Lecture (12)
Apr. 14, 2000

# DYNAMIC PROGRAMMING

(1) $\dot{x} = f(x(t), u(t), t), \qquad x(t_0) = x_0$

(2) $J = \int_{t_0}^{t_f} g(x(t), u(t), t)\,dt + R(x(t_f))$

Fund $u^{opt}$ which minimizes (2) subject to (1)

$u^{opt} = f(x(t), t)$ = closed loop (feedback) control

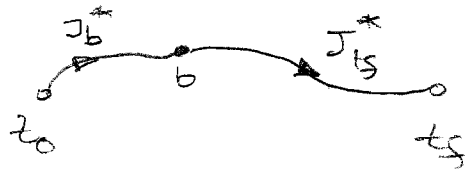
Techniques: 1) Dynamic programming (Bellman)
2) Minimum principle (Pontryagin)

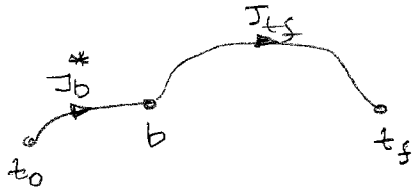
## (3.1). THE OPTIMAL CONTROL LAW

$u^* = f(x(t), t)$ closed-loop or feedback

## (3.2). THE PRINCIPLE OF OPTIMALITY

optimal path



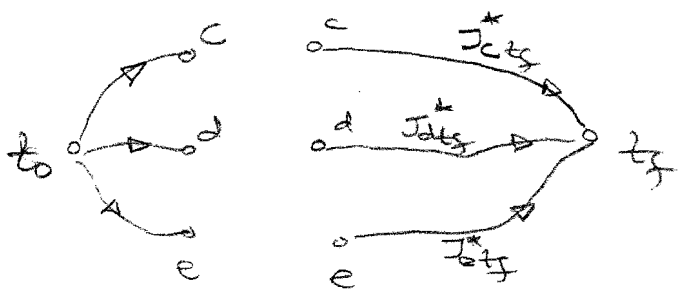




$J^* = J_b^* + J_{tf}^*$ $\qquad$ $J = J_b^* + J_{tf}$ $\qquad$ $J^* < J$

BELLMAN: "AN OPTIMAL POLICY HAS THE PROPERTY THAT WHATEVER THE INITIAL STATE AND INITIAL DECISION ARE, THE REMAINIG DECISIONS MUST CONSTITUTE AN OPTIMAL POLICY WITH REGARD TO THE STATE RESULTING FROM THE FIRST DECISION."

## 3.3. APPLICATION OF THE PRINCIPLE OF OPTIMALITY TO DECISION-MAKING

"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."




Need to compare J's only from $t_0$ to $\begin{cases} c \\ d \\ e \end{cases}$

that is

$J_{0c}$ , $J_{0d}$ , $J_{0e}$

If $t_0 d$ is an optimal ~~trajectory~~ $\Rightarrow t_0 d t_f$ is optimal

(3.4) ROUTING PROBLEM - for COMMUNICATIONS - EXAMPLE

(3.5) AN OPTIMAL CONTROL PROBLEM EXAMPLE

(ex)

$$\begin{cases} \frac{d}{dt}x = ax(t) + bu(t) \\ J = \int_0^+ \lambda u^2(t)\,dt + x^2(T) \end{cases}$$

$0 \le x(t) \le 1.5$
$-1 \le u(t) \le 1$
} admissible set

$t_f = T$

$\lambda$ = weighting factor

PROBLEM:

Drive the system to $x(T) = 0$ without excessive expenditure of control effort.

DISCRETIZATION $\Rightarrow$

$$x(t+\Delta t) = [1 + a\Delta t]\, x(t) + b\,\Delta t\, u(t)$$

$\Delta t$ = small enough

$t = k\Delta t$ , $k = 0, 1, 2, --, N-1$.

$$\Rightarrow \boxed{x((k+1)\Delta t) = [1 + a\Delta t]\, x(k\Delta t) + b\,\Delta t\, u(k\Delta t)}$$

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

$$J = x^2(N) + \lambda\Delta t\,[u^2(0) + u^2(1) + -- + u^2(N-1)]$$

$$J = x^2(N) + \lambda(\Delta t)\sum_{k=0}^{N-1} u^2(k)$$

Assume in our example that

$a = 0$, $T = 2$

$b = 1$, $\Delta t = 1$

$\lambda = 2$

$\Rightarrow$

$$\begin{cases} x(k+1) = x(k) + u(k) & k = 0, 1 \\ J = 2u^2(0) + 2u^2(1) + x^2(2) \end{cases}$$

subject to the constraints

$$0 \le x(k) \le 1.5$$

$$-1 \le u(k) \le 1$$

In addition, we assume the quantized values

$x(k) = 0,\ 0.5,\ 1,\ 1.5$

$u(k) = -1,\ -0.5,\ 0,\ 0.5,\ 1$ $\Rightarrow$ TABLE 3-2

$\Downarrow$ trivial search for the discrete minimum

$\Rightarrow$ DYNAMIC OPTIMAZATION CONVERTED INTO STATIC CONSTRAINED OPTIMIZATION PROBLEM

## 3.7 A RECURRENCE RELATION OF DYNAMIC PROGRAMMIN

$$\dot{x} = a(x(t), u(t)) \quad \text{- time invariant}, \quad u \in U$$

$$J = R(x(t_f)) + \int_0^{t_f} g(x(t), u(t))\,dt \quad , t_f = \text{fixed}$$

$0 \le t \le t_f$, $N$ equally spaced time instants

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

$$x(t+\Delta t) = x(t) + \Delta t\, a(x(t), u(t))$$

$$x(k+1) = x(k) + \Delta t\, a(x(k), u(k))$$

$$\boxed{x(k+1) \triangleq a_D(x(k), u(k))}$$

$$J = R(x(N(\Delta t)) + \int_0^{\Delta t} g\,dt + \int_{\Delta t}^{2\Delta t} g\,dt + \cdots + \int_{(N-1)\Delta t}^{N\Delta t} g\,dt$$

$$\boxed{J = R(x(N)) + \Delta t \sum_{k=0}^{N-1} g(x(k), u(k))}$$

or

$$J = R(x(N)) + \sum_{k=0}^{N-1} g_D(x(k), u(k))$$

RECURSIVE FORMULA

$$J = R(x(N)) + \sum_{k=0}^{N-1} g_D(x(k), u(k))$$

$$x(k+1) \triangleq a_D(x(k), u(k))$$

Define:

(N)

$$J_{NN}(x(N)) = R(x(N))$$

(N-1) $$J_{N-1,N}(x(N-1), u(N-1)) = g_D(x(N-1), u(N-1)) + \underbrace{R(x(N))}_{J_{NN}(x(N))}$$

$$J_{N-1,N}(x(N-1), u(N-1)) = J_{NN}(x(N)) + g_D(x(N-1), u(N-1))$$

one stage process

$$x(N) = a_D(x(N-1), u(N-1)) \quad , \quad x(N-1) = \text{initial state}$$

$$J_{N-1,N}(x(N-1), u(N-1)) = J_{NN}(a_D(x(N-1), u(N-1))) + g_D(x(N-1), u(N-1))$$

optimal cost:

$$J^*_{N-1,N}(x(N-1), u(N-1)) = \min_{u(N-1)} \{g_D(x(N-1), u(N-1)) + J_{NN}(a_D(x(N-1), u(N-1)))\}$$

$$\Rightarrow \boxed{u^*(x(N-1), N-1)}$$

(N-2)

$$J_{N-2,N}(x(N-2), u(N-2), u(N-1)) = g_D(x(N-2), u(N-2)) + J_{N-1,N}(x(N-1), u(N-1))$$

$$J^*_{N-2,N} = \min_{u(N-2)} \{ g_D(x(N-2), u(N-2)) + J_{N-1,N}(x(N-1), u(N-1)) \}$$

$u(N-1)$: found in the previous step, $u(N-1) = f(x(N-1))$

$x(N-1) = a_D(x(N-2), u(N-2))$

$J^*_{N-1,N}$

$\Rightarrow u^*(x(N-2), N-2)$

so that

$$J^*_{N-2,N} = \min_{u(N-2)} \{ J^*_{N-1,N}(a_D(x(N-2), u(N-2)) + g_D(x(N-2), u(N-2)) \}$$

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

$$J^*_{N-K,N}(x(N-K)) = \min_{\substack{u(N-K) \\ u(N-K+1) \\ \vdots \\ u(N-1)}} \{ h(x(N)) + \sum_{k=N-K}^{N-1} g_D(x(k), u(k)) \}$$

which by applying the principle of optimality becomes

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

FUNCTIONAL EQUATION OF DYNAMIC PROGRAMMING

$\Rightarrow u(N-K) = f(x(N-K), N-K)$

feedback form

## 3.8. COMPUTATIONAL PROCEDURE FOR SOLVING OPTIMAL CONTROL PROBLEMS

$$x(k+1) = g_D(x(k), u(k))$$

$$J = h(x(N)) + \sum_{k=0}^{N-1} g_D(x(k), u(k))$$

Dynamic programming $\Rightarrow$

$$J^*_{N-k,N}(x(N-k)) = \min_{u(N-k)} \{ g_D(x(N-k), u(N-k)) + J^*_{N-k-1,N}(g_D(x(N-k), u(N-k)) \}$$

$$J^*_{N,N}(x(N)) = h(x(N))$$

The solution of this recurrence equation is an optimal control law or optimal policy $u^*(x(N-k), N-k)$, which is obtained by trying all admissible control values at each admissible state values. To make the computation procedure feasible it is necessary to quantize the admissible state and control values using finite number of levels.

$\Rightarrow$ DIMENSIONALITY PROBLEM

SEARCH THROUGH ALL ADMISSIBLE REGIONS

$\Rightarrow$ algorithm page 74

direct search is used $\Rightarrow$ absolute (global) minimum

$\Rightarrow$ Dynamic programming yields the optimal control in closed loop form (for every state value in the admissible region we know what the

$$u^* = f(x(t), t)$$