# ROBOTIRÁNYÍTÁS

11. előadás

Pólusáthelyezés állapotvisszacsatolással

Dr. habil. Kovács Levente
egyetemi docens

Dr. Haidegger Tamás
egyetemi adjunktus

# Tartalom

# 1. Állapotegyenlet és kanonikus alakjai

# Áttérés állapotteres leírásból átviteli függvényre

## Átviteli függvény reprezentáció

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

## Állapotteres reprezentáció

$$\dot{x} = Ax + Bu$$
$$y = Cx + Du$$

Áttérés állapotteres leírásból átviteli függvényre (Laplace transzformációval)

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

Kezdeti állapot: $x(0) = 0$

MATLAB

# Hasonlósági transzformáció

## Hasonlósági transzformáció (vagy állapotteres transzformáció)

- a transzformáció egy T invertálható transzformációs mátrixszal definiált

$$z = Tx$$
$$x = T^{-1}z$$

- z az új állapotváltozó
- az állapotegyenlet a z állapotváltozóval felírva

$$\dot{z} = TAT^{-1}z + TBu$$
$$y = CT^{-1}x + Du$$

✓ egy mátrix sajátértékei invariánsak a hasonlósági transzformációra

➡ A és $TAT^{-1}$ sajátértékei ugyanazok, és egyben az átviteli függvény pólusai is

# Kanonikus alakok

## Kanonikus alakok

- Az állapotegyenlet megfelelő kanonikus alakja lehetővé teszi az átviteli függvény nevezőjében a pólusok/ együtthatók közvetlen leolvasását.

- Matlab
  - ✓ meg tudja határozni a transzformációs mátrixokat, amelyek ahhoz kellenek, hogy az adott állapotteres leírás kanonikus formára alakítható legyen
  - ✓ és megadja magát a transzformációt

- `ss2ss, canon, ss(tf(.))`

# Kanonikus alakok

## Állapotteres transzformáció adott T transzformációs mátrixszal

**ss2ss**

- szintaxis: `sysT = ss2ss(sys,T)`
- 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
- `ss2ss` használható folytonos és diszkrét modellek esetén is

- példa

```
[A,B,C,D] =tf2ss(num,den)
T = balance(A)
sysT = ss2ss(sys,inv(T))
```

megadja a hasonlósági transzformációt

# Kanonikus alakok

## `canon`

- szintaxis: `[csys,T]= canon(sys,type)`

- a `sys` lineáris modellt transzformálja a `csys` kanonikus állapotteres modellé
- szintén megadja a `T` állapot-koordináta transzformációt, amely megfelelteti a `sys` állapotteres modell állapotit a `csys` állapotteres modell állapotainak
- a `type` argumentum specifikálja, hogy `csys` modal vagy companion alakú
- `sys`: tetszőleges dinamikus lineáris modell

# Kanonikus alakok

## Modális/blokkdiagonális kanonikus alak

```
[csys,T]= canon(sys,'modal')
```

- modal alakban `A` egy blokkdiagonális mátrix (→ `csys.A`)
- a blokk mérete tipikusan
  - ✓ 1x1-es valós sajátértékek esetén
  - ✓ 2x2-es komplex sajátértékek esetén
  - ✓ ha többszörös sajátértékek vannak, akkor a blokkok nagyobbak lehetnek
- például, egy rendszernek, melynek sajátértékei:

$$p_1 = \lambda_1, p_2 = \sigma + j\omega, p_3 = \sigma - j\omega, p_4 = \lambda_2$$

akkor a modal `A` mátrix az alábbi alakú:

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

# Kanonikus alakok

## Megfigyelhetőségi kanonikus alak

```
[csys,T]= canon(sys,'companion')
```

• a companion realizációban a rendszer karakterisztikus egyenlete explicit módon megjelenik az A mátrix jobb szélső oszlopában

• ha egy rendszer karakterisztikus polinomja

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

• akkor a hozzá tartozó companion alakú A mátrix

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

*Megjegyzés*

• a companion transzformáció megköveteli, hogy a rendszer irányítható legyen az első bementről

• hátránya, hogy rosszul kondicionált a legtöbb állapotteres számításra, így ha lehetséges, használata kerülendő

# Kanonikus alakok

## Irányíthatósági kanonikus alak

**`ss(tf(.))`**

- a `tf`-ről `ss`-re történő áttérésnél a `ss(sys_tf)` megadja az irányíthatósági kanonikus alakot
- 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
- példa

```
n=[1 1]; d=[1 1 10];
[A,B,C,D]=tf2ss(n,d);
A
 A =
-1 -10
 1   0
```

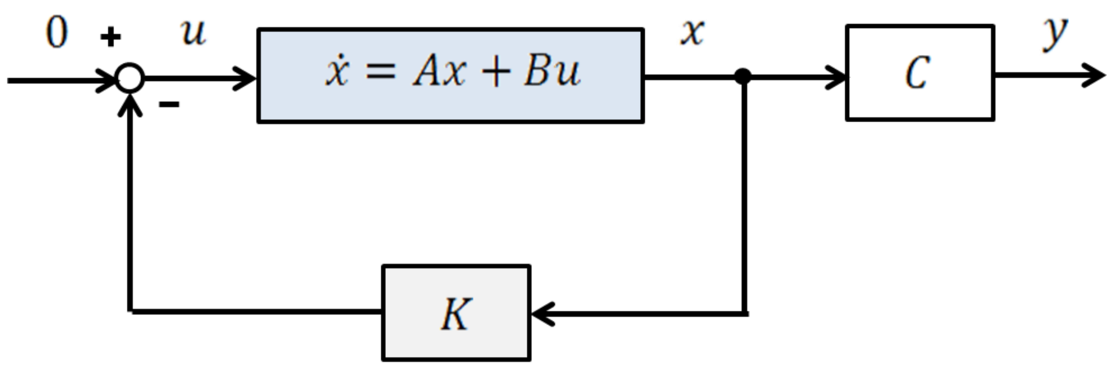
```
n=[1 1]; d=[1 1 10];
sys=ss(tf(n,d));
sys.a
ans =
-1 -5
 2  0
```

# 2. Pólusáthelyezés állapotvisszacsatolással

# Állapotvisszacsatolási modell

- feltesszük, hogy az az állapotváltozók elérhetők a szabályozó számára minden időpillanatban
- a szabályozó jel az állapotváltozó lineáris kombinációjaként áll elő:

  $u = -Kx$ , ahol K egy n elemű sorvektor (n = dim x), ha a rendszer egyetlen bemenetű
- így megkapjuk az állapotvisszacsatolási modellt:




- vegyük észre, hogy a visszacsatolásban nem egyetlen skalár érték van, hanem n skalár értékű időfüggvény egy vektorba foglalva (az összes állapotváltozó visszacsatolása).

# A szabályozási cél

- a zárt kör állapotegyenlete állapotvisszacsatolás esetén:

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

ami azt jelenti, hogy a zárt kör pólusai az $\mathrm{A - BK}$ mátrix sajátértékei

A szabályozási cél

- a szakasz pólusai (az $\mathrm{A}$ mátrix sajátértékei) tetszőleges helyre áthelyezhetők legyenek a komplex számsíkon
- ez megoldható egy megfelelően megválasztott $\mathrm{K}$ vektor segítségével, ha az $(\mathrm{A,B})$ pár (állapot)irányítható

# Irányíthatóság

$$\dot{x} = Ax + Bu$$

$$y = Cx + Du$$

- Egy rendszer állapotváltozója – ami a rendszer változóinak értékét tartalmazza – teljesen leírja a rendszert minden adott pillanatban.
- A teljes állapotirányíthatóság megadja, hogy egy külső bement képes a rendszer belső állapotát átvinni tetszőleges kezdeti állapotból tetszőleges végső állapotba véges idő alatt.

Az irányíthatóság feltétele: az $M_c=[B\ AB\ \ldots\ A^{n-1}B]$ irányíthatósági mátrix teljes rangú kell, hogy legyen

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

# Megfigyelhetőség

$$\dot{x} = Ax + Bu$$

$$y = Cx + Du$$

- 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 megfigyelhetőség feltétele: az $M_o=[C\ CA\ \ldots\ CA^{n-1}]^T$ megfigyelhetőségi mátrix teljes rangú kell, hogy legyen

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

# Hasznos Matlab függvények

- `ctrb(sys)` – az irányíthatósági mátrixát számítja egy állapotteres rendszernek. Egy nxn-es A mátrix és egy nxm-es B mátrix esetén a `ctrb(A,B)` parancs a $Co = \begin{bmatrix} B & AB & A^2B & \dots & A^{n-1}B \end{bmatrix}$ irányíthatósági mátrixot adja meg, ahol `Co` n soros és nm oszlopos
- `obsv(sys)` – a megfigyelhetőségi mátrixát számítja egy állapotteres rendszernek. Egy nxn-es A mátrix és egy nxp-s B mátrix esetén az `obsv(A,C)` megfigyelhetőségi mátrixot adja meg n oszloppal és np sorral

$$Ob = \begin{bmatrix} C \\ CA \\ CA^2 \\ \vdots \\ CA^{n-1} \end{bmatrix}$$

- `rank(A)` – az `A` mátrix rangját adja meg (azaz egy becslést a mátrix lineárisan független sorainak/oszlopainak számáról)

# 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
  - ✓ gyorsítás: a valós tengely mentén a -∞ irányába mozgatjuk el a pólusokat
  - ✓ példa
    - eredetileg a rendszer pólusai: $s_1 = 0.5, s_2 = -1.5$
    - gyorsítsuk kétszeres értékükre a pólusokat (ha kell, stabilizáljuk is):

    $s_{a1} = -1, s_{a2} = -3$
- kérdés: milyen $\mathrm{K}$ állapotvisszacsatolási vektorral érhetjük ezt el?

## Megoldás

- a $\mathrm{K}$ állapotvisszacsatolási vektor az Ackermann-formula segítségével számítható

# Ackermann-formula

- ha a zárt kör előírt sajátértékeinek vektora

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

- akkor az $\mathrm{A} - \mathrm{BK}$ mátrix karakterisztikus egyenlete

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

- és a hozzátartozó állapotvisszacsatolási $\mathrm{K}$ vektor

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

Ackermann-formula

MATLAB **`acker`** Pole placement design for single-input systems

`K = acker(A,B,p)` Ha adott a SISO állapotteres rendszer ($\dot{x} = Ax + bu$ ) és az előírt pólusok `p` vektora, az `acker(A,B,p)` parancs az Ackermann formulát használja a `K` vektor számítására, ahol az $\mathrm{u} = -\mathrm{Kx}$ Állapotvisszacsatolás a zárt rendszer pólusait az előírt `p` helyekre teszi.

# Ackermann-formula – számolós példa

## Feladat

- adott az alábbi lineáris rendszerünk: $\dot{x} = \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix} x + \begin{bmatrix} 1 \\ -1 \end{bmatrix} u$
- azt szeretnénk, hogy az állapotvisszacsatolás eredményeképp a zárt rendszer pólusai a következők legyenek: $s_{c1,c2} = -2 \pm 2j$
- milyen K állapotvisszacsatolási vektort kell használnunk ehhez?

## Megoldás

① Probléma megfogalmazása, a jelenlegi rendszer pólusainak számítása

② $\varphi_c(A)$ számítása

③ $M_c^{-1}$ számítása

④ $e_n^T$ számítása

⑤ K számítása

# Ackermann-formula – számolós példa

## ① Probléma megfogalmazása, a jelenlegi rendszer pólusainak számítása

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

$$\varphi(s) = \det(sI - A) = 0$$

✓ az egységmátrix dimenziója megegyezik az A mátrixéval (jelen esetben 2x2-es) $I = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$

✓ $sI = \begin{bmatrix} s & 0 \\ 0 & s \end{bmatrix} \quad A = \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix} \quad sI - A = \begin{bmatrix} s & 0 \\ 0 & s \end{bmatrix} - \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix} = \begin{array}{c} \begin{array}{cc} a & b \end{array} \\ \begin{bmatrix} s & -1 \\ 1 & s \end{bmatrix} \\ \begin{array}{cc} c & d \end{array} \end{array}$

✓ 2x2-es mátrix determinánsának számítása

$$\varphi(s) = \det(sI - A) = ad - bc = s \cdot s - (-1) \cdot 1 = s^2 + 1$$

✓ a karakterisztikus egyenlet megoldása, a rendszer pólusai:

$$\varphi(s) = s^2 + 1 = 0 \Rightarrow s^2 = -1 \Rightarrow$$

$$s = \sqrt{-1} = j \Rightarrow s_1 = +j, s_2 = -j$$

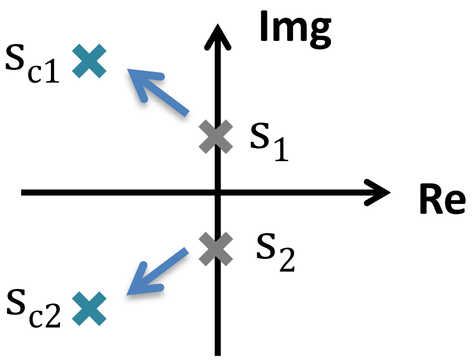
# Ackermann-formula – számolós példa

## ② $\varphi_c(A)$ számítása

• azt szeretnénk tehát, hogyha a jelenlegi pólusok a zárt körben gyorsulnának

$$s_1 = +j, s_2 = -j$$

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




• a zárt kör karakterisztikus egyenlete állapotvisszacsatolás esetén (definíció szerint):

$$\varphi_c(s) = \det\big(sI - (A - BK)\big) = 0$$

• ezt most a gyöktényezős alakból tudjuk számítani:

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

# Ackermann-formula – számolós példa

## ② $\varphi_c(A)$ számítása

- a zárt kör karakterisztikus egyenletének számítása a gyöktényezős alakból:

$$\varphi_c(s) = [s - (-2 + 2j)] \cdot [s - (-2 - 2j)] =$$
$$= [s + (2 - 2j)] \cdot [s + (2 + 2j)] =$$
$$= s^2 + s(2 + 2j) + s(2 - 2j) + (2 - 2j)(2 + 2j)$$

a) komplex számok összeadása $(a, b) + (c, d) = (a + c, b + d)$

$$s(2 + 2j) + s(2 - 2j) = (4s, 0) = 4s$$

b) komplex számok szorzása $(a, b) \cdot (c, d) = (ac - bd, bc + ad)$

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

$$\Rightarrow \quad \varphi_c(s) = s^2 + 4s + 8$$

# Ackermann-formula – számolós példa

## ② $\varphi_c(A)$ számítása

- a zárt kör karakterisztikus egyenletébe most behelyettesítjük az $A$ mátrixot:

$$\varphi_c(s) = s^2 + 4s + 8$$

$$s = A$$

$$\varphi_c(A) = A^2 + 4 \cdot A + 8 \cdot I$$

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

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

# Ackermann-formula – számolós példa

## ③ $M_c^{-1}$ számítása

- irányíthatósági mátrix számítása általános esetben:

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

- az irányíthatósági mátrix esetünkben:

$$n = 2 \Rightarrow n - 1 = 1$$

$$M_c = [B \quad AB]$$

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

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

# Ackermann-formula – számolós példa

## ③ $M_c^{-1}$ számítása

- irányíthatósági mátrix inverzének számítása
  - ✓ 2x2-es mátrix inverzének számítása

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

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

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

# Ackermann-formula – számolós példa

## ④ $e_n{}^T$ számítása

$$n = 2 \Rightarrow e_n = \begin{bmatrix} 0 \\ 1 \end{bmatrix} \Rightarrow e_n{}^T = \begin{bmatrix} 0 & 1 \end{bmatrix}$$

## ⑤ K számítása

- a K állapotvisszacsatolási vektort az Ackermann-formula segítségével kapjuk meg

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

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

$$= \begin{bmatrix} 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} 5.5 & -1.5 \\ -1.5 & -5.5 \end{bmatrix} = \begin{bmatrix} -1.5 & -5.5 \end{bmatrix}$$

# Ackermann-formula – Matlab példa

- a szakasz állapotteres alakban adott

```
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;
```

- az állapotmátrix sajátértékei számíthatók

```
eig(A)
ans =
-0.5000
 0.2500
-0.1429
```

➡ a szakasz instabil

➡ az elsődleges célunk az, hogy stabil zárt kört állítsunk elő egy megfelelően megválasztott állapotvisszacsatolás segítségével

# Ackermann-formula – Matlab példa

- először nézzük meg, hogy a szakasz irányítható-e

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

→ az irányíthatósági mátrix rangja megegyezik az állapotváltozók számával

→ a szakasz irányítható

- írjuk elő az alábbi értékeket a zárt rendszer pólusai számára:

$$s_{c1} = -2.5000, s_{c2} = -1.2500, s_{c3} = -0.7143$$

- az $u = -Kx$ állapotvisszacsatolást megvalósító $K$ vektor, mely az előírt pólusokat állítja elő zárt körben, az `acker` paranccsal számítható

```
K = acker(A,B,[-2.5000 -1.2500 -0.7143])
K =
4.0714   11.7857   18.0000
```

# Ackermann-formula – Matlab példa

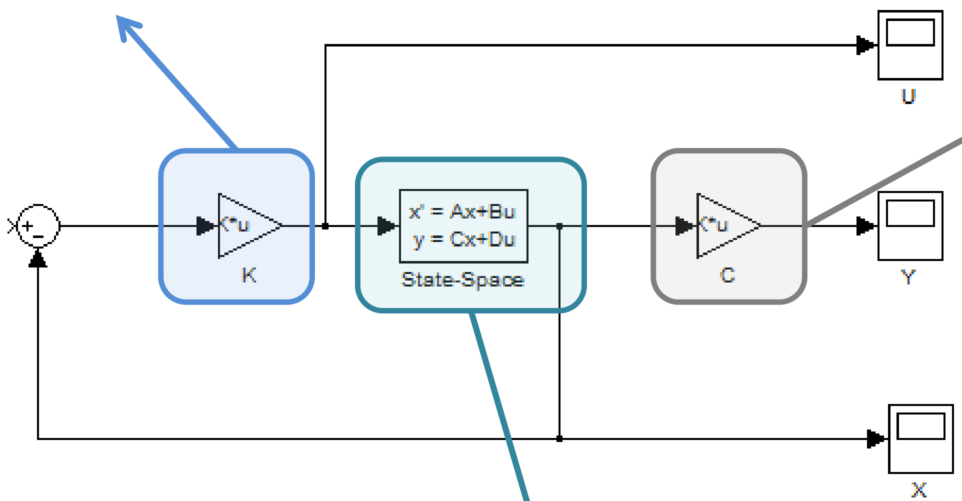
- az eredmény ellenőrizhető a zárt kör pólusainak kiszámításával

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

➡ nagy pontossággal át tudtuk helyezni a pólusokat a kívánt helyre

# Pólusáthelyezés állapotvisszacsatolással Simulinkben

A szabályozó a Matlabban `acker` paranccsal számított K.

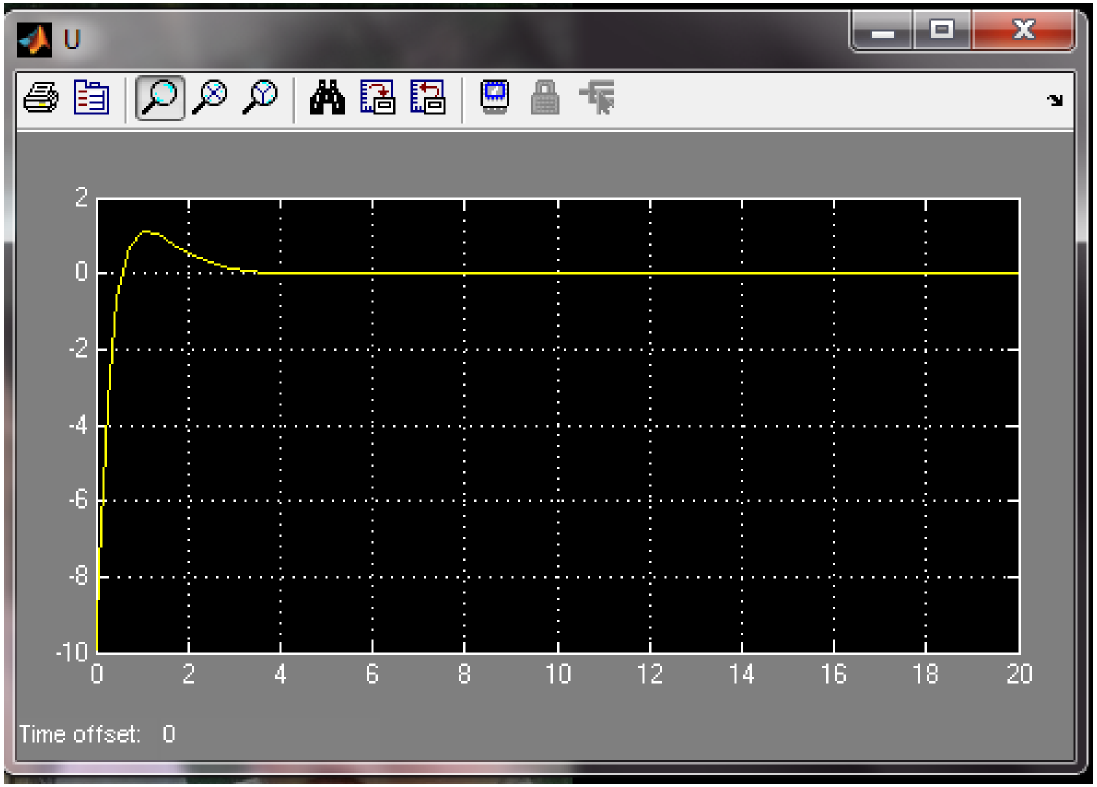



Az állapotokból ezután egy külön C mátrix-szorzással határozzuk meg a kimenetet. Ez a C mátrix eredetileg a rendszer része!

A C mátrix értékét a state-space blokkban egységmátrixra állítjuk, hogy a blokk kimenetén az állapotváltozókat kapjuk meg.

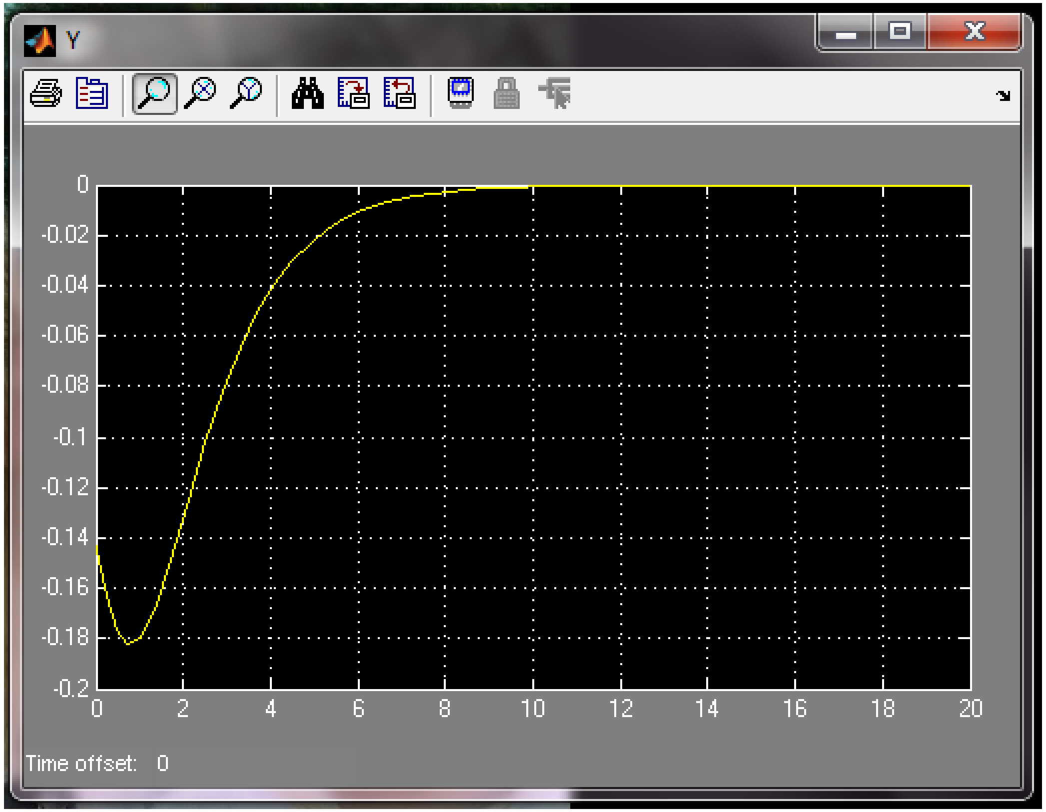
# Pólusáthelyezés állapotvisszacsatolással Simulinkben

A bemenet időfüggvénye



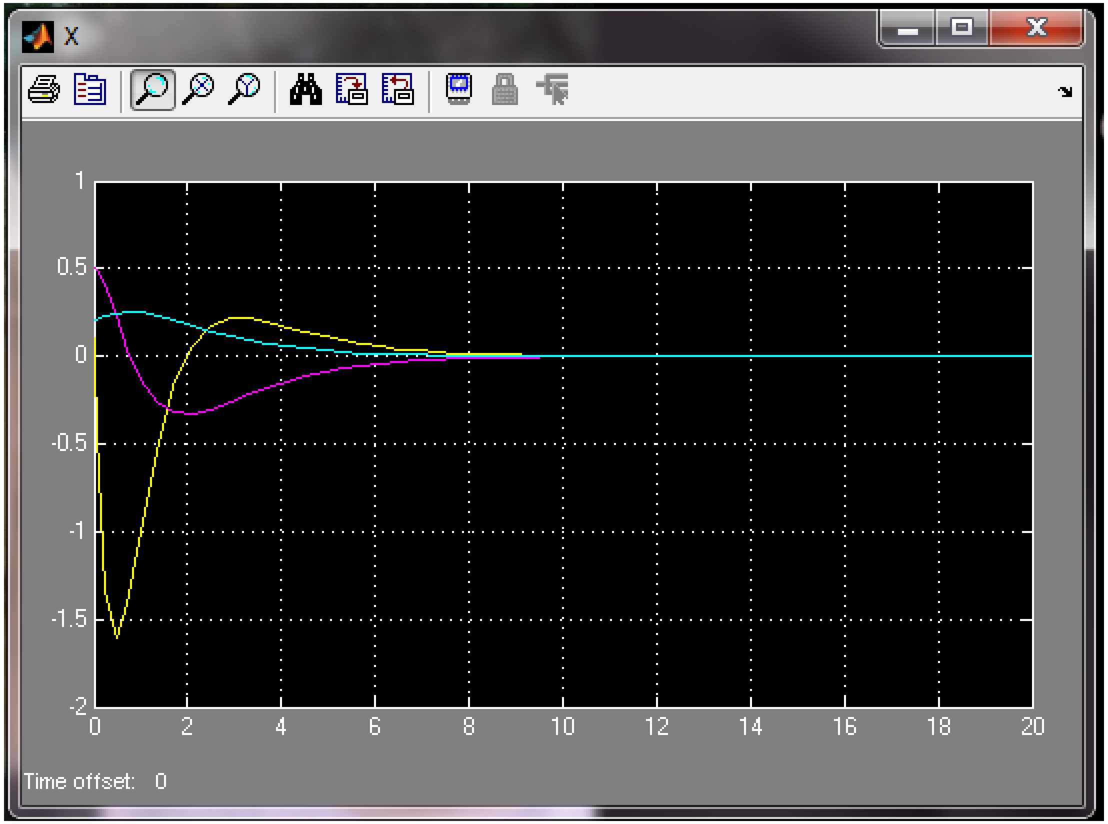
# Pólusáthelyezés állapotvisszacsatolással Simulinkben

A kimenet időfüggvénye



# Pólusáthelyezés állapotvisszacsatolással Simulinkben

Az állapotváltozók időfüggvénye



# Videó

[Pólusáthelyezéses állapotvisszacsatolás megvalósítása]

# Köszönöm a figyelmet!

Dr. habil. Kovács Levente
egyetemi docens

kovacs.levente@nik.uni-obuda.hu

Dr. Haidegger Tamás
egyetemi adjunktus

haidegger@irob.uni-obuda.hu

EKIK
Egyetemi Kutató és Innovációs Központ

K-MOOC

ÓBUDAI EGYETEM
Bejczy Antal iRobottechnikai Központ

Physiological Controls Group

Élettani Szabályozások Csoport
Óbudai Egyetem