14. Динамично програмиране (дискретен вариант)

- Постановка на задачата.
- Принцип на оптималността.
- Предимства и недостатъци на динамичното програмиране.

Динамичното програмиране е разработено в началото на 50-те години на миналия век от американския математик Р. Белман.

Методът се основава на *принципа на оптималността*.

1. Постановка на задачата.

Дадено: $$\dot{\mathbf{X}}(t) = \mathbf{f}(\mathbf{X}(t), \mathbf{U}(t))$$

където: $\mathbf{X}(t)$ е n-мерен вектор на състоянието на системата;

$\mathbf{U}(t)$ - r-мерен вектор на управлението на с-мата;

$\mathbf{f}(.)$ - n-мерен вектор с компоненти нелинейни функции, описващи динамиката на с-мата;

Началното и крайното състояние са съответно:

$$\mathbf{X}_0 = \mathbf{X}(0)\,; \qquad \mathbf{X}_T = \mathbf{X}(T).$$

Търси се: управление $\mathbf{U}(t)$, принадлежащо на областта на допустимите управления $\mathbf{U}(t) \subset \Omega_{\mathbf{U}}$, което да привежда системата от начално състояние $\mathbf{X}_\mathbf{0}$ в крайно състояние $\mathbf{X}_T$ за време $t \in [0;T]$, така че да минимизира критерия за качество

$$I = \varphi(\mathbf{X}_\mathbf{T}) + \int_0^T f_0(\mathbf{X}(t), \mathbf{U}(t))dt \qquad (1)$$

14. Динамично програмиране - дискретен вариант.

Нека решението на задачата на динамичното програмиране е:

оптималната траектория $\mathbf{X}^{*}(t)$, породена от

оптималното управление $\mathbf{U}^{*}(t)$.

Разглеждат се само автономни системи.

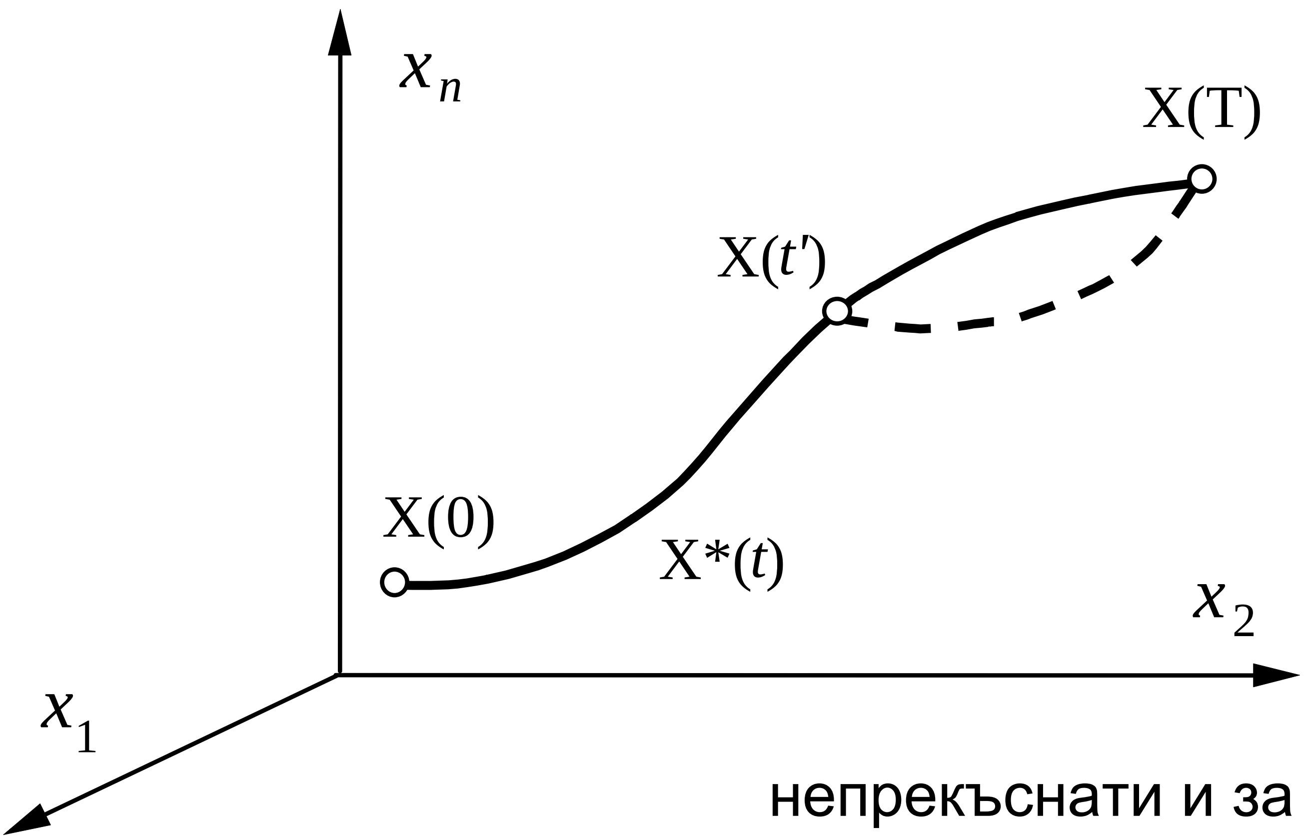
Това не води до загуба на общност, тъй като неавтономната система може да се сведе до горния запис като вместо време се въведе:

- нова променлива $x_{n+1}(t)=t$ и съответно
- още едно диференциално уравнение $\dot{x}_{n+1}(t)=1$, с
- начално условие $x_{n+1}(0)=0$.

2. Принцип на оптималността (ПО).

Нека $\mathbf{X}^*(t)$ е търсената оптимална траектория, $t = t'$, $0 < t' < T$

Според принципа на оптималността, частта от оптималната траектория от междинна точка $\mathbf{X}(t')$ до крайната точка $\mathbf{X}(T)$ е също оптимална траектория. Това значи, че частта от траекторията между тези две точки е оптимална независимо от



предисторията на системата, т.е. по какъв начин тя е достигнала до състоянието $\mathbf{X}(t')$.

ПО дава достатъчно общо ***необходимо условие*** за оптималност, което може да се използва за непрекъснати и за дискретни системи.

3. Идея на метода.

Дадена е крайната цел и се знае управлението в крайната точка (състояние).

Допуска се, че е известна оптималната траектория.

Времето се разглежда в обратна посока и се минимизира функционала (1) :

$$I = \varphi(\mathbf{X_T}) + \int_0^T f_0(\mathbf{X}(t), \mathbf{U}(t))dt$$

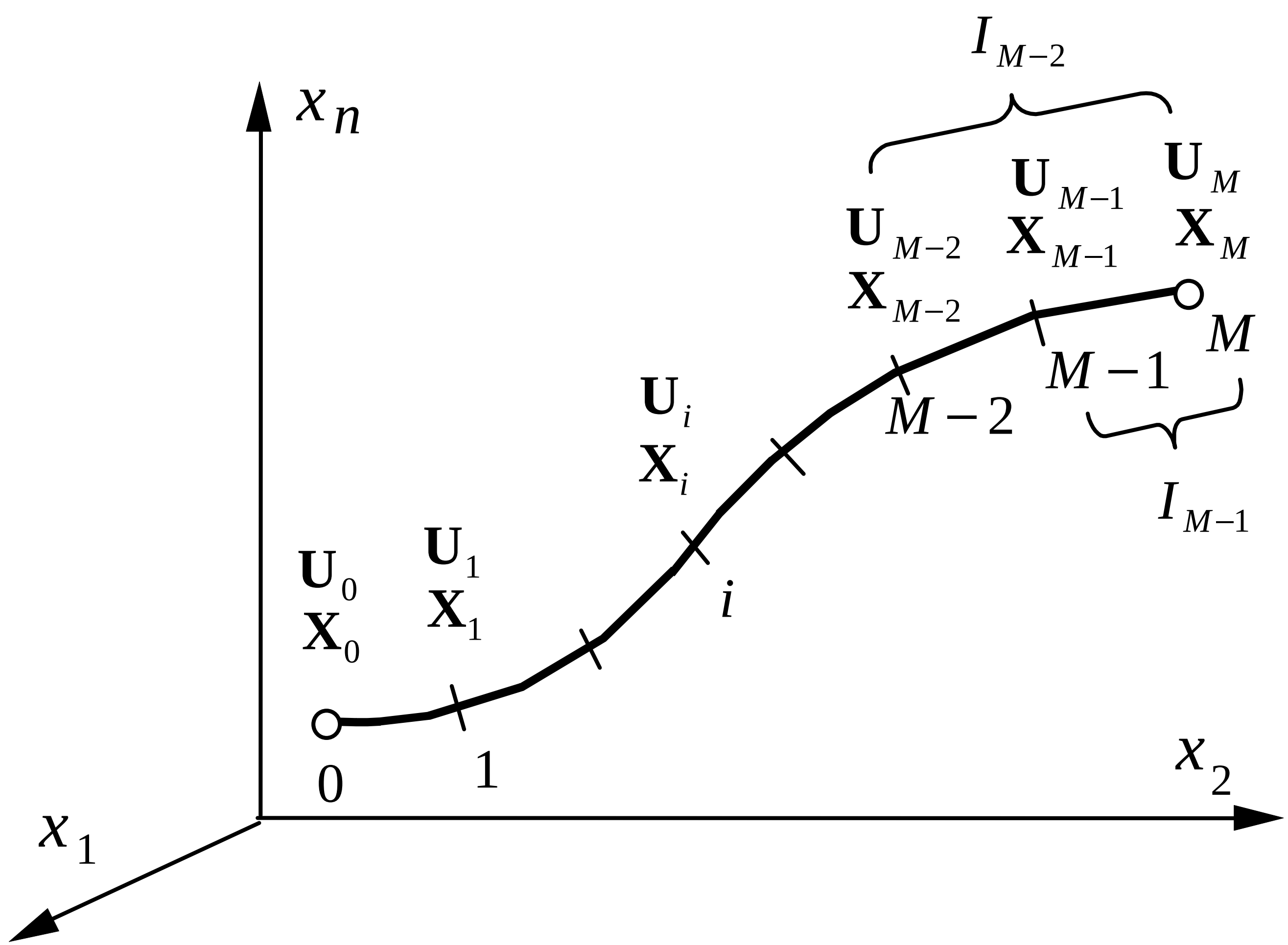
по $\mathbf{U}$, а $\mathbf{X}$ се приема за известно. Когато се стигне в началото се получава $I = I(\mathbf{X})$, т.е. оптимизация на (1) с точност до $\mathbf{X}$.

В началната точка $\mathbf{X}$ се замества с началните условия $\mathbf{X_0}$. Възстановява се нормалният ход на времето (в права посока) и се намират оптималната траектория $\mathbf{X}^*$ и оптималното управление $\mathbf{U}^*$.

Следователно, задачата се разбива на два етапа: обратен (I) и прав (II).

4. Дискретен вариант.

Траекторията от $\mathbf{X_0}$ до $\mathbf{X}_T$ се разделя на M достатъчно малки интервала. Времето за привеждане на системата от начално в крайно състояние е $T = M\Delta t$.



14. Динамично програмиране - дискретен вариант.

Уравнението на състоянието в дискретен вид е:

$$\frac{\mathbf{X}_{i+1}-\mathbf{X}_i}{\Delta t}=\mathbf{f}(\mathbf{X}_i,\mathbf{U}_i), \qquad i=0,1,\dots,M-1, \qquad (2)$$

а критерият за качеството е

$$I=\varphi(\mathbf{X}_\mathrm{T})+\sum_{i=0}^{M-1} f_0(\mathbf{X}_i,\mathbf{U}_i)\Delta t \quad . \qquad (3)$$

I етап (обратен):

Времето се разглежда в обратна посока. Нека $t=T$ и $\mathbf{X}_M=\mathbf{X}_\mathrm{T}$. Системата е в установен режим, грешката и производните и са нули и не се подава допълнително въздействие. Известни са управлението $\mathbf{U}_M=\mathbf{U}_M^*$ и състоянието $\mathbf{X}_M=\mathbf{X}_M^*$.

За функцията на Белман

$$S(\mathbf{X})=\min_{\mathbf{U}\subset\Omega_\mathbf{U}} I(\mathbf{X},\mathbf{U})$$

при $i=M$ от (3) се получава

$$S_M(\mathbf{X}_M)=\varphi(\mathbf{X}_M) \ .$$

Нека $i = M - 1$. Допуска се, че векторите $\mathbf{U}_i, (i = 0,1,\ldots,M-2)$ по някакъв начин са избрани и трябва за дадено състояние $\mathbf{X}_{M-1}$ да се определи векторът $\mathbf{U}_{M-1}$. Съгласно принципа на оптималността $\mathbf{U}_{M-1}$ не зависи от предисторията на системата, а се определя само от състоянието $\mathbf{X}_{M-1}$ и целта на управлението (минимизиране на критерия на качеството). Допуска се че е известно $\mathbf{X}_{M-1} = \mathbf{X}^*_{M-1}$ и се търси управление $\mathbf{U}_{M-1}$, което да минимизира критерия за качество (3) за последния интервал от т. $\mathbf{X}_{M-1}$ до края - т. $\mathbf{X}_M$:

$$I_{M-1} = \varphi(\mathbf{X}_M) + f_0(\mathbf{X}_{M-1}, \mathbf{U}_{M-1})\Delta t \, ,$$

$$S_{M-1} = \min_{\mathbf{U}_{M-1} \subset \Omega_{\mathbf{U}}} I_{M-1} = \min_{\mathbf{U}_{M-1} \subset \Omega_{\mathbf{U}}} [\varphi(\mathbf{X}_M) + f_0(\mathbf{X}_{M-1}, \mathbf{U}_{M-1})\Delta t] . \quad (4)$$

Векторът на състоянието се изразява от (2):

$$\mathbf{X}_M = \mathbf{X}_{M-1} + \mathbf{f}(\mathbf{X}_{M-1}, \mathbf{U}_{M-1})\Delta t$$

и се замества в (4):

$$S_{M-1}(\mathbf{X}^*_{M-1}) = \min_{\mathbf{U}_{M-1} \subset \Omega_{\mathbf{U}}} [\varphi(\mathbf{X}^*_{M-1} + \mathbf{f}(\mathbf{X}^*_{M-1}, \mathbf{U}_{M-1})\Delta t) + f_0(\mathbf{X}^*_{M-1}, \mathbf{U}_{M-1})\Delta t]$$

14. Динамично програмиране - дискретен вариант.

Нека $i = M - 2$. Допуска се, че е известно $\mathbf{X}_{M-2} = \mathbf{X}^*_{M-2}$.
Критерият на качеството за отрязъка на траекторията от т. $\mathbf{X}_{M-2}$ до края (т. $\mathbf{X}_M$) е:

$$I_{M-2} = I_{M-1} + f_0(\mathbf{X}_{M-2}, \mathbf{U}_{M-2})\Delta t .$$

Търси се минимумът на I_{M-2} по векторите $\mathbf{U}_{M-2}$ и $\mathbf{U}^*_{M-1}$.
Минимумът на I_{M-1}, т.е. $S_{M-1}(\mathbf{X}^*_{M-1})$, за произволно състояние $\mathbf{X}^*_{M-1}$ и оптимално управление $\mathbf{U}^*_{M-1}$ вече е намерен.
Следователно:

$$S_{M-2}(\mathbf{X}^*_{M-2}) = \min_{\substack{\mathbf{U}_{M-2}\subset\Omega_{\mathbf{U}} \\ \mathbf{U}^*_{M-1}\subset\Omega_{\mathbf{U}}}} I_{M-2} = \min_{\mathbf{U}_{M-2}\subset\Omega_{\mathbf{U}}} [S_{M-1}(\mathbf{X}^*_{M-1}) + f_0(\mathbf{X}^*_{M-2}, \mathbf{U}_{M-2})\Delta t]$$

$$= \min_{\mathbf{U}_{M-2}\subset\Omega_{\mathbf{U}}} [S_{M-1}(\mathbf{X}^*_{M-2} + \mathbf{f}(\mathbf{X}^*_{M-2}, \mathbf{U}_{M-2})\Delta t) + f_0(\mathbf{X}^*_{M-2}, \mathbf{U}_{M-2})\Delta t].$$

14. Динамично програмиране - дискретен вариант.

При $i = M - k$ критерият за качество за последните k интервала (от т. $\mathbf{X}_{M-k}$ до края) се дава с рекурентната формула:

$$I_{M-k} = I_{M-k+1} + f_0(\mathbf{X}_{M-k}, \mathbf{U}_{M-k})\Delta t \ .$$

а оптималната му стойност - с рекурентната формула на Белман за дискретни процеси:

$$S_{M-k}(\mathbf{X}^*_{M-k}) = \min_{\substack{\mathbf{U}_{M-k} \subset \Omega_{\mathbf{U}} \\ \mathbf{U}^*_{M-k+1} \subset \Omega_{\mathbf{U}}}} I_{M-k} =$$

$$= \min_{\mathbf{U}_{M-k} \subset \Omega_{\mathbf{U}}} [S_{M-k+1}(\mathbf{X}^*_{M-k+1}) + f_0(\mathbf{X}^*_{M-k}, \mathbf{U}_{M-k})\Delta t] =$$

$$= \min_{\mathbf{U}_{M-k} \subset \Omega_{\mathbf{U}}} [S_{M-k+1}(\mathbf{X}^*_{M-k} + \mathbf{f}(\mathbf{X}^*_{M-k}, \mathbf{U}_{M-k})\Delta t) + f_0(\mathbf{X}^*_{M-k}, \mathbf{U}_{M-k})\Delta t]$$

където $k = 1,2,\ldots,M$.

При $i=0$ се получава оптималната стойност на (3) за цялата траектория:

$$S_0(\mathbf{X_0}) = \min_{\mathbf{U} \subset \Omega_{\mathbf{U}}} I(\mathbf{X_0}, \mathbf{U}) \quad .$$

Така по обратен път в края на I етап се получават $S_i = S_i(\mathbf{X}_i^*)$ и съответните условни оптимални управления $\mathbf{U}_i$, $i = 0,1,\dots,M-1$ при допускането, че е известна оптималната траектория.

II етап (прав):

През този етап, на базата на изчислените условни оптимални управления $\mathbf{U}_i$, $i = 0,1,\dots,M-1$, чрез уравнението на състоянието на обекта (2) и началните условия $\mathbf{X}_0^* = \mathbf{X}_0$ се определят действителните оптимални управления $\mathbf{U}_i^*$, $i = 0,1,\dots,M-1$ и оптималната траектория $\mathbf{X}^* = \mathbf{X}(\mathbf{U}^*)$. За тази цел се определя $\mathbf{U}_0^* = \mathbf{U}_0(\mathbf{X}_0^*)$ и след заместване в (2) се изчислява $\mathbf{X}_1^*$. Аналогично $\mathbf{U}_1^* = \mathbf{U}_1(\mathbf{X}_1^*)$ се замества в (2) за получаване на $\mathbf{X}_2^*$ и т.н.

5. Предимства и недостатъци на динамичното програмиране. Критерият на качеството може да се разглежда като сложна функция на много променливи.

$$I = I(\mathbf{X}_0, \mathbf{X}_1, ..., \mathbf{X}_M; \mathbf{U}_0, \mathbf{U}_1, ..., \mathbf{U}_{M-1})$$

- Главното преимущество на метода на динамичното програмиране се състои в това, че задачата за минимизация за сложна скаларна функция на няколко векторни аргумента се свежда до минимизация на поредица от скаларни функции само по един векторен аргумент.
- Аналитичното решение на тази задача е възможно в много редки и прости случаи, тъй като е свързано с решаване на системи нелинейни диференциални уравнения с частни производни. Затова, обикновено, динамичното програмиране се свързва с използуването на изчислителна техника.
- В дискретния вариант решението на задачата изисква съхранение на големи масиви от данни за оптималното управление във всяка точка от фазовото пространство и за всеки интервал от решението, както и данни за стойностите на функцията на Белман за два съседни интервала.

<u>*14. Динамично програмиране - дискретен вариант.*</u>

Необходимостта от голяма памет се увеличава чувствително с нарастване на реда на системата. Този проблем е известен като "*проклятие на размерността*". Затова понякога директното съхранение на необходимите стойности се заменя с въвеждане на правила или алгоритми за възпроизвеждането им, т.е. използуване на *апроксимация*.

- Тъй като се работи с дискретни стойности на управлението и на координатите на състоянието, то стойностите на функцията на Белман $S(\mathbf{X})$ за два съседни интервала и на условното оптимално управление $\mathbf{U}(\mathbf{X})$ за всичките интервали са известни и се помнят само в точките на фазовата мрежа. Като правило новите стойности на вектора $\mathbf{X}$, изчислени в (2), няма да попаднат точно в тези точки (на фазовата решетка), поради което се налага липсващите стойности на S и $\mathbf{U}$, необходими в I и II етапи, да бъдат определени чрез *интерполация* или понякога чрез *екстраполация*.

Динамично програмиране – изчислителни аспекти: