# ME 433 – STATE SPACE CONTROL

**Lecture 12**



# Dynamic Programming

## 1. Bellman's Principle of Optimality

So far we have considered the variational approach to optimal control. We will consider now dynamic programming, which is based on Bellman's principle of optimality:

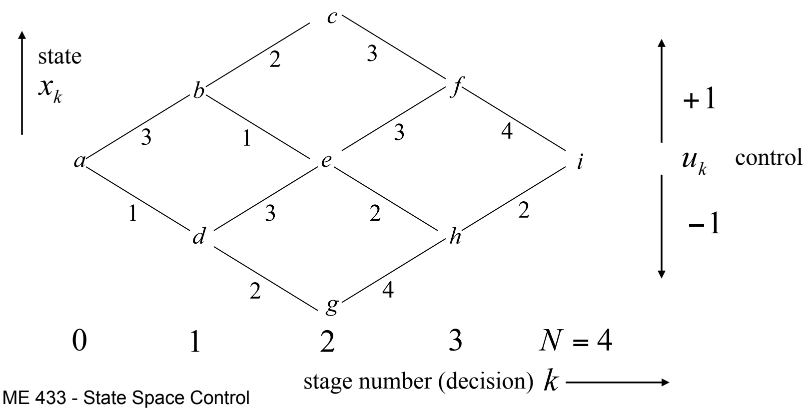
***An optimal policy has the property that no matter what the previous decision (i.e., controls) have been, the remaining decisions must constitute an optimal policy with regard to the state resulting from those previous decisions.***

The principle of optimality plays a role similar to that played by Pontryagin's minimum principle in the variational approach to system control. It serves to limit the number of potentially optimal control strategies that must be investigated. It also implies that optimal control strategies must be determined by working backward from the final state; the optimal control problem is inherently a backward-in-time problem.

# Dynamic Programming

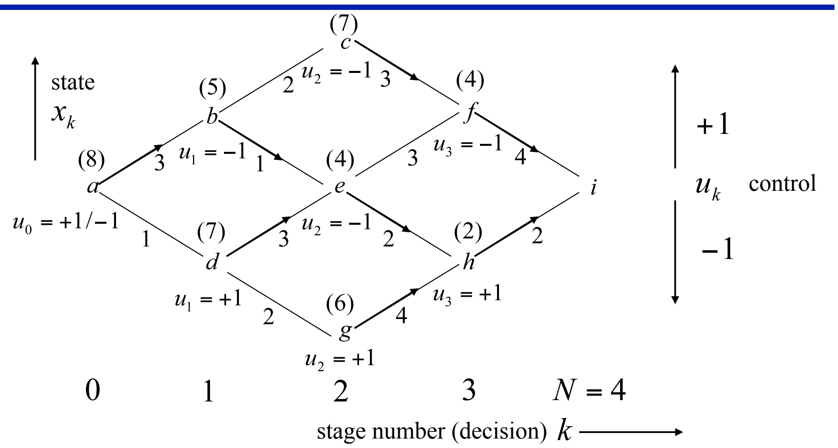
Example: An Aircraft Routing Problem

An aircraft can fly from left to right along the paths shown in the figure below. Intersections *a, b, c, ...* represent cities, and the numbers represent the fuel required to complete each path. We will use Bellman's principle of optimality to solve the minimum-fuel problem with fixed final state and constrained control and state values.






# Dynamic Programming




- State feedback control law
- Solution by dynamic programming may not be unique
- Working forward? Not optimal!
- Any portion of an optimal path is optimal
- Bellman's principle of optimality has reduced the number of decisions

# Dynamic Programming

2. Discrete-Time Systems

The plant is described by the general nonlinear discrete-time dynamical equation

$$x_{k+1} = f(k, x_k, u_k), \qquad k = i, \ldots, N-1$$

with initial condition $x_i$ given. The vector $x_k$ has $n$ components and the vector $u_k$ has $m$ components. Suppose we associate with this plant the performance index

$$J_i(x_i) = \phi(N, x_N) + \sum_{k=i}^{N-1} L(k, x_k, u_k)$$

where $[i, N]$ is the time interval of interest. We want to use Bellman's principle of optimality to find the sequence $u_k$ that minimizes the performance index.



# Dynamic Programming

Suppose we have computed the optimal cost

$$J^*_{k+1}(x_{k+1})$$

for time $k$+1 to the terminal time $N$ for all possible states $x_{k+1}$, and that we have also found the optimal control sequences from time $k$+1 to $N$ for all possible states $x_{k+1}$. The optimal cost results when the optimal control sequences $u^*_{k+1}$, $u^*_{k+2}$, ..., $u^*_{N-1}$ is applied to the plant with a state of $x_{k+1}$. Note that the optimal control sequence depends on $x_{k+1}$. If we apply any arbitrary control $u_k$ at time $k$ and then use the known optimal control sequence from $k$+1 on, the resulting cost will be

$$J_k(x_k) = L(k, x_k, u_k) + J^*_{k+1}(x_{k+1})$$

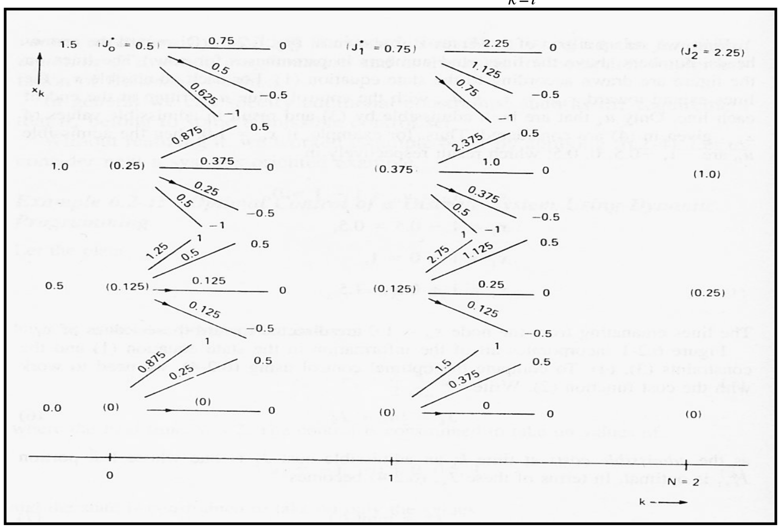
where $x_k$ is the state at time $k$, and $x_{k+1}$ is given by the state equation. According to Bellman, the optimal cost from time $k$ is equal to

$$\boxed{J^*_k(x_k) = \min_{u_k}\left(L(k, x_k, u_k) + J^*_{k+1}(x_{k+1})\right)}$$

and the optimal control $u^*_k$ at time $k$ is the $u_k$ that achieves this minimum.

# Dynamic Programming

Example: $x_{k+1} = x_k + u_k;$ $J_i = x_N^2 + \sum_{k=i}^{N-1} \frac{u_k^2}{2}$






# Dynamic Programming

## 3. Discrete LQR via Dynamic Programming

The plant is described by the linear discrete-time dynamical equation

$$x_{k+1} = A_k x_k + B_k u_k,$$

with initial condition $x_i$ given and final state $x_N$ free. We want to find the sequence $u_k$ on the interval $[i,N]$ that minimizes the performance index:

$$J_i = \frac{1}{2} x_N^T S_N x_N + \frac{1}{2} \sum_{k=i}^{N-1} \left( x_k^T Q_k x_k + u_k^T R_k u_k \right), \qquad S_N \geq 0, Q \geq 0, R > 0$$

Let $k=N$ and write

$$J_N = \frac{1}{2} x_N^T S_N x_N = J_N^*$$

Now let $k=N-1$ and write

$$J_{N-1} = \frac{1}{2} x_{N-1}^T Q_{N-1} x_{N-1} + \frac{1}{2} u_{N-1}^T R_{N-1} u_{N-1} + \frac{1}{2} x_N^T S_N x_N$$

# Dynamic Programming

According to Bellman's principle of optimality,

$$J_k^*(x_k) = \min_{u_k}\left(L(k, x_k, u_k) + J_{k+1}^*(x_{k+1})\right)$$

we find $u_{N-1}$ by minimizing $J_{N-1}$, which can be rewritten as

$$J_{N-1} = \frac{1}{2}x_{N-1}^T Q_{N-1} x_{N-1} + \frac{1}{2}u_{N-1}^T R_{N-1} u_{N-1} + \frac{1}{2}\left(A_{N-1}x_{N-1} + B_{N-1}u_{N-1}\right)^T S_N \left(A_{N-1}x_{N-1} + B_{N-1}u_{N-1}\right)$$

Since there is no input constraint, the minimum is found by setting

$$0 = \frac{\partial J_{N-1}}{\partial u_{N-1}} = R_{N-1}u_{N-1} + B_{N-1}^T S_N \left(A_{N-1}x_{N-1} + B_{N-1}u_{N-1}\right)$$

which gives

$$u_{N-1} = -\left(B_{N-1}^T S_N B_{N-1} + R_{N-1}\right)^{-1} B_{N-1}^T S_N A_{N-1} x_{N-1}$$



# Dynamic Programming

Defining

$$K_{N-1} = \left(B_{N-1}^T S_N B_{N-1} + R_{N-1}\right)^{-1} B_{N-1}^T S_N A_{N-1}$$

we can rewrite

$$u_{N-1}^* = -K_{N-1}x_{N-1}$$

The optimal cost to go from $k=N-1$ is found substituting the optimal control in the expression for $J_{N-1}$,

$$J_{N-1}^* = \frac{1}{2}x_{N-1}^T\left[\left(A_{N-1} - B_{N-1}K_{N-1}\right)^T S_N \left(A_{N-1} - B_{N-1}K_{N-1}\right) + K_{N-1}^T R_{N-1} K_{N-1} + Q_{N-1}\right]x_{N-1}$$

if we define

$$S_{N-1} = \left(A_{N-1} - B_{N-1}K_{N-1}\right)^T S_N \left(A_{N-1} - B_{N-1}K_{N-1}\right) + K_{N-1}^T R_{N-1} K_{N-1} + Q_{N-1}$$

this can be written as

$$J_{N-1}^* = \frac{1}{2}x_{N-1}^T S_{N-1} x_{N-1}$$

# Dynamic Programming

For $k$=$N$

$$J_N = \frac{1}{2} x_N^T S_N x_N$$

For $k$=$N$-1

$$J_{N-1} = \frac{1}{2} x_{N-1}^T Q_{N-1} x_{N-1} + \frac{1}{2} u_{N-1}^T R_{N-1} u_{N-1} + \underbrace{\frac{1}{2} x_N^T S_N x_N}_{J_N^*}$$

For $k$=$N$-2

$$J_{N-2} = \frac{1}{2} x_{N-2}^T Q_{N-2} x_{N-2} + \frac{1}{2} u_{N-2}^T R_{N-2} u_{N-2} + \underbrace{\frac{1}{2} x_{N-1}^T S_{N-1} x_{N-1}}_{J_{N-1}^*}$$

The structure of the problem is the same. To obtain $u^*_{N-2}$ we just need to replace $N$-1 by $N$-2. If we continued to decrement $k$ and apply the optimality principle, the result for each $k$=$N$-1, …, 1, 0 is

$$u_k = -K_k x_k, \quad K_k = \left(B_k^T S_{k+1} B_k + R_k\right)^{-1} B_k^T S_{k+1} A_k$$ Kalman Gain Sequence

$$S_k = \left(A_k - B_k K_k\right)^T S_{k+1} \left(A_k - B_k K_k\right) + K_k^T R_k K_k + Q_k$$ Joseph Stabilized Ricatti Difference Equation (RDE)

$$J_k^* = \frac{1}{2} x_k^T S_k x_k$$ Optimal Cost



# Dynamic Programming

4. Continuous-Time Systems

The plant is described by the general nonlinear continuous-time dynamical equation

$$\dot{x} = f(t, x, u), \qquad t_0 < t < T$$

with initial condition $x_0$ given. The vector $x$ has $n$ components and the vector $u$ has $m$ components. Suppose we associate with this plant the performance index

$$J\left(x(t_0), t_0\right) = \phi\left(T, x(T)\right) + \int_{t_0}^{T} L\left(t, x(t), u(t)\right) dt$$

where $[t_0, T]$ is the time interval of interest. We want to use Bellman's principle of optimality to find the control $u$ that minimizes the performance index and drives the initial state to a final state satisfying

$$\psi\left(T, x(T)\right) = 0$$

# Dynamic Programming

Suppose $t$ is the current time and $t+\Delta t$ is a future time close to $t$. Then, the cost can be rewritten as

$$J(x,t) = \phi(T,x(T)) + \int_{t+\Delta t}^{T} L(\tau,x(\tau),u(\tau))d\tau + \int_{t}^{t+\Delta t} L(\tau,x(\tau),u(\tau))d\tau$$

$$= \int_{t}^{t+\Delta t} L(\tau,x(\tau),u(\tau))d\tau + J(x+\Delta x,t+\Delta t)$$

where $x+\ \Delta x$ is the state at time $t+\Delta t$ that results when the current $u(t)$ and $x(t)$ are used in the state equation. This expression describes all possible costs to go from time $t$ to the final time $T$. According to the optimality principle, the only candidates for $J^*(x,t)$ are those costs that are optimal from $t+\Delta t$ to $T$. Suppose we have computed the optimal cost

$$J^*(x+\Delta x,t+\Delta t)$$

for all possible states $x+\ \Delta x$. Suppose also that the optimal control has determined on the interval $[t+\Delta t,T]$ for each $x+\ \Delta x$. Then, it remains to select the current $u(t)$ on the interval $[t,t+\Delta t]$.



# Dynamic Programming

Hence,

$$\boxed{J^*(x,t) = \min_{\substack{u(\tau)\\ t\le\tau\le t+\Delta t}} \left[ \int_{t}^{t+\Delta t} L(\tau,x(\tau),u(\tau))d\tau + J^*(x+\Delta x,t+\Delta t) \right]}$$

We perform a Taylor series expansion of $J^*(x+\ \Delta x,t+\ \Delta t\ )$ around $(x,t)$ and take an approximation for the integral to write, to first order,

$$J^*(x,t) = \min_{\substack{u(\tau)\\ t\le\tau\le t+\Delta t}} \left[ L\Delta t + J^*(x,t) + \left(\frac{\partial J^*}{\partial x}\right)^T \Delta x + \frac{\partial J^*}{\partial t}\Delta t \right]$$

$$= J^*(x,t) + \frac{\partial J^*}{\partial t}\Delta t + \min_{\substack{u(\tau)\\ t\le\tau\le t+\Delta t}} \left[ L\Delta t + \left(\frac{\partial J^*}{\partial x}\right)^T f\Delta t \right]$$

where we have used that, to the first order, $\Delta x = f\,\Delta t$.

# Dynamic Programming

Thus,

$$-\frac{\partial J^*}{\partial t}\Delta t = \min_{\substack{u(\tau)\\ t\le\tau\le t+\Delta t}}\left[L\Delta t + \left(\frac{\partial J^*}{\partial x}\right)^T f\Delta t\right]$$

Letting $\Delta t \to 0$,

$$-\frac{\partial J^*}{\partial t} = \min_{\substack{u(\tau)\\ t\le\tau\le t+\Delta t}}\left[L + \left(\frac{\partial J^*}{\partial x}\right)^T f\right]$$

Hamilton-Jacobi-Bellman (HJB) Equation

It is solved backward in time from $t=T$, with boundary condition

$J^*((T, x(T))) = \phi(T, x(T))$ on the hypersurface $\psi(T, x(T)) = 0$

The HJB is usually written as

$$-\frac{\partial J^*}{\partial t} = \min_{\substack{u(\tau)\\ t\le\tau\le\Delta t}}\left[H\left(x, u, \frac{\partial J^*}{\partial x}, t\right)\right], \quad H(x,u,\lambda,t) = L(x,u,t) + \lambda^T f(x,u,t)$$



# Dynamic Programming

Example:

# Dynamic Programming

5. Continuous LQR via Dynamic Programming

The plant is described by the linear continuous-time dynamical equation

$$\dot{x} = A(t)x + B(t)u,$$

with initial condition $x_0$ given. We assume that the final time $T$ is fixed and given, and that no function of the final state $\psi$ is specified. We want to find the sequence $u^*(t)$ that minimizes the performance index:

$$J(t_0) = \frac{1}{2}x^T(T)S(T)x(T) + \frac{1}{2}\int_{t_0}^{T}\left(x^T Q(t)x + u^T R(t)u\right)dt$$

The Hamiltonian

$$H = \frac{1}{2}x^T Q(t)x + \frac{1}{2}u^T R(t)u + \lambda^T\left(A(t)x + B(t)u\right)$$

is minimized setting

$$\frac{\partial H}{\partial u} = R(t)u + \lambda^T B(t) = 0 \Rightarrow u^* = -R^{-1}(t)B^T(t)\lambda, \quad \frac{\partial^2 H}{\partial u^2} = R(t) > 0$$



# Dynamic Programming

Then,

$$H^* = \frac{1}{2}x^T Q(t)x + \lambda^T A(t)x - \frac{1}{2}\lambda^T BR^{-1}(t)B^T\lambda$$

Setting $\lambda = J^*_x$, the HJB equation is

$$-J_t^* = \frac{1}{2}x^T Q(t)x + \left(J_x^*\right)^T A(t)x - \frac{1}{2}\left(J_x^*\right)^T BR^{-1}(t)B^T J_x^*$$

with boundary condition

$$J^*(T) = \frac{1}{2}x^T(T)S(T)x(T)$$

We assume that there is a symmetric matrix $S(t)$ such that

$$J^*(t) = \frac{1}{2}x^T(t)S(t)x(t) \qquad \forall t \le T$$

# Dynamic Programming

Substituting this last expression in the HJB equation yields

$$0 = \frac{1}{2}x^T\dot{S}(t)x + \frac{1}{2}x^T Q(t)x + x^T S(t)A(t)x - \frac{1}{2}x^T S(t)BR^{-1}(t)B^T S(t)x$$

$$0 = \frac{1}{2}x^T\left(\dot{S}(t) + 2S(t)A(t) - S(t)BR^{-1}(t)B^T S(t) + Q(t)\right)x$$

$$0 = \frac{1}{2}x^T\left(\dot{S}(t) + A^T(t)S(t) + S(t)A(t) - S(t)BR^{-1}(t)B^T S(t) + Q(t)\right)x$$

Hence,

$$-\dot{S}(t) = A^T(t)S(t) + S(t)A(t) - S(t)BR^{-1}(t)B^T S(t) + Q(t) \qquad \text{(RDE)}$$

$$u^*(t) = -R^{-1}(t)B^T(t)S(t)x(t)$$

$$J^*(t) = \frac{1}{2}x^T(t)S(t)x(t)$$