ROBOTIRÁNYÍTÁS

12. előadás

Alapjel miatti korrekció, állapotmegfigyelő, LQ szabályozás

Dr. habil. Kovács Levente
egyetemi docens

kovacs.levente@nik.uni-obuda.hu

Dr. Haidegger Tamás
egyetemi adjunktus

haidegger@irob.uni-obuda.hu

Az előadás témája és célja

A kurzus záró előadása az állapotvisszacsatolás „speciális eseteit” mutatja be. Alapjel miatti korrekciót akkor használunk, ha a szabályozás célja nem csupán az, hogy stabilizáljuk a rendszert, hanem az is, hogy egy zérustól különböző referencia jelet kövessen. Ha nem ismerjük a rendszer állapotváltozóit, akkor tervezhetünk állapotmegfigyelőt, ami meg tudja becsülni az állapotváltozók értékeit adott bemenetek és kimenetek mellett. Ebben az esetben nem is állapotvisszacsatolás történik, hanem a kimenet kerül visszacsatolásra. Végezetül ahhoz, hogy optimális állapotvisszacsatolást tervezzünk, használhatjuk az LQ szabályozást, ahol a két súlyozó mátrixot a szabályozási körrel szembeni elvárások alapján kell megválasztanunk.

Az előadás célja, hogy a hallgatók az állapotvisszacsatolás komplexebb megoldásait is elsajátítsák.

Kulcsszavak

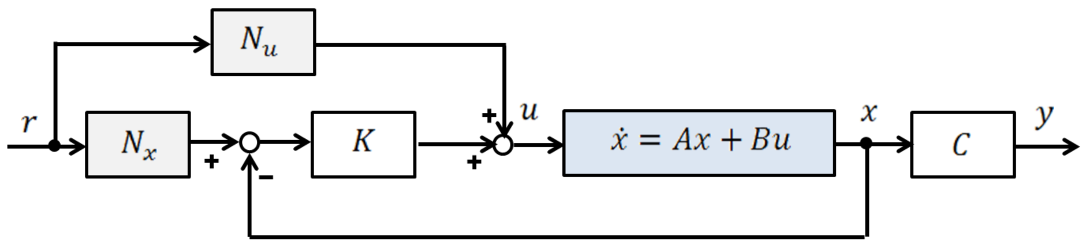
alapjel miatti korrekció, állapotmegfigyelő, LQ optimális szabályozás, költségfüggvény, súlyozómátrix, Algebraic Riccati Equation (CARE)

Tartalomjegyzék

1. Alapjel miatti korrekció

1.1. Alapjel miatti korrekció megvalósítása

Az előző előadásban úgy terveztünk állapotvisszacsatolást pólusáthelyezéssel, hogy a szabályozási körben nem szerepelt előírt alapjel, ami azt jelenti, hogy a szabályozás 0 alapjelet valósít meg. Ha a szabályozás célja nem csak az, hogy stabilizáljuk a rendszert, hanem az is, hogy egy nem nulla referencia jelet kövessen, akkor a blokk diagramot ki kell terjeszteni olyan speciális blokkokkal, amelyek a referencia jel követését lehetővé teszik (N_x és N_u). A kiegészített blokkdiagramot az 1. ábra mutatja. Mint általában, itt is feltesszük, hogy a szakasz D mátrixa nulla, így $y = Cx$.



1. ábra: Alapjel miatti korrekciót tartalmazó állapotvisszacsatolásos szabályozási kör

Amikor az N_x és N_u vektorokat számítjuk, akkor feltesszük, hogy *egységugrás bemenet esetén*

1. a kivonó tag után a jel értéke 0 állandósult állapotban, azaz

$$N_x r - x_\infty = 0 \Rightarrow x_\infty = N_x r \tag{1}$$

2. és a kimenet értéke 1 állandósult állapotban

$$y_\infty = r. \tag{2}$$

Felhasználva, hogy $y = Cx$, majd a (1) és a (2) egyenletet is felhasználva adódik, hogy

$$y_\infty = Cx_\infty = CN_x r = r. \tag{3}$$

Ebből adódóan

$$CN_x = I. \tag{4}$$

A blokkdiagramról leolvasva u állandósult állapotbeli értéke

$$u_\infty = N_u r. \tag{5}$$

Az állapotegyenlet állandósult állapotban tehát

$$\dot{x}_\infty = Ax_\infty + Bu_\infty = 0 \Rightarrow (AN_x + BN_u)r = 0 \Rightarrow AN_x + BN_u = 0. \tag{6}$$

A követelmények az alábbi egyenletekben összegezhetőek:

$$AN_x + BN_u = 0 \quad (7)$$

$$CN_x = I. \quad (8)$$

Mátrix alakban felírva a (7) és (8) egyenlet:

$$\begin{bmatrix} A & B \\ C & 0 \end{bmatrix} \begin{bmatrix} N_x \\ N_u \end{bmatrix} = \begin{bmatrix} 0 \\ 1 \end{bmatrix}. \quad (9)$$

A (9) egyenletet megoldva a keresett, N_x-et és N_u-t tartalmazó mátrixra:

$$\begin{bmatrix} N_x \\ N_u \end{bmatrix} = \begin{bmatrix} A & B \\ C & 0 \end{bmatrix}^{-1} \begin{bmatrix} 0 \\ 1 \end{bmatrix}. \quad (10)$$

1.2. Alapjel miatti korrekció – példa

Az előző előadás „Ackermann-formula – Matlab példa” mintarendszerét használva, a szakasz

```
A=[-0.3929 0.1786 0.1429; 0.5 0 0; 0 0.25 0];

B=[1;0;0];

C=[0 0 -0.7143];

D=0;
```

Matlabban N_x és N_u könnyen számítható

```
N=inv([A B; C D])*[0;0;0;1];

Nx=N(1:3)

Nx =

 0

 0

-1.4000

Nu = N(4)

Nu =

 0.2000
```

2. Kimeneti visszacsatolás állapotmegfigyelő használatával

2.1. Állapotmegfigyelő megvalósítása

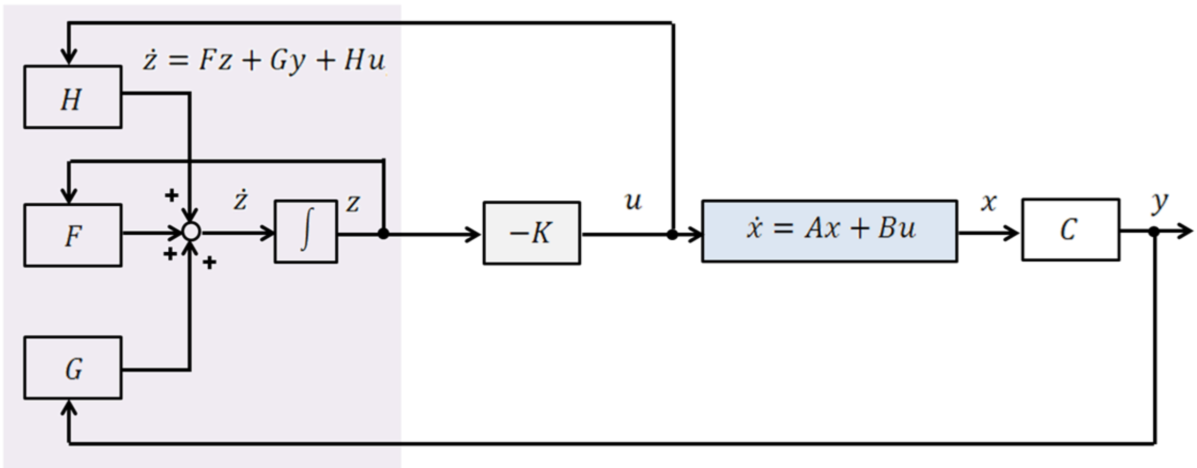
Egy erős hátránya az állapotvisszacsatolásnak, hogy az összes állapotváltozó értéke ismert kell, hogy legyen a visszacsatolás meghatározásához. Valós rendszerek esetén ez extrém drága (vagy lehetetlen). Hasznos lehet egy olyan tagot használni, amely

1. meg tudja becsülni az állapotváltozók értékeit adott bemenetek és kimenetek mellett,
2. biztosítani tudja, hogy megfelelően pontos modell esetén a becsült értékek a valósakhoz tartanak.

Az állapotmegfigyelő egy dinamikus rendszer, melynek állapotai a becsült állapotváltozók. Az állapotmegfigyelő állapotegyenlete (a szabályozó által megvalósítva):

$$\dot{z} = Fz + Gy + Hu \tag{11}$$

Az állapotmegfigyelőt tartalmazó szabályozási kör blokkdiagramját a 2. ábra mutatja. Figyeljük meg, hogy **ebben az esetben nem állapotvisszacsatolás van, hanem a kimenet van visszacsatolva**, mely jelet az állapotmegfigyelő G mátrixa kapja meg bemenetként.



2. ábra: Kimeneti visszacsatolást tartalmazó szabályozási kör állapotmegfigyelővel

Legyen a becslés hibája

$$\tilde{x} = x - z, \tag{12}$$

akkor a

$$H = B \tag{13}$$

választással élve a hiba tranziense az alábbi differenciál-egyenlettel írható le:

$$\dot{\tilde{x}} = F\tilde{x} = (A - GC)\tilde{x}, \tag{14}$$

ahol az F mátrix sajátértékei tetszőlegesen megválaszthatók egy megfelelő G választással élve, ha a (A, C) pár megfigyelhető. Az F mátrix sajátértékeinek elhelyezése algebrailag hasonló a pólusáthelyezés problémájához, tehát az **Ackermann formula** használható, azzal a különbséggel, hogy ebben az esetben az A és C mátrix *transzponáltjait* kell behelyettesíteni a képletbe.

2.2. Állapotmegfigyelő – példa

Folytatva a korábbi példát, ahol a zárt kör K erősítése az alábbi volt:

```
K = acker(A,B,[-2.5000 -1.2500 -0.7143])
```

Szeretnénk olyan tranzienst előírni az állapotmegfigyelőnek, amely gyorsabb, mint a zárt kör tranziense, így az állapotmegfigyelő sajátértékeit a bal félsíkon távolabb kell elhelyezni, mint a zárt körét

```
Gt = acker(A',C',[-5.0000 -2.5000 -1.4286])

Gt =

-120.9577 -111.7226 -11.9500

G = Gt'

G =

-120.9577

-111.7226

-11.9500
```

Miután kiszámítottuk az állapotmegfigyelő G mátrixának az értékét az Ackermann képlettel, az F mátrix is meghatározható:

```
F=A-G*C

F =

 -0.3929    0.1786  -86.2555

  0.5000         0  -79.8018

       0    0.2500   -8.5357
```

Végezetül pedig az állapotmegfigyelő H mátrixát úgy választjuk meg, hogy a szabályozandó rendszer B mátrixával legyen egyenlő:

```
H=B
H =
     1
     0
     0
```

3. LQ optimális szabályozás

A pólusáthelyezéses állapotvisszacsatolás esetben a zárt kör pólusait a tervező írja elő. Azonban semmi nem garantálja, hogy az így tervezett szabályozási kör optimális lesz. Ahhoz, hogy optimális állapotvisszacsatolást tudjunk tervezni, az úgynevezett LQ (Linear-quadratic) szabályozást (vagy más néven LQR, Linear-quadratic regulator) tudjuk használni.

3.1. Az LQ szabályozási probléma

A szabályozandó rendszer egy LTI állapotteres (state space) modell:

$$\begin{aligned} \dot{x} &= Ax + Bu \\ y &= Cx \end{aligned} \tag{15}$$

A szabályozás célja állapotvisszacsatolás tervezése a rendszer stabilizálására, tehát a szabályozó jel

$$u = -Kx, \tag{16}$$

ahol a K állapotvisszacsatolási vektor megtervezése egy kompromisszum a bementi és kimeneti előírások teljesítésére.

Az optimális szabályozás megvalósításához **költségfüggvény**t definiálunk:

$$J(x,u) = \frac{1}{2}\int_0^\infty [x^T(t)Qx(t) + u^T(t)Ru(t)]dt, \tag{17}$$

ahol Q egy $n\,x\,n$ szimmetrikus pozitív definit mátrix ($Q^T = Q, Q > 0$), R pedig egy $m\,x\,m$ szimmetrikus pozitív definit mátrix ($P^T = P, P > 0$). A Q és R mátrixokat **súlyozómátrix**nak hívjuk. Ezek függvényében **azt az $\boldsymbol{u}$ szabályozó jelet keressük, ami minimalizálja a $\boldsymbol{J}$ költségfüggvényt**.

3.2. Az LQ szabályozási probléma megoldása

A megoldás levezetését mellőzve, az a K állapotvisszacsatolási vektor, amely használatával megtaláljuk azt az u szabályozó jelet, ami minimalizálja a J költségfüggvényt:

$$K = R^{-1}B^TP, \tag{18}$$

ahol B a szabályozandó rendszerhez tartozó bemeneti mátrix, R az egyik súlyozómátrix, P pedig az úgynevezett folytonos idejű **Control Algebraic Riccati Equation (CARE)** megoldásából származik.

$$P \to PA + A^TP - PBR^{-1}B^TP + Q = 0 \tag{19}$$

3.3. Az LQ szabályozási probléma megoldása Matlabbal

Matlabban az alábbi parancs használható a probléma megoldására:

```
[K,S,e] = lqr(SYS,Q,R,N)
```

Az `lqr` parancs az optimális `K` mátrixot számítja, úgy, hogy a (15) egyenletben felírt alakú rendszerre a (16) egyenletben található szabályozó jel minimalizálja a (17) egyenletben található költségfüggvényt. A visszacsatoló `K` állapotvisszacsatolási mátrix mellett az `lqr` még a Ricatti egyenlet `S` megoldását is megadja, valamint a zárt rendszer pólusait (`e`) is.

```
e = eig(A-B*K)
```

Amikor `N` nincs megadva, akkor `N = 0` (általában ezt használjuk).

3.4. Az LQ szabályozás tulajdonságai

Statikus visszacsatolás

Az LQ módszer egy statikus K visszacsatoló mátrixot ad meg, ami nem egy dinamikus rendszer, így a zárt rendszer fokszáma ugyanaz marad, ami a szakaszé volt.

Robusztusság

Az LQ végtelen amplitúdótartalékot biztosít:

$$Gm_{LQR} = \infty. \tag{20}$$

Az LQ a fázis tartalékot is garantálja:

$$\varphi_t > 60°. \tag{21}$$

3.5. A Q és R súlyozó mátrix tipikus megválasztása

Míg a pólusáthelyezéses állapotvisszacsatolásnál a zárt kör pólusait, addig **az LQ optimális szabályozásnál a $\boldsymbol{Q}$ és az $\boldsymbol{R}$ súlyozó mátrixokat kell nekünk megválasztani**. Az alábbiakban ismertetünk egy tipikus megválasztást, de hangsúlyozzuk, hogy *ez csak egy lehetséges megválasztás, de nem minden esetben így választjuk meg a súlyozó mátrixokat. A megválasztás az elvégzendő feladat specifikációjától függ!*

A Q mátrix

$$Q = \begin{bmatrix} q_1 & & & \\ & q_2 & & \\ & & \ddots & \\ & & & q_n \end{bmatrix} \tag{22}$$

egy tipikus megválasztása:

$$q_i = \frac{1}{t_{s,i}(x_{i,max})^2}, \tag{23}$$

ahol $t_{s,i}$ az előírt beállási ideje az x_i állapotváltozónak, $x_{i,max}$ pedig az $|x_i|$ megszorítása (limitje).

Hasonlóképpen az alábbi alakú R mátrix

$$R = \rho \begin{bmatrix} r_1 & & & \\ & r_2 & & \\ & & \ddots & \\ & & & r_m \end{bmatrix} \tag{24}$$

egy tipikus megválasztása:

$$r_i = \frac{1}{(u_{i,max})^2}, \tag{25}$$

ahol $u_{i,max}$ pedig az $|u_i|$ megszorítása (limitje).

A $\boldsymbol{Q}$ súlyozó mátrix az állapotváltozókra hat, a rendszer minőségi jellemzőit súlyozza, míg az $\boldsymbol{R}$ súlyozó mátrix a szabályozó jelre hat, a bemenetet bünteti. Ezt a tervezésnél figyelembe kell venni.

3.6. LQ optimális szabályozás – példa

Feladat

Tervezzünk LQ szabályozót a $W(s)$ szakaszhoz.

$$W(s) = \frac{5}{(7s+1)(4s+1)(2s+1)} \tag{26}$$

A súlyozó mátrixokat a (22)-(25) egyenletek szerint válasszuk meg, az alábbi paraméterekkel:

$$\begin{aligned} x_{1,max} &= 0.8 \\ x_{2,max} &= 0.4 \\ x_{3,max} &= 0.1 \\ t_{s,1} &= t_{s,2} = t_{s,3} = 12 \\ u_{max} &= 0.6 \\ \rho &= 1 \end{aligned} \tag{27}$$

Megoldás Matlabban

1. Először adjuk meg a szakaszt átviteli függvény alakjában.

```
Aplant=5; T1=7; T2=4; T3=2;

num=Aplant;

den=conv(conv([T1 1],[T2 1]),[T3 1]);

sys_tf=tf(num,den)
```

2. Térjünk át állapotteres leírásba és mentsük ki az állapotér mátrixait.

```
sys_ss=ss(sys_tf)

A=sys_ss.a;

B=sys_ss.b;

C=sys_ss.c;

D=sys_ss.d;
```

3. Paraméterek deklarálása, Q és R súlyozó mátrixok megadása.

```
% adott értékek

x_1max = 0.8;

x_2max = 0.4;

x_3max = 0.1;
```

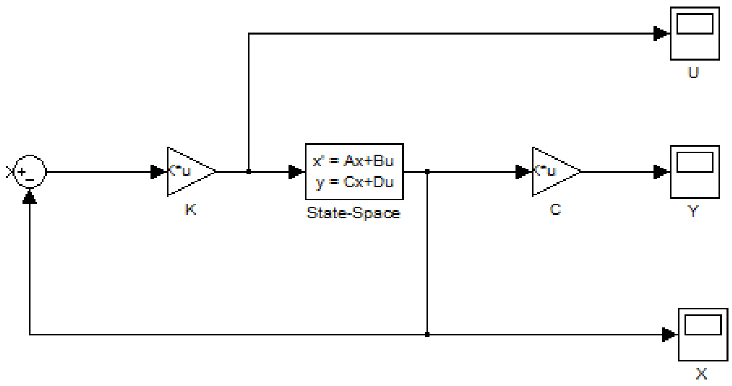
```
t_s1 = 12;
t_s2 = 12;
t_s3 = 12;
u_max = 0.6;
% rho
rho = 1;
% Q elemeinek számítása
q1 = 1 / (t_s1 * x_1max^2);
q2 = 1 / (t_s2 * x_2max^2);
q3 = 1 / (t_s3 * x_3max^2);
% Q definiálása
Q = [q1 0 0
    0 q2 0
    0 0 q3]
% R elemeinek számítása
r = 1 / (u_max^2); % egy bemenetünk van
% R definiálása
R  = rho * r
```

4. K állapotvisszacsatolási mátrix számítása.

```
[K_LQ, P, eig_cl] = lqr(A, B, Q, R, 0);
% K_LQ a visszacsatoló mátrix
% P a Riccati egyenlet megoldása
% eig_cl tartalmazza a zárt kör sajátértékeit
((A-BK) sajátértékei)
```

5. Megvalósítás Simulinkben

A Simulinkes blokkdiagramot a 3. ábra mutatja.

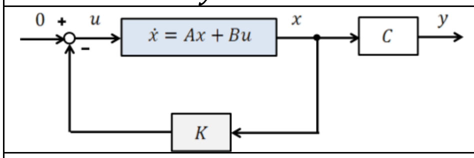
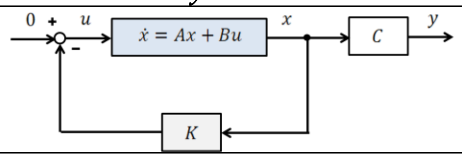


3. ábra: LQ optimális állapotvisszacsatolás megvalósítása Simulinkben

- A szabályozó a Matlabban `lqr` paranccsal számított K.
- A C mátrix értékét a `state-space` blokkban egységmátrixra állítjuk, hogy a blokk kimenetén az állapotváltozókat kapjuk meg.
- Az állapotokból ezután egy külön C mátrix-szorzással határozzuk meg a kimenetet. Ez a C mátrix eredetileg a rendszer része!

4. A pólusáthelyezés és az LQ szabályozás összehasonlítása

Végezetül hasonlítsuk össze a pólusáthelyezéses és az LQ optimális állapotvisszacsatolást. Az összehasonlítást az 1. táblázat foglalja össze.

Pólusáthelyezéses állapotvisszacsatolás		LQ optimális állapotvisszacsatolás
állapotteres modell $\dot{x} = Ax + Bu$ $y = Cx$	szakasz	állapotteres modell $\dot{x} = Ax + Bu$ $y = Cx$
	blokk diagram	
$u = -Kx$	szabályozó jel	$u = -Kx$
Ackermann-formula $K = e_n{}^T \cdot M_C^{-1} \cdot \varphi_c(A)$	megoldás	$K = R^{-1}B^TP$ Control Algebraic Riccati Equation $P \rightarrow PA + A^TP - PBR^{-1}B^TP + Q = 0$
a zárt rendszer pólusai s_i	megválasz-tandó paraméterek	súlyozó mátrixok Q, R

1. táblázat: A pólusáthelyezés és az LQ szabályozás összehasonlítása

A szabályozandó szakasz mind a két esetben állapottérben van megadva. A blokkdiagram pontosan ugyanaz mind a két esetben, hiszen állapotvisszacsatolás mind a kettő, a szabályozó jelre is ugyanez igaz. A különbség abban van, hogy a K állapotvisszacsatolási vektort, azaz a szabályozási probléma megoldását hogyan számítjuk. Pólusáthelyezéses állapotvisszacsatolás esetén az Ackermann-formula segítségével, míg LQ optimális állapotvisszacsatolásnál a Control Algebraic Riccati Equation (CARE) megoldásából számolt P mátrix segítségével tehetjük meg. A szabályozás megvalósításához pólusáthelyezés esetén a zárt kör pólusait kell nekünk megválasztani, míg LQ optimális szabályozás esetén a Q és az R súlyozómátrixot.

Az előadás összefoglalása

Az állapotvisszacsatolás alapmodellje az igényeknek megfelelően kibővíthető. Alapjel miatti korrekciót esetén a blokk diagramot ki kell terjeszteni olyan speciális blokkokkal, amelyek a referencia jel követését lehetővé teszik (N_x és N_u). Ha nem ismerjük a rendszer állapotváltozóit, akkor tervezhetünk állapotmegfigyelőt. Az állapotmegfigyelő egy dinamikus rendszer, melynek állapotai a becsült állapotváltozók, állapotegyenlete az F, G és H mátrixokat tartalmazza. Az F mátrix sajátértékeinek elhelyezése algebrailag hasonló a pólusáthelyezés problémájához, így az Ackermann formula használható (azzal a különbséggel, hogy ebben az esetben az A és C mátrix transzponáltjait kell behelyettesíteni a képletbe).

LQ szabályozás esetén a K állapotvisszacsatolási vektor megtervezése egy kompromisszum a bementi és kimeneti előírások teljesítésére. Azt az u szabályozó jelet keressük ilyenkor, ami minimalizálja a megadott J költségfüggvényt. LQ optimális állapotvisszacsatolásnál a K vektor a Control Algebraic Riccati Equation (CARE) megoldásából számolt P mátrix segítségével állítható elő.

Ellenőrző kérdések

1. Állapotvisszacsatolás nem nulla alapjel esetén milyen új blokkokat kell figyelembe venni, ezeket hogy tudjuk számítani?

2. Miért használunk állapotmegfigyelőt?

3. Mi az állapotmegfigyelő állapotegyenlete és az együtthatómátrixok megválasztása?

4. LQ szabályozás esetén mik azok a paraméterek, amiket nekünk kell megválasztani, mi az értelmezési tartományuk és mire hatnak?

5. LQ szabályozás esetén hogy számítjuk a K állapotvisszacsatolási vektort?