ROBOTIRÁNYÍTÁS

11. előadás

Pólusáthelyezés állapotvisszacsatolással

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

Az előző előadásokon soros kompenzátorokat terveztünk, azaz a szabályozó a szabályozási körben a szakasz előtt helyezkedett el, a szabályozó kimenete a szakasz bemenete volt, a szakasz kimenete pedig negatívan vissza volt csatolva. Jelen előadásban azonban olyan szabályozást tervezünk, ahol nem a kimenet, hanem a rendszer állapotváltozói vannak visszacsatolva egy K vektoron keresztül. Ehhez azonban a rendszert állapotteres leírásban kell vizsgálnunk, emellett a rendszer irányíthatóságát is ellenőriznünk kell.

Az előadás célja, hogy a hallgatók megismerkednek a pólusáthelyezéses állapotvisszacsatolás tervezésével, amit egy papíron levezetett számolós példa mellett egy Matlab és Simulink segítségével megoldott példa is segít.

Kulcsszavak

hasonlósági transzformáció, kanonikus alak, pólusáthelyezés, állapotvisszacsatolás, irányíthatóság, megfigyelhetőség, Ackermann-formula

Tartalomjegyzék

1. Állapotegyenlet és kanonikus alakjai

1.1. Áttérés állapotteres leírásból átviteli függvényre

A 4. előadáson megismerkedtünk a különböző modellreprezentációkkal: időtartományban az állapotteres leírást használtuk, Laplace tartományban pedig az átviteli függvényt (és a pólus-zérus-erősítés alakot).

A 8. előadástól szabályozótervezéssel foglalkoztunk (empirikus szabályozások; P, PI és PID soros kompenzátorok), és ha visszaemlékszünk, a szabályozandó szakaszunkat mindig átviteli függvény formájában (lásd (1) egyenlet) adtuk meg. Ennek az az oka, hogy a Laplace transzformáció rendelkezik egy nagy előnnyel, miszerint a differenciálegyenleteket algebrai egyenletté alakítja (s operátor tartományban). Soros kompenzátorok tervezésénél ezt az előnyt ki tudjuk használni, hiszen a szabályozandó szakasz kimenetének ismeretére van csak szükségünk, amit negatívan visszacsatolunk.

$$W(s) = \frac{Y(s)}{U(s)} = \frac{b_m s^m + b_{m-1} s^{m-1} + \cdots + b_1 s + b_0}{a_n s^n + a_{n-1} s^{n-1} + \cdots + a_1 s + a_0} \tag{1}$$

Más a helyzet viszont, amikor a rendszer „belsejében” lévő jelek ismeretére is szükségünk van, azaz a rendszer állapotváltozóit (x) is ismerni és használni szeretnénk. Ebben az esetben az állapotteres leírást kell használnunk (lásd (2) és (3) egyenlet). Az állapotegyenlet (a (2) egyenlet) az x állapotváltozó időbeli változását írja le, míg a kimeneti egyenlet (a (3) egyenlet) az y kimenet (szabályozott jellemző) adott t időpillanatbeli értékét adja meg.

$$\frac{dx(t)}{dt} = Ax(t) + Bu(t) \tag{2}$$

$$y(t) = Cx(t) + Dx(t) \tag{3}$$

A kapcsolat az állapotteres leírás és az átviteli függvény között az, hogy az A mátrix sajátértékei (állapotteres leírás) a rendszer pólusai (átviteli függvény).

A két reprezentáció között áttérés lehetséges mind a két irányban, azonban a megfeleltetés csak az állapotteres leírásról az átviteli függvényre egyértelmű, ennek egyenlete:

$$W(s) = \frac{Y(s)}{U(s)} = C(sI - A)^{-1} B(s) + D \tag{4}$$

Amikor átviteli függvényről állapottérbe térünk át, akkor az így kapott leírás több formában is előállhat.

Matlabban az áttérésekre a `tf2ss()`, illetve az `ss2tf()` függvények használhatók.

1.2. Hasonlósági transzformáció

Ahogy említettük, egyetlen rendszernek több állapotteres leírása létezik; ezek ugyanazt a rendszert írják le, hasonlósági transzformációval átalakíthatók egymásba.

A *hasonlósági transzformáció* (vagy *állapotteres transzformáció*) egy T invertálható transzformációs mátrixszal definiált:

$$z = Tx \Rightarrow x = T^{-1}z \tag{5}$$

A transzformáció következtében egy új állapotváltozó jön létre, a z állapotváltozó. Az állapotegyenlet a z állapotváltozóval felírva:

$$\dot{z} = TAT^{-1}z + TBu \tag{6}$$

$$y = CT^{-1}x + Du \tag{7}$$

Azt, hogy ez a (6) és (7) egyenlettel reprezentált rendszer valóban ugyanaz a rendszer, mint amit az x állapotváltozóval (lásd (2) és (3) egyenlet) írtunk le, az garantálja, hogy egy mátrix sajátértékei invariánsak a hasonlósági transzformációra. Tehát A és TAT^{-1} sajátértékei ugyanazok, és egyben az átviteli függvény pólusai is.

1.3. Kanonikus alakok

Az állapotegyenlet megfelelő *kanonikus alak*ja lehetővé teszi az átviteli függvény nevezőjében a pólusok/ együtthatók közvetlen leolvasását.

A Matlab meg tudja határozni a transzformációs mátrixokat, amelyek ahhoz kellenek, hogy az adott állapotteres leírás kanonikus formára alakítható legyen, valamint megadja magát a transzformációt is. A továbbiakban röviden ismertetjük azon Matlab által kínált függvényeket, melyeknek segítségével a rendszer egy kanonikus alakjához juthatunk.

Állapotteres transzformáció adott T transzformációs mátrixszal (`ss2ss`)

Matlab szintaxis: `sysT = ss2ss(sys,T).`

Az `ss2ss` függvény megadja a transzformált állapotteres modellt (`sysT`) adott `sys` rendszer és `T` transzformáció mellett (a `sys` állapotteres leírású kell, hogy legyen, a `T` mátrix pedig invertálható).

Modális/blokkdiagonális kanonikus alak (`canon modal` típusa)

Matlab szintaxis: `[csys,T]= canon(sys,'modal').`

Nem csak a kanonikus alakot (`csys`), de a használt `T` transzformációt is megadja. A kapott alakban A egy blokkdiagonális mátrix, a blokk mérete tipikusan

- 1x1-es valós sajátértékek esetén
- 2x2-es komplex sajátértékek esetén
- ha többszörös sajátértékek vannak, akkor a blokkok nagyobbak lehetnek.

Például ha a rendszer pólusai (az A mátrix sajátértékei):

$$p_1 = \lambda_1, p_2 = \sigma + j\omega, p_3 = \sigma - j\omega, p_4 = \lambda_2, \tag{8}$$

akkor az A mátrix az alábbi blokkdiagonális alakú:

$$A = \begin{bmatrix} \lambda_1 & 0 & 0 & 0 \\ 0 & \sigma & \omega & 0 \\ 0 & -\omega & \sigma & 0 \\ 0 & 0 & 0 & \lambda_2 \end{bmatrix} \tag{9}$$

Megfigyelhetőségi kanonikus alak (`canon companion` típusa)

Matlab szintaxis: `[csys,T]= canon(sys,'companion').`

Nem csak a kanonikus alakot (`csys`), de a használt `T` transzformációt is megadja. A rendszer karakterisztikus egyenlete explicit módon megjelenik az A mátrix jobb szélső oszlopában

Például ha a rendszer karakterisztikus polinomja:

$$\Delta(s) = a_n s^n + a_{n-1} s^{n-1} + \cdots + a_1 s + a_0, \tag{10}$$

akkor az A mátrix az alábbi alakú:

$$A = \begin{bmatrix} 0 & 0 & \cdots & \cdots & 0 & -a_0 \\ 1 & 0 & 0 & \cdots & 0 & -a_1 \\ 0 & 1 & 0 & \cdots & \cdots & \cdots \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ 0 & \cdots & \cdots & 1 & 0 & -a_{n-1} \\ 0 & \cdots & \cdots & 0 & 1 & -a_n \end{bmatrix} \tag{11}$$

Irányíthatósági kanonikus alak (`ss(tf)`)

Matlab szintaxis: `sys=ss(tf(n,d)).`

Hasonló algoritmust használ, mint a `tf2ss`, de átskálázza az állapot-vektort annak érdekében, hogy összenyomja az A állapotmátrix numerikus tartományát, és hogy a numerikus számításokat javítsa.

2. Pólusáthelyezés állapotvisszacsatolással

2.1. Állapotvisszacsatolási modell

Állapotvisszacsatolási modell tervezésénél feltesszük, hogy az állapotváltozók elérhetők a szabályozó számára minden időpillanatban. A soros kompenzátorokkal szemben itt nem a kimenet (y), hanem az állapotváltozó (x) van negatívan visszacsatolva (1. ábra). Ahhoz viszont, hogy az állapotváltozóhoz hozzáférjünk, a rendszert ketté kell bontani, először csak a (2) egyenlettel leírt rendszer található a körben; ennek kimeneteként férünk hozzá az állapotváltozóhoz. Ezután van sorba kapcsolva a C mátrix, aminek a kimenete már a $Cx = y$ (ez pontosan az állapotteres leírás második, kimeneti egyenlete – a (3) egyenlet – csak a D mátrix nélkül, ami valós rendszerekben nem szerepel).

A szabályozó jel az állapotváltozó lineáris kombinációjaként áll elő:

$$u = -Kx, \tag{12}$$

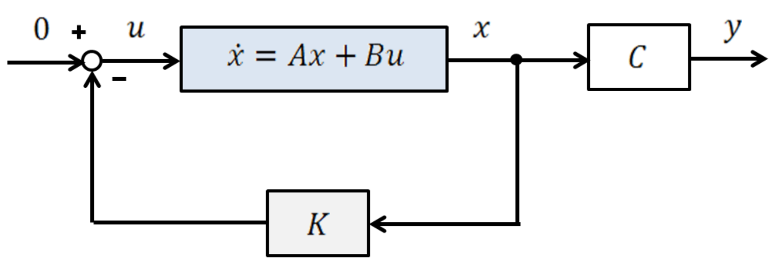
ahol K egy n elemű sorvektor ($n = dim\, x$), ha a rendszer egyetlen bemenetű. Vegyük észre, hogy a visszacsatolásban nem egyetlen skalár érték van, hanem n skalár értékű időfüggvény egy vektorba foglalva (az összes állapotváltozó visszacsatolása).

A zárt kör állapotegyenlete állapotvisszacsatolás esetén (behelyettesítve a (12) egyenletet a (2) egyenletbe):

$$\dot{x} = (A - BK)x, \tag{13}$$

ami azt jelenti, hogy a zárt kör pólusai az $A - BK$ mátrix sajátértékei.

A szabályozási cél, hogy a szakasz pólusai (az A mátrix sajátértékei) tetszőleges helyre áthelyezhetők legyenek a komplex számsíkon. Ez megoldható egy megfelelően megválasztott K vektor segítségével, ha az (A, B) pár (állapot)irányítható.



1. ábra: Állapotvisszacsatolási modell

Külön felhívnánk a figyelmet arra, hogy azt, hogy a szakasz pólusait hova helyezzük át (hova írjuk elő a zárt rendszer pólusait), a **tervező dönti el**.

2.2. Irányíthatóság, megfigyelhetőség

2.2.1. Irányíthatóság (controllability)

A teljes állapotirányíthatóság megadja, hogy egy külső bement képes a rendszer belső állapotát átvinni tetszőleges kezdeti állapotból tetszőleges végső állapotba véges idő alatt.

Az irányíthatósági mátrix:

$$M_c = [B \quad AB \quad \cdots \quad A^{n-1}B] \tag{14}$$

Az irányíthatóság feltétele: az M_c irányíthatósági mátrix teljes rangú kell, hogy legyen, azaz

$$\mathrm{rank}(M_c) = n \tag{15}$$

2.2.2. Megfigyelhetőség (observability)

Egy rendszer megfigyelhető, ha minden lehetséges állapotvektor és szabályozó vektor érték mellett az aktuális állapot előállítható véges idő alatt csupán a kimenetek használatával. Ez azt is jelenti, hogy csupán a kimenetekből a teljes rendszer viselkedése meghatározható.

A megfigyelhetőségi mátrix:

$$M_o = \begin{bmatrix} C \\ CA \\ \vdots \\ CA^{n-1} \end{bmatrix} \tag{16}$$

A megfigyelhetőség feltétele: az M_o megfigyelhetőségi mátrix teljes rangú kell, hogy legyen, azaz

$$\mathrm{rank}(M_o) = n \tag{17}$$

2.3. Pólusáthelyezés állapotvisszacsatolással

Alapprobléma. Adott a rendszer állapotteres alakban. A rendszer pólusai vagy instabilak, vagy stabilak, de lassúak.

Cél. A cél az, hogy az általunk előírt értékekre gyorsítsuk a rendszer pólusait. A gyorsítás azt jelenti, hogy a valós tengely mentén a $-\infty$ irányába mozgatjuk el a pólusokat.

Például az eredetileg a rendszer pólusai legyenek $s_1 = 0.5, s_2 = -1.5$. A feladat, hogy gyorsítsuk kétszeres értékükre a pólusokat (ha kell, stabilizáljuk is). Ennek eredményeképpen a zárt kör pólusai: $s_{a1} = -1, s_{a2} = -3$. A kérdés az, hogy ezt milyen K állapotvisszacsatolási vektorral érhetjük ezt el?

Megoldás. A K állapotvisszacsatolási vektor az *Ackermann-formula* segítségével számítható.

2.4. Ackermann-formula

Ha a zárt kör előírt sajátértékeinek vektora

$$[s_{c1} \quad s_{c2} \quad \cdots \quad s_{cn}], \tag{18}$$

akkor az $A - BK$ mátrix karakterisztikus egyenlete

$$\varphi_c(s) = \prod_{i=1}^{n}(s - s_{ci}) \tag{19}$$

és a hozzátartozó állapotvisszacsatolási K vektor

$$K = e_n{}^T \cdot M_C^{-1} \cdot \varphi_c(A), \tag{20}$$

ahol e_n az n. egységvektor, M_c pedig az irányíthatósági mátrix. A (20) egyenletet hívjuk **Ackermann-formulának, ami tehát a pólusáthelyezéses állapotvisszacsatolási probléma megoldását adja meg**.

Matlabban a `K = acker(A,B,p)` parancs használható, ahol `A` és `B` a rendszer mátrixai, `p` az előírt pólusok vektora, `K` pedig a keresett állapotvisszacsatolási vektor.

2.4.1. Ackermann-formula – számolós példa

Feladat

Adott az alábbi lineáris rendszerünk:

$$\dot{x} = \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix} x + \begin{bmatrix} 1 \\ -1 \end{bmatrix} u \tag{21}$$

Azt szeretnénk, hogy az állapotvisszacsatolás eredményeképp a zárt rendszer pólusai a következők legyenek:

$$s_{c1,c2} = -2 \pm 2j \tag{22}$$

Milyen K állapotvisszacsatolási vektort kell használnunk ehhez?

Megoldás

1. Probléma megfogalmazása, a jelenlegi rendszer pólusainak számítása

A rendszer pólusai az A mátrix sajátértékei, azaz a karakterisztikus egyenlet gyökei:

$$\varphi(s) = \det(sI - A) = 0 \tag{23}$$

Az egységmátrix (I) dimenziója megegyezik az A mátrixéval (jelen esetben 2x2-es):

$$I = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \tag{24}$$

Az $sI - A$ számítása:

$$sI = \begin{bmatrix} s & 0 \\ 0 & s \end{bmatrix}, A = \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix} \Rightarrow sI - A = \begin{bmatrix} s & 0 \\ 0 & s \end{bmatrix} - \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix} = \begin{bmatrix} s & -1 \\ 1 & s \end{bmatrix} \tag{25}$$

Felhasználva, hogy a 2x2-es mátrix determinánsának számítási szabálya

$$A = \begin{bmatrix} a & b \\ c & d \end{bmatrix} \Rightarrow \det(A) = ad - bc, \tag{26}$$

a karakterisztikus polinomra az alábbi adódik:

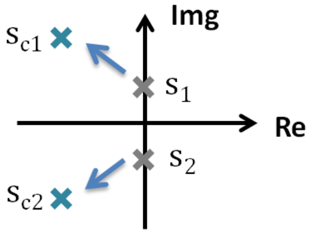
$$\varphi(s) = \det(sI - A) = s \cdot s - (-1) \cdot 1 = s^2 + 1 \tag{27}$$

A karakterisztikus egyenlet megoldása, a rendszer pólusai tehát:

$$\begin{gathered} \varphi(s) = s^2 + 1 = 0 \Rightarrow s^2 = -1 \Rightarrow \; s = \sqrt{-1} = j \Rightarrow \\ s_1 = +j, s_2 = -j \end{gathered} \tag{28}$$

2. $\boldsymbol{\varphi_c(A)}$ számítása

Azt szeretnénk tehát, hogyha a jelenlegi pólusok ($s_1 = +j, s_2 = -j$) a zárt körben gyorsulnának ($s_{c1,c2} = -2 \pm 2j$), lásd 2. ábrát.



2. ábra: Pólusok gyorsítása

A zárt kör karakterisztikus egyenlete állapotvisszacsatolás esetén (definíció szerint):

$$\varphi_c(s) = \det\big(sI - (A - BK)\big) = 0 \tag{29}$$

Ezt most a gyöktényezős alakból tudjuk számítani:

$$\varphi_c(s) = (s - s_{c1}) \cdot (s - s_{c2}) \tag{30}$$

A zárt kör karakterisztikus egyenletének számítása a gyöktényezős alakból:

$$\begin{aligned} \varphi_c(s) &= [s - (-2 + 2j)] \cdot [s - (-2 - 2j)] = \\ &= [s + (2 - 2j)] \cdot [s + (2 + 2j)] = \\ &= s^2 + s(2 + 2j) + s(2 - 2j) + (2 - 2j)(2 + 2j) \end{aligned} \tag{31}$$

Felhasználva a komplex számok összeadására vonatkozó

$$(a, b) + (c, d) = (a + c, b + d) \tag{32}$$

szabályt, a (31) egyenlet elsőfokú tagja:

$$s(2+2j)+s(2-2j)=(4s,0)=4s \tag{33}$$

Ugyancsak felhasználva a komplex számok szorzására vonatkozó

$$(a,b)\cdot(c,d)=(ac-bd,bc+ad) \tag{34}$$

szabályt, a (31) egyenlet konstans tagja:

$$(2-2j)(2+2j)=(8{,}0)=8 \tag{35}$$

A zárt kör karakterisztikus egyenlete tehát:

$$\varphi_c(s)=s^2+4s+8 \tag{36}$$

A zárt kör karakterisztikus egyenletébe most behelyettesítjük az A mátrixot (azaz az $s=A$ helyettesítéssel élünk):

$$\varphi_c(A)=A^2+4\cdot A+8\cdot I \tag{37}$$

Felhasználva, hogy

$$A^2=\begin{bmatrix}0 & 1\\ -1 & 0\end{bmatrix}\cdot\begin{bmatrix}0 & 1\\ -1 & 0\end{bmatrix}=\begin{bmatrix}-1 & 0\\ 0 & -1\end{bmatrix} \tag{38}$$

a (37) egyenlet megoldása:

$$\varphi_c(A)=\begin{bmatrix}-1 & 0\\ 0 & -1\end{bmatrix}+\begin{bmatrix}0 & 4\\ -4 & 0\end{bmatrix}+\begin{bmatrix}8 & 0\\ 0 & 8\end{bmatrix}=\begin{bmatrix}7 & 4\\ -4 & 7\end{bmatrix} \tag{39}$$

3. $\boldsymbol{M_c}^{-1}$ számítása

Tudjuk, hogy az irányíthatósági mátrix a (14) egyenletben megadott alakú. Mivel esetünkben az állapotváltozók száma $n=2\Rightarrow n-1=1$, ezért

$$M_c=[B\quad AB] \tag{40}$$

Felhasználva, hogy

$$AB=\begin{bmatrix}0 & 1\\ -1 & 0\end{bmatrix}\begin{bmatrix}1\\ -1\end{bmatrix}=\begin{bmatrix}-1\\ -1\end{bmatrix} \tag{41}$$

a (40) egyenlet megoldása:

$$M_c=\begin{bmatrix}1 & -1\\ -1 & -1\end{bmatrix} \tag{42}$$

Az irányíthatósági mátrix inverzének számításánál felhasználjuk, hogy egy 2x2-es mátrix inverze az alábbiak szerint számítható:

$$A^{-1} = \begin{bmatrix} a & b \\ c & d \end{bmatrix}^{-1} = \frac{1}{ad - bc}\begin{bmatrix} d & -b \\ -c & a \end{bmatrix} \tag{43}$$

Tehát az irányíthatósági mátrix inverze:

$$M_c{}^{-1} = \begin{bmatrix} 1 & -1 \\ -1 & -1 \end{bmatrix}^{-1} = \frac{1}{-2}\begin{bmatrix} -1 & 1 \\ 1 & 1 \end{bmatrix} = \begin{bmatrix} \frac{1}{2} & -\frac{1}{2} \\ -\frac{1}{2} & -\frac{1}{2} \end{bmatrix} \tag{44}$$

4. $e_n{}^T$ számítása

Mivel $n = 2$, ezért

$$e_n = \begin{bmatrix} 0 \\ 1 \end{bmatrix} \Rightarrow e_n{}^T = \begin{bmatrix} 0 & 1 \end{bmatrix} \tag{45}$$

5. K számítása

A K állapotvisszacsatolási vektort az Ackermann-formula segítségével (lásd (20) egyenlet) kapjuk meg:

$$K = \begin{bmatrix} 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} \frac{1}{2} & -\frac{1}{2} \\ -\frac{1}{2} & -\frac{1}{2} \end{bmatrix} \cdot \begin{bmatrix} 7 & 4 \\ -4 & 7 \end{bmatrix} = \begin{bmatrix} 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} 5.5 & -1.5 \\ -1.5 & -5.5 \end{bmatrix} = \begin{bmatrix} -1.5 & -5.5 \end{bmatrix} \tag{46}$$

2.4.2. Ackermann-formula – Matlab példa

1. A szakaszt állapotteres leírásban adjuk meg, tehát az A, B, C és D mátrixokkal.

```
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;
```

2. A szakasz pólusai számíthatók az állapotmátrix sajátértékeinek számításával:

```
eig(A)

-0.5000

 0.2500

-0.1429
```

3. Ha a szakasz instabil (vagy stabil, csak lassúak a pólusok), akkor az elsődleges célunk az, hogy stabil zárt kört állítsunk elő egy megfelelően megválasztott állapotvisszacsatolás segítségével (esetleg gyorsítsuk a stabil, de lassú pólusokat).

4. Először nézzük meg, hogy a szakasz irányítható-e:

```
Mc=ctrb(A,B);
rank(Mc)
ans = 3
```

Ha az irányíthatósági mátrix rangja megegyezik az állapotváltozók számával, akkor a szakasz irányítható.

5. Írjuk elő a kívánt értékeket a zárt rendszer pólusai számára.

```
p = [-2.5000 -1.2500 -0.7143];
```

6. Számítsuk ki az ehhez szükséges K állapotvisszacsatolási vektort:

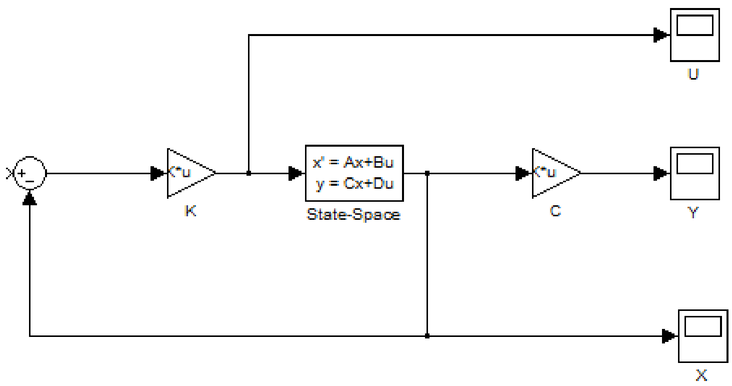
```
K = acker(A,B,p)
K = 4.0714   11.7857   18.0000
```

7. Az eredmény ellenőrizhető a zárt kör pólusainak kiszámításával:

```
eig(A-B*K)
-2.5000
-1.2500
-0.7143
```

8. Megvalósítás Simulinkben

A Simulinkes blokkdiagramot a 3. ábra mutatja.



3. ábra: Állapotvisszacsatolás megvalósítása Simulinkben

- A szabályozó a Matlabban `acker` 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!

Az előadás összefoglalása

A modelljeinket reprezentálhatjuk időtartományban és Laplace tartományban is. Időtartományban az állapotteres leírást használtuk, Laplace tartományban pedig az átviteli függvényt. A két reprezentáció közötti áttérés lehetséges mind a két irányban, de csak az állapotteres leírásról az átviteli függvényre történő áttérés egyértelmű. Abban az esetben, amikor átviteli függvényről állapottérbe lépünk át, a rendszerünket végtelen sok, egymásba hasonlósági transzformációval átalakítható állapot-és kimeneti egyenlettel írhatjuk le. A sok állapotteres leírás közül kiemelkednek az úgynevezett kanonikus alakok, melyekből az átviteli függvény nevezőjének gyökei (a pólusok) vagy a karakterisztikus polinom együtthatói közvetlenül leolvashatók. Matlabban olyan kanonikus alakot is előállíthatunk, amelyben az A állapotmátrix numerikus tartományát redukáljuk a numerikus számítások javítsa céljából.

Pólusáthelyezéses állapotvisszacsatolás esetén a rendszerünk állapotteres alakban adott. A rendszer pólusai vagy instabilak, vagy stabilak, de lassúak. A szabályozási cél az, hogy az általunk előírt értékekre gyorsítsuk a rendszer pólusait, így stabilizálva és gyorsítva a zárt kört. Ezt úgy tudjuk elérni, hogy az állapotváltozókat negatívan visszacsatoljuk egy K állapotvisszacsatolási vektor segítségével. Arra, hogy ezt a K vektort hogyan lehet kiszámítani, az Ackermann-formula ad választ.

Ellenőrző kérdések

1. Mi a kanonikus alak?

2. Mi az állapotvisszacsatolás lényege, hogy néz ki az állapotvisszacsatolás blokk diagramja?

3. Mi az irányíthatóság és a megfigyelhetőség feltétele?

4. Pólusáthelyezéses állapotvisszacsatolás esetén mik azok a paraméterek, amiket nekünk kell megválasztani?

5. Pólusáthelyezéses állapotvisszacsatolás esetén hogy számítjuk a K állapotvisszacsatolási vektort?