

# Szervomotor pályakövetést megvalósító irányítása

## 1. A gyakorlat célja

Szervomotor pozíciószabályozásának megoldása előírt pálya mentén. Időben optimális pálya és a pályakövetést megvalósító irányítási algoritmus implementálása.

## 2. Elméleti bevezető

A robotirányítási algoritmus felelős azért, hogy a robot végberendezése eljusson a célpontba, vagy előre definiált pályát járjon be. Amennyiben a robot csak PTP (ponttól pontig) mozgást végez, a robot irányítására egyszerű PID típusú irányítási algoritmusokat használhatunk. Pályakövetés esetében figyelembe kell, hogy vegyük a robot dinamikus modelljét is és így nemlineáris algoritmusokhoz jutunk.

Pályatervezés: a robotok térbeli mozgásának megtervezése, az előírt pálya meghatározása. Kétféle mozgást különböztetünk meg:

- Pont-pont mozgás (PTP – point to point)
- Pályakövetés

A PTP mozgás esetén a cél a robot eljuttatása egy adott végpontba. Fontos, hogy véges idő alatt tegye meg a robot az utat a kezdőponttól a végpontig. A két pont közti szakasz nem fontos.

Pályakövetés esetén a cél ugyanaz, eljutni egy kezdőpontból a végpontba, viszont itt a kezdőpont és a végpont között minden időpillanatban előírjuk a robot aktuális helyzetét. A robotirányítási algoritmusban az előírt érték minden időpillanatban más és más lesz, ami pontosabb irányításhoz vezet.

### 2.1 Polinomiális pályatervezés

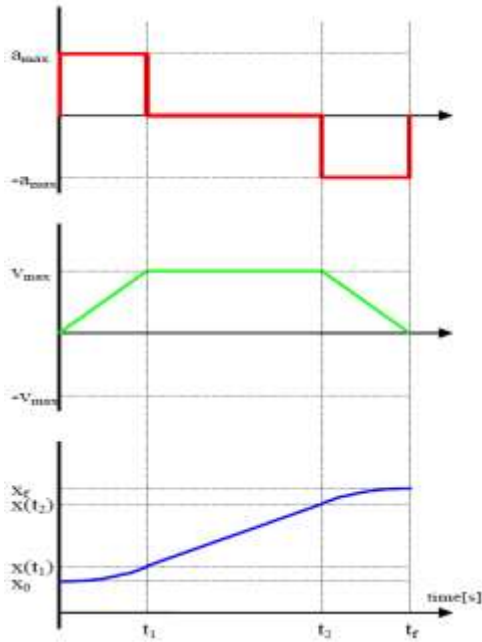
Adott  $x_0$  kezdeti és  $x_f$  végpozíció ismeretében tervezünk egy pályát ami, a mi rendszerünk paramétereinek (maximális gyorsulás  $a_{sys\_max}$ , maximális sebesség  $v_{sys\_max}$ ) megfelelően, időben optimális mozgást eredményez.

Az első szakaszban növeljük a sebességet (a gyorsulás  $a_{sys\_max}$ ) amíg elérjük a  $v_{sys\_max}$  sebességet. A második szakaszban a mozgást a maximális sebességgel folytatom. A harmadik szakasz hossza megegyezik az első szakaszéval. Itt a sebességet  $v_{sys\_max}$ -ról lecsökkentjük nullára (gyorsulás  $-a_{sys\_max}$ ).

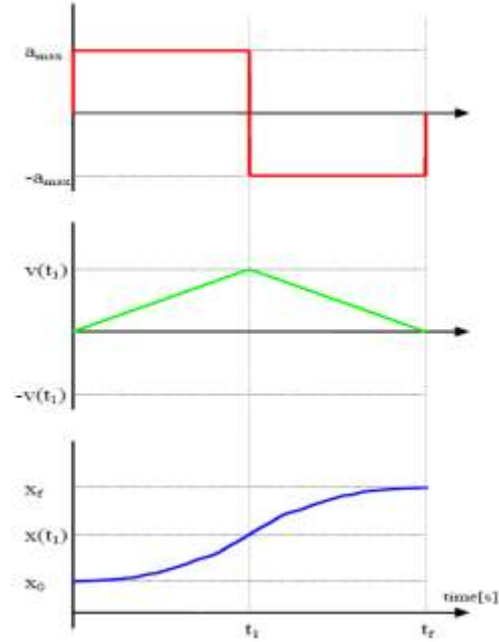
A megfelelő sebesség és gyorsulás irányok biztosítására bevezetjük a következő változókat:

$$\begin{aligned} a_{\max} &= \text{sign}(x_f - x_0) \cdot a_{\text{sys\_max}} \\ v_{\max} &= \text{sign}(x_f - x_0) \cdot v_{\text{sys\_max}} \end{aligned} \quad (1)$$

Az  $x_0, x_f$  közti távolság függvényében két illetve három szakaszból álló pályákat különböztetünk meg. Ha a távolság nem elég arra, hogy elérjük a maximális sebességet és utána a sebességet nullára csökkentjük, akkor csak két szakasza lesz a pályának (egy gyorsulási és egy lassulási szakasz).



1 Ábra: Három szakasz



2 Ábra: Két szakasz

Első eset (három szakasz):  $v_{\text{sys\_max}}^2 / a_{\text{sys\_max}} < |x_f - x_0|$

- első szakasz (gyorsulás):

$$\begin{cases} a(t) = a_{\max} \\ v(t) = a_{\max} \cdot t \\ x(t) = x_0 + \frac{a_{\max} \cdot t^2}{2} \end{cases} \quad (2)$$

ahol  $t_1 = v_{\max} / a_{\max}$ .

- második szakasz (mozgás maximális sebességgel):

$$\begin{cases} a(t) = 0 \\ v(t) = v_{\max} \\ x(t) = x(t_1) + v_{\max} \cdot t \end{cases} \quad (3)$$

ahol: 
$$t_2 = \frac{x(t_2) - x(t_1)}{v_{\max}} \quad (4)$$

$$x(t_1) = \frac{v_{\max}^2}{2 \cdot a_{\max}} + x_0 \quad (5)$$

$$x(t_2) = x_f - (x(t_1) - x_0) \quad (6)$$

- harmadik szakasz (lassítás):

$$\begin{cases} a(t) = -a_{\max} \\ v(t) = v_{\max} - a_{\max} \cdot t \\ x(t) = x(t_2) + v_{\max} \cdot t - \frac{a_{\max} \cdot t^2}{2} \end{cases} \quad (7)$$

*Második eset (két szakasz):*  $v_{\text{sys\_max}}^2 / a_{\text{sys\_max}} > |x_f - x_0|$

- első szakasz (gyorsulás):

$$\begin{cases} a(t) = a_{\max} \\ v(t) = a_{\max} \cdot t \\ x(t) = x_0 + \frac{a_{\max} \cdot t^2}{2} \end{cases} \quad (8)$$

ahol: 
$$t_1 = \sqrt{\frac{x_f - x_0}{a_{\max}}}.$$

- második szakasz (lassítás):

$$\begin{cases} a(t) = -a_{\max} \\ v(t) = v(t_1) - a_{\max} \cdot t \\ x(t) = x(t_1) + v(t_1) \cdot t - \frac{a_{\max} \cdot t^2}{2} \end{cases} \quad (9)$$

ahol:

$$x(t_1) = \frac{x_0 + x_f}{2} \quad (10)$$

$$v(t_1) = a_{\max} \cdot t_1 \quad (11)$$

## 2.2 A motor modellezése az irányítás tervezéséhez

Induljunk ki a motor mechanikus és elektromos egyenleteiből:

$$\begin{aligned} J \frac{dv}{dt} &= c_2 i \\ L \frac{di}{dt} + i \cdot R + c_1 v &= u \end{aligned} \quad (12)$$

ahol  $v$  a szögsebesség,  $i$  a rotoron folyó áram,  $u$  a bemeneti feszültség. A paraméterek  $J$  a terhelés inerciája,  $R$  a rotor ellenállása,  $L$  a rotor induktivitása,  $c_2$  a nyomatékállandó,  $c_1$  a sebességállandó.

Elosztva az ellenállás értékével és feltételezve, hogy az  $R/L$  elektromos időállandója kis értékű, kapjuk:

$$\begin{aligned} \frac{L}{R} \frac{di}{dt} + i + \frac{c_1}{R} v &= \frac{1}{R} u \\ \underbrace{\frac{L}{R}}_{\approx 0} \frac{di}{dt} + i + \frac{c_1}{R} v &= \frac{1}{R} u \\ i &= \frac{1}{R} u - \frac{c_1}{R} v \end{aligned} \quad (13)$$

Visszahelyettesítve a mechanikai egyenletbe, kapjuk az irányítás tervezéséhez alkalmazható modellt:

$$\begin{aligned} J \frac{dv}{dt} &= \frac{c_2}{R} u - \frac{c_1 c_2}{R} v \\ \frac{dv}{dt} &= \frac{c_2}{J \cdot R} u - \frac{c_1 c_2}{J \cdot R} v \end{aligned} \quad (14)$$

Vezessük be az alábbi jelöléseket:

$$K_u = \frac{c_2}{J \cdot R} \quad K_\omega = \frac{c_1 c_2}{J \cdot R} \quad (15)$$

A jelölésekkel kapjuk:

$$\frac{dv}{dt} = K_u \cdot u - K_\omega \cdot v \quad (16)$$

Ha  $x$ -el jelöljük a pozíciót:

$$\ddot{x} + K_\omega \dot{x} = K_u \cdot u \quad (17)$$

### 3. A mérés menete

#### 3.1 Az irányítási algoritmus tervezése

A pályakövetési feladat megoldásához feltételezzük, hogy adott az előírt pálya (előírt pozíció, sebesség, gyorsulás) valamint a motor (17) dinamikus egyenlete.

Definiáljuk a pályakövetési hibametrikát:

$$S = (v_{ref} - v) + \lambda(x_{ref} - x) \quad (18)$$

ahol  $\lambda > 0$  a súlyzótenyező.

Tervezzük úgy az irányítást, hogy  $S$  értéke nullába tartson. Határozzuk meg  $S$  időbeli változását:

$$\dot{S} = a_{ref} - \ddot{x} + \lambda(v_{ref} - v) \quad (19)$$

Behelyettesítve a (17) dinamikát:

$$\dot{S} = a_{ref} + \lambda(v_{ref} - v) + K_v v - K_u \cdot u \quad (20)$$

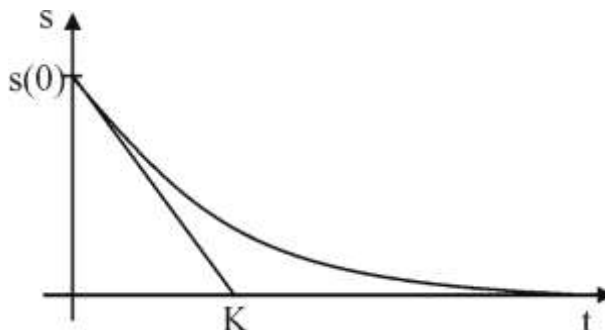
Válasszuk a beavatkozó jelet:

$$u = \frac{1}{K_u} (a_{ref} + K_v v + \lambda(v_{ref} - v) + K_s S) \quad (21)$$

A beavatkozó jellel kapjuk:

$$\dot{S} = -K_s S \quad (22)$$

Ha  $K_s > 0$ -ra választjuk, akkor a (22) egy stabil elsőfokú homogén differenciálegyenlet ( $\dot{S} + K_s S = 0$ ), tehát  $S$  értéke aszimptotikusan nullába fog konvergálni.



### 3 Ábra: Az S hibametrika konvergenciája

A folyamat paraméterek kiszámításához felhasználjuk a motor katalógusadatait:

$$\left\{ \begin{array}{l} R = 7.13 \\ c_1 = \frac{1}{26.16} \\ c_2 = 38.2 \cdot 10^{-3} \\ J = 1.89 \cdot 10^{-4} \end{array} \right. \Rightarrow K_u = \frac{38.2 \cdot 10^{-3}}{1.89 \cdot 10^{-4} \cdot 7.13} = \frac{382}{1.89 \cdot 7.13} = 28.35, \quad K_\omega = \frac{28.35}{26.16} = 1.08$$

A  $\lambda$  szabályozóparaméter megválasztásánál azt vesszük figyelembe, hogy (18) hibametrikában a pozícióhiba nagyobb súllyal kell megjelenjen, mint a sebességhiba. Mivel a sebességhiba súlyzója 1, ezért értékét egységénél nagyobboknak kell választani. Robotikai alkalmazásoknál tipikusan 10-nek választjuk.

### 3.2 A pályatervezés és a szabályozás implementálása

A feladat megoldásához felhasználjuk a pozíciószabályozásánál és a sebességszabályozásánál alkalmazott méréseket és a beavatkozó jel kiküldését. Szükségünk van a sebesség, pozíció és a mintavételi periódus mérésére. A sebesség mértékegysége *fok/mp* a pozíció mértékegysége *fok* a mintavételi periódus mértékegysége *mp* (*másodperc*) kell, hogy legyen.

A beavatkozó jel kiküldésénél szükségünk van a súrlódás miatt megjelenő holtáv kompenzálására, a beavatkozó jel korlátozására ( $\pm 5000$  mV) és a hardvernek specifikus kiküldésre: a beavatkozó jel abszolút értékét az analóg kimeneten, előjelét digitális kimeneten küldjük ki.

A pályatervezés és pályakövetést megvalósító szabályozás akkor kell végrehajtsdjon, amikor a *MotorControl* alkalmazásban a *TRACKING CONTROL* gomb van lenyomva. Természetesen ezt is az *OnTimer* függvényben implementáljuk.

A pályatervezéshez először az időalapot állítjuk elő egy statikus változóban. A program indításakor ennek a változónak az értéke nulla, és a szabályozás indításakor értékét minden mintavételi periódusban növeljük. A pályát számítása:

```
amax=5000;
umax=3000;
xf=3000;
t1=umax/amax;
x1=umax2/2amax;
x2=xf-x1;
t2=t1+(x2-x1)/umax;
t3=t1+t2;
t=t+TPeriod; // t-inicializálni 0-ra az OnInitDialog-ban
if (t<t1)
    aref=amax;
    uref=amax t;
    xref=amax/2 t2;
```

```

else if (t<t2)
    aref=0;
    uref=umax;
    xref=umax (t-t1)+x1;
else else if (t<t3)
    aref=amax;
    uref=umax-amax (t-t2);
    xref=xf-umax2/2amax +umax (t-t2)+amax (t-t2)2/2;
else aref=0;
    uref=0;
    xref=xf;

```

A beavatkozó jel kiszámításánál a  $K_S$  változó értékét a *KPSlider* csúszkáról olvassuk le. A beavatkozó feszültség számítása:

```

l=10;
Ku=28.35;
Kv=1.08;
getTextboxVal("KP_trackBar", out tempPID);
K_S = (double)tempPID;
s=(vref-v)+l(xref-x);
u=(aref+ l(vref-v)+Kv v+Ks s)/Ku;

```

A program futtatásánál vizsgáljuk a szabályozás minőségét  $K_S=20$  -ra majd más erősítésértékekre is. A TRACE segítségével vizsgáljuk a pályakövetési pontosságot.

#### **4. Kérdések és feladatok**

1. Módosítsuk úgy a program pályatervező részét, hogy  $x_0 \neq 0$ .
2. Bővítsük ki a programot úgy, hogy a pályatervező rész tartalmazza a két szakaszos pályát is.
3. Módosítsuk a mintavételi periódust 15 ezredmásodpercről 60 ezredmásodpercre a *SetTimer* függvényben. Milyen hatással van a pályakövetés pontosságára?