.0	0.0	$(0.0)^2 + 2(-1.0)^2 = 2.00$		
0.5	1.0	1.5	$(1.5)^2 + 2(1.0)^2 = 4.25$		
	0.5	1.0	$(1.0)^2 + 2(0.5)^2 = 1.50$		
	0.0	0.5	$(0.5)^2 + 2(0.0)^2 = 0.25$	$J^*_{12}(0.5) = 0.25$	$u^*(0.5, 1) = 0.0$
	−0.5	0.0	$(0.0)^2 + 2(-0.5)^2 = 0.50$		
0.0	1.0	1.0	$(1.0)^2 + 2(1.0)^2 = 3.00$		
	0.5	0.5	$(0.5)^2 + 2(0.5)^2 = 0.75$		
	0.0	0.0	$(0.0)^2 + 2(0.0)^2 = 0.00$	$J^*_{12}(0.0) = 0.00$	$u^*(0.0, 1) = 0.0$

Table 3-3 COSTS OF OPERATION OVER THE LAST TWO STAGES

Current state $x(0)$	*Control* $u(0)$	*Next state* $x(1) = x(0) + u(0)$	*Minimum cost over last two stages for trial value* $u(0)$ $J_{01}(x(0), u(0)) + J^*_{12}(x(1)) = 2u^2(0) + J^*_{12}(x(1)) = C^*_{02}(x(0), u(0))$	*Minimum cost over last two stages* $J^*_{02}(x(0))$	*Optimal control applied at* $k = 0$ $u^*(x(0), 0)$
1.5	0.0	1.5	$2(0.0)^2 + 1.50 = 1.50$		
	−0.5	1.0	$2(-0.5)^2 + 0.75 = 1.25$	$J^*_{02}(1.5) = 1.25$	$u^*(1.5, 0) = -0.5$
	−1.0	0.5	$2(-1.0)^2 + 0.25 = 2.25$		
1.0	0.5	1.5	$2(0.5)^2 + 1.50 = 2.00$		
	0.0	1.0	$2(0.0)^2 + 0.75 = 0.75$		
	−0.5	0.5	$2(-0.5)^2 + 0.25 = 0.75$	$J^*_{02}(1.0) = \begin{Bmatrix} 0.75 \\ 0.75 \end{Bmatrix}$	$u^*(1.0, 0) = \begin{Bmatrix} 0.0 \\ -0.5 \end{Bmatrix}$
	−1.0	0.0	$2(-1.0)^2 + 0.00 = 2.00$		
0.5	1.0	1.5	$2(1.0)^2 + 1.50 = 3.50$		
	0.5	1.0	$2(0.5)^2 + 0.75 = 1.25$		
	0.0	0.5	$2(0.0)^2 + 0.25 = 0.25$	$J^*_{02}(0.5) = 0.25$	$u^*(0.5, 0) = 0.0$
	−0.5	0.0	$2(-0.5)^2 + 0.00 = 0.50$		
0.0	1.0	1.0	$2(1.0)^2 + 0.75 = 2.75$		
	0.5	0.5	$2(0.5)^2 + 0.25 = 0.75$		
	0.0	0.0	$2(0.0)^2 + 0.00 = 0.00$	$J^*_{02}(0.0) = 0.00$	$u^*(0.0, 0) = 0.0$

to the final state $x(2)$. For example, if $x(0) = 1.5$, Table 3-3 indicates that $u^*(1.5, 0) = -0.5$ and $J_{02}^*(1.5) = 1.25$. Application of $u^*(1.5, 0)$ at $x(0) = 1.5$ makes $x(1) = 1.0$, and Table 3-2 gives the optimal control applied at $k = 1$ as $u^*(1.0, 1) = -0.5$. Thus, for $x(0) = 1.5$ the optimal control sequence is $\{-0.5, -0.5\}$, and the minimum cost is 1.25.

In a similar way, the optimal policies and trajectories can be determined from the tables for the other values of $x(0)$. Observe that if $x(0) = 1.0$ the optimal policy is nonunique—the control sequences $\{0, -0.5\}$ and $\{-0.5, 0\}$ are both optimal. Notice also that in this problem there is no requirement that all the trajectories end at the same value of $x(2)$. A problem in which $x(T)$ is specified is included in the problems at the end of the chapter (Problem 3-3).

If a problem is segmented into more than two stages, the procedure must simply be extended by repeating the calculations of Table 3-3 for each preceding stage. In general, to determine the optimal control applied at $t = k\,\Delta t$ in an N-stage process the appropriate forms for (3.5-13) and (3.5-14) are

$$C_{kN}^*(x(k), u(k)) = J_{k,k+1}(x(k), u(k)) + J_{k+1,N}^*(x(k+1)), \tag{3.5-13a}$$

$$J_{kN}^*(x(k)) = \min_{u(k)} [C_{kN}^*(x(k), u(k))]. \tag{3.5-14a}$$

Taken together, equations (3.5-13a) and (3.5-14a) form the *functional equation of dynamic programming;* we shall have more to say about this in Section 3.7.

In more practical problems a digital computer would normally be needed, and it often becomes important to minimize the amount of storage required for the retention of intermediate results. The calculations in Table 3-3 and the determination of the optimal policy and trajectory for any allowable value of $x(0)$ require only the data in the last two columns of Tables 3-2 and 3-3; therefore, only these data need be stored.

3.6 INTERPOLATION

In the preceding control example all of the trial control values drive the state of the system either to a computational "grid" point or to a value outside of the allowable range. Had the numerical values not been carefully selected, this happy situation would not have been obtained and interpolation would have been required. For example, suppose that the trial values for $u(k)$ had been -1, -0.75, -0.5, -0.25, 0, 0.25, 0.5, 0.75, 1. The values of $J_{12}^*(x(1))$ and $u^*(x(1), 1)$ shown next to the state points in Fig. 3-4(a) are

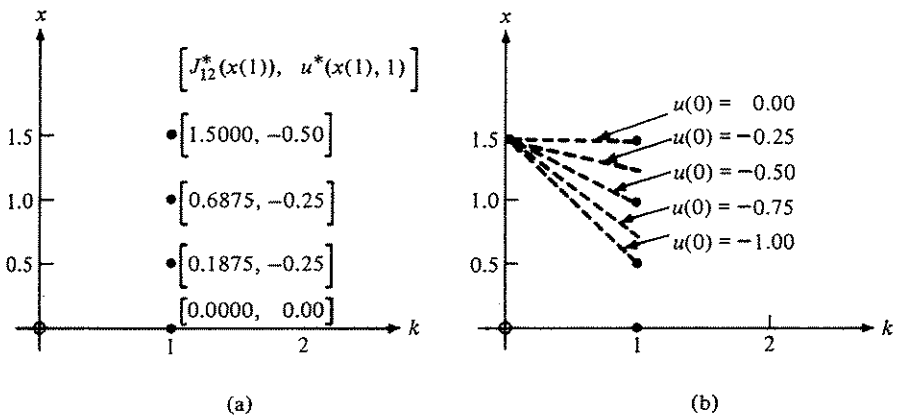

Figure 3-4 (a) Minimum costs and optimal controls for quantized values of $x(1)$. (b) Paths resulting from the application of quantized control values at $x(0) = 1.5$

the results of repeating the calculations in Table 3-2 with the new trial values for $u(1)$.

Next, suppose that all of the quantized values of the control are applied for a state value of $x(0) = 1.5$. The resulting values of $x(1)$ are shown in Fig. 3-4(b), where it can be seen that two of the end points do not coincide with the grid points of Fig. 3-4(a). But, by linear interpolation,

$$\begin{aligned} J_{12}^*(1.25) &= 0.68750 + \tfrac{1}{2}[1.50000 - 0.68750] \\ &= 1.09375 \end{aligned} \tag{3.6-1}$$

and

$$\begin{aligned} J_{12}^*(0.75) &= 0.18750 + \tfrac{1}{2}[0.68750 - 0.18750] \\ &= 0.43750 \end{aligned} \tag{3.6-2}$$

Finally, the result of repeating the calculations in Table 3-3 [for $x(0) = 1.5$ only], the interpolated values of $J_{12}^*(x(1))$ being used where required, is shown in Table 3-4.

Interpolation may also be required when one is using stored data to calculate an optimal control sequence. For example, if the optimal control applied at some value of $x(0)$ drives the system to a state value $x(1)$ that is halfway between two points where the optimal controls are -1 and -0.5, then by linear interpolation the optimal control is -0.75.

In summary, although a finite grid of state and control values must be employed in the numerical procedure, interpolation makes available approximate information about intermediate points. Naturally, the degree of approxi-

Table 3-4 COSTS OF OPERATION OVER THE LAST TWO STAGES FOR $x(0) = 1.50$

Current state	*Control*	*Next state*	*Minimum cost over last two stages for trial value u(0)* $J_{01}(x(0), u(0)) + J_{12}^*(x(1)) =$		*Minimum cost over last two stages*	*Optimal control applied at* $k = 0$
$x(0)$	$u(0)$	$x(1) = x(0) + u(0)$	$2u^2(0) + J_{12}^*(x(1)) = C_{02}^*(x(0), u(0))$		$J_{02}^*(x(0))$	$u^*(x(0), 0)$
1.50	0.00	1.50	$2(0.00)^2 + 1.50000$	$= 1.50000$		
	−0.25	1.25	$2(-0.25)^2 + 1.09375$	$= 1.21875$		
	−0.50	1.00	$2(-0.50)^2 + 0.68750$	$= 1.18750$	$J_{02}^*(1.5) = 1.18750$	$u^*(1.5, 0) = -0.50$
	−0.75	0.75	$2(-0.75)^2 + 0.43750$	$= 1.56250$		
	−1.00	0.50	$2(-1.00)^2 + 0.18750$	$= 2.18750$		

mation depends on the separation of the grid points, the interpolation scheme used, and the system dynamics and performance measure. A finer grid generally means greater accuracy, but also increased storage requirements and computation time. The effects of these factors are illustrated in some of the exercises at the end of the chapter (Problems 3-14 through 3-18).

3.7 A RECURRENCE RELATION OF DYNAMIC PROGRAMMING

In this section we shall begin to formalize some of the ideas introduced intuitively in preceding sections. In particular, we wish to generalize the procedure in Section 3.5 which led to equations (3.5-13a) and (3.5-14a). Since our attention is focused on control systems, a recurrence relation will be derived by applying dynamic programming to a control process.

An nth-order time-invariant system† is described by the state equation

$$\dot{\mathbf{x}}(t) = \mathbf{a}(\mathbf{x}(t), \mathbf{u}(t)). \tag{3.7-1}$$

It is desired to determine the control law which minimizes the performance measure

$$J = h(\mathbf{x}(t_f)) + \int_0^{t_f} g(\mathbf{x}(t), \mathbf{u}(t))\, dt, \tag{3.7-2}$$

where t_f is assumed fixed. The admissible controls are constrained to lie in a set U; i.e., $\mathbf{u} \in U$. As before, we first approximate the continuously operating system of Eq. (3.7-1) by a discrete system; this is accomplished by considering N equally spaced time increments in the interval $0 \leq t \leq t_f$. From (3.7-1)

$$\frac{\mathbf{x}(t + \Delta t) - \mathbf{x}(t)}{\Delta t} \approx \mathbf{a}(\mathbf{x}(t), \mathbf{u}(t)) \tag{3.7-3}$$

or

$$\mathbf{x}(t + \Delta t) = \mathbf{x}(t) + \Delta t\, \mathbf{a}(\mathbf{x}(t), \mathbf{u}(t)). \tag{3.7-4}$$

Using the shorthand notation developed earlier for $\mathbf{x}(k\,\Delta t)$ gives

$$\mathbf{x}(k + 1) = \mathbf{x}(k) + \Delta t\, \mathbf{a}(\mathbf{x}(k), \mathbf{u}(k)), \tag{3.7-5}$$

which we will denote by

$$\mathbf{x}(k + 1) \triangleq \mathbf{a}_D(\mathbf{x}(k), \mathbf{u}(k)). \tag{3.7-6}$$

† The following derivation can be applied to time-varying systems as well; time-invariance is assumed only to simplify the notation.

Operating on the performance measure in a similar manner, we obtain

$$J = h(\mathbf{x}(N\,\Delta t)) + \int_0^{\Delta t} g\,dt + \int_{\Delta t}^{2\,\Delta t} g\,dt + \cdots + \int_{(N-1)\,\Delta t}^{N\,\Delta t} g\,dt, \quad (3.7\text{-}7)$$

which becomes for small Δt,

$$J \approx h(\mathbf{x}(N)) + \Delta t \sum_{k=0}^{N-1} g(\mathbf{x}(k), \mathbf{u}(k)), \quad (3.7\text{-}8)$$

which we shall denote by

$$J = h(\mathbf{x}(N)) + \sum_{k=0}^{N-1} g_D(\mathbf{x}(k), \mathbf{u}(k)). \quad (3.7\text{-}8a)$$

By making the problem discrete as we have done, it is now required that the optimal control law $\mathbf{u}^*(\mathbf{x}(0), 0), \mathbf{u}^*(\mathbf{x}(1), 1), \ldots, \mathbf{u}^*(\mathbf{x}(N-1), N-1)$ be determined for the system given by Eq. (3.7-6) which has the performance measure given by (3.7-8a). We are now ready to derive the recurrence equation.

Begin by defining

$$J_{NN}(\mathbf{x}(N)) \triangleq h(\mathbf{x}(N)); \quad (3.7\text{-}9)$$

J_{NN} is the cost of reaching the final state value $\mathbf{x}(N)$. Next, define

$$\begin{aligned} J_{N-1,\,N}(\mathbf{x}(N-1), \mathbf{u}(N-1)) &\triangleq g_D(\mathbf{x}(N-1), \mathbf{u}(N-1)) + h(\mathbf{x}(N)) \\ &= g_D(\mathbf{x}(N-1), \mathbf{u}(N-1)) + J_{NN}(\mathbf{x}(N)), \end{aligned} \quad (3.7\text{-}10)$$

which is the cost of operation during the interval $(N-1)\,\Delta t \leq t \leq N\,\Delta t$. Observe that $J_{N-1,\,N}$ is also the cost of a *one-stage process* with initial state $\mathbf{x}(N-1)$. The value of $J_{N-1,\,N}$ is dependent only on $\mathbf{x}(N-1)$ and $\mathbf{u}(N-1)$, since $\mathbf{x}(N)$ is related to $\mathbf{x}(N-1)$ and $\mathbf{u}(N-1)$ through the state equation (3.7-6), so we write

$$\begin{aligned} J_{N-1,\,N}(\mathbf{x}(N-1), \mathbf{u}(N-1)) &= g_D(\mathbf{x}(N-1), \mathbf{u}(N-1)) \\ &\quad + J_{NN}(\mathbf{a}_D(\mathbf{x}(N-1), \mathbf{u}(N-1))). \end{aligned} \quad (3.7\text{-}11)$$

The optimal cost is then

$$J^*_{N-1,\,N}(\mathbf{x}(N-1)) \triangleq \min_{\mathbf{u}(N-1)} \{g_D(\mathbf{x}(N-1), \mathbf{u}(N-1)) + J_{NN}(\mathbf{a}_D(\mathbf{x}(N-1), \mathbf{u}(N-1)))\}\dagger \quad (3.7\text{-}12)$$

† Notice that the minimization is performed with only admissible control values being used.

We know that the optimal choice of $\mathbf{u}(N-1)$ will depend on $\mathbf{x}(N-1)$, so we denote the minimizing control by $\mathbf{u}^*(\mathbf{x}(N-1), N-1)$.

The cost of operation over the last two intervals is given by

$$\begin{aligned} &J_{N-2,\,N}(\mathbf{x}(N-2), \mathbf{u}(N-2), \mathbf{u}(N-1)) \\ &\quad = g_D(\mathbf{x}(N-2), \mathbf{u}(N-2)) + g_D(\mathbf{x}(N-1), \mathbf{u}(N-1)) + h(\mathbf{x}(N)) \\ &\quad = g_D(\mathbf{x}(N-2), \mathbf{u}(N-2)) + J_{N-1,\,N}(\mathbf{x}(N-1), \mathbf{u}(N-1)), \end{aligned} \quad (3.7\text{-}13)$$

where again we have used the dependence of $\mathbf{x}(N)$ on $\mathbf{x}(N-1)$ and $\mathbf{u}(N-1)$. As before, observe that $J_{N-2,\,N}$ is the cost of a *two-stage process* with initial state $\mathbf{x}(N-2)$. The optimal policy during the last two intervals is found from

$$\begin{aligned} &J^*_{N-2,\,N}(\mathbf{x}(N-2)) \\ &\quad \triangleq \min_{\mathbf{u}(N-2),\,\mathbf{u}(N-1)} \{g_D(\mathbf{x}(N-2), \mathbf{u}(N-2)) + J_{N-1,\,N}(\mathbf{x}(N-1), \mathbf{u}(N-1))\} \end{aligned} \quad (3.7\text{-}14)$$

The principle of optimality states that for this two-stage process, whatever the initial state $\mathbf{x}(N-2)$ and initial decision $\mathbf{u}(N-2)$, the remaining decision $\mathbf{u}(N-1)$ must be optimal with respect to the value of $\mathbf{x}(N-1)$ that results from application of $\mathbf{u}(N-2)$; therefore,

$$J^*_{N-2,\,N}(\mathbf{x}(N-2)) = \min_{\mathbf{u}(N-2)} \{g_D(\mathbf{x}(N-2), \mathbf{u}(N-2)) + J^*_{N-1,\,N}(\mathbf{x}(N-1))\}. \quad (3.7\text{-}15)$$

Since $\mathbf{x}(N-1)$ is related to $\mathbf{x}(N-2)$ and $\mathbf{u}(N-2)$ by the state equation, $J^*_{N-2,\,N}$ depends only on $\mathbf{x}(N-2)$; thus

$$\begin{aligned} &J^*_{N-2,\,N}(\mathbf{x}(N-2)) \\ &\quad = \min_{\mathbf{u}(N-2)} \{g_D(\mathbf{x}(N-2), \mathbf{u}(N-2)) + J^*_{N-1,\,N}(\mathbf{a}_D(\mathbf{x}(N-2), \mathbf{u}(N-2)))\} \end{aligned} \quad (3.7\text{-}15a)$$

By considering the cost of operation over the final *three* stages—a *three-stage process* with initial state $\mathbf{x}(N-3)$—we can follow exactly the same reasoning which led to Eqs. (3.7-13) through (3.7-15a) to obtain

$$\begin{aligned} &J^*_{N-3,\,N}(\mathbf{x}(N-3)) \\ &\quad = \min_{\mathbf{u}(N-3)} \{g_D(\mathbf{x}(N-3), \mathbf{u}(N-3)) + J^*_{N-2,\,N}(\mathbf{a}_D(\mathbf{x}(N-3), \mathbf{u}(N-3)))\} \end{aligned} \quad (3.7\text{-}16)$$

Continuing backward in this manner, we obtain for a K-stage process the result

$$\begin{aligned} &J^*_{N-K,\,N}(\mathbf{x}(N-K)) \\ &\quad = \min_{\mathbf{u}(N-K),\,\mathbf{u}(N-K+1),\,\ldots,\,\mathbf{u}(N-1)} \left\{h(\mathbf{x}(N)) + \sum_{k=N-K}^{N-1} g_D(\mathbf{x}(k), \mathbf{u}(k))\right\}, \end{aligned} \quad (3.7\text{-}17)$$

which by applying the principle of optimality becomes

$$J^*_{N-K,\,N}(\mathbf{x}(N-K)) = \min_{\mathbf{u}(N-K)} \{g_D(\mathbf{x}(N-K), \mathbf{u}(N-K)) + J^*_{N-(K-1),\,N}(\mathbf{a}_D(\mathbf{x}(N-K), \mathbf{u}(N-K)))\}.\dagger \tag{3.7-18}$$

Equation (3.7-18) is the recurrence relation that we set out to obtain. By knowing $J^*_{N-(K-1),\,N}$, the optimal cost for a $(K-1)$-stage policy, we can generate $J^*_{N-K,\,N}$, the optimal cost for a K-stage policy. To begin the process we simply start with a zero-stage process and generate $J^*_{NN} \triangleq J_{NN}$ (the * is just a notational convenience here; no choice of a control is implied). Next, the optimal cost can be found for a one-stage process by using J^*_{NN} and (3.7-18), and so on. Notice that beginning with a zero-stage process corresponds to starting at the terminal state h in the routing problem of Section 3.4 and starting at the final time $t = 2\,\Delta t$ in the control example of Section 3.5.

This derivation of the recurrence equation has also revealed another important concept—the *imbedding principle*. $J^*_{N-K,\,N}(\mathbf{x}(N-K))$ is the minimum cost possible for the final K stages of an N-stage process with state value $\mathbf{x}(N-K)$ at the beginning of the $(N-K)$th stage; however, $J^*_{N-K,\,N}(\mathbf{x}(N-K))$ is also the minimum cost possible for a K-stage process with initial state numerically equal to the value $\mathbf{x}(N-K)$. This means that the optimal policy and minimum costs for a K-stage process are contained (or imbedded) in the results for an N-stage process, provided that $N \geq K$.

Our discussion has been concerned primarily with the solution of optimal control problems; however, dynamic programming can also be applied to other types of optimization problems. For a more general treatment of dynamic programming and its applications, see references [B-2] and [N-1].

3.8 COMPUTATIONAL PROCEDURE FOR SOLVING OPTIMAL CONTROL PROBLEMS

Let us now summarize the dynamic programming computational procedure for determining optimal policies.

† An alternative notation often used is

$$J^*_K(\mathbf{x}(N-K)) = \min_{\mathbf{u}(N-K)} \{g_D(\mathbf{x}(N-K), \mathbf{u}(N-K)) + J^*_{K-1}(\mathbf{a}_D(\mathbf{x}(N-K), \mathbf{u}(N-K)))\},$$

where the subscripts of J^* indicate the number of stages. We shall use the notation of Eq. (3.7-18) because it more clearly indicates the computational procedure to be followed.

A system is described by the state difference equation†

$$\mathbf{x}(k+1) = \mathbf{a}_D(\mathbf{x}(k), \mathbf{u}(k)); \qquad k = 0, 1, \ldots, N-1. \tag{3.8-1}$$

It is desired to determine the control law that minimizes the criterion

$$J = h(\mathbf{x}(N)) + \sum_{k=0}^{N-1} g_D(\mathbf{x}(k), \mathbf{u}(k)).\ddagger \tag{3.8-2}$$

As shown in Section 3.7, the application of dynamic programming to this problem leads to the recurrence equation

$$J^*_{N-K,\,N}(\mathbf{x}(N-K)) = \min_{\mathbf{u}(N-K)} \{g_D(\mathbf{x}(N-K), \mathbf{u}(N-K)) + J^*_{N-(K-1),\,N}(\mathbf{a}_D(\mathbf{x}(N-K), \mathbf{u}(N-K)))\}; \qquad K = 1, 2, \ldots, N \tag{3.8-3}$$

with initial value

$$J^*_{NN}(\mathbf{x}(N)) = h(\mathbf{x}(N)). \tag{3.8-4}$$

It should be re-emphasized that Eq. (3.8-3) is simply a formalization of the computational procedure followed in solving the control problem in Section 3.5.

The solution of this recurrence equation is an optimal control law or optimal policy, $\mathbf{u}^*(\mathbf{x}(N-K), N-K)$, $K = 1, 2, \ldots, N$, which is obtained by trying *all* admissible control values at *each* admissible state value. To make the computational procedure feasible it is necessary to quantize the admissible state and control values into a finite number of levels. For example, if the system is second order, the grid of state values would appear as shown in Fig. 3-5. The heavily dotted points are the state values at which each of the quantized control values is to be tried. In this second-order example, the total number of state grid points for each time, $k\,\Delta t$, is $s_1 s_2$, where s_1 is the number of points in the x_1 coordinate direction and s_2 is the number of points in the x_2 coordinate direction. s_1 and s_2 are determined by the relationship

$$s_r = \frac{x_{r\max} - x_{r\min}}{\Delta x_r} + 1; \qquad r = 1, 2, \tag{3.8-5}$$

where it is assumed that Δx_r is selected so that the interval $x_{r\max} - x_{r\min}$

† This difference equation and the performance measure may be a discrete approximation to a continuous system, or they may represent a system that is actually discrete.

‡ To simplify the notation, it is assumed that the state equations and performance measure do not contain k explicitly. The algorithm is easily modified if this is not the case.

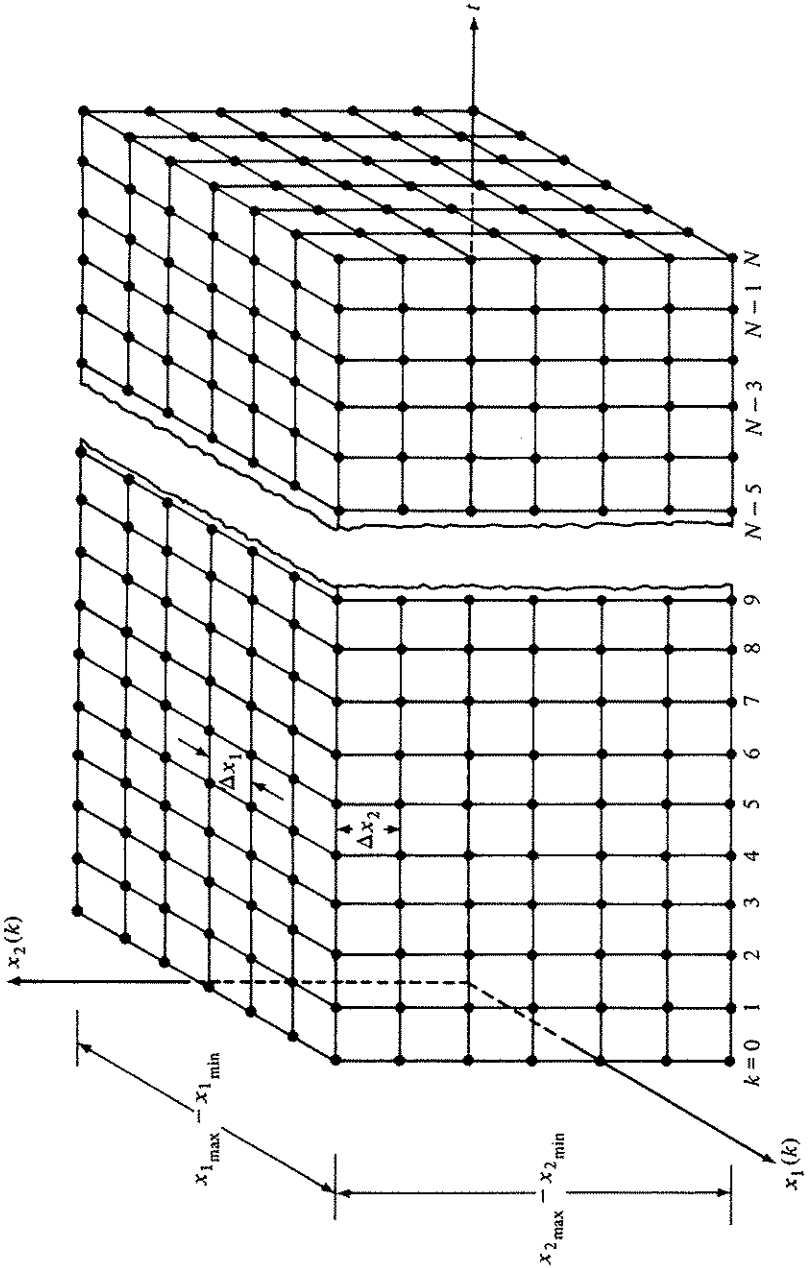

Figure 3-5 Grid of admissible state values

contains an integer number of points. For an nth-order system the number of state grid points for each time, $t = k\,\Delta t$, is

$$S = s_1 \cdot s_2 \cdot \cdots \cdot s_n \tag{3.8-6}$$

where

$$s_r = \frac{x_{r_{max}} - x_{r_{min}}}{\Delta x_r} + 1; \qquad r = 1, 2, \ldots, n, \tag{3.8-7}$$

if it is assumed that the ratio $[x_{r_{max}} - x_{r_{min}}]/\Delta x_r$ is an integer. The admissible range of control values is quantized in exactly the same way; if C is the total number of quantized values of $\mathbf{u}(k)$, then

$$C = c_1 \cdot c_2 \cdot \cdots \cdot c_m \tag{3.8-8}$$

where

$$c_q = \frac{u_{q_{max}} - u_{q_{min}}}{\Delta u_q} + 1; \qquad q = 1, 2, \ldots, m. \tag{3.8-9}$$

In the following development $\mathbf{x}^{(i)}(k)(i = 1, 2, \ldots, S)$ and $\mathbf{u}^{(j)}(k)$ $(j = 1, 2, \ldots, C)$ denote the admissible quantized state and control values at time $t = k\,\Delta t$.

The first step in the computational procedure is to calculate the values of $J^*_{NN}(\mathbf{x}^{(i)}(N))$ $(i = 1, 2, \ldots, S)$ which are used to begin solution of the recurrence equation.

Next, we set $K = 1$, and select the first trial state point by putting $i = 1$ in the subroutine which generates the points $\mathbf{x}^{(i)}(N - K)$. Each control value, $\mathbf{u}^{(j)}(N - K)$ $(j = 1, 2, \ldots, C)$, is then tried at the state value $\mathbf{x}^{(i)}(N - K)$ to determine the next state value, $\mathbf{x}^{(i,j)}(N - K + 1)$, which is used to look up the appropriate value of $J^*_{N-(K-1),N}(\mathbf{x}^{(i,j)}(N - K + 1))$ in computer memory—interpolation will be required if $\mathbf{x}^{(i,j)}(N - K + 1)$ does not fall exactly on a grid value. Using this value of $J^*_{N-(K-1),N}(\mathbf{x}^{(i,j)}(N - K + 1))$ we evaluate

$$C^*_{N-K,N}(\mathbf{x}^{(i)}(N - K), \mathbf{u}^{(j)}(N - K)) = g_D(\mathbf{x}^{(i)}(N - K), \mathbf{u}^{(j)}(N - K)) + J^*_{N-(K-1),N}(\mathbf{x}^{(i,j)}(N - K + 1)), \tag{3.8-10}$$

which is the minimum cost of operation over the final K stages of an N-stage process *assuming that* the control value $\mathbf{u}^{(j)}(N - K)$ is applied at the state value $\mathbf{x}^{(i)}(N - K)$. The idea is to find the value of $\mathbf{u}^{(j)}(N - K)$ that yields $J^*_{N-K,N}(\mathbf{x}^{(i)}(N - K))$, the minimum of $C^*_{N-K,N}(\mathbf{x}^{(i)}(N - K), \mathbf{u}^{(j)}(N - K))$. Only the smallest value of $C^*_{N-K,N}(\mathbf{x}^{(i)}(N - K), \mathbf{u}^{(j)}(N - K))$ and the associated control need to be retained in storage; thus, as each control value is applied at $\mathbf{x}^{(i)}(N - K)$ the $C^*_{N-K,N}(\mathbf{x}^{(i)}(N - K), \mathbf{u}^{(j)}(N - K))$ that results is compared with the variable named COSMIN—the COSt which is the MINi-

mum of those which have been previously calculated. If $C^*_{N-K,N}(\mathbf{x}^{(i)}(N-K), \mathbf{u}^{(j)}(N-K)) < \text{COSMIN}$, then the current value of COSMIN is replaced by this new smaller value. The control that corresponds to the value of COSMIN is also retained—as the variable named UMIN. Naturally, when COSMIN is changed, so is UMIN.

After all control values have been tried at the state value $\mathbf{x}^{(i)}(N-K)$, the numbers stored in COSMIN and UMIN are transferred to storage in arrays named COST$(N-K, I)$ and UOPT$(N-K, I)$, respectively. The arguments $(N-K)$ and I indicate that these values correspond to the state value $\mathbf{x}^{(i)}(N-K)$.

The above procedure is carried out for each quantized state value; then K is increased by one and the procedure is repeated until $K = N$, at which point the COST$(N-K, I)$ and UOPT$(N-K, I)$ arrays are printed out for $K = 1, 2, \ldots, N$ and $I = 1, 2, \ldots, S$. A flow chart of the computational procedure is shown in Fig. 3-6.

The result of the computational procedure is a number for the optimal

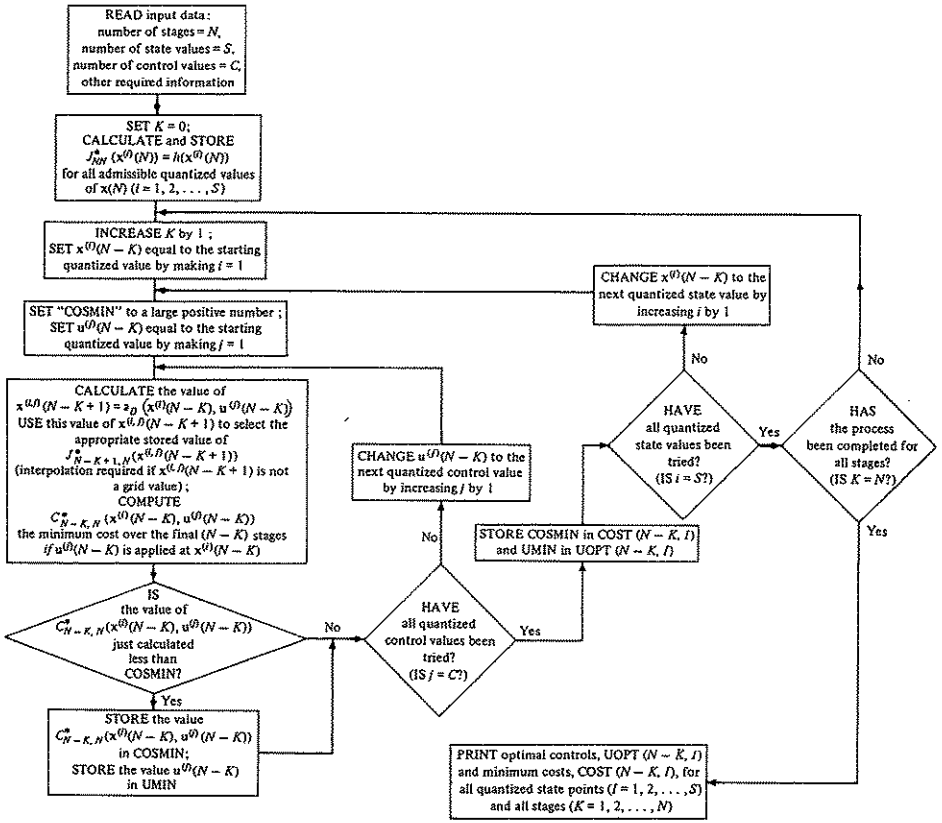

Figure 3-6 Flow chart of the computational procedure

control and the minimum cost at every point on the $(n+1)$-dimensional state-time grid. To calculate the optimal control sequence for a given initial condition, we enter the storage location corresponding to the specified initial condition and extract the control value $\mathbf{u}^*(0)$ and the minimum cost. Next, by solving the state equation we determine the state of the system at $k = 1$ which results from applying $\mathbf{u}^*(0)$ at $k = 0$. The resulting value of $\mathbf{x}(1)$ is then used to reenter the table and extract $\mathbf{u}^*(1)$, and so on. We see that the optimal controller is physically realized by a table look-up device and a generator of piecewise-constant signals.

3.9 CHARACTERISTICS OF DYNAMIC PROGRAMMING SOLUTION

In Section 3.8 we formalized the algorithm for computing the optimal control law from the functional equation

$$J^*_{N-K,N}(\mathbf{x}(N-K)) = \min_{\mathbf{u}(N-K)} \{g_D(\mathbf{x}(N-K), \mathbf{u}(N-K)) + J^*_{N-(K-1),N}(\mathbf{a}_D(\mathbf{x}(N-K), \mathbf{u}(N-K)))\}. \tag{3.8-3}$$

Let us now summarize the important characteristics of the computational procedure and the solution it provides.

Absolute Minimum

Since a direct search is used to solve the functional recurrence equation (3.8-3), the solution obtained is the absolute (or global) minimum. Dynamic programming makes the direct search feasible because instead of searching among the set of *all* admissible controls that cause admissible trajectories, we consider only those controls that satisfy an additional necessary condition—the principle of optimality. This concept is illustrated in Fig. 3-7. S_1 is the set of all controls; S_2 is the set of admissible controls; S_3 is the set of controls that yield admissible state trajectories; S_4 is the set of controls that satisfy the principle of optimality. Without the principle of optimality we would search in the intersection of sets S_2 and S_3.† The dynamic programming algorithm, however, searches only in the shaded region—the intersection of S_2, S_3, and S_4 $(S_2 \cap S_3 \cap S_4)$.

† The set that is the intersection of S_2 and S_3, denoted by $S_2 \cap S_3$, is composed of the elements that belong to *both* S_2 and S_3.

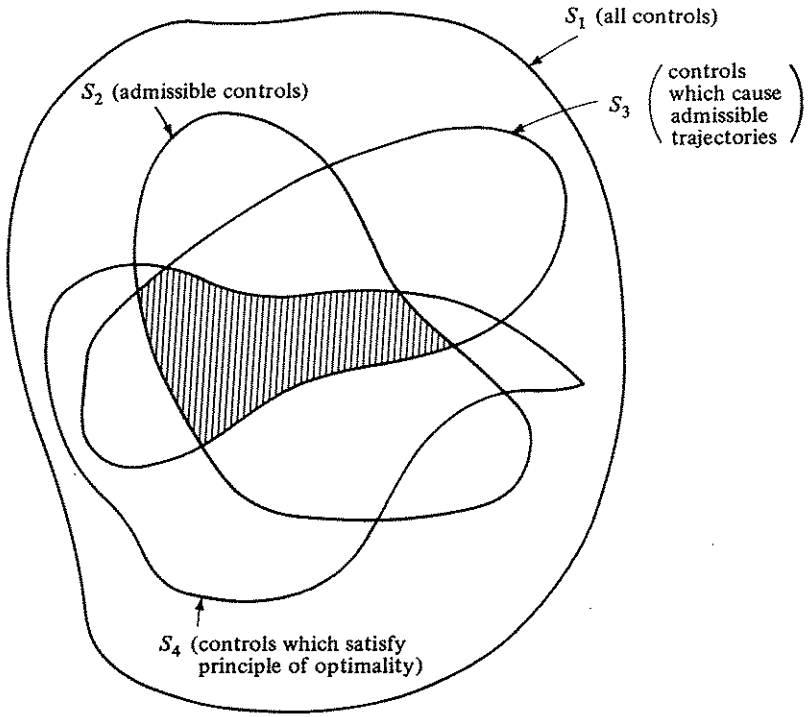

Figure 3-7 Subsets of the control space

Presence of Constraints

As shown in Fig. 3-7, the presence of constraining relations on admissible state and/or control values simplifies the numerical procedure. For example, if the control is a scalar and is constrained by the relationship

$$-1.0 \leq u(t) \leq 1.0, \tag{3.9-1}$$

then in the direct search procedure we need to try only values of u in the allowed interval instead of values of u throughout the interval

$$-\infty < u(t) < \infty. \tag{3.9-2}$$

Form of the Optimal Control

Dynamic programming yields the optimal control in closed-loop or feedback form—for every state value in the admissible region we know what the optimal control is. However, although $\mathbf{u}^*$ is obtained in the form

$$\mathbf{u}^*(t) = \mathbf{f}(\mathbf{x}(t), t), \tag{3.9-3}$$

unfortunately the computational procedure does not yield a nice analytical expression for $\mathbf{f}$. It may be possible to approximate $\mathbf{f}$ in some fashion, but if this cannot be done, the optimal control law must be implemented by extracting the control values from a storage device that contains the solution of Eq. (3.8-3) in tabular form.

A Comparison of Dynamic Programming and Direct Enumeration

Dynamic programming uses the principle of optimality to reduce dramatically the number of calculations required to determine the optimal control law. In order to appreciate more fully the importance of the principle of optimality, let us compare the dynamic programming algorithm with direct enumeration of all possible control sequences.

Consider a first-order control process with one control input. Assume that the admissible state values are quantized into 10 levels, and the admissible control values into four levels. In direct enumeration we try all of the four control values at each of the 10 initial state values for one time increment Δt. In general, this will allow $x(\Delta t)$ to assume any of 40 admissible state values. Assuming that all of these state values are admissible, we apply all four control values at each of the 40 state values and determine the resulting values of $x(2\,\Delta t)$. This procedure continues for the appropriate number of

Table 3-5 AN EXAMPLE COMPARISON OF DYNAMIC PROGRAMMING AND DIRECT ENUMERATION

Number of stages in the process N	*Number of calculations required by dynamic programming*	*Number of calculations required by direct enumeration*	*Number of calculations required by direct enumeration (assuming 50% of state values admissible and distinct)*
1	40	40	40
2	80	200	120
3	120	840	280
4	160	3,400	600
5	200	13,640	1,240
6	240	54,600	2,520
L	$40L$	$\sum_{k=1}^{L} [10 \cdot 4^k]$	$\sum_{k=1}^{L} [20 \cdot 2^k]$

stages. In dynamic programming, at every stage we try four control values at each of 10 state values. Table 3-5 shows a comparison of the number of calculations required by the two methods. The table also includes the number of calculations required for direct enumeration if it is assumed that at the end of each stage only half of the state values are distinct and admissible. The important point is that the number of calculations required by direct enumeration increases exponentially with the number of stages, while the computational requirements of dynamic programming increase linearly.

The Curse of Dimensionality

From the preceding discussion it may seem that perhaps dynamic programming is the answer to all of our problems; unfortunately, there is one serious drawback: for high-dimensional systems the number of high-speed storage locations becomes prohibitive. Bellman calls this difficulty the "curse of dimensionality." To appreciate the nature of the problem, recall that to evaluate $J^*_{N-K,N}$ we need access to the values of $J^*_{N-(K-1),N}$ which have been previously computed. For a third-order system with 100 quantization levels in each state coordinate direction, this means that $10^2 \times 10^2 \times 10^2 = 10^6$ storage locations are required; this number approaches the limit of rapid-access storage available with current computers. There is nothing to prevent us from using low-speed storage; however, this will drastically increase computation time. Of the techniques that have been developed to alleviate the curse of dimensionality, Larson's "state increment dynamic programming"[L-1]seems to be the most promising. There are other methods, however, several of which are explained in [N-1]. [L-2] contains an excellent survey of computational procedures used in dynamic programming.

3.10 ANALYTICAL RESULTS—DISCRETE LINEAR REGULATOR PROBLEMS

In this section we consider the discrete system described by the state equation

$$\mathbf{x}(k+1) = \mathbf{A}(k)\mathbf{x}(k) + \mathbf{B}(k)\mathbf{u}(k). \tag{3.10-1}$$

The states and controls are not constrained by any boundaries. The problem is to find an optimal policy $\mathbf{u}^*(\mathbf{x}(k), k)$ that minimizes the performance measure

$$J = \tfrac{1}{2}\mathbf{x}^T(N)\mathbf{H}\mathbf{x}(N) + \tfrac{1}{2}\sum_{k=0}^{N-1}[\mathbf{x}^T(k)\mathbf{Q}(k)\mathbf{x}(k) + \mathbf{u}^T(k)\mathbf{R}(k)\mathbf{u}(k)], \tag{3.10-2}$$

where

$\mathbf{H}$ and $\mathbf{Q}(k)$ are real symmetric positive semi-definite $n \times n$ matrices.
$\mathbf{R}(k)$ is a real symmetric positive definite $m \times m$ matrix.
N is a fixed integer greater than 0.

The above problem is the discrete counterpart of the continuous linear regulator problem considered in Sections 3.12 and 5.2.† To simplify the notation in the derivation that follows, let us make the assumption that $\mathbf{A}$, $\mathbf{B}$, $\mathbf{R}$, and $\mathbf{Q}$ are constant matrices. The approach we will take is to solve the functional equation (3.7-18). We begin by defining

$$J_{NN}(\mathbf{x}(N)) = \tfrac{1}{2}\mathbf{x}^T(N)\mathbf{H}\mathbf{x}(N) = J^*_{NN}(\mathbf{x}(N)) \triangleq \tfrac{1}{2}\mathbf{x}^T(N)\mathbf{P}(0)\mathbf{x}(N) \tag{3.10-3}$$

where $\mathbf{P}(0) \triangleq \mathbf{H}$. The cost over the final interval is given by

$$\begin{aligned} J_{N-1,N}(\mathbf{x}(N-1), \mathbf{u}(N-1)) &= \tfrac{1}{2}\mathbf{x}^T(N-1)\mathbf{Q}\mathbf{x}(N-1) \\ &\quad + \tfrac{1}{2}\mathbf{u}^T(N-1)\mathbf{R}\mathbf{u}(N-1) + \tfrac{1}{2}\mathbf{x}^T(N)\mathbf{P}(0)\mathbf{x}(N), \end{aligned} \tag{3.10-4}$$

and the minimum cost is

$$J^*_{N-1,N}(\mathbf{x}(N-1)) \triangleq \min_{\mathbf{u}(N-1)} \{J_{N-1,N}(\mathbf{x}(N-1), \mathbf{u}(N-1))\}. \tag{3.10-5}$$

Now $\mathbf{x}(N)$ is related to $\mathbf{u}(N-1)$ by the state equation, so

$$\begin{aligned} J^*_{N-1,N}(\mathbf{x}(N-1)) = \min_{\mathbf{u}(N-1)} \{&\tfrac{1}{2}\mathbf{x}^T(N-1)\mathbf{Q}\mathbf{x}(N-1) + \tfrac{1}{2}\mathbf{u}^T(N-1)\mathbf{R}\mathbf{u}(N-1) \\ &+ \tfrac{1}{2}[\mathbf{A}\mathbf{x}(N-1) + \mathbf{B}\mathbf{u}(N-1)]^T\mathbf{P}(0)[\mathbf{A}\mathbf{x}(N-1) + \mathbf{B}\mathbf{u}(N-1)]\}. \end{aligned} \tag{3.10-6}$$

It is assumed that the admissible controls are not bounded; therefore, to minimize $J_{N-1,N}$ with respect to $\mathbf{u}(N-1)$ we need to consider only those control values for which

$$\begin{bmatrix} \dfrac{\partial J_{N-1,N}}{\partial u_1(N-1)} \\ \dfrac{\partial J_{N-1,N}}{\partial u_2(N-1)} \\ \vdots \\ \dfrac{\partial J_{N-1,N}}{\partial u_m(N-1)} \end{bmatrix} \triangleq \frac{\partial J_{N-1,N}}{\partial \mathbf{u}(N-1)} = \mathbf{0}. \tag{3.10-7}$$

† Equations (3.10-1) and (3.10-2) may be the result of a discrete approximation to a continuous problem, or the formulation for a linear, sampled-data system (see Appendix 2).

Evaluating the indicated partial derivatives gives

$$\mathbf{R}\mathbf{u}(N-1) + \mathbf{B}^T\mathbf{P}(0)[\mathbf{A}\mathbf{x}(N-1) + \mathbf{B}\mathbf{u}(N-1)] = \mathbf{0}.\dagger \quad (3.10\text{-}8)$$

The control values that satisfy this equation may yield a minimum of $J_{N-1,N}$, a maximum, or neither. To investigate further, we form the matrix of second partials given by

$$\begin{bmatrix} \frac{\partial^2 J_{N-1,N}}{\partial u_1^2(N-1)} & \frac{\partial^2 J_{N-1,N}}{\partial u_1(N-1)\partial u_2(N-1)} & \cdots & \frac{\partial^2 J_{N-1,N}}{\partial u_1(N-1)\partial u_m(N-1)} \\ \frac{\partial^2 J_{N-1,N}}{\partial u_2(N-1)\partial u_1(N-1)} & \frac{\partial^2 J_{N-1,N}}{\partial u_2^2(N-1)} & \cdots & \frac{\partial^2 J_{N-1,N}}{\partial u_2(N-1)\partial u_m(N-1)} \\ \vdots & \vdots & & \vdots \\ \frac{\partial^2 J_{N-1,N}}{\partial u_m(N-1)\partial u_1(N-1)} & \frac{\partial^2 J_{N-1,N}}{\partial u_m(N-1)\partial u_2(N-1)} & \cdots & \frac{\partial^2 J_{N-1,N}}{\partial u_m^2(N-1)} \end{bmatrix}$$

$$\triangleq \frac{\partial^2 J_{N-1,N}}{\partial \mathbf{u}^2(N-1)} = \mathbf{R} + \mathbf{B}^T\mathbf{P}(0)\mathbf{B}. \quad (3.10\text{-}9)$$

By assumption $\mathbf{H}$ [and hence $\mathbf{P}(0)$] is a positive semi-definite matrix, and $\mathbf{R}$ is a positive definite matrix. It can be shown that since $\mathbf{P}(0)$ is positive semi-definite, so is $\mathbf{B}^T\mathbf{P}(0)\mathbf{B}$. This means that $\mathbf{R} + \mathbf{B}^T\mathbf{P}(0)\mathbf{B}$ is the sum of a positive definite matrix and a positive semi-definite matrix, and this implies that $\mathbf{R} + \mathbf{B}^T\mathbf{P}(0)\mathbf{B}$ is positive definite.‡ Since $J_{N-1,N}$ is a quadratic function of $\mathbf{u}(N-1)$ and the matrix $\partial^2 J_{N-1,N}/\partial \mathbf{u}^2(N-1)$ is positive definite, the control that satisfies Eq. (3.10-8) yields the absolute, or global, minimum of $J_{N-1,N}$.

Solving (3.10-8) for the optimal control gives

$$\mathbf{u}^*(N-1) = -[\mathbf{R} + \mathbf{B}^T\mathbf{P}(0)\mathbf{B}]^{-1}\mathbf{B}^T\mathbf{P}(0)\mathbf{A}\mathbf{x}(N-1) \triangleq \mathbf{F}(N-1)\mathbf{x}(N-1). \quad (3.10\text{-}10)$$

Since $\mathbf{R} + \mathbf{B}^T\mathbf{P}(0)\mathbf{B}$ is positive definite, the indicated inverse is guaranteed to exist. Substituting the expression for $\mathbf{u}^*(N-1)$ into the equation for $J_{N-1,N}$ gives $J^*_{N-1,N}$, which after terms have been collected becomes

$$J^*_{N-1,N}(\mathbf{x}(N-1)) = \tfrac{1}{2}\mathbf{x}^T(N-1)\{[\mathbf{A} + \mathbf{B}\mathbf{F}(N-1)]^T\mathbf{P}(0)[\mathbf{A} + \mathbf{B}\mathbf{F}(N-1)] + \mathbf{F}^T(N-1)\mathbf{R}\mathbf{F}(N-1) + \mathbf{Q}\}\mathbf{x}(N-1) \triangleq \tfrac{1}{2}\mathbf{x}^T(N-1)\mathbf{P}(1)\mathbf{x}(N-1). \quad (3.10\text{-}11)$$

† The symmetry of $\mathbf{R}$ and $\mathbf{P}(0)$ have also been used here. The reader will find the matrix calculus relationships given in Appendix 1 helpful in following the steps of this derivation.

‡ See Appendix 1.

The definition for $\mathbf{P}(1)$ is clear, by inspection of (3.10-11). The important point is that $J^*_{N-1,N}$ *is of exactly the same form as* $J^*_{N,N}$, which means that when we continue the process one stage further back, the results will have exactly the same form; i.e.,

$$\mathbf{u}^*(N-2) = -[\mathbf{R} + \mathbf{B}^T\mathbf{P}(1)\mathbf{B}]^{-1}\mathbf{B}^T\mathbf{P}(1)\mathbf{A}\mathbf{x}(N-2) \triangleq \mathbf{F}(N-2)\mathbf{x}(N-2), \quad (3.10\text{-}12)$$

and

$$J^*_{N-2,N}(\mathbf{x}(N-2)) = \tfrac{1}{2}\mathbf{x}^T(N-2)\{[\mathbf{A} + \mathbf{B}\mathbf{F}(N-2)]^T\mathbf{P}(1)[\mathbf{A} + \mathbf{B}\mathbf{F}(N-2)] + \mathbf{F}^T(N-2)\mathbf{R}\mathbf{F}(N-2) + \mathbf{Q}\}\mathbf{x}(N-2) \triangleq \tfrac{1}{2}\mathbf{x}^T(N-2)\mathbf{P}(2)\mathbf{x}(N-2). \quad (3.10\text{-}13)$$

If you do not believe this, try it and see.

By induction, for the Kth stage

$$\mathbf{u}^*(N-K) = -[\mathbf{R} + \mathbf{B}^T\mathbf{P}(K-1)\mathbf{B}]^{-1}\mathbf{B}^T\mathbf{P}(K-1)\mathbf{A}\mathbf{x}(N-K) \triangleq \mathbf{F}(N-K)\mathbf{x}(N-K) \quad (3.10\text{-}14)$$

and

$$J^*_{N-K,N}(\mathbf{x}(N-K)) = \tfrac{1}{2}\mathbf{x}^T(N-K)\{[\mathbf{A}+\mathbf{B}\mathbf{F}(N-K)]^T\mathbf{P}(K-1)[\mathbf{A}+\mathbf{B}\mathbf{F}(N-K)] + \mathbf{F}^T(N-K)\mathbf{R}\mathbf{F}(N-K) + \mathbf{Q}\}\mathbf{x}(N-K) \triangleq \tfrac{1}{2}\mathbf{x}^T(N-K)\mathbf{P}(K)\mathbf{x}(N-K). \quad (3.10\text{-}15)$$

In the general time-varying case the same derivation gives

$$\mathbf{u}^*(N-K) = -[\mathbf{R}(N-K) + \mathbf{B}^T(N-K)\mathbf{P}(K-1)\mathbf{B}(N-K)]^{-1} \times \mathbf{B}^T(N-K)\mathbf{P}(K-1)\mathbf{A}(N-K)\mathbf{x}(N-K) \triangleq \mathbf{F}(N-K)\mathbf{x}(N-K) \quad (3.10\text{-}16)$$

$$J^*_{N-K,N}(\mathbf{x}(N-K)) = \tfrac{1}{2}\mathbf{x}^T(N-K)\{[\mathbf{A}(N-K) + \mathbf{B}(N-K)\mathbf{F}(N-K)]^T \times \mathbf{P}(K-1)[\mathbf{A}(N-K)+\mathbf{B}(N-K)\mathbf{F}(N-K)] + \mathbf{F}^T(N-K)\mathbf{R}(N-K)\mathbf{F}(N-K) + \mathbf{Q}(N-K)\}\mathbf{x}(N-K) \triangleq \tfrac{1}{2}\mathbf{x}^T(N-K)\mathbf{P}(K)\mathbf{x}(N-K). \quad (3.10\text{-}17)$$

What are the implications of these results? First, and most important, observe that *the optimal control at each stage is a linear combination of the states;* therefore, the optimal policy is linear state-variable feedback. Notice that the feedback is time-varying, even if **A**, **B**, **R**, and **Q** are *all* constant matrices—this means that the controller for the optimal policy can be implemented by the m time-varying amplifier-summers each with n inputs shown in Fig. 3-8. At the conclusion of Section 3.8 we remarked, "... the optimal controller is physically realized by a table look-up device and a generator of piecewise-constant signals"; when the system is linear and the performance measure quadratic in the states and controls, the only table

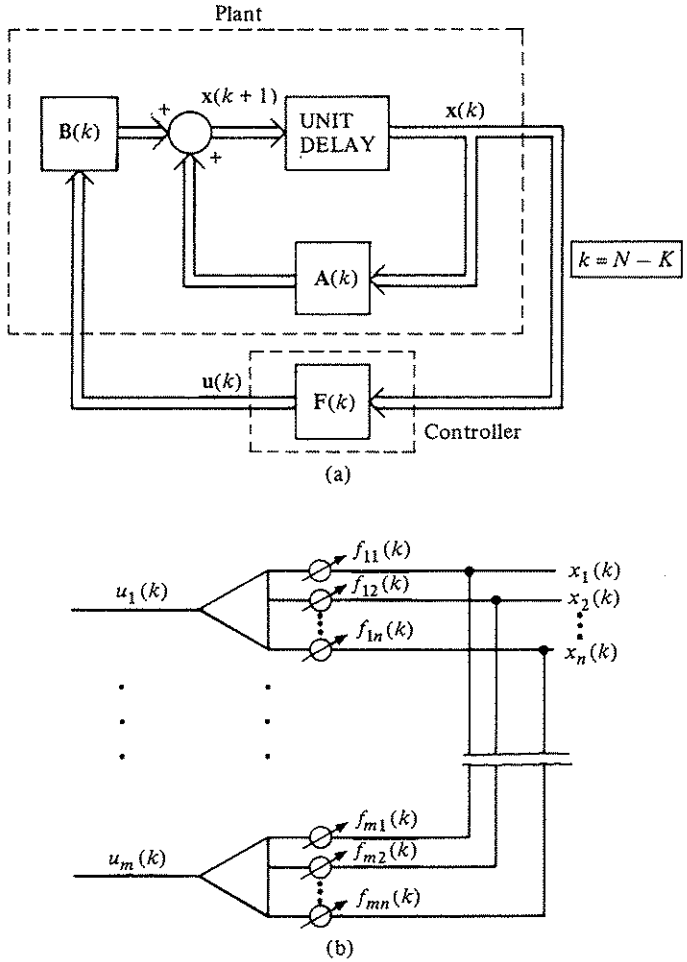

Figure 3-8 (a) Plant and linear time-varying feedback controller (b) Controller configuration

look-up involved in the controller is to determine the appropriate gain settings from stage to stage.

Another important result of the derivation is that the minimum cost for an N-stage process with initial state $\mathbf{x}_0$ is given by

$$\boxed{J^*_{0,N}(\mathbf{x}_0) = \tfrac{1}{2}\mathbf{x}_0^T\mathbf{P}(N)\mathbf{x}_0,} \tag{3.10-18}$$

which follows directly from the definition of $\mathbf{P}(N-K)$. This means that storage of the $\mathbf{P}(N-K)$ matrices for $K = 1, 2, \ldots, N$ provides us with a means of determining the minimum costs for processes of from 1 to N stages.

The computational implications of these results are also important. In order to evaluate the feedback gains and the minimum cost for any initial state, it is necessary only to solve the equations

$$\begin{aligned}\mathbf{F}(N-K) = -&[\mathbf{R}(N-K) + \mathbf{B}^T(N-K)\mathbf{P}(K-1)\mathbf{B}(N-K)]^{-1} \\ &\times \mathbf{B}^T(N-K)\mathbf{P}(K-1)\mathbf{A}(N-K)\end{aligned} \tag{3.10-19}$$

and

$$\begin{aligned}\mathbf{P}(K) = &[\mathbf{A}(N-K) + \mathbf{B}(N-K)\mathbf{F}(N-K)]^T\mathbf{P}(K-1) \\ &\times [\mathbf{A}(N-K) + \mathbf{B}(N-K)\mathbf{F}(N-K)] \\ &+ \mathbf{F}^T(N-K)\mathbf{R}(N-K)\mathbf{F}(N-K) + \mathbf{Q}(N-K)\end{aligned} \tag{3.10-20}$$

with $\mathbf{P}(0) = \mathbf{H}$. We obtain the solution by evaluating $\mathbf{F}(N-1)$ using $\mathbf{P}(0) = \mathbf{H}$, and then substituting $\mathbf{F}(N-1)$ in (3.10-20) to determine $\mathbf{P}(1)$. This constitutes one cycle of the procedure, which we then continue by calculating $\mathbf{F}(N-2)$, $\mathbf{P}(2)$, and so on. The solution is best done by a digital computer; for a reduction in the number of arithmetic operations, it is helpful to define

$$\mathbf{V}(N-K) \triangleq \mathbf{A}(N-K) + \mathbf{B}(N-K)\mathbf{F}(N-K) \tag{3.10-21}$$

so that the procedure is to solve (3.10-19), then (3.10-21), and finally the equation

$$\begin{aligned}\mathbf{P}(K) = &\mathbf{V}^T(N-K)\mathbf{P}(K-1)\mathbf{V}(N-K) \\ &+ \mathbf{F}^T(N-K)\mathbf{R}(N-K)\mathbf{F}(N-K) + \mathbf{Q}(N-K).\end{aligned} \tag{3.10-20a}$$

The **F** and **P** matrices are printed for use in synthesizing optimal controls and determining minimum costs.

It is important to realize that the solution of these equations is equivalent to the computational procedure outlined in Section 3.8; however, because

of the *linear plant dynamics and quadratic performance measure* we obtain the closed-form results given in Eqs. (3.10-16) through (3.10-20a).

The reader may have noticed that the control problem of Section 3.5 is of the linear regulator type. Why then are not the optimal controls in the rightmost columns of Tables 3-2 and 3-3 linear functions of the state values? The answer is that the quantized grid of points is very coarse, causing numerical inaccuracies. When the quantization increments are made much smaller, the linear relationship between the optimal control and state values is apparent; this effect is illustrated in Problems 3-14 through 3-17 at the end of the chapter.

Another important characteristic of the linear regulator problem is that if the system (3.10-1) is completely controllable† and time-invariant, $\mathbf{H} = \mathbf{0}$, and $\mathbf{R}$ and $\mathbf{Q}$ are constant matrices, then the optimal control law is time-invariant for an infinite-stage process; that is

$$\mathbf{F}(N - K) \longrightarrow \mathbf{F} \text{ (a constant matrix)} \quad \text{as} \quad N \longrightarrow \infty.$$

From a physical point of view this means that if a process is to be controlled for a large number of stages the optimal control can be implemented by feedback of the states through a configuration of amplifier-summers as shown in Fig. 3-8(b), but with *fixed* gain factors. One way of determining the constant $\mathbf{F}$ matrix is to solve the recurrence relations for as many stages as required for $\mathbf{F}(N - K)$ to converge to a constant matrix.

Let us now conclude our consideration of the discrete linear regulator problem with the following example.

Example 3.10-1. The linear discrete system

$$\mathbf{x}(k+1) = \begin{bmatrix} 0.9974 & 0.0539 \\ -0.1078 & 1.1591 \end{bmatrix} \mathbf{x}(k) + \begin{bmatrix} 0.0013 \\ 0.0539 \end{bmatrix} u(k) \tag{3.10-22}$$

is to be controlled to minimize the performance measure

$$J = \tfrac{1}{2} \sum_{k=0}^{N-1} [0.25x_1^2(k) + 0.05x_2^2(k) + 0.05u^2(k)]. \tag{3.10-23}$$

Determine the optimal control law.

Equations (3.10-19), (3.10-21), and (3.10-20a) are most easily solved by using a digital computer with $\mathbf{A}$ and $\mathbf{B}$ as specified in Eq. (3.10-22),

† The discrete system of Eq. (3.10-1) with $\mathbf{A}$ and $\mathbf{B}$ constant matrices is completely controllable if and only if the $n \times mn$ matrix

$$[\mathbf{B} \mid \mathbf{AB} \mid \cdots \mid \mathbf{A}^{n-1}\mathbf{B}]$$

is of rank n. For a proof of this theorem, see [P-2].

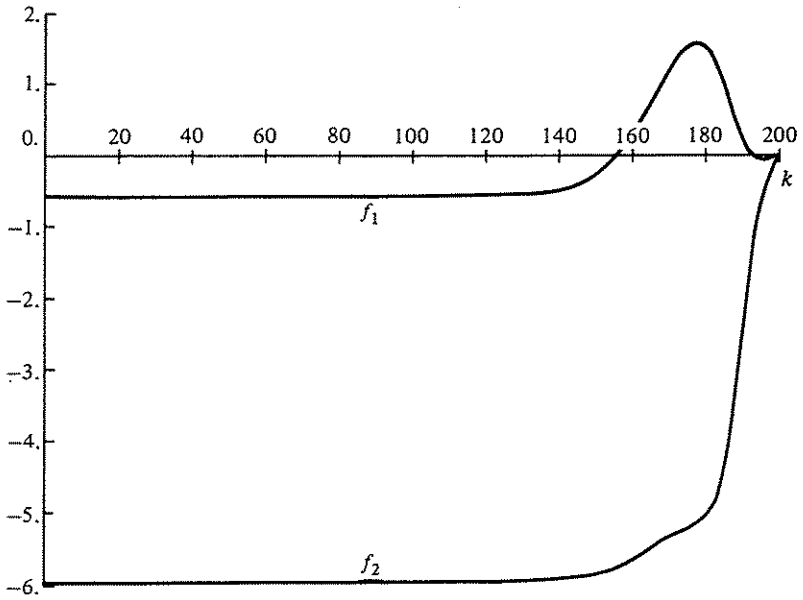

Figure 3-9(a) Feedback gain coefficients for optimal control of a second-order discrete linear regulator

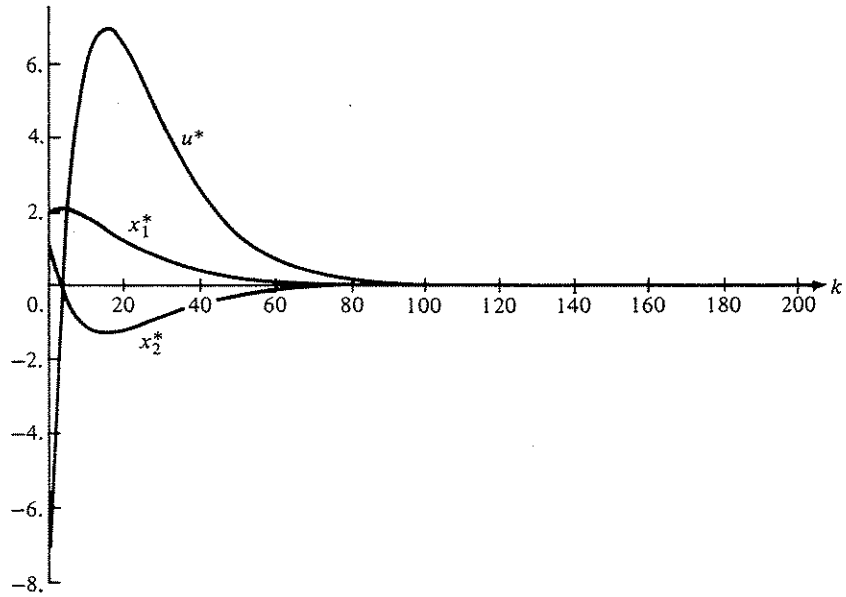

Figure 3-9(b) Optimal control and trajectory for a second-order discrete linear regulator

$$\mathbf{H} = \mathbf{0}, \qquad \mathbf{Q} = \begin{bmatrix} 0.25 & 0.00 \\ 0.00 & 0.05 \end{bmatrix}, \quad \text{and} \quad R = 0.05.$$

The optimal feedback gain matrix $\mathbf{F}(k)$ is shown in Fig. 3-9(a) for $N = 200$. Looking backward from $k = 199$, we observe that at $k \approx 130$ the $\mathbf{F}(k)$ matrix has reached the steady-state value

$$\mathbf{F}(k) = [-0.5522 \quad -5.9668], \qquad 0 \leq k \lesssim 130. \tag{3.10-24}$$

The optimal control history and the optimal trajectory for $\mathbf{x}(0) = [2 \quad 1]^T$ are shown in Fig. 3-9(b). Notice that the optimal trajectory has essentially reached $\mathbf{0}$ at $k = 100$. Thus, we would expect that insignificant performance degradation would be caused by simply using the steady-state value of $\mathbf{F}$ given in (3.10-24) rather than $\mathbf{F}(k)$ as specified in Fig. 3-9(a).

3.11 THE HAMILTON-JACOBI-BELLMAN EQUATION

In our initial exposure to dynamic programming, we approximated continuously operating systems by discrete systems. This approach leads to a recurrence relation that is ideally suited for digital computer solution. In this section we shall consider an alternative approach which leads to a nonlinear *partial* differential equation—the Hamilton-Jacobi-Bellman (H-J-B) equation. The derivation that will be given in this section parallels the development of the functional recurrence equation (3.7-18) in Section 3.7.

The process described by the state equation

$$\dot{\mathbf{x}}(t) = \mathbf{a}(\mathbf{x}(t), \mathbf{u}(t), t) \tag{3.11-1}$$

is to be controlled to minimize the performance measure

$$J = h(\mathbf{x}(t_f), t_f) + \int_{t_0}^{t_f} g(\mathbf{x}(\tau), \mathbf{u}(\tau), \tau)\, d\tau, \tag{3.11-2}$$

where h and g are specified functions, t_0 and t_f are fixed, and τ is a dummy variable of integration. Let us now use the *imbedding principle* to include this problem in a larger class of problems by considering the performance measure

$$J(\mathbf{x}(t), t, \underset{t \leq \tau \leq t_f}{\mathbf{u}(\tau)}) = h(\mathbf{x}(t_f), t_f) + \int_t^{t_f} g(\mathbf{x}(\tau), \mathbf{u}(\tau), \tau)\, d\tau, \tag{3.11-3}$$

where t can be any value less than or equal to t_f, and $\mathbf{x}(t)$ can be any admissible state value. Notice that the performance measure will depend on the

numerical values for $\mathbf{x}(t)$ and t, and on the optimal control history in the interval $[t, t_f]$.

Let us now attempt to determine the controls that minimize (3.11-3) for all admissible $\mathbf{x}(t)$, and for all $t \leq t_f$. The minimum cost function is then

$$J^*(\mathbf{x}(t), t) = \min_{\substack{\mathbf{u}(\tau) \\ t \leq \tau \leq t_f}} \left\{ \int_t^{t_f} g(\mathbf{x}(\tau), \mathbf{u}(\tau), \tau)\, d\tau + h(\mathbf{x}(t_f), t_f) \right\}. \tag{3.11-4}$$

By subdividing the interval, we obtain

$$J^*(\mathbf{x}(t), t) = \min_{\substack{\mathbf{u}(\tau) \\ t \leq \tau \leq t_f}} \left\{ \int_t^{t+\Delta t} g\, d\tau + \int_{t+\Delta t}^{t_f} g\, d\tau + h(\mathbf{x}(t_f), t_f) \right\}. \tag{3.11-5}$$

The principle of optimality requires that

$$J^*(\mathbf{x}(t), t) = \min_{\substack{\mathbf{u}(\tau) \\ t \leq \tau \leq t+\Delta t}} \left\{ \int_t^{t+\Delta t} g\, d\tau + J^*(\mathbf{x}(t + \Delta t), t + \Delta t) \right\}, \tag{3.11-6}$$

where $J^*(\mathbf{x}(t + \Delta t), t + \Delta t)$ is the minimum cost of the process for the time interval $t + \Delta t \leq \tau \leq t_f$ with "initial" state $\mathbf{x}(t + \Delta t)$.

Assuming that the second partial derivatives of J^* exist and are bounded, we can expand $J^*(\mathbf{x}(t + \Delta t), t + \Delta t)$ in a Taylor series about the point $(\mathbf{x}(t), t)$ to obtain

$$\begin{aligned} J^*(\mathbf{x}(t), t) = \min_{\substack{\mathbf{u}(\tau) \\ t \leq \tau \leq t+\Delta t}} \Bigg\{ & \int_t^{t+\Delta t} g\, d\tau + J^*(\mathbf{x}(t), t) + \left[\frac{\partial J^*}{\partial t}(\mathbf{x}(t), t)\right] \Delta t \\ & + \left[\frac{\partial J^*}{\partial \mathbf{x}}(\mathbf{x}(t), t)\right]^T \left[\mathbf{x}(t + \Delta t) - \mathbf{x}(t)\right] \\ & + \text{terms of higher order} \Bigg\}. \end{aligned} \tag{3.11-7}$$

Now for small Δt

$$\begin{aligned} J^*(\mathbf{x}(t), t) = \min_{\mathbf{u}(t)} \{ & g(\mathbf{x}(t), \mathbf{u}(t), t)\, \Delta t + J^*(\mathbf{x}(t), t) \\ & + J_t^*(\mathbf{x}(t), t)\, \Delta t + J_{\mathbf{x}}^{*T}(\mathbf{x}(t), t)[\mathbf{a}(\mathbf{x}(t), \mathbf{u}(t), t)]\, \Delta t \\ & + o(\Delta t) \}, \dagger \end{aligned} \tag{3.11-8}$$

where $o(\Delta t)$ denotes the terms containing $[\Delta t]^2$ and higher orders of Δt that arise from the approximation of the integral and the truncation of the Taylor series expansion. Next, removing the terms involving $J^*(\mathbf{x}(t), t)$ and $J_t^*(\mathbf{x}(t), t)$

† $J_{\mathbf{x}}^* \triangleq \dfrac{\partial J^*}{\partial \mathbf{x}} = \left[\dfrac{\partial J^*}{\partial x_1} \quad \dfrac{\partial J^*}{\partial x_2} \quad \cdots \quad \dfrac{\partial J^*}{\partial x_n}\right]^T$ and $J_t^* \triangleq \dfrac{\partial J^*}{\partial t}$.

from the minimization [since they do not depend on $\mathbf{u}(t)$], we obtain

$$0 = J_t^*(\mathbf{x}(t), t)\,\Delta t + \min_{\mathbf{u}(t)} \{g(\mathbf{x}(t), \mathbf{u}(t), t)\,\Delta t + J_{\mathbf{x}}^{*T}(\mathbf{x}(t), t)[\mathbf{a}(\mathbf{x}(t), \mathbf{u}(t), t)]\,\Delta t + o(\Delta t)\}. \tag{3.11-9}$$

Dividing by Δt and taking the limit as $\Delta t \to 0$ gives†

$$\boxed{0 = J_t^*(\mathbf{x}(t), t) + \min_{\mathbf{u}(t)} \{g(\mathbf{x}(t), \mathbf{u}(t), t) + J_{\mathbf{x}}^{*T}(\mathbf{x}(t), t)[\mathbf{a}(\mathbf{x}(t), \mathbf{u}(t), t)]\}.} \tag{3.11-10}$$

To find the boundary value for this partial differential equation, set $t = t_f$; from Eq. (3.11-4) it is apparent that

$$\boxed{J^*(\mathbf{x}(t_f), t_f) = h(\mathbf{x}(t_f), t_f).} \tag{3.11-11}$$

We define the Hamiltonian $\mathscr{H}$ as

$$\mathscr{H}(\mathbf{x}(t), \mathbf{u}(t), J_{\mathbf{x}}^*, t) \triangleq g(\mathbf{x}(t), \mathbf{u}(t), t) + J_{\mathbf{x}}^{*T}(\mathbf{x}(t), t)[\mathbf{a}(\mathbf{x}(t), \mathbf{u}(t), t)] \tag{3.11-12}$$

and

$$\mathscr{H}(\mathbf{x}(t), \mathbf{u}^*(\mathbf{x}(t), J_{\mathbf{x}}^*, t), J_{\mathbf{x}}^*, t) = \min_{\mathbf{u}(t)} \mathscr{H}(\mathbf{x}(t), \mathbf{u}(t), J_{\mathbf{x}}^*, t), \tag{3.11-13}$$

since the minimizing control will depend on $\mathbf{x}$, $J_{\mathbf{x}}^*$, and t. Using these definitions, we have obtained the Hamilton-Jacobi equation

$$\boxed{0 = J_t^*(\mathbf{x}(t), t) + \mathscr{H}(\mathbf{x}(t), \mathbf{u}^*(\mathbf{x}(t), J_{\mathbf{x}}^*, t), J_{\mathbf{x}}^*, t).} \tag{3.11-10a}$$

This equation is the continuous-time analog of Bellman's recurrence equation (3.7-18); therefore, we shall refer to (3.11-10a) as the "Hamilton-Jacobi-Bellman equation."

Example 3.11-1. A first-order system is described by the differential equation

$$\dot{x}(t) = x(t) + u(t); \tag{3.11-14}$$

† $\lim_{\Delta t \to 0} \left| \frac{o(\Delta t)}{\Delta t} \right| = 0.$

it is desired to find the control law that minimizes the performance measure

$$J = \tfrac{1}{4}x^2(T) + \int_0^T \tfrac{1}{4}u^2(t)\,dt. \tag{3.11-15}$$

The final time T is specified, and the admissible state and control values are not constrained by any boundaries.

Substituting $g = \frac{1}{4}u^2(t)$ and $a = x(t) + u(t)$ into Eq. (3.11-12), we find that the Hamiltonian is (omitting the arguments of J_x^*)

$$\mathscr{H}(x(t), u(t), J_x^*, t) = \tfrac{1}{4}u^2(t) + J_x^*[x(t) + u(t)], \tag{3.11-16}$$

and since the control is unconstrained, a necessary condition that the optimal control must satisfy is

$$\frac{\partial \mathscr{H}}{\partial u} = \frac{1}{2}u(t) + J_x^*(x(t), t) = 0. \tag{3.11-17}$$

Observe that

$$\frac{\partial^2 \mathscr{H}}{\partial u^2} = \frac{1}{2} > 0; \tag{3.11-18}$$

thus, the control that satisfies Eq. (3.11-17) does minimize $\mathscr{H}$. From (3.11-17)

$$u^*(t) = -2J_x^*(x(t), t), \tag{3.11-19}$$

which when substituted in the Hamilton-Jacobi-Bellman equation gives

$$\begin{aligned} 0 &= J_t^* + \tfrac{1}{4}[-2J_x^*]^2 + [J_x^*]x(t) - 2[J_x^*]^2 \\ &= J_t^* - [J_x^*]^2 + [J_x^*]x(t). \end{aligned} \tag{3.11-20}$$

The boundary value is, from (3.11-15),

$$J^*(x(T), T) = \tfrac{1}{4}x^2(T). \tag{3.11-21}$$

One way to solve the Hamilton-Jacobi-Bellman equation is to guess a form for the solution and see if it can be made to satisfy the differential equation and the boundary conditions. Let us assume a solution of the form

$$J^*(x(t), t) = \tfrac{1}{2}K(t)x^2(t), \tag{3.11-22}$$

where $K(t)$ represents an unknown scalar function of t that is to be determined. Notice that

$$J_x^*(x(t), t) = K(t)x(t), \tag{3.11-23}$$

which, together with Eq. (3.11-19), implies that

$$u^*(t) = -2K(t)x(t). \tag{3.11-24}$$

Thus, if a function $K(t)$ can be found such that (3.11-20) and (3.11-21) are satisfied, the optimal control is *linear* feedback of the state—indeed, this was the motivation for selecting the form (3.11-22).

By making $K(T) = \frac{1}{2}$, the assumed solution matches the boundary condition specified by Eq. (3.11-21).

Substituting (3.11-23) for J_x^* and

$$J_t^*(x(t), t) = \tfrac{1}{2}\dot{K}(t)x^2(t)$$

into Eq. (3.11-20) gives

$$0 = \tfrac{1}{2}\dot{K}(t)x^2(t) - K^2(t)x^2(t) + K(t)x^2(t). \tag{3.11-25}$$

Since this equation must be satisfied for all $x(t)$,

$$\tfrac{1}{2}\dot{K}(t) - K^2(t) + K(t) = 0. \tag{3.11-26}$$

$K(t)$ is a scalar function of t; therefore, the solution can be obtained by separation of variables with the result

$$K(t) = \frac{\epsilon^{(T-t)}}{\epsilon^{(T-t)} + \epsilon^{-(T-t)}}. \tag{3.11-27}$$

The optimal control law is then

$$\begin{aligned} u^*(t) &= -2J_x^*(x(t), t) \\ &= -2K(t)x(t). \end{aligned} \tag{3.11-28}$$

Notice that as $T \longrightarrow \infty$, the linear time-varying feedback approaches constant feedback ($K(t) \longrightarrow 1$), and that the controlled system

$$\begin{aligned} \dot{x}(t) &= x(t) - 2x(t) \\ &= -x(t) \end{aligned} \tag{3.11-29}$$

is stable. If this were not the case, the performance measure would be infinite.

3.12 CONTINUOUS LINEAR REGULATOR PROBLEMS

Problems like Example 3.11-1 with linear plant dynamics and quadratic performance criteria are referred to as linear regulator problems. In this section we investigate the use of the Hamilton-Jacobi-Bellman equation as a means of solving the general form of the continuous linear regulator problem.†

The process to be controlled is described by the state equations

$$\dot{\mathbf{x}}(t) = \mathbf{A}(t)\mathbf{x}(t) + \mathbf{B}(t)\mathbf{u}(t), \tag{3.12-1}$$

and the performance measure to be minimized is

$$J = \tfrac{1}{2}\mathbf{x}^T(t_f)\mathbf{H}\mathbf{x}(t_f) + \int_{t_0}^{t_f} \tfrac{1}{2}\left[\mathbf{x}^T(t)\mathbf{Q}(t)\mathbf{x}(t) + \mathbf{u}^T(t)\mathbf{R}(t)\mathbf{u}(t)\right] dt. \tag{3.12-2}$$

$\mathbf{H}$ and $\mathbf{Q}$ are real symmetric positive semi-definite matrices, $\mathbf{R}$ is a real, symmetric positive definite matrix, the initial time t_0 and the final time t_f are specified, and $\mathbf{u}(t)$ and $\mathbf{x}(t)$ are not constrained by any boundaries.

To use the Hamilton-Jacobi-Bellman equation, we first form the Hamiltonian:

$$\begin{aligned} \mathscr{H}(\mathbf{x}(t), \mathbf{u}(t), J_{\mathbf{x}}^*, t) = {} & \tfrac{1}{2}\mathbf{x}^T(t)\mathbf{Q}(t)\mathbf{x}(t) + \tfrac{1}{2}\mathbf{u}^T(t)\mathbf{R}(t)\mathbf{u}(t) + J_{\mathbf{x}}^{*T}(\mathbf{x}(t), t) \\ & \cdot[\mathbf{A}(t)\mathbf{x}(t) + \mathbf{B}(t)\mathbf{u}(t)]. \end{aligned} \tag{3.12-3}$$

A necessary condition for $\mathbf{u}(t)$ to minimize $\mathscr{H}$ is that $\partial\mathscr{H}/\partial\mathbf{u} = \mathbf{0}$; thus

$$\frac{\partial\mathscr{H}}{\partial\mathbf{u}}(\mathbf{x}(t), \mathbf{u}(t), J_{\mathbf{x}}^*, t) = \mathbf{R}(t)\mathbf{u}(t) + \mathbf{B}^T(t)J_{\mathbf{x}}^*(\mathbf{x}(t), t) = \mathbf{0}. \tag{3.12-4}$$

Since the matrix

$$\frac{\partial^2\mathscr{H}}{\partial\mathbf{u}^2} = \mathbf{R}(t) \tag{3.12-5}$$

is positive definite and $\mathscr{H}$ is a quadratic form in $\mathbf{u}$, the control that satisfies Eq. (3.12-4) does minimize $\mathscr{H}$ (globally). Solving Eq. (3.12-4) for $\mathbf{u}^*(t)$ gives

$$\mathbf{u}^*(t) = -\mathbf{R}^{-1}(t)\mathbf{B}^T(t)J_{\mathbf{x}}^*(\mathbf{x}(t), t), \tag{3.12-6}$$

which when substituted in (3.12-3) yields

$$\begin{aligned} \mathscr{H}(\mathbf{x}(t), \mathbf{u}^*(t), J_{\mathbf{x}}^*, t) &= \tfrac{1}{2}\mathbf{x}^T\mathbf{Q}\mathbf{x} + \tfrac{1}{2}J_{\mathbf{x}}^{*T}\mathbf{B}\mathbf{R}^{-1}\mathbf{B}^TJ_{\mathbf{x}}^* \\ &\quad + J_{\mathbf{x}}^{*T}\mathbf{A}\mathbf{x} - J_{\mathbf{x}}^{*T}\mathbf{B}\mathbf{R}^{-1}\mathbf{B}^TJ_{\mathbf{x}}^* \\ &= \tfrac{1}{2}\mathbf{x}^T\mathbf{Q}\mathbf{x} - \tfrac{1}{2}J_{\mathbf{x}}^{*T}\mathbf{B}\mathbf{R}^{-1}\mathbf{B}^TJ_{\mathbf{x}}^* + J_{\mathbf{x}}^{*T}\mathbf{A}\mathbf{x}.‡ \end{aligned} \tag{3.12-7}$$

The Hamilton-Jacobi-Bellman equation is

$$0 = J_t^* + \tfrac{1}{2}\mathbf{x}^T\mathbf{Q}\mathbf{x} - \tfrac{1}{2}J_{\mathbf{x}}^{*T}\mathbf{B}\mathbf{R}^{-1}\mathbf{B}^TJ_{\mathbf{x}}^* + J_{\mathbf{x}}^{*T}\mathbf{A}\mathbf{x}. \tag{3.12-8}$$

† Refer also to Section 5.2, where this same problem is considered and the variational approach is used.

‡ Where no ambiguity exists, the arguments will be omitted.

From Eq. (3.12-2) the boundary condition is

$$J^*(\mathbf{x}(t_f), t_f) = \tfrac{1}{2}\mathbf{x}^T(t_f)\mathbf{H}\mathbf{x}(t_f). \tag{3.12-9}$$

Since we found in Section 3.10 that the minimum cost for the discrete linear regulator problem is a quadratic function of the state, it seems reasonable to guess as a solution the form

$$J^*(\mathbf{x}(t), t) = \tfrac{1}{2}\mathbf{x}^T(t)\mathbf{K}(t)\mathbf{x}(t), \tag{3.12-10}$$

where $\mathbf{K}(t)$ is a real symmetric positive-definite matrix that is to be determined. Substituting this assumed solution in Eq. (3.12-8) yields the result

$$\begin{aligned} 0 = {} & \tfrac{1}{2}\mathbf{x}^T\dot{\mathbf{K}}\mathbf{x} + \tfrac{1}{2}\mathbf{x}^T\mathbf{Q}\mathbf{x} - \tfrac{1}{2}\mathbf{x}^T\mathbf{K}\mathbf{B}\mathbf{R}^{-1}\mathbf{B}^T\mathbf{K}\mathbf{x} \\ & + \mathbf{x}^T\mathbf{K}\mathbf{A}\mathbf{x}. \end{aligned} \tag{3.12-11}$$

The matrix product $\mathbf{KA}$ appearing in the last term can be written as the sum of a symmetric part and an unsymmetric part,

$$\mathbf{KA} = \tfrac{1}{2}[\mathbf{KA} + (\mathbf{KA})^T] + \tfrac{1}{2}[\mathbf{KA} - (\mathbf{KA})^T]. \tag{3.12-12}$$

Using the matrix property $(\mathbf{CD})^T = \mathbf{D}^T\mathbf{C}^T$ and the knowledge that the transpose of a scalar equals itself, we can show that only the symmetric part of $\mathbf{KA}$ contributes anything to (3.12-11). Thus Eq. (3.12-11) can be written

$$\begin{aligned} 0 = {} & \tfrac{1}{2}\mathbf{x}^T\dot{\mathbf{K}}\mathbf{x} + \tfrac{1}{2}\mathbf{x}^T\mathbf{Q}\mathbf{x} - \tfrac{1}{2}\mathbf{x}^T\mathbf{K}\mathbf{B}\mathbf{R}^{-1}\mathbf{B}^T\mathbf{K}\mathbf{x} \\ & + \tfrac{1}{2}\mathbf{x}^T\mathbf{K}\mathbf{A}\mathbf{x} + \tfrac{1}{2}\mathbf{x}^T\mathbf{A}^T\mathbf{K}\mathbf{x}. \end{aligned} \tag{3.12-13}$$

This equation must hold for all $\mathbf{x}(t)$, so

$$\boxed{\begin{aligned} \mathbf{0} = {} & \dot{\mathbf{K}}(t) + \mathbf{Q}(t) - \mathbf{K}(t)\mathbf{B}(t)\mathbf{R}^{-1}(t)\mathbf{B}^T(t)\mathbf{K}(t) \\ & + \mathbf{K}(t)\mathbf{A}(t) + \mathbf{A}^T(t)\mathbf{K}(t), \end{aligned}} \tag{3.12-14}$$

and the boundary condition is [from (3.12-9) and (3.12-10)]

$$\boxed{\mathbf{K}(t_f) = \mathbf{H}.} \tag{3.12.15}$$

Let us consider the implications of this result: first, the H-J-B partial differential equation reduces to a set of ordinary nonlinear differential equations. Second, the $\mathbf{K}(t)$ matrix can be determined by numerical integration

of Eq. (3.12-14) from $t = t_f$ to $t = t_0$ by using the boundary condition $\mathbf{K}(t_f) = \mathbf{H}$. Actually, since the $n \times n$ $\mathbf{K}(t)$ matrix is symmetric, we need to integrate only $n(n+1)/2$ differential equations.

Once $\mathbf{K}(t)$ has been determined, the optimal control law is given by

$$\mathbf{u}^*(t) = -\mathbf{R}^{-1}(t)\mathbf{B}^T(t)\mathbf{K}(t)\mathbf{x}(t). \tag{3.12-16}$$

Thus, by assuming a solution of the form (3.12-10) the optimal control law is linear, time-varying state feedback. It should be pointed out, however, that other forms are possible as solutions of the Hamilton-Jacobi-Bellman equation. Reference [J-1] gives an alternative approach which, under certain conditions, leads to a nonlinear but time-invariant form for the optimal control law.

Our approach in this section leads to Eq. (3.12-14), which is a differential equation of the Riccati type, and thus is referred to as "the Riccati equation"; in Section 5.2 this same equation is developed by variational methods—in linear regulator problems all routes lead to the same destination.

3.13 THE HAMILTON-JACOBI-BELLMAN EQUATION—SOME OBSERVATIONS

We have derived the Hamilton-Jacobi-Bellman equation and used it to solve two examples of the linear regulator type. Let us now make some observations concerning the H-J-B functional equation.

Boundary Conditions

In our derivation we have assumed that t_f is fixed; however, the results still apply if t_f is free. For example, if S represents some hypersurface in the state space and t_f is defined as the first time the system's trajectory intersects S, then the boundary condition is

$$J^*(\mathbf{x}(t_f), t_f) = h(\mathbf{x}(t_f), t_f). \tag{3.13-1}$$

A Necessary Condition

The results we have obtained represent a necessary condition for optimality; that is, the minimum cost function $J^*(\mathbf{x}(t), t)$ must satisfy the Hamilton-Jacobi-Bellman equation.

A Sufficient Condition

Although we have not derived it here, it is also true that if there is a cost function $J'(\mathbf{x}(t), t)$ that satisfies the Hamilton-Jacobi-Bellman equation, then J' is the minimum cost function; i.e.,

$$J'(\mathbf{x}(t), t) = J^*(\mathbf{x}(t), t). \tag{3.13-2}$$

Rigorous proofs of the necessary and sufficient conditions embodied in the H-J-B equation are given in [K-5] and also in [A-2], which contains several examples.

Solution of the Hamilton-Jacobi-Bellman Equation

In both of the examples that we considered, a solution was obtained by guessing a form for the minimum cost function. Unfortunately, we are normally unable to find a solution so easily. In general, the H-J-B equation must be solved by numerical techniques—see [F-1], for example. Actually, a numerical solution involves some sort of a discrete approximation to the exact optimization relationship [Eq. (3.11-10)]; alternatively, by solving the recurrence relation [Eq. (3.7-18)] we obtain the exact solution to a discrete approximation of the Hamilton-Jacobi-Bellman functional equation.

Applications of the Hamilton-Jacobi-Bellman Equation

Two examples of the use of the H-J-B equation to find a solution to optimal control problems have been given; in these examples we used the necessary condition.

Alternatively, if we have in our possession a proposed solution to an optimal control problem, the sufficiency condition can be used to verify the optimality. Several examples of this type are given in [A-2]. It should be pointed out that the derivation of the sufficient condition requires that trajectories remain in certain regions in state-time space. Unfortunately, these regions are not specified in advance—they must be determined in order to use the Hamilton-Jacobi-Bellman equation.

In Chapter 7 we shall see that the Hamilton-Jacobi-Bellman equation provides us with a bridge from the dynamic programming approach to variational methods.

3.14 SUMMARY

The central theme in this chapter has been the development of dynamic programming as it applies to a class of control problems. The principle of optimality is the cornerstone upon which the computational algorithm is built. We have seen that dynamic programming leads to a functional recurrence relation [Eq. (3.7-18)] when a continuous process is approximated by a discrete system. Alternatively, when we deal with a continuous process, the H-J-B partial differential equation results. In either case, a digital computer solution is generally required, and the curse of dimensionality rears its ugly head. In solving the recurrence equation (3.7-18) we obtain an *exact solution to a discrete approximation of the optimization equation*, whereas in performing a numerical solution to the H-J-B equation we obtain an *approximate solution to the exact optimization equation*. Both approaches lead to an optimal control law (closed-loop optimal control). In linear regulator problems we are able to obtain the optimal control law in closed form.

REFERENCES

A-2 Athans, M., and P. L. Falb, *Optimal Control: An Introduction to the Theory and Its Applications*. New York: McGraw-Hill, Inc., 1966.

B-1 Bellman, R. E., and S. E. Dreyfus, *Applied Dynamic Programming*. Princeton, N.J.: Princeton University Press, 1962.

B-2 Bellman, R. E., and R. E. Kalaba, *Dynamic Programming and Modern Control Theory*. New York: Academic Press, 1965.

B-3 Bellman, R. E., *Dynamic Programming*. Princeton, N.J.: Princeton University Press, 1957.

F-1 Fox, L., *Numerical Solution of Ordinary and Partial Differential Equations*. Reading, Mass.: Addison-Wesley Publishing Company, Inc., 1962.

J-1 Johnson, C. D., and J. E. Gibson, "Optimal Control with Quadratic Performance Index and Fixed Terminal Time," *IEEE Trans. Automatic Control* (1964), 355–360.

K-4 Kirk, D. E., "An Introduction to Dynamic Programming," *IEEE Trans. Education* (1967), 212–219.

K-5 Kalman, R. E., "The Theory of Optimal Control and the Calculus of Variations," *Mathematical Optimization Techniques*, R. E. Bellman, ed. Santa Monica, Cal.: The RAND Corporation, 1963.

L-1 Larson, R. E., "Dynamic Programming with Reduced Computational Requirements," *IEEE Trans. Automatic Control* (1965), 135–143.

L-2 Larson, R. E., "A Survey of Dynamic Programming Computational Procedures," *IEEE Trans. Automatic Control* (1967), 767–774.

N-1 Nemhauser, G. L., *Introduction to Dynamic Programming*. New York: John Wiley & Sons, Inc., 1966.

P-1 Pontryagin, L. S., V. G. Boltyanskii, R. V. Gamkrelidze, and E. F. Mischenko,