

# Programare dinamică

## 1 Problema de control optimal

Pentru un sistem discret descris în spațiul stărilor de:

$$\mathbf{x}_{k+1} = \mathbf{f}(\mathbf{x}_k, \mathbf{u}_k), \quad k = \overline{1, N-1}, \quad \text{cu condiția inițială } \mathbf{x}_1 = C$$

se cere să se determine o secvență de comenzi admisibile  $u_1^*, u_1^*, \dots, u_{N-1}^*$  pentru care sistemul să urmeze o traiectorie optimală  $x_1^*, x_1^*, \dots, x_N^*$  și să minimizeze funcția de cost

$$J = h(\mathbf{x}_N) + \sum_{k=1}^{N-1} L(\mathbf{x}_k, \mathbf{u}_k)$$

Relația de recurență a programării dinamice este:

$$J_{k,N}^*(\mathbf{x}_k) = \min_{\mathbf{u}_k} \left( L(\mathbf{x}_k, \mathbf{u}_k) + J_{k+1,N}^*(\mathbf{x}_{k+1}) \right) \quad (1)$$

## 2 Exemplu

Se consideră un sistem de ordinul 1 discret, cu ecuația de stare:

$$x_{k+1} = x_k + u_k, \quad k = 1, \dots, N-1 \quad (2)$$

Se caută secvența de comenzi optime  $u_1^*, u_2^*, \dots, u_{N-1}^*$  care minimizează funcția de performanță:

$$J = x_N^2 + \frac{1}{2} \sum_{k=1}^{N-1} u_k^2 \quad (3)$$

Pentru demonstrarea algoritmului se consideră  $N = 3$ . Valorile admisibile pentru  $u_k$  și  $x_k$  sunt:

- $u_k \in (-1, -1/2, 0, 1/2, 1)$
- $x_k \in (0, 1/2, 1, 3/2)$ .

*Soluție*

Pentru  $N = 3$  funcția de performanță este:

$$J = x_3^2 + \frac{1}{2} \sum_{k=1}^2 u_k^2 \quad (4)$$

$$\underline{k=3}, \quad J_{3,3}^*(x_3) = x_3^2$$

| $x_3$ | $J_{3,3}^*(x_3) = x_3^2$ |
|-------|--------------------------|
| 0     | 0                        |
| 1/2   | 1/4                      |
| 1     | 1                        |
| 3/2   | 9/4                      |

$$\underline{k=2}, \quad J_{2,3}^*(x_2) = \min_{u_2} (J_{3,3}^*(x_3) + \frac{1}{2} u_2^2)$$

| $x_2$ | $u_2$ | $x_3 = x_2 + u_2$ | $J_{3,3}^*(x_3)$ | $J_{2,3} = J_{3,3}^*(x_3) + \frac{1}{2}u_2^2$ | valori minime                            |
|-------|-------|-------------------|------------------|-----------------------------------------------|------------------------------------------|
| 0     | -1    | -1                | -                | -                                             | $u_2^* = 0$<br>$J_{2,3}^*(0) = 0$        |
|       | -1/2  | -1/2              | -                | -                                             |                                          |
|       | 0     | 0                 | 0                | $0 + 1/2 \cdot 0^2 = 0$                       |                                          |
|       | 1/2   | 1/2               | 1/4              | $1/4 + 1/2 \cdot 1/4 = 3/8$                   |                                          |
|       | 1     | 1                 | 1                | $1 + 1/2 \cdot 1^2 = 3/2$                     |                                          |
| 1/2   | -1    | -1/2              | -                | -                                             | $u_2^* = -1/2$<br>$J_{2,3}^*(1/2) = 1/8$ |
|       | -1/2  | 0                 | 0                | $0 + 1/2 \cdot 1/4 = 1/8$                     |                                          |
|       | 0     | 1/2               | 1/4              | $1/4 + 1/2 \cdot 0^2 = 1/4$                   |                                          |
|       | 1/2   | 1                 | 1                | $1 + 1/2 \cdot 1/4 = 9/8$                     |                                          |
|       | 1     | 3/2               | 9/4              | $9/4 + 1/2 \cdot 1^2 = 11/4$                  |                                          |
| 1     | -1    | 0                 | 0                | $0 + 1/2 \cdot 1 = 1/2$                       | $u_2^* = -1/2$<br>$J_{2,3}^*(1) = 3/8$   |
|       | -1/2  | 1/2               | 1/4              | $1/4 + 1/2 \cdot 1/4 = 3/8$                   |                                          |
|       | 0     | 1                 | 1                | $1 + 1/2 \cdot 0^2 = 1$                       |                                          |
|       | 1/2   | 3/2               | 9/4              | $9/4 + 1/2 \cdot 1/4 = 19/8$                  |                                          |
|       | 1     | 2                 | -                | -                                             |                                          |
| 3/2   | -1    | 1/2               | 1/4              | $1/4 + 1/2 \cdot 1 = 3/4$                     | $u_2^* = -1$<br>$J_{2,3}^*(3/2) = 3/4$   |
|       | -1/2  | 1                 | 1                | $1 + 1/2 \cdot 1/4 = 9/8$                     |                                          |
|       | 0     | 3/2               | 9/4              | $9/4 + 1/2 \cdot 0^2 = 9/4$                   |                                          |
|       | 1/2   | 2                 | -                | -                                             |                                          |
|       | 1     | 5/2               | -                | -                                             |                                          |

| $x_2$ | $J_{2,3}^*(x_2)$ |
|-------|------------------|
| 0     | 0                |
| 1/2   | 1/8              |
| 1     | 3/8              |
| 3/2   | 3/4              |

| $u_k^*$   | $k = 1$ | $k = 2$ |
|-----------|---------|---------|
| $x_k=0$   | -       | 0       |
| $x_k=1/2$ | -       | -1/2    |
| $x_k=1$   | -       | -1/2    |
| $x_k=3/2$ | -       | -1      |

$\underline{k=1}, J_{1,2}^*(x_1) = \min_{u_1} (J_{2,3}^*(x_2) + \frac{1}{2}u_1^2)$

| $x_1$ | $u_1$ | $x_2 = x_1 + u_1$ | $J_{2,3}^*(x_2)$ | $J_{1,3} = J_{2,3}^*(x_2) + \frac{1}{2}u_1^2$ | valori minime                                                   |
|-------|-------|-------------------|------------------|-----------------------------------------------|-----------------------------------------------------------------|
| 0     | -1    | -1                | -                | -                                             | $u_1^* = 0$<br>$J_{1,3}^*(0) = 0$                               |
|       | -1/2  | -1/2              | -                | -                                             |                                                                 |
|       | 0     | 0                 | 0                | $0 + 1/2 \cdot 0^2 = 0$                       |                                                                 |
|       | 1/2   | 1/2               | 1/8              | $1/8 + 1/2 \cdot 1/4 = 1/4$                   |                                                                 |
|       | 1     | 1                 | 3/8              | $3/8 + 1/2 \cdot 1^2 = 7/8$                   |                                                                 |
| 1/2   | -1    | -1/2              | -                | -                                             | $u_1^* = -1/2 \text{ sau } u_1^* = 0$<br>$J_{1,3}^*(1/2) = 1/8$ |
|       | -1/2  | 0                 | 0                | $0 + 1/2 \cdot 1/4 = 1/8$                     |                                                                 |
|       | 0     | 1/2               | 1/8              | $1/8 + 1/2 \cdot 0^2 = 1/8$                   |                                                                 |
|       | 1/2   | 1                 | 3/8              | $3/8 + 1/2 \cdot 1/4 = 1/2$                   |                                                                 |
|       | 1     | 3/2               | 3/4              | $3/4 + 1/2 \cdot 1^2 = 5/4$                   |                                                                 |
| 1     | -1    | 0                 | 0                | $0 + 1/2 \cdot 1 = 1/2$                       | $u_1^* = -1/2$<br>$J_{1,3}^*(1) = 1/4$                          |
|       | -1/2  | 1/2               | 1/8              | $1/8 + 1/2 \cdot 1/4 = 1/4$                   |                                                                 |
|       | 0     | 1                 | 3/8              | $3/8 + 1/2 \cdot 0^2 = 3/8$                   |                                                                 |
|       | 1/2   | 3/2               | 3/4              | $3/4 + 1/2 \cdot 1/4 = 7/8$                   |                                                                 |
|       | 1     | 2                 | -                | -                                             |                                                                 |
| 3/2   | -1    | 1/2               | 1/8              | $1/8 + 1/2 \cdot 1 = 5/8$                     | $u_1^* = -1/2$<br>$J_{1,3}^*(3/2) = 1/2$                        |
|       | -1/2  | 1                 | 3/8              | $3/8 + 1/2 \cdot 1/4 = 1/2$                   |                                                                 |
|       | 0     | 3/2               | 3/4              | $3/4 + 1/2 \cdot 0^2 = 3/4$                   |                                                                 |
|       | 1/2   | 2                 | -                | -                                             |                                                                 |
|       | 1     | 5/2               | -                | -                                             |                                                                 |

| $x_1$ | $J_{1,3}^*(x_1)$ |
|-------|------------------|
| 0     | 0                |
| 1/2   | 1/8              |
| 1     | 1/4              |
| 3/2   | 1/2              |

| $u_k^*$   | $k = 1$ | $k = 2$ |
|-----------|---------|---------|
| $x_k=0$   | 0       | 0       |
| $x_k=1/2$ | -1/2    | -1/2    |
| $x_k=1$   | -1/2    | -1/2    |
| $x_k=3/2$ | -1/2    | -1      |

Dacă starea inițială este  $x_1 = 1$ , din ultimul tabel se alege valoarea costului optim de la  $x_1 = 1$  la starea finală  $x_3$ :

$$J_{1,3}^*(1) = \frac{1}{4}$$

și valoarea optimă a variabilei de comandă la etapa  $k = 1$   $u_1^* = -1/2$  care transferă sistemul din starea  $x_1 = 1$  în starea  $x_2^* = x_1 + u_1^* = 1 - 1/2 = 1/2$ .

Din tabelul calculat pentru  $k = 2$  se alege valoarea optimă a variabilei de comandă  $u_2^* = -1/2$  care transferă sistemul din  $x_2^* = 1/2$  în  $x_3^* = x_2^* + u_2^* = 1/2 + (-1/2) = 0$ .

Astfel, secvența de comenzi optime este:

$$u^* : -\frac{1}{2}, -\frac{1}{2}$$

și valorile traiectoriei de stare optime începând cu starea inițială  $x_1 = 1$ :

$$x^* : 1, \frac{1}{2}, 0$$

Costul minim este:

$$J_{1,3}^* = \frac{1}{4}$$

### 3 Probleme

**Problema 1.** Urmăriți exemplul din secțiunea 2. Implementați algoritmul în Matlab și utilizați exemplul pentru verificare. Pentru implementare puteți folosi Algoritmul 1 în pseudocod sau scriptul MATLAB din Listing 1.

**Problema 2.** Rezolvați problema din exemplu prin calcul analitic. Considerați  $N = 3$  și utilizați algoritmul din 2- *Programare dinamica-slides.pdf*, pagina 16 (sau din notele de curs 4- *Dynamic Programming.pdf*, pagina 10). Comparati rezultatele cu cele obținute în Problema 1. În scriptul din Listing 1 alegeți un pas mai mic pentru discretizarea valorilor posibile ale stării și comenzii, de exemplu  $dx = 0.1$ ,  $du = 0.1$  și comentați rezultatul.

**Problema 3.** Se consideră un sistem de ordinul 1 cu funcția de transfer:

$$H(s) = \frac{10}{s + 10}$$

1. Simulați sistemul în buclă deschisă pentru o intrare treaptă unitară.
2. Determinați modelul în spațiul stărilor.
3. Obiectivul este ca ieșirea acestui sistem să urmărească o treaptă de 2 unități, într-un interval de timp  $t \in [0, 0.1] \text{ sec}$ , cu consum minim de energie de comandă. Scrieți o funcție de performanță care să fie minimizată pentru a determina variabila de comandă care să fie aplicată ca intrare sistemului.
4. Discretizați problema (ecuația de stare și funcția de performanță).
5. Rezolvați problema utilizând programare dinamică. Implementați algoritmul pe calculator și reprezentați grafic comanda optimală și traiectoria de stare optimală.
6. Modificați și rezolvați problema dacă obiectivul este ca ieșirea sistemului să urmărească o referință  $r(t) = \sin 14t$  pe intervalul  $t \in [0, 1]$ . Reprezentați grafic comanda obținută, traiectoria de stare și referința  $r(t)$ .

## Algoritmul programării dinamice

---

**Algorithm 1** Programare dinamică

---

Se definește sistemul discret

Se definește funcția de cost (expresia din sumă:  $L(x, u)$ )

Se stabilesc comenzile admisibile  $u_{min} \leq u \leq u_{max}$

Se stabilesc stările admisibile  $x_{min} \leq x \leq x_{max}$

Se calculează  $J_N^*(x_N) = h(x_N)$  pentru toate valorile posibile ale stării

**for**  $k = N - 1, N - 2, \dots, 1$  **do**

**for** fiecare stare admisibilă  $x_i$  **do**

**for** fiecare comandă admisibilă  $u_j$  **do**

            Se calculează starea următoare  $x_{urm} = f(x_i, u_j)$

**if**  $x_{urm}$  este admisibilă **then**

                Se calculează  $J_{k+1}^*(x_{urm})$  prin interpolare între valorile anterior calculate ale lui  $J_{k+1}^*$  pentru fiecare valoare admisibilă a stării.

                Se calculează costul până la starea finală  $J_k(x_i, u_j) = J_{k+1}^*(x_{urm}) + L(x_i, u_j)$ .

**else**

$J(x_i, u_j) \leftarrow$  valoare foarte mare

**end if**

**end for**

    Se calculează costul optim pentru pasul  $k$  și starea  $x_i$  ca  $\min_{u_j} J(x_i, u_j)$ . Această valoare se salvează într-un vector  $J_k^*$ , cu lungimea egală cu numărul de stări admisibile.

Se determină valoarea lui  $u_j$  care minimizează pe  $J_k(x_i, u_j)$ . Această valoare se salvează într-o matrice ( $MU_{optim}$ ), pentru fiecare pas  $k$  și fiecare valoarea admisibilă a stării. Matricea va avea o dimensiune egală cu numărul de stări admisibile și cealaltă egală cu numărul de comenzi admisibile).

**end for**

Se formează vectorul de costuri optime pentru fiecare valoare admisibilă a stării la pasul  $k$ :  $J_k^*(x)$

**end for**

Pornind de la starea inițială  $x_1 = C$  și utilizând valorile optime ale comenzii pentru fiecare stare și la fiecare pas, reconstruiți traiectoria de stare optimală  $x_1, x_2, \dots, x_N$  și secvența de comenzi optime  $u_1, u_2, \dots, u_{N-1}$ , astfel:

Setați starea curentă ca starea inițială  $x_1$ .

**for**  $k = 1 : N - 1$  **do**

    Se calculează comanda optimă  $u_{optim_k}$  corespunzătoare stării curente: se selectează din matricea  $MU_{optim}$  valoarea comenzii optime la pasul  $k$ , prin interpolare între valorile calculate ale comenzii la pasul  $k$ , în funcție de valorile admisibile ale stării

    Se calculează starea următoare  $x_{k+1} = f(x_k, u_{optim_k})$ .

    Valorile calculate ale comenzii și stării se salvează pentru reprezentare grafică.

**end for**

---

Listing 1: exemplul1.m

```

1 close all
2 clear all
3 clc
4
5 % Definirea functiei de cost si a dinamicii sistemului
6 L = @(uk) 1/2*uk.^2; % Functia L din functia de cost (din suma)=costul etapei k
7 h = @(xk) xk.^2; % Functia h din functia de cost = costul final
8 xurmator = @(xk,uk) xk+uk; % Ecuatia de stare:  $x_{k+1}$  in functie de  $x_k$  si  $u_k$ 
9
10 % Definirea setului de valori admisibile pentru control si stare
11 umin = -1; umax = 1; du = 0.5; % valoare maxima, minima si pas pentru u
12 xmin = -1; xmax = 1; dx = 0.5; % valoare maxima, minima si pas pentru x
13
14 u = umin:du:umax; % Valorile posibile pentru comanda u (valori discrete)
15 x = xmin:dx:xmax; % Valorile posibile pentru starea x (valori discrete)
16
17 % Definirea parametrilor de timp
18 dt = 0.5; % Pasul de timp (perioada de esantionare)
19 t = 0:dt:1; % Intervalul de timp
20 N = length(t); % Numarul total de pasi de timp
21
22 % Initializarea costului optim cu costul final
23 Jopt = h(x);
24
25 % Algoritmul de programare dinamica - iterativ inapoi in timp
26 for k = N-1:-1:1 % Iteram de la ultimul moment in timp spre primul
27     k
28     for ix = 1:length(x) % Pentru toate valorile posibile ale starii
29         for iu = 1:length(u) % Pentru toate valorile posibile ale comenzii
30             xu = xurmator(x(ix), u(iu)); % Starea urmatoare din ecuatia de stare
31
32             % Verificam daca starea urmatoare este in domeniul admisibil
33             if (xu <= x(end) && xu >= x(1))
34                 Jaux(iu) = L(u(iu)) + interp1(x, Jopt, xu); % Costul etapei curente +
35                     % costul optim de la etapa urmatoare la final
36             else
37                 Jaux(iu) = 99999; % Costul etapei curente = o valoare mare
38                     % pentru starea din afara domeniului
39             end
40         end
41     end
42
43     % Calculam costul minim dintre cele obtinute pentru o stare si toate valorile posibile
44     % ale comenzii
45     [Joptaux(ix), ind] = min(Jaux);
46     % Salvam valoarea comenzii pentru care am obtinut cost minim
47     MU(ix, k) = u(ind(1));
48 end
49
50 Jopt = Joptaux; % Suprascrim vectorul de valori optime ale lui J obtinut pentru
51     % fiecare valoare a starii
52 end
53
54 % Simularea traiectoriei optime folosind secventa de comenzi optime
55 xopt(1) = 1; % Conditia initiala pentru variabila de stare
56 for k = 1:N-1
57     % Din tabelul cu valori optime a comenzii se selecteaza u_optim pentru
58     % valoarea curenta a starii si timpul discret curent, prin interpolare
59     uopt(k) = interp1(x, MU(:,k), xopt(k));
60     % Se calculeaza starea urmatoare din ecuatia de stare
61     xopt(k+1) = xurmator(xopt(k), uopt(k));
62 end
63
64 % Reprezentarea grafica a rezultatelor
65 plot(t, xopt, 'r-*', t(1:N-1), uopt, 'g-*'), grid
66 legend('x-{opt}', 'u-{opt}')
67
68 % Afisarea traiectoriei de stare si a controlului optime
69 xopt
70 uopt

```