ME 433 – STATE SPACE CONTROL

Lecture 9

Discrete Dynamic Optimization

1. Multiple-step Discrete-time Finite-Horizon Optimal Control

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

$$x_{k+1} = f\left(k, x_k, u_k\right), \qquad k = 0, \ldots, N-1$$

with initial condition x_0 given. The vector x_k has n components and the vector u_k has m components. Note that this equation contains a set of successive equality constraints which define the state x_k, in terms of the controls u_k, and the known initial condition x_0.

The problem is to find the sequence u_k that minimizes the performance index:

$$J = \phi\left(N, x_N\right) + \sum_{k=0}^{N-1} L\left(k, x_k, u_k\right)$$

Since none of the u_k depends on any of the x_k other than x_0, this is open-loop control.

Discrete Dynamic Optimization

We adjoin the system equations (constraints) to the performance index J with a multiplier sequence $\lambda(k) \in R^n$.

$$\bar{J} = \phi(N, x_N) + \sum_{k=0}^{N-1} \left\{ L(k, x_k, u_k) + \lambda_{k+1}^T \left[f(k, x_k, u_k) - x_{k+1} \right] \right\}$$

For convenience, we define a Hamiltonian at each step k

$$H_k = L(k, x_k, u_k) + \lambda_{k+1}^T f(k, x_k, u_k)$$

Thus,

$$\bar{J} = \phi(N, x_N) + \sum_{k=0}^{N-1} \left\{ H_k - \lambda_{k+1}^T x_{k+1} \right\}$$

$$= \phi(N, x_N) - \lambda_N^T x_N + \sum_{k=1}^{N-1} \left\{ H_k - \lambda_k^T x_k \right\} + H_0$$

Discrete Dynamic Optimization

We want to examine now the increment in $\bar{J}$ due to increments in all the variables x_k, λ_k, and u_k. The final time N is fixed and the initial condition x_0 is given.

$$d\bar{J} = \left[\frac{\partial \phi}{\partial x_N} - \lambda_N^T \right] dx_N + \sum_{k=1}^{N-1} \left\{ \left[\frac{\partial H_k}{\partial x_k} - \lambda_k^T \right] dx_k + \frac{\partial H_k}{\partial u_k} du_k \right\}$$

$$+ \frac{\partial H_0}{\partial x_0} dx_0 + \frac{\partial H_0}{\partial u_0} du_0 + \sum_{k=1}^{N} \left[\frac{\partial H_{k-1}}{\partial \lambda_k} - x_k \right]^T d\lambda_k$$

We make

$$\frac{\partial H_{k-1}}{\partial \lambda_k} - x_k = 0 \Rightarrow x_{k+1} = \frac{\partial H_k}{\partial \lambda_{k+1}} = f(k, x_k, u_k), \qquad k = 0, \ldots, N-1 \quad (1)$$

with initial boundary condition

$$x_{k=0} = x_0$$

Difference equation solved forward in time

Discrete Dynamic Optimization

When the constraint is satisfied, we have

$$d\bar{J} = \left[\frac{\partial \phi}{\partial x_N} - \lambda_N^T\right] dx_N + \sum_{k=1}^{N-1} \left\{ \left[\frac{\partial H_k}{\partial x_k} - \lambda_k^T\right] dx_k + \frac{\partial H_k}{\partial u_k} du_k \right\}$$
$$+ \frac{\partial H_0}{\partial x_0} dx_0 + \frac{\partial H_0}{\partial u_0} du_0$$

We choose the multiplier sequence $\lambda(k) \in R^n$ so that we have

$$\frac{\partial H_k}{\partial x_k} - \lambda_k^T = 0 \Rightarrow \lambda_k^T = \frac{\partial L_k}{\partial x_k} + \lambda_{k+1}^T \frac{\partial f_k}{\partial x_k}, \qquad k = 0, \ldots, N-1 \qquad (2)$$

Difference equation solved backward in time

Discrete Dynamic Optimization

With this choice of λ_k we have

$$d\bar{J} = \left[\frac{\partial \phi}{\partial x_N} - \lambda_N^T\right] dx_N + \sum_{k=0}^{N-1} \frac{\partial H_k}{\partial u_k} du_k + \frac{\partial H_0}{\partial x_0} dx_0$$

- The initial condition x_0 is given, then $dx_0=0$.

- For a *fixed final state*, x_N is given, then $dx_N=0$. For a *free final state*, we need

$$\lambda_{k=N}^T = \frac{\partial \phi}{\partial x_N} \qquad (3)$$

The initial condition for the Two-point Boundary-value Problem (1)-(2) is the known value for x_0. The final condition is either a desired value of x_N or the value of λ_N in (3).

Discrete Dynamic Optimization

Now we have

$$d\bar{J} = \sum_{k=0}^{N-1} \frac{\partial H_k}{\partial u_k} du_k$$

For an extremum, the increment in $\bar{J}$ must be zero for any arbitrary du_k. This can only happen if we have

$$\frac{\partial H_k}{\partial u_k} = \frac{\partial L_k}{\partial u_k} + \lambda_{k+1}^T \frac{\partial f_k}{\partial u_k} = 0, \qquad k = 0,\ldots,N-1$$

Discrete Dynamic Optimization

SUMMARY

System Properties

System Model

$$x_{k+1} = f(k, x_k, u_k)$$

Performance Index

$$J = \phi(N, x_N) + \sum_{k=0}^{N-1} L(k, x_k, u_k)$$

Hamiltonian

$$H_k = L(k, x_k, u_k) + \lambda_{k+1}^T f(k, x_k, u_k)$$

Controller Properties

State Equation

$$x_{k+1} = \frac{\partial H_k}{\partial \lambda_{k+1}} = f(k, x_k, u_k)$$

Costate Equation

$$\lambda_k^T = \frac{\partial H_k}{\partial x_k} = \frac{\partial L_k}{\partial x_k} + \lambda_{k+1}^T \frac{\partial f_k}{\partial x_k}$$

Stationary Condition

$$\frac{\partial H_k}{\partial u_k} = \frac{\partial L_k}{\partial u_k} + \lambda_{k+1}^T \frac{\partial f_k}{\partial u_k} = 0$$

Boundary Condition

$$\left[\frac{\partial \phi}{\partial x_N} - \lambda_N^T\right] dx_N = 0, \quad \frac{\partial H_0}{\partial x_0} dx_0 = 0$$

Discrete Dynamic Optimization

So far, we have derived *necessary conditions* for a stationary point of J that also satisfies the constrains $x_{k+1}=f(k,x_k,u_k)$. We are interested now in *sufficient conditions* for a local minimum. This requires satisfaction of the stationary conditions above, plus establishment of the property that $dJ \geq 0$ for small changes du_k about the stationary point.

$$dJ = \frac{1}{2} dx_N^T \frac{\partial^2 \phi}{\partial x_N^2} dx_N + \frac{1}{2} \sum_{k=0}^{N-1} \begin{bmatrix} dx_k^T & du_k^T \end{bmatrix} \begin{bmatrix} \dfrac{\partial^2 H_k}{\partial x_k^2} & \dfrac{\partial^2 H_k}{\partial x_k \partial u_k} \\ \dfrac{\partial^2 H_k}{\partial u_k \partial x_k} & \dfrac{\partial^2 H_k}{\partial u_k^2} \end{bmatrix} \begin{bmatrix} dx_k \\ du_u \end{bmatrix}$$

The values of dx_k are determined by the sequence du_k from the differential of the plant dynamics

$$dx_{k+1} = \frac{\partial f}{\partial x_k} dx_k + \frac{\partial f}{\partial u_k} du_k, \qquad dx_0 = 0$$

Discrete Dynamic Optimization

Examples:

$$x_{k+1} = a x_k + b u_k, \qquad x_{k=0} = x_0$$

(a) Fixed final state $\quad x_{k=N} = r_N \qquad J_0 = \frac{r}{2} \sum_{k=0}^{N-1} u_k^2$

(b) Free final state $\quad x_{k=N} \rightarrow r_N \qquad J_0 = \frac{1}{2}\left(x_N - r_N\right)^2 + \frac{r}{2} \sum_{k=0}^{N-1} u_k^2$

Discrete Dynamic Optimization

2. Linear Quadratic Regulator (LQR) Problem

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_0 given. We want to find the sequence u_k that minimizes the performance index:

$$J = \frac{1}{2} x_N^T S_N x_N + \frac{1}{2} \sum_{k=0}^{N-1} \left(x_k^T Q_k x_k + u_k^T R_k u_k \right)$$

Linear because of the system dynamics

Quadratic because of the performance index

Regulator because of the absence of a tracking objective---we are interested in regulation around the zero state.

Discrete Dynamic Optimization

We adjoin the system equations (constraints) to the performance index J with a multiplier sequence $\lambda(k) \in R^n$.

$$\bar{J} = \frac{1}{2} x_N^T S_N x_N + \frac{1}{2} \sum_{k=0}^{N-1} \left(x_k^T Q_k x_k + u_k^T R_k u_k \right) + \lambda_{k+1}^T \left(A_k x_k + B_k u_k - x_{k+1} \right)$$

We define a Hamiltonian at each step k

$$H_k = x_k^T Q_k x_k + u_k^T R_k u_k + \lambda_{k+1}^T \left[A_k x_k + B_k u_k \right]$$

Thus, the necessary conditions for a stationary point are:

$$x_{k+1} = \frac{\partial H_k}{\partial \lambda_{k+1}} = A_k x_k + B_k u_k$$

$$\lambda_k^T = \frac{\partial H_k}{\partial x_k} = x_k^T Q_k + \lambda_{k+1}^T A_k$$

$$\frac{\partial H_k}{\partial u_k} = u_k^T R_k + \lambda_{k+1}^T B_k = 0 \Rightarrow \boxed{u_k^T = -\lambda_{k+1}^T B_k R_k^{-1}}$$

Discrete Dynamic Optimization

We must solve the Two-point Boundary-value Problem

$$x_{k+1} = A_k x_k - B_k R_k^{-1} B_k^T \lambda_{k+1}$$
$$\lambda_k = A_k^T \lambda_{k+1} + Q_k x_k$$

for k=0, … , N-1 with boundary conditions

$$x_{k=0} = x_0$$
$$\lambda_{k=N} = S_N x_{k=N} \text{ or } x_{k=N} = x_N \quad \Rightarrow \quad \left[\frac{\partial \phi}{\partial x_N} - \lambda_N^T\right] dx_N = 0, \quad \frac{\partial H_0}{\partial x_0} dx_0 = 0$$

If $|A| \neq 0$ we can invert A in the x_k recursion to yield a reverse-time variant.

$$x_{k+1} = A_k x_k - B_k R_k^{-1} B_k^T \lambda_{k+1} \Rightarrow x_k = A_k^{-1} x_{k+1} + A_k^{-1} B_k R_k^{-1} B_k^T \lambda_{k+1}$$
$$\lambda_k = A_k^T \lambda_{k+1} + Q_k x_k$$

Unfortunately, we are given x_0, not x_N and λ_N simultaneously.

Discrete Dynamic Optimization

2.1 Fixed-Final State and Open-Loop Control

$$x_{k+1} = Ax_k + Bu_k, \qquad x_N = r_N$$
$$J_0 = \frac{1}{2}\sum_{k=0}^{N-1} u_k^T R u_k$$

If $Q \neq 0$, the problem is intractable. The Two-point Boundary-value Problem is now simplified:

$$\begin{aligned} x_{k+1} &= Ax_k - BR^{-1}B^T \lambda_{k+1} \\ \lambda_k &= A^T \lambda_{k+1} + Qx_k \end{aligned} \quad \Rightarrow \quad \begin{aligned} x_{k+1} &= Ax_k - BR^{-1}B^T \lambda_{k+1} \\ \lambda_k &= A^T \lambda_{k+1} \end{aligned}$$

Discrete Dynamic Optimization

The costate equation is decoupled from the state equation, and it has an easy solution:

$$\lambda_k = A^T \lambda_{k+1} \Rightarrow \boxed{\lambda_k = \left(A^T\right)^{N-k} \lambda_N}$$

We replace λ_{k+1} in the state equation and solve:

$$x_{k+1} = Ax_k - BR^{-1}B^T\left(A^T\right)^{N-k-1}\lambda_N \Rightarrow \boxed{x_k = A^k x_0 - \sum_{i=0}^{k-1} A^{k-i-1}BR^{-1}B^T\left(A^T\right)^{N-i-1}\lambda_N}$$

We solve now for λ_N:

$$x_N = A^N x_0 - \sum_{i=0}^{N-1} A^{N-i-1}BR^{-1}B^T\left(A^T\right)^{N-i-1}\lambda_N = r_N$$

$$\lambda_N = -W_C^{-1}(0,N)\left(r_N - A^N x_0\right) \qquad W_C(0,N) = \sum_{i=0}^{N-1} A^{N-i-1}BR^{-1}B^T\left(A^T\right)^{N-i-1}$$

Weighted Controllability Gramian of $[A,B]$

Discrete Dynamic Optimization

Summary: $$\lambda_N = -W_C^{-1}(0,N)\left(r_N - A^N x_0\right)$$

$$W_C(0,N) = \sum_{i=0}^{N-1} A^{N-i-1}BR^{-1}B^T\left(A^T\right)^{N-i-1} = U_N \begin{bmatrix} R^{-1} & & \\ & \ddots & \\ & & R^{-1} \end{bmatrix} U_N^T$$

The inverse of the gramian $W_C(0,N)$ exits if and only if $U_N=[B\ AB\ A^2B\ \ldots\ A^{N-1}B]$ is full rank (system is controllable).

$$\lambda_k = -\left(A^T\right)^{N-k} W_C^{-1}(0,N)\left(r_N - A^N x_0\right)$$

$$x_k = A^k x_0 + \sum_{i=0}^{k-1} A^{k-i-1}BR^{-1}B^T\left(A^T\right)^{N-i-1} W_C^{-1}(0,N)\left(r_N - A^N x_0\right)$$

$$u_k^* = BR^{-1}B^T\left(A^T\right)^{N-k-1} W_C^{-1}(0,N)\left(r_N - A^N x_0\right)$$

Discrete Dynamic Optimization

2.2 Free-Final-State and Closed-Loop Control

$$x_{k+1} = A_k x_k + B_k u_k, \qquad J_0 = \frac{1}{2} x_N^T S_N x_N + \frac{1}{2}\sum_{k=0}^{N-1}\left(x_k^T Q_k x_k + u_k^T R_k u_k\right)$$

The Two-point Boundary-value Problem is:

$$x_{k+1} = A_k x_k - B_k R_k^{-1} B_k^T \lambda_{k+1}$$
$$\lambda_k = A_k^T \lambda_{k+1} + Q_k x_k$$

We need $\left[\frac{\partial \phi}{\partial x_N} - \lambda_N^T\right] dx_N = 0 \Rightarrow \lambda_N^T = \frac{\partial \phi}{\partial x_N} = x_N^T S_N$

Let us assume that this relationship holds for all $k \le N$ (Sweep Method)

$$\lambda_k = S_k x_k$$

Discrete Dynamic Optimization

Substituting in the state equation,

$$x_{k+1} = A_k x_k - B_k R_k^{-1} B_k^T S_{k+1} x_{k+1} \quad \Rightarrow \quad x_{k+1} = \left(I + B_k R_k^{-1} B_k^T S_{k+1}\right)^{-1} A_k x_k$$

Substituting in the costate equation,

$$S_k x_k = A_k^T S_{k+1} x_{k+1} + Q_k x_k = Q_k x_k + A_k^T S_{k+1}\left(I + B_k R_k^{-1} B_k^T S_{k+1}\right)^{-1} A_k x_k$$

Since this must hold for any sequence x_k,

$$S_k = Q_k + A_k^T S_{k+1}\left(I + B_k R_k^{-1} B_k^T S_{k+1}\right)^{-1} A_k$$

Using the matrix inversion lemma $(A + BCD)^{-1} = A^{-1} - A^{-1}B\left(DA^{-1}B + C^{-1}\right)^{-1} DA^{-1}$

$$S_k = A_k^T S_{k+1} A_k - A_k^T S_{k+1} B_k \left(B_k^T S_{k+1} B_k + R_k\right)^{-1} B_k^T S_{k+1} A_k + Q_k$$

Ricatti Difference Equation (RDE)

Discrete Dynamic Optimization

The optimal control is given by,

$$u_k = -R_k^{-1}B_k^T\lambda_{k+1} = -R_k^{-1}B_k^T S_{k+1}x_{k+1} = -R_k^{-1}B_k^T S_{k+1}\left(A_k x_k + B_k u_k\right)$$

Solving for u_k,

$$\begin{aligned} u_k &= -\left(I + R_k^{-1}B_k^T S_{k+1}B_k\right)^{-1} R_k^{-1}B_k^T S_{k+1}A_k x_k \\ &= -\left(R_k + B_k^T S_{k+1}B_k\right)^{-1} B_k^T S_{k+1}A_k x_k \\ &= -K_k x_k \end{aligned}$$

Feedback Control!!!

$$K_k = \left(R_k + B_k^T S_{k+1}B_k\right)^{-1} B_k^T S_{k+1}A_k$$

Kalman Gain Sequence

This expresses u_k as a time-varying, linear, state-variable, feedback control. The feedback gain K_k is computed ahead of time via the sequence S_k, which satisfies the RDE with terminal condition S_N.

Discrete Dynamic Optimization

The optimal control is given by,

$$u_k = -R_k^{-1}B_k^T\lambda_{k+1} = -R_k^{-1}B_k^T S_{k+1}x_{k+1} = -R_k^{-1}B_k^T S_{k+1}\left(A_k x_k + B_k u_k\right)$$

Solving for u_k,

$$\begin{aligned} u_k &= -\left(I + R_k^{-1}B_k^T S_{k+1}B_k\right)^{-1} R_k^{-1}B_k^T S_{k+1}A_k x_k \\ &= -\left(R_k + B_k^T S_{k+1}B_k\right)^{-1} B_k^T S_{k+1}A_k x_k \\ &= -K_k x_k \end{aligned}$$

Feedback Control!!!

$$K_k = \left(R_k + B_k^T S_{k+1}B_k\right)^{-1} B_k^T S_{k+1}A_k$$

Kalman Gain Sequence

This expresses u_k as a time-varying, linear, state-variable, feedback control. The feedback gain K_k is computed ahead of time via the sequence S_k, which satisfies the RDE with terminal condition S_N.

Discrete Dynamic Optimization

$$J_0 = \frac{1}{2} x_N^T S_N x_N + \frac{1}{2}\sum_{k=0}^{N-1}\left(x_k^T Q_k x_k + u_k^T R_k u_k\right)$$

$$= \frac{1}{2} x_0^T S_0 x_0 + \frac{1}{2}\sum_{k=0}^{N-1}\left(x_{k+1}^T S_{k+1} x_{k+1} + x_k^T\left(Q_k - S_k\right)x_k + u_k^T R_k u_k\right)$$

Where we have used the fact that

$$\sum_{k=0}^{N-1} x_{k+1}^T S_{k+1} x_{k+1} - x_k^T S_k x_k = x_N^T S_N x_N - x_0^T S_0 x_0$$

Using the state equation $\boxed{x_{k+1} = A_k x_k + B_k u_k}$

$$J_0 = \frac{1}{2} x_0^T S_0 x_0 + \frac{1}{2}\sum_{k=0}^{N-1}\Big[x_k^T\left(A_k^T S_{k+1} A_k + Q_k - S_k\right)x_k + x_k^T A_k^T S_{k+1} B_k u_k$$

$$+u_k^T B_k^T S_{k+1} A_k x_k + u_k^T\left(B_k^T S_{k+1} B_k + R_k\right)u_k\Big]$$

Discrete Dynamic Optimization

Using the Riccati equation

$$\boxed{S_k = A_k^T S_{k+1} A_k - A_k^T S_{k+1} B_k\left(B_k^T S_{k+1} B_k + R_k\right)^{-1} B_k^T S_{k+1} A_k + Q_k}$$

we can obtain

$$J_0 = \frac{1}{2} x_0^T S_0 x_0 + \frac{1}{2}\sum_{k=0}^{N-1}\Big[x_k^T A_k^T S_{k+1} B_k\left(B_k^T S_{k+1} B_k + R_k\right)^{-1} B_k^T S_{k+1} A_k x_k$$

$$+x_k^T A_k^T S_{k+1} B_k u_k + u_k^T B_k^T S_{k+1} A_k x_k + u_k^T\left(B_k^T S_{k+1} B_k + R_k\right)u_k\Big]$$

$$= \frac{1}{2} x_0^T S_0 x_0 + \frac{1}{2}\sum_{k=0}^{N-1}\left\|\left(R_k + B_k^T S_{k+1} B_k\right)^{-1} B_k^T S_{k+1} A_k x_k + u_k\right\|^2_{R_k + B_k^T S_{k+1} B_k}$$

$$= \frac{1}{2} x_0^T S_0 x_0 \qquad \boxed{u_k = -\left(R_k + B_k^T S_{k+1} B_k\right)^{-1} B_k^T S_{k+1} A_k x_k}$$