

5. Gyakorlat

5. Tantermi gyakorlat – Folytonosidejű szabályozások tervezése állapotterben

A gyakorlat célja, hogy bemutassuk a szabályozók állapotteres méretezésének módszereit: pólus (sajátérték) áthelyezés állapot-visszacsatolással; állapotmegfigyelő tervezése; az alapjel figyelembevétele; integráló hatás vagy terhelésbecslő beépítése. A gyakorlat során csak egybemenetű és egykimenetű (SISO – Single Input, Single Output) rendszerekkel foglalkozunk, a módszerek ugyanakkor egyszerűen kiterjeszthetők több bemenetet és kimenetet tartalmazó rendszerek kompenzálásához.

Állapotegyenlet és kanonikus alakjai

Ismeretes (ld. Tankönyv, illetve röviden az 1. gyakorlat), hogy egy lineáris, időinvariáns, folytonos idejű rendszer dinamikája állapotegyenletével is megadható a

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

alakban, ahol $x(t)$ az állapotváltozók vektorának értéke t időpillanatban az R^n vektortérben. A szokásos módon $u(t)$ a bemenet, míg $y(t)$ a kimenet időfüggvényei, n az állapotter dimenziója és $x(0) = x_0$ az állapotok kezdeti értékeinek vektora. Az első gyakorlat anyagából ismeretes, hogy egy LTI rendszert az `ss` utasítás segítségével adhatunk meg a Matlab környezetben.

Ismeretes továbbá, hogy az állapotegyenletből nulla kezdeti értéket feltételezve a

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

összefüggéssel felírható a ki- és bemenet közötti átviteli függvény. Ez az átviteli függvény invariáns (megőrzi alakját, ld. tankönyv) az állapotter hasonlósági transzformációval történő átalakításaira, amelyek ugyanakkor módosítják az állapotegyenlet alakját. Egy hasonlósági transzformációt (más néven állapotter-transzformációt) egy invertálható T transzformációs mátrix definiál, hogy $z = Tx$ és $x = T^{-1}z$. Az állapotegyenlet z állapotváltozóban felírva

$$\begin{aligned}\dot{z} &= TAT^{-1}z + TBu \\ y &= CT^{-1}z + Du\end{aligned}$$

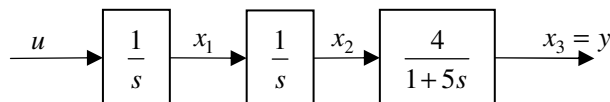
alakú. Mivel egy mátrix sajátértékei nem változnak a hasonlósági transzformációra, az A és TAT^{-1} mátrixok sajátértékei azonosak, egyben az átviteli függvény pólusai.

Az állapotegyenlet egyes kanonikus alakjaiból közvetlenül kiolvashatók az átviteli függvény nevezőjének pólusai vagy együtthatói. A Matlab képes megkeresni adott állapotegyenlethez azokat a transzformációs mátrixokat, amelyekkel az egyes kanonikus alakok előállíthatók, illetve elvégzi a szükséges transzformációt is. Ezen utasításokat összegzi az alábbi táblázat.

Utasítás	Leírás
<code>ss2ss</code>	Állapottér-transzformáció végrehajtása tetszőleges, de invertálható T transzformációs mátrix megadása mellett.
<code>canon</code>	Állapottér-transzformáció kanonikus alakra, a szükséges transzformációs mátrix meghatározása mellett. A támogatott kanonikus alakok: 'companion' – megfigyelhetőségi kanonikus alak 'modal' – modális vagy blokkdiagonális kanonikus alak
<code>ss(tf(.))</code>	Állapottér-transzformáció végrehajtása, a kapott alak az ún. irányíthatósági lépcsős alak.

Az utolsó transzformáció azt használja ki, hogy az `ss` utasítás, ha átviteli függvényből (`tf`) vagy annak zérus-pólus-erősítés (`zpk`) alakjából indulunk ki, akkor irányíthatósági lépcsős alakban adja meg az állapotegyenletet.

Példaként tekintsük az alábbi tagot.



5.1. ábra. Lineáris tag hatásvázlata

Az eredő átviteli függvény

$$W(s) = \frac{4}{s^2(1+5s)}$$

Az állapotok választását az ábrán bejelöltük. A tagot alkotó elemek átviteli függvényeinek nevezői mind elsőfokúak, így az állapotegyenlet felírása egyszerű

$$\frac{X_1(s)}{U(s)} = \frac{1}{s}$$

$$\frac{X_2(s)}{X_1(s)} = \frac{1}{s}$$

$$\frac{X_3(s)}{X_2(s)} = \frac{4}{1+5s}$$

$$\dot{x}_1 = u$$

$$\dot{x}_2 = x_1$$

$$x_3 + 5\dot{x}_3 = 4x_2$$

$$\dot{x}_3 = -\frac{1}{5}x_3 + \frac{4}{5}x_2$$

amelyet felírhatunk mátrixos alakban is, figyelembe véve, hogy $y = x_3$.

$$\dot{x} = \begin{bmatrix} 0 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & \frac{4}{5} & -\frac{1}{5} \end{bmatrix} x + \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} u$$

$$y = \begin{bmatrix} 0 & 0 & 1 \end{bmatrix} x + 0u$$

A Matlab segítségével ellenőrizhetjük, hogy ez az állapotegyenlet a felírt átviteli függvényre vezet.

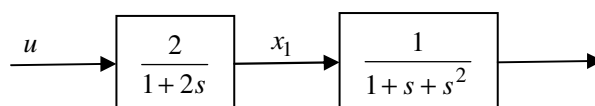
```
>> sys = ss([0 0 0; 1 0 0; 0 4/5 -1/5], [1;0;0], [0 0 1], 0);
>> sys2 = zpk(sys)
```

Zero/pole/gain:

0.8

s^2 (s+0.2)

Most tekintsünk egy másik lineáris tagot, melynek hatásvázlata az ábrán látható.



5.2. ábra. Lineáris tag hatásvázlata

A lineáris tag átviteli függvénye

$$W(s) = \frac{2}{(1+2s)(1+s+s^2)}.$$

Az állapotválasztás legyen az alábbi: $x_2 = y$, $x_3 = \dot{y} = \dot{x}_2$, az ábrán bejelölt x_1 mellett. Az állapotegyenlet felállítása során ismét az elemek által megadott átviteli függvényekből kell kiindulni.

$$\frac{X_1(s)}{U(s)} = \frac{2}{1+2s} \quad x_1 + 2\dot{x}_1 = 2u \quad \dot{x}_1 = -\frac{1}{2}x_1 + u$$

$$\frac{Y(s)}{X_1(s)} = \frac{1}{1+s+s^2} \quad y + \dot{y} + \ddot{y} = x_1 \quad \dot{x}_2 = x_3$$

$$x_2 + x_3 + \dot{x}_3 = x_1 \quad \dot{x}_3 = x_1 - x_2 - x_3,$$

amelyet ismét csak felírhatunk mátrixos alakban is, majd Matlab segítségével ellenőrizhetjük, hogy visszakapjuk-e az átviteli függvényt

$$\dot{x} = \begin{bmatrix} -\frac{1}{2} & 0 & 0 \\ 0 & 0 & 1 \\ 1 & -1 & -1 \end{bmatrix} x + \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} u$$

$$y = \begin{bmatrix} 0 & 1 & 0 \end{bmatrix} x + 0u$$

```
>> syslin = ss([-1/2 0 0; 0 0 1; 1 -1 -1], [1;0;0], [0 1 0], 0);
>> syslin2 = tf(syslin)
```

Transfer function:

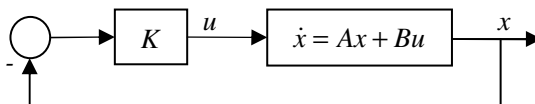
$$\frac{1}{s^3 + 1.5 s^2 + 1.5 s + 0.5}$$

Pólusát helyezés állapot-visszacsatolással

Feltesszük, hogy a szabályzó számára minden időpillanatban rendelkezésre állnak a rendszer állapotai. A beavatkozó jelet minden időpillanatban az állapotok lineáris kombinációjaként állítjuk elő, azaz legyen

$$u = -Kx,$$

ahol K egy n elemű sorvektor ($n = \dim x$), ha a rendszernek egy bemente van. Ez az 5.3. ábrán mutatott állapot-visszacsatolásnak felel meg.



5.3. ábra. Állapot-visszacsatolás

Vegyük észre, hogy a visszacsatoló ágban szereplő összekötetés ezúttal nem egy skalárértékű időfüggvény hatására utal, hanem n darab, vektorba rendezett skalárértékű időfüggvény jelerjedését (a teljes állapot visszacsatolását) jelenti.

A zárt kör állapotegyenlete a visszacsatolás kifejezésének behelyettesítése után

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

amely alapján zárt rendszer sajátértékei az $A - BK$ mátrix sajátértékei lesznek. Ezek a sajátértékek határozzák meg, hogy nemnulla kezdeti értékről az állapotvektor milyen tranziens után éri el az állapottér origóját. Fontos tény, hogy a K vektor megfelelő megválasztásával csak akkor lehet valamely megadott helyre kimozdítani a szakasz sajátértékeit (az A mátrix sajátértékeit), ha az (A, B) pár (állapot)irányítható, amelynek szükséges és elégséges feltétele, hogy az

$$M_c = \begin{bmatrix} B & AB & \cdots & A^{n-1}B \end{bmatrix}$$

ún. irányíthatósági mátrix (lásd tankönyv) rangja n legyen. Ha a kívánt sajátértékek vektora

$$[\lambda_1 \quad \lambda_2 \quad \dots \quad \lambda_n],$$

akkor az $A - BK$ mátrix karakterisztikus egyenlete

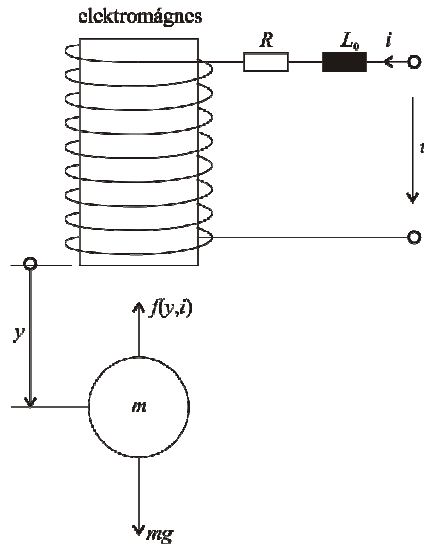
$$\varphi_c(\lambda) = \prod_{i=1}^n (\lambda - \lambda_i)$$

és az ehhez tartozó K (sor)vektort az

$$K = [0 \quad 0 \quad \dots \quad 1] M_c^{-1} \varphi_c(A)$$

Ackermann-képlet segítségével határozhatjuk meg. Vegyük észre, hogy egy polinom változója helyére egy négyzetes mátrixot is behelyettesíthetünk, hiszen egy négyzetes mátrix hatványai is négyzetes mátrixok. A képlet levezetését a tankönyv tartalmazza. A képletben szerepel az M_c mátrix inverze, amelyről láttuk, hogy akkor és csak akkor létezik, ha az (A, B) pár irányítható. Az Ackermann-képletet egyszerűen kódolhatjuk a Matlab segítségével, de a Control System Toolbox-ot készítő programozók ezt már megtették helyettünk, így a képlet kiértékelésére rendelkezésünkre áll az `acker` utasítás (lásd még `doc acker`), amely használatát az alábbi példát mutatjuk meg.

Tekintsünk egy egyváltozós nemlineáris rendszert, egy mágnesesen lebegtetett (magnetically levitated - maglev) golyót. Az 5.4. ábrán a lebegtető rendszer sematikus szerkezeti vázlata látható.



5.4. ábra. Maglev eszköz egyszerűsített vázlata

Az elektromágnes tekercsére kapcsolt feszültség a golyóra ható mágneses erőt befolyásolja, lehetővé téve annak függőleges gyorsítását. A rendszer dinamikus modelljét $L = \frac{Q}{y} + L_0$ esetén az

$$m\ddot{y} = mg + \frac{i^2}{2} \frac{dL}{dy} = mg + \frac{i^2}{2} \left(-\frac{Q}{y^2} \right) \Rightarrow \ddot{y} = g - \frac{Qi^2}{2my^2}$$

$$u = Ri + \frac{d}{dt}(Li) = Ri + \left\{ -\frac{Q}{y^2} \frac{dy}{dt} i + \left(\frac{Q}{y} + L_0 \right) \frac{di}{dt} \right\} \Rightarrow \frac{di}{dt} = \left(\frac{Q}{y} + L_0 \right)^{-1} \left(u - Ri + \frac{Qyi}{y^2} \right)$$

differentiálegyenletek definiálják a táblázatban ismertetett paraméterek mellett.

Paraméter	Jelölés	Érték
golyó tömege	m	$m = 0.8 \text{ kg}$
tekercs induktivitása	L_0	$L_0 = 0.5 \text{ H}$
tekercs ellenállása	R	$R = 10\Omega$
légrészhez tartozó induktivitás	Q	$Q = 0.001 \text{ Hm}$
nehézségi gyorsulás	g	$g = 9.8 \text{ kgm/s}^2$

A lebegtetett golyó függőleges pozícióját y jelöli, i az elektromágnes tekercsében folyó áram, u pedig a tekercs kapcsaira kapcsolt feszültség. A modell kimenete a golyó pozíciója.

Az állapotok $x = (x_1, x_2, x_3)^T = (y, \dot{y}, i)^T$. Innen az állapotegyenlet

$$\dot{x}_1 = x_2$$

$$\dot{x}_2 = g - \frac{Qx_3^2}{2mx_1^2}$$

$$\dot{x}_3 = \left(\frac{Q}{x_1} + L_0 \right)^{-1} \left(u - Rx_3 + \frac{Qx_2x_3}{x_1^2} \right)$$

$$y = x_1$$

Munkapontnak egy adott magasságban mozdulatlanul álló golyóhoz tartozó egyensúlyi értékeket választjuk. Egyensúly esetén az állapotok nem változnak, $x_2 = \dot{y} = 0$. Legyen az egyensúlyi magasság $x_{10} = y_0 = 0.01 \text{ m}$. Értelemszerűen $x_{20} = 0$. A golyónak ebben a magasságban történő megtartásához az elektromágnes tekercsében folyó áram egyszerűen meghatározható a második egyenletből

$$x_{30} = i_0 = \sqrt{\frac{2mgy_0^2}{Q}} = 1.2522 \text{ A}$$

az ehhez tartozó feszültség pedig a harmadikból

$$u_0 = Ri_0 = 12.522 \text{ V}$$

A linearizált rendszer állapotegyenletével kívánunk dolgozni. A parciális deriváltakat tartalmazó mátrixok meghatározása és a munkapontnak megfelelő értékek behelyettesítése után a következő eredményeket kapjuk:

$$A = \begin{bmatrix} 0 & 1 & 0 \\ \frac{Qx_3^2}{mx_1^{-3}} & 0 & -\frac{Qx_3}{mx_1^2} \\ a_{31} & \frac{Qx_1x_3}{x_1^2(Q+L_0x_1)} & \frac{Qx_2-Rx_1^2}{x_1(Q+L_0x_1)} \end{bmatrix}_{x_0, u_0} = \begin{bmatrix} 0 & 1 & 0 \\ 1960 & 0 & -15.65 \\ 0 & 20.87 & -16.66 \end{bmatrix}$$

$$B = \begin{bmatrix} 0 \\ 0 \\ \frac{x_1}{Q+L_0x_1} \end{bmatrix}_{x_0, u_0} = \begin{bmatrix} 0 \\ 0 \\ 1.6667 \end{bmatrix} \quad C = [1 \quad 0 \quad 0]$$

ahol

$$a_{31} = \frac{(u-Rx_3)(Q+L_0x_1)-L_0x_1(u-Rx_3)}{(Q+L_0x_1)^2} + \frac{Qx_1^2x_2x_3(Q+L_0x_1)-Q^2x_1^2x_2x_3(2+3L_0x_1)}{(Qx_1^2+L_0x_1^3)}$$

A linearizált rendszer

$$\delta\dot{x} = A\delta x + B\delta u$$

$$\delta y = C\delta x + D\delta u$$

állapotegyenlete a munkapont körüli eltérésekre igaz, azaz $\delta u = u - u_0$, $\delta x = x - x_0$ és $\delta y = y - y_0$. A továbbiakban ezzel a modellel foglalkozunk a tantermi gyakorlat során és a lineáris állapotegyenlet felírásakor a jelölésben a δ szimbólumokat elhagyjuk. Az állapotmátrix sajátértékeit a Matlab segítségével számíthatjuk

```
>> A=[0 1 0; 1960 0 -15.66; 0 20.87 -16.66];
>> B=[0;0;1.6667];
```

```
>> C=[1 0 0];
>> D=0;
>> eig(A)
```

```
ans =
    41.5554
   -36.9484
   -21.2671
```

Megállapítható, hogy a szakasz labilis. Az elsődleges célunk az állapot-visszacsatolással, hogy zárt körben stabilis rendszert kapjunk. Vizsgáljuk meg először, hogy a rendszer irányítható-e.

```
>> Mc=ctrb(A,B);
>> rank(Mc)
```

```
ans =
     3
```

Itt a `ctrb` utasítás az irányíthatósági mátrixot számítja, a `rank` utasítás pedig egy mátrix rangját határozza meg. Megállapítható, hogy az irányíthatósági mátrix rangja megegyezik a rendszer állapotainak számával, tehát a rendszer irányítható.

A stabil működésű zárt kör sajátértékei legyenek $\lambda_1 = \lambda_2 = \lambda_3 = -20$. Az ehhez szükséges $u = -Kx$ állapot-visszacsatolás K erősítésvektorát az `acker` utasítással határozzuk meg.

```
>> K = acker(A,B,[-20 -20 -20])
```

```
K =
    1.0e+003 *
   -4.8152   -0.1086    0.0260
```

Ellenőrzésképpen kiszámíthatjuk a zárt kör állapotegyenletének sajátértékeit.

```
>> eig(A-B*K)
```

```
ans =
   -19.9998 + 0.0003i
   -19.9998 - 0.0003i
   -20.0003
```

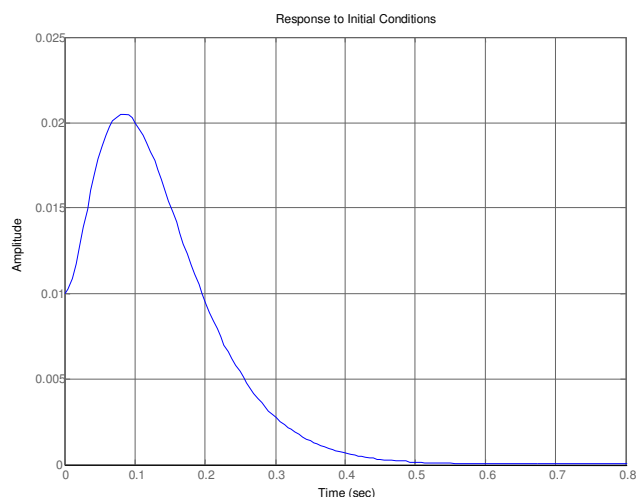
Látható, hogy nagy pontossággal sikerült a zárt kör sajátértékeit az előírt pozícióba mozgatni.

Most vizsgáljuk meg a rendszer viselkedését zárt körben. Ehhez az `initial` utasítást (doc `initial`) használjuk. Az utasítás számára meg kell adni az

állapotegyenletet és az állapotvektor x_0 kezdeti értékét. Az utasítás a kimenet (és opcionálisan az állapotok) alakulását számítja azonosan nulla bemenő jel mellett. Jelen esetben az utasítás alakja

```
>> initial(ss(A-B*K, [], [1 0 0], []), [0.01 0 0])
Warning: Some of the specified systems have no input and/or output.
> In rfinputs at 93
  In lti.initialplot at 82
  In lti.initial at 97
```

ahol a figyelmeztetés (Warning) oka, hogy a zárt kör állapotegyenletét megadva a bemenetet szorozó mátrix helyén üres [] mátrix szerepel. De tekintettel arra, hogy az `initial` utasítás a bemenetre úgyis azonosan nulla jelet ad, ennek *itt* nincsen jelentősége és a kimenet alakulására (a golyó pozíciójának tranziensére) az alábbi ábrát kapjuk a $\delta x_0 = [0.01 \ 0 \ 0]$ kezdeti állapot esetén, azaz a golyó egy centiméterrel történő kimozdítása után.

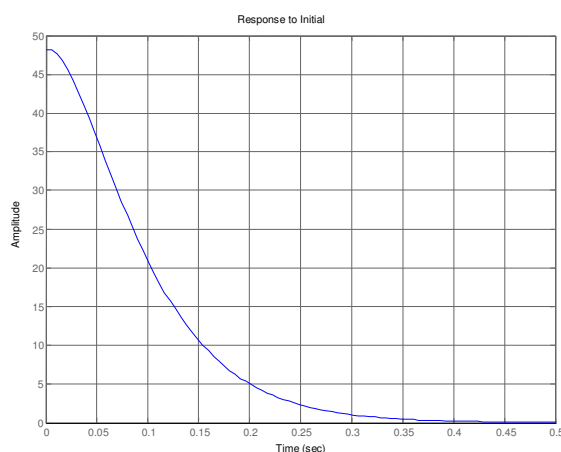


5.5. ábra. A lebegtetett golyó pozíciója az egyensúlyi helyzethez képest stabilizált zárt körben

Az ábra azt mutatja, hogy a golyó munkaponti pozícióját (az ábra az ehhez képesti eltéréseket mutatja) sikerült stabilizálni az állapot-visszacsatolásnak köszönhetően. Vegyük észre, hogy egy egyszerű módszerrel a bemenet alakulását is megjeleníthetjük, ha kimenetként a $-Kx$ jelet kívánjuk ábrázolni.

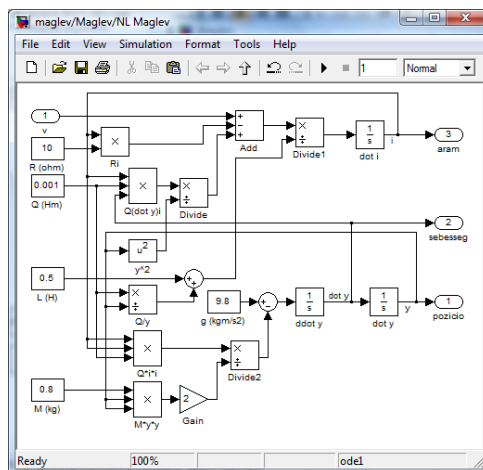
```
>> initial(ss(A-B*K, [], -K, []), [0.01 0 0])
Warning: Some of the specified systems have no input and/or output.
> In rfinputs at 93
  In lti.initialplot at 82
  In lti.initial at 97
```

A munkaponti egyensúlyi helyzet eléréséhez szükséges feszültség alakulását mutatja az ábra, most is az egyensúlyi helyzethez tartozó u_0 feszültséghez képest.

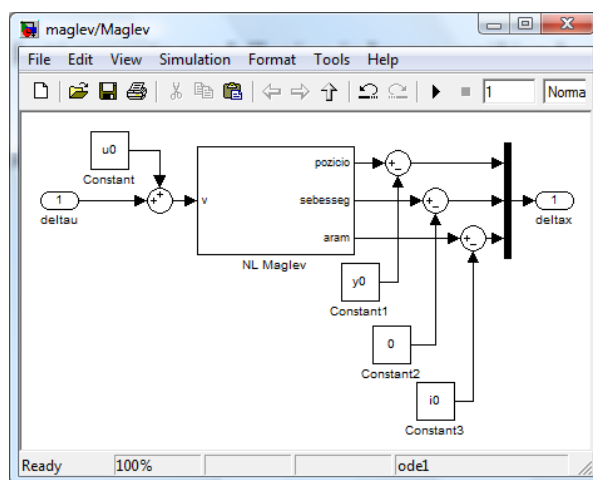


5.6. ábra. Az elektromágnes kapcsaira jutó feszültség alakulása (a munkaponti feszültséghez képest).

Tekintettel arra, hogy az első gyakorlatról ismeretes a Simulink segítségével felépített nemlineáris állapotegyenlet (5.7. ábra), azt is megvizsgálhatjuk, hogy mennyire tér el a zárt rendszer viselkedése, ha ugyanahhoz a (lineáris) állapot-visszacsatoláshoz nem a linearizált rendszer állapotegyenletét, hanem a nemlineáris rendszer állapotegyenletét szimuláljuk. Nyilván ez utóbbi eset áll közelebb a valóságos rendszer viselkedéséhez.

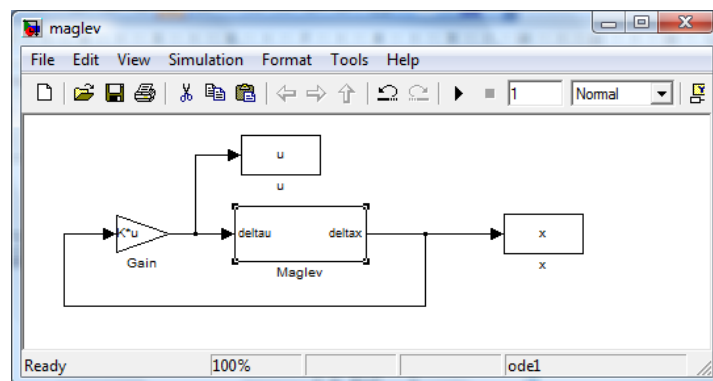


Mivel az állapot-visszacsatolás a munkaponthoz képesti eltérések alapján állítja elő a munkaponti értékehez képest eltolat beavatkozó jelet, ezért ezeket az eltolásokat el kell végezni.



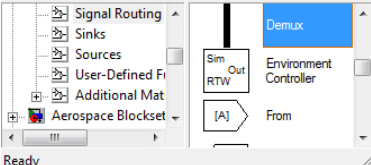
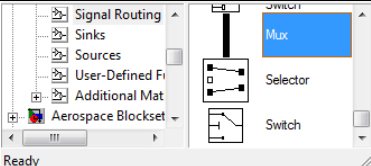
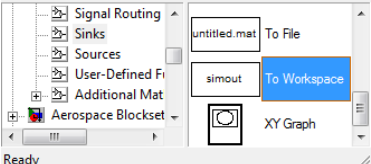
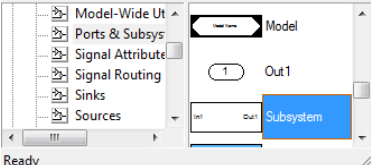
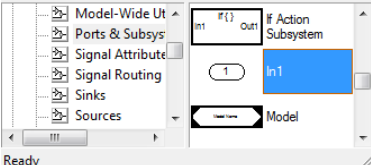
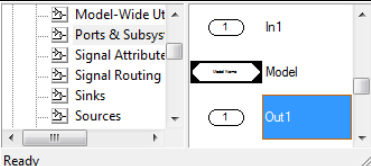
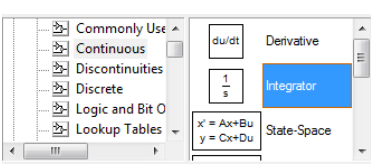
5.8. ábra. Korrekció a munkaponti értékekkel (az 5.7. ábra diagramját az itt szereplő NL Maglev blokk tartalmazza).

Végül pedig a Simulink segítségével meg kell valósítani az állapot-visszacsatolást, ezt mutatja az 5.9. ábra.



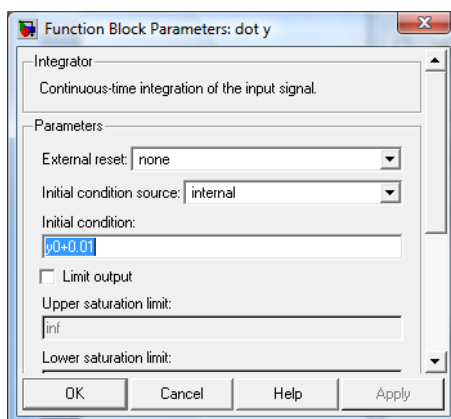
5.9. ábra. Az állapot-visszacsatolás megvalósítása (az 5.8. ábra diagramját az itt szereplő Maglev blokk tartalmazza)

A diagram felépítése egyszerű, ugyanakkor a teljesség kedvéért néhány újonnan használt Simulink elem tulajdonságait és fellelhetőségét foglalja össze az alábbi táblázat.

Név (hely a könyvtárban)	Szimbólum	Magyarázat
Demux (Signal Routing)		Vektorba rendezett jelcsoportot szétbont. A kimenetek számát kell megadni.
Mux (Signal Routing)		Vektorba rendez több jelet. A bemenetek számát kell megadni.
To Workspace (Sinks)		Jel (jelek vektorának) alakulását a munkatérbe menti további műveletek (például ábrázolás) céljából.
Subsystem (Ports & Subsystems)		Alrendszer definiálására szolgál. Tartalmazhat tetszőleges diagram elemeket, így újabb alrendszereket is.
In1 (Ports & Subsystems)		Alrendszer bemeneti jele. Az alrendszer „dobozán” nyelőként fog megjelenni.
Out1 (Ports & Subsystems)		Alrendszer kimeneti jele. Az alrendszer „dobozán” forrásként fog megjelenni.
Integrator (Continuous)		Integrátor. Itt egy integrátorhoz egy állapotot rendelünk. Az integrátor elemekhez kezdeti feltételt lehet hozzárendelni.

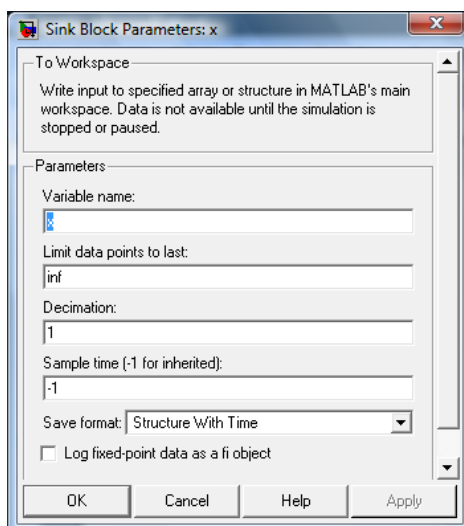
Ügyeljünk arra, hogy a nemlineáris modell esetében a kezdeti feltételt *nem* a munkaponti értékhez képest kell megadni az állapotoknak megfelelő

integrátoroknál. Tehát az $x_1 = y$ állapotnak kezdeti értéként $0.01 + 0.01 = 0.02$ métert kell megadni, ha a linearizált rendszert és a nemlineáris rendszert azonos kezdeti feltételről kívánjuk indítani. Ezt a beállítást mutatja az 5.10. ábra.



5.10. ábra. Az 5.7. ábra diagramjában szereplő dot y integrátorhoz tartozó párbeszédablak a kezdeti érték beállításával.

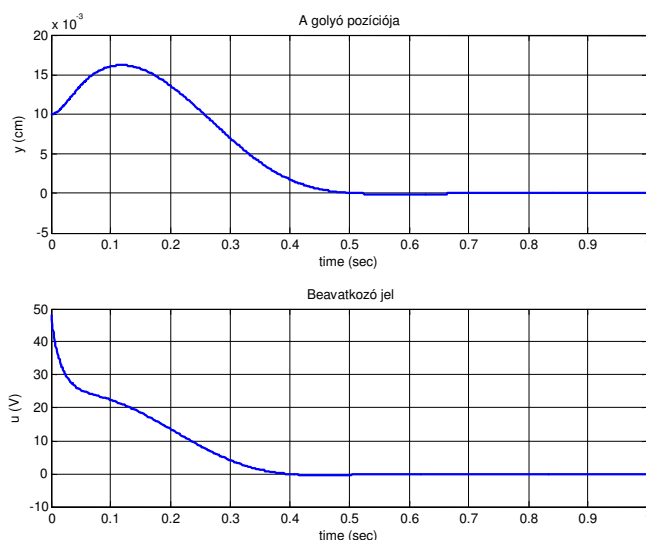
A szimuláció eredményéhez a To Workspace elemek beiktatása miatt a szimuláció lefuttatása nyomán a munkatérben is hozzáférhetünk. Ehhez az elemhez tartozó párbeszédablakban a változó nevét és struktúráját is meg kell adnunk.



5.11. ábra. Jel munkatérbe mentésére szolgáló elem paramétereinek beállításai.

A szimuláció futtatása után egyszerű `plot` utasítások segítségével ábrázolhatóak az adatok, illetve más műveleteket is végezhetünk velük.

```
>> subplot(2,1,1);
>> plot(x.time,x.signals.values(:,1));
>> subplot(2,1,2);
>> plot(u.time,u.signals.values);
```



5.12. ábra. A kimenet és a beavatkozó jel alakulása a munkaponthoz képest a nemlineáris modellel szimulálva a szakasz működését

Látható, hogy a stabilizálás a nemlineáris modell esetén is sikeres az adott kezdeti eltérés mellett, bár a tranziensek némileg másképpen alakulnak.

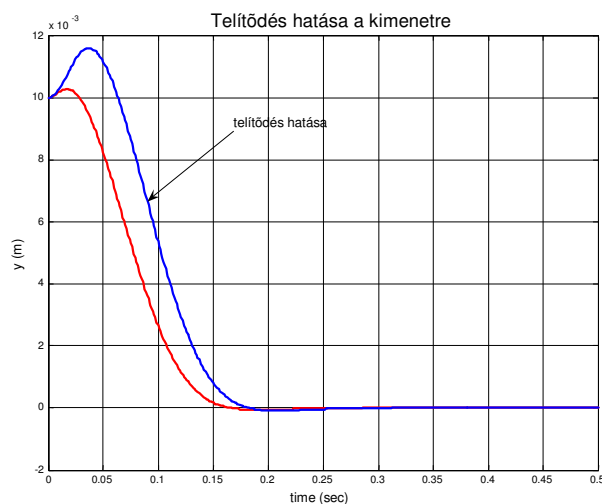
A pólusáthelyezés korlátai

Állapotirányítható rendszer esetében az állapot-visszacsatolás segítségével elviekben a komplex számsík bal félsíkján tetszőleges helyre helyezhetőek el a zárt kör pólusai feltéve, hogy a komplex pólusok konjugált párjukkal fordulnak elő.

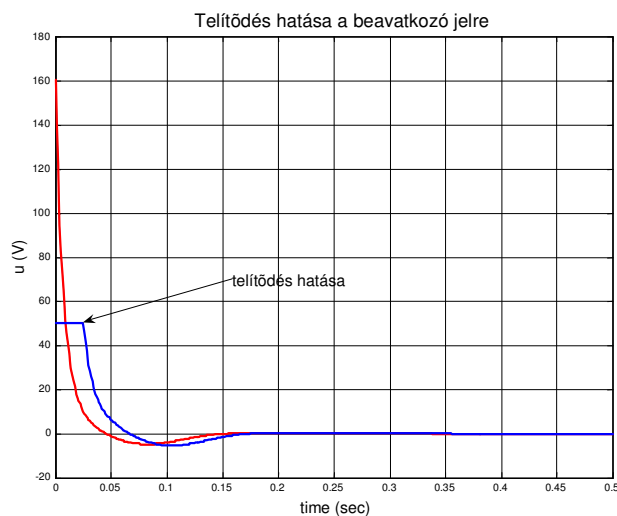
A zárt kör stabilitása mellett általában a követési tulajdonságok javítása érdekében minél gyorsabb viselkedésre törekszünk, azaz zárt körben lehetőleg abszolút értékben nagy, negatív valós részű pólusokat írunk elő. Ennek az szab határt, hogy a beavatkozó jel az őt előállító beavatkozó szerv fizikai (például teljesítménybeli) korlátai miatt nem lehet tetszőlegesen nagy. Ezt a mágneses lebegtetés példáján kívánjuk illusztrálni.

Feltesszük, hogy az elektromágneset vezérlő elektronika a tekercs kapcsaira nem képes a ± 50 Voltos tartományon kívüli jelet adni (a munkaponti feszültséget

Az 5.14. és 5.15. ábrák mutatják a telítődés hatását a rendszer tranzienseire. A kezdeti érték a munkaponthoz képest továbbra is $\delta x_0 = [0.01 \ 0 \ 0]^T$.



5.14. ábra. A beavatkozó szerv telítődéses jellegének hatása a kimenet tranziensében



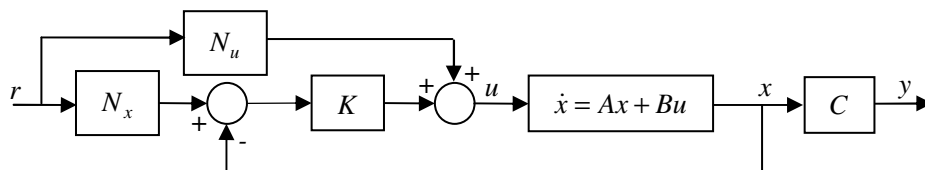
5.15. ábra. A beavatkozó szerv telítődéses jellegének hatása a beavatkozó jel tranziensében

Látható, hogy a telítődéses jelleg ugyan ebben az esetben nem vezetett labilis viselkedéshez, de a kimenet tranziensében legalábbis késleltetést okozott.

Az alapjel figyelembevétele

Amennyiben a szabályozás célja nem egyszerűen az állapottér origójának stabilizálása, hanem valamely nullától különböző alapjel követése, a hatásvázlatot ki kell egészíteni az alapjel figyelembevételére szolgáló elemekkel.

Feltesszük, hogy az állapotegyenlet D mátrixa nulla így $y = Cx$. Az alapjel figyelembevételére szolgáló struktúrát mutatja az ábra.



5.16. ábra. Az alapjel figyelembevételére szolgáló elemek (N_x és N_u) a hatásvázlatban

Az N_x és N_u erősítések kiszámítása során abból az előírásból indulunk ki, hogy egységugrás alapjel esetén állandósult állapotban a különbségképző szerv kimenetén 0 értékű vektort találjunk, a szabályozott jellemző értéke pedig egy legyen. Ez az

$$\begin{aligned} AN_x + BN_u &= 0 \\ CN_x &= 1 \end{aligned}$$

egyenletekre vezet, amelyeket mátrixos alakba átírva a

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

összefüggéseket kapjuk. A mágneses lebegtetés példája esetében az N_x és N_u erősítések értékét a Matlab segítségével számítjuk.

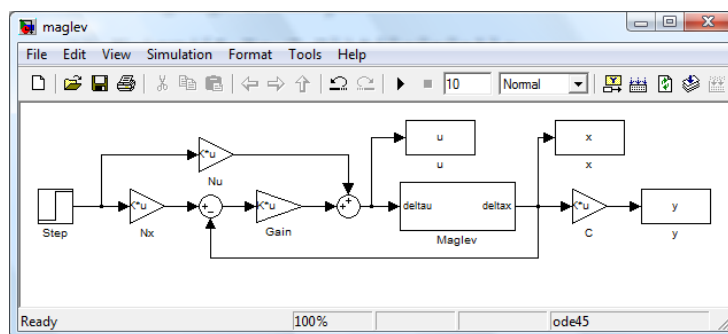
```
>> N=inv([A B; C 0])*[0;0;0;1];
>> Nx=N(1:3)
```

```
Nx =
    1.0000
   -0.0000
   125.2496
```

```
>> Nu = N(4)
```

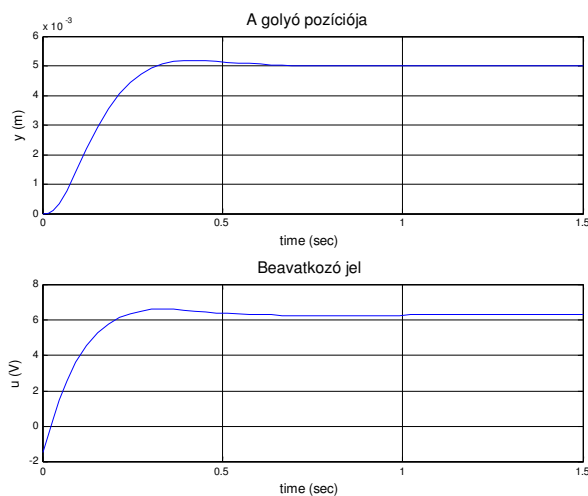
```
Nu =
    1.2519e+003
```

Az erősítéseket hozzáadjuk a Simulink segítségével felépített diagramhoz. Az erősítés elemek párbeszédablakában be kell állítani, hogy a Simulink a szorzásokat a mátrixszorzás szabályai szerint végezze (ennek az N_x erősítésnél van jelentősége).



5.17. ábra. A szabályozási kör hatásvázlata Simulinkben az alapjel figyelembevételére szolgáló elemekkel és a lebegtetés nemlineáris modelljével

A szabályozási kör viselkedését 5mm nagyságú alapjelugrásra vizsgáljuk, ezúttal a kezdeti állapotot nem módosítottuk, az a korábban meghatározott x_0 munkapontnak felel meg.



5.18. ábra. A golyó pozíciójának és a beavatkozó jelnek transziensei egységugrás alapjel (5 mm) esetén

Kimeneti visszacsatolás állapotmegfigyelő használatával

Az állapot-visszacsatolás hátránya, hogy minden állapot pillanatnyi értékét ismerni kell a beavatkozó jel meghatározásához. Ez egy valódi szakasz esetében az állapotok

számának megfelelő érzékelő beépítését és bekábelezését feltételezi, amely adott esetben nagy költséget jelenthet. Célszerű lenne egy olyan elem használata, amely a kimenet és a bemenet alakulásának ismeretében becsli az állapotok pillanatnyi értékét és megfelelően pontos modell esetén garantálni tudja, hogy a becsült érték a valódi értékhez tart. Ezt az elemet állapotmegfigyelőnek (state observer) nevezik. Az állapotmegfigyelő egy dinamikus rendszer, melynek állapotai a becsült állapotok. Az állapotmegfigyelő állapotegyenlete

$$\dot{\hat{x}} = F\hat{x} + Gy + Hu,$$

amelyet a szabályzó valósít meg. Ha a becslés hibáját az $\tilde{x} = x - \hat{x}$ összefüggéssel definiáljuk, akkor a $H = B$ választás mellett a becslő hibájának tranziense a

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

differenciálegyenletet elégíti ki, ahol F mátrix sajátértékeit G vektor megfelelő megválasztásával tetszőlegesen beállíthatjuk, feltéve, hogy az (A, C) pár megfigyelhető. Az F mátrix sajátértékeinek beállítása algebrailag hasonló a pólusáthelyezési feladathoz, itt is az Ackermann-képletet lehet használni, de az A mátrix és a C vektor transzponáltjaival kell számolni.

Az állapot-visszacsatolásban a becslő állapotait használhatjuk fel, az alapjel figyelembevételére szolgáló elemek változatlanok maradnak.

Az állapotmegfigyelő tranzienseit szeretnénk gyorsabbra választani, mint a zárt kör átvitelének tranzienseit, így annak sajátértékeit a komplex számsíkon a képzetes tengelytől balra írnak elő. A számítások Matlabban az alábbiak a mágneses lebegtetés példáját tekintve

```
>> Gt = acker(A',C',[-50 -50 -50])
```

```
Gt =
    1.0e+003 *
         0.1333         6.9119         0.0671
```

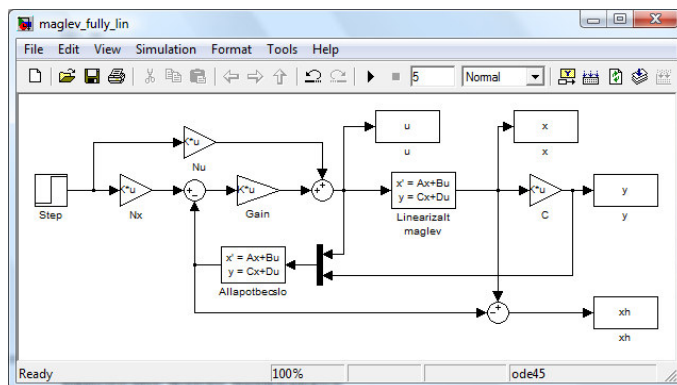
```
>> G = Gt'
```

```
G =
    1.0e+003 *
         0.1333
         6.9119
         0.0671
```

A Simulink diagrammban az állapotbecslőt állapotegyenletével valósítjuk meg. Ügyelni kell arra, hogy az állapotbecslő egy kétbemenetű és három kimenetű (MIMO) rendszer. Az egyszerűség kedvéért a zárt kör viselkedését a linearizált

szakaszmodellt felhasználva vizsgáljuk. Az alkalmazott Simulink modellt mutatja az 5.19. ábra.

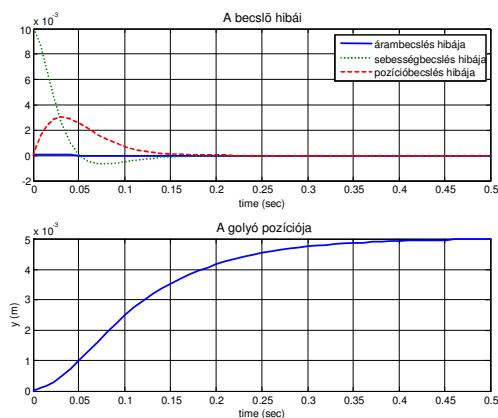
Az állapotbecslő működésének vizsgálatához más kezdeti értékeket állítunk be a szakasz modelljében és a becslőben, így az állapotbecslő nem nulla kezdeti hibáról indul. Legyen $\hat{x}_0 = [0 \ 0 \ 0]$ és $\delta x_0 = [0 \ 0.01 \ 0]$.



5.19. ábra. A zárt szabályozás kör a linearizált szakaszmodellel (Linearizált maglev), az állapotbecslővel, a pólusáthelyezéssel és az alapjel figyelembevételére szolgáló elemekkel

A becslés hibájának (ezt a munkatérbe az xh változóként mentjük el) és a kimenet transziensének felrajzolását az alábbi Matlab kód végzi el.

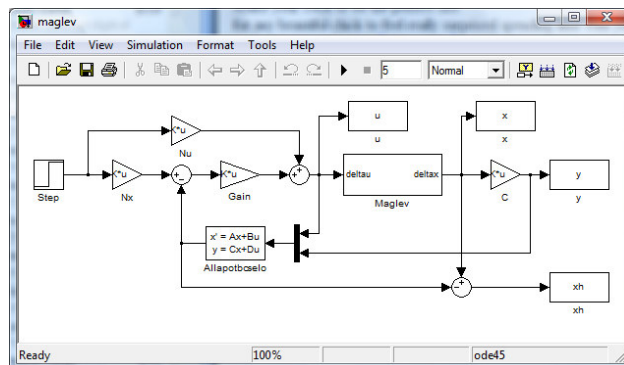
```
>> subplot(2,1,1);
>> plot(xh.time,xh.signals.values);
>> subplot(2,1,2);
>> plot(y.time,y.signals.values);
```



5.20. ábra. Az állapotbecslői hibái és a kimenet zárt körben. Az egységugrás alapjel végértéke itt is 5 mm.

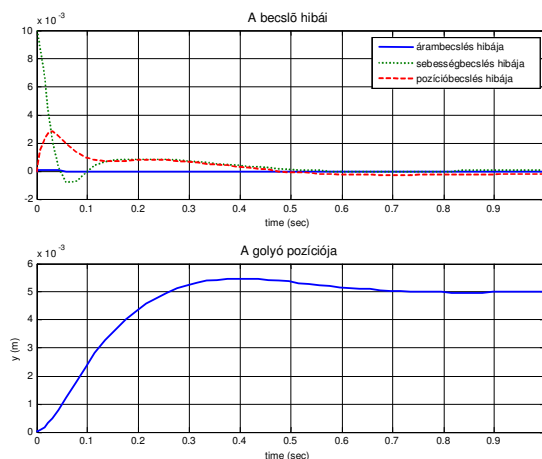
A várakozásoknak megfelelően a becslés kezdeti hibája egy tranziens után eltűnik.

Most tekintsük azt az esetet, amikor a becslőt, az állapot-visszacsatolást és az alapjel figyelembevételére szolgáló elemeket az eredeti nemlineáris modell köré építjük, ahogy ezt a 5.21. ábra mutatja.



5.21. ábra. A zárt szabályozás kör a nemlineáris szakaszmodellel (Maglev), az állapotbecslővel, a polusát helyezéssel és az alapjel figyelembevételére szolgáló elemekkel

A korábbi szimulációhoz hasonlóan itt is megjelenítjük a becslés hibáját és a kimenet tranziensét. Az alapjel és az állapotok kezdeti értéke a korábbi esettel megegyezik.



5.22. ábra. Az állapotbecslői hibái és a kimenet zárt körben nemlineáris szakaszmodellel szimulálva. Az egységugrás alapjel végértéke továbbra 5 mm.

Tekintettel arra, hogy az állapotbecslőben a linearizált szakaszmodell mátrixait használjuk fel és a rendszert az alapjelugrás eltávolítja a munkaponttól, (az ábrán

alig látható) maradó hiba lép fel a becslésben és ennek következtében a kimeneten is. A maradó hibát integráló hatás beiktatásával vagy terhelésbecslővel kívánjuk elnyomni.

Integráló hatás beiktatása

Az integráló hatás beiktatását új állapot bevezetésével érjük el. Az új állapothoz tartozó differenciálegyenletet – az állapotbecslő differenciálegyenletéhez hasonló módon – a szabályzó tartalmazza és működése során numerikusan integrálja. Legyen az új állapotváltozó a kimenet integrálja

$$x_I = \int y dt ,$$

ahonnan az új állapothoz tartozó differenciálegyenlet

$$\dot{x}_I = y = Cx ,$$

tehát az integrátor állapotával bővített lineáris szakaszmodell most

$$\begin{bmatrix} \dot{x} \\ \dot{x}_I \end{bmatrix} = \begin{bmatrix} A & 0 \\ C & 0 \end{bmatrix} \begin{bmatrix} x \\ x_I \end{bmatrix} + \begin{bmatrix} B \\ 0 \end{bmatrix} u$$

alakú. Az állapot-visszacsatolást most ehhez a bővített rendszerhez kívánjuk megtervezni, azaz

$$u = - \begin{bmatrix} K_i & K_I \end{bmatrix} \begin{bmatrix} x \\ x_I \end{bmatrix} ,$$

amihez ezúttal négy pólust kell megadni az Ackermann-képletben az alábbi Matlab kód szerint

```
>> Kh = acker([A zeros(3,1); C 0],[B; 0],[-30 -30 -30 -30]);
>> KI = Kh(4)

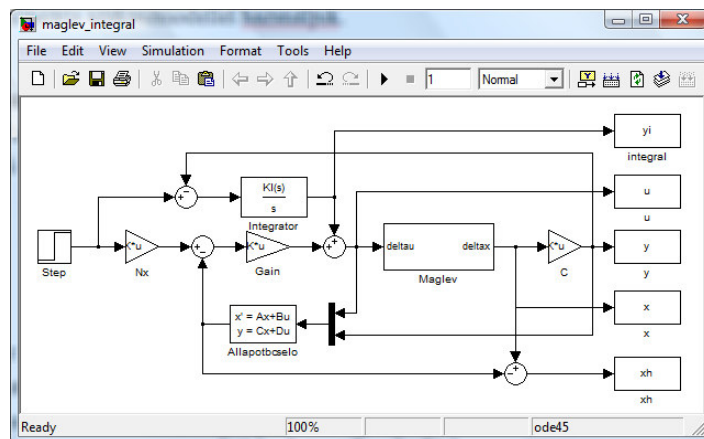
KI =
-3.1054e+004

>> Ki = Kh(1:3)

Ki =
1.0e+004 *
-1.3158    -0.0270    0.0062
```

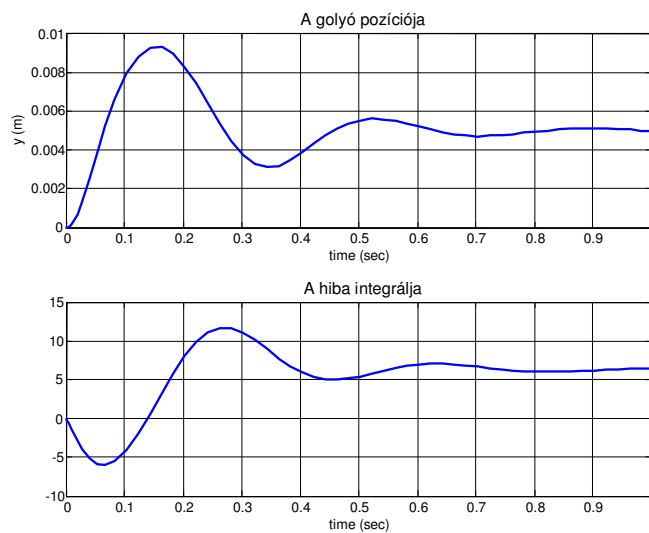
Az integrátort tartalmazó hatásvázlatot mutatja az 5.23. ábra. A zárt körben a mágneses lebegtetés linearizált modelljét használjuk fel. Az integráló hatás az N_u elem hatását is kiváltja. A kezdeti állapot a becslő és a szakasz esetében a golyó

Most tekintsük azt az esetet, amikor az integráló hatást tartalmazó visszacsatolást a nemlineáris szakaszmodellel használjuk.



5.25. ábra. Az integráló hatást tartalmazó szabályozási kör a mágneses lebegtetés nemlineáris modelljével.

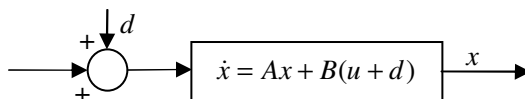
A tranziensek az előzőekben megadott beállításokkal azonos esetben az 5.26. ábrán láthatók.



5.26. ábra. Az integráló hatást tartalmazó szabályozási kör tranziensei: a golyó pozíciója és a hibajel integrálja (lineáris szakaszmodellt használva).

A bemenetre ható zavarás kompenzálása terhelésbecslővel

A terhelésbecslő lehetővé teszi a szakasz bemenetén ható konstans, de ismeretlen értékű d zavarás kompenzálását. A zavarás állandó volta miatt teljesül, hogy $\dot{d} = 0$.



5.27. ábra. A szakasz bemenetén ható zavarás

Az elgondolás szerint a zavarást x_d új állapotként bevesszük a becslendő állapotok közé és $\hat{x}_d$ becsült értékével kompenzáljuk a zavarás hatását. A becslőt tehát a bővített

$$\begin{bmatrix} \dot{x} \\ \dot{x}_d \end{bmatrix} = \begin{bmatrix} A & B \\ 0 & 0 \end{bmatrix} \begin{bmatrix} x \\ x_d \end{bmatrix} + \begin{bmatrix} B \\ 0 \end{bmatrix} u$$

$$y = \begin{bmatrix} C & 0 \end{bmatrix} \begin{bmatrix} x \\ x_d \end{bmatrix}$$

rendszerhez tervezzük. A becslő differenciálegyenlete most

$$\begin{bmatrix} \dot{\hat{x}} \\ \dot{\hat{x}}_d \end{bmatrix} = \tilde{F} \begin{bmatrix} \hat{x} \\ \hat{x}_d \end{bmatrix} + \tilde{G}y + \tilde{H}u.$$

Ha most a

$$\tilde{H} = \begin{bmatrix} B \\ 0 \end{bmatrix} \quad \tilde{F} = \begin{bmatrix} A & B \\ 0 & 0 \end{bmatrix} - \tilde{G}\tilde{C}$$

választásokkal élünk, akkor a becslés hibája a

$$\begin{bmatrix} \dot{\tilde{x}} \\ \dot{\tilde{x}}_d \end{bmatrix} = \tilde{F} \begin{bmatrix} \tilde{x} \\ \tilde{x}_d \end{bmatrix}$$

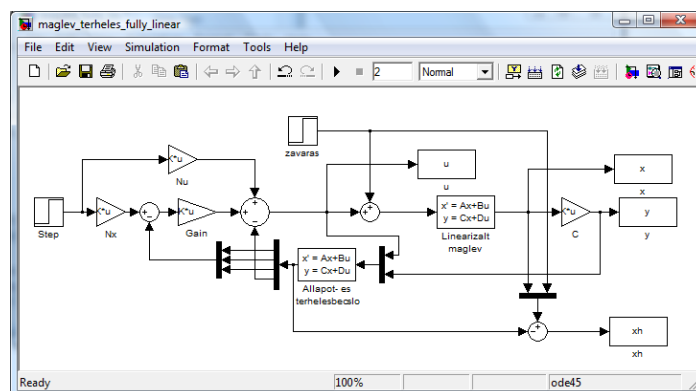
differenciálegyenletet elégíti ki, ahol $\tilde{F}$ mátrix sajátértékeit a korábbiakhoz hasonló módon $\tilde{G}$ vektor megfelelő megválasztásával állíthatjuk be. Az erre szolgáló Matlab kód a mágneses lebegtetéses példájához az alábbi.

```
>> Gth = acker([A B; 0 0 0 0]', [C 0]', [-40 -40 -40 -40])
```

```
Gth =  
1.0e+004 *
```

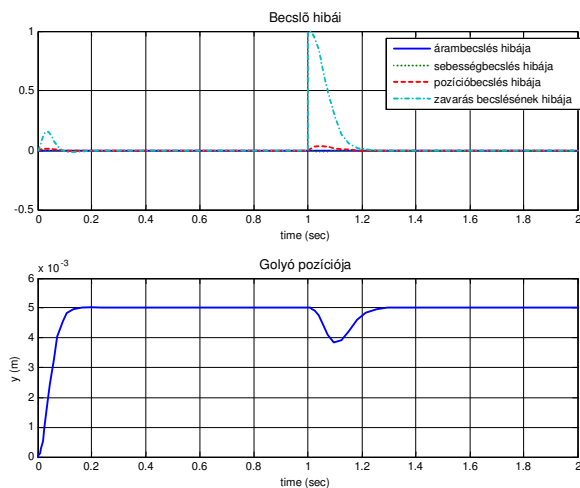
0.0143 0.8845 -0.6037 -9.8145

A szabályozási kört a linearizált szakasszal, a zavaró jellel, a terhelésbecslőt is tartalmazó állapotmegfigyelővel, a pólusáthelyezést biztosító állapot-visszacsatolással, valamint az alapjel figyelembevételére szolgáló elemekkel mutatja az 5.28. ábra.



5.28. ábra. Terhelésbecslőt tartalmazó szabályozási kör a mágneses lebegtetés linearizált modelljével.

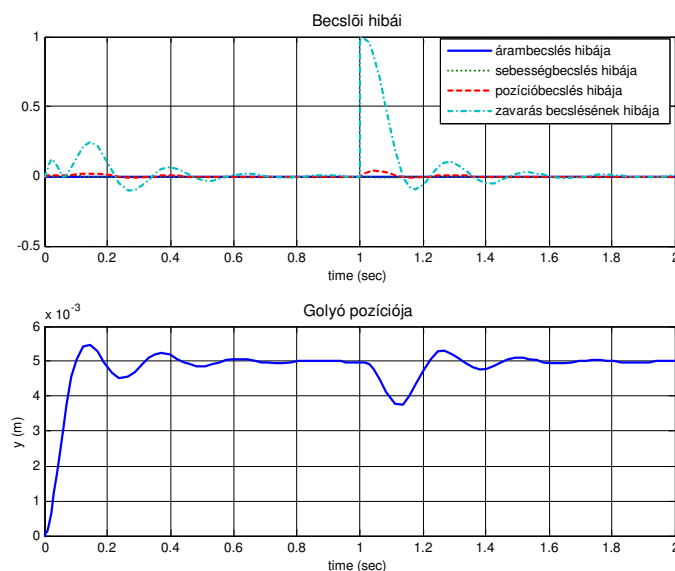
A becslő kezdeti állapota továbbra is $x_0 = 0$, a rendszer kezdeti állapota pedig ezúttal szintén $\hat{x}_0 = [0 \ 0.01 \ 0]$. Az alapjel egy 5 mm-es egységugrás, a bemenetre ható zavarás pedig egy 1 Volt nagyságú egységugrás, amelyik a szimuláció kezdete után egy másodperccel érkezik.



5.29. ábra. Az állapot- és terhelésbecslő hibája, a kimenet tranziense.

A terhelésbecslő hibája egy (1 másodperc után, a terhelés ugrásával induló) tranziens után nullára áll be. A zavarás hatása a kimeneten is jelentkezik, ami annak a következménye, hogy a terhelésbecslő csak valamilyen tranziens után képes pontosan kompenzálni a zavarás hatását.

Most vizsgáljuk meg azt az esetet, amikor a mágneses lebegtetés nemlineáris modelljét illesztjük be a szabályozási körbe.

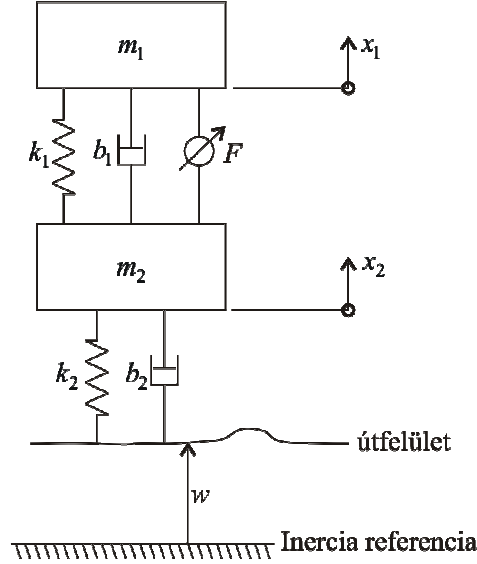


5.30. ábra. Az állapot- és terhelésbecslő hibája, a kimenet tranziense a lebegtetés nemlineáris modelljét használva.

Busz aktív kerékfelfüggesztés szabályozása

A (negyed) busz aktív kerékfelfüggesztésének dinamikus modelljét a III.5 pontban tárgyaltuk, a 3. gyakorlat keretében pedig megvizsgáltuk az analóg szabályozás tervezésének lehetőségeit frekvenciatartományban. Megadtuk az aktív kerékfelfüggesztés egyszerűsített vázlatát, amelyet a 5.31. ábrán megismétlünk.

A busz 4 részre osztható 4 kerék esetén, minden rész egy összetett mechanikai lengőrendszer, amely a karosszéria negyedrészeinek tömegéből (utasokkal együtt) és a kerék tömegéből áll, melyek függőleges helyzetét a felfelé mutató x_1 és x_2 írja le. Az útfelület és a kerék talppontja közötti távolság a mozgás közben az útfelülettel együtt változó w (zavaró jellemző). Az egyes részek közötti kapcsolat flexibilis, melyet viszkózus súrlódással és rugóállandóval modellezünk. A felfüggesztés aktív, amelyet az F erő képvisel (beavatkozó jel).



5.31. ábra. Busz aktív kerékfelfüggesztésének egyszerűsített vázlata

A (negyed) karosszéria és a kerék tömegét rendre m_1 és m_2 , az egyes komponensek közötti flexibilis kapcsolatot a viszkózus súrlódási tényezők (b_1 , b_2) és rugóállandók (k_1 , k_2) írják le.

A III.5 pontban megadtuk a mozgásegyenletet és abból kiindulva levezettük az eredő állapotegyenletet:

Mozgásegyenlet:

$$m_1 \ddot{x}_1 = F - b_1(\dot{x}_1 - \dot{x}_2) - k_1(x_1 - x_2)$$

$$m_2 \ddot{x}_2 = -F + b_1(\dot{x}_1 - \dot{x}_2) + k_1(x_1 - x_2) - b_2(\dot{x}_2 - \dot{w}) - k_2(x_2 - w)$$

Állapotegyenlet: $\dot{\bar{x}} = A\bar{x} + Bu$, $y = C\bar{x}$

$$\bar{x} := (x_1, \dot{x}_1, y_1, \dot{y}_1)^T, \quad u = (F, w)^T, \quad y = x_1 - x_2 = y_1$$

$$A = \begin{bmatrix} 0 & 1 & 0 & 0 \\ -\frac{b_1 b_2}{m_1 m_2} & 0 & \frac{b_1}{m_1} \left(\frac{b_1}{m_1} + \frac{b_1}{m_2} + \frac{b_2}{m_2} \right) - \frac{k_1}{m_1} & -\frac{b_1}{m_1} \\ \frac{b_2}{m_2} & 0 & -\left(\frac{b_1}{m_1} + \frac{b_1}{m_2} + \frac{b_2}{m_2} \right) & 1 \\ \frac{k_2}{m_2} & 0 & -\left(\frac{k_1}{m_1} + \frac{k_1}{m_2} + \frac{k_2}{m_2} \right) & 0 \end{bmatrix}, \quad B = \begin{bmatrix} 0 & 0 \\ \frac{1}{m_1} & \frac{b_1 b_2}{m_1 m_2} \\ 0 & -\frac{b_2}{m_2} \\ \left(\frac{1}{m_1} + \frac{1}{m_2} \right) & -\frac{k_2}{m_2} \end{bmatrix}$$

$$C = \begin{bmatrix} 0 & 0 & 1 & 0 \end{bmatrix}$$

Numerikus értékek busz esetén SI egységekben:

$$m_1 = 2500 \quad b_1 = 350 \quad k_1 = 80000$$

$$m_2 = 320 \quad b_2 = 15020 \quad k_2 = 500000$$

A paraméterek értéktartományából következik, hogy az F beavatkozó jel (N) értéktartománya is nagy kell, hogy legyen, amelynek biztosítására hidraulikus segédenergiát valamint rugós előfeszítést célszerű alkalmazni.

Mérhető jelek:

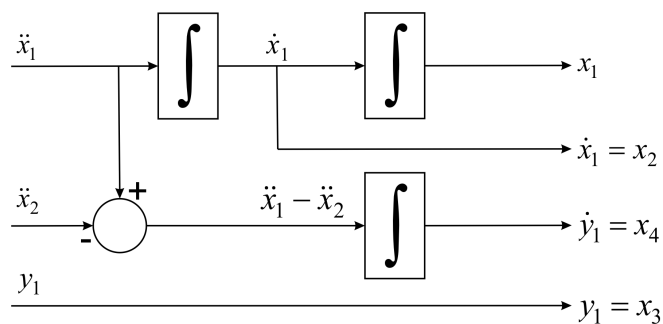
$$\ddot{x}_1, \ddot{x}_2, y_1 = x_1 - x_2$$

Elvárások a szabályozással szemben:

Az utazási komfort (kis $x_1 - x_2 = y_1 = y$) biztosításához 10sec-nál kisebb szabályozási idő és 5%-nál kisebb túllövés biztosítása szükséges. Az utazási komfortot a tervezés számára úgy specifikáljuk, hogy ha a busz felfut az útegyenetlenségek miatt egy 10cm-es lépcsőre (azaz $w(t) = 0.1 \cdot 1(t)$ [m] zavaró jellemző hat a rendszerre), akkor az oszcilláció maradjon ± 5 mm között (azaz ± 0.005 m között), és az oszcilláció szűnjön meg 5sec-on belül.

Az állapotok előállítása a mérhető jelekből:

Az állapotok a mérhető jelekből kiegészítő hálózattal képezhetők, ezért az állapot-visszacsatolás realizálásakor mérhető állapotokat feltételezünk, emiatt nincs szükség állapotbecslőre. A kiegészítő hálózatot az 5.32. ábra mutatja be. Ebben elhanyagoltuk, hogy az érzékelők tipikusan additív zajjal terheltek, amelyek várható értéke eltérhet nullától (bias). A kiegészítő hálózat a szabályozó része.



5.32. ábra. Az állapotok előállítása a mérhető jelekből

Mivel a zavarás nem mérhető, és a szigorú specifikációk mellett gyors becslése nehezen megoldható, ezért integrátort helyezünk el a rendszer kimenetén a maradó hiba elnyomására állandósult állapotban zavarás esetén, amelyet bevonunk az állapot-visszacsatolásba. Az integrátor a szabályozó része, hatását az állapotegyenletben az állapotváltozók bővítésével vesszük figyelembe. Az állapot-visszacsatolás természetesen csak u értékét tudja befolyásolni, ezért az állapot-visszacsatolás tervezésekor $\tilde{A} - \tilde{B} \cdot [1 \ 0]^T \tilde{K}$ sajátértékeit írjuk elő $\tilde{\varphi}_c(s)$ -ben, ahol az integrátorral bővített állapotvektor $\tilde{x} = (x^T \ x_I)^T$ és $x_I = \int y \, dt \Leftrightarrow \dot{x}_I = y = Cx$. Az integrátorral kiegészített állapotegyenlet mátrixai az állapot-visszacsatolás tervezéséhez a következők:

$$\tilde{A} = \begin{bmatrix} A & 0 \\ C & 0 \end{bmatrix}, \quad \tilde{B} = \begin{bmatrix} B \\ 0 \end{bmatrix}.$$

A pólusáthelyezéshez $\tilde{\varphi}_c(s)$ -ben 5 sajátértéket írhatunk elő. Ebből az egyik a tisztán valósnak választott $s_{dom} = -1/T_{dom}$ domináns pólus lesz, melynek időállandóját úgy választjuk meg, hogy az exponenciális lecsengés befejeződjék a specifikációban szereplő 5 sec időpont feléig $\Rightarrow T_{dom} = 0.5 \text{ sec} \Rightarrow p_{dom} = -1/T_{dom} = -2$.

Meghatározzuk az integrátor nélküli szakasz P/Z eloszlását és csillapítását, és a gyengén csillapított lassúbb pólusokat pólus/zérus kiejtésbe kényszerítjük, míg a gyorsabb maradék pólusok csillapítását megtartjuk, de csillapítatlan sajátfrekvenciáját jelentősen felgyorsítjuk. A szükséges értékek meghatározható a 3. gyakorlat során már használt programrészlettel:

```
%Suspension

clear all
close all
clc

%-----
%1/4 Bus Active Suspension
%See Section III.5 and Fig. III.5
%-----

%Parameters
m1=2500;
m2=320;
k1=80000;
k2=500000;
b1=350;
```

```

b2=15020;

%State Equation
%state x=(x1,dx1,y1,dy1)' where y1=x1-x2 (see comfort)
%input (u,w)' where u: active suspension force, w: road
disturbance
%output y=y1=x1-x2 (see comfort)
A=[0 1 0 0;...
   -b1*b2/(m1*m2) 0 b1/m1*(b1/m1+b1/m2+b2/m2)-k1/m1 -
b1/m1;...
   b2/m2 0 -(b1/m1+b1/m2+b2/m2) 1;...
   k2/m2 0 -(k1/m1+k1/m2+k2/m2) 0];
B=[0 0;...
   1/m1 b1*b2/(m1*m2);
   0 -b2/m2;
   (1/m1+1/m2) -k2/m2];
C=[0 0 1 0];
D=0;
sys=ss(A,B,C,D);

%Transfer functions
W=tf(sys); %Full Transfer Function
Wyu=W(1,1); %Transfer function u->y
Wyw=W(1,2); %Transfer function w->y
numyu=Wyu.num{1};
denyu=Wyu.den{1};
numyw=Wyw.num{1};
denyw=Wyw.den{1};

%Wyu Z/P distribution
[z,p,k]=zpkdata(Wyu,'v');
damp(z)
damp(p)
k

```

Az eredmények felhasználásával a zárt rendszer p_c sajátértékeit a következőkre választottuk:

Eigenvalue	Damping	Freq. (rad/s)
-2.00e+000	1.00e+000	2.00e+000
-2.66e+000 - 1.30e+001i	2.00e-001	1.33e+001
-2.66e+000 + 1.30e+001i	2.00e-001	1.33e+001
-7.04e+002 - 1.03e+003i	5.64e-001	1.25e+003
-7.04e+002 + 1.03e+003i	5.64e-001	1.25e+003

Határozzuk meg ezután $\tilde{A}$, $\tilde{B}$, $\tilde{C}$, $\tilde{K}$ értékét következő programrészlettel:

```

%Computation of the state feedback
At=[A zeros(4,1); C 0];

```

```

Bt=[B;0 0];
Ct=[C 0];
Dt=[0 0];
syst=ss(At,Bt,Ct,Dt);
%Attention: series(sys,sysI) makes xI to 1st component!

```

Az állapot-visszacsatolást az Ackermann-képlettel határozzuk meg:

```

%State feedback computation
Kt=acker(At,Bt*[1 0]',p_c);
%Kt=place(At,Bt*[1 0]',p_c); %only for different poles

```

A rendszer tipikusan nulla alapjel mellett működik, a szabályozás feladata a zavarások kompenzálása.

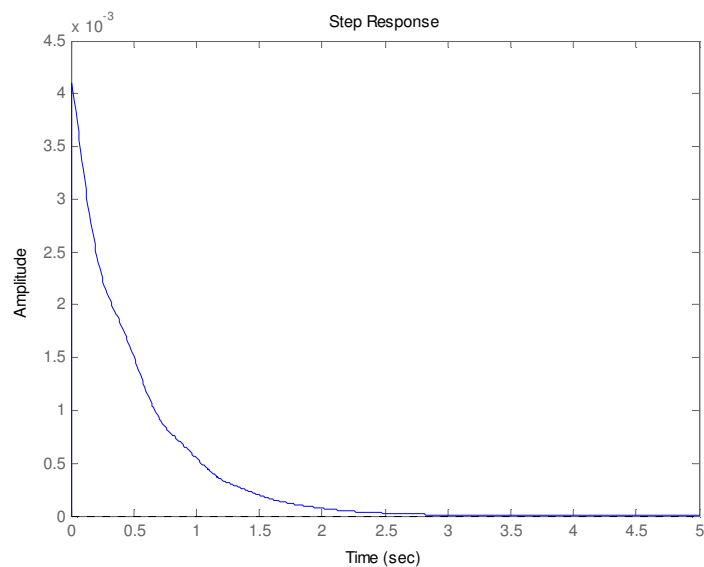
A zárt rendszer tranziense meghatározható a következő utasítással:

```

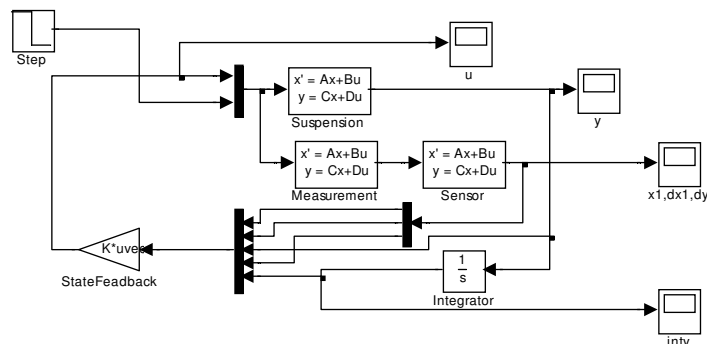
%Closed loop transient computation for disturbance jump
figure(1)
t=0:0.01:5;
step(At-Bt*[1 0]'*Kt,-0.1*Bt,Ct,-0.1*Dt,2,t)

```

Látható az 5.33. ábrából, hogy az állapot-visszacsatoláson alapuló integráló szabályozás teljesíti a specifikációkat.



5.33. ábra. A zárt rendszer tranziense zavarójel ugrás esetén állapot-visszacsatoláson alapuló integráló szabályozással



5.34. ábra. A zárt szabályozási kör Simulink modellje

A zárt szabályozási kör Simulink modelljét az 5.34. ábra mutatja be. A Suspension tag a felfüggesztés állapotegyenlete (A, B, C, D) , amely nincs kiegészítve a kimenőjel integrálásával.

A Measurement tag az érzékelőt szimulálja $\ddot{x}_1$ és $\ddot{x}_2 = -(\ddot{x}_1 - \ddot{x}_2) + \ddot{x}_1 = -\ddot{y} + \ddot{x}_1$ alapján, ahol az első a Suspension második állapotegyenletének jobb oldala, a második pedig a második és a negyedik állapotegyenlet jobb oldalainak különbsége, lásd A_m, B_m, C_m, D_m mátrixok a későbbi programban. A kimenőjeleket $ddx1$, $ddx2$ jelöli.

A Sensor tag az x_1 , $\dot{x}_1$, $\dot{y}$ jeleket állítja elő integrátorok bevonásával az

$$\dot{x}_1 = dx1, \quad \dot{dx1} = u_1 = \ddot{x}_1, \quad \dot{dy1} = u_1 - u_2 = \ddot{x}_1 - \ddot{x}_2$$

összefüggések alapján, lásd A_s, B_s, C_s, D_s mátrixok a későbbi programban. A kimenőjeleket x_1 , $\dot{x}_1 = dx1$, $\dot{y} = dy1$ adja, melyekhez még hozzá-multiplexeljük az y kimenőjel értékét, valamint a kimenőjel integrálját, a melyet külön állítunk elő. A sorrend $(x_1, \dot{x}_1, y, \dot{y}, \int y dt)^T$ lesz, ahogy ez szükséges az állapot-visszacsatoláshoz, amelynek mátrixa K_t .

Melléklet – Matlab kód az aktív kerékfelfüggesztés irányításának vizsgálatához állapot-visszacsatolással

```
%SuspensionControl.m
%Suspension control using state feedback and
measured/estimated state
```

```

clear all
close all
clc

%-----
%1/4 Bus Active Suspension
%See Section III.5 and Fig. III.5
%-----

%Parameters
m1=2500;
m2=320;
k1=80000;
k2=500000;
b1=350;
b2=15020;

%State Equation
%state x=(x1,dx1,y1,dy1)' where y1=x1-x2 (see comfort)
%input (u,w)' where u: active suspension force, w: road
disturbance
%output y=y1=x1-x2 (see comfort)
A=[0 1 0 0;...
   -b1*b2/(m1*m2) 0 b1/m1*(b1/m1+b1/m2+b2/m2)-k1/m1 -
   b1/m1;...
   b2/m2 0 -(b1/m1+b1/m2+b2/m2) 1;...
   k2/m2 0 -(k1/m1+k1/m2+k2/m2) 0];
B=[0 0;...
   1/m1 b1*b2/(m1*m2);
   0 -b2/m2;
   (1/m1+1/m2) -k2/m2];
C=[0 0 1 0];
D=[0 0];
sys=ss(A,B,C,D);
sys.InputName={'u','w'};
sys.OutputName={'y'};

%Transfer functions
W=tf(sys); %Full Transfer Function
Wyu=W(1,1); %Transfer function u->y
Wyw=W(1,2); %Transfer function w->y
numyu=Wyu.num{1};
denyu=Wyu.den{1};
numyw=Wyw.num{1};
denyw=Wyw.den{1};

%Step response for disturbance input
figure(1)
step(0.1*Wyw);
title('Step Response for w=0.1 m without Control')

```

```

%Wyu Z/P distribution
[z,p,k]=zpkdata(Wyu,'v')
damp(z)
damp(p)
k

%Choice of closed loop poles
p1=-2 %-1.580859552
xi2=0.2 %2.25e-001
w2=13.3 %1.31e+001
xi4=0.563 %1 %3.59e-001
w4=1.25e3 %2e3 %1.34e+003
p2=-xi2*w2+i*sqrt(1-xi2^2)*w2
p3=conj(p2)
p4=-xi4*w4+i*sqrt(1-xi4^2)*w4
p5=conj(p4)
p_c=[p1 p2 p3 p4 p5]'
%phit_c=poly(p_c)

%Computation of the state feedback
At=[A zeros(4,1); C 0];
Bt=[B;0 0];
Ct=[C 0; 0 0 0 0 1];
Dt=[0 0; 0 0];
syst=ss(At,Bt,Ct,Dt);
%Attention: series(sys,sysI) would put xI to be the first
component

%State feedback computation
Kt=acker(At,Bt*[1 0]',p_c);
%Kt=place(At,Bt*[1 0]',p_c); %only for different poles

%Closed loop transient computation for disturbance jump
figure(2)
t=0:0.01:5;
step(At-Bt*[1 0]'*Kt,-0.1*Bt,Ct,-0.1*Dt,2,t)
title('Closed loop step response (y, inty) for disturbance
jump')

%Simulate measurement system
Am=A;
Bm=B;
Cm=[A(2,:); -A(4,:)+A(2,:)]; %ddx1, ddx2
Dm=[B(2,:); -B(4,:)+B(2,:)];
sysm=ss(Am,Bm,Cm,Dm);
sysm.InputName={'u','w'};
sysm.OutputName={'ddx1','ddx2'};

%Simulate state estimator
As=[0 1 0; 0 0 0; 0 0 0];

```

```
Bs=[0 0; 1 0; 1 -1];  
Cs=[1 0 0; 0 1 0; 0 0 1];  
Ds=[0 0; 0 0; 0 0];  
syss=ss(As,Bs,Cs,Ds);  
syss.InputName={'ddx1','ddx2'};  
syss.OutputName={'x1','dx1','dy'};  
  
%Simulate sensor outputs  
open('SuspContSys.mdl')  
fprintf('SuspContSys.mdl has been opened\n')  
fprintf('Start simulation\n')
```

5. Számítógéptermi gyakorlat: Állapot-visszacsatolás és megfigyelő tervezése folytonos időben

A szabályozott szakasz leírása nemcsak átviteli függvény segítségével, hanem állapotterres leírással is történhet. Az állapotterben megadott modell információban gazdagabb (a nem irányítható és nem megfigyelhető alrendszerrel is képet kapunk), ezért a szabályozótervezés sok esetben az állapotterres leíráson alapul.

Az állapotterben végrehajtott szabályozótervezések alapja az állapot-visszacsatolás, ezért a gyakorlat során is ezt sajátítjuk el először. Az eddigi soros kompenzálásoktól eltérően, nem a szakasz kimenetét, hanem annak állapotát csatoljuk vissza a bemenetre.

Sajnos sok praktikus alkalmazásban az állapotok nem mérhetők, vagy az érzékelők beépítése jelentős járulékos költséget jelentenek. A probléma megoldására megfigyelőt alkalmazunk, amely a szakasz bemenő és kimenő jeleinek mérése alapján a szakasz állapotait megbecsüli. Az állapot-visszacsatolás ezt a becsült állapotot használja fel a valódi állapot helyett. Ebből is látható, hogy ismeretlen állapotok esetén egy jó megfigyelő tervezése elengedhetetlen.

Az állapot-visszacsatolás önmagában csak a zárt kör dinamikáját határozza meg (azaz, hogy milyen tranzienseken keresztül érjük el az állandósult állapotot), de a kívánt statikus működést külön kell biztosítani az alapjel megfelelő bevezetésével. Ez a gyakorlat során speciális N_x , N_u mátrixok segítségével történik.

Az állapot-visszacsatolás bővítésével lehetőség van arra, hogy a szakasz bemenetén megjelenő egységugrás alakú zavarjel (terhelés) kimenetre vett kártékony hatását (egy tranziens lefutása után) elimináljuk. A módszer a terhelésbecslő tervezésén alapul, amelyet a gyakorlat végén tárgyalunk.

A korábbiakhoz hasonlóan legyen a szabályozott szakasz átviteli függvénye:

$$W_p(s) = \frac{A}{(1+sT_1)(1+sT_2)(1+sT_3)}$$

$$\begin{aligned} A &= 2 \\ T_1 &= 10 \text{ sec} \\ T_2 &= 4 \text{ sec} \\ T_3 &= 1 \text{ sec} \end{aligned}$$

Matlab környezetben:

```
Aplant=2; T1=10; T2=4; T3=1;
numps=Aplant;
denps=conv(conv([T1 1],[T2 1]),[T3 1]);

sysp_tf=tf(numps,denps)
```

Az állapotterbeli szabályozótervezéshez szükség lesz a szakasz állapotteres leírására is:

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

$$y = Cx + Du$$

ami Matlab környezetben a

```
sysp_ss=ss(sysp_tf)
A=sysp_ss.a;
B=sysp_ss.b;
C=sysp_ss.c;
D=sysp_ss.d;
```

utasításokkal állítható elő.

Állapot-visszacsatolás tervezése

Feladat: Tervezzon szabályozót $W_p(s)$ átviteli függvénnyel rendelkező rendszerre, ahol a zárt kör egységugrás válaszát a

$$\xi = 0.7$$

$$\omega_0 = 1$$

csillapítással és csillapítatlan sajátfrekvenciával rendelkező domináns póluspár határozza meg. A zárt rendszer minden más pólusa legyen

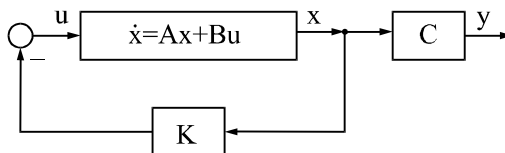
$$s_{c\infty} = -3\omega_0\xi$$

Megoldás:

Állapot-visszacsatolás esetén a beavatkozó jel

$$u = -Kx$$

alakú lesz (5.35. ábra).



5.35. ábra. Állapot-visszacsatolás blokk-sémája

Amennyiben előírjuk a zárt rendszer állapotmátrixának sajátértékeit (pólusait), azaz a zárt rendszer

$$\varphi_{closed}(s) = \det(sI - (A - BK)) = 0$$

karakterisztikus egyenletének gyökeit, akkor egyúttal előírjuk a zárt kör tranzienseinek lecsengését (a zárt kör dinamikáját). A feladat specifikációjának ismeretében a zárt rendszer előírt pólusai:

$$s_{1,2} = -\omega_0 \xi \pm j\omega_0 \sqrt{1 - \xi^2}$$

$$s_3 = s_{c\infty} = -3\omega_0 \xi$$

ami Matlab utasítással:

```
>> sc = [-w0*ksi+j* w0*sqrt(1-ksi^2)
-w0*ksi-j* w0*sqrt(1-ksi^2)
-3*w0*ksi]
```

sc =

```
-0.7000 - 0.7141i
-0.7000 + 0.7141i
-2.1000
```

azaz az előírt sajátértékeket az sc vektorban tároljuk. Az előírt sajátértékek (dinamika) ismeretében a szabályozáshoz szükséges K állapotvisszacsatolás a

$$K = (0 \quad \dots \quad 0 \quad 1)M_c^{-1}\varphi_{closed}(A) \qquad M_c = \begin{bmatrix} B & AB & \dots & A^{n-1}B \end{bmatrix}$$

Ackermann-képlettel meghatározható. Az Ackermann-képletet rövid jelölése a továbbiakban:

$$(A, B) \xrightarrow[M_c]{\varphi_{closed}(s)} K.$$

A Control System Toolbox `acker` szolgáltatásával a megoldás egy függvényhívással meghatározható, anélkül, hogy az M_c irányíthatósági mátrixot kiszámítanánk:

```
>> K=acker(A,B,sc)
```

K =

```
4.3000    7.1300   33.2000
```

azaz a megoldás:

$$K = [4.3 \quad 7.13 \quad 33.2]$$

Az `acker` függvénynek az állapotterez leírás A, B mátrixain kívül egy sc oszlopvektorban meg kell adni a kívánt zárt rendszer állapotmátrixának sajátértékeit. Mellékesen jegyezzük meg, hogy a fenti utasítás könnyen kiváltható hagyományos Matlab utasításokkal is:

```
>> Mc = ctrb(A,B);
>> K = [0 0 1]*inv(Mc)*polyvalm(poly(sc),A)
```

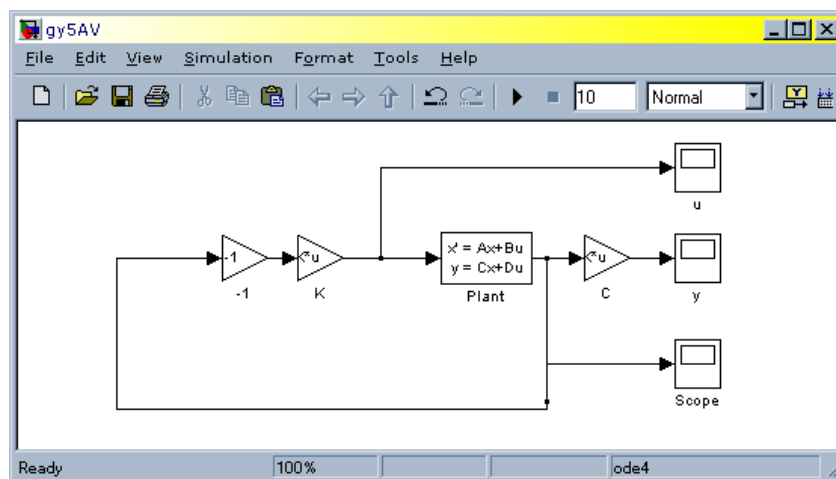
K =

```
4.3000    7.1300   33.2000
```

Ellenőrzés: A zárt szabályozási kör kimenetei és állapotai legegyszerűbben Simulinkben szimulálhatók. A Simulink használatát az is indokolja, hogy a továbbiakban a szabályozási kör további blokkokkal egészül ki, amelyek összekapcsolását legegyszerűbben Simulinkben oldhatjuk meg. A Matlab

```
>>simulink
```

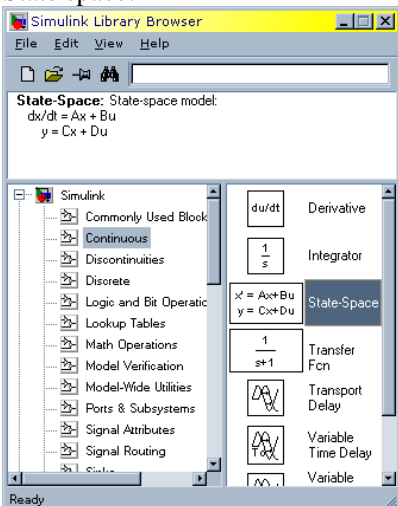
hívása után építsük meg az 5.36. ábrán látható szabályozási kört



5.36. ábra. Állapot-visszacsatolás blokkdiagramja Simulinkben

A szabályozási kör az 5.37. ábrán bemutatott elemekből és paraméter-beállítások mellett építhető fel (baloldalon láthatjuk a blokk megtalálási helyét, jobboldalon a blokk paramétereinek beállítását)

State space:



Simulink Library Browser

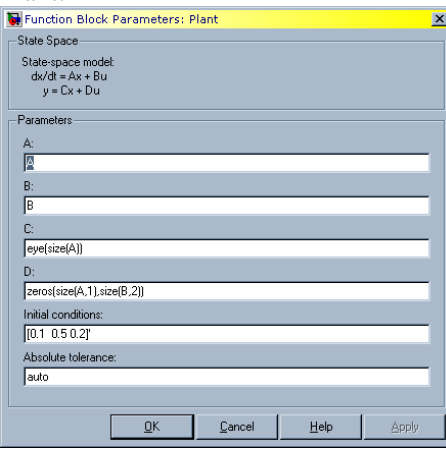
State-Space: State-space model:
 $\dot{x} = Ax + Bu$
 $y = Cx + Du$

Math Operations

State-Space

Ready

Plant:



Function Block Parameters: Plant

State Space

State-space model:
 $\dot{x} = Ax + Bu$
 $y = Cx + Du$

Parameters:

A: [A]

B: [B]

C: [eye(size(A))]

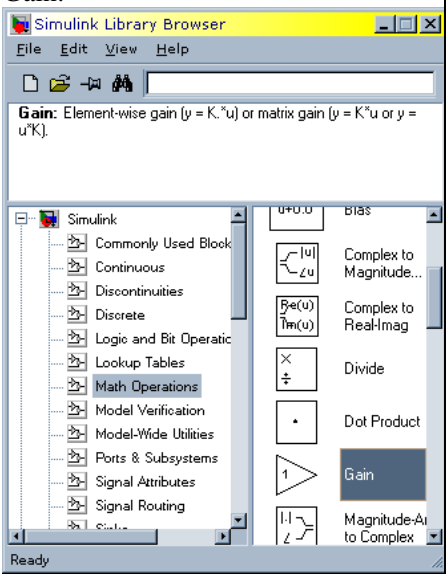
D: [zeros(size(A,1),size(B,2))]

Initial conditions: [0.1 0.5 0.2]

Absolute tolerance: auto

OK Cancel Help Apply

Gain:



Simulink Library Browser

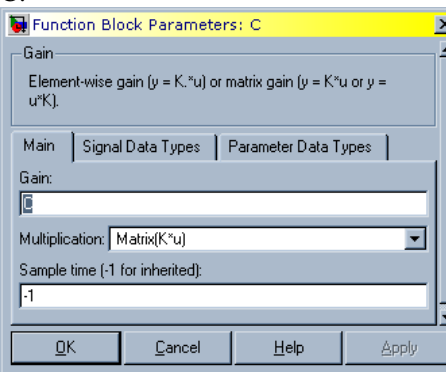
Gain: Element-wise gain ($y = K \cdot u$) or matrix gain ($y = K \cdot u$ or $y = u \cdot K$).

Math Operations

Gain

Ready

C:



Function Block Parameters: C

Gain

Element-wise gain ($y = K \cdot u$) or matrix gain ($y = K \cdot u$ or $y = u \cdot K$).

Main Signal Data Types Parameter Data Types

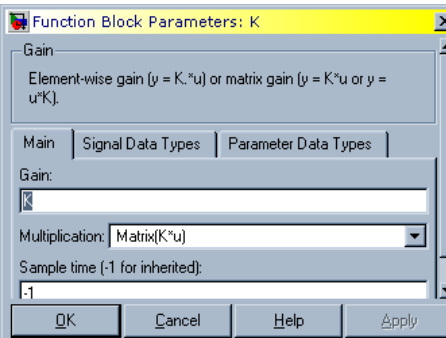
Gain: [C]

Multiplication: Matrix($K \cdot u$)

Sample time (-1 for inherited): -1

OK Cancel Help Apply

K:



Function Block Parameters: K

Gain

Element-wise gain ($y = K \cdot u$) or matrix gain ($y = K \cdot u$ or $y = u \cdot K$).

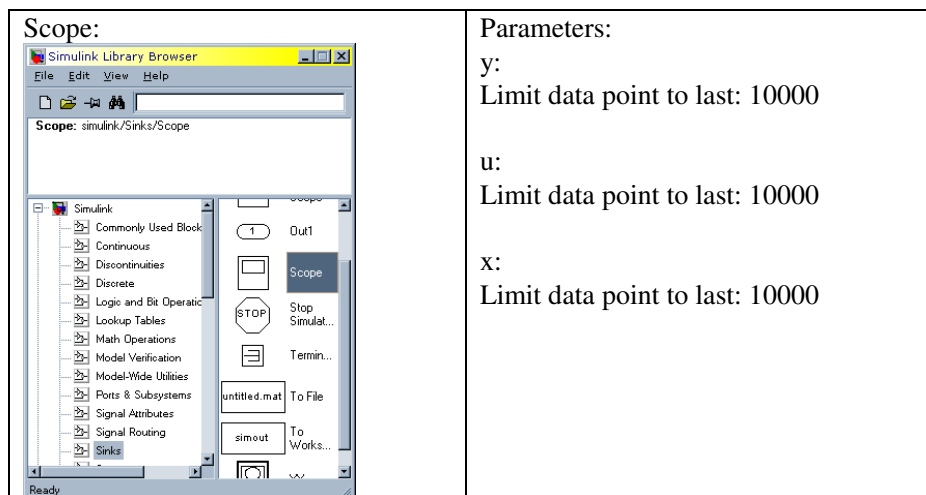
Main Signal Data Types Parameter Data Types

Gain: [K]

Multiplication: Matrix($K \cdot u$)

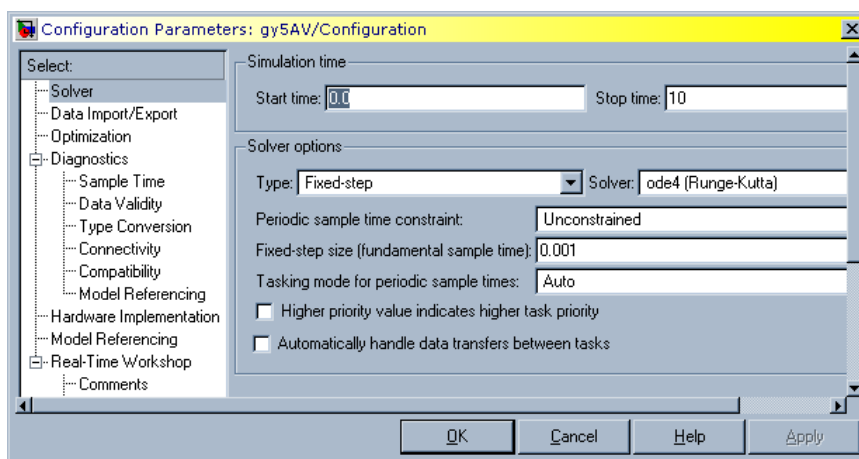
Sample time (-1 for inherited): -1

OK Cancel Help Apply



5.37. ábra. Az állapot-visszacsatolást megvalósító Simulink modell blokkjai

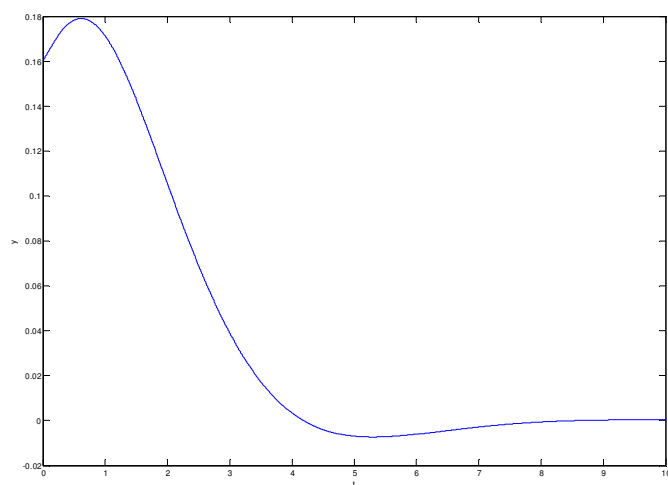
Ügyeljünk arra, hogy az erősítések a (negatív visszacsatolást megvalósító erősítés kivételével) mátrixszorzást valósítanak meg. A state-space blokkban a C mátrix értékét egységmátrixnak választottuk annak érdekében, hogy lehetőség legyen az állapotok oszcilloszkópra történő kivezetésére. Az állapotokból ezután egy külön C mátrixszorzással határozzuk meg a kimenetet. Ez a C mátrix eredetileg a rendszer része! Ezenkívül be kell állítani a Simulinknél a differenciálegyenlet megoldás paramétereit (5.38. ábra). Ez a Simulink/Configuration Parameters menüpontban történik:



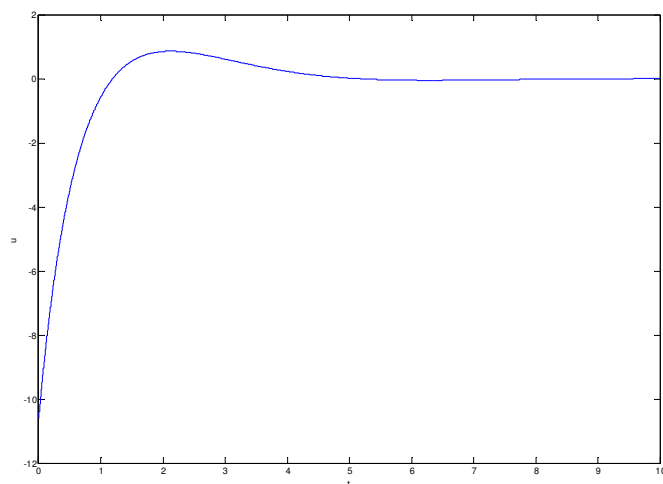
5.38. ábra. A szimuláció konfigurációs paraméterei.

A szimuláció idejét 10 másodpercben határoztuk meg, ami lehetővé teszi a teljes tranziens megjelenítését. Ehhez azonban az 5.36. ábrának megfelelően az oszcilloszkópok 10000 szimulációs pont tárolását kell, hogy elvégezzék.

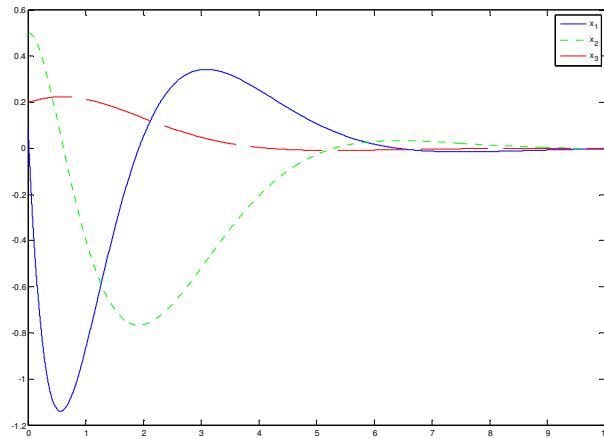
A szimuláció lefuttatása (és az oszcilloszkópok képeinek automatikus skálázása) után a szabályozási kör kimenő jele az 5.39. ábrán, a beavatkozó jel, az 5.40. ábrán, az állapotok az 5.41. ábrán láthatók.



5.39. ábra. Kimenő jel állapot-visszacsatolás esetén.



5.40. ábra. A beavatkozó jel állapot-visszacsatolás esetén.



5.41. ábra. Az állapotok alakulása állapot-visszacsatolás esetén.

Az elvárásnak megfelelően látható, hogy a kimenő jel és az állapotok a nemnulla kezdeti állapotokból az állapot-visszacsatolás hatására exponenciálisan konvergálnak a nullához.

Állapotmegfigyelő tervezése

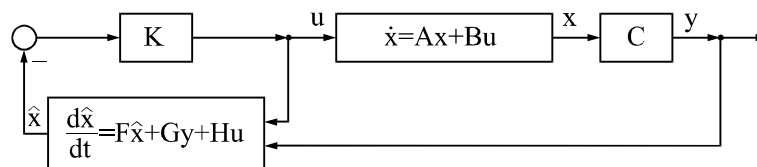
Feladat: A folyamat technológiájából adódóan nem állnak rendelkezésre a $W_p(s)$ folyamat állapotai. A feladat most a korábbi specifikációk mellett egy olyan állapot-visszacsatolás tervezése, amely a rendszer állapotai helyett annak becslését használja fel az $u = -K\hat{x}$ irányítási törvényben ($\hat{x}$ a folyamat eredeti állapotainak becslése). Az állapotok becsléséhez tervezzen állapotmegfigyelőt realizáló

$$\frac{d\hat{x}}{dt} = F\hat{x} + G y + H u$$

dinamikus rendszert, amely

$$s_{o\infty} = -5\omega_0\xi$$

sajátértékekkel rendelkezik.



5.42. ábra. Állapot-visszacsatolást és állapotmegfigyelőt tartalmazó irányítás blokksémája

Megjegyzés: Vegyük észre, hogy a megfigyelő pólusai értelemszerűen gyorsabbak, mint a zárt rendszer előírt pólusai (azaz a megfigyelő sajátértékei negatívabbak, mint az előírt zárt kör sajátértékeinek valós része).

Megjegyzés: Vegyük észre, hogy amennyiben ismertek lennének a folyamat kezdeti állapotai, a pontosan ismert A, B, C, D mátrixok segítségével az állapotok alakulása az állapotegyenlet (numerikus) megoldásával nyomon követhető lenne és nem lenne szükség állapotmegfigyelőre. Ezt azonban nem feltételezhetjük, ezért az állapotmegfigyelő tervezése szükséges lépés. A szimuláció kedvéért legyen a gyakorlat során az állapotok ismeretlen kezdeti értéke: $x_1(0) = 0.1$, $x_2(0) = 0.5$, $x_3(0) = 0.2$.

Megoldás:

Az állapotmegfigyelőt úgy tervezzük meg az F, G, H mátrixok megfelelő beállításával, hogy a becslési hiba tranziensét az s_{∞} megfigyelő pólusok határozzák meg. Ismeretes, hogy az $\tilde{x}$ becslési hiba nullához konvergál a

$$\begin{aligned}\tilde{x} \rightarrow 0: \quad & F = A - G C \\ & H = B \\ & \frac{d\tilde{x}}{dt} = F \tilde{x}\end{aligned}$$

feltételek mellett. Látható, hogy H értéke közvetlenül meghatározható. Ezenkívül a kívánt hibatranziens akkor érhető el, ha F állapotmátrixának sajátértékei s_{∞} . Az F meghatározásához csak G ismeretére van szükség. Természetesen G értékét úgy kell meghatározni, hogy F állapotmátrix sajátértékei az előírt s_{∞} pólusok legyenek. A feladat algebrailag visszavezethető az Ackermann-képletre:

$$(A, C)_I \leftrightarrow (A^T, C^T)_{II} \xrightarrow{\varphi_{obs}(s)} K_{II} \rightarrow G = K_{II}^T \rightarrow F = A - GC$$

$$M_{cII} = M_{oI}^T$$

A fentiek ismeretében a megfigyelő tervezését az alábbi Matlab utasításokkal végezhetjük el:

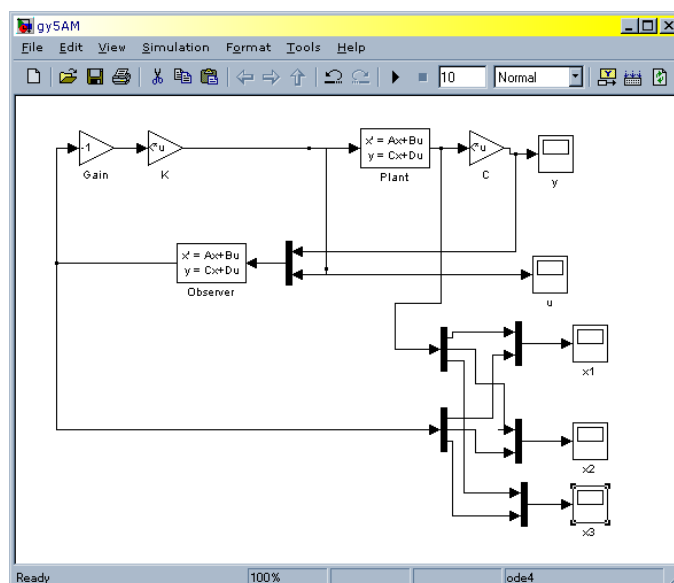
```
% Megfigyelo eloirt polusai
soinf=-5*w0*ksi;
%Megfigyelotervezés
Gt=acker(A', C', soinf*ones(1, length(A)))
G=Gt'
F=A-G*C
H=B
```

Ügyeljünk arra, hogy az `acker` függvény által visszaadott G_t mátrixot (jelenleg sorvektort) az elméletnek megfelelően transzponálni kell ($G=G_t'$). Látható, hogy a megfigyelő pólusait az előírt $s_{\infty} = -5\omega_0\xi$ testesíti meg (többszörös multiplicitással). Az utasításokat végrehajtva a megfigyelő paraméterei az alábbiak:

$$F = \begin{bmatrix} -1.35 & -0.375 & -56.107 \\ 1 & 0 & -192.18 \\ 0 & 0.125 & -9.15 \end{bmatrix} \quad G = \begin{bmatrix} 69.884 \\ 240.225 \\ 11.438 \end{bmatrix} \quad H = \begin{bmatrix} 0.5 \\ 0 \\ 0 \end{bmatrix}$$

Ellenőrzés:

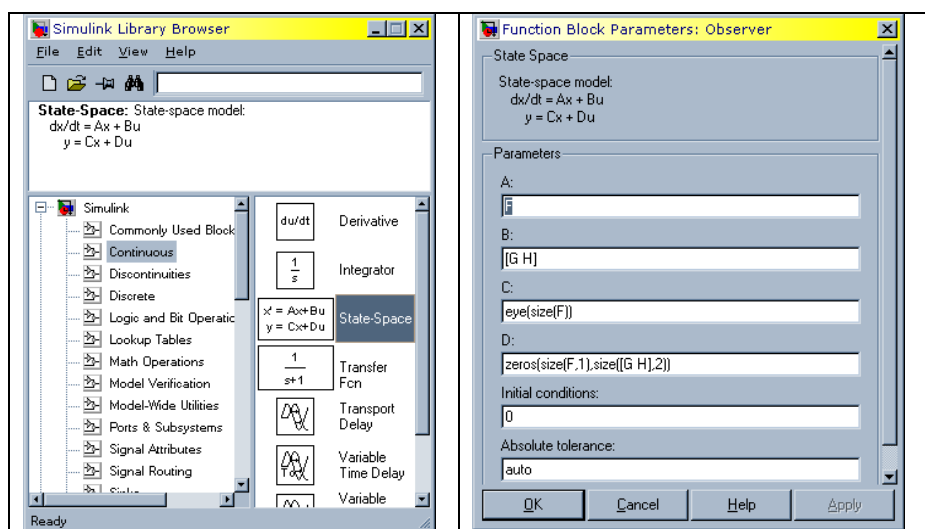
Az eredmények verifikálásához módosítanunk kell a korábbi Simulink modellt az 5.42. ábrának megfelelően. A Simulink modellben az egyetlen új elem a megfigyelő, amelynek beállításait az 5.43. ábra illusztrálja.



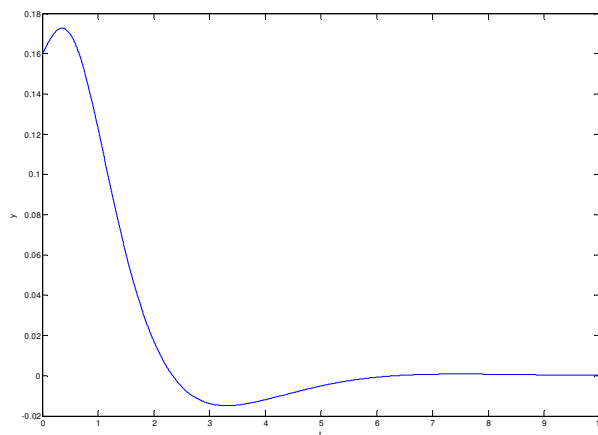
5.43. ábra. Az állapot-visszacsatolást és állapotmegfigyelőt tartalmazó Simulink modell.

Ügyeljünk az állapotteres modell megadáskor arra, hogy a megfigyelőnek két bemenete van, ezért a B mátrixának is két oszlopa lesz. A megfigyelő kimeneti C mátrixa egységmátrix, hisz minden becsült állapotra szükségünk van az állapot-visszacsatoláshoz (és az oszcilloszkópon való megjelenítéshez). Az 5.45. ábra alapján beláthatjuk, hogy a kimenő jel a tranziensek lecsengése után a nullához konvergál, ami várható volt. Az 5.46. ábra az ehhez tartozó beavatkozó jelet mutatja. A megfigyelő működését az 5.47. ábra, az 5.48. ábra és az 5.49. ábra illusztrálja. Az ábrákon a folyamat állapotait láthatjuk, a hozzájuk tartozó és a

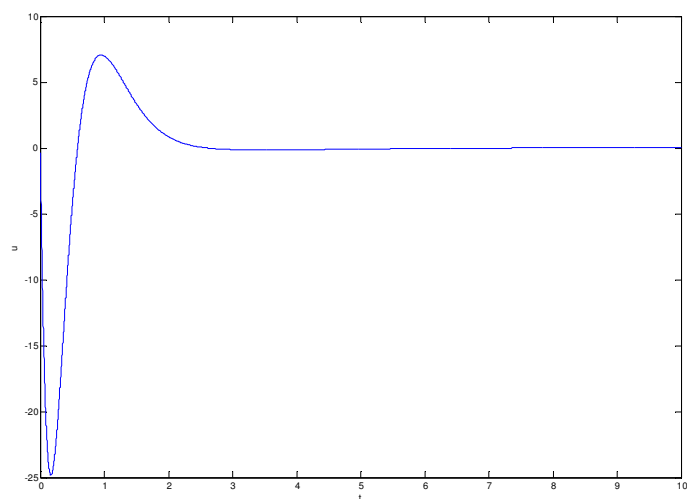
megfigyelő által becsült állapotokkal. Az ábrák alapján levonhatjuk azt a következtetést, hogy a megfigyelt állapotok még a zárt kör tranziensének lecsengése előtt rásimulnak a valódi állapotokra. Ez annak köszönhető, hogy a megfigyelő dinamikáját gyorsabbra terveztük, mint a zárt körét. Erre szükség is van, hisz akkor tudunk állapot-visszacsatolással a szabályozási körbe hatékonyan beavatkozni, ha az állapotokat már megfelelő pontossággal ismerjük.



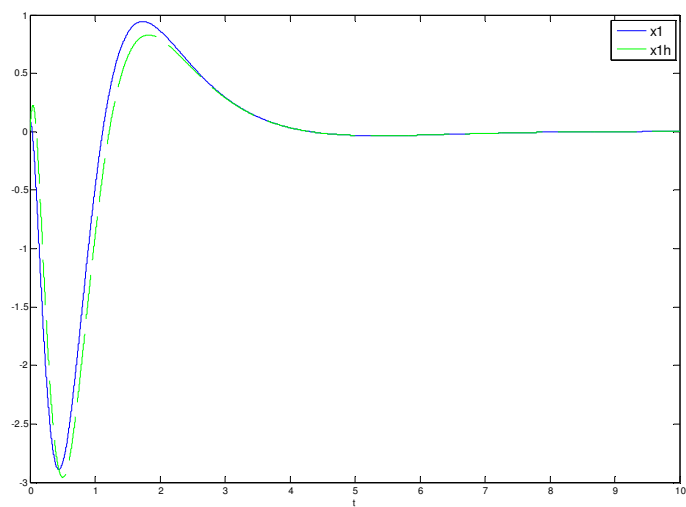
5.44. ábra. A megfigyelő beépítése a Simulink modellbe.



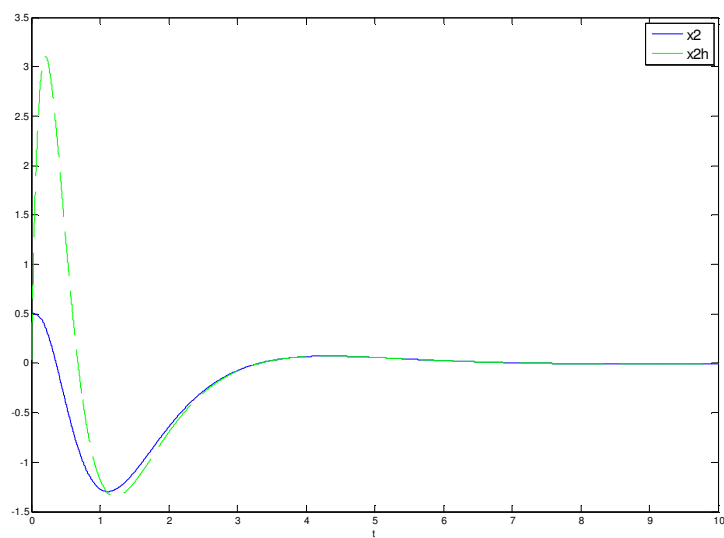
5.45. ábra. A kimenő jel állapot-visszacsatolást és megfigyelőt tartalmazó szabályozó esetén.



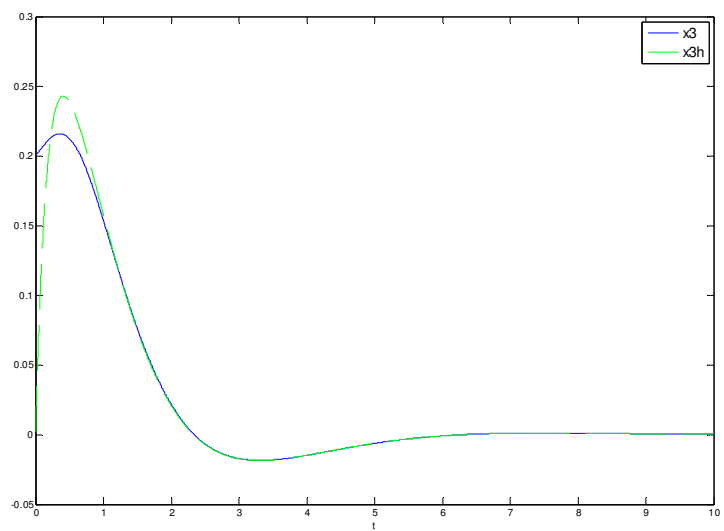
5.46. ábra. A beavatkozó jel állapot-visszacsatolást és megfigyelőt tartalmazó szabályozó esetén.



5.47. ábra. A valódi x_1 és a becsült $\hat{x}_1$ állapot alakulása állapot-visszacsatolást és megfigyelőt tartalmazó szabályozó esetén.



5.48. ábra. A valódi x_2 és a becsült $\hat{x}_2$ állapot alakulása állapot-visszacsatolást és megfigyelőt tartalmazó szabályozó esetén.

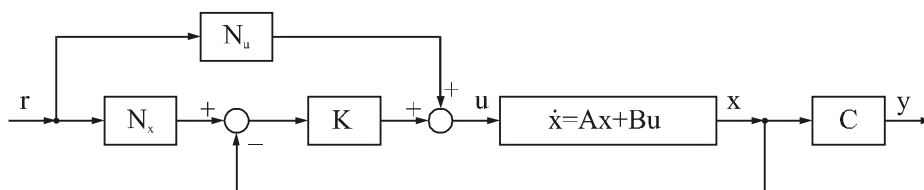


5.49. ábra. A valódi x_3 és a becsült $\hat{x}_3$ állapot alakulása állapot-visszacsatolást és megfigyelőt tartalmazó szabályozó esetén.

Egységugrás alapjel statikus követésének biztosítása

Feladat: Az eddigi szabályozás körhöz tervezzünk az 5.50. ábrának megfelelően olyan N_x, N_u rendszertechnikai blokkokat, amelyek biztosítják a belépő alapjel statikus követését.

Megjegyzés: A korábban elvégzett tervezési módszerek csak a zárt rendszer dinamikáját befolyásolták és alapjel nem jelent meg a rendszerben. Egy szabályozási körben azonban tipikus feladat egy egységugrás alakú alapjel követése. Ezzel foglalkozunk most a feladat keretében.



5.50. ábra. Egységugrás alakú alapjel követését biztosító szabályozási kör.

Megoldás: A koncepció lényege, hogy állandósult állapotban a K állapot-visszacsatoló mátrixon (SISO esetben sorvektor) keresztül nem jut jel a folyamat bemenetére, mert azt az N_x erősítésen keresztül érkező alapjel semlegesíti. A folyamat állandósult kimenetét biztosító beavatkozó jelet csak az N_u erősítésen keresztül érkező alapjel biztosítja. Elméleti levezetések után megoldásra a

$$\begin{pmatrix} N_x \\ N_u \end{pmatrix} = \begin{bmatrix} A & B \\ C & 0 \end{bmatrix}^{-1} \begin{pmatrix} 0 \\ I \end{pmatrix}$$

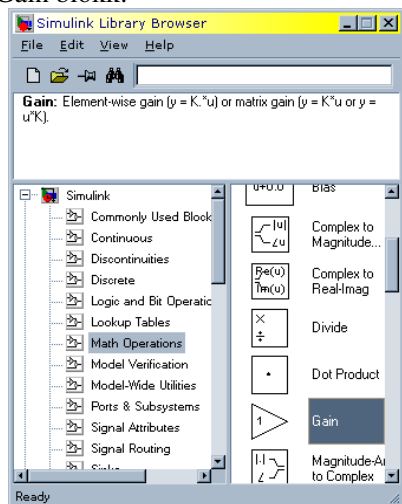
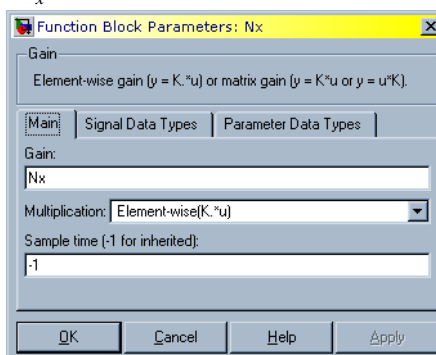
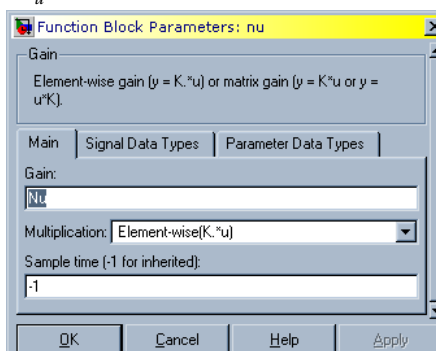
Kifejezést kapjuk. SISO esetben az I egységmátrix 1-re redukálódik. A megoldást Matlab környezetben is előállíthatjuk:

```
NxNu=inv([A B;C 0])* [zeros(size(B)); 1]
Nx=NxNu(1:length(A))
Nu=NxNu(length(A)+1)
```

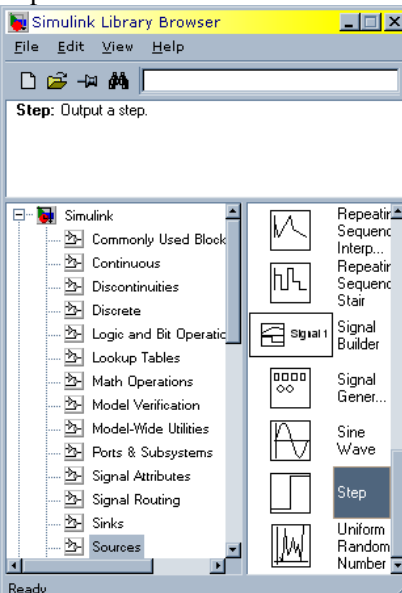
ami a következő eredményre vezet:

$$N_x = \begin{bmatrix} 0 \\ 0 \\ 1.25 \end{bmatrix} \quad N_u = 0.5$$

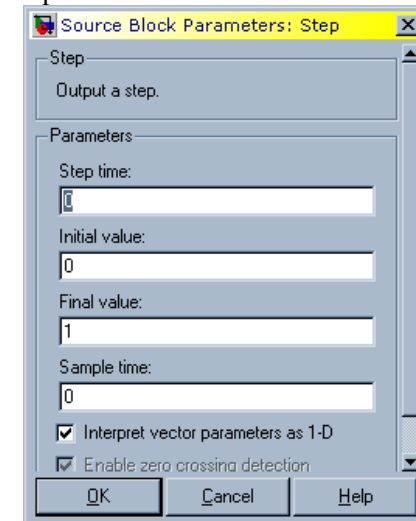
Gain blokk:

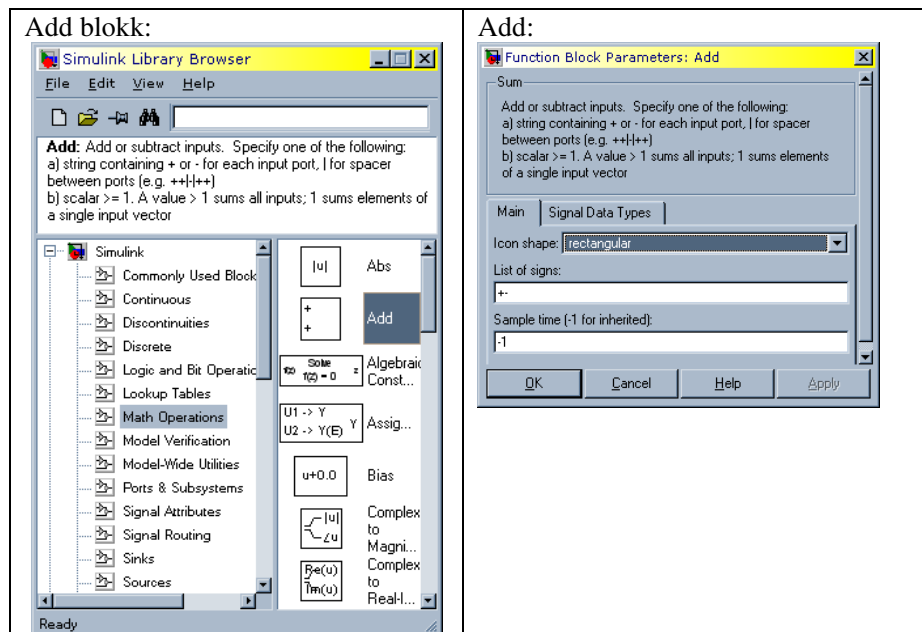
 N_x : N_u :

Step blokk:

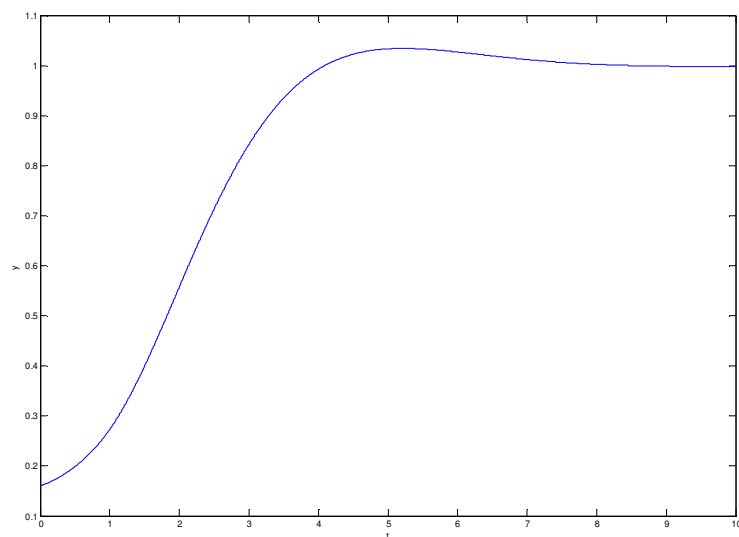


Step:

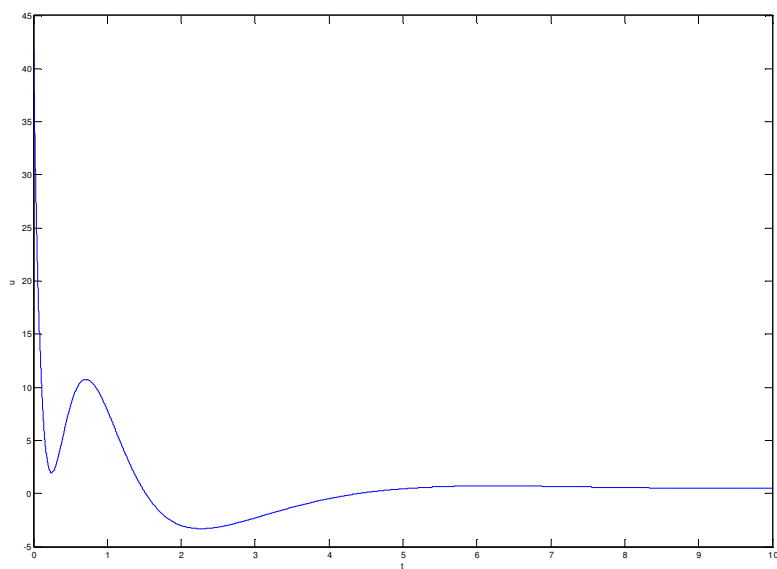




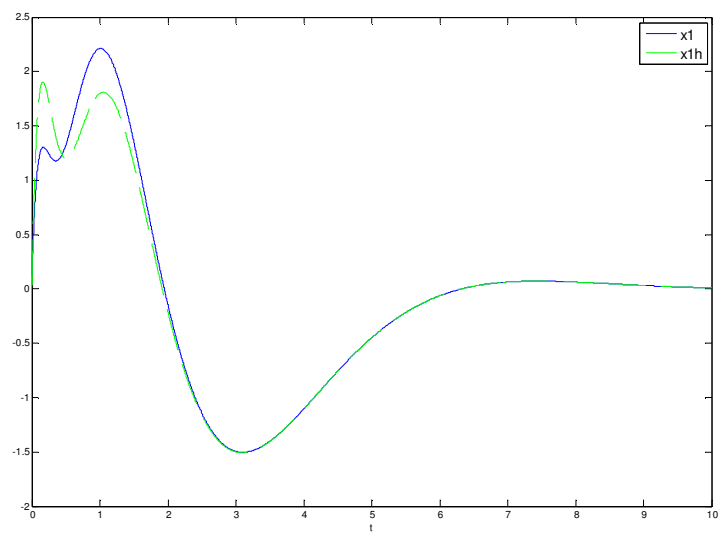
5.52. ábra. Új rendszertechnikai blokkok az egységugrás alapjel statikus követését biztosító szabályozás Simulink modelljében.



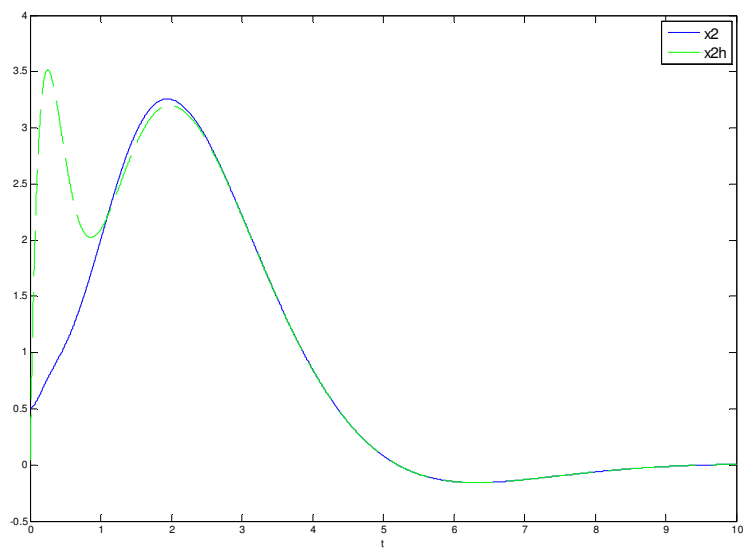
5.53. ábra. Kimenő jel egységugrás alapjel esetén N_x , N_u alkalmazása mellett.



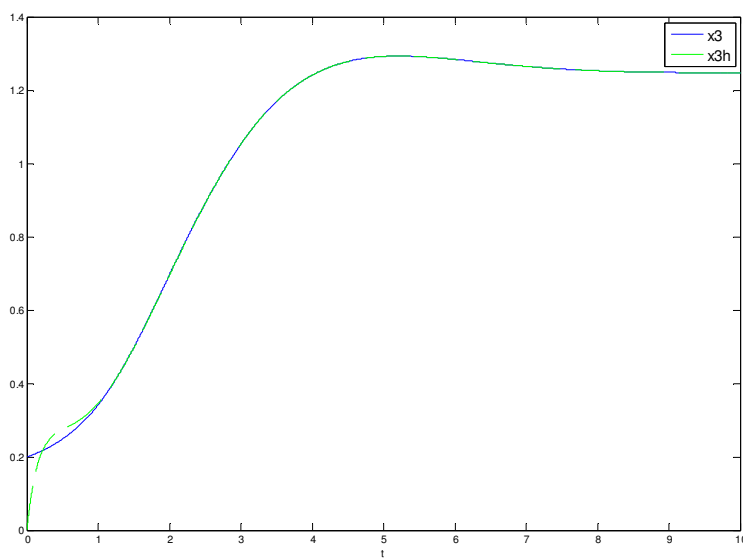
5.54. ábra. Beavatkozó jel egységugrás alapjel esetén N_x , N_u alkalmazása mellett.



5.55. ábra. Az x_1 állapot és annak $\hat{x}_1$ ($x1h$) becslése egységugrás alapjel esetén N_x , N_u alkalmazása mellett.



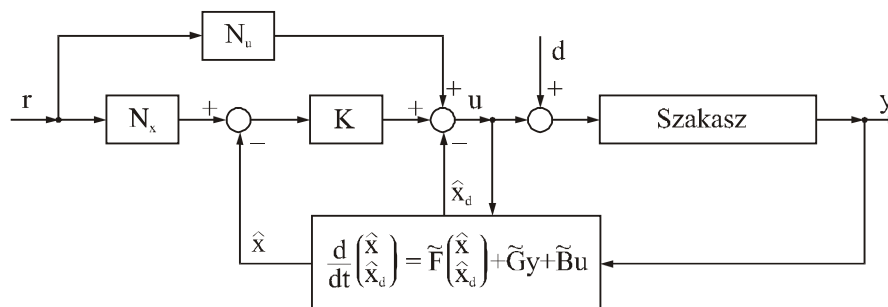
5.56. ábra. Az x_2 állapot és annak $\hat{x}_2$ ($x2h$) becslése egységugrás alapjel esetén N_x , N_u alkalmazása mellett.



5.57. ábra. Az x_3 állapot és annak $\hat{x}_3$ ($x3h$) becslése egységugrás alapjel esetén N_x , N_u alkalmazása mellett.

Terhelésbecslő tervezése állapot-visszacsatoláshoz

Feladat: Tervezzünk olyan állapotmegfigyelőt az állapot-visszacsatoláson alapuló szabályozáshoz, amely képes a bemeneten ható d értékű egységugrás alakú terhelés becslésére és annak kompenzálására az 5.58. ábrának megfelelően.



5.58. ábra. Állapot-visszacsatolás terhelésbecslésre alkalmas megfigyelővel.

Megoldás: A bemeneti terhelést a folyamat egy extra állapotának fogjuk fel, így lehetőség nyílik az állapotegyenletbe való integrálására. A $\dot{d} = 0$ (egységugrás alakú, de ismeretlen nagyságú terhelésnek) megfelelően a bővített rendszer állapotegyenlete:

$$\begin{pmatrix} \dot{x} \\ \dot{x}_d \end{pmatrix} = \begin{bmatrix} A & B \\ 0 & 0 \end{bmatrix} \begin{pmatrix} x \\ x_d \end{pmatrix} + \begin{bmatrix} B \\ 0 \end{bmatrix} u = \tilde{A} \begin{pmatrix} x \\ x_d \end{pmatrix} + \tilde{B} u$$

$$y = [C \quad 0] \begin{pmatrix} x \\ x_d \end{pmatrix} = \tilde{C} \begin{pmatrix} x \\ x_d \end{pmatrix}$$

Ezt a Matlabban könnyen megvalósíthatjuk:

```
Ah=[A B; zeros(1,size([A B],2))]  
Bh=[B;0]  
Ch=[C 0]
```

ahol az Ah változó reprezentálja a kibővített rendszer $\tilde{A}$ állapotmátrixát, Bh reprezentálja a kibővített rendszer $\tilde{B}$ mátrixát, Ch reprezentálja a kibővített rendszer $\tilde{C}$ mátrixát. A kibővített rendszerre ezek után az állapotmegfigyelő tervezését ugyanúgy elvégezhetjük, mint korábban:

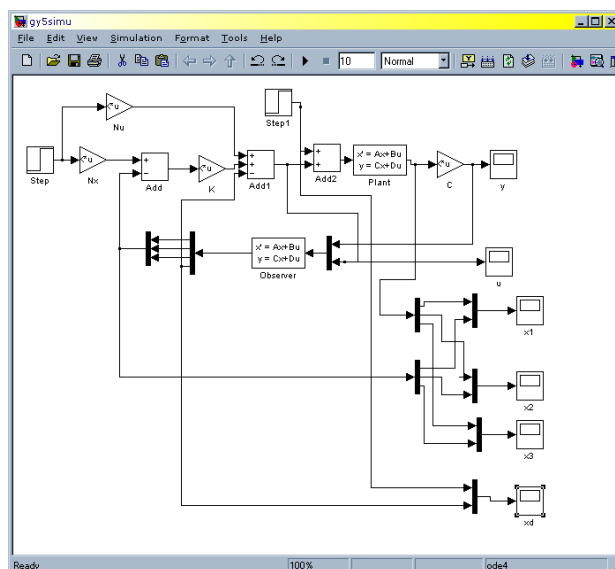
```
Ght=acker(Ah', Ch', soinf*ones(1,length(Ah)))  
Gh=Ght'  
Fh=Ah-Gh*Ch  
Hh=Bh
```


Egyedül arra kell ügyelni, hogy az állapotmátrix megnövelt mérete miatt az `acker` függvénynek eggyel több (az $\tilde{A}$ méretének megfelelő) sajátértéket kell megadni. Ez természetesen azt is jelenti, hogy a szakaszban található 3 időállandó (pólus) ellenére a megfigyelőnek négy sajátértéke lesz, hisz nem csak a folyamat 3 állapotát, hanem a d bemeneti terhelésnek megfelelő x_d állapotot is meg kell becsülni. A programkódban a `Fh` reprezentálja a megfigyelő $\tilde{F}$ mátrixát, `Gh` reprezentálja a megfigyelő $\tilde{G}$ mátrixát és `Hh` reprezentálja a megfigyelő $\tilde{H}$ mátrixát. Az utasításokat kiértékelve az új megfigyelő paraméterei:

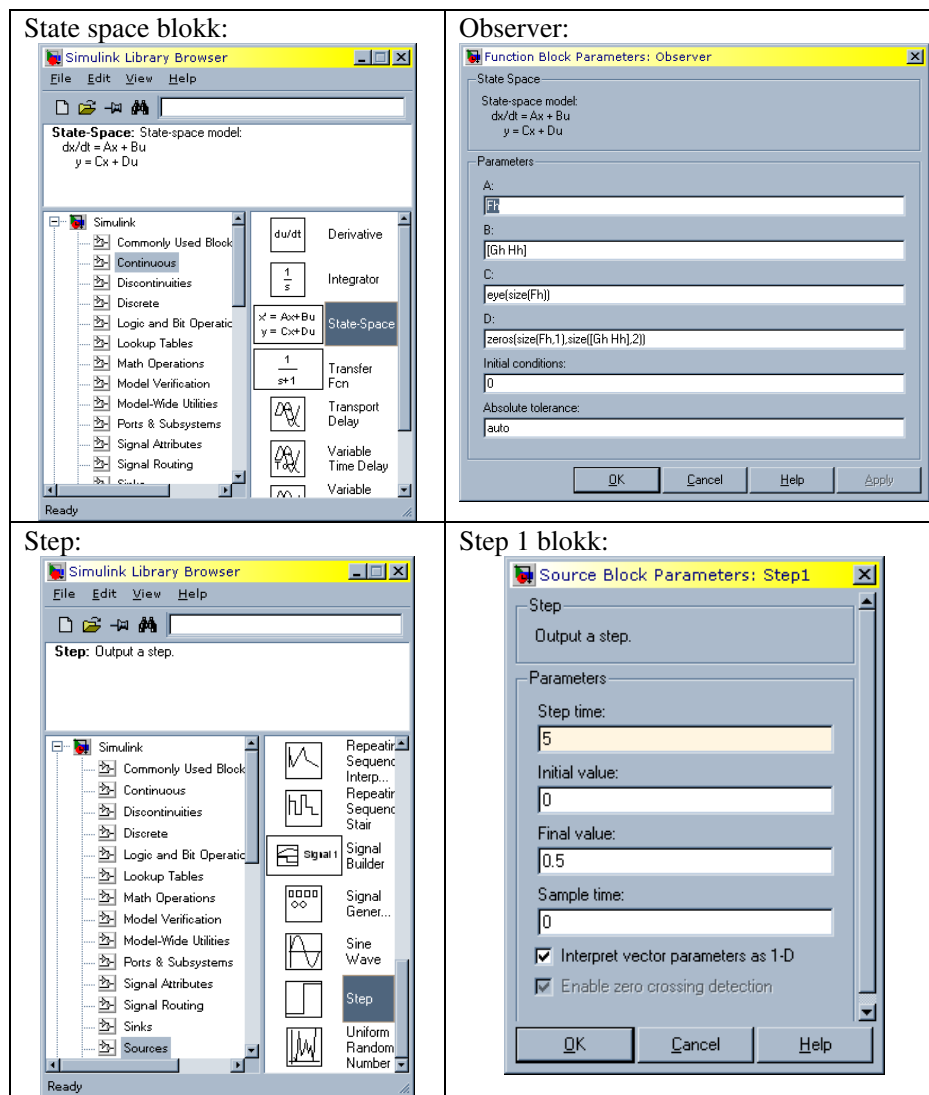
$$\tilde{F} = \begin{bmatrix} -1.4 & -0.4 & -728.4 & 0.5 \\ 1 & 0 & -448.4 & 0 \\ 0 & 0.1 & -12.7 & 0 \\ 0 & 0 & -2401 & 0 \end{bmatrix} \quad \tilde{G} = \begin{bmatrix} 9107 \\ 5605 \\ 15.8 \\ 3001.3 \end{bmatrix} \quad \tilde{H} = \begin{bmatrix} 0.5 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

Figyelem! A K állapot-visszacatolást és az N_x , N_u mátrixokat továbbra is az eredeti rendszer A , B mátrixai alapján kell megtervezni! Ez azt jelenti, hogy a már korábban megtervezett K , N_x , N_u mátrixokat (a jelenlegi SISO rendszerben oszlopvektorokat és skalárt) most is felhasználhatjuk, függetlenül attól, hogy a megfigyelőt lecseréltük.

Ellenőrzés: A terhelésbecslés ellenőrzését az 5.59. ábrán bemutatott Simulink modell segítségével végezhetjük el. Az új állapotmegfigyelő definícióját az 5.60. ábra mutatja.



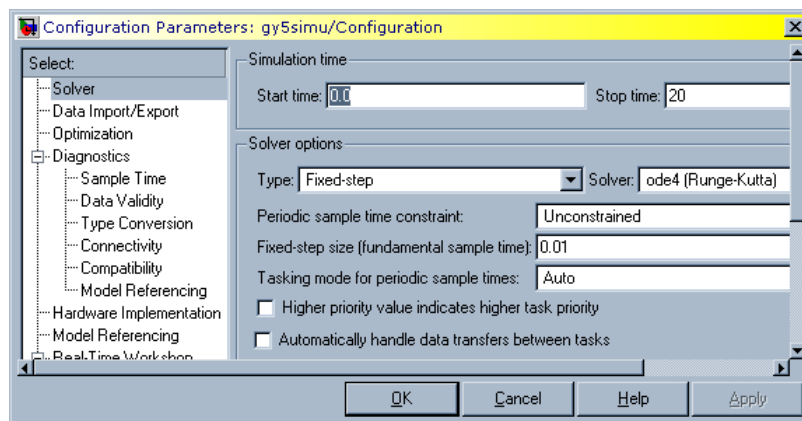
5.59. ábra. Az állapot-visszacatoláson alapuló és terhelésbecslést is alkalmazó szabályozás.



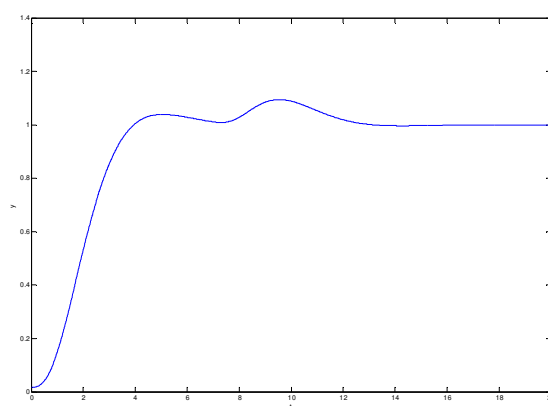
5.60. ábra. A terheléscsclést is tartalmazó megfigyelő paramétereinek beállítása a Simulink modellben.

Vegyük észre, hogy a terhelés bevezetése miatt egy extra összeadó blokk jelenik meg (*Add2* néven). A korábbi *Add1* a negatív visszacsatolást és alapjel bevezetését megvalósító összegző blokk pedig egy jellel bővül, mivel a megfigyelőből kijövő $\hat{x}_d$ becslt zavarjelet a kompenzálás céljából le kell vonni a beavatkozó jelből. A bemeneti terhelést szintén egy *step* függvényvel szimuláljuk, de az ugrás mértéke

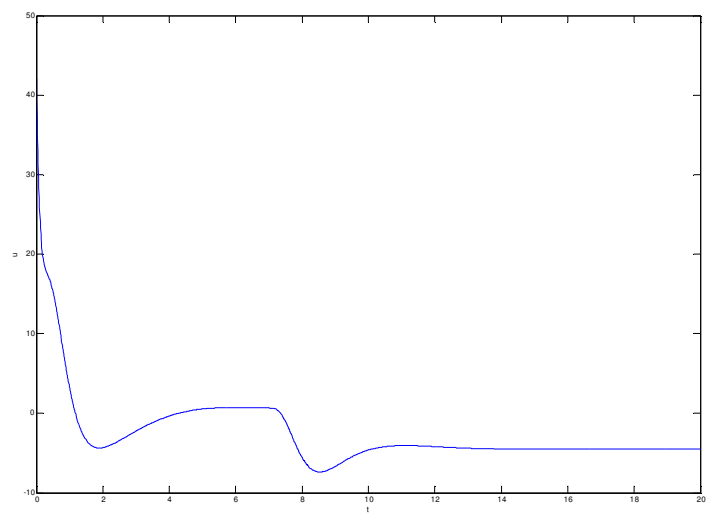
5 és az egységugrás $t = 7$ sec időpontban jelenik meg a rendszerben (erről a szabályozó nem tud semmit) annak érdekében, hogy az alapjel és a terhelés hatását külön tudjuk vizsgálni. Az 5.61. ábra mutatja a Simulink szimuláció konfigurációs paramétereit. A szimulációt most 20 sec ideig futtatjuk. Mivel a zárt rendszer kimenetét az 5.62. ábra, a beavatkozó jelet az 5.63. ábra mutatja. Látható, hogy a $t = 7$ sec időpontban belépő zavarás a lecsillapodni készülő kimenő jelet újra kibillenti. Mindez a jelenség kisebb nagyobb mértékben megfigyelhető az 5.64. ábrán, az 5.65. ábrán és az 5.66. ábrán is az állapotok alakulásánál. Azonban a terhelésbecslőnek köszönhetően miután az x_d állapot becslését sikerül elvégezni (lásd 5.67. ábra $t = 7$ sec utáni tranziensét), a kimenő jel is eléri a kívánt alapjel értékét, illetve az állapotbecslőben (főleg $\hat{x}_1$ és $\hat{x}_2$ ágán) megjelenő zavarok is fokozatosan eltűnnek.



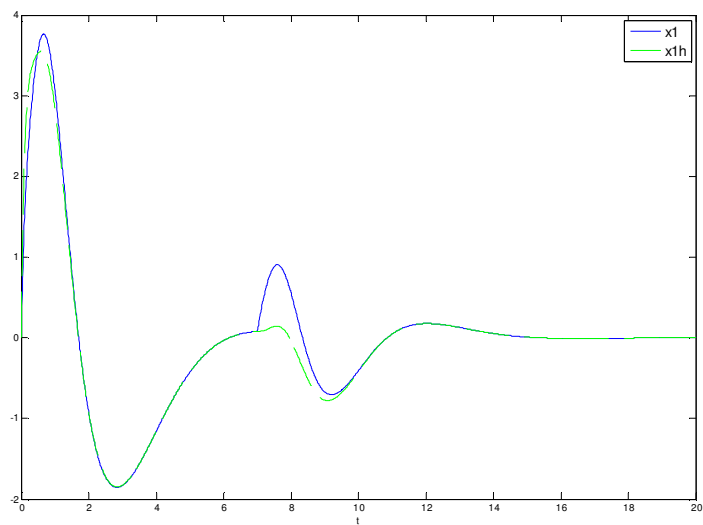
5.61. ábra. A szimuláció paramétereinek beállítása.



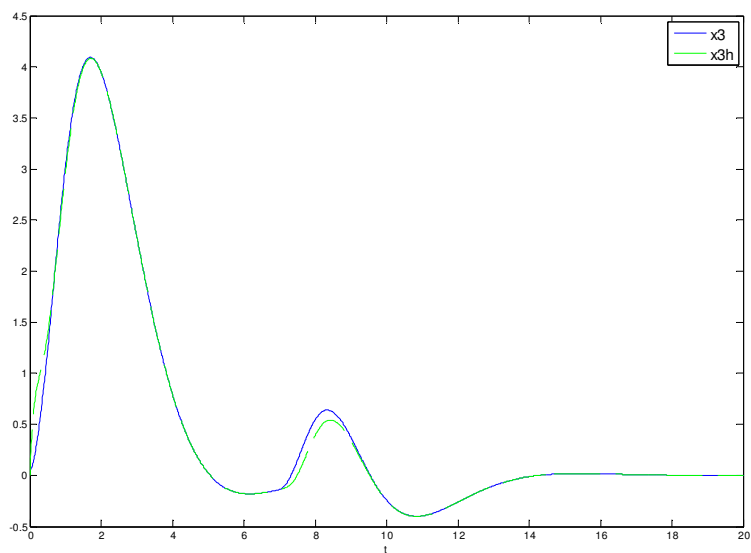
5.62. ábra. A kimenő jel terhelésbecslés mellett (terhelés: $d = 5$, $t = 7$ sec időponttól).



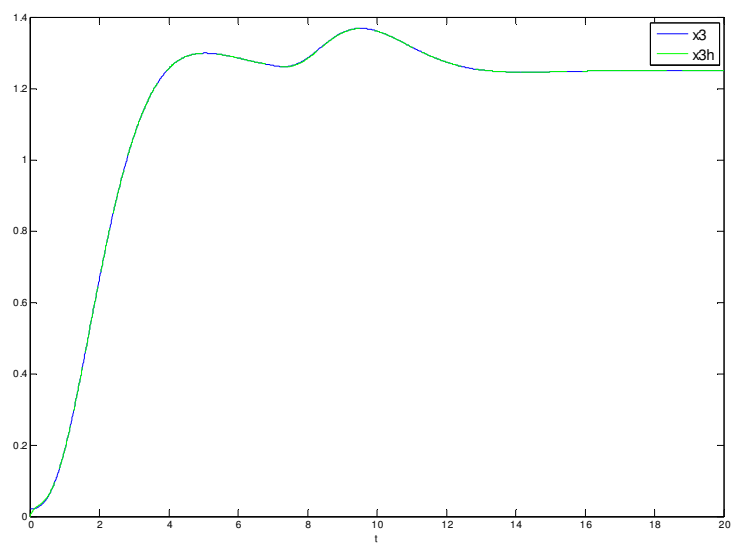
5.63. ábra. A beavatkozó jel terhelésbecslés mellett (terhelés: $d = 5$, $t = 7$ sec időponttól).



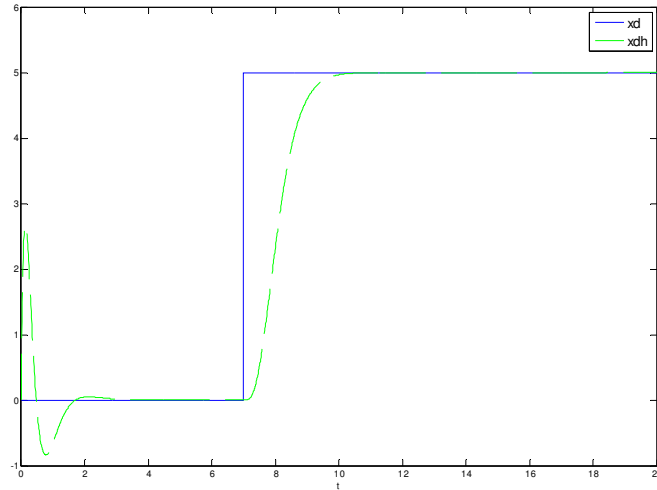
5.64. ábra. Az x_1 állapot (x_1) és annak $\hat{x}_1$ (x_{1h}) becslése terhelésbecslés alkalmazása mellett (terhelés: $d = 5$, $t = 7$ sec időponttól).



5.65. ábra. Az x_2 állapot (x_2) és annak $\hat{x}_2$ (x_2h) becslése terhelésbecslés alkalmazása mellett (terhelés: $d = 5$, $t = 7$ sec időponttól).



5.66. ábra. Az x_3 állapot (x_3) és annak $\hat{x}_3$ (x_3h) becslése terhelésbecslés alkalmazása mellett (terhelés: $d = 5$, $t = 7$ sec időponttól).



5.67. ábra. Az x_d állapot (x_d) és annak $\hat{x}_d$ (x_{dh}) becslése terhelésbecslés alkalmazása mellett (terhelés: $d = 5$, $t = 7$ sec időponttól).

Szatellit orientációszabályozása állapotér módszerrel

Az irányítási cél egy asztronómiai feladatot ellátó szatellit tudományos célú szenzoregységének az orientációjának a beállítása. A rendszer dinamikus modelljét a III.3 pontban már tárgyaltuk, ahol megadtuk az orientáció beállító egység egyszerűsített vázlatát a III.3. ábrán, amelyet most az 5.68. ábrán megismétlünk.

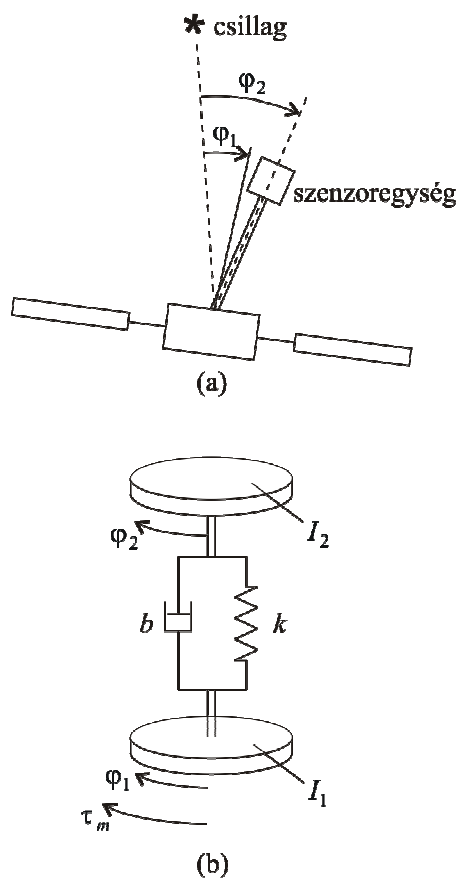
A φ_2 szög a szenzoregység és a csillag érzékelője közötti szög, amely felé a szatellit néz. A φ_1 szög a szatellit fő egységének orientációja a csillaghoz képest. A szenzoregység és a meghajtó motor közötti tengelykapcsolat elasztikus. A τ_m motor nyomaték a rendszer bemenőjele, melyet gyors tranziensű, alacsony szintű nyomatékszabályozás valósít meg, amellyel itt nem foglalkozunk. Elvárás, hogy az orientáció váltáskor a szabályozási időnek 20 sec-nál, a túllövésnek 15%-nál kisebbnek kell lennie.

A motor és a szenzor egység inerciáját rendre I_1 és I_2 , a flexibilis tengely viszkózus súrlódási tényezőjét b , rugóállandóját k jelöli. A zavaró jel nyomatékokat elhanyagoljuk. Az orientáció beállítást, felhasználva az akció-reakció (D'Alambert-) elvet is, a következő egyenletek írják le:

$$I_1 \ddot{\varphi}_1 = \tau_m - b(\dot{\varphi}_1 - \dot{\varphi}_2) - k(\varphi_1 - \varphi_2)$$

$$I_2 \ddot{\varphi}_2 = b(\dot{\varphi}_1 - \dot{\varphi}_2) + k(\varphi_1 - \varphi_2)$$

A tipikus paraméterek $I_1 = 1$, $I_2 = 0.1$, $0.09 \leq k \leq 0.4$, $0.038\sqrt{\frac{k}{10}} \leq b \leq 0.2\sqrt{\frac{k}{10}}$, minden SI egységben értendő.



5.68. ábra. Szatellit orientáció beállító egység egyszerűsített vázlata

Állapotegyenlet: $\dot{x} = Ax + Bu$, $y = Cx$

$$x = (\varphi_2, \dot{\varphi}_2, \varphi_1, \dot{\varphi}_1)^T, \quad u = \tau_m, \quad y = \varphi_2$$

$$\dot{x} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ -k/I_2 & -b/I_2 & k/I_2 & b/I_2 \\ 0 & 0 & 0 & 1 \\ k/I_1 & b/I_1 & -k/I_1 & -b/I_1 \end{bmatrix} x + \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1/I_1 \end{bmatrix} u$$

$$y = [1 \ 0 \ 0 \ 0]x$$

A szabályozás mérőrendszere a φ_1 és φ_2 ún. pozíciójeleket méri. A $\dot{\varphi}_1$ és $\dot{\varphi}_2$ ún. sebességjeleket az állapot-visszacsatoláshoz állapotmegfigyelő határozza meg.

A szabályozást a nominális rendszerhez tervezzük meg, amelynek paraméterei:

$$k = 0.091, \quad b = 0.0036, \quad I_1 = 1, \quad I_2 = 0.1 \quad (\text{mind SI egységben értendő}).$$

A szimmetrikus gyökhelygörbe módszer (SRL)

Gyakran kell a tervezés során kompromisszumot találni a gyors működés és a beavatkozó jel nagysága között. Ez a következő probléma megoldását igényli, ahol $z(t)$ a szakasz kimenőjelétől vagy az állapotoktól függő változó a tranzienssel kapcsolatos kívánalmak megfogalmazására, $u(t)$ a szabályozó kimenőjele, ρ pedig a keresett arány közöttük:

$$J = \int_0^\infty [\rho z^2(t) + u^2(t)] dt \rightarrow \min$$

$$u = -Kx$$

Jelölje a SISO rendszer u bemente és z kimenete közötti átviteli függvényt

$$W_{zu}(s) \Leftrightarrow Z(s) = W_{zu}(s)U(s),$$

akkor Letov (1960) és Chang (1961) eredményei szerint az optimális megoldásnak ki kell elégítenie a következő feltételt:

$$1 + \rho W_{zu}(-s)W_{zu}(s) = 0 \quad (\text{SRL feltétel})$$

Vegyük észre, hogy ekkor a $H(s) := W_{zu}(-s)W(s)$ komplex függvénynek a gyökei párban fordulnak elő, azaz ha az s_{left} gyök a bal félsíkon helyezkedik el, akkor az $s_{right} = -s_{left}$ párja szintén gyök, amely a jobb félsíkon helyezkedik el. Vegyük észre, hogy a gyökök meghatározhatók, ha felrajzoljuk $H(s)$ gyökhelygörbéjét (Symmetric Root Locus), mintha $\rho > 0$ volna a körerősítés. A gyökök ismeretében megválaszthatók a zárt rendszer domináns pólusai a stabil tartományban, amihez leolvasható a hozzátartozó optimális ρ arány. A továbbiakban $z = y$ választással élünk.

A szabályozás tervezését a mellékelt `SatelliteControl` programmal követhetjük nyomon.

Először meghatározzuk a nominális rendszert állapotegyenlet (`sys`) és átviteli függvény (`Wyu`) alakban. Meghatározzuk az átviteli függvény zérus helyét és

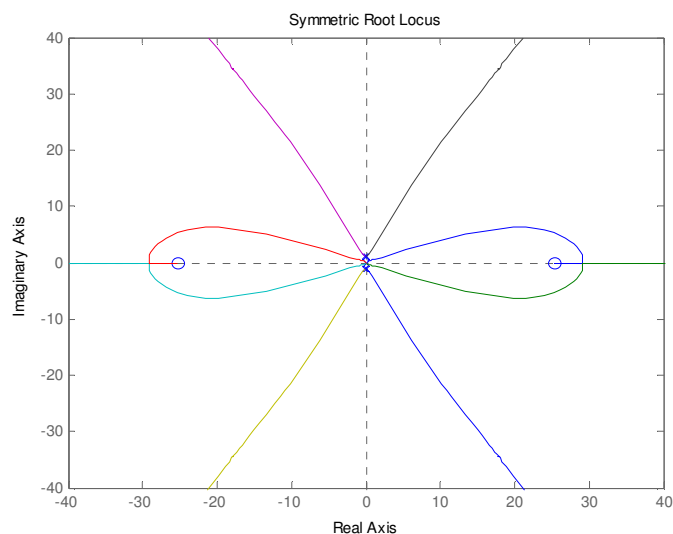
pólusait, valamint ezek csillapítását és csillapítatlan sajátfrekvenciáját, amelyből látható, hogy van egy gyengén csillapított póluspár is a két integrátor mellett:

Eigenvalue	Damping	Freq. (rad/s)
0.00e+000	-1.00e+000	0.00e+000
0.00e+000	-1.00e+000	0.00e+000
-1.98e-002 + 1.00e+000i	1.98e-002	1.00e+000
-1.98e-002 - 1.00e+000i	1.98e-002	1.00e+000

Ezután meghatározzuk a $H(s) = W_{yu}(-s)W_{yu}(s)$ komplex függvényt az SRT-hez:

```
Transfer function:
-0.001296 s^2 + 0.8281
-----
s^8 + 2 s^6 + 1.002 s^4
```

Az SRT-t az 5.69. ábra mutatja.



5.69. ábra. Szimmetrikus gyökhelygörbe (SRT)

Sajnos a MATLAB CST adottságai és a távoli zérushely miatt az SRT-nek számunkra fontos induló része nem értékelhető, ezért $\text{Rho}=0.1:0.1:1$ között külön is meghatároztuk a gyököket, és kiválasztottuk közülük a bal félsíkon lévőket. A 10 pólus oszlopokba szervezve helyezkedik el, minden oszlop egy-egy Rho

értéknek felel meg. Ezek közül a negyediket választottuk ki, és ehhez megadtuk a csillapítatlan sajátfrekvencia, csillapítás és túllövés értékeket is:

$P_{cl} =$

$$\begin{aligned} & -0.1989 + 1.0964i \\ & -0.1989 - 1.0964i \\ & -0.5594 + 0.3882i \\ & -0.5594 - 0.3882i \end{aligned}$$

$wn_xi_dv =$

$$\begin{array}{ccc} 1.1142 & 0.1785 & 56.5605 \\ 1.1142 & 0.1785 & 56.5605 \\ 0.6809 & 0.8216 & 1.0810 \\ 0.6809 & 0.8216 & 1.0810 \end{array}$$

A választott P_{cl} zárt rendszer pólusokhoz meghatároztuk az állapot-visszacsatolás az Ackermann-képlettel, eredményül a következőt kaptuk:

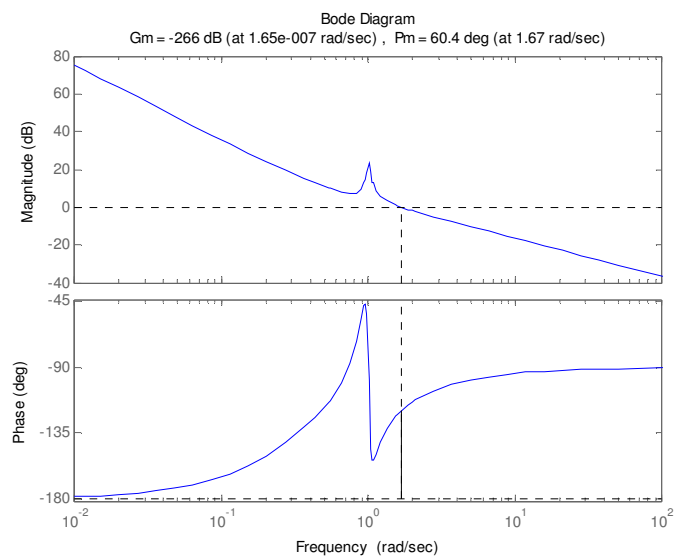
$K =$

$$\begin{array}{cccc} -0.4553 & 0.2270 & 1.0877 & 1.4769 \end{array}$$

Az így megtervezett állapot-visszacsatolást alkalmazó szabályozóval (nulla referencia jelet feltételezve), de a szabályozási kört az állapot-visszacsatolás után felvágva meghatároztuk a felnyitott kör állapotegyenletét:

$$\dot{x} = Ax + Bu_{in}, u_{out} = Kx.$$

Az állapotegyenletnek a $sysu=ss(A, B, K, 0)$ rendszer felel meg, amelynek meghatároztuk a Bode-diagramját és a fázistöbbletét a `margin` utasítással, lásd 5.70.ábra. Az ábra alapján a fázistöbblet $\Phi=60.4$ deg.



5.70. ábra. A felnyitott kör Bode-diagramja és fázistöbblete

Ezután meghatároztuk a referencia jel figyelembevételéhez a szabályozóban szükséges N_x , N_u értékeket (utóbbi nulla, mivel a szakasz két integrátort is tartalmaz):

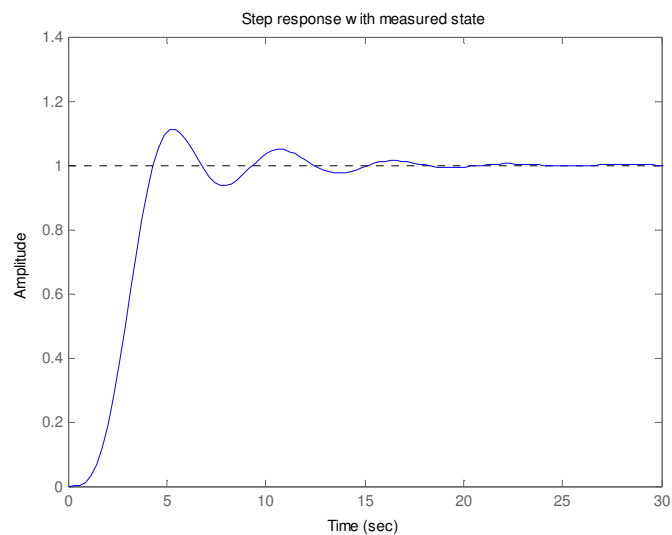
$N_x =$

1
0
1
0

$N_u =$

0

Ezt követően felépítjük a zárt szabályozási kört előbb mérhető állapotváltozókat feltételezve: `syscl=ss(A-B*K,B*(K*Nx+Nu),C,0)`, majd ehhez a zárt rendszer átmeneti függvényét, lásd 5.71. ábra.



5.71. ábra. A zárt rendszer átmeneti függvénye mért állapotok esetén

Ezután megtervezzük az állapotmegfigyelőt, sajátértékekként a zárt rendszer pólusainak 20-szorosát választva.

```
%Observer design
Pobs=Pc1*20
G=acker(A',C',Pobs) '
F=A-G*C
H=B
```

Az eredmények:

```
Pobs =

    -3.9774 +21.9271i
    -3.9774 -21.9271i
   -11.1874 +  7.7632i
   -11.1874 -  7.7632i

G =

1.0e+005 *

    0.0003
    0.0086
    0.0983
    1.0111
```

F =

```
1.0e+005 *
-0.0003    0.0000    0    0
-0.0086   -0.0000    0.0000  0.0000
-0.0983    0    0    0.0000
-1.0111    0.0000  -0.0000  -0.0000
```

H =

```
0
0
0
1
```

Ezután meghatároztuk az eredő szabályozót, amely tartalmazza az alapjel miatti korrekciót, az állapot-visszacsatolást és az állapotmegfigyelőt.

```
%Resulting controller
Ac=F-H*K
Bc=[H*(K*Nx+Nu) G]
Cc=-K
Dc=[K*Nx+Nu 0]
sysc=ss(Ac,Bc,Cc,Dc);
sysc.Inputname={'r','y'};
sysc.OutputName='u';
Wc=tf(sysc)
```

Az eredmények a következők:

Ac =

```
1.0e+005 *
-0.0003    0.0000    0    0
-0.0086   -0.0000    0.0000  0.0000
-0.0983    0    0    0.0000
-1.0111   -0.0000  -0.0000  -0.0000
```

Bc =

```
1.0e+005 *
0    0.0003
0    0.0086
0    0.0983
0.0000  1.0111
```

Cc =

0.4553 -0.2270 -1.0877 -1.4769

Dc =

0.6325 0

Transfer function from input "r" to output "u":

0.6325 s^4 + 19.18 s^3 + 543.9 s^2 + 7961 s + 5.824e004

s^4 + 31.81 s^3 + 905.9 s^2 + 1.389e004 s + 1.058e005

Transfer function from input "y" to output "u":

-1.602e005 s^3 - 1.169e005 s^2 - 1.649e005 s - 5.824e004

s^4 + 31.81 s^3 + 905.9 s^2 + 1.389e004 s + 1.058e005

Ezután összekötöttük a szakaszt a szabályozóval:

```
%Closed loop with observer
syscl1=connect(sys,sysc,'r',{'y','u'})
figure(4)
step(syscl1)
title('Closed loop step response with nominal plant')
```

Az erdő syscl1 zárt rendszer állapotegyenlete a szabályozóval és a nominális szakasszal:

```
a =
```

	x1	x2	x3	x4	x5
x1	0	1	0	0	0
x2	-0.91	-0.036	0.91	0.036	0
x3	0	0	0	1	0
x4	0.091	0.0036	-0.091	-0.0036	0.4553
x5	30.29	0	0	0	-30.29
x6	857.8	0	0	0	-858.7
x7	9825	0	0	0	-9825
x8	1.011e+005	0	0	0	-1.011e+005

	x6	x7	x8
x1	0	0	0
x2	0	0	0
x3	0	0	0
x4	-0.227	-1.088	-1.477
x5	1	0	0
x6	-0.036	0.91	0.036
x7	0	0	1
x8	-0.2234	-1.179	-1.48

b =

```

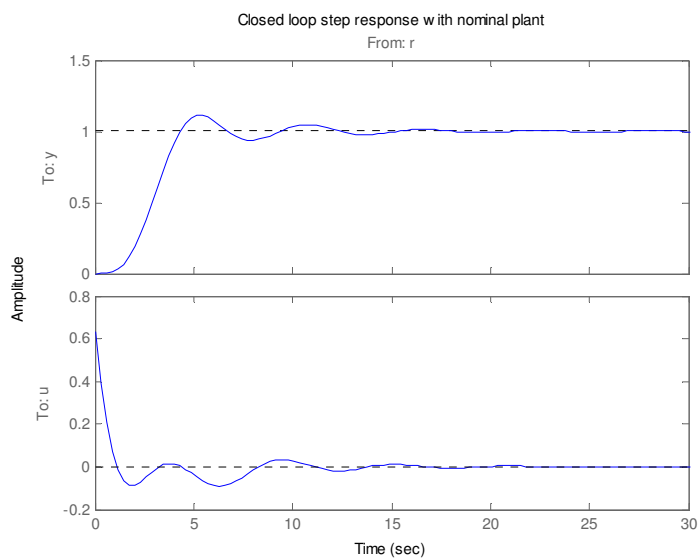
      r
x1      0
x2      0
x3      0
x4 0.6325
x5      0
x6      0
x7      0
x8 0.6325

c =
      x1      x2      x3      x4      x5      x6      x7      x8
y      1      0      0      0      0      0      0      0
u      0      0      0      0 0.4553 -0.227 -1.088 -1.477

d =
      r
y      0
u 0.6325

```

A `syscl1` zárt rendszer átmeneti függvényét a nominális szakasszal az 5.72. ábra mutatja.



5.72. ábra. A zárt rendszer átmeneti függvénye a nominális szakasszal.

Ezután módosítjuk a szakaszt, megváltoztatván a rugóállandó értékét a legnagyobb üzemi $k=0.4$ értékűre (stiff spring):

```

%Perturbed plant
k2=0.4; b2=0.0036; I1=1; I2=0.1;
A2 = [0 1 0 0;
      -k2/I2 -b2/I2 k2/I2 b2/I2;
      0 0 0 1;

```

```

      k2/I1  b2/I1  -k2/I1  -b2/I1]
B2 = [0 0 0 1/I1] '
C2 = [1 0 0 0]
D2 = 0
t=[0:0.01:100];
sys2=ss(A2,B2,C2,D2);
sys2.InputName='u';
sys2.OutputName='y';

```

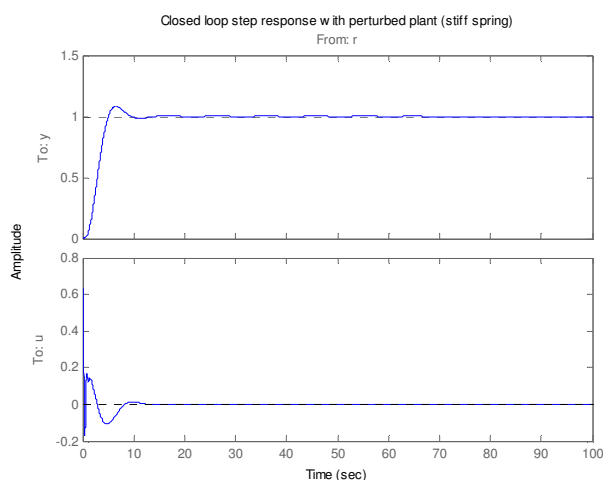
Megvizsgáljuk, mennyire tolerálja a nominális szabályozó a megváltozott szakaszt a zárt szabályozási körben:

```

%
Wyu2=tf(sys2)
num2=Wyu2.num{:};
num2=num2(min(find(num2~=0)):length(num2));
den2=Wyu2.den{:};
den2(4:5)=0;
Wyu2=tf(num2,den2)
Wyu2.InputName='u';
Wyu2.OutputName='y';
syscl2=connect(Wyu2,sysc,'r',{'y','u'})
%
figure(5)
step(syscl2,t)
title('Closed loop step response with perturbed plant
(stiff spring)')

```

A syscl2 zárt rendszer átmeneti függvényét a nominális szabályozóval és a megváltozott szakasszal az 5.73. ábra mutatja.



5.73. ábra. A zárt kör átmeneti függvénye nominális szabályozó és perturbált szakasz esetén (merev rugó)

Melléklet – Matlab kód a szatellit orientáció szabályozás vizsgálatához állapot-visszacsatolással és megfigyelővel

```
%SatelliteControl.m
%Satellite orientation control in state space
clear all
close all
clc

%Nominal system
k=0.091; b=0.0036; I1=1; I2=0.1;

A = [0 1 0 0;
     -k/I2 -b/I2 k/I2 b/I2;
     0 0 0 1;
     k/I1 b/I1 -k/I1 -b/I1]
B = [0 0 0 1/I1]'
C = [1 0 0 0]
D = 0

%Nominal system
sys=ss(A,B,C,D);
sys.InputName='u';
sys.OutputName='y';
format long
eigsys=eig(sys.a)
format short
%
Wyu=tf(sys)
num=Wyu.num{:};
num=num(min(find(num~=0)):length(num));
den=Wyu.den{:};
den(4:5)=0;
Wyu=tf(num,den)
zvec=roots(num)
pvec=roots(den)
damp(pvec)

%Symmetric Root Locus
%H(s)=Wyu(-s)*Wyu(s)
num_minus=[-1 1].*num
den_minus=[1 -1 1 -1 1].*den
Wyu_minus=tf(num_minus,den_minus)
numH=conv(num,num_minus)
denH=conv(den,den_minus)
H=tf(numH,denH)
figure(1)
rlocus(H)
title('Symmetric Root Locus')
```

```

[P,Rho]=rlocus(H,[0.1:0.1:1]);
kkmax=length(Rho);
PP=zeros(4,kkmax);
for kk=1:kkmax
    Pkk=P(:,kk);
    ip=find(real(Pkk<0));
    PP(:,kk)=Pkk(ip);
end
PP
kk=4 %k=input('Choose column k in {1,...,10} for closed
loop poles, k=?');
Pcl=PP(:,kk)
[wn,xi]=damp(Pcl);
dv=exp(-pi*xi./sqrt(1-xi.^2))*100;
wn_xi_dv=[wn xi dv]

%State feedback design
K=acker(A,B,Pcl)
sysu=ss(A,B,K,0);
figure(2)
margin(sysu)

%Reference signal correction
Nxu=inv([A B; C 0])*[0 0 0 0 1]';
Nx=Nxu(1:4)
Nu=Nxu(5)

%Closed loop with measured state
syscl=ss(A-B*K,B*(K*Nx+Nu),C,0);
figure(3)
step(syscl)
title('Step response with measured state')

%Observer design
Pobs=Pcl*20
G=acker(A',C',Pobs)'
F=A-G*C
H=B

%Resulting controller
Ac=F-H*K
Bc=[H*(K*Nx+Nu) G]
Cc=-K
Dc=[K*Nx+Nu 0]
sysc=ss(Ac,Bc,Cc,Dc);
sysc.Inputname={'r','y'};
sysc.OutputName='u';
Wc=tf(sysc)

%Closed loop with observer

```

```

syscl1=connect(sys,sysc,'r',{'y','u'})
figure(4)
step(syscl1)
title('Closed loop step response with nominal plant')

%Perturbed plant
k2=0.4; b2=0.0036; I1=1; I2=0.1;
A2 = [0 1 0 0;
      -k2/I2 -b2/I2 k2/I2 b2/I2;
      0 0 0 1;
      k2/I1 b2/I1 -k2/I1 -b2/I1]
B2 = [0 0 0 1/I1]'
C2 = [1 0 0 0]
D2 = 0
t=[0:0.01:100];
sys2=ss(A2,B2,C2,D2);
sys2.InputName='u';
sys2.OutputName='y';

%
Wyu2=tf(sys2)
num2=Wyu2.num{:};
num2=num2(min(find(num2~=0)):length(num2));
den2=Wyu2.den{:};
den2(4:5)=0;
Wyu2=tf(num2,den2)
Wyu2.InputName='u';
Wyu2.OutputName='y';
syscl2=connect(Wyu2,sysc,'r',{'y','u'})
%
figure(5)
step(syscl2,t)
title('Closed loop step response with perturbed plant
(stiff spring)')

```

5. Ellenőrző kérdések a gyakorlathoz

1. Adja meg az irányíthatóság és elérhetőség szabatos definícióját folytonosidejű rendszer esetén.
2. Adja meg a $P(\tau, t)$ irányíthatósági Gram-mátrixot időben változó (LTV) és időinvariáns (LTI) lineáris rendszer esetén, és kapcsolatát az irányítható és nem irányítható állapotok altereivel.
3. Adja meg az M_c irányíthatósági mátrixot, az irányítható állapotok L alterét és a teljes irányíthatóság feltételét.
4. Fogalmazza meg a pólusáthelyezési feladatot állapot-visszacsatolás esetén, és a megoldás meghatározására szolgáló Ackermann-képletet SISO rendszert feltételezve. Adja meg a zárt rendszer hatásvázlatát az állapot-visszacsatolás és mérhető állapot esetén.
5. Mit értünk irányíthatósági lépcsős alak alatt, és erre alapozva adja meg a stabilizálható rendszer definícióját.
6. Adja meg az alapjel miatti korrekcióhoz szükséges N_x , N_u mátrixok számítási szabályát, méretüket speciálisan SISO rendszer esetén, és a zárt rendszer hatásvázlatát állapot-visszacsatolás és az alapjel miatti korrekció feltüntetésével.
7. Adja meg a megfigyelhetőség és rekonstruálhatóság szabatos definícióját, és értelmezését speciálisan lineáris rendszer esetén.
8. Adja meg a $Q(\tau, t)$ megfigyelhetőségi Gram-mátrixot időben változó (LTV) és időinvariáns (LTI) lineáris rendszer esetén. Adja meg azt a fiktív rendszert, amelynek $P_H(\tau, t)$ irányíthatósági Gram-mátrixa azonos a $Q(\tau, t)$ megfigyelhetőségi Gram-mátrixszal (megfigyelhetőség és irányíthatóság dualitása).
9. Adja meg az M_o megfigyelhetőségi mátrixot, a megfigyelhetőségi lépcsős alakot, és erre alapozva a detektálható rendszer definícióját.
10. Adja meg a folytonosidejű teljesrendű megfigyelő állapotegyenletét és a benne szereplő mátrixok megválasztását.
11. Adja meg a megfigyelőtervezési feladat megoldásának sémáját a dualitás elve és az Ackermann-képlet felhasználásával.
12. Adja meg az állapot-visszacsatolás, alapjel miatti korrekció és állapotmegfigyelő együttes alkalmazása esetén a zárt rendszer hatásvázlatát.

13. Fogalmazza meg az integrátort is tartalmazó állapot-visszacsatolási feladatot, adja meg a tervezés lépéseit és rajzolja fel alkalmazása esetén a zárt rendszer hatásvázlatát.
14. Adja meg a terhelésbecslést (bemeneti zavaráskompenzálást) alkalmazó állapotmegfigyelő tervezési lépéseit, a benne szereplő mátrixok megválasztását és az Ackermann-képletre visszavezethető feladat alakját.
15. Adja meg az állapot-visszacsatolást, alapjel miatti korrekciót és terhelésbecslőt alkalmazó szabályozó tervezési lépéseit, és a zárt rendszer hatásvázlatát együttes alkalmazásukkor.
16. Adja meg az időinvariáns lineáris rendszer Kalman-féle felbontását rajzban és a felbontásnak megfelelő állapotegyenletet. Melyik rendszer rész befolyásolja az átviteli függvényt, és mit okoz a többi rendszerkomponens?