

Szabályozástechnika Matlab Gyakorlatok, Villamosmérnöki

**Hetthéssy Jenő
Bars Ruth
Barta András**



*Budapesti Műszaki és Gazdaságtudományi Egyetem
Villamosmérnöki és Informatikai Kar
Automatizálási és Alkalmazott Informatikai Tanszék*

2005

Tartalomjegyzék

1. Bevezetés a Matlab Használatába	2
2. Bevezetés a Matlab Control System Toolbox Használatába.....	9
3. A frekvenciafüggvény	17
4. Lineáris Rendszer Elemei.....	21
5. Visszacsatolás Vizsgálata.....	28
6. Stabilitásvizsgálat.....	31
7. Soros PID Kompenzáció	36
8. Holtidős Rendszer Soros Kompenzációja	44
9. Labilis Rendszer Soros Kompenzációja.....	47
10. Mintavételes Rendszerek.....	50
11. Mintavételes PID Szabályozó Tervezése Kisfrekvenciás Közelítés Alapján	54
12. Állapotteres Leírás, Irányíthatóság, Megfigyelhetőség.....	61
13. Állapotvisszacsatolás	65

1. Bevezetés a Matlab Használatába

A Matlab egy interaktív programcsomag tudományos és mérnöki számítások, szimulációk és grafikus adatmegjelenítés elvégzésére. A Matlab erős háttérrel biztosít mátrix algebra, differenciálegyenletek és más matematikai problémák megoldására. A Matlab elterjedtsége és széleskörű alkalmazása abból adódik, hogy a Matlab utasításkészlete kiterjeszhető *toolbox*-ok segítségével. A *toolbox* tulajdonképpen egy függvénykönyvtár, amit különféle szakterületek támogatására fejlesztettek ki. Ilyen szakterületek például a jelfeldolgozás (Signal Processing Toolbox), a szabályozástechnika (Control System Toolbox), a képfeldolgozás (Image Processing Toolbox), a neurális hálózatok alkalmazása (Neural Network Toolbox), stb. Ennek a bevezetésnek a célja, hogy minél gyorsabban el lehessen jutni egy olyan szintre, ami szükséges az alapproblémák megoldására. Részletesebb információt a Matlab felhasználói segédletek tartalmaznak. Elektronikus formában ezek a '*matlab/help*' könyvtárban találhatóak meg. Egy on-line sűgó is a felhasználó rendelkezésére áll.

» **helpdesk**

A *help* utasítás nyújt segítséget az egyes utasítások használatáról, például:

» **help sqrt**

A Matlab utasításokból egy szkript fájl is létre lehet hozni. Ez egy text fájl *.m* kiterjesztéssel. Ez a szkript új utasításként használható (kiterjesztés nélkül).

Változó nevek: A változók nevének mérete maximálisan 31 karakter lehet (betűk, számok és a '_' karakter). Az első karakter nem lehet szám. A kis és nagybetűk nem azonosak. A **casesen** utasítás megváltoztatja ezt a működési módot (azaz azonosá teheti a kis és nagybetűket, de ennek használata nem ajánlott). Minden változó egy mátrix. Egy skaláris változó egyszer egyes mátrixnak tekinthető.

Adatbevitel: A Matlab számos adattípust használ, de ezeket nem kell megadni, automatikusan deklarálja őket.

Integer:

» **k=2**

Ha az utasítás pontosvesszővel végződik, akkor az eredmény nem jelenik meg a képernyőn.

» **J=-4;**

Valós:

» **s=3.6**

» **F2=-12.6e-5**

Komplex:

» **z=3+4*i**

» **r=5*exp(i*pi/3)**

Az $i=\sqrt{-1}$ változó, az egység képzetes vektor előre meghatározott. Ehelyett azonban más változó is használható, például:

» **j=sqrt(-1)**

Vektor:

» **x=[1, 2, 3]** % sorvektor, az elemeket vesszővel vagy üres karakterrel (space) kell elválasztani

» **q=[4; 5; 6]** % oszlopvektor, az elemeket pontosvesszővel kell elválasztani

A sorvektorból a transzponálás műveletével is képezhető oszlopvektor:

» **v=[4, 5, 6]'** % ugyanaz mint q

Megjegyzés: A transzponálás komplex változók esetén a transzponált vektor komplex konjugáltját képezi.

» **i'**

0 - 1.000i

Mátrix:

» **A=[7, 8, 9; 5, 6, 7]** % A egy 2x3-as mátrix, » MATRIX=[row1; row2; ...; rowN];

Speciális vektorok és mátrixok:

```
» u=1:3; % létrehozza a u=[1 2 3] vektort, » u=start:stop
» w=1:2:10; % létrehozza a w=[1 3 5 7 9] vektort, » =start:increment:stop
» E=eye(4)
```

```
E=
    1    0    0    0
    0    1    0    0
    0    0    1    0
    0    0    0    1
```

```
» B=eye(3,4)
```

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

```
» C=zeros(2,4)
```

```
C=
    0    0    0    0
    0    0    0    0
```

```
» D=ones(3,5)
```

```
D=
    1    1    1    1    1
    1    1    1    1    1
    1    1    1    1    1
```

Változó értékek: A változó nevének beírása kijelzi a változó értékét.

```
» A
```

```
A=
    7    8    9
    5    6    7
```

```
» A(2, 3)
```

```
ans =
    7
```

Az első index a sor számát jelzi, a második pedig az oszlop sorszámát adja meg. A választ az *ans* változó veszi fel.

Egy mátrix vagy vektor egy elemének megváltoztatása után a változó összes eleme látható a képernyőn, kivéve ha az utasítás pontosvesszővel végződik.

```
» v(2) = -6
```

```
v=
    4
   -6
    6
```

Indexelés (Subscripting): A kettőspont (:) segítségével a mátrix több eleme is hozzáférhető egyszerre. Többféleképpen is használható ez az operátor.

Kezdeti index : végső index - egy mátrix egy részéhez enged hozzáférni

: - a kettőspont az összes elemet jelöli ki egy adott sorban vagy oszlopban.

Vektorok esetén: $v=[v(1) \ v(2) \ \dots \ v(N)]$

Mátrixok esetén: $M=[M(1,1)...M(1,m); M(2,1)...M(2,m); \dots ; M(n,1)...M(n,m)]$

Ha **B** egy 8x8 mátrix, akkor

`B(1:5,3)` egy oszlopvektor, [`B(1,3); B(2,3); B(3,3); B(4,3); B(5,3)`]
`B(2:3,4:5)` egy mátrix [`B(2,4) B(2,5); B(3,4) B(3,5)`]
`B(:,3)` a B mátrix mindegyik sorát és harmadik oszlopát, azaz a harmadik oszlopot jelenti
`B(2,:)` a B mátrix második sorát jelenti
`B(1:3,:)` a B mátrix első három sorát jelenti.
 » `A(2,1:2)` % az A mátrix második sorának első és második elemét jelöli ki
 » `A(:,2)` % az A mátrix második oszlopát jelöli ki

Munkaterület (Workspace): A használt változók a *workspace*-nek nevezett memória területen helyezkednek el. A *workspace* a következő utasításokkal jeleníthető meg:

» `who`
 » `whos` % kiírja a változó méreteit is

A változók méreteit a `length` és a `size` utasítással lehet megadni.

Vektorok esetén:

» `lng=length(v)`
 `lng=`
 3

Vektorok és mátrixok esetén:

» `[m,n]=size(A)`
 `m=`
 2 % a sorok száma
 `n=`
 3 % az oszlopok száma

A *workspace*-t el lehet menteni, betölteni és kitörölni:

» `save` % elmenti a *workspace*-t a default *matlab.mat* fájlba.
 » `save filename.mat` % elmenti a *workspace*-t a *filename.mat* fájlba.
 » `clear` % kitörli az összes változót.
 » `load` % betölti a *workspace*-be a *matlab.mat* fájl tartalmát
 » `load filename.mat` % betölti a *workspace*-be a *filename.mat* fájl tartalmát

Aritmetikai operátorok:

Összeadás és kivonás:

» `A=[1 2; 3 4];`
 » `B=A';`
 » `C=A+B;`
 `C=`
 2 5
 5 8
 » `D=A-B`
 `D=`
 0 -1
 1 0
 » `x=[-1 0 2]';`
 » `y=x-1` (Vegyük észre, hogy a vektor mindegyik eleme megváltozott!)
 `y=`
 -2
 -1
 1

Szorzás:

Vektor szorzása skalárral:

```
» 2*x
ans=
-2
0
4
```

Mátrix szorzása skalárral:

```
» 3*A
ans=
3    6
9   12
```

Skaláris szorzat:

```
» s=x'*y
s=
4
```

```
» y'*x
ans=
4
```

Diadikus szorzat(mátrixot eredményez):

```
» M=x*y'
M=
2    1   -1
0    0    0
-4   -2    2

» y*x'
ans=
2    0   -4
1    0   -2
-1   0    2
```

Mátrix szorzása vektorral:

```
» b=M*x
b=
-4
0
8
```

Osztás:

```
» C/2
```

Mátrixok esetén: B/A megfelel BA^{-1} -nek
 $A \setminus B$ megfelel $A^{-1}B$ -nek

Hatványozás: A^p , ahol A egy négyzetes mátrix és p egy valós konstans
például A inverze: A^{-1} , vagy ezzel egyenértékű $inv(A)$

Műveletek komplex számokon:

```
» r
r=
2.5000 + 4.3301i

» real(r)
ans=
2.5000
```

```

>> imag(r)
ans=
    4.3301
>> abs(r)
ans=
    5
>> angle(r)
ans=
    1.0472

```

A szög fokokban

```

>> 180/pi*angle(r)
ans=
    60.0000

```

Tömbök elemein végzett műveletek:

A tömbök elemein végzett aritmetikai műveletek a műveletek fontos csoportját alkotják. Ha az aritmetikai műveletet megelőzi egy pont, akkor ez a művelet a változók elemeire vonatkozik.

A .* művelet összeszorozza a változók megfelelő indexű elemeit. Például sorvektorok esetén:

$a.*b=[a(1)*b(1), a(2)*b(2), \dots, a(n)*b(n)]$. A változók méretének meg kell egyeznie.

Például:

```

>> a=[2 4 6]
>> b=[5 3 1]
>> a.*b
ans=
    10    12     6

```

Az elemenként végzendő műveletek értelemszerűen más operátorokra is alkalmazhatók, például osztás és hatványozás esetén: ./, .^

Alapvető matematikai függvények: (Az egyes függvények használatát a **help** utasítás mutatja meg)

abs	absolute value or magnitude of complex numbers
sqrt	square root
real	real part
imag	imaginary part
conj	complex conjugate
round	round to nearest integer
fix	round towards zero
floor	round towards -infinity
ceil	round towards +infinity
sign	signum function
rem	remainder
sin	sine
cos	cosine
tan	tangent
asin	arcsine
acos	arccosine
atan	arctangent
atan2	four quadrant arctangent
sinh	hyperbolic sine
cosh	hyperbolic cosine
tanh	hyperbolic tangent
exp	exponential base e

log	natural logarithm
log10	log base 10
bessel	Bessel function
rat	rational approximation
expm	matrix exponential
logm	matrix logarithm
sqrtm	matrix square root.

Például:

```
» help sqrt
» g=sqrt(2)
```

Grafikus kimenet:

A legegyszerűbb grafikus utasítás a *plot*.

```
» plot(2,3) % kirajzolja az x=2, y=3 pontot
```

Egyszerre több pontot is ki lehet rajzolni, ha a koordináta értékeket egy vektorban tároljuk el.

```
» x=[1,2,3]
» y=[0,2,1]
» plot(x,y) % a pontok össze vannak kötve egyenesekkel
» plot(x,y,'*') % így az utasítás csak a pontokat rajzolja ki
```

Ezzel a módszerrel bonyolultabb görbék is kirajzolhatók.

Például:

```
» t=0:0.05:4*pi;
» y=sin(t);
» plot(t,y)
» title('Sine function'), xlabel('Time'), ylabel('sin(t)'), grid;
```

ahol a *title*, *xlabel*, *ylabel* és *grid* utasítások opcionálisak.

Egyszerre több görbe is kirajzolható:

```
» y1=3*sin(2*t);
» plot(t,y,'r',t,y1,'b'); % r-red (piros), b-blue (kék)
```

A kirajzolás színét és típusát a következő formátumban lehet megadni: » `plot(t,y,'@#')`, ahol '@' jelenti a vonal típusát:

- folytonos
- szaggatott
- : pontvonal
- . pontok
- + plusz
- * csillag
- o kör
- x x-jel

és '#' jelenti a színt:

- r red, piros
- g green, zöld
- b blue, kék
- w white, fehér
- y yellow, sárga

Polinomok: A Matlab-ban a polinomokat együtthatóikkal adhatjuk meg. Például a $P(x)=4x^4-6x^3+9x^2-5$ polinomot megadhatjuk egy vektorral, aminek az elemei a polinom együtthatói.

```
» v=[4 -6 9 0 -5]
```

A polinom gyökeit, azaz a $P(x)=0$ egyenlet megoldásait a *roots* utasítás számolja ki.


```
» p=roots (v)
```

```
p =
    0.6198 + 1.4337i
    0.6198 - 1.4337i
    0.8577
   -0.5974
```

A polinom p_1, p_2, \dots, p_n gyökeiből kiszámolhatjuk a $P(x)=(x-p_1)(x-p_2)\dots(x-p_n)$ polinom együtthatóit.

```
» v1=poly (p)
```

```
v1 =
    1.0000   -1.5000    2.2500    0   -1.2500
```

A `poly` utasítás a legnagyobb hatványú együtthatót egynek veszi (normálja). Ha vissza akarjuk kapni az eredeti polinomot, akkor meg kell szorozni az összes együtthatót az első együtthatóval, a mi esetünkben 4-gyel.

```
» v2=4*poly (p)
```

```
v2 =
    4.0000   -6.0000    9.0000    0   -5.0000
```

Vegyük fel a következő mátrixot:

```
» M=[3 5 ; 7 -1]
```

Az M mátrix sajátértékeit az `eig` utasítás határozza meg.

```
» e=eig (M)
```

```
e =
    7.2450
   -5.2450
```

A sajátértékeket egy polinom gyökeinek tekintve a `poly` utasítással egy polinom meghatározható:

```
» poly (e)
```

```
ans =
    1   -2   -38
```

A fenti polinom, azaz $x^2-2x-38$ az M mátrix *karakterisztikus polinomja*, ami a $\det(xI-M)$ egyenlettel határozható meg. A karakterisztikus polinom az M mátrixból közvetlenül is kiszámítható:

```
» poly (M)
```

```
ans =
    1   -2   -38
```

Látható, hogy sokszor egy utasítás többféleképpen is meghívható. A Matlab help egy utasítás valamennyi meghívási lehetőségét megadja.

2. Bevezetés a Matlab Control System Toolbox Használatába

Tekintsünk egy egy-bemenetű, egy-kimenetű folytonos lineáris rendszert, amely átviteli függvényével adott:

$$\frac{u(t)}{U(s)} \rightarrow \boxed{H(s) = \frac{Y(s)}{U(s)} = \frac{num}{den}} \rightarrow \frac{y(t)}{Y(s)}$$

Állítsuk elő a Matlab segítségével a $H(s) = \frac{2}{s^2 + 2s + 4}$ rendszer *átmeneti függvényét*, azaz az $y(t)$

kimenetet egységugrás bemenet esetén. Az átviteli függvényt a számláló és a nevező polinom felvételével határozzuk meg. Egy polinomot az együtthatóival adhatunk meg, így a fenti példában $num = 2$,

$den = s^2 + 2s + 4$.

» **num=2**

» **den=[1 2 4]**

Általános esetben az együtthatókat csökkenő fokszám szerinti sorrendben kell egy vektorba gyűjteni. Az átmeneti függvényt a Matlab *step* utasításával lehet megjeleníteni:

» **step(num,den) ;**

Ugyanezt közvetlenül is meg lehet adni:

» **step(2,[1 2 4]) ;**

Vegyük észre, hogy az időskálát a Matlab automatikusan választotta.

Amennyiben tárolni kívánjuk a kiszámított átmeneti függvény értékeit, akkor a *step* függvényt bal oldali argumentummal - annak a változó nevének megadásával, amelyben tárolni kívánjuk a kiszámított függvényértékeket - kell meghívunk:

» **[y,x,t]=step(num,den)** vagy egyszerűbben

» **y=step(num,den)**

Látható és érdemes megjegyezni, hogy a Matlab függvényhívások jellemzője, hogy az argumentumok száma változó lehet mindkét oldalon. A fenti három bal oldali változót megjelölő függvényhívásban az y a kimenőjel értékeit, az x a rendszer belső állapotváltozóinak értékeit és a t a számítások időpontjainak értékeit tartalmazza. Vegyük észre, hogy ebben az esetben grafikus ábrázolás nem történt. Ha pontosvesszőt teszünk a sor végére, akkor az adatokat sem írja ki a Matlab. Az értékek kiírhatók a változó nevének beírásával.

» **y**

Látható, hogy egy oszlopvektort kaptunk, amelynek elemei a mintavételezési pontokban felvett értékek. A mintavétel természetes, hiszen nem analóg számítógéppel dolgozunk, így csakis diszkrét időpontokban várhatjuk a szimuláció eredményét. Kérdés, hogy milyen időfelbontással dolgoztunk. Az ábrából látható, hogy a Matlab automatikusan a $0 \leq t \leq 6$ időtartományt választotta ki az átmeneti függvény megjelenítésére. A Matlab a rendszer dinamikai tulajdonságai (zérusai, pólusai) alapján választja meg az idővektort. Az y vektor elemeinek számából így már vissza tudunk következtetni a mintavételi időre.

» **n=length(y)**

$n = 109$

» **T=6/n**

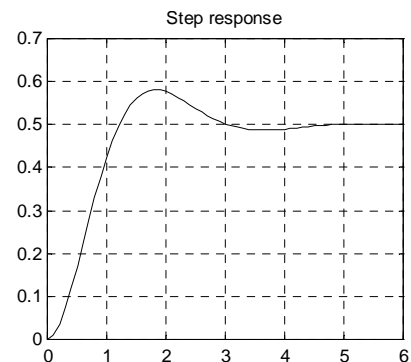
$T = 0.055$

A számított mintavételi idő: $T=6/109=0.055$ sec.

A *help* utasítás megmutatja az utasítás részletes használati módjait:

» **help step**

Természetesen többféleképpen megadható a fenti szimulációs futtatás meghívása. Az utasítás egy fontos alakja, amikor nem kívánjuk a Matlab automatikus szimulációs időközét és időtartamát választani, hanem



ezt rögzíteni kívánjuk a szimuláció végrehajtása előtt, pl. 10 sec-ig szeretnénk látni az átmeneti függvényt, mégpedig 0.1 sec felbontási idővel. Adjuk meg először a 0-tól 10-ig terjedő intervallumot 0.1-es lépésközökkel

```
»t=0:0.1:10
```

majd jelenítsük meg az átmeneti függvényt:

```
» y=step(num,den,t)
```

A kimenőjelet grafikusan megjeleníthetjük a *plot* utasítással:

```
» plot(t,y);
```

A könnyebb értelmezhetőség érdekében egy koordináta hálót is rajzolhatunk az ábrára a *grid* utasítással.

```
» plot(t,y),grid;
```

A *plot* utasítás a pontok között lineáris interpolációt alkalmaz, azaz összeköti a pontokat egy egyenessel. Az egyes pontok összekötése csak a vizuális megjelenítést (végül is folytonos rendszer folytonos kimenetét szeretnénk látni) szolgálja, valójában nincs pontosan kiszámított információnk a kimenőjel két szimulációs pont közötti viselkedését illetően. Ezt az interpolációt elhagyhatjuk, ha pontokkal ábrázoljuk a mintavételi értékeket.

```
» plot(t,y,'.');
```

Ezeket az *y* értékeket további számításokra is fel lehet használni. Az átmeneti függvény maximális értékét megkaphatjuk a *max* függvény hívással.

```
» ym=max(y)
```

```
ym = 0.5815
```

Stacionárius értékét a *dcgain* utasítás adja meg.

```
» ys=dcgain(num,den)
```

```
ys = 0.5
```

A százalékos túllüvést ezekből ki tudjuk számolni.

```
» yovrsht=(ym-ys)/ys*100
```

```
yovrsht = 16.2971
```

Inverz Laplace Transzformáció:

Analitikus vizsgálatokhoz szükséges az inverz Laplace transzformáció számítása. A módszer lényege, hogy a függvény Laplace transzformáltját olyan tagok összegére bontjuk fel, amelyeknek inverz transzformáltját már ismerjük. A leggyakrabban előforduló tagok:

$$k \xrightarrow{L^{-1}} k1(t)$$

$$\frac{r}{s+p} \xrightarrow{L^{-1}} re^{-pt}$$

$$\frac{r}{(s+p)^2} \xrightarrow{L^{-1}} rte^{-pt}$$

Ezt az alakot részlettörtekre bontással lehet meghatározni, amit a *residue* utasítással lehet végrehajtani.

Legyen a feladat az

$$Y(s) = \frac{3s^2 + 13s + 16}{(s+2)(s+3)^2}$$

valós törttel a Laplace tartományban leírt jel visszatranszformálása az időtartományba, azaz *y(t)* analitikus meghatározása.

Először adjuk meg a jel Laplace transzformáltját számláló és nevező polinomjával:

```
» num=[3 13 16];
```

```
» den=poly([-3 -3 -2]);
```

A részlettörtekre való bontás:

```
» [r,p,k]=residue(num,den)
```

```
r = 1.0000
```

```

-4.0000
 2.0000
p = -3.0000
     -3.0000
     -2.0000
k = []

```

A eredmény a Laplace tartományban (komplex frekvencia tartományban):

$$Y(s) = \frac{r(1)}{s-p(1)} + \frac{r(2)}{(s-p(2))^2} + \frac{r(3)}{s-p(3)} + k = \frac{1}{s+3} - \frac{4}{(s+3)^2} + \frac{2}{s+2}$$

és az időtartományban:

$$y(t) = e^{-3t} - 4te^{-3t} + 2e^{-2t}, \quad t \geq 0$$

Figyeljük meg a kettős pólusnak megfelelő struktúrát mind az r és p vektorban, mind a részlettörtes Laplace transzformált alakban, mind pedig az időtartománybeli kifejezésben.

Az analitikus kifejezés alapján természetesen elő tudjuk állítani az időfüggvényt:

```

» t=0:0.05:6;
» y=r(1)*exp(p(1)*t)+r(2)*t.*exp(p(2)*t)+r(3)*exp(p(3)*t);

```

(A második kifejezésben a t melletti pont azt jelöli, hogy a műveletet nem a vektorra, hanem annak elemeire értelmezzük). Ugyanezt egyszerűbben is meg lehet határozni numerikusan.

```

» yi=impz(num,den,t);
» plot(t,[y,yi]),grid;

```

LTI modell struktúra (sys):

Az egyszerűbb használat érdekében a Control System Toolbox bevezeti az LTI adatstruktúrát (Linear Time Invariant system). Egy LTI struktúrában egy lineáris rendszert háromféle alakban lehet megadni.

- Átviteli függvény alak: $H_{tf}(s) = \frac{s^m + b_{m-1}s^{m-1} + \dots + b_2s^2 + b_1s + b_0}{a_n s^n + a_{n-1}s^{n-1} + \dots + a_2s^2 + a_1s + a_0} = \frac{2}{s^2 + 3s + 2}$
- Zérus-pólus-erősítés alak: $H_{zpk}(s) = k \frac{(s-z_1)(s-z_2)\dots(s-z_m)}{(s-p_1)(s-p_2)\dots(s-p_n)} = \frac{2}{(s+1)(s+2)}$
- Állapotteres alak: $\dot{\mathbf{x}} = \mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{u}$
 $\mathbf{y} = \mathbf{C}\mathbf{x} + \mathbf{D}\mathbf{u}$ $\mathbf{A} = \begin{bmatrix} -3 & -1 \\ 2 & 0 \end{bmatrix}, \mathbf{B} = \begin{bmatrix} 1 \\ 0 \end{bmatrix}, \mathbf{C} = [0 \quad 1], \mathbf{D} = 0.$

A Matlab-ban a *tf*, *zpk* és *ss* utasításokkal lehet megadni egy lineáris rendszert LTI sys struktúrában.

Legyen a rendszer átviteli függvénye $H(s)$.

```

» num=2
» den=[1, 3, 2]
» H=tf(num,den)
Transfer function:
      2
-----
s^2 + 3 s + 2

```

vagy közvetlenül:

```

» Htf=tf(2,[1, 3, 2])

```

A többi modell hasonlóan adható meg:

```

» Hzpk=zpk([], [-1, -2], 2)
Zero/pole/gain:

```

2

(s+1) (s+2)

```

» A=[-3, -1; 2, 0]; B=[1; 0]; C=[0, 1]; D=0;
» Hss=ss(A,B,C,D);

```

A modellek átkonvertálhatók egymásba.

```

» Hzpk1=zpk(Htf)
» Hss1=ss(Htf)
» Htf1=tf(Hss)

```

A modellek számos paramétert (properties) foglalnak magukban. Ezeket a *get* utasítással nézhetjük meg.

```

» get(Htf)
» get(Hzpk)
» get(Hss)

```

Az LTI modell paramétereit megkaphatjuk a *tfdata*, *zpkdata*, *ssdata* utasításokkal. A 'v' flag szükséges, hogy vektor alakban kapjuk meg a paramétereit.

```

» [num1,den1]=tfdata(Htf,'v')
num1 = 0      0      2
den1 = 1      3      2
» [z,p,k]=zpkdata(Hzpk,'v')

```

A modell paraméterek közvetlenül is elérhetők a nevükön keresztül:

```

» num2=Htf.num{1}
num2 = 0      0      2

```

Az LTI adatstruktúra néhány paramétert ún. *cell array* típusban tárol, hogy több bemenetű, több kimenetű (MIMO, multi- input multi-output) rendszerek esetén is használható legyen.

A *cell array* egy olyan mátrix, melynek elemei mátrixok. Egy *cell array* például a következőképpen adható meg.

```

» ca={1, [1,2], [1,2,3]}
ca = [1]      [1x2 double]      [1x3 double]
» ca{2}
ans = 1      2

```

A {1} jelöléssel hivatkozunk a *cell array* első elemére. Egy bemenetű, egy kimenetű (SISO single-input single-output) rendszerre mindig {1} jelölést kell használni, mivel csak az első elem létezik. A *num*, *den*, *z*, *p* paraméterek *cell array* formátumban adottak. A *get(...)* utasítással lehet megnézni a paraméterek típusait.

A többi modell paraméter hasonló módon érhető el:

```

» Hzpk=zpk([], [-1, -2], 2)
2

```

(s+1) (s+2)

```

» Hzpk.p{1}
» Hss.a

```

```

ans = -3      -1
      2      0 .

```

Szimbólikus adatbevitellel tovább egyszerűsíthető az adat megadás. Az *s* a Laplace transzformációs változó.

```

» s=zpk('s')
» H=1/((s+1)*(s+2))

```

Aritmetikai műveletek is alkalmazhatók az LTI struktúrákra. A leggyakrabban használatos műveletek: +, -, *, /, \, ', inv, ^ . Például egy zárt rendszer eredő átviteli függvénye kiszámítható közvetlenül az alábbi szimbólikus összefüggéssel:

```

» Hcl=H/(1+H)

```

Az LTI sys struktúrák egy hierarchikus sorrendben helyezkednek el: tf -> zpk -> ss. Ha egy utasításban vagy számításban különböző modellek szerepelnek, akkor az eredmény mindig a magasabb hierarchiájú modell formájában keletkezik. Például a

```
» Htf*Hzpk
```

utasítás eredménye *zpk* alakban jelenik meg.

Ha az *s* változót **s=zpk('s')** formában adjuk meg, akkor az ezzel megadott átviteli függvényeket *zpk* alakban kapjuk. Ha az *s* változót **s=tf('s')** formában adjuk meg, akkor az ezzel megadott átviteli függvényeket *tf*, azaz polinom/polinom alakban kapjuk.

Időtartománybeli vizsgálat:

A Control System Toolbox több utasítást tartalmaz, amelyek segítségével a lineáris rendszerek időtartománybeli viselkedését lehet vizsgálni. Vizsgáljuk ismét a következő rendszert: $H(s) = \frac{2}{s^2 + 2s + 4}$

```
» H=2/(s^2+2*s+4)
```

Vegyünk fel egy időtartományt:

```
» t=0:0.1:10;
```

- **Átmeneti függvény (step response):** Az átmeneti függvény a rendszer kimenőjele egységugrás (step) bemenőjel esetén. A *step* utasítás előzőekben tárgyalt alakjai mellett az alábbi alak is hívható:

```
» step(H);
```

- **Súlyfüggvény (impulse response):** A súlyfüggvény a rendszer kimenete Dirac-delta bemenet esetén.

```
» impulse(H);
```

```
» yi=impulse(H,t);
```

```
» plot(t,yi)
```

- **Kezdeti feltétel (initial condition):** A rendszer viselkedése kezdeti feltétel esetén az *initial* utasítás segítségével vizsgálható. Csak állapotterezs reprezentációra alkalmazható ez az utasítás.

```
» H=ss(H)
```

```
» x0=[1, -2]
```

```
» [y,t,x]=initial(H,x0);
```

```
» plot(t,y),grid
```

Vegyünk észre, hogy az *x* változó egy mátrix, amelynek annyi oszlopa van, ahány állapotváltozója van a rendszernek (ebben az esetben 2). Sorainak száma megegyezik az idővektor elemeinek számával. Ellenőrizzük:

```
» size(x)
```

```
ans = 109      2
```

A rendszer állapottrajektóriáját is tudjuk vizsgálni. Az *x* mátrix első oszlopa tartalmazza az első állapotváltozót, a második pedig a másodikat. A **:** kiválasztja az adott vektor minden egyes elemét. Az állapottrajektória az egyik állapotváltozót a másik állapotváltozó függvényében ábrázolja.

```
» x1=x(:,1); x2=x(:,2);
```

```
» plot(x1,x2)
```

- **Tetszőleges bemenet:** A rendszer kimenetét tetszőleges bemenet esetén is ki lehet számítani.

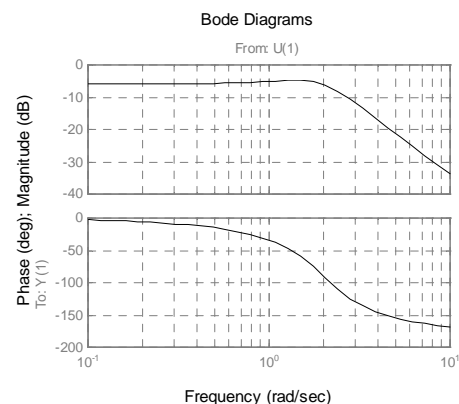
Határozzuk meg a kimenetet $u(t) = 2\sin(3t)$ bemenet esetén.

```
» usin=2*sin(3*t);
```

```
» ysin=lsim(H,usin,t);
```

Ábrázoljuk a bemenőjelet piros, a kimenőjelet pedig kék görbével.

```
» plot(t,usin,'r',t,ysin,'b'),grid;
```



Frekvenciatartománybeli vizsgálat:

A rendszer viselkedése a frekvenciatartományban is vizsgálható.

- **Bode diagram** A rendszer Bode diagramját a *bode* utasítással számíthatjuk. Számos módon alkalmazhatjuk ezt az utasítást. Egy megadott frekvencián kiszámolhatjuk a rendszer erősítését és fázisszögét. Számítsuk ki az erősítést az $\omega = 5$ frekvencián.

```
» w=5;
» [gain,phase]=bode(H,w);
```

Az eredmény: $gain = 0.0860$, $phase = -154.5367$

A számítás megismételhető különböző frekvenciákon. A *bode* utasítás azonban meghívható úgy is, hogy egyidejűleg több frekvencián is kiszámítja a frekvenciafüggvény abszolút értékét és fázisszögét.

A Bode diagram közvetlenül grafikusán is megjeleníthető. Ebben az esetben a Matlab automatikusan felvesz egy frekvenciatartományt a rendszer dinamikus tulajdonságai alapján (a *step* utasításhoz hasonlóan).

```
» bode(H),grid
```

A frekvencia pontok logaritmikus skálán helyezkednek el, mivel ez szemléletesebb, nagyobb frekvenciatartományt átfogó megjelenítést biztosít. A frekvencia vektor közvetlenül is meghatározható:

```
» w=logspace(-1,1,200);
```

A *logspace* utasítás létrehoz egy 200 pontból álló logaritmikusan egyenlő távolságra elhelyezkedő pontokból álló frekvencia vektort a 0.01 and 100 közti frekvenciatartományban. Ezzel a Bode diagram értékei már kiszámíthatók és ábrázolhatók.

```
» [gain,phase]=bode(H,w);
```

- **Nyquist diagram** : A rendszer erősítését és fázistolását ábrázoljuk a komplex síkon úgy, hogy a frekvencia változik.
- » `nyquist(H)` ;
- **Margin** : ez az utasítás kiértékeli a frekvenciafüggvény jellemzőit, kiszámítja a stabilitásra jellemző mutatókat.
- » `margin(H)` ;

- **Zérusok, pólusok:**

A karakterisztikus polinom (az átviteli függvény nevezője) gyökei

a rendszer pólusai.

```
» [num,den]=tfdata(H,'v');
» poles=roots(den);
```

A zérusok az átviteli függvény számlálójának a gyökei.

```
» zeros=roots(num);
```

A zérusok és pólusok megkaphatók közvetlenül a *zpk* modellből:

```
» [z,p,k]=zpkdata(H,'v');
```

A zérusok és pólusok ábrázolhatók a komplex síkon.

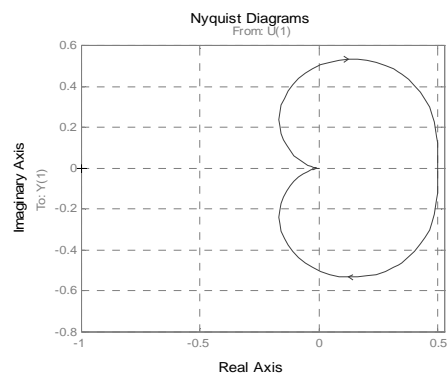
```
» subplot(111);
» pzmap(H);
```

A *damp* utasítás megadja az összes pólust és komplex póluspár esetén a csillapítási tényezőt és a sajátfrekvenciát is:

```
» damp(H);
```

A rendszer DC (egyenáramú) erősítése meghatározható:

```
» K=dcgain(H);
```



LTI Viewer:

Egy lineáris rendszer egyszerűen vizsgálható az *LTI Viewer* segítségével. Az *LTI Viewer* egy grafikus felület egy rendszer idő- és frekvenciatartománybeli tulajdonságainak vizsgálatára. Több rendszer egyszerre is vizsgálható a különböző menüpontok és a jobb egér billentyű segítségével.

» `ltiview`

vagy konkrétan,

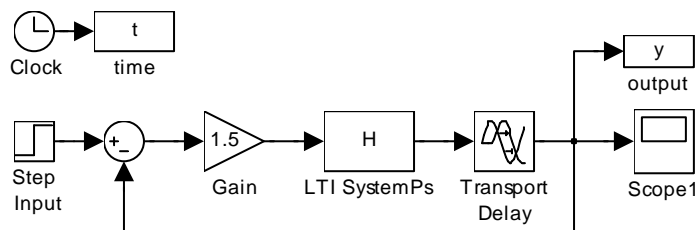
» `ltiview('bode',H);`

Simulink:

A Simulink egy grafikus programcsomag, amely blokk diagramon alapuló szimuláció végrehajtására képes. Egy Simulink szimuláció két lépésből áll. Először létrehozuk a modellt, majd futtatásokat végzünk a kívánt paraméterbeállítások mellett. A Simulink-ben egy dinamikus rendszert blokk diagrammal lehet megadni. A rendszer definiálása a blokk diagram megrajzolását jelenti. A blokkokat nem kell külön előállítani, egy könyvtárból ki lehet őket másolni. A könyvtár blokkjai működésük szerint külön alkönyvtárakban helyezkednek el. A Simulink blokk könyvtár megnyitható a Matlabból a *simulink* utasítással, vagy a Simulink ikonnal. Egy új modellt létre lehet hozni **File -> New** menü aktiválásával. Ez létrehoz egy új ablakot, amelybe a blokkok már bemásolhatóak a könyvtárból. Az egér bal gombjának lenyomva tartásával lehet a könyvtári blokkokat áthúzni a fájlunkba. A blokkokat össze tudjuk kötni az egér bal gombjával, rákliccelünk a blokkok bemenetére vagy kimenetére, majd a vonalat odahúzzuk a kívánt helyre. Egy blokk duplikálható a jobb egér billentyű klikkelésével. A blokkokat méretezni és elforgatni is lehet. A blokk paramétereit dupla klikkeléssel lehet megváltoztatni. Az új fájl elmenthető a **File->Save** menüponttal.

A szimuláció elindítható a **Start -> Simulation** menüvel, vagy a Run ikonnal (▶). A szimuláció beállítási paramétereit megváltoztathatjuk. A futtatási eredményeket a Scope blokkal lehet megvizsgálni vagy a To Workspace blokkal visszaküldeni a Matlab felületre. A Simulink a modellek szimulációját közönséges differenciálegyenletek integrálásával végzi. A Simulink számos integrálási módszer használatát teszi lehetővé. Pontos eredmények elérésére a megfelelő módszer kiválasztása is egy fontos szempont lehet.

A Simulink működésének megismerése céljából vegyünk egy egyszerű példát. Hozzunk létre egy új fájlt és másoljuk be a különféle blokkokat. A blokkok paramétereit változtassuk meg a kívánt értékre. Állítsuk be a menüből a Simulation->Parameters->Stop time paramétert 50-re. A Simulink a Matlab-ban definiált változókat használja.



$H(s)$: Control System Toolbox ->LTI system : H

Különbségképző: Simulink->Math->Sum: +-

Késleltetés, holtidő: Simulink->Continuous->Transport Delay: 1

Erősítés: Simulink->Math->Gain: 1.5

Step input: Simulink->Sources->Step

Scope: Simulink->Sinks->Scope

Clock: Simulink->Sources->ClockOutput, time: Simulink->Sinks->To Workspace: y,t

Változtassuk a Gain paraméter értékét 0.5 és 2 között. Állapítsuk meg, a paraméter milyen értékénél lépnek fel a rendszerben állandósult lengések.

A *Scope* blokk segítségével is visszaküldhetjük a szimuláció eredményét a Matlab felületre. A *Scope* grafikus ablak paramétereit állítsuk be a következőképpen:

A 'properties' menü alatt.

Data history: Save data to workspace

Variable name: ty (tu for the control signal)

Matrix format

Így a t idő és az y kimenőjel vektorai egyszerűen kinyerhetők a szimuláció után, ezekből pedig a minőségi jellemzők meghatározhatók (túllövés, beállási idő, maximális beavatkozó jel, stb.).

```
» t=ty(:,1)
```

```
» y=ty(:,2)
```

```
» plot(t,y),grid
```

3. A frekvenciafüggvény

Stabilis lineáris rendszerek egy alapvető tulajdonsága, hogy szinuszos bemenőjelekre *állandósult állapotban*, a tranziensek lecsengése után a bemenőjel frekvenciájával megegyező frekvenciájú szinuszos jelekkel válaszolnak. Legyen

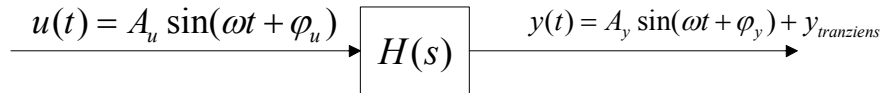
$$u(t) = A_u \sin(\omega t + \varphi_u)$$

$t \geq 0$ esetén a rendszerünk bemenete. A kimenőjel

$$y(t) = y_{\text{állandósult}}(t) + y_{\text{tranziens}}(t)$$

A kimenőjel állandósult (kvázistacionárius) állapotban

$$y_{\text{állandósult}}(t) = A_y \sin(\omega t + \varphi_y)$$



A frekvenciafüggvény az A_y/A_u amplitúdó arány és a $(\varphi_y - \varphi_u)$ fáziskülönbség frekvenciafüggését leíró függvény. Mivel a frekvenciafüggvény szimultán módon két rendszerjellemező tulajdonság frekvenciafüggését kívánja reprezentálni, nem meglepő, hogy a *frekvenciafüggvény komplex függvény*. Bebizonyítható, hogy *formailag* a frekvenciafüggvényt az átviteli függvényből az $s=j\omega$ helyettesítéssel lehet származtatni.

$$H(j\omega) = H(s)|_{s=j\omega} = M(\omega)e^{j\varphi(\omega)}$$

A frekvenciafüggvény kifejezésében $M(\omega)$ az *amplitúdófüggvény* (a frekvenciafüggvény abszolút értéke), $\varphi(\omega)$ pedig a *fázisfüggvény*:

$$M(\omega) = |H(j\omega)| = \frac{A_y(\omega)}{A_u(\omega)} \quad \varphi(\omega) = \arg\{H(j\omega)\} = \varphi_y(\omega) - \varphi_u(\omega)$$

A frekvenciafüggvény többféleképpen is megjeleníthető. Az $M(\omega)$ és $\varphi(\omega)$ függvényeket külön-külön ábrázolhatjuk egy kijelölt frekvenciatartományban. Ez a technika a *Bode* diagramot eredményezi. A frekvenciaskála léptéke rendszerint logaritmikus. A *Nyquist* diagram a frekvenciafüggvényt a komplex számsíkon ábrázolja. A kiválasztott frekvenciatartomány minden egyes ω értékére a komplex síkban az $M(\omega)$ és $\varphi(\omega)$ értékpárnak megfelelő pontot adhatunk meg. E pontok kontúrral való összekötése eredményezi a Nyquist diagramot. A Nyquist diagram ábrázolásakor a frekvenciát rendszerint nulla és végtelen között változtatjuk.

Példa: Legyen egy rendszer átviteli függvénye

$$H(s) = \frac{10}{s^2 + 2s + 10}$$

Határozzuk meg a rendszer kimenőjelét, ha bemenőjele $u(t) = 2 \cdot \sin(3 \cdot t)$.

```

» num=10;
» den=[1, 2, 10];
» H=tf(num,den);
» t=0:0.05:10;
» u=2*sin(3*t);
» y1=lsim(H,u,t);

```

Ábrázoljuk a rendszer bemenőjelét és kimenőjelét egy diagramban (bemenőjel: piros (red), kimenőjel: kék (blue)):

```
» plot(t,u,'r',t,y1,'b'), grid;
```

Állandósult állapotban, a tranziens lecsengése után a kimenőjel szinuszos, frekvenciája megegyezik a bemenőjel frekvenciájával, amplitúdója és fázisszöge függ a bemenőjel frekvenciájától. Az amplitúdó és a fázisszög kiszámítható a $H(s=j\omega)$ frekvenciafüggvényből. A Matlab-ban a *bode* utasítás alkalmazható ezen értékek kiszámítására egy adott frekvencián vagy egy adott frekvenciatartományban. Például az $\omega = 3$ frekvencián:

```
» [gain,phase]=bode(H,3);
```

Vizsgáljuk a rendszer válaszát az alábbi bemenőjelekre: $u(t) = \sin(\omega t)$, $\omega_1 = 1$, $\omega_2 = 2$, $\omega_3 = 5$, $\omega_4 = 10$.

```
» w1=1; w2=2; w3=5; w4=10;
» u1=sin(w1*t); u2=sin(w2*t); u3=sin(w3*t); u4=sin(w4*t);
» y1=lsim(H,u1,t); y2=lsim(H,u2,t); y3=lsim(H,u3,t);
» y4=lsim(H,u4,t);
» plot(t,y1,'r',t,y2,'g',t,y3,'b',t,y4,'m'), grid;
```

Számítsuk ki az amplitúdót és a fázisszöget a megadott frekvenciákra:

```
» [gain1,phase1]=bode(H,w1);
» [gain2,phase2]=bode(H,w2);
» [gain3,phase3]=bode(H,w3);
» [gain4,phase4]=bode(H,w4);
```

Készítsünk táblázatot a fenti frekvenciák, amplitúdók és fázisszögek értékeivel.

Az amplitúdó és fázisértékek egyszerűbben számíthatók, ha a frekvenciákat vektorban adjuk meg. Hozzunk létre egy lineáris frekvencia vektort.

```
» w=1:0.1:10;
» [gain,phase]=bode(num,den,w);
```

Az amplitúdó és a fázisdiagramok egyidejűleg a képernyő különböző ablakaiban jeleníthetők meg a subplot parancs segítségével. (A subplot(211) parancs 2 x 1 ablakot hoz létre a képernyőn és ezek közül az elsőre hivatkozik.)

```
» subplot(211), plot(w,gain)
» subplot(212), plot(w,phase)
```

E két görbe alkotja a rendszer Bode diagramját.

```
» subplot(111) % visszaállítjuk az egy ablakos ábrázolást.
```

A Nyquist diagram a komplex számsíkon ábrázolja a frekvenciafüggvény pontjait.

```
» clf % clear figure, törli az előző ábrát.
» re=real(gain.*exp(j*phase*pi/180));
» im=imag(gain.*exp(j*phase*pi/180));
» plot(re,im)
```

Az adott frekvenciavektor pontjaihoz a frekvenciafüggvény valós és képzetes részei közvetlenül is számíthatók a *nyquist* parancssal.

```
» [re,im]=nyquist(num,den,w)
» plot(re,im)
```

A frekvenciafüggvényt közvetlenül is ábrázolhatjuk.

Megjelenítés: Ha a *bode* parancsot bal oldali változók megadása nélkül hívjuk meg, a diagram megjelenik a képernyőn, de a számított adatok nem őrződnek meg.

```
» clf % törli az ábra ablakot.
» bode(H), grid
```

Figyeljük meg, hogy mind a frekvencia, mind pedig az amplitúdó léptéke logaritmikus. Hasonlóan a Nyquist diagram az alábbi utasítás meghívásával jeleníthető meg:

```
» nyquist(H)
```

A frekvenciafüggvény pontjait előzetesen kiszámíthatjuk, majd azután ábrázolhatjuk.

Számítás: Az amplitúdó és fázisértékeket közvetlenül kiszámíthatjuk, ha az utasításokat bal oldali változók megadásával hívjuk meg. A számításhoz az átviteli függvényt számláló és nevező polinomjaival adjuk meg (*num, den*), mivel az LTI struktúra egy 3 dimenziós elrendezést (array) hoz létre.

» **[gain, phase, w]=bode(num, den)**

Ebben az esetben a Matlab számítja a frekvenciavektort is. (Egy Matlab utasításban a jobb oldali változók bemeneti, a bal oldali változók kimeneti változók.)

Ábrázoljuk az amplitúdót kétszeresen logaritmikus léptékben a Matlab *loglog* utasításának meghívásával. Ez az utasítás hasonló a *plot* utasításhoz, de a megjelenítéshez mind az x, mind az y tengelyen logaritmikus skálát használ.

» **loglog(w, gain), grid**

Alkalmazzuk ismét a subplot utasítást az amplitúdó és a fázisszög megjelenítéséhez.

» **subplot(211)**

» **loglog(w, gain)**

» **subplot(212)**

» **semilogx(w, phase)**

Logaritmikus frekvencia vektort közvetlenül a *logspace* utasítással hozhatunk létre.

» **w=logspace(-2, 2, 100)**

Ez az utasítás 100 logaritmikusan ekvidisztáns pontot számol a $10^{-2}=0.01$ és $10^2=100$ alsó és felső frekvenciaértékek között. (Alapértelmezésben, a harmadik változó megadása nélkül az utasítás 50 pontot számol.) Ezzel a frekvenciavektorral számítsuk újra a Bode diagramot.

» **[gain, phase]=bode(num, den, w)**

» **subplot(111)**

» **loglog(w, gain)**

Vegyük észre, hogy a frekvenciatartomány eltér az előbbitől.

Feladat:

A rendszer átviteli függvénye: $H(s)=10/(s^2+2s+5)$

A bemenőjel: $u(t)=A*\sin(2t)$, $t \geq 0$

A kimenőjel kvázistacionárius állapotban: $y(t)=5*\sin(2t+phase)$

Határozzuk meg az *A* és *phase* paraméterek értékeit (alkalmazzuk a *bode* utasítást).

A Nyquist diagram alakja jellemzi a rendszert. Analizálva a Nyquist diagramot a rendszer fontos tulajdonságairól kaphatunk képet.

A Bode diagram egyik előnye rendszerek soros eredői frekvenciafüggvényeinek számításakor mutatkozik. A soros eredő a frekvenciafüggvények összesorzásával kapható meg. A logaritmikus lépték miatt a Bode diagramok egyszerűen összeadódnak. A Bode diagram másik előnye, hogy rendszerint jól közelíthető aszimptotáival. A közelítő diagramok jellegéből és töréspontjaiból a rendszer tulajdonságairól gyors értékelést adhatunk.

Az alábbi példa szemlélteti, hogy a közelítő Bode amplitúdó diagram széles frekvenciatartományban valóban jól közelíti a pontos görbét.

Pontos és közelítő Bode diagram:

Határozzuk meg az alábbi átviteli függvénnyel adott rendszer pontos és közelítő Bode amplitúdó diagramját.

$$H(s) = \frac{10}{s(1+s)(1+10s)^2}$$

A rendszer $H(s)$ átviteli függvénye ún. időállandós formában adott. Határozzuk meg erősítés-zérus-pólus alakban is.

$$H(s) = 0.1 \frac{1}{s} \frac{1}{(0.1+s)^2} \frac{1}{1+s}$$

```

» s=zpk('s')
» H=10/(s*(1+s)*(1+10*s)*(1+10*s))
» [num,den]=tfdata(H,'v')
» p=roots(den)

```

A rendszer pólusai: $0, -0.1, -0.1, -1$. A pólusok (a negatív előjel nélkül) adják a közelítő Bode diagram töréspontjait.

Vezessük be az alábbi frekvenciatartományokat: $w1:0.02-0.1$; $w2:0.1-1$; $w3:1-4$

```

» w1=logspace(log10(0.02),-1,10);
» w2=logspace(-1,0,10);
» w3=logspace(0,log10(4),10);

```

A teljes frekvenciatartományt a részterületek egyesítésével képezzük:

```

» w=[w1 w2 w3];

```

1. Számítsuk ki a közelítő amplitúdó értékeket az $\frac{1}{s}$ tényezőre.

```

» gain1=1./w;

```

2. Határozzuk meg a közelítő amplitúdó értékeket az $\frac{1}{(0.1+s)^2}$ tényezőre. A törésponti frekvenciánál

kisebb frekvenciákra (kisfrekvenciás közelítés) a tag az $\frac{1}{0.1^2}$ konstanssal közelíthető. A töréspont felett

(nagyfrekvenciás közelítés) a tag az $\frac{1}{s^2}$ átviteli

függvénnyel közelíthető.

```

» w2L=w1; w2H=[w2 w3];
» gain2L=1./((0.1+0*w2L).^2);
» gain2H=1./(w2H.^2);
» gain2=[gain2L gain2H];

```

3. Hasonlóan adjuk meg a közelítő amplitúdó

értékeket az $\frac{1}{1+s}$ tényezőre.

```

» w3L=[w1 w2]; w3H=w3;
» gain3L=0*w3L+1; gain3H=1./w3H;
» gain3=[gain3L gain3H];

```

A teljes rendszer erősítése:

```

» gainappr=0.1*gain1.*gain2.*gain3;

```

Számítsuk ki a pontos Bode diagram amplitúdó értékeit.

```

» gainexact=bode(num,den,w);

```

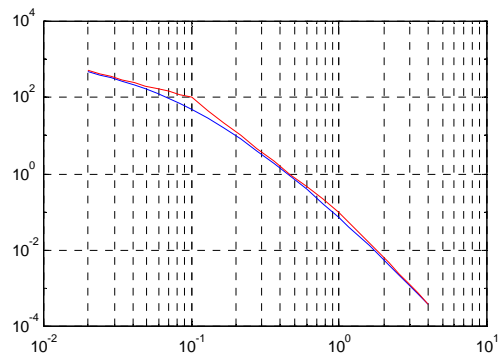
Ábrázoljuk a két görbét egy diagramban.

```

» loglog(w, gainexact,'b',w,gainappr,'r'),grid

```

Látható, hogy a közelítés elfogadható, az aszimptoták jól közelítik a pontos amplitúdó görbét. A legnagyobb az eltérés az $\omega = 0.1$ frekvencián, ami egy kettős pólushoz tartozó töréspont. Megjegyezzük, hogy az eltérés jóval nagyobb lehet gyengén csillapított konjugált komplex pólusok esetén ($\zeta < 0.3$).



4. Lineáris Rendszer Elemei

Egy lineáris rendszer ún. időállandós alakban általánosan az alábbi átviteli függvénnyel adható meg:

$$H(s) = \frac{k \prod_{j=1}^c (1 + s\tau_j) \prod_{j=1}^d (1 + 2\zeta_j \tau_{0j} s + s^2 \tau_{0j}^2)}{s^i \prod_{j=1}^e (1 + sT_j) \prod_{j=1}^f (1 + 2\zeta_j T_{0j} s + s^2 T_{0j}^2)} e^{-sT_D}$$

ahol k az átviteli vagy erősítési tényező, i az integrátorok száma, T_D a holtidő, τ és T időállandók, a ζ paraméterek a csillapítási tényezők.

Az alábbiakban a legfontosabb elemek idő- és frekvencia tartománybeli tulajdonságait vizsgáljuk. Ezek az elemek az arányos, integráló, differenciáló, holtidős és tárolós jellegű tagok, illetve ezek soros kapcsolásával adódó eredők.

1. Arányos (P) tag: $H(s) = k$

Az erősítési tényező k , a fázisszög pedig zérus minden frekvencián.

2. Integráló (I) tag: $H(s) = \frac{k}{s}$

$i=1$, a szakasz egyetlen integráló hatást tartalmaz. További időállandós tagok nincsenek.

Az integráló tag memória tulajdonságú, kimenetén a jel akkor lehet konstans, ha bemenetén a jel értéke zérus. A kimenőjel aktuális értéke a múltbeli bemenőjel értékeitől függ.

Vizsgáljuk a $\frac{k}{s}$ átviteli függvénnyel adott tiszta integráló tag viselkedését $k=1$ és $k=5$ erősítési tényezők mellett.

$$H_1(s) = \frac{1}{s}, \quad H_2(s) = \frac{5}{s}$$

```

» clear           % törölünk minden korábban definiált változót.
» s=zpk('s')     % definiáljuk az s szimbólikus változót zpk alakban.
» H1=1/s
» H2=5/s

```

Az átmeneti függvény (egységugrásra adott válasz):

```
» figure(1), step(H1,'r',H2,'g'), grid
```

Bode diagram:

```
» figure(2), bode(H1,'r',H2,'g'), grid
```

Nyquist diagram:

```
» figure(3), nyquist(H1,'r',H2,'g'), grid
```

Az átmeneti függvény végértékei:

```
» y1inf=dcgain(H1)
» y2inf=dcgain(H2)

```

Látható, hogy az átmeneti függvény végértéke végtelen. A frekvenciafüggvény amplitúdója kisfrekvencián végtelen, míg fázisszöge minden frekvencián -90° .

3. Egytárolós arányos (PT1) tag: $H(s) = \frac{k}{1+Ts}$

Adjuk meg az alábbi tag átmeneti függvényét, Bode és Nyquist diagramját.

$$H_1(s) = \frac{k}{1+Ts} = \frac{2}{1+10s}$$

Definiáljuk a rendszert.

» **H1=2/(1+10*s)**

illetve

» **H1=tf(2,[10, 1])**

Az átmeneti függvény:

» **t=0:0.1:50;**

» **y1=step(H1,t);**

» **plot(t,y1),grid**

vagy egyszerűen

» **step(H1)**

Vizsgáljuk a k és T paraméterek jelentését, hatását a rendszer válaszára.

A kimenőjel *állandósult értéke*, $y(t \rightarrow \infty)$ a Laplace transzformáció végértéktételével számítható.

$$y(t \rightarrow \infty) = \lim_{s \rightarrow 0} sY(s) = \lim_{s \rightarrow 0} sH(s)U(s)$$

ahol $U(s)$ a bemenőjel Laplace transzformáltja. Egységugrás bemenőjel esetén

$$y(t \rightarrow \infty) = \lim_{s \rightarrow 0} sH(s)U(s) = \lim_{s \rightarrow 0} sH(s) \frac{1}{s} = \lim_{s \rightarrow 0} H(s)$$

Ha $H(s) = H_1(s)$,

$$y(t \rightarrow \infty) = \left. \frac{2}{1+10s} \right|_{s=0} = 2$$

Matlab segítségével:

» **ylimf=dcgain(H1)**

Vizsgáljuk a T időállandó hatását a tranzien viselkedésre. Ismételjük meg az

» **y1=step(2,[T 1],t);**

Matlab utasítást különböző T értékekre.

A szakasz átviteli függvénye megadható *zérus-pólus* alakban is:

$$H_1(s) = \frac{k_p}{s-p} = \frac{2}{10(0.1+s)} = \frac{0.2}{s+0.1}$$

ahol

$$p = -\frac{1}{T}, \quad k_p = \frac{k}{T}$$

A pólus abszolút értéke megadja a közelítő amplitúdó-frekvencia diagram töréspontját (sarokfrekvencia).

Az átmeneti függvény állandósult értéke

$$y(t \rightarrow \infty) = \lim_{s \rightarrow 0} H(s)$$

megegyezik az amplitúdó-frekvencia függvény kisfrekvenciás értékével.

A tag *Bode* és *Nyquist* diagramja:

» **bode(H1);**

» **nyquist(H1);**

Hasonlóan vizsgáljuk az alábbi egytárolós, kéttárolós és háromtárolós arányos (PT1, PT2, PT3) tagok átmeneti és frekvenciafüggvényeit:

$$H_1 = \frac{2}{1+10s}; H_2 = \frac{2}{(1+10s)(1+2s)}; H_3 = \frac{2}{(1+10s)(1+2s)(1+0.1s)}$$

(1-piros, 2-zöld, 3- kék)

- » **H2=2/((1+10*s)*(1+2*s))**
- » **H3=2/((1+10*s)*(1+2*s)*(1+0.1*s))**

Átmeneti függvények:

- » **figure(1), step(H1,'r',H2,'g',H3,'b'),grid**

Bode diagramok:

- » **figure(2), bode(H1,'r',H2,'g',H3,'b'), grid**

Nyquist diagramok:

- » **figure(3), nyquist(H1,'r',H2,'g',H3,'b'),grid**

Megjegyezzük, hogy a görbék kijelölt része megnagyítható a figure ablakban aktivizált zoom paranccsal.

4. Integráló tárolós (IT) tag: $H(s) = \frac{k}{s(1+Ts)}$

$$H_1(s) = \frac{2}{s}, \quad H_2(s) = \frac{2}{s(1+10s)}, \quad H_3(s) = \frac{2}{s(1+10s)(1+2s)}$$

Az integráló taghoz sorbakapcsolt tárolós tagok csatlakoznak.

- » **H1=2/s**
- » **H2=2/(s*(1+10*s))**
- » **H3=2/(s*(1+10*s)*(1+2*s))**

Átmeneti függvények:

- » **figure(1), step(H1,'r',H2,'g',H3,'b'),grid**

Bode diagramok:

- » **figure(2), bode(H1,'r',H2,'g',H3,'b'),grid,**

Nyquist diagramok:

- » **figure(3), nyquist(H1,'r',H2,'g',H3,'b'),grid**

5. Kéttárolós lengő tag: $H(s) = \frac{1}{s^2 T_0^2 + 2\zeta T_0 s + 1}$

Vizsgáljuk az alábbi rendszert:

$$H(s) = \frac{1}{9s^2 + 2s + 1} = \frac{1}{s^2 T_0^2 + 2\zeta T_0 s + 1}$$

ahol $\omega_0 = \frac{1}{T_0}$ a természetes körfrekvencia és ζ a csillapítási tényező ($T_0 = \frac{1}{\omega_0} = 3$, $\zeta = 1/3$).

- » **num=1;**
- » **den=[9, 2, 1]**
- » **H=tf(num,den)**

A rendszer pólusai a *roots* utasítással számolva

- » **roots(den)**

illetve a *damp* utasítással:

- » **damp(H)**

A két konjugált komplex pólus megadható az alábbi alakban: $p_1 = a + jb$, $p_2 = a - jb$

Az átmeneti függvény v_t túllövése a következőképpen számítható:

$$\omega_0^2 = a^2 + b^2, \quad \zeta = -\frac{a}{b}, \quad v_t = e^{\frac{-\zeta\pi}{\sqrt{1-\zeta^2}}} = e^{-\frac{\pi a}{b}}$$

A lengési körfrekvencia:

$$\omega_p = b = \omega_0 \sqrt{1 - \zeta^2}$$

Az átmeneti függvény első maximumának időpontja (csúcsidő): $T_p = \frac{\pi}{\omega_p} = \frac{\pi}{b}$.

```
» zeta=1/3
» vt=exp(-zeta *pi/sqrt(1- zeta * zeta))
```

Az átmeneti függvény:

```
» [y,t]=step(H) ;
» plot(t,y) , grid
```

Az átmeneti függvény maximális értéke:

```
» ym=max(y)
```

Állandósult értéke:

```
» ys=dcgain(H)
```

A túllövés értéke másképp számítva:

```
» yo=(ym-ys)/ys
```

Vizsgáljuk az átmeneti függvényeket, a Bode és a Nyquist diagramokat különböző csillapítási tényezők mellett.

$\zeta = 0.5, 1, 2$.

```
» zeta1=0.5, zeta2=1, zeta3=2
» T0=3
» H1=1/(s*s*T0*T0+2*zeta1*T0*s+1)
» H2=1/(s*s*T0*T0+2*zeta2*T0*s+1)
» H3=1/(s*s*T0*T0+2*zeta3*T0*s+1)
```

Az átmeneti függvények:

```
» figure(1) , step(H1,'r',H2,'g',H3,'b') ,grid
```

A Bode diagramok:

```
» figure(2) , bode(H1,'r',H2,'g',H3,'b') ,grid
```

A Nyquist diagramok:

```
» figure(3) , nyquist(H1,'r',H2,'g',H3,'b') ,grid
```

A pólus-zérus konfigurációk:

```
» pzmap(H1)
» pzmap(H2)
» pzmap(H3)
```

Látható, hogy 0.3 csillapítási tényező mellett az átmeneti függvény a leghangosabb, a túllendülés a legnagyobb, és a Bode amplitúdó diagram kiemelése is a legnagyobb. A Nyquist diagram metszéke az imaginárius tengellyel ennél a csillapításnál a legnagyobb. A pólusok imaginárius értéke, ami a lengési körfrekvenciát meghatározza szintén itt a legnagyobb. A Bode amplitúdó diagram kiemelése összefüggésben van az átmeneti függvény túllendülésével. Ha el akarjuk kerülni az időtartományban a nagy túllendülést, nem engedünk meg a frekvenciatartományban nagy amplitúdó kiemelést. A $\zeta = 0.7$ csillapítási tényező kedvező viselkedést, gyors beállást és kis túllendülést mutat. Szabályozási rendszerek tervezhetők ilyen dinamikus viselkedésre.

6. Differenciáló (D és DT) tag: $H(s) = sT_d$

$$H_1(s) = sT_d = 2s, H_2(s) = \frac{2s}{1+10s}, H_3 = \frac{2s}{(1+10s)(1+2s)}, H_4 = \frac{2s}{(1+10s)(1+2s)(1+0.1s)}$$

Adjuk meg ismét a tag átmeneti függvényét, Bode és Nyquist diagramját.

```
» H1=2*s
```

```

» H2=(2*s)/(1+10*s)
» H3=(2*s)/((1+10*s)*(1+2*s))
» H4=(2*s)/((1+10*s)*(1+2*s)*(1+2*s))

```

$H_1(s)$ átmeneti függvénye:

```

» figure(1), step(H1)

```

A Matlab nem tudja kiértékelni a rendszer viselkedését, mivel a tag nem realizálható, számlálója magasabbfokú a nevezőjénél. (Az átmeneti függvény Dirac delta).

Átmeneti függvények:

```

» figure(1), step(H2,'r',H3,'g',H4,'b'),grid

```

Bode diagramok:

```

» figure(2), bode(H2,'r',H3,'g',H4,'b'),grid,

```

Nyquist diagramok:

```

» figure(3), nyquist(H2,'r',H3,'g',H4,'b'),grid

```

7. A zérusok hatása: $H(s) = \frac{k(s-z_1)(s-z_2)\dots(s-z_M)}{D(s)}$

Vizsgáljuk a zérus hatását az átmeneti függvényre és a frekvenciafüggvényre az alábbi átviteli függvény esetén:

$$H(s) = \frac{1 + \tau s}{(1 + s)(1 + 10s)}$$

τ , a számláló időállandója (a zérus értéke $-1/\tau$) -4 és 4 értékek között változik. Pozitív (a komplex számsík jobb oldalán lévő) zérus esetén a tagot nem-minimumfázisúnak nevezzük.

```

» s=tf('s')
» tau=[-4 -2 0 2 4];
» D=10*s*s+11*s+1;
» for i=1:5,
»     H(i)=(s*tau(i)+1)/D
» end
» t=0:0.1:60;
» Y1=step(H(1),t); Y2=step(H(2),t); Y3=step(H(3),t);
» Y4=step(H(4),t); Y5=step(H(5),t);
» plot(t,[Y1,Y2,Y3,Y4,Y5]),grid,shg

```

Megjegyezzük, hogy a nem-minimumfázisú rendszerek szokatlan viselkedést mutathatnak, például egy jobb oldali zérus esetén az átmeneti függvény kezdetben az állandósult értékével ellentétes irányban indul el, majd irányt váltva éri el végül állandósult értékét. A zérus beiktatása gyorsítja a rendszert.

Bode diagramok:

```

» figure(1)
» bode(H(1),'r',H(2),'g',H(3),'k',H(4),'y',H(5),'b'),grid

```

Nyquist diagramok:

```

» figure(2)
» nyquist(H(1),'r',H(2),'g',H(3),'k',H(4),'y',H(5),'b'),grid

```

Értékeljük a zérus hatását a frekvenciatartományban a Bode és a Nyquist diagramok alakulására!

8. Holtidős tag: $H(s) = e^{-sT_D}$

A holtidős tag leírása az idő- és a Laplace operátor tartományban:

$$y(t) \rightarrow y(t - T_D)$$

$$Y(s) \rightarrow Y(s)e^{-sT_D}$$

Az átviteli függvény:

$$H_D(s) = H(s)e^{-sT_D}$$

A frekvenciatartományban:

$$\left| e^{-j\omega T_D} \right| = 1, \quad \arg\{e^{-j\omega T_D}\} = -\omega T_D$$

$$\text{erősítés: } |H_D| = |H|$$

$$\text{fázisszög: } \arg(H_D) = \arg(H) - \omega T_D$$

Vizsgáljuk az alábbi átviteli függvényű tagok frekvenciafüggvényeit:

$$H_1(s) = \frac{1}{1+10s}, \quad H_2(s) = \frac{1}{(1+10s)} e^{-2s}$$

A holtidős tag amplitúdója és fázisszöge:

$$|H_1| = |H_2|, \quad \text{gain2} = \text{gain1}$$

$$\arg(H_2) = \arg(H_1) - \omega T_D, \quad \text{phase2} = \text{phase1} - \omega T_D$$

```

» Td=2
» num1=1
» den1=[10, 1]

```

A Bode diagram számításakor most a *num*, *den* számláló és nevezőpolinommal adott alakot kell használni. Először hozzuk létre a frekvenciavektort.

```

» w=logspace(-2, 0, 100)
» [gain1, phase1]=bode(num1, den1, w)
» delay=180/pi*Td*w'           % a holtidős tag  $-\omega T_D$  radiánról fokra számítjuk át.

```

Az amplitúdó és a fázisszög a holtidő figyelembe vételével:

```

» gain2=gain1
» phase2=phase1-delay
» subplot(211), loglog(w, gain1, 'r', w, gain2, 'b'), grid;
» subplot(212), semilogx(w, phase1, 'r', w, phase2, 'b'), grid

```

A fázisszög linearitása jól látható, ha ábrázolására a *semilogx* utasítás helyett a lineáris léptékű *plot* utasítást használjuk.

```

» figure(2), subplot(111), plot(w, phase1, 'r', w, phase2, 'b'), grid

```

Ábrázoljuk a Nyquist diagramot. Számítsuk ki először a valós és a képzetes értékeket.

```

» h1= gain1.*exp(j*phase1*pi/180)
» h2= gain2.*exp(j*phase2*pi/180)
» plot(real(h1), imag(h1), 'r', real(h2), imag(h2), 'b')

```

A nagyfrekvenciás viselkedés jobban vizsgálható, ha nagyobb frekvenciatartományt jelölünk ki a *logspace* utasítással.

Az időtartománybeli viselkedés Simulink blokk-diagram megépítésével és a program futtatásával követhető a legjobban, mivel a holtidős tag a Simulink program építőeleme.

A holtidős tag átviteli függvénye nem racionális törtfüggvény, a Matlab program csupán közelíteni tudja egy nem-minimumfázisú racionális törtfüggvénnyel, az ún. *Pade* közelítéssel. A holtidős átviteli függvény és a *Pade* közelítés Taylor sorának első elemei megegyeznek. Minél magasabbfokú a törtfüggvény, annál jobb a közelítés. A közelítő átmeneti függvény zérus helyett +1 vagy -1 értékből indul. A Matlab-ban a *pade* utasítás szolgál a közelítés megadására.

A *Pade* közelítés szemléltetésére alkalmazzunk ötödfokú közelítést.

$$H_{\text{deadtime}}(s) = e^{-sT_D}$$

- » `H1=tf(num1,den1)`
- » `[numpade,denpade]=pade(Td,5)`
- » `Hdeadtime=tf(numpade,denpade)`
- » `H2=H1*Hdeadtime`

Az átmeneti függvény:

- » `figure(1), step(H1,'r',H2,'g'),grid`

A Bode diagram:

- » `figure(2), bode(H1,'r',H2,'g'),grid`

A Nyquist diagram:

- » `figure(3), nyquist(H1,'r',H2,'g'),grid`

Feladat: Vizsgáljuk a *Pade* közelítés jóságát a fenti egytárolós holtidős tag esetén. Építsünk egy Simulink programot a “transport delay” blokk, illetve a *Pade* közelítő ötödfokú racionális törtfüggvénnyel. Hasonlítsuk össze az átmeneti függvényeket.

9. Kettős integrátor: $H(s) = \frac{k}{s^2}$

$$H_1(s) = \frac{4}{s^2}, H_2(s) = \frac{4}{s^2(1+10s)}, H_3 = \frac{4}{s^2(1+10s)(1+2s)}, H_4 = \frac{4}{s^2(1+10s)(1+2s)(1+0.1s)}$$

Feladat: Határozzuk meg a megadott tagokra az átmeneti függvényeket valamint a Bode és a Nyquist diagramokat.

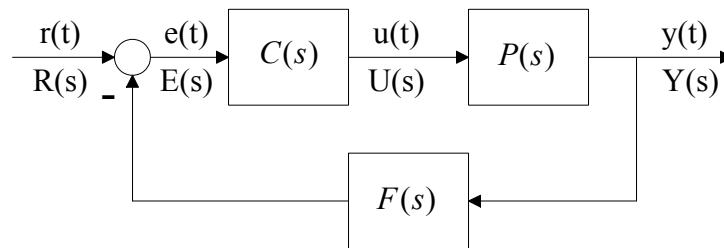
Összefoglalás:

Az átmeneti függvény értéke állandósult állapotban ($t \rightarrow \infty$) és a frekvenciafüggvény amplitúdója $\omega \rightarrow 0$ értéknél megegyezik.

Az arányos jellegű (tiszta arányos tagot és tárolós tagokat tartalmazó) elemek Nyquist diagramja $\omega = 0$ értéknél a komplex számsík valós tengelyének egy pozitív pontjából indul, ez az érték a szakasz átviteli tényezője. Integráló jellegű elemek Nyquist diagramja az imaginárius tengely negatív végtelen irányából indul. A kétszeresen integráló (esetleg tárolós tagokat is tartalmazó) elem Nyquist diagramja a negatív valós tengely végtelen irányából indul. Differenciáló jellegű elemek Nyquist diagramja a komplex számsík origójából indul a pozitív imaginárius tengely irányába. Tárolós tagok esetén (zérusok nélkül) a Nyquist diagram annyi síknegyeden halad át, amennyi a tárolók száma. A fázisszög a frekvenciával monoton módon változik. A zérusok eltorzítják a fázisszög monoton változását, meghatározott frekvenciatartományokban a görbén „behorpadásokat” okoznak. Arányos jellegű tagok Bode amplitúdó diagramja a frekvencia tengellyel párhuzamosan indul, zérus fáziseltolással. Egy integrátort tartalmazó rendszer Bode amplitúdó diagramja -20dB/dekád meredekséggel indul -90° fázisszöggel, míg a kettős integrátort tartalmazó rendszer Bode amplitúdó diagramja -40dB/dekád meredekséggel indul -180° fázisszöggel. A tárolós tagok a töréspontokban -20dB/dekád meredekséggel változtatják az aszimptotikus Bode amplitúdó görbe meredekségét. A zérusok megjelenése $+20\text{dB/dekád}$ meredekségváltozást okoz.

5. Visszacsatolás Vizsgálata

A szabályozástechnika alapvető struktúráis eleme a visszacsatolás. A zárt körben az egyes jelek közötti eredő átviteli függvények (pl. a szabályozott jellemző és az alapjel közötti eredő átviteli függvény) könnyen kiszámíthatók. Az eredő frekvenciafüggvények egyszerűen származtathatók az átviteli függvényekből $s=j\omega$ helyettesítéssel. A Matlab közvetlenül támogatja a tipikus blokkdiagram-algebrai struktúrákat, a sorosan kapcsolt illetve a visszacsatolt körök átviteli függvényeinek számítását. E függvényhívások rövid összefoglalására tekintünk az alábbi zárt szabályozási kört (C - szabályozó, P - folyamat, F - visszacsatolás):



A szabályozási körben az alábbi eredő átviteli függvények számíthatók:

A felnyitott kör átviteli függvénye (hurokátviteli függvény):

$$L(s) = C(s)P(s)F(s)$$

A zárt szabályozási kör eredő átviteli függvényei, a hibajel, a beavatkozójel, illetve a kimenőjel (szabályozott jellemző) alapjelre vonatkozó eredő átviteli függvényei:

$$W_e(s) = \frac{E(s)}{R(s)} = \frac{1}{1+L(s)}, \quad W_u(s) = \frac{U(s)}{R(s)} = \frac{C(s)}{1+L(s)}, \quad T(s) = \frac{Y(s)}{R(s)} = \frac{C(s)P(s)}{1+L(s)}$$

Az előrevezető ág átviteli függvénye: $E=CP$:

$$\gg \mathbf{E=series(C,P)}; \text{ vagy } \gg \mathbf{E=C*P};$$

A hurokátviteli függvény: $L=CPF=EF$

$$\gg \mathbf{L=series(E,F)}; \text{ vagy } \gg \mathbf{L=C*P*F};$$

A zárt körben a kimenőjelenek az alapjelre vonatkozó eredő átviteli függvénye:

$$\gg \mathbf{T=feedback(E,F,-1)};$$

Egységnyi visszacsatolás esetén ($F=1$) a zárt kör eredő átviteli függvénye a *loop* utasítással számítható:

$$\gg \mathbf{T=cloop(E,-1)};$$

Az eredő átviteli függvény közvetlenül az LTI struktúrával szimbólikusan is számítható:

$$\gg \mathbf{T=C*P/(1+C*P*F)}$$

A *minreal* utasítás meghívásával kijelölhetők a megegyező zérusok és pólusok.

$$\gg \mathbf{T=minreal(T)}$$

Az egyszerűsítéshez tolerancia is megadható. A tolerancia értéke alapértelmezésben

$$\mathit{sqrt}(\mathit{eps})=1.4901e-008.$$

$$\gg \mathbf{T=minreal(T,0.001)}$$

Ezesetben az egyszerűsítés akkor történik meg, ha a pólusok és zérusok értékének különbsége kisebb 0.001-nél. Az átviteli függvényekből ezután a *bode* vagy a *nyquist* függvényhívással számítható illetve jeleníthető meg a frekvenciafüggvény.

A frekvencia függvény:

Vizsgáljuk a felnyitott és a zárt rendszer frekvenciafüggvényeinek kapcsolatát, a zárt szabályozási kör átmeneti függvényét és annak kapcsolatát a frekvenciafüggvénnyel.

1. Példa.

A felnyitott kör átviteli függvénye:

$$L(s) = \frac{K}{s(1+sT_a)}$$

Egységnyi negatív (merek) visszacsatolást alkalmazunk. A zárt kör eredő átviteli függvénye:

$$T(s) = \frac{K}{s^2 T_a + s + K} = \frac{1}{1 + s \frac{1}{K} + s^2 \frac{T_a}{K}}$$

Az eredő másodrendű lengő tag, amely általánosan az alábbi alakban írható fel: $\frac{1}{1 + 2\zeta T_0 s + T_0^2 s^2}$

Az együtthatók összehasonlításával:

$$T_0 = \sqrt{\frac{T_a}{K}}, \quad \zeta = \frac{1}{2\sqrt{KT_a}}$$

Ábrázoljuk a felnyitott és a zárt rendszer Bode amplitúdó diagramjait egy ábrában $T_1=1$, és $K=0.1$ valamint $K=4$ esetén. Számítsuk ki a felnyitott kör frekvenciafüggvényének jellegzetes pontjait.

```

» s=tf('s')
» Ta=1
» K1=0.1
» K2=4
» L1=K1/(s*(s+1))
» L2=K2/(s*(s+1))

```

A kimenőjel és az alapjel közötti eredő átviteli függvény az alábbi összefüggéssel számítható:

$$L(s) = \frac{num}{den}, \quad T(s) = \frac{L(s)}{1+L(s)} = \frac{num}{num+den}$$

```

» T1=L1/(1+L1)
» T1=minreal(T1) % kiejti az azonos pólus-zérus párokat.

```

vagy

```

» T1=feedback(L1,1)
» T2=feedback(L2,1)

```

Adjuk meg az átmeneti függvényeket, határozzuk meg a felnyitott és a zárt kör frekvenciafüggvényeit.

```

» figure(1)
» step(T1,'r',T2,'y')
» figure(2)
» bode(L1,'b',L2,'c',T1,'r',T2,'y')

```

Egységnyi merev visszacsatolás esetén a zárt rendszer Bode amplitúdó-körfrekvencia diagramja a kisfrekvenciás tartományban közel egységnyi (nagy kisfrekvenciás erősítés esetén), a nagyfrekvenciás tartományban közelítően egybeesik a felnyitott rendszer amplitúdó menetével.

A felnyitott rendszer diagramjából számított vágási körfrekvencia környékén nagy K értékeknél a zárt rendszer amplitúdó diagramjában nagy kiemelés lehet. Megfigyelhetjük, hogy minél nagyobb a zárt rendszer frekvenciafüggvényének kiemelése, annál nagyobb az átmeneti függvény túllendülése.

Határozzuk meg a zárt rendszer csillapítási tényezőjét.

» **damp (T1)**

Eigenvalue	Damping	Freq. (rad/s)
-1.13e-001	1.00e+000	1.13e-001
-8.87e-001	1.00e+000	8.87e-001

» **damp (T2)**

Eigenvalue	Damping	Freq. (rad/s)
-5.00e-001 + 1.94e+000i	2.50e-001	2.00e+000
-5.00e-001 - 1.94e+000i	2.50e-001	2.00e+000

Tárolhatjuk a csillapítási tényezőt és a frekvencia értékét változóknak is.

» **[w01, zeta1]=damp (T1)**

» **[w02, zeta2]=damp (T2)**

A különböző erősítési tényezők lényegesen eltérő viselkedést eredményeznek. Az első esetben a pólusok valósak, míg a második esetben konjugált komplexek, amelyek az időtartományban lengő viselkedést eredményeznek.

2. Példa.

Egy rendszer hurokátviteli függvénye:

$$L(s) = \frac{k}{(1+s)(1+5s)} = \frac{10}{(1+s)(1+5s)}$$

Vizsgáljuk a felnyitott és a zárt rendszer viselkedését.

» **s=zpk ('s')**

» **L=10 / ((1+s) * (5*s+1))**

» **T=feedback (L, 1)**

Határozzuk meg a felnyitott és a zárt kör átmeneti függvényeinek állandósult értékeit.

A Laplace transzformáció végérték tétele szerint egységugrás alapjel esetén

$$r(t) = 1(t), R(s) = \frac{1}{s}$$

$$y(t \rightarrow \infty) = \lim_{s \rightarrow 0} sR(s)H(s) = \lim_{s \rightarrow 0} s \frac{1}{s} H(s) = \lim_{s \rightarrow 0} H(s)$$

A felnyitott kör és a zárt kör átmeneti függvényeinek állandósult értéke:

$$y_{nyitott}(t \rightarrow \infty) = \lim_{s \rightarrow 0} L(s) = k = 10$$

$$y_{zárt}(t \rightarrow \infty) = \lim_{s \rightarrow 0} T(s) = \frac{y_{nyitott}(t \rightarrow \infty)}{1 + y_{nyitott}(t \rightarrow \infty)} = \frac{k}{1+k} = \frac{10}{1+10}$$

A hiba értéke állandósult állapotban: $e(t \rightarrow \infty) = 1 - y_{nyitott}(t \rightarrow \infty) = \frac{1}{1+k} = \frac{1}{11}$

Ábrázoljuk a felnyitott és a zárt kör átmeneti függvényeit:

» **step (L, 'r', T, 'b')**

Az állandósult értékek leolvashatók a görbékről vagy kiszámíthatók a *dcgain* utasítással.

» **yos=dcgain (L)**

» **yos=dcgain (T)**

Jelenítsük meg a felnyitott és a zárt kör Bode diagramjait egy ábrán.

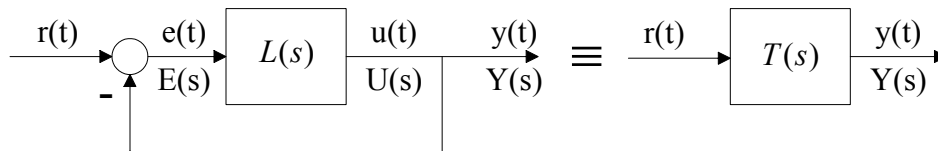
» **bode (L, 'r', T, 'b')**

Határozzuk meg az állandósult hiba értékét $k=1, 20, 100$ esetére.

6. Stabilitásvizsgálat

Egy rendszer stabilis, ha kitérés után a magára hagyott rendszer visszatér nyugalmi állapotába, tranziensei lecsengenek. A stabilitás egy másik megfogalmazása szerint egy rendszer stabilis, ha korlátos bemenet esetén a kimenet is korlátos.

Tekintsük az alábbi hatásvázlattal megadott szabályozási kört. A felnyitott rendszer átviteli függvénye $L(s)$. A rendszerben merev (egységnyi) negatív visszacsatolást alkalmazunk:



A zárt rendszer eredő átviteli függvényét a következőképpen számolhatjuk:

$$T(s) = \frac{L(s)}{1 + L(s)}$$

1. Stabilitásvizsgálat a zárt rendszer pólusai alapján:

A rendszer stabilitása meghatározható a zárt rendszer pólusai alapján. A rendszer stabilis, ha pólusai a komplex bal félsíkra esnek, azaz összes pólusának a valós része negatív.

1. Példa

Egy zárt rendszer eredő átviteli függvénye:

$$T(s) = \frac{s + 5}{s^5 - 3s^4 + 4s^3 + 10s^2 + 5s - 10}$$

Állapítsuk meg, hogy a rendszer stabilis-e.

Megoldás: Vizsgáljuk meg a pólusok elhelyezkedését.

```

» num=[1, 5]
» den=[1, -3, 4, 10, 5, -10]
» T=tf(num,den)
» poles=roots(den)
poles =
    2.1150 + 2.1652i
    2.1150 - 2.1652i
   -0.9824 + 0.7214i
   -0.9824 - 0.7214i
    0.7348

```

vagy másképpen

```

» [z,p,k]=zpkdata(T,'v')

```

A rendszer labilis, mivel vannak pozitív valós részű pólusai.

A `pzmap` utasítás grafikusan jeleníti meg a pólusok elhelyezkedését:

```

» pzmap(T)

```

Látható, hogy a rendszer labilis, mivel vannak pólusai a komplex számsík jobb oldali részén.

2. A Nyquist stabilitási kritérium alkalmazása:

A zárt rendszer stabilitása meghatározható a felnyitott rendszer frekvenciafüggvénye alapján is.

a. Az egyszerűsített Nyquist kritérium akkor használható, ha a felnyitott rendszernek nincs labilis (pozitív valós részű) pólusa. A zárt rendszer stabilis, ha a felnyitott rendszer Nyquist diagramja nem veszi körül a $(-1+0j)$ pontot.

b. Az általánosított Nyquist kritériumot kell használni akkor, ha a felnyitott rendszernek van labilis pólusa. A zárt rendszer stabilis, ha a felnyitott rendszer Nyquist diagramja annyiszor veszi körül a $(-1+0j)$ pontot az óramutató járásával ellentétes (pozitív) irányban, ahány labilis pólusa van a felnyitott rendszernek.

2. Példa

A felnyitott kör átviteli függvénye

$$L(s) = \frac{10}{(1+10s)(1+s)}.$$

Negatív merev visszacsatolást alkalmazunk. A Nyquist kritérium alapján határozzuk meg, hogy a zárt rendszer stabilis-e.

» **s=zk('s')**

» **L=10/((1+10*s)*(1+s))**

» **[z,p,k]=zpkdata(L,'v')**

Van-e labilis (pozitív valós részű) pólusa a nyílt rendszernek? *(Nincs, tehát az egyszerűsített Nyquist kritérium használható.)*

» **nyquist(L),grid**

Körülveszi-e a Nyquist diagram a $(-1+0j)$ pontot? *(Nem, tehát a rendszer stabilis.)*

Stabilis marad-e a rendszer, ha a 10-es erősítést a számlálóban megnöveljük?

Ellenőrizzük a stabilitást a zárt rendszer pólusainak vizsgálatával. A *feedback* utasításban az *1* a merev (egységnyi), a *-1* pedig a negatív visszacsatolást jelenti. (A *-1* elhagyható, mert az alapértelmezés a negatív visszacsatolás)

» **T=feedback(L,1,-1)**

vagy

» **T=L/(1+L); T=minreal(T)**

A *minreal* utasítás elvégzi a lehetséges egyszerűsítéseket, kiejti a közös zérusokat és pólusokat.

» **[z,p,k]=zpkdata(T,'v')**

Ábrázoljuk a zárt rendszer pólusait:

» **pzmap(T)**

Láthatjuk, hogy a rendszer struktúráisan stabilis. A Nyquist diagram nem veszi körül a $(-1+0j)$ pontot még megnövelt erősítés esetén sem.

3. Példa

A felnyitott rendszer átviteli függvénye

$$L(s) = \frac{-5}{(1-10s)(1+0.1s)}.$$

Negatív merev visszacsatolást alkalmazunk. A Nyquist kritérium alapján határozzuk meg, hogy a zárt rendszer stabilis-e?

» **L=-5/((1-10*s)*(1+0.1*s))**

» **[z,p,k]=zpkdata(L,'v')**

Van-e labilis pólusa a felnyitott rendszernek? *(Igen, $p_2 = 0.1$)*

» **nyquist(L)**

Milyen irányban kerüli meg a görbe a $(-1+0j)$ pontot? *(Az óramutató járásával ellentétes irányban)*

Stabilis-e a zárt rendszer? *(Igen)*

Ellenőrizzük az eredményt a zárt rendszer pólusainak kiszámításával.

- » **T=feedback(L,1)**
- » **step(T)**
- » **[z,p,k]=zpkdata(T,'v')**
- » **pzmap(T)**

Vizsgáljuk meg a rendszer stabilitását, ha a pólusok előjelét megváltoztatjuk.

$$L(s) = \frac{-5}{(1+10s)(1-0.1s)}$$

Milyen irányban kerüli meg a Nyquist diagram a -1 pontot? Stabilis-e a zárt rendszer?

4. Példa

Egy felnyitott rendszer átviteli függvénye

$$L(s) = k \frac{1-s}{(1+s)(1+0.5s)}$$

Negatív merev visszacsatolást alkalmazunk. Határozzuk meg a k paraméter azon értékeit, amelyekre a zárt rendszer stabilis lesz.

a. $k=1$ értéket feltételezünk:

- » **L=(1-s)/((1+s)*(1+0.5*s))**
- » **[z,p,k]=zpkdata(L,'v')**

Van-e labilis pólusa a felnyitott rendszernek? (Nincs, tehát az egyszerűsített Nyquist kritérium használható)

- » **nyquist(L),grid**

Határozzuk meg azt a pontot, ahol a görbe metszi a valós tengelyt (-0.666). A *zoom* utasítás vagy a *zoom* menüpont és az egér segítségével a Nyquist diagramról leolvashatjuk ezt az értéket.

Ha egy rendszer erősítését k -val megszorozzuk, akkor a Nyquist diagramját k -szorosára nagyítjuk. A rendszer akkor van a stabilitás határhelyzetében, ha a metszéspont a -1 értékre esik. Tehát a rendszer erősítését még megnövelhetjük $k=1/0.666=1.5$ értékkel. A rendszer stabilis, ha teljesül, hogy $0 < k < 1.5$ (pozitív k -ra)

b. $k=-1$ értéket feltételezünk:

- » **nyquist(-L), grid**

Határozzuk meg ismét azt a pontot, ahol a görbe metszi a valós tengelyt (-1). Tehát a rendszer stabilis, ha $k > -1 > 0$.

Tegyük össze a két feltételt. A zárt rendszer stabilis, ha $-1 < k < 1.5$

A stabilitás meghatározható az *rlocus* utasítással is, amely megadja a rendszer gyökhely-görbét, vagyis a zárt rendszer pólusainak változását a k paraméter függvényében.

- » **help rlocus**
- » **rlocus(L)**
- » **rlocfind(L)**

Az *rlocfind* utasítással és az egér segítségével a gyökhelygörbe egy pontjához meghatározhatjuk az erősítést.

Fázistartalék (fázistöbblet), erősítési tartalék:

Egy rendszer stabilitásvizsgálatánál nemcsak az a fontos, hogy a rendszer stabilis-e, hanem az is, hogy milyen messze van a stabilitás határhelyzetétől. A fázistartalék és az erősítési tartalék fejezik ki ezeket a jellemzőket. A fázistartalék azt adja meg, hogy a vágási körfrekvenciánál a rendszer fázisszögét még mennyivel lehet megnövelni, hogy a rendszer a stabilitás határhelyzetébe (-180°) kerüljön. A fázistartalékot meghatározhatjuk az ω_c vágási körfrekvencián felvett fázisszög értékéből.

$$\varphi_m = \varphi(\omega_c) + 180^\circ$$

ω_c az a frekvencia, ahol a felnyitott rendszer frekvenciafüggvényének abszolút értéke 1.

$$\omega_c : |L(j\omega)|_{\omega=\omega_c} = 1$$

Ha a fázistartalék pozitív, a rendszer stabilis. Például, ha a rendszer fázisszöge a vágási körfrekvencián $\varphi = -120^\circ$, akkor a fázistartalék $\varphi_m = \varphi + 180^\circ = -120^\circ + 180^\circ = 60^\circ$, azaz a rendszer stabilis.

Az erősítési tartalékkal a rendszer erősítését megszorozva a rendszer a stabilitás határhelyzetébe kerül.

$$gm = \frac{1}{|L(\omega_\pi)|},$$

ahol

$$\omega_\pi : \varphi(\omega)_{\omega=\omega_\pi} = -180$$

ω_π az a frekvencia, ahol a fázisszög -180° . Ha az erősítési tartalék nagyobb mint egy, akkor a rendszer stabilis. A fázistartalék szemléltethető a Nyquist és a Bode diagramon is. A *margin* utasítás segítségével a frekvenciafüggvény jellemzői egyszerűen kiértékelhetők és ábrázolhatók.

5. Példa

Egy felnyitott rendszer átviteli függvénye:

$$L(s) = \frac{1}{(0.5 + s)(s^2 + 2s + 1)}$$

Negatív merev visszacsatolást alkalmazunk. Határozzuk meg a fázistartaléket, az erősítési tartaléket és a felnyitott rendszer vágási körfrekvenciáját.

- » `L=1/((0.5+s)*(s^2+2*s+1))`
- » `[gm,pm,wg,wc]=margin(L)`

A *gm* (gain margin) az erősítési tartalék, *pm* (phase margin) a fázistartalék, *wc* (cut-off frequency) a vágási körfrekvencia és *wg* az a körfrekvencia, ahol a fázisszög értéke -180° .

Grafikusan is meg lehet jeleníteni ezen értékek elhelyezkedését:

- » `margin(L)` ;

A grafikus megjelenítés esetén a *Gm* érték decibelben adott, $Gm=20*\log_{10}(gm)$

Az erősítés, fázis és frekvencia értékek táblázatban is megjeleníthetők.

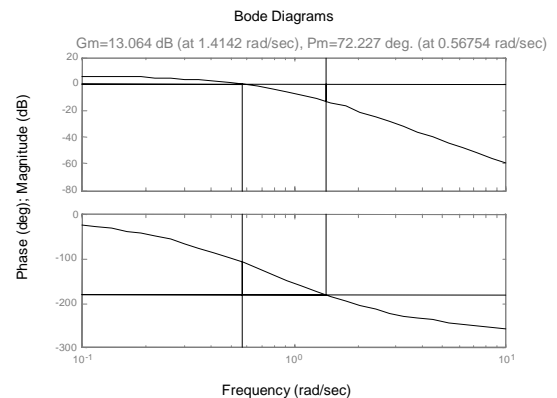
- » `w=logspace(-1,1,100)` ;
- » `[num,den]=tfdata(L,'v')`
- » `[mag,phase]=bode(num,den,w)` ;
- » `Tabl=[mag, phase, w']`

<i>mag</i>	<i>phase</i>	<i>w</i>
1.1123	-99.5242	0.5094
1.0643	-103.0406	0.5337
1.0158	-106.6104	0.5591 $\approx wc$
0.9669	-110.2286	0.5857
.....		
0.2449	-176.7848	1.3530
0.2211	-180.1658	1.4175 $\approx wg$
0.1991	-183.4774	1.4850
0.1789	-186.7160	1.5557

A táblázat az LTI sys struktúrából is számolható

- » `[mag,phase,w]=bode(L)` ;
- » `Tabl=[mag(:),phase(:),w]`

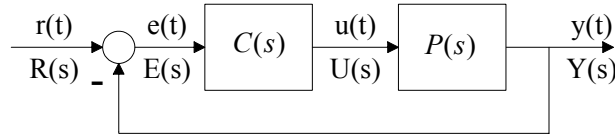
Az LTI struktúra több bemenetű, több kimenetű rendszerekre értelmezett, ezért a *mag* és *phase* változók 3 dimenziós tömbként adódnak. Egy bemenet, egy kimenet esetén a `(:)` operátor hozza a 3 dimenziós tömböket vektor alakra.



A fázistartalék kiolvasható a táblázatból, az 1.0 erősítési értékhez tartozó fázisszög értékéből lehet kiszámolni. $pm=180-106.6=73.4$. A vágási körfrekvencia $\omega=\omega_c=0.5591$. Az erősítési tartalék a -180° -hoz tartozó erősítés reciproka, $gm=1/0.221=4.52$. Az így számolt értékek pontosságát a táblázat felbontása korlátozza.

7. Soros PID Kompenzáció

Tekintsük az alábbi zárt szabályozási kört. $P(s)$ a folyamat (szabályozott szakasz), $C(s)$ a szabályozó átviteli függvénye.



Az adott szabályozandó folyamathoz olyan sorbaiktatott szabályozót kívánunk megadni, amellyel a zárt szabályozási rendszer stabilis működésű és megfelel a minőségi előírásoknak.

Először megfogalmazzuk a leglényegesebb minőségi előírásokat, majd néhány szabályozó tervezési eljárást mutatunk be.

Minőségi előírások:

Zárt szabályozási körökkel szemben általában az alábbi követelmények, gyakorlati igények merülnek fel:

Stabilitás: Alapvető követelmény a szabályozási rendszer stabilis működése. A stabilitást különböző módon fogalmazhatjuk meg. A BIBO (Bounded Input - Bounded Output) stabilitás azt jelenti, hogy a rendszer korlátos bemenőjelre korlátos kimenőjelet ad. Aszimptotikusan stabilis a rendszer, ha tranziensei lecsengenek.

Robusztusság: a zárt szabályozási kör viselkedése ne legyen érzékeny a folyamatról rendelkezésünkre álló információ pontatlanságára. A stabilitást akkor is biztosítani kell, ha a rendszer paraméterei nem pontosan ismertek, a névleges értékük körül adott tartományban változnak.

Statikus viselkedés: Egy másik fontos követelmény a szabályozási rendszer statikus vagy állandósult állapotbeli pontossága.

Az állandósult állapotra vonatkozó statikus előírások:

- Alapjelkövetés: a szabályozott jellemző kövesse az alapjelet (referencia jelet). A követési hiba az előírt értéken belül legyen.
- Zavarelhárítás: a szabályozás állandósult állapotban küszöbölje ki a fellépő zavarások hatását.

A statikus pontosság függ a rendszer struktúrájától és a bemenőjelektől is.

Tranziens viselkedés: A rendszer tranziens válasza a szabályozási kör lényeges dinamikus tulajdonsága. A tranziens viselkedés tulajdonságai megfogalmazhatók az időtartományban a rendszer egységugrás alapjelre vagy zavarásra adott válaszának tranziens tulajdonságaival.

A tranziens beállásra vonatkozó előírások:

- A kimenőjel túllövése,
- beállítás ideje (szabályozási idő).

A beállási idő alatt a szabályozott jellemző 1-2%-on belül megközelíti állandósult értékét. Általában egy kis, 5-10%-on belüli túllövés tolerálható, de vannak olyan folyamatok, ahol aperiodikus tranzienseket írnak elő.

A minőségi előírások a frekvenciatartományban is megfogalmazhatók. Az átmeneti függvény túllendülése összefüggésben van a zárt kör amplitúdó - frekvencia függvényének maximális értékével illetve a felnyitott rendszer frekvenciafüggvényéből számítható fázistartalékkal. Ha a fázistartalék körülbelül 60° , akkor a zárt rendszer átmeneti függvényének túllövése általában 5-10%-on belül lesz.

A beállási idő a vágási körfrekvenciától függ és az alábbi összefüggéssel becsülhető: $\frac{3}{\omega_c} \leq t_s \leq \frac{10}{\omega_c}$.

Beavatkozó jel nagysága: A gyakorlatban a beavatkozójel rendszerint korlátos. Követelmény, hogy a beavatkozójel maximuma ne haladja meg az előírt korlátot.

A szabályozó paramétereit a tervezési követelmények alapján kell megválasztani.

Bonyolultabb szabályozási problémák esetén a kimenőjel és a beavatkozójel teljes lefolyására előírhatók korlátozások. Például a szabályozási hiba és a beavatkozójel négyzetintegrálját minimalizáljuk. Ezek a korlátozások nemlineárisak és egymásnak ellentmondóak is lehetnek. Általános esetben csak valamilyen optimalizálási eljárással lehet a szabályozó paramétereit meghatározni.

Itt egy egyszerű és a gyakorlatban is gyakran alkalmazott módszert mutatunk be, amikor a rendszer fázistartalékát írjuk elő. A tervezés a felnyitott kör frekvenciafüggvénye alapján történik, azaz a felnyitott kör viselkedéséből a zárt rendszer tulajdonságaira következtetünk.

A felnyitott kör átviteli függvénye $L(s) = C(s)P(s)$. A zárt szabályozási kör eredő átviteli függvényei:

$$W_e(s) = \frac{E(s)}{R(s)} = \frac{1}{1 + C(s)P(s)}, \quad W_u(s) = \frac{U(s)}{R(s)} = \frac{C(s)}{1 + C(s)P(s)}, \quad T(s) = \frac{Y(s)}{R(s)} = \frac{C(s)P(s)}{1 + C(s)P(s)}$$

Itt $r(t)$ az alapjel, $u(t)$ a beavatkozójel és $y(t)$ a kimenőjel (szabályozott jellemző).

Soros, a szakasszal sorba beiktatott P (Proporcionális), PI (Proporcionális+Integráló), PD (Proporcionális+Differenciáló) és PID (Proporcionális+Integráló+Differenciáló) jellegű szabályozókat alkalmaznak a leggyakrabban, amelyeknek átviteli függvényei:

$$C_P(s) = A = k_c$$

$$C_{PI}(s) = A\left(1 + \frac{1}{sT_i}\right) = k_c \frac{1 + T_i s}{s}$$

$$C_{PD}(s) = k_c \left(1 + \frac{s\tau}{1 + T_1 s}\right) = k_c \frac{1 + T_d s}{1 + T_1 s}$$

$$C_{PID}(s) = C_{PI}(s)C_{PD}(s) = A\left(1 + \frac{1}{sT_i} + \frac{s\tau}{1 + T_1 s}\right) \approx k_c \frac{1 + T_i s}{s} \frac{1 + T_d s}{1 + T_1 s}$$

A PID tag PI és PD tagok szorzatával megadott közelítése akkor alkalmazható, ha $T_i > \tau > T_1$. A szabályozó átviteli függvényét megadhatjuk összeg illetve szorzat alakban.

PI szabályozót akkor alkalmazunk, ha állandósult állapotban pontos beállást követelünk meg egységugrás alapjelre. Az integráló hatás fogja biztosítani a hibamentes beállást. A PD szabályozó beiktatása gyorsítja a rendszert. Ideális PD szabályozóban a T_1 paraméter értéke zérus. Ez a szabályozó azonban nem realizálható. PID szabályozót akkor alkalmazunk, ha a szabályozás pontosságát és gyorsaságát is növelni kívánjuk. A PD tag a beavatkozójel jelentős kezdeti megnövekedését eredményezi, amely a rendszer gyorsításáért felelős.

A szabályozót a folyamathoz (illetve annak modelljéhez) tervezzük a minőségi követelmények kielégítésére. Gyakori kompenzációs technika a póluskiejtés, amikor a szabályozó átviteli függvényének zérusaival kiejtjük a szakasz kedvezőtlen pólusait, és a szabályozó pólusaival kedvezőbb dinamikát

biztosítunk a szabályozási körben. A póluskiejtéses technika alkalmazásához célszerű a szabályozó átviteli függvényét szorzat alakban megadni.

Egy PID szabályozóban négy modell paraméter van: k_c, T_i, T_d, T_1 . Póluskiejtéses szabályozó tervezés esetén ezeket a következőképpen választjuk meg: a T_i paramétert a folyamat legnagyobb időállandójával megegyezőnek választjuk (legkisebb frekvenciájú pólus), a T_d paramétert pedig a második legnagyobb időállandóval tesszük egyenlővé. Így elérhetjük, hogy a behozott zérusok kiejtsék a folyamat pólusait. A T_i paramétert $T_1 = \frac{T_d}{n_p}$ alakban adjuk meg, ahol n_p a pólus áthelyezési arány, amely megadja, hogy a PD

tag hányszoros frekvenciára tolja el a kompenzált pólust. Jó gyakorlati szabály, hogy n_p -t az 2-10 tartományba vesszük fel. Ha nagyobbra választjuk, akkor a rendszer gyorsabb lesz, de ennek az ára az, hogy a beavatkozó jel maximuma is nagyobb lesz. Mivel a k_c paraméter értéke nem befolyásolja a felnyitott kör fázismenetét, ezért ezt arra használhatjuk, hogy beállítsuk vele az előírt fázistartalék értéket.

A soros P, PI, PD, PID szabályozó tervezés lépéseit az alábbi folyamat kompenzálására mutatjuk be.

$$P(s) = \frac{1}{(1+10s)(1+s)(1+0.2s)}$$

Tervezzünk soros P, PI, PD, PID típusú szabályozókat 60° fázistöbblet biztosítására. Adjuk meg a minőségi jellemzők értékeit. Számítsuk ki és ábrázoljuk a kimenőjelet és a beavatkozójelet egységugrás alapjel esetén.

P szabályozó tervezése:

A szabályozót $C(s) = k_c$ alakban adjuk meg. Tehát csak a k_c paraméter értékét kell meghatározni.

Adjuk meg a szakasz átviteli függvényét.

- » **s=zpk('s')**
- » **P=1/((1+10*s)*(1+s)*(1+0.2*s))**

Első lépésben vegyük fel a $kc=1$ értéket.

- » **kc=1**
- » **C=kc**
- » **L=C*P**

Ábrázoljuk a felnyitott rendszer Bode diagramját és határozzuk meg annak jellemző paramétereit (fázistartalék, erősítési tartalék, vágási körfrekvencia) a *margin* utasítás segítségével.

- » **margin(L)**

Látható, hogy a rendszernek jelentős fázis- illetve erősítési tartaléka van. A fázistolás monoton csökken nullától -270° -ig, tehát az erősítés megváltoztatásával be lehet állítani a kívánt fázistartalékot. A $\varphi = \varphi_m - 180^\circ$ -hoz tartozó erősítés reciproka lesz a szabályozó erősítési tényezője. Ezt le lehet olvasni a rendszer Bode diagramjáról, vagy ki lehet olvasni egy táblázatból vagy a *margin* utasítás segítségével lehet számolni.

A *margin* utasítás az erősítési tartalék (gm) értékét a -180° -hoz tartozó erősítési értékből számolja. Ha a rendszer fázis értékeit lecsökkentjük a kívánt fázistartalék értékkel, akkor a *margin* utasítás éppen az adott fázistartalékhoz tartozó erősítési érték reciprokát adja vissza gm paraméterként.

- » **[mag, phase, w]=bode(L);**
- » **gm=margin(mag, phase-60, w)**
- » **kc=gm**

Tehát a szabályozó: $C_p(s) = k_c = 7.51$

Ezzel egyenértékűen a kc parameter meghatározható a frekvenciafüggvény adatait tartalmazó táblázatból is az alábbiak szerint:

» **Table=[mag(:), phase(:), w]**

<i>mag</i>	<i>phase</i>	<i>w</i>
0.2756	-95.0730	0.3290
0.1960	-107.0164	0.4520
0.1340	-119.7735	0.6210
0.0873	-133.4679	0.8532
0.0537	-147.8469	1.1721

A -120° -hoz tartozó erősítés reciproka lesz a kc , a hozzátartozó frekvencia pedig a vágási körfrekvencia:

$$kc = 1/0.1340 = 7.4627, \omega_c = 0.621$$

A w frekvenciavektor felbontását finomítva mindkét módszer azonos eredményt ad.

Fontos, hogy ellenőrizzük a rendszer viselkedését.

» **C=kc**

» **L=kc*L**

Ellenőrizzük a stabilitási tartalékra jellemző paramétereket (erősítési tartalék, fázistöbblet).

» **margin(L)**

» **[gm, pm, wg, wc]=margin(L)**

A fázistöbblet valóban 60° . A vágási körfrekvencia $\omega_c = 0.6245$.

Számoljuk ki a zárt rendszer eredő átviteli függvényét.

» **T=feedback(L, 1)**

Adjuk meg a felnyitott és a zárt kör Bode diagramját.

» **bode(L, 'r', T, 'b')**

Ábrázoljuk a zárt rendszer átmeneti függvényét.

» **step(T)**

Számítsuk is ki az átmeneti függvényt.

» **t=0:0.05:10;**

» **y=step(T, t);**

Az átmeneti függvény maximális értéke:

» **ym=max(y)**

Az átmeneti függvény állandósult értéke:

» **ys=dcgain(T)**

Ezekből az értékekből a százalékos túllendülés értéke kiszámítható.

» **yt=(ym-ys)/ys**

Az állandósult hiba:

» **es=1-ys**

Még meg kell vizsgálni az $u(t)$ beavatkozó jel viselkedését is. Ez azért fontos, mert ez a valóságos folyamat bemenőjele és itt lépnek fel a fizikai korlátozások. Számoljuk ki a beavatkozójel és az alapjel közötti eredő átviteli függvényt.

» **U=feedback(C, P)**

vagy

» **U=C/(1+C*P)**

Egységugrás alapjel esetén:

» **ut=step(U, t);**

» **plot(t, ut)**

A leggyakrabban korlátozást a beavatkozó jel maximumára adnak meg, számoljuk ezt ki.

» **um=max(ut)**

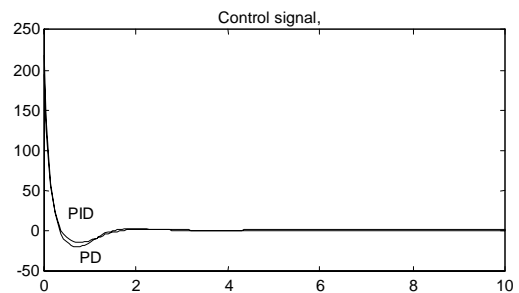
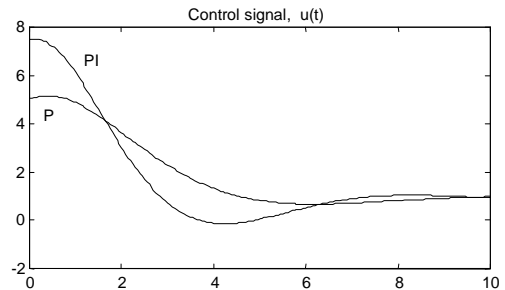
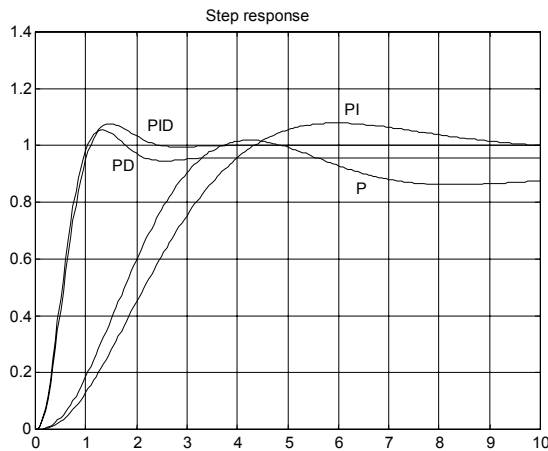
P,PI,PD,PID szabályozók tervezése:

A PI,PD,PID szabályozókat hasonló módon lehet megtervezni.

Hozzunk létre egy *m file*-t a szabályozó tervezésre a szabályozás jellemző paramétereinek kiszámítására.

Az alábbi táblázat összefoglalja az eredményeket.

	$C(s)$	$L(s) = C(S)P(s)$	k_c	ω_c	yt	es	um	~ ts
Sza- kasz		$\frac{1}{(1+10s)(1+s)(1+0.2s)}$		0	0	0.5	2	12
P	k_c	$\frac{k_c}{(1+10s)(1+s)(1+0.2s)}$	7.51	0.62	0.153	0.117	7.51	8
PI	$k_c \frac{(1+10s)}{s}$	$\frac{k_c}{s(1+s)(1+0.2s)}$	0.504	0.46	0.078	0	5.16	9
PD	$k_c \frac{(1+s)}{(1+0.1s)}$	$\frac{k_c}{(1+10s)(1+0.1s)(1+0.2s)}$	21.69	1.98	0.103	0.044	217	2
PID	$k_c \frac{(1+10s)(1+s)}{s(1+0.1s)}$	$\frac{k_c}{s(1+0.1s)(1+0.2s)}$	19.34	1.79	0.076	0	193	2



A szabályozott rendszerrel szemben támasztott követelmény a gyors beállítás és az alapjel követés volt. Látható, hogy a P kompenzáció egyik feltételt sem teljesíti. A beállítás elég lassú és nem is éri el a várt $y=1$ értéket. A PI szabályozás hatására a stacionárius hiba nullára csökkent, de a gyorsaság nem változott. A PD kompenzáció felgyorsítja a rendszert, de csak maradó hibával követi az egységugrást. Ezt a gyorsítást úgy érte el, hogy jelentősen megnövekedett az $u(t)$ beavatkozó jel értéke. A PID szabályozó hatására a rendszer gyorsabb is lett és a stacionárius hiba is lecsökkent nullára

Írjunk Matlab programot (hozzunk létre egy m fájlt, ez egy szövegfájl m kiterjesztéssel). Nyissunk meg egy új m -fájlt a Matlab *file* menüjéből. Írjuk a Matlab utasításokat az üres fájlba. Mentsük el a fájlt: *Save As, C:/Matlab/work/myfile.m*

A program meghívásához egyszerűen írjuk be a program nevét kiterjesztés nélkül a Matlab parancs ablakába.

» **myfile**

Az alábbi Matlab program szolgálja a szabályozó tervezést.

```

clear
s=tf('s');
P=1/((1+10*s)*(1+s)*(1+0.2*s))
P=zpk(P)

Cp=1
Cpi=(1+10*s)/(10*s)
Cpd=(1+s)/(1+0.1*s)
Cpid=Cpi*Cpd
Cpid=zpk(Cpid)

[mag,phase,w]=bode(Cp*P);
kp=margin(mag,phase-60,w)
Cp=kp*Cp;

[mag,phase,w]=bode(Cpi*P);
kpi=margin(mag,phase-60,w);
Cpi=kpi*Cpi;

[mag,phase,w]=bode(Cpd*P);
kpd=margin(mag,phase-60,w);
Cpd=kpd*Cpd;

[mag,phase,w]=bode(Cpid*P);
kpid=margin(mag,phase-60,w);
Cpid=kpid*Cpid;

Tp=feedback(Cp*P,1);
Tpi=feedback(Cpi*P,1);
Tpd=feedback(Cpd*P,1);
Tpid=feedback(Cpid*P,1);

Up=feedback(Cp,P);
Upi=feedback(Cpi,P);

figure(1),step(Tp,'r',Tpi,'b',Tpd,'g',Tpid,'y')
figure(2),step(Up,'r',Upi,'b')
figure(3),step(Upd,'g',Upid,'y')

t=0:0.05:10;
yp=step(Tp,t);
ypi=step(Tpi,t);
ypd=step(Tpd,t);
ypid=step(Tpid,t);

yvsp=dcgain(Tp)
yvspi=dcgain(Tpi)
yvspd=dcgain(Tpd)
yvspid=dcgain(Tpid)

ep=1-yvsp
epi=1-yvspi
epd=1-yvspd
epid=1-yvspid

ytp=(max(yp)-yvsp)/yvsp
ytpi=(max(yypi)-yvspi)/yvspi
ytpd=(max(ypd)-yvspd)/yvspd
ytpid=(max(ypid)-yvspid)/yvspid

up=step(Up,t);
upi=step(Upi,t);
upd=step(Upd,t);
upid=step(Upid,t);
upim=max(upi)
updm=max(upd)
upidm=max(upid)

```

A szabályozási kör viselkedése vizsgálható Simulink blokk-diagram megépítésével és futtatásával az adott szakasszal és a megtervezett szabályozókkal.

Másodrendű lengő tag kompenzálása

A szakasz az alábbi átviteli függvénnyel adható meg:

$$P(s) = \frac{A}{(s-p_1)(s-p_2)} = \frac{A}{s^2 T_0^2 + 2\zeta T_0 s + 1}, \text{ ahol } p_{1,2} = a \pm jb$$

A pólusok konjugált komplexek. A Bode diagram törésponti frekvenciája $\omega_0 = \frac{1}{T_0}$. Itt a közelítő Bode

diagram meredeksége 0-ról -40dB/dekád ra változik. Ezesetben egy lehetséges PID póluskiejtéses kompenzációs technika, ha mind a PI, mind pedig a PD tag időállandóját a természetes körfrekvencia reciprokára választjuk, vagyis $T_i = T_0$ és $T_d = T_0$, $C(s) = k_c \frac{1+T_0 s}{s} \frac{1+T_0 s}{1+T_1 s}$.

Egy másik lehetőség egy tiszta integráló tag alkalmazása kompenzáló tagként, amelynek erősítését úgy választjuk meg, hogy a fázistöbblet 60° körüli legyen.

Érzékenység vizsgálata:

A szakasz paraméterei általában nem pontos értékek, hiszen mérés, azonosítással, azonosítással határozzuk meg őket. Az átmeneti függvényből például közelítően meghatározhatjuk a szakasz jellegét (pl. egytárolós holtidős tag, kéttárolós tag, stb.) és megbecsülhetjük paramétereit. A paraméterek értéke egy minimális és egy maximális érték között változhat. A szabályozási rendszer viselkedésének vizsgálatakor fontos szempont a paraméter bizonytalanságok hatásának figyelembe vétele.

A szabályozó tervezésekor figyelembe kell vennünk a paraméterek értékeinek bizonytalanságát. A szabályozásnak a paraméterek névleges értékeinél az előírásoknak megfelelően kell viselkednie, de elfogadható viselkedést kell kapnunk, ha a paraméterek a névleges értéküktől eltérnek, de még a megadott sávban vannak. Különböző szabályozó tervezési eljárások vannak a robusztus viselkedés biztosítására.

A paraméter bizonytalanságok hatásának szemléltetésére tervezzünk egy PID szabályozót az alábbi lengő taghoz, és vizsgáljuk a szabályozási rendszer viselkedését, ha a szakasz csillapítási tényezője változik.

$$P(s) = \frac{1}{4s^2 + 4\zeta s + 1}, \text{ a } \zeta \text{ csillapítási tényező } 0.2 \text{ és } 1 \text{ között változik.}$$

Tervezzük a szabályozót 60° fázistöbbletre.

A Matlab utasítások:

```
» clear
» s=tf('s');
```

Adjuk meg a szakaszt:

```
» P1=1/(4*s*s+0.8*s+1)           %zeta=0.2
» P2=1/(4*s*s+2.8*s+1)         %zeta=0.7, névleges
» P3=1/(4*s*s+4*s+1)           %zeta=1
» P=zpk(P); P1=zpk(P1); P2=zpk(P2); P3=zpk(P3);
```

A szabályozó $k_p=1$ erősítéssel:

```
» Cpid=(1+2*s)*(1+2*s)/((2*s)*(1+0.2*s))
» Cpid=zpk(Cpid)
```

A Bode diagram a névleges P2 szakaszra:

```
» [mag,phase,w]=bode(Cpid*P2);
```

Határozzuk meg a szabályozó átviteli tényezőjét a 60° fázistöbblet biztosításához:

```
» kp=margin(mag,phase-60,w)
» Cpid=kp*Cpid;
```

```

» figure(1); bode(Cpid*P2)    % a Bode diagram ábrázolása
» Tpid=feedback(Cpid*P2,1);  % az eredő átviteli függvények
számítása

```

```

» Upid=feedback(Cpid,P2);

```

Számítsuk ki és ábrázoljuk a kimenőjelet és a beavatkozójelet.

```

» t=0:0.05:10;
» ypid=step(Tpid,t) ;
» figure(2); plot(t,ypid),grid,shg
» upid=step(Upid,t);
» figure(3); plot(t,upid),grid,shg

```

```

» Tpid1=feedback(Cpid*P1,1)

```

```

» Tpid3=feedback(Cpid*P3,1)

```

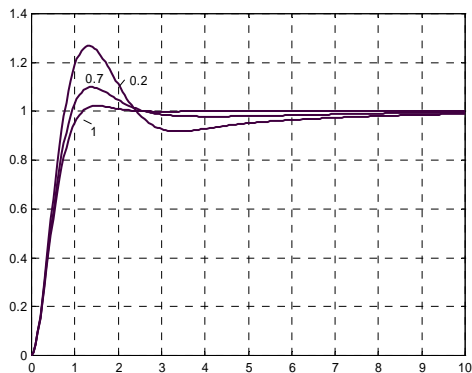
Számítsuk ki és ábrázoljuk a kimenőjelet a névleges szakasszal és a megváltozott csillapítási tényezőkkel.

```

» ypid1=step(Tpid1,t);
» ypid3=step(Tpid3,t);
» figure(4); plot(t,[ypid,ypid1,ypid3]),grid

```

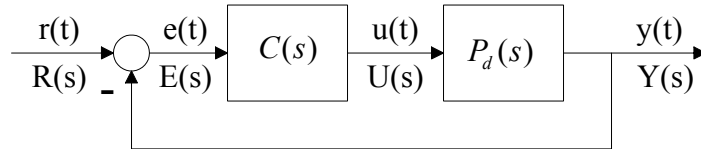
Az alábbi ábra mutatja a szabályozás kimenőjelét a névleges szakaszhoz ($\zeta = 0.7$) tervezett szabályozóval a névleges szakasszal és a módosított csillapítási tényezőjű szakasszal. Látható, hogy a túllendülés lényegesen nagyobb a kisebb csillapítási tényezővel. A szabályozás érzékeny a csillapítási tényező megváltozására.



8. Holtidős Rendszer Soros Kompenzációja

Holtidős tagot tartalmazó rendszer kompenzálása bonyolultabb, mint egy holtidő mentes rendszeré, mivel a holtidős tag nem reprezentálható pontosan egy racionális törtfüggvénnyel. A holtidős tag által létrehozott fázistolást a tervezés során külön kell figyelembe venni.

Tekintsük az alábbi szabályozási kört:



ahol $P(s)$ a folyamat átviteli függvénye holtidő nélkül, $C(s)$ a szabályozó átviteli függvénye és $L(s) = C(s)P_d(s)$ a felnyitott kör átviteli függvénye.

Vegyük a következő példát:

$$P_d(s) = P(s)e^{-sT_d} = \frac{e^{-s}}{1+20s}.$$

A $C(s)$ szabályozót úgy kell megválasztani, hogy az előírt minőségi jellemzők teljesüljenek.

Előírások: Egységugrás alapjel esetén a szabályozott rendszer statikus hiba nélkül kövesse az alapjelet és a kimenőjel túllövése körülbelül 10% legyen.

Ezek alapján egy PI kompenzáló tagot választunk:

$$C(s) = k_c \frac{1+20s}{s}$$

A zárt rendszer eredő átviteli függvénye:

$$L(s) = C(s)P(s) = k_c \frac{1+20s}{s} \frac{e^{-s}}{1+20s} = k_c \frac{e^{-s}}{s}$$

A k_c konstans úgy választjuk meg, hogy a fázistartalék 60° legyen.

A folyamat frekvenciafüggvényének amplitúdóját meghatározhatjuk a késleltetés nélküli rendszerből:

$$|P_d(j\omega)| = |P(j\omega)e^{-j\omega T_d}| = |P(j\omega)|, \text{ mivel } |e^{-j\omega T_d}| = 1;$$

A fázisszög pedig

$$\text{Arg}\{P_d(j\omega)\} = \text{Arg}\{P(j\omega)\} + \text{Arg}\{e^{-j\omega T_d}\} = \text{Arg}\{P(j\omega)\} - \omega T_d$$

- » **s=zpk('s')**
- » **P=1/(1+20*s)**
- » **Td=1**
- » **kc=1**
- » **C=kc*(1+20*s)/s**

A felnyitott kör átviteli függvénye:

- » **L=C*P**
- » **L=minreal(L)**

A felnyitott kör frekvenciafüggvényének amplitúdója megegyezik a késleltetés nélküli rendszer amplitúdójával, a fázisszöge pedig egy lineáris taggal módosul:

- » **[mag, phase, w]=bode(L);**
- » **magd=mag(:);**
- » **phased=phase(:)-w*Td*180/pi;**

A k_c erősítést kétféleképpen számolhatjuk:

1. módszer: A *margin* utasítás segítségével:

Az erősítési tartalék -120° -os fázisra:

» **gm=margin(magd,phased-60,w)**

Ez lesz a k_c tényező értéke

» **kc=gm**

0.5229

2. módszer: Táblázat segítségével:

» **T=[phased, magd, w']**

-116.5943	2.1544	0.4642	
-118.5162	2.0092	0.4977	
-120.5770	1.8738	0.5337	<=
-122.7868	1.7475	0.5722	
-125.1562	1.6298	0.6136	

A *mag* értéke *phased* = -120° -nál **1.8738**.

Ebből $k_c=1/1.8738$

» **kc=1/1.8738**

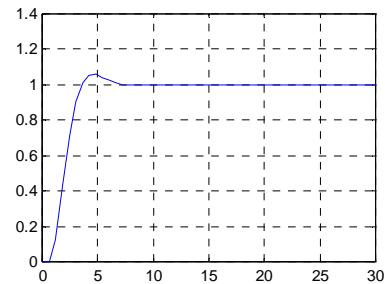
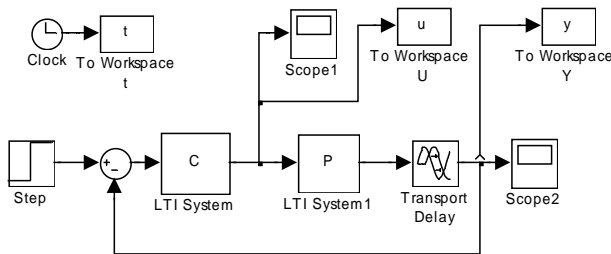
0.5337

Számoljuk újra a szabályozót:

» **C=kc*(1+20*s)/s**

Ellenőrizzük, hogy a fázistartalék valóban 60° lett-e.

A rendszer viselkedését a következő Simulink modellel lehet vizsgálni. A Simulink lehetővé teszi, hogy a holtidőt egyszerűen szimuláljuk.



A szimuláció eredményét visszaküldhetjük a Matlab felületre további vizsgálatok és grafikus megjelenítés céljából. Ezt kétféleképpen tehetjük meg. Egyrészt a *To Workspace* blokkal, másrészt a *Scope* blokk segítségével. A *To Workspace* blokkban be kell állítani a Matlab-ban használandó változó nevét, és a típusát *Matrix*-ként kell megadni. Ezután ábrázolhatjuk a jeleket a Matlab-ból is:

» **plot(t,y),grid**

A *Scope* grafikus ablak paramétereit állítsuk be a következőképpen

A 'properties' menü alatt.

Data history: Save data to workspace
 Variable name: ty (tu for the control signal)
 Matrix format

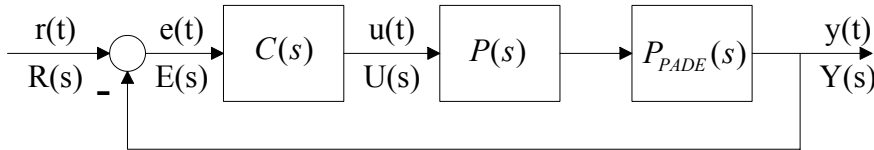
Így a t idő és az y kimenőjel vektorai egyszerűen kinyerhetők a szimuláció után. Ezekből pedig a minőségi jellemzők meghatározhatók (túllövés, beállási idő, maximális beavatkozó jel, stb.).

```

» t=ty(:,1)
» y=ty(:,2)
» plot(t,y),grid

```

Pade közelítés: Holtidős rendszer kezelése Pade közelítés segítségével is elvégezhető. A késleltető tag közelíthető egy racionális törtfüggvénnyel, $P_{PADE}(s) \approx e^{-sT_d}$. (A Pade törtfüggvény Taylor sorának első néhány tagja megegyezik a holtidős tag átviteli függvénye Taylor sorának első néhány tagjával.)



Matlab-ban a `pade` utasítás számítja ki a közelítést a kívánt fokszámra. Például használjunk 5-ödrendű közelítést.

```

» [numpade,denpade]=pade(Td,5)
» Ppade=tf(numpade,denpade)
» Pd=P*Ppade

```

A kc erősítés ezzel már meghatározható.

```

» [mag,phase,w]=bode(Pd);
» kc=margin(magd,phased-60,w)
0.5229

```

Ki kell hangsúlyozni, hogy az első részben megadott módszer pontosabb, mint a Pade közelítés alkalmazása és a kapott felnyitott és zárt átviteli függvények jóval bonyolultabbak a magasabb fokú közelítés miatt. A Pade közelítés előnye, hogy a tervezés menete nagyon hasonló a késleltetés nélküli rendszerek tervezési módszeréhez.

9. Labilis Rendszer Soros Kompenzációja

1. *Példa.* Vizsgáljuk a következő labilis folyamatot:

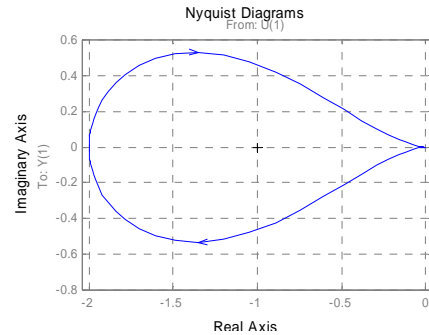
$$P(s) = \frac{20}{(s+2)(s-5)}$$

Stabilizálható-e a folyamat egy arányos (P) $C(s) = k_c$ szabályozóval?

```

» s=zpk('s')
» P=20/((s+2)*(s-5))
» figure(1); grid on; nyquist(P);
» figure(2); grid on; bode(P);
» figure(3); rlocus(P);

```



Látható, hogy a stabilitást nem lehet elérni, mivel a Nyquist diagram nem veheti körbe a $(-1+j0)$ pontot az óramutató járásával ellentétes irányban, és a fázistartalék is mindig kisebb mint nulla (sajnos a Matlab itt rosszul ábrázolja a Bode diagramban a fázisszöget, azt 360° -kal módosítja). Ugyanez az eredmény kapható meg a karakterisztikus egyenlet és a pólusok alapján. A pólusokat az $s^2-3s-10+kc=0$ karakterisztikus egyenletből határozhatjuk meg. A stabilitás szükséges feltétele, hogy az együtthatóknak azonos előjelűeknek kell lenniük. Ez nem teljesülhet, mert a kc értéke nem befolyásolja a -3 együtthatót. A gyökhelygörbe ($0 \leq kc < \infty$) alapján is ugyanezt kapjuk. Látható, hogy minden körerősítés értéknél a komplex számsík jobb oldalára mindig esik legalább egy pólus.

2. *Példa.* Vizsgáljuk meg most a következő folyamatot:

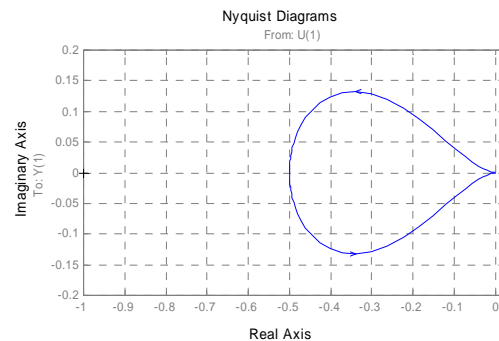
$$P(s) = \frac{5}{(s-2)(s+5)}$$

Stabilizálható -e a folyamat egy arányos (P) $C(s) = k_c$ szabályozóval?

```

» clear
» s=zpk('s')
» P=5/((s-2)*(s+5))
» figure(1); grid on; nyquist(P);
» figure(2); grid on; bode(P);
» figure(3); rlocus(P);

```



Látható, hogy a Nyquist diagram az óramutató járásával ellenkező irányban körülveheti a -1 pontot. Az erősítés növelésével tehát teljesíthető az általános Nyquist kritérium, $k_c > 2$ esetben a zárt rendszer stabilis lesz. A Bode diagramról is leolvasható, hogy a fázistartalék pozitív lehet. Válasszuk meg a k_c értékét úgy, hogy a vágási körfrekvencia a fázisszög maximumára essen.

Először vizsgáljuk a felnyitott kört $kc=1$ esetén.

```

» C=1
» L=C*P
» [mag,phase,w]=bode(L);

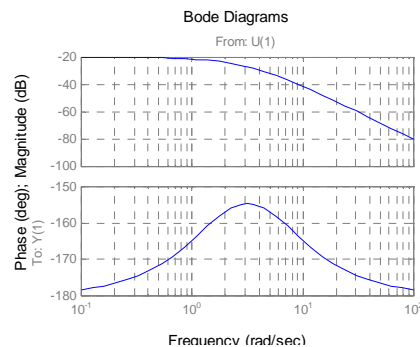
```

1. *Módszer:* Táblázatból olvassuk ki az értékeket.

```

» T=[ phase(:), mag(:), w]
-158.1444 0.3559 1.7433
-155.9825 0.3066 2.2122
-154.7797 0.2530 2.8072
-154.6231 0.2259 3.1623
-154.7797 0.1994 3.5622
-155.9825 0.1501 4.5204

```



A Bode fázisgörbe a $\varphi = -154.62$ mag=0.2259 $\omega=3.16$ -nél éri el a maximumát. Az arányos szabályozóval elérhető maximális fázistartalék tehát $pm=180-154.6=25.8^\circ$.

Ebben az esetben az erősítést $kc=1/0.045=4.42$ értékre kell választani.

» **kc=4.42**

2. *Módszer:* A maximális értéket meghatározhatjuk a *max* utasítással is.

» **[maxphase, index]=max(phase)**

» **kc=1/mag(index)**

Számoljuk újra a szabályozót:

» **C=kc;**

» **L=C*P;**

A *margin* utasítással grafikusán is ellenőrizhetjük a fázistartalékot.

» **margin(L);**

A rendszernek elég kicsi a fázistartaléka, $Pm = 25.4^\circ$ (60° -ot szeretnénk elérni).

Nézzük meg a zárt rendszer viselkedését:

» **T=feedback(L,1)**

» **step(T)**

A stabilitást sikerült elérni, de a viselkedés nem megfelelő, a rendszernek közel 100%-os statikus hibája van. PID szabályozó alkalmazásával a szabályozás minőségi viselkedése javítható.

3. *Példa.* Tervezzünk PID szabályozót a statikus hiba lecsökkentése érdekében.

$$C(s) = k_c \frac{(s+2)}{s} \frac{(s+5)}{s+50}$$

A labilis $p_1 = 2$ pólust nem ejthetjük ki közvetlenül egy labilis zérussal, mivel a paramétereket rendszerint mérési eredmények alapján határozzuk meg, és a pólus és az őt kiejtő zérus kis eltérése esetén is már a rendszer labilissá válik. A labilis pólust ehelyett egy stabilis PI taggal kompenzáljuk. A rendszer gyorsítása érdekében a stabilis $p_1 = -5$ polust nagyobb frekvenciára toljuk el egy PD taggal ($p=-50$, a póluseltolási arány 10).

A k_c konstans ismét úgy választjuk meg, hogy a maximális fázistartalékot kapjuk.

Zárjuk be a grafikus ablakokat.

» **clear**

» **s=zpk('s')**

» **P=5/(s-2)*(s+5)**

» **C=((s+2)*(s+5))/(s*(s+50))**

» **L=C*P**

» **L=minreal(L)**

» **bode(L)**

» **[mag, phase, w]=bode(L);**

Határozzuk meg az erősítést a maximális fázishoz:

» **[maxphase, index]=max(phase)**

» **kc=1/mag(index)**

Az erősítési tényező $kc=152$, a fázistöbblet pedig

$pm=180+maxphase=58$.

Ellenőrizzük a rendszer viselkedését.

» **C=kc*((s+2)*(s+5))/(s*(s+50))**

» **L=C*P, L=minreal(L)**

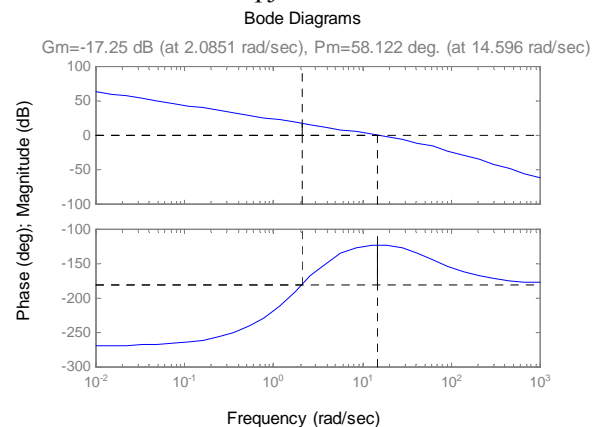
» **margin(L)**

A zárt rendszer átmeneti függvényét az alábbi ábra mutatja.

» **T=L/(1+L), T=minreal(T)**

A statikus hiba lecsökkent nullára.

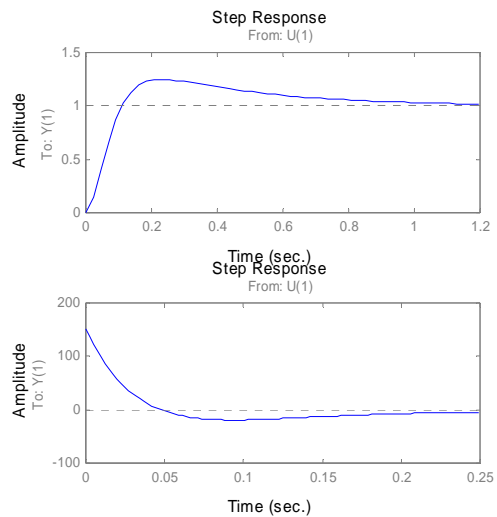
» **es=1-dcgain(T)**



Ellenőrizzük az u beavatkozó jelet:

- » $U=C/(1+L)$, $U=\text{minreal}(U)$
- » `subplot(211), step(T)`
- » `subplot(212), step(U)`
- » `u=step(U)`
- » `um=max(u)`

A beavatkozó jel um maximuma elég nagy. Ez az érték a póluseltolási arány csökkentésével mérsékelhető.



10. Mintavételes Rendszerek

z-transzformáció:

Digitális szabályozási rendszerekben a jeleket digitális alakban tároljuk és dolgozzuk fel. Ezekben a rendszerekben működési elvük következtében a jelfeldolgozás csak diszkrét mintavételi időpontokban történik. Ebből adódóan a szabályozási körökben megjelenő analóg jeleket mintavételezni és digitalizálni kell.

Egy folytonos $y(t)$ jel digitális alakját eltolt impulzusok összegével lehet leírni.

$$y_d(t) = \{y(nT_s)\} = y(0)\delta(t) + y(T_s)\delta(t - T_s) + y(2T_s)\delta(t - 2T_s) + y(3T_s)\delta(t - 3T_s) + \dots$$

Az $y_d(t)$ jel Laplace transzformáltja:

$$y_d(s) = y(0) + y(T_s)e^{-sT_s} + y(2T_s)e^{-2sT_s} + y(3T_s)e^{-3sT_s} + \dots$$

Vezessük be a következő jelölést:

$$z = e^{sT_s}, \quad z^{-1} = e^{-sT_s}$$

$$y_d(z) = y(0) + y(T_s)z^{-1} + y(2T_s)z^{-2} + y(3T_s)z^{-3} + \dots = \sum_{k=0}^{\infty} y(kT_s)z^{-k}$$

Ez az átalakítás a mintavételezett jel z -transzformációja, ahol a z^{-1} -et az eltolási operátorként lehet értelmezni.

1. Példa.

Egy jel z -transzformáltja

$$y_d = \frac{2z^2 - z}{z^2 - z + 0.24}, \quad T_s = 0.5$$

Számoljuk ki a jel inverz z -transzformációjának első $N_s = 5$ mintavételi pontban felvett értékét.

a/ Matlab utasítások:

```
» num=[2, -1, 0]
» den=[1, -1, 0.24]
» yd=dimpulse(num,den,5)
» plot(yd,'*')
» plot(1:5,yd,'*')
```

Szimbólikus adatbevitellel és LTI (sys) struktúrával is elvégezhető a számítás. Hasonlóan az s változóhoz egy z változó is definiálható, de itt a T_s mintavételi időt is meg kell adni.

```
» Ts=0.5
» z=tf('z',Ts)
» Y=(2*z^2-z)/(z*z-z+0.24)
```

Az LTI (sys) struktúra használatának egyik előnye, hogy ugyanazt az utasítást lehet alkalmazni folytonos és diszkrét rendszerekre. Ha a sys.Ts mintavételi idő (sys.Ts) paraméter nulla, akkor a Matlab a rendszert folytonosnak, ha pedig nullától eltérő, akkor diszkrétnek feltételezi. A *help* utasítás megadja az *impulse* utasítás pontos használatát.

```
» help impulse
```

```
IMPULSE(SYS,TFINAL) simulates the impulse response from t=0 to the
final time t=TFINAL.
```

A *tfinal* = 2 idő szükséges $T_s = 0.5$ mintavételi idő esetén ahhoz, hogy 5 mintavételi pontot kapjunk

(Az első érték $t=0$ -hoz tartozik)

```
» impulse(Y,2);
» yd=impulse(Y,2);
```

```
» plot(Ts*(0:4),yd,'*');
```

b/ Részlettrötekre való bontás

Egy exponenciális tag z-transzformáltja:

$$y(t) = e^{-at}, y_d(nT_s) = e^{-anT_s} \xrightarrow{z} \frac{z}{z - e^{-aT_s}}$$

A jelet részlettrötekre bonthatjuk a Matlab *residue* utasításával.

$$y_d(z) = \frac{num}{den} = z \frac{num1}{den} = z \left(k_0 + \frac{r_1}{z - p_1} + \frac{r_2}{z - p_2} \right)$$

```
» num1=[2, -1]
» den=[1, -1, 0.24]
» [r,p,k0]=residue(num1,den)
```

```
r =
     1
     1
p =
 0.6000
 0.4000
```

```
k0 =
 []
» Ts=0.5
```

$$y_d(z) = z \left(\frac{r_1}{z - p_1} + \frac{r_2}{z - p_2} \right) = z \left(\frac{1}{z - 0.6} + \frac{1}{z - 0.4} \right) = \frac{z}{z - 0.6} + \frac{z}{z - 0.4}$$

$$y_d(t) = \{y(nT_s)\} = \{e^{anT_s} + e^{bnT_s}\}, \text{ ahol } e^{aT_s} = 0.6, e^{bT_s} = 0.4$$

```
» a=log(0.6)/Ts
```

```
» b=log(0.4)/Ts
```

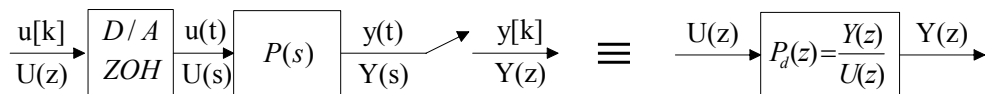
$$a = \frac{\ln(0.6)}{T_s} = -1.02, b = \frac{\ln(0.4)}{T_s} = -1.83$$

$$y(t) = e^{-1.02t} + e^{-1.83t}, \quad t \geq 0$$

```
» t=[0:Ts:2]';
» yd=exp(a*t)+exp(b*t);
» plot(t,yd,'*');
```

Impulzusátviteli függvény (Diszkrét átviteli függvény):

Egy folytonos folyamatot a D/A átalakítóval együtt az impulzusátviteli függvénnyel írhatjuk le. Az impulzusátviteli függvény a kimenőjel és a bemenőjel z-transzformáltjainak hányadosát adja meg nulladrendű tartószervet feltételezve a bemeneten.



2. Példa.

Határozzuk meg az alábbi folytonos folyamat impulzusátviteli függvényét. A mintavételezési idő $T_s = 1$.

$$P(s) = \frac{1 + 5s}{(1 + 10s)(1 + 8s)(1 + 4s)(1 + 2s)}, \quad T_s = 1$$

» **s=zpk('s')**

» **P=(1+5*s) / ((1+10*s) * (1+8*s) * (1+4*s) * (1+2*s))**

A folytonos folyamat átviteli függvénye zérus-pólus alakban:

$$P(s) = 0.0078 \frac{s + 0.2}{(s + 0.5)(s + 0.25)(s + 0.125)(s + 0.1)}$$

» **[zerusf, polusf, kf]=zpkdata(P, 'v')**

A folytonos rendszer pólusai: $-0.5, -0.25, -0.125, -0.1$

A folytonos rendszer zérusa: -0.2

A Matlab-ban egy folytonos folyamat diszkrét impulzusátviteli függvénye a *c2d* utasítással számolható: **sysd = c2d(sysc, ts, method)**, ahol *ts* a mintavételi idő és *method* pedig a tartószerv típusát adja meg. A default értéke 'zoh', ami nulladrendű tartószervet jelent.

» **Ts=1;**

» **Pd=c2d(P, Ts, 'zoh')**

$$P_d(z) = 0.0011 \frac{(z + 3.088)(z - 0.8187)(z + 0.2198)}{(z - 0.9048)(z - 0.8825)(z - 0.7788)(z - 0.6065)}$$

A diszkrét zérusok és pólusok:

» **[zerusd, polusd, kd]= zpkdata(Pd, 'v')**

A diszkrét pólusok: $0.9048, 0.8825, 0.7788, 0.6065$

A diszkrét zérusok: $-3.088, 0.8187, -0.2198$

Vegyük észre, hogy a diszkrét pólusokat a $z = e^{sT_s}$ transzformációval kapjuk a folytonos pólusokból.

» **exp(-0.5)**

0.6065

Hasonlóan

» **exp(polusf)**

Ez a transzformáció a folytonos zérusok és a diszkrét zérusok közt nem érvényes, az összefüggés ennél bonyolultabb.

Végérték tételek

Egy diszkrét jel kezdeti értékét meghatározhatjuk a következő összefüggéssel:

$$\lim_{n \rightarrow 0} y(nT_s) = \lim_{z \rightarrow \infty} y(z),$$

végértékét pedig az alábbi összefüggéssel adhatjuk meg:

$$\lim_{n \rightarrow \infty} y(nT_s) = \lim_{z \rightarrow 1} (1 - z^{-1})y(z).$$

3. Példa.

Az előző példában megadott rendszerre egységugrás bemenetet alkalmazunk. Az egységugrás diszkrét megfelelője egy egységnyi amplitúdójú impulzus sorozat, amelynek a *z*-transzformáltja $\frac{z}{z-1}$.

Határozzuk meg a kimenőjel kezdeti- és végértékét.

$$y(z) = \frac{z}{z-1} P_d(z)$$

A kezdeti érték nulla, mivel a számláló fokszáma magasabb, mint a nevező fokszáma. A végértéket meghatározhatjuk a $P_d(z)$ impulzusátviteli függvényből a $z=1$ helyettesítéssel. A Matlab-ban a végértéket

a *dcgain* utasítással számolhatjuk folytonos és diszkrét esetre is (diszkrét esetre a *ddcgain(numd,dend)* utasítás szintén használható). Hasonlóan itt is a mintavételi idő paraméter határozza meg, hogy a rendszer folytonos vagy diszkrét.

- » **P.Ts**
- » **A=dcgain(P)**
- » **Ad=dcgain(Pd)**

Mindkét, **A** (folytonos) és **Ad** (diszkrét) értékre 1 értéket kaptunk, ahogy ezt vártuk.

Stabilitás

Egy diszkrét rendszer stabilis, ha a pólusai (karakterisztikus egyenletének, azaz a nevezőjének a gyökei) a komplex sík egységsugarú körének a belsejébe esnek.

4. Példa.

Stabilis-e az alábbi diszkrét rendszer?

$$P_d(z) = \frac{z^2 - 0.3z - 0.1}{z^3 + 3z^2 + 2.5z + 1}, T_s = 1$$

Definiáljuk a diszkrét rendszert:

- » **Ts=1**
- » **z=tf('z',Ts)**
- » **Pd=(z*z-0.3*z-1)/(z^3+3*z^2+2.5*z+1)**

Határozzuk meg a pólusokat:

- » **zerusd,polusd,kd]=zpkdata(Pd,'v')**

A pólusok abszolút értékei:

- » **abs(polusd)**
- ```
ans =
 2.0000
 0.7071
 0.7071
```

A rendszer labilis, mert egyik pólusának abszolút értéke nagyobb mint 1.

A stabilitás meghatározható a pólusok complex síkon történő ábrázolásával.

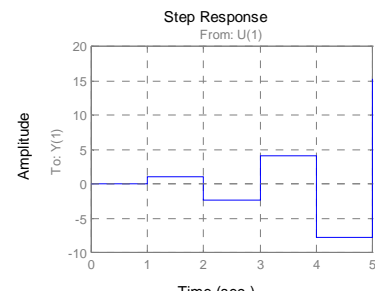
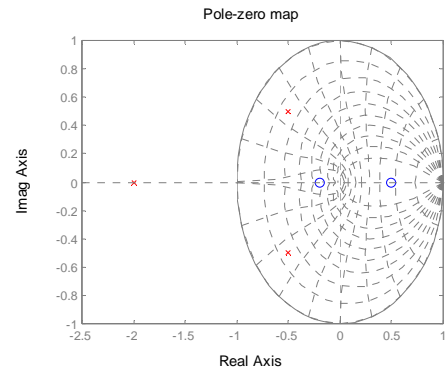
- » **pzmap(Pd)**
- » **zgrid**

A rendszer labilis, mert a -2 pólus az egységsugarú körön kívül helyezkedik el. A *zgrid* utasítás kirajzolja az egységsugarú kört és a konstans csillapítási tényezőhöz és természetes frekvenciához tartozó vonalakat.

A stabilitás az időtartományban is meghatározható az átmeneti függvényből.

- » **step(Pd)**

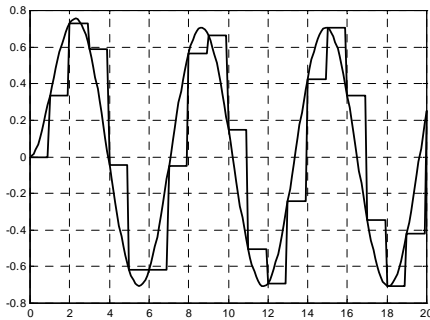
A kimenet nem korlátos, tehát a rendszer labilis.



## 11. Mintavételes PID Szabályozó Tervezése Kisfrekvenciás Közelítés Alapján

Mintavételes szabályozási körben a szabályozó tervezhető a frekvenciatartományban annak szem előtt tartásával, hogy a mintavételezés és zérusrendű tartószerv alkalmazása úgy tekinthető, mintha járulékos holtidő lépne fel a rendszerben.

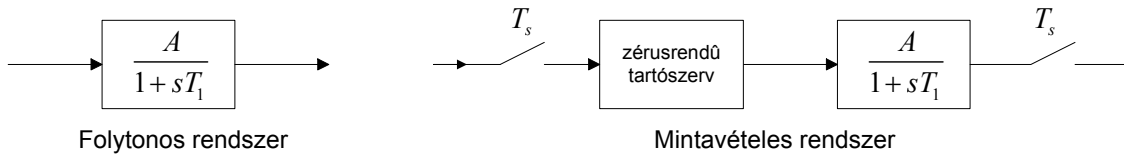
Egy egytárolós arányos tagot szinuszos jellel gerjesztünk. A tag kimenőjelét 1 sec mintavételi idővel mintavételezzük, majd zérusrendű tartószervet alkalmazunk. Ábrázoljuk a bemenőjelet, valamint a folytonos és a mintavételezett szakasz kimenőjelét egy diagramban az alábbi ábra szerint..



A folytonos kimenőjel kvázistacionárius állapotban a bemenőjelhez képest amplitudóban és fázisban eltér, de a bemenőjellel megegyező frekvenciájú szinuszos jel. Szemmel láthatóan a mintavételezett jel késik a folytonos kimenőjelhez képest, és annak alapharmonikusa kb. fél mintavételi idejű holtidős késleltetést mutat a folytonos kimenőjelhez képest.

Elemezzük a jelenséget a frekvenciatartományban. Hasonlítsuk össze egy folytonos egytárolós arányos tag frekvenciafüggvényét a mintavételezővel és zérusrendű

tartószervvel ellátott egytárolós arányos tag frekvenciafüggvényével.



A mintavételes rendszer impulzusátviteli függvénye:

$$P(z) = \frac{1 - e^{-T_s/T_1}}{z - e^{-T_s/T_1}}$$

A frekvenciafüggvényt  $z = e^{j\omega T_s}$  helyettesítéssel nyerjük. Az exponenciális tagokat közelítjük Taylor sorokkal.

$$P(z = e^{j\omega T_s}) = \frac{1 - e^{-T_s/T_1}}{e^{j\omega T_s} - e^{-T_s/T_1}} \approx \frac{1 - (1 - \frac{T_s}{T_1} + \frac{1}{2}(\frac{T_s}{T_1})^2 - \dots)}{1 + j\omega T_s - \frac{(\omega T_s)^2}{2} + \dots - (1 - \frac{T_s}{T_1} + \frac{1}{2}(\frac{T_s}{T_1})^2 - \dots)}$$

Ha  $T_s < T_1$  és  $\omega T_s < 1$ ,  $T_s/T_1$  valamint  $\omega T_s$  magasabb hatványai elhanyagolhatók, és a mintavételezett szakasz frekvenciafüggvénye közelítően megegyezik a folytonos szakasz frekvenciafüggvényével.

$$P(z = e^{j\omega T_s}) \approx \frac{T_s/T_1}{j\omega T_s + T_s/T_1} = \frac{1}{1 + j\omega T_1}$$

Az elhanyagolásokat egy további  $T_j$  járulékos holtidővel vehetjük figyelembe, amelynek értéke  $T_s/2$  és  $T_s$

között becsülhető.  $P(z = e^{j\omega T_s}) \approx \frac{1}{1 + j\omega T_1} e^{-j\omega T_j}$ .

### 1. Példa

Hasonlítsuk össze a folytonos rendszer és a mintavételes rendszer frekvenciafüggvényeit.

Legyen  $A=1$ ,  $T_1=0.1$  és a mintavételezési idő  $T_s=0.1$ .

$$P(s) = \frac{1}{1+0.1s}$$

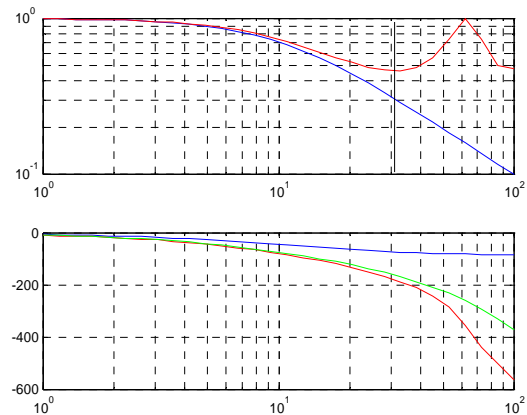
- » `s=zpk('s')`
- » `Ps=1/(1+0.1*s)`
- » `Ts=0.1;`
- » `Pz=c2d(Ps, Ts)`

$$P_d(z) = \frac{0.63212}{z-0.3679}$$

- » `[mags, phases, w]=bode(Ps);`
- » `[magz, phasez]=bode(Pz, w);`
- » `subplot(211), loglog(w, mags(:), 'b', w, magz(:), 'r'), grid;`
- » `PhasesWithDel=phases(:)-w*(Ts/2)*180/pi;`
- » `subplot(212)`
- » `semilogx(w, phases(:), 'b', w, phasez(:), 'r', w, PhasesWithDel, 'g')`
- » `grid`

Látható, hogy a z-transzformációnak megfelelő (kék) amplitúdó-frekvencia görbék kb.  $\omega=1/T_s=10$ -ig jól követik a folytonos rendszer (piros) amplitúdó-frekvencia görbéjét. A nagyobb frekvenciák tartományában a közelítés nem elfogadható, az eltérések igen nagyok. A fázisszögben az eltérés a folytonos (piros) és a mintavételes (kék) rendszer között már a kisfrekvenciás tartományban is észrevehető,  $\omega=1/T_s=10$ -nél már kb. 0.5 radián, a járulékos holtidős fázisszöggel kiegészített folytonos fázisszög görbe (zöld) azonban jó közelítést ad.

Látható, hogy a diszkrét frekvenciafüggvény amplitúdó menete (piros) az  $\omega = \pi/T_s = 31.4$  körfrekvencia után nagymértékben eltér a folytonostól a Shannon mintavételezési tételnek megfelelően.

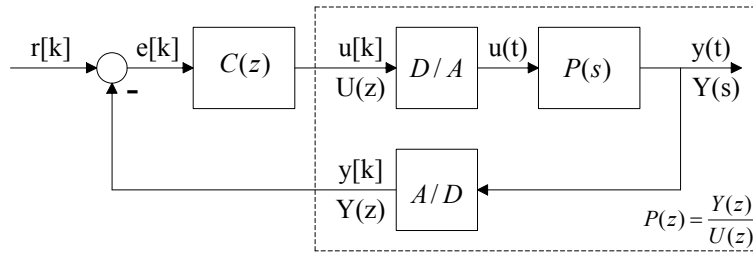


A mintavételezésből eredő járulékos holtidő (bármilyen kicsi is legyen az) megváltoztatja az eredeti rendszer strukturális tulajdonságait, pl. egy strukturálisan stabilis rendszer mintavételes szabályozás megvalósításakor nem lesz strukturálisan stabilis.

### Mintavételes PID Szabályozó Tervezése

A mintavételes zárt szabályozási rendszer alapvető strukturáját az alábbi blokkvázlat adja meg. Ez a rendszer hibrid rendszer abban az értelemben, hogy folytonos és diszkrét idejű jelek egyaránt szerepelnek benne.





A fenti ábrán  $C(z)$  a tervezendő diszkrét idejű szabályozó,  $P(s)$  az adott folytonos szakasz.

A szabályozással szemben minőségi követelményeket támasztunk, amelyek egyrészt a statikus alapjelkövetési illetve zavarelhárítási tulajdonságokat, másrészt a rendszer dinamikus tulajdonságait írják elő.

Tervezhetünk PID jellegű folytonos szabályozót a folytonos szakaszhoz, figyelembe véve, hogy a mintavételezés járulékos holtidőt jelent. Ezután a folytonos szabályozót diszkrét algoritmussá transzformáljuk.

Célszerű azonban közvetlenül a D/A nulladrendű tartószerv és a  $P(s)$  folytonos szakasz *együttes* mintavételes alakjából, a szakasz  $P(z)$  impulzusátviteli függvényéből kiindulva közvetlenül diszkrét PID jellegű szabályozó algoritmust tervezni. A tervezés póluskiejtéses technikán alapulhat. A diszkrét felnyitott rendszer Bode diagramját úgy módosítjuk, hogy kiejtjük a rendszer kedvezőtlen pólusait, és helyettük megfelelő pólusokat hozunk be.

#### Az eljárás lényege a következő:

A tárolós jellegű szakaszok impulzusátviteli függvényének nevezője  $(z - e^{-T_s/T_1})(z - e^{-T_s/T_2}) \dots$  alakú tényezőket tartalmaz.

A diszkrét P, PI, PD, PID jellegű algoritmusok az alábbi impulzusátviteli függvényekkel adhatók meg:

**P** szabályozó:  $C(z) = A$

**PI** szabályozó:  $C(z) = A \frac{z - e^{-T_s/T_1}}{z - 1}$

(Kiejthető a szakasz legnagyobb időállandója, helyette integráló hatást hozunk be.)

Differenciaegyenlet:  $u[k] = Ae[k] - A \exp(-T_s/T_1)e[k-1] + u[k-1]$ , ahol  $u[k]$

a beavatkozójel,  $e[k]$  pedig a hibajel aktuális értékét jelöli.

**PD** szabályozó:  $C(z) = A \frac{z - e^{-T_s/T_2}}{z - e^{-T_s/T_2^*}}$ , ahol  $T_2^* < T_2$ . (Kiejthető a szakasz egy kedvezőtlen

időállandója, amely helyett egy kisebb időállandót hozunk be.)

Ideális PD szabályozó:

$$C(z) = A \frac{z - e^{-T_s/T_2}}{z}$$

Differenciaegyenlete:  $u[k] = Ae[k] - A \exp(-T_s/T_2)e[k-1]$ .

(A folytonos PD algoritmustól eltérően diszkrétben az ideális PD hatás realizálható, mivel nem eredményez végtelen túlvezérlést.)

**PID** szabályozó: 
$$C(z) = A \frac{(z - e^{-T_s/T_1})(z - e^{-T_s/T_2})}{(z - 1)z}$$

Differenciaegyenlete:

$u[k] = A e[k] - A(\exp(-T_s/T_1) + \exp(-T_s/T_2))e[k-1] + A(\exp(-T_s/T_1)\exp(-T_s/T_2))e[k-2] + u[k-1]$  Megjegyzés: Többszörös PI, illetve PD hatás is alkalmazható, amennyiben az  $u$  beavatkozájra megadott korlátot nem lépjük túl.

A kompenzáló algoritmusoknak megfelelő differenciaegyenlet rekurzív összefüggés, amely valós időben realizálható.

## 2. Példa

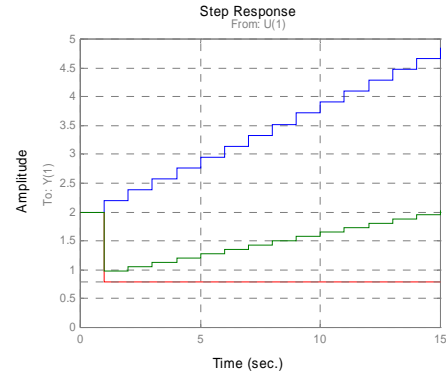
Vizsgáljuk az egyes szabályozók egységugrás választát.

```

>> T1=10; T2=2; Ts=1; A=2;
>> z=zpk('z',Ts)
>> pd1=exp(-Ts/T1), pd2=exp(-Ts/T2)
>> Cpi=A*(z-pd1)/(z-1)
>> Cpd=A*(z-pd2)/z
>> Cpid=A*(z-pd1)*(z-pd2)/(z*(z-1))
>> t=0:Ts:15;
>>

```

```
step(Cpi,'b',Cpd,'r',Cpid,'g',t),grid
```



Látható, hogy a diszkrét PID szabályozó algoritmusokkal a folytonos PID algoritmusokéhoz hasonló hatások érhetők el.

A szabályozók zérusaival kiejtjük a szakasz nemkívánatos pólusait. A szabályozó  $A$  átviteli tényezőjét úgy választjuk meg, hogy a diszkrét rendszer Bode diagramja alapján a fázistöbblet az előírt (rendszerint  $\sim 60^\circ$ ) legyen. A Bode diagram a  $dbode$  (ill. a rendszer LTI megadásakor a  $bode$  Matlab utasítással számítható). A frekvenciatartomány kb. az  $1/T_s$  értékig veendő fel. A vágási körfrekvencia nagyjából az  $\omega_c \approx 1/2(T_{holtidő} + T_s)$  értékre fog adódni. (A szakasz mintavételezéséből  $T_s/2$ , a szabályozó algoritmusból további  $T_s/2$  járulékos holtidő vehető figyelembe.) A frekvenciatartománynak tehát csupán a kisfrekvenciás része érdekes a szabályozás tervezése szempontjából, ahol a tárgyalat közelítés érvényes.

Mivel a fenti szabályozási algoritmusok a szakasz zérusait nem kompenzálják, a mintavételi pontok között nem várhatók lengések. A beavatkozájra rendszerint korlátok érvényesek. Ellenőrizni kell, hogy a beavatkozáj a korlátokon belül maradjon-e.

## 3. Példa

A folytonos rendszer adott az átviteli függvényével:

$$P(s) = \frac{e^{-s}}{(1+10s)(1+5s)} = P_1(s)e^{-s}$$

A rendszerben egy holtidős tag is szerepel  $Td=1$  késleltetéssel. A mintavételi idő  $T_s=1$ . Tervezzünk diszkrét soros szabályozót, amely teljesíti a következő minőségi előírásokat.

- Fázistartalék  $\geq 60^\circ$
- A beállási idő minél kisebb legyen;
- Egységugrás alapjel esetén a statikus hiba legyen nulla.

Először vegyük fel a folytonos rendszert késleltetés nélkül:

$$\gg \mathbf{s}=\mathbf{zpk}('s');$$

$$\gg \mathbf{P1s}=1/((1+10*s)*(1+5*s));$$

Határozzuk meg a diszkrét megfelelőjét nulladrendű tartószerv alkalmazásával.

$$P_1(z) = (1 - z^{-1})Z \left\{ \frac{P_1(s)}{s} \right\}$$

$$\gg \mathbf{Ts}=1;$$

$$\gg \mathbf{P1z}=\mathbf{c2d}(\mathbf{P1s}, \mathbf{Ts}, 'zoh');$$

(A default 'zoh' method megadása nem kötelező).

A késleltetést hozzáadjuk a rendszerhez, ha a  $P_1(z)$  impulzusátviteli függvényt megszorozzuk  $z^{-1}$ -gyel, mivel  $Z\{e^{-s}\} = z^{-1}$ .

A  $z$  változót definiáljuk hasonlóan mint a folytonos rendszerekre az  $s$  változót. Itt a mintavételi időt is meg kell adni.

$$\gg \mathbf{z}=\mathbf{zpk}('z', \mathbf{Ts})$$

$$\gg \mathbf{Pz}=\mathbf{P1z}/\mathbf{z}$$

$$P(z) = z^{-1} \cdot P_1(z) = \frac{1}{z} \cdot P_1(z) = 0.00905 \frac{(z + 0.9048)}{(z - 0.9048)(z - 0.8187)z}$$

Határozzuk meg a diszkrét folyamat zérusait és pólusait:

$$\gg [\mathbf{zd}, \mathbf{pd}, \mathbf{kd}] = \mathbf{zpkdata}(\mathbf{Pz}, 'v')$$

PI kompenzáló tag szükséges a nulla statikus hiba eléréséhez, a PD kompenzáló tagot pedig a rendszer gyorsítása érdekében alkalmazzuk.

A PI és PD tagok törési frekvenciáját a folytonos folyamat időállandói (pólusai) alapján választjuk meg: A  $T_i$  integrálási időállandót a folyamat legnagyobb időállandójával megegyező értékre választjuk, a  $T_d$  differenciálási időállandót pedig a második legnagyobb időállandóval tesszük egyenlővé. Ezen sarokfrekvenciák diszkrét megfelelőit a  $z = e^{sT_s}$  transzformáció alapján számíthatjuk:

$$\text{PI sarokfrekvencia: } 0.1 \Rightarrow e^{-\frac{T_s}{T_i}} = e^{-\frac{1}{10}} = e^{-0.1} = 0.9048$$

$$\text{PD sarokfrekvencia: } 0.2 \Rightarrow e^{-\frac{T_s}{T_d}} = e^{-\frac{1}{0.5}} = e^{-0.2} = 0.8187$$

A diszkrét szabályozó tehát:

$$C(z) = k_c \frac{z - 0.9048}{z - 1} \frac{z - 0.8187}{z}$$

A  $k_c$  paramétert úgy választjuk meg, hogy körülbelül  $60^\circ$  fázistartalékot érjünk el.

Először tételezzük fel, hogy  $k_c = 1$ :

$$\gg \mathbf{kc}=1;$$

$$\gg \mathbf{Cz} = ((\mathbf{z}-0.9048) * (\mathbf{z}-0.8187)) / ((\mathbf{z}-1) * \mathbf{z})$$

vagy közvetlenül a diszkrét átviteli függvény pólusaiból:

$$\gg \mathbf{Cz} = ((\mathbf{z}-\mathbf{pd}(1)) * (\mathbf{z}-\mathbf{pd}(2))) / ((\mathbf{z}-1) * \mathbf{z})$$

A felnyitott kör diszkrét átviteli függvénye:  $L(z) = C(z) \cdot P(z)$ .

$$\gg \mathbf{Lz}=\mathbf{Cz} * \mathbf{Pz};$$

$$\gg \mathbf{Lz}=\mathbf{minreal}(\mathbf{Lz}, 0.001);$$

A *minreal* utasítás akkor ejti ki az azonos zérus-pólus párt, ha az eltérésük kisebb a megadott pontosságnál. Ha a pontosság nincs megadva, a Matlab az előre definiált *eps* változót tekinti pontossági korlátnak.

» **eps**

A  $k_c$  paramétert a *margin* utasítással számolhatjuk vagy kiolvashatjuk egy táblázatból.

a. A *margin* utasítás használata. A *bode* utasítás a  $T_s$  mintavételi idő alapján dönti el, hogy a folyamat folytonos vagy diszkrét.

```
» [mag, phase, w]=bode(Lz);
» kc=margin(mag, phase-60, w);
```

b. Táblázat alkalmazásával

```
» [mag, phase, w]=bode(Lz);
» T=[mag(:), phase(:), w] % táblázat képzése
```

T =

```
0.0781 -114.8936 0.2200
0.0697 -117.8611 0.2462
0.0622 -121.1822 0.2756
0.0555 -124.8991 0.3084
0.0495 -129.0589 0.3452
```

Látható, hogy körülbelül  $-120^\circ$  fázisszöveget **0.062** erősítésnél érhetünk el, ezért a  $k_c$ -t  $1/0.0622=16$ -re kell választani.

```
» kc=1/0.0622
```

Megjegyezzük, hogy a *margin* utasítás kétféleképpen hívható meg. A bemenetei lehetnek a Bode amplitúdó, fázis és körfrekvencia értékei vagy az átviteli függvény LTI (sys) alakban.

Ellenőrizzük a viselkedést. Számoljuk ki a fázisstartalék értékét a *margin* utasítással.

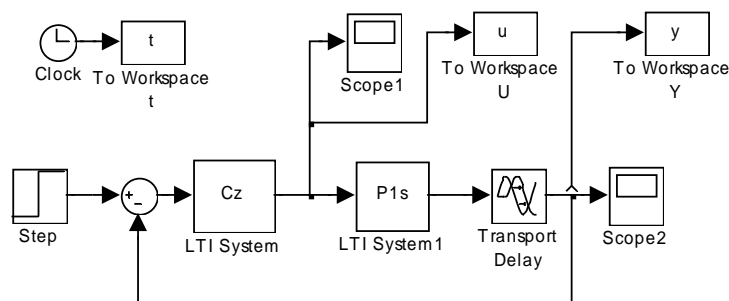
```
» Cz=kc*((z-pd(1))*(z-pd(2)))/((z-1)*z);
» Lz=Cz*Pz;
» Lz=minreal(Lz);
» margin(Lz);
```

Használjuk a Simulink-et a zárt rendszer viselkedésének vizsgálatára.

```
» simulink % elindítja a Simulink-et
```

A rendszer viselkedését a következő Simulink modellel lehet vizsgálni. A Simulink lehetővé teszi, hogy a folytonos folyamatot, a diszkrét szabályozót és a holtidőt egyszerre szimuláljuk.

Hozzunk létre egy új fájlt és másoljuk be az egyes blokkokat. A blokkok paramétereit változtassuk meg a kívánt értékre:



- $C(z)$ ,  $PI(s)$ : Control System Toolbox → LTI system :Cz, P1s
- Sum: Simulink → Math → Sum: +-
- Dead time, delay: Simulink → Continuous → Transport Delay: 1
- Step input: Simulink → Sources → Step, Változtassuk meg a Step time paramétert nullára
- Zero-Order-Hold: Simulink → Discrete → Zero-Order-Hold, Sampling time: Ts

- Scope: Simulink→Sinks→Scope

A szimuláció eredményét visszaküldhetjük a Matlab felületre további vizsgálatok és grafikus megjelenítés céljából. Ezt kétféleképpen tehetjük meg. Egyrészt a *To Workspace* blokkal, másrészt a *Scope* blokk segítségével. A *To Workspace* blokkban be kell állítani a Matlab-ban használandó változó nevét, és a típusát *Matrix*-ként kell megadni. Ezután ábrázolhatjuk a jeleket a Matlab-ból is.

A *Scope* blokk segítségével egyszerűen visszaküldhetjük a szimuláció eredményét a Matlab felületre. Változtassuk meg a paramétereket a 'properties' menü alatt a Scope grafikus ablakában.

Data history: Save data to workspace  
Variable name: My for ScopeY and Mu for ScopeU  
Matrix format

Változtassuk meg a Simulation→Parameters→Stop Time parametert 20-ra.

Így a  $t$  idő és az  $y$  kimenőjel vektorai egyszerűen kinyerhetőek a szimuláció után. Ezekből pedig a minőségi jellemzők meghatározhatók (túllövés, beállási idő, maximális beavatkozó jel, stb.).

```
» ty=My(:,1), y=My(:,2)
```

```
» tu=Mu(:,1), u=Mu(:,2)
```

Ábrázoljuk az  $y(t)$  kimenőjelet

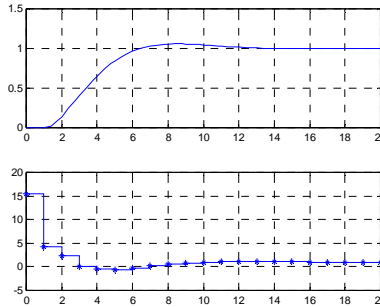
```
» subplot(211), plot(ty,y), grid
```

és az  $u(t)$  beavatkozó jelet, amely a nulladrendű tartószerv kimenete.

```
» subplot(212), stairs(tu,u), grid
```

A mintavételezett  $u[k]$  jelet is ki lehet rajzolni.

```
» hold on, plot(tu,u,'*')
```



## 12. Állapotteres Leírás, Irányíthatóság, Megfigyelhetőség

Egy lineáris rendszer jellemezhető bemenőjeleivel, kimenőjeleivel és állapotváltozóival. Az állapotváltozók olyan mennyiségek, amelyek a bemenőjel hirtelen megváltozására nem változnak meg hirtelen, pillanatnyi értéküket a rendszert a múltban ért hatások alakítják ki. Másképpen a rendszer azon változói, amelyek tároló funkciót látnak el.

Tekintsünk egy egy bemenetű, egy kimenetű (SISO - single input - single output) rendszert. A rendszer  $u$  bemenőjele,  $x$  állapotváltozói és  $y$  kimenőjele között az alábbi állapotegyenlet adja meg az összefüggést:

$$\begin{aligned}\dot{x} &= Ax + bu \\ y &= cx + du\end{aligned}$$

ahol az  $\{A, b, c, d\}$  mátrixok a rendszert jellemző paramétermátrixok.

Az állapotegyenlettel egy  $n$ -edrendű differenciálegyenlettel megadható rendszert  $n$  számú elsőrendű differenciálegyenlettel írunk le.

Az állapotegyenletnek végtelen sok reprezentációja létezik, mivel az állapotváltozók bármilyen lineáris kombinációja új állapotváltozókat eredményez. A különböző reprezentációk a bemenet és a kimenet között ugyanazt az átviteli kapcsolatot adják.

A MATLAB az átviteli függvényből kiindulva az állapotegyenlet egy lehetséges reprezentációját adja meg a *tf2ss* (transfer function to state space) utasítással.

Példaként tekintsük az alábbi átviteli függvénnyel adott másodrendű rendszert:

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

Adjuk meg a rendszer egy állapotteres reprezentációját.

```
» num=1;
» den=[1 1 1];
» [A,b,c,d]=tf2ss(num,den) % transzformálás állapotteres alakra
» [num1,den1]=ss2tf(A,b,c,d) % ellenőrzésképpen visszatranszformálás átviteli
% függvény alakra
```

A kétféle megadási forma egyenértékűségét nézzük meg az átmeneti függvények felrajzolásával.

```
» step(num,den);
» step(A,b,c,d);
```

A rendszer egy másik állapotreprezentációját koordináta-transzformációval (*hasonlósági transzformáció*) kaphatjuk meg. Az új  $\bar{x}$  állapotváltozók és az eredeti  $x$  állapotváltozók között a  $P$  transzformációs mátrix adja meg a kapcsolatot.

$$\begin{aligned}\dot{\bar{x}} &= \bar{A}\bar{x} + \bar{b}u \\ y &= \bar{c}\bar{x} + \bar{d}u\end{aligned}$$

ahol

$$\begin{aligned}\bar{x} &= Px \Rightarrow x = P^{-1}\bar{x} \\ \bar{A} &= PAP^{-1}, \bar{b} = Pb, \bar{c} = cP^{-1}, \bar{d} = d\end{aligned}$$

Ha a  $P^{-1}$  mátrix oszlopvektorai az  $A$  mátrix sajátvektorai, az  $\bar{A}$  mátrix diagonális lesz (kanonikus transzformáció).

Példa:

- »  $\mathbf{A} = [-1 \ -0.5 \ 0.5; \ 2 \ -3 \ 0; \ 2 \ -1 \ -2]$
- »  $\mathbf{b} = [2; 3; 1]$
- »  $\mathbf{c} = [0 \ 0 \ 1]$
- »  $\mathbf{d} = 0$
- »  $\mathbf{H} = \text{ss}(\mathbf{A}, \mathbf{b}, \mathbf{c}, \mathbf{d})$

Határozzuk meg a sajátvektorokat ( $\mathbf{V}$ ) és a sajátértékeket ( $\mathbf{ev}$ ).

- »  $[\mathbf{V}, \mathbf{ev}] = \text{eig}(\mathbf{A})$

Transzformáljuk a rendszert kanonikus alakra. Ekkor a sajátvektorokat megadó  $\mathbf{V}$  mátrix inverze a transzformációs mátrix.

- »  $\mathbf{P}_i = \mathbf{V}; \mathbf{P} = \text{inv}(\mathbf{V})$
- »  $\mathbf{A}_p = \mathbf{P} * \mathbf{A} * \mathbf{P}_i$
- »  $\mathbf{b}_p = \mathbf{P} * \mathbf{b}$
- »  $\mathbf{c}_p = \mathbf{c} * \mathbf{P}_i$
- »  $\mathbf{d}_p = \mathbf{d}$

Figyeljük meg, hogy  $\mathbf{A}_p$  valóban diagonális.

A fenti számítások egyszerűbben is elvégezhetők:

- »  $[\mathbf{A}_p, \mathbf{b}_p, \mathbf{c}_p, \mathbf{d}_p] = \text{ss2ss}(\mathbf{A}, \mathbf{b}, \mathbf{c}, \mathbf{d}, \text{inv}(\mathbf{V}))$

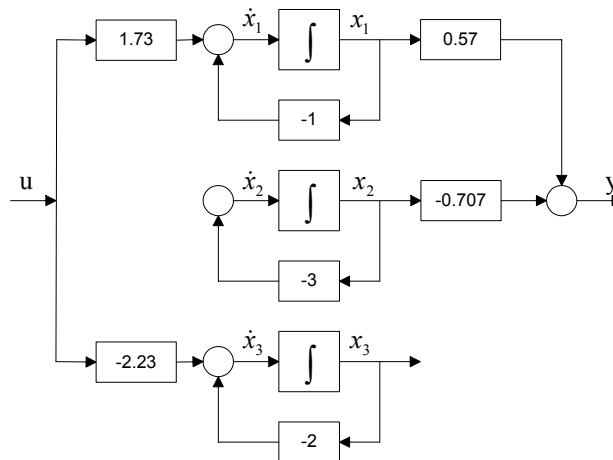
vagy

- »  $[\mathbf{A}_p, \mathbf{b}_p, \mathbf{c}_p, \mathbf{d}_p] = \text{canon}(\mathbf{A}, \mathbf{b}, \mathbf{c}, \mathbf{d}, 'modal')$

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \end{bmatrix} = \begin{bmatrix} -1 & 0 & 0 \\ 0 & -3 & 0 \\ 0 & 0 & -2 \end{bmatrix} \cdot \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} + \begin{bmatrix} 1.73 \\ 0 \\ -2.23 \end{bmatrix} \cdot u$$

$$y = [0.57 \quad -0.707 \quad 0] \cdot \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} + 0 \cdot u$$

Rajzoljuk fel a rendszer blokk-diagramját. Ez a rendszer *párhuzamos reprezentációja*. Figyeljük meg, hogy az átviteli tényező értéke  $b_1 \cdot c_1 = 1.73 \cdot 0.57 = 1$ . Bármilyen  $b_1$  és  $c_1$  felbontással, amelyre  $b_1 \cdot c_1 = 1$  a kimenőjel és a bemenőjel között azonos összefüggést kapunk, miközben az állapotegyenlet paraméterei különböznek.



A fenti párhuzamos struktúrából (az állapotegyenlet kanonikus alakjából) közvetlenül megadhatjuk a rendszer irányíthatósági és megfigyelhetőségi tulajdonságait. Látható, hogy az  $u$  bemenőjel nem befolyásolja az  $x_2$  állapotváltozót (az  $x_2$  állapotváltozó nem irányítható), az  $y$  kimenőjel pedig nem

tartalmaz információt az  $x_3$  állapotváltozóról (az  $x_3$  állapotváltozó nem megfigyelhető). Az  $y$  kimenőjel irányítható, mivel az  $u$  bemenőjel az  $x_1$  állapotváltozón keresztül befolyásolja. Ezeket a tulajdonságokat az eredeti alakból kiindulva is megállapíthatjuk a Kalman-féle irányíthatósági és megfigyelhetőségi mátrixok rangjának vizsgálatával.

A rendszer **irányíthatósági mátrixa**  $C_o = [b \quad Ab \quad \dots \quad A^{n-1}b]$  számítható a **ctrb** utasítással.

» **Co=ctrb(A,b)**

vagy az LTI struktúrával:

» **Co =ctrb(H)**

A rendszer teljesen állapotirányítható, ha  $C_o$  rangja megegyezik az állapotváltozók  $n$  számával.

» **rank(Co)**

Esetünkben a rendszer nem teljesen állapotirányítható, mivel  $rank(C_o) = 2 < n$ .

A rendszer **kimeneti irányítható**, ha a  $c_{oy} = c * C_o$  mátrix rangja megegyezik a kimenőjelek számával.

» **rank(c\*Co)**

Mivel ez az érték 1, megegyezik a kimenőjelek számával, a rendszer kimeneti irányítható.

A rendszer **megfigyelhetőségi mátrixa**  $O_b = \begin{bmatrix} c \\ cA \\ \dots \\ cA^{n-1} \end{bmatrix}$

A MATLAB **obsv** utasításával számolva:

» **Ob=obsv(A,c)**

» **Ob =obsv(H)**

» **rank(Ob)**

A rendszer megfigyelhető, ha a megfigyelhetőségi mátrix rangja  $n$ . Mivel  $rank(O_b) = 2 < n$ , a rendszer nem teljesen megfigyelhető.

Határozzuk meg a rendszer átviteli függvényét.

» **[num,den]=ss2tf(A,b,c,d)**

Az átviteli függvény zérus-pólus alakban:

» **Hzpk=zpk(H)**

Figyeljük meg, hogy két zérus kiejt két pólust.

» **Hzpk=minreal(Hzpk)**

Az átviteli függvény alakban információt veszünk a rendszer állapotairól. Az átviteli függvény csak az irányítható és megfigyelhető állapotváltozókról tartalmaz információt.

A párhuzamos reprezentációt megkaphatjuk a részlettörtes felbontással is:

» **[num,den]=tfdata(H,'v')**

» **[r,p,k]=residue(num,den)**

$$H(s) = \frac{0}{s+3} + \frac{0}{s+2} + \frac{1}{s+1}$$



*Megjegyzés:* Látható, hogy az  $\frac{1}{s+1}$  átviteli függvény az állapotegyenlettel megadott teljes rendszernek csak egy részrendszerét írja le. Ez a részrendszer a teljes rendszer irányítható és megfigyelhető része.

### 13. Állapotvisszacsatolás

Tekintsük először a folytonos rendszereket. Egy folytonos, lineáris, időinvariáns, több bemenetű/több kimenetű rendszer állapotteretes reprezentációja az  $\mathbf{A}, \mathbf{B}, \mathbf{C}, \mathbf{D}$  paraméter mátrixokkal írható le:

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

$$\mathbf{y} = \mathbf{C}\mathbf{x} + \mathbf{D}\mathbf{u}$$

Ismeretes, hogy a fenti egyenletek (nevezetesen az állapotegyenlet illetve a kimeneti egyenlet) az  $\mathbf{u}$  bemenet és az  $\mathbf{y}$  kimenet között a

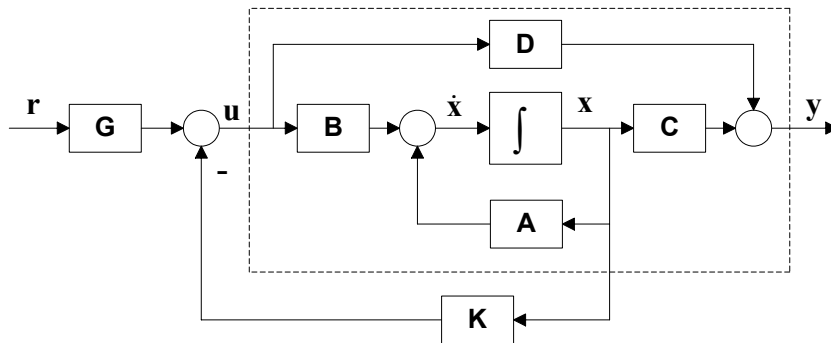
$$\mathbf{C}(s\mathbf{I} - \mathbf{A})^{-1}\mathbf{B} + \mathbf{D}$$

átviteli függvényt valósítja meg. Az  $\{\mathbf{A}, \mathbf{B}, \mathbf{C}, \mathbf{D}\}$  négyessel adott rendszerleírást állapotmodell névvel is illetik, amely modell pólusait a

$$\det(s\mathbf{I} - \mathbf{A}) = 0$$

karakterisztikus egyenlet gyökei határozzák meg.

A rendszer állapotváltozóiról megvalósított, az alábbi ábrán bemutatott visszacsatolás az  $\mathbf{r}$  referencia jel (alapjel) és az  $\mathbf{y}$  kimenőjel közötti átvitel, mind statikus, mind pedig dinamikus értelemben módosítja:



A visszacsatolásban  $\mathbf{K}$  visszacsatolási mátrixot, az előrevezető ágban  $\mathbf{G}$  erősítési mátrixot feltételezve a visszacsatolt (zárt) rendszer egyenletei  $\mathbf{u} = \mathbf{G}\mathbf{r} - \mathbf{K}\mathbf{x}$  következtében az alábbiak szerint írhatók fel:

$$\dot{\mathbf{x}} = (\mathbf{A} - \mathbf{B}\mathbf{K})\mathbf{x} + \mathbf{G}\mathbf{B}\mathbf{r}$$

$$\mathbf{y} = (\mathbf{C} - \mathbf{D}\mathbf{K})\mathbf{x} + \mathbf{D}\mathbf{G}\mathbf{r}$$

Bevezetve az  $\mathbf{A}_F = \mathbf{A} - \mathbf{B}\mathbf{K}$ ,  $\mathbf{B}_F = \mathbf{G}\mathbf{B}$ ,  $\mathbf{C}_F = \mathbf{C} - \mathbf{D}\mathbf{K}$ ,  $\mathbf{D}_F = \mathbf{D}\mathbf{G}$  jelöléseket

$$\dot{\mathbf{x}} = \mathbf{A}_F\mathbf{x} + \mathbf{B}_F\mathbf{r}$$

$$\mathbf{y} = \mathbf{C}_F\mathbf{x} + \mathbf{D}_F\mathbf{r}$$

adódik a zárt rendszerre, amelynek karakterisztikus egyenlete:

$$\det(s\mathbf{I} - \mathbf{A}_F) = \det(s\mathbf{I} - \mathbf{A} + \mathbf{B}\mathbf{K}) = 0.$$

A nyitott és zárt rendszer karakterisztikus egyenletét összevetve látható, a nyitott rendszer pólusai csak  $\mathbf{A}$  értékétől, míg a zárt rendszer pólusai az  $\{\mathbf{A}, \mathbf{B}, \mathbf{K}\}$  hármastól függenek. A zárt rendszer tulajdonságait úgy tervezzük meg, hogy előírjuk, hová kerüljenek a zárt rendszer pólusai a komplex síkban. Matematikailag olyan  $\mathbf{K}$  visszacsatolási mátrixot keresünk, amely a karakterisztikus egyenlet gyökeit éppen e kívánt értékekre helyezi el.

Az állapotvisszacsatolós rendszer tervezése az alábbi lépésekben történik:

- válasszuk meg a zárt rendszer pólusait
- egy bemenetű/egy kimenetű rendszerek esetén a  $\mathbf{K}$  visszacsatolási mátrixot az Ackermann formula segítségével határozhatjuk meg (a Matlab *acker* parancsának közvetlen alkalmazásával)
- az állandósult állapotbeli átvitelre megfogalmazott követelmények alapján határozzuk meg a  $\mathbf{G}$  erősítési mátrixot.

*Példa.*

Tekintsük az alábbi átviteli függvénnyel adott rendszert:

$$H(s) = \frac{6}{(s+1)(s+2)(s+3)}$$

A választott rendszer egy bemenetű/egy kimenetű, így a  $\mathbf{K}$  visszacsatolási mátrix ebben az esetben visszacsatolási vektort, a  $\mathbf{G}$  erősítési mátrix pedig erősítési vektort fog jelenteni. Vegyük észre, hogy a rendszer statikus erősítése egységnyi. A rendszert Matlab szinten adjuk meg  $\mathbf{p}_o$  pólusaival, majd transzformáljuk át a rendszert állapotteres formába:

- »  $\mathbf{p}_o = [-1; -2; -3];$
- »  $[\mathbf{A}, \mathbf{b}, \mathbf{c}, \mathbf{d}] = \mathbf{tf2ss}(6, \mathbf{poly}(\mathbf{p}_o))$

$$\mathbf{A} = \begin{bmatrix} -6 & -11 & -6 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix}, \mathbf{B} = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}, \mathbf{C} = [0 \ 0 \ 6], \mathbf{D} = 0$$

Jelöljük ki a zárt rendszer  $\mathbf{p}_c$  pólusait:

- »  $\mathbf{p}_c = [-6; -3+i*4; -3-i*4];$

majd határozzuk meg azt a visszacsatolási vektort, amely a nyitott rendszer  $\mathbf{p}_o = [-1 \ -2 \ -3]$  pólusait a zárt rendszer  $\mathbf{p}_c = [-6 \ -3+4j \ -3-4j]$  pólusaiba helyezi át. Az Ackermann formula analitikus alakja

$$\mathbf{k} = [0, \dots, 0, 1] \mathbf{C}_x^{-1} \alpha_c(\mathbf{A})$$

formában állítja elő a keresett visszacsatolási vektort, ahol  $\mathbf{C}_x$  a nyitott rendszer irányíthatósági mátrixa,  $\alpha_c(s)$  a zárt rendszer karakterisztikus egyenlete (amelyet az előírt pólusok határoznak meg!),  $\alpha_c(\mathbf{A})$  pedig e polinom, mint mátrix polinom helyettesítési értéke az  $\mathbf{A}$  helyen. Matlab szinten mindez egyetlen utasítás:

- »  $\mathbf{k} = \mathbf{acker}(\mathbf{A}, \mathbf{b}, \mathbf{p}_c)$

$$\mathbf{k} = [6 \ 50 \ 144]$$

- »  $\mathbf{t} = 0:0.1:6;$
- »  $\mathbf{subplot}(211), \mathbf{step}(\mathbf{A}, \mathbf{b}, \mathbf{c}, \mathbf{d}, 1, \mathbf{t});$
- »  $\mathbf{subplot}(212), \mathbf{step}(\mathbf{A}-\mathbf{b}*\mathbf{k}, \mathbf{b}, \mathbf{c}, \mathbf{d}, 1, \mathbf{t});$

Látható, hogy a pólusok balra tolásával az átmeneti függvény tranziense gyorsabb lefutásúvá vált, ugyanakkor – feltételezve, hogy a zárt körben is egységnyi statikus átvitelt szeretnénk biztosítani - a zárt rendszer állandósult állapotbeli viselkedése nem elfogadható. A megfelelő állandósult állapot biztosításának egy lehetséges módja a  $\mathbf{G}$  erősítési vektor beállítása. Az eljárás részletei jól követhetők az alábbi Matlab utasítás sorsal, ahol *GainOL* a nyitott kör, *GainCL* pedig a zárt kör statikus erősítését jelöli:

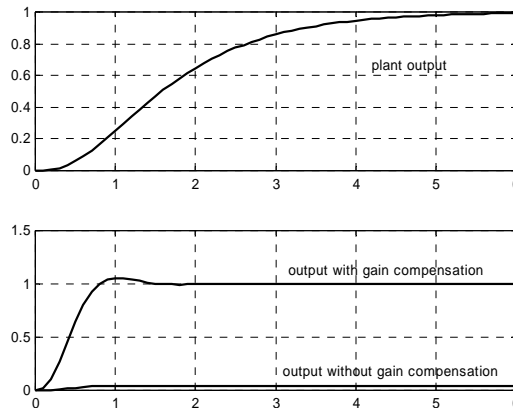
- »  $\mathbf{GainOL} = \mathbf{dcgain}(\mathbf{A}, \mathbf{b}, \mathbf{c}, \mathbf{d})$
- $\mathbf{GainOL} = 1$

```

» GainCL=dcgain(A-b*k,b,c,d)
GainCL = 0.04
» G=GainOL/ GainCL
» subplot(212),step(A-b*k,G*b,c,d)

```

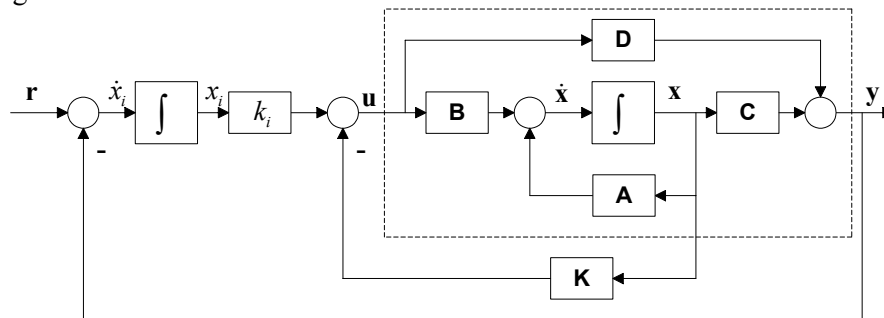
Az alábbi ábra a nyitott és zárt rendszer átmeneti függvényét mutatja be, a zárt rendszer viselkedését  $G=1$ , valamint  $G = GainOL / GainOC = 25$  értékre is láthatjuk:



Az ábrán jól látható a dinamikus tulajdonságoknak a pólusát helyezés következtében előálló javulása.

### Integrátor beiktatása a visszacsatolási hurokba

Ami a zárt kör dinamikus tulajdonságait illeti, az állapotviszacsatolás önmagában a korábbiakban frekvenciatartománybeli megfontolások alapján származtatott PD kompenzáció alkalmazásánál tapasztalt hatást idézi elő. A statikus viszonyokat ugyanakkor egy a visszacsatolási *hurkon kívül elhelyezett* erősítés határozza meg, amelyet a rendszerparaméterek ismeretében határoztunk meg. Ez azt jelenti, hogy az erősítés meghatározása érzékeny a rendszerparaméterek ismeretének pontosságára. Ezen túlmenően, a kimenetet érő zavaró hatásokat hurkon kívül elhelyezett elemekkel nem lehet kompenzálni. Következésképpen a statikus hiba elhárítására - a frekvenciatartománybeli megfontolásokhoz hasonló módon - integrátort iktatunk be a hurokba. A beiktatott integrátort is tartalmazó megoldást az alábbi ábra részletezi:



Látható, hogy az integrátor  $x_i$  kimenetét állapotváltozóként figyelembe véve az állapotváltozók száma eggyel növekedett. Egy bemenetű/egy kimenetű rendszert és  $\mathbf{D} = \mathbf{0}$  közvetlen input-output átvitelt feltételezve a *zárt rendszer* egyenletei egyetlen vektor-mátrix egyenletbe foglalhatók:

$$\dot{\mathbf{x}}_e = \begin{bmatrix} \dot{\mathbf{x}} \\ \dot{x}_i \end{bmatrix} = \begin{bmatrix} \mathbf{A} - \mathbf{b}\mathbf{k} & \mathbf{b}k_i \\ -\mathbf{c} & 0 \end{bmatrix} \begin{bmatrix} \mathbf{x} \\ x_i \end{bmatrix} + \begin{bmatrix} \mathbf{0} \\ 1 \end{bmatrix} r = \mathbf{A}_c \mathbf{x}_e + \mathbf{b}_c r \quad ,$$

ahol  $\mathbf{A}_c$  tovább részletezhető az alábbiak szerint:

$$\mathbf{A}_c = \begin{bmatrix} \mathbf{A} - \mathbf{b}\mathbf{k} & \mathbf{b}k_i \\ -\mathbf{c} & 0 \end{bmatrix} = \begin{bmatrix} \mathbf{A} & \mathbf{0} \\ -\mathbf{c} & 0 \end{bmatrix} - \begin{bmatrix} \mathbf{b} \\ 0 \end{bmatrix} [\mathbf{k} \quad -k_i] = \mathbf{A}_e - \mathbf{b}_e \mathbf{k}_e$$

A zárt rendszer pólusait a

$$\det(s\mathbf{I} - \mathbf{A}_e + \mathbf{b}_e \mathbf{k}_e) = 0$$

karakterisztikus egyenlet gyökei határozzák meg, így a kibővített rendszer megválasztott  $\mathbf{p}_{ce}$  pólusait biztosító  $\mathbf{k}_e$  visszacsatolási vektort ismét az Ackermann formulával számíthatjuk ki. Ehhez azonban először a kibővített rendszer paramétermátrixait kell előállítani:

```

» z=[0;0;0];
» Ae=[A z;c 0]
Ae =
 -6 -11 -6 0
 1 0 0 0
 0 1 0 0
 0 0 6 0

» be=[b;0]
be =
 1
 0
 0
 0

```

Jelöljük ki a zárt rendszer pólusait:

```

» pce=[-9 -6 -3+i*4 -3-i*4];
» ke=acker(Ae,be,pce)

```

$$\mathbf{k}_e = [15 \quad 158 \quad 693 \quad 225]$$

A kibővített erősítési vektor első három eleme az eredeti állapotváltozóról történő visszacsatolást valósítja meg, míg a negyedik  $k_i$  elem a mesterségesen bevezetett integrátorhoz tartozik:

```

» k=ke(1:3)

```

```

k =
 15 158 693
» ki=ke(4)
ki =
 225
» Ac=[A-b*k b*ki;-c 0]
Ac =
 -21 -169 -699 225
 1 0 0 0
 0 1 0 0
 0 0 -6 0

```

```

» bc=[z;1]

```

```

bc =
 0
 0

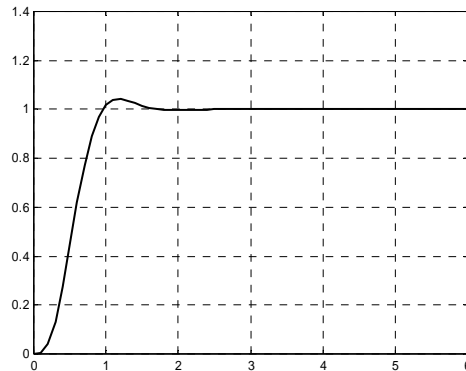
```

```

0
1
» cc=[c 0]
cc =
0 0 6 0
» dc=0;
» subplot(111), step(Ac,bc,cc,dc,1,t)

```

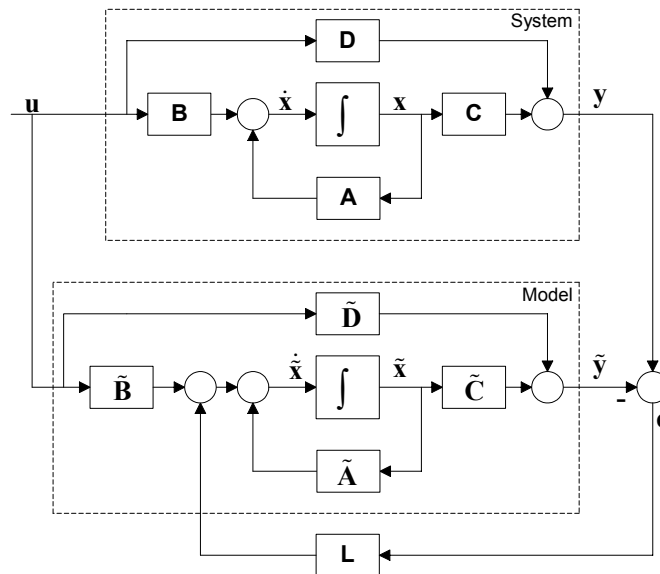
A zárt kör átmeneti függvénye:



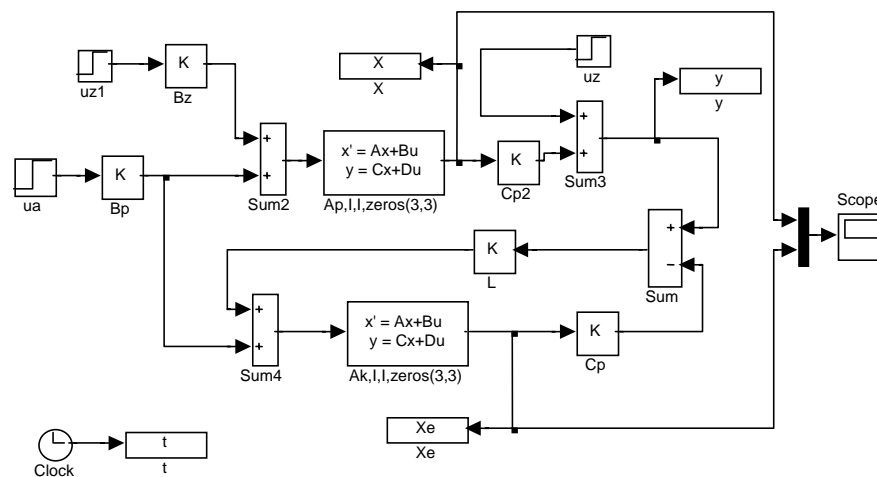
Látható, hogy a zárt kör dinamikus és statikus viselkedése is megfelelő.

### Állapotbecslés

A gyakorlati alkalmazások döntő részében a folyamatok műszerezése magába foglalja a rendszer kimenetének mérését, azonban az állapotváltozók közvetlen méréssel nem hozzáférhetők. Ilyen esetekben az állapotviszacsatolásos szabályozási technika alkalmazhatóságát a nem mért állapotváltozók becslése kell, hogy kiegészítse. Az állapotváltozók becslésére szolgáló ún. Kalman szűrő blokkvázlata az alábbi ábrán látható. A Kalman szűrő részét képező modell felépítése követi a rendszer felépítését, a rendszer és a modell kimenetének különbsége pedig egy hibajelet állít elő. E hibának a modellbe történő visszacsatolása biztosítja azt, hogy a modell állapotváltozói minél kisebb hibával kövessék (másolják le) a rendszer állapotváltozóit. A becslő hálózat Simulink-ben könnyen realizálható, ugyanis a becslő hálózat előírt pólusai alapján az  $L$  erősítés ismét az Ackermann formulával számítható. A becslő hálózat pólusait úgy szokásos megválasztani, hogy a zárt rendszer tranziens viselkedését az állapotbecslés túlságosan ne lassítsa le, ugyanis a zárt rendszer állapotviszacsatolása a becsült állapotváltozó felhasználásával történik.



A becslő hálózat Simulink modelljét az alábbi ábra adja meg.



A szakaszt és modelljét a Simulink Continuous könyvtárának State-Space valamint Math könyvtárának Matrix Gain blokkjaiból építjük fel. A  $C$  paraméter különválasztására azért van szükség, mert nemcsak a kimenőjelhez, hanem az állapotváltozókhöz is hozzá akarunk férni. A  $B$  paramétert is különválasztjuk az állapotmodell bloktól, mivel a modell állapotváltozóinak deriváltjait módosítjuk, így a deriváltakhoz is hozzá kell férnünk.

A becslő hálózat paramétereit (az  $L$  vektor elemeit) a zárt becslő kör karakterisztikus egyenlete gyökeinek előírásával az Ackermann képlettel, a Matlab »  $L = \text{acker}(A', C', Se)$  ' utasításával határozhatók meg.  $Se$  a becslő kör előírt pólusainak vektora. A becslő kör legyen gyorsabb, mint a szakasz, és legyen gyorsabb, mint a szabályozás az állapotviszacsatolással. (A transzponálásra a mátrixok, vektorok méreteinek egyeztetése miatt van szükség.)

A szakasz legyen a szabályozási példánál is vizsgált háromtárolós arányos szakasz (az integráló állapotváltozóval való kiterjesztés nélkül). Legyen mindhárom állapotváltozó kezdeti értéke 1. Az alapjel és a zavarás értéke legyen zérus.

A becslő hálózat előírt pólusai legyenek

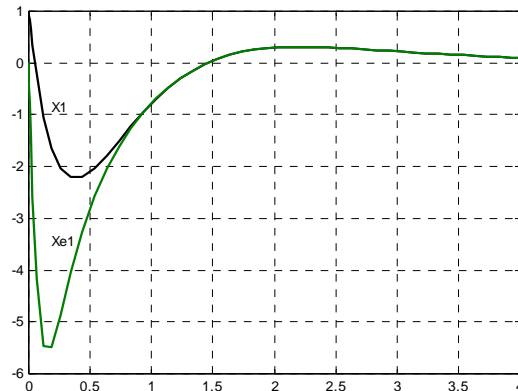
$$\gg \mathbf{S_e} = [-7 \quad -7 \quad -7]$$

Ezzel  $\mathbf{L}' = [-17.3333 \quad 7.6667 \quad 2.5000]$  érték adódik.

Szimuláljuk az állapotbecslés lefolyását, majd a Matlab felületről ábrázoljuk az első állapotváltozó és becslült értékének időbeli lefolyását.

$$\gg \text{plot}(t, [\mathbf{x}(:, 1), \mathbf{x}_e(:, 1)]), \text{grid}$$

A szimuláció az állapotváltozók gyors beállítását mutatja.



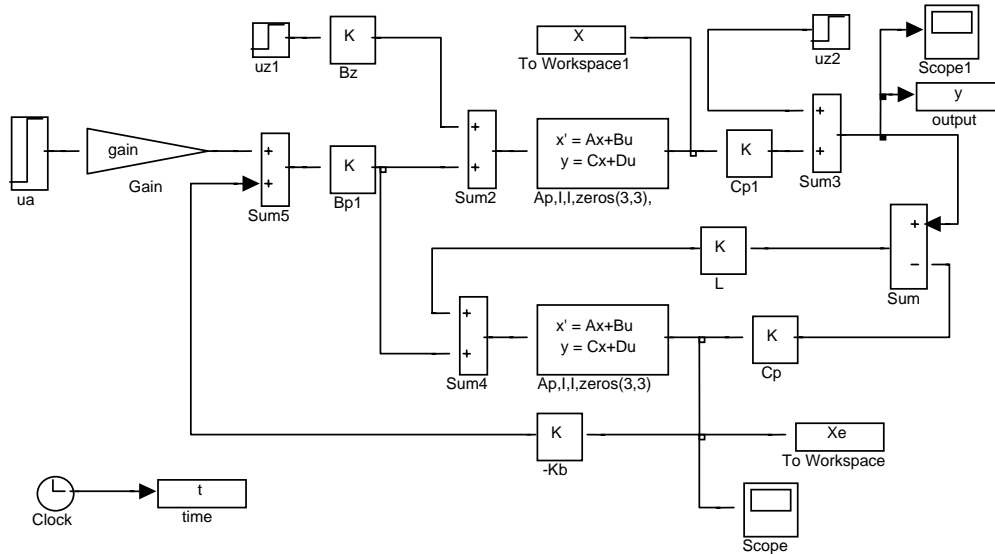
**Feladat:** Állítsuk a kezdeti feltételek értékét zérusra, és az  $uz$  zavarás értékét 1-re. A szimulációt lefuttatva látható, hogy statikus eltérés lép fel a tényleges és becslült állapotváltozók között.

Ennek kiküszöböléséhez a zavarás állapotváltozóival célszerű kibővíteni a szakasz állapotegyenletét és az állapotbecslést a kibővített rendszerre ajánlatos elvégezni. (Erre azonban a továbbiakban nem térünk ki.)

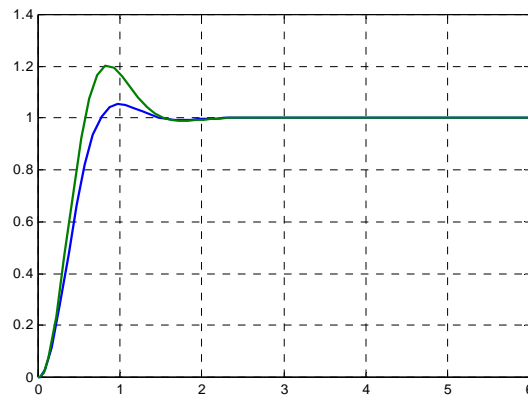
### Állapotviszacsatolás a becslült állapotváltozókról

Az állapotbecslés és állapotviszacsatolás egymástól függetlenül elvégezhető feladatok (szeparációs elv). Ha az állapotváltozók nem hozzáférhetők, az állapotviszacsatolós szabályozást a becslült állapotváltozók visszacsatolásával valósíthatjuk meg azzal a  $\mathbf{K}$  visszacsatoló mátrix-szal, amellyel a hozzáférhető állapotváltozókat csatolnánk vissza. A Simulink vázlatot az alábbi ábra mutatja. Kezdeti feltételekre és alapjelre az alapjel statikus kompenzációjával a szabályozás jól működik. A zavarások hatását azonban csak statikus hibával tudja kiküszöbölni.





Futtassuk le azonos kezdeti értékekre  $[1 \ 0 \ 0]$  és egységugrás alapjelre a Simulink programot a visszacsatolást a tényleges, illetve a becült állapotváltozókról megvalósítva. A kimenőjeleket (illetve az állapotváltozókat) ábrázoljuk egy diagramban. Látható, hogy a becült állapotváltozókról való visszacsatolással a túllendülés kissé nagyobb. Hasonlítsuk össze a beavatkozójeleket is.



### Feladat:

Építsünk Simulink vázlatot az integráló állapotváltozóval kibővítve a rendszert, állapotbecsléssel és a becült állapotváltozókról történő visszacsatolással.

## Állapotviszacsatolás mintavételes rendszerekre

Vizsgáljuk meg az állapotviszacsatolást mintavételes rendszerekre. A folytonos rendszerek körében folytatott vizsgálataink azt mutatták, hogy a szabályozási kör pólusai a folyamat állapotváltozóinak a folyamat bemenetére való visszacsatolásával előírt értékekre állíthatók. Most az állapotviszacsatolást mintavételes rendszerre kívánjuk alkalmazni.

Induljunk ki a folytonos állapotegyenlet általános megoldásából:

$$\mathbf{x}(t) = e^{\mathbf{A}(t-t_0)} \mathbf{x}(0) + \int_{t_0}^t e^{\mathbf{A}(t-\tau)} \mathbf{B} \mathbf{u}(\tau) d\tau$$

A mintavételezés során alkalmazzunk zérusrendű tartószervet.

Legyen  $t_0$  egy mintavételi intervallum kezdete,  $t_0 = iT$ , ahol  $T$  a mintavételezési idő, az intervallum vége ekkor  $t = (i+1)T$ . Ebben a tartományban  $u(\tau) = \text{állandó} = u(iT)$ , és így kiemelhető az integrálás elé. Az integrálást elvégezve

$$\mathbf{x}[(i+1)T] = e^{\mathbf{A}T} \mathbf{x}(iT) + \mathbf{A}^{-1} (e^{\mathbf{A}T} - \mathbf{I}) \mathbf{B} u(iT)$$

A mintavételezett rendszer állapotegyenlete:

$$\mathbf{x}_{i+1} = \mathbf{A}_d \mathbf{x}_i + \mathbf{B}_d \mathbf{u}_i$$

$$\mathbf{y}_i = \mathbf{C}_d \mathbf{x}_i + \mathbf{D}_d \mathbf{u}_i$$

ahol

$$\mathbf{A}_d = e^{\mathbf{A}T}; \quad \mathbf{B}_d = \mathbf{A}^{-1} (e^{\mathbf{A}T} - \mathbf{I}) \mathbf{B}; \quad \mathbf{C}_d = \mathbf{C}; \quad \mathbf{D}_d = \mathbf{D}.$$

Ismeretes, hogy a gyakorlati rendszerekre tipikusan  $\mathbf{D}_d = 0$  áll fenn, továbbá a nyitott rendszer

$$\det(z\mathbf{I} - \mathbf{A}_d) = 0$$

karakterisztikus egyenlete a  $z$  tartományban adja meg a rendszer pólusait.

Az Ackermann formula a megoldás egyik kulcseleme volt a folytonos rendszereknél, vizsgáljuk meg ezt az összefüggést mintavételes rendszerekre. A nyitott rendszer diszkrét állapotegyenlete:

$$\mathbf{x}_{i+1} = \mathbf{A}_d \mathbf{x}_i + \mathbf{B}_d \mathbf{u}_i$$

$$\mathbf{y}_i = \mathbf{C}_d \mathbf{x}_i$$

Állapotviszacsatolás:  $\mathbf{u}_i = -\mathbf{K} \mathbf{x}_i$

A zárt rendszer állapotegyenlete:  $\mathbf{x}_{i+1} = (\mathbf{A}_d - \mathbf{B}_d \mathbf{K}) \mathbf{x}_i$

A zárt rendszer előírt karakterisztikus polinomja:

$$\alpha_c(z) = \det(z\mathbf{I} - (\mathbf{A}_d - \mathbf{B}_d \mathbf{K})) = (z - p_{c1})(z - p_{c2}) \dots (z - p_{cn})$$

ahol  $p_{c1}, p_{c2}, \dots, p_{cn}$  a zárt rendszernek a  $z$  tartományban előírt pólusai.

Legyen a szakasz irányíthatósági mátrixa:  $\mathbf{M}_c$

$$\mathbf{M}_c = \begin{bmatrix} \mathbf{B}_d & \mathbf{A}_d \mathbf{B}_d & \dots & \mathbf{A}_d^{n-1} \mathbf{B}_d \end{bmatrix}, \quad n = \dim x$$

Az Ackermann képlet szerint a szakasz  $\mathbf{A}_d, \mathbf{B}_d$  mátrixaiból és a zárt rendszer előírt pólusaihoz tartozó karakterisztikus polinomból a  $\mathbf{K}$  visszacsatoló vektor az alábbi összefüggéssel számítható:

$$\mathbf{K} = [0 \quad \dots \quad 0 \quad 1] \mathbf{M}_c^{-1} \alpha_c(\mathbf{A}_d)$$

ahol  $\alpha_c(\mathbf{A}_d)$  a zárt rendszer karakterisztikus polinomja  $z = \mathbf{A}_d$  helyettesítéssel.

Az állapotokat visszacsatoló  $\mathbf{K}$  vektor kiszámítását a Matlab az *acker* utasítással támogatja:

$$\mathbf{K} = \text{acker}(\mathbf{A}_d, \mathbf{B}_d, \mathbf{S}_d)$$

$\mathbf{S}_d$  a visszacsatolt rendszer előírt pólusait (az  $\alpha_c(z) = 0$ ) egyenlet előírt gyökeit) tartalmazó vektor.

Ha az előírt pólusokat folytonos időben adjuk meg, ezek diszkrét megfelelőit a  $z = e^{sT}$  transzformációval határozhatjuk meg, ahol  $T$  a mintavételezési idő.

*Példa*

Legyen a folytonos szabályozott szakasz az alábbi háromtárolós arányos tag:

$$\mathbf{u} \quad \boxed{\frac{A}{1+sT_3}} \quad \mathbf{x}_3 \quad \boxed{\frac{1}{1+sT_2}} \quad \mathbf{x}_2 \quad \boxed{\frac{1}{1+sT_1}} \quad \mathbf{x}_1 = \mathbf{y}$$



A paraméterek értékei:  $A=1$ ,  $T_1=0.2$ ,  $T_2=2$ ,  $T_3=4$ .

Az állapotváltozókat vegyük fel az ábra szerint. Adjuk meg a szakasz folytonos állapotváltozós egyenleteit, majd  $T=0.2$  mintavételezési idő mellett határozzuk meg a diszkrét állapotegyenletet zérusrendű tartószerv feltételezésével.

a./ Tervezzünk állapotviszacsatolást a diszkrét zárt rendszer pólusainak előírásával. A pólusokat folytonos időben specifikáljuk, majd a  $z = e^{sT}$  transzformációval térjünk át diszkrét időre. Bevezetjük a  $T_{sum} = T_1 + T_2 + T_3$  összeg időállandót (ennek az  $A$  átviteli tényezővel megszorított értéke adja a szakasz átmeneti függvénye és annak állandósult értéke eltérésének integrálját, az ún. lassító területet). Ehhez képest gyorsítjuk fel a rendszert.

A zárt rendszer előírt pólusai legyenek:

$$s_{1,2} \text{ egy kéttárolós lengő tag pólusai, } \omega_0 = \frac{5}{T_{sum}}; \quad \zeta = 0.7.$$

$$s_3 = -\max\{\omega_0, 1/\min\{T_1, T_2, T_3\}\}$$

Vizsgáljuk a rendszer működését kezdeti feltételekre, illetve alapjelkövetésre és zavarelhárításra.

*Megoldás*

A folytonos szakasz állapotegyenlete:

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \end{bmatrix} = \begin{bmatrix} -5 & 5 & 0 \\ 0 & -0.5 & 0.5 \\ 0 & 0 & -0.25 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 0.25 \end{bmatrix} u$$

$$y = [1 \quad 0 \quad 0]X + 0 \cdot u$$

A zárt rendszerre előírt pólusok:  $s_{1,2} = -\zeta\omega_0 \pm j\omega_0\sqrt{1-\zeta^2} = -0.5645 \pm j0.576$

$$s_3 = -5.$$

$$\gg \mathbf{S} = [-0.5645 + i*0.576, -0.5645 - i*0.576, -5]$$

Diszkrét megfelelője  $T_s=0.2$  mintavételezési idővel:

$$\gg \mathbf{T}_s = 0.2$$

$$\gg \mathbf{S}_d = \mathbf{exp}(\mathbf{S} * \mathbf{T}_s)$$

A kapott értékek:  $0.8873 + 0.1027i$ ,  $0.8873 - 0.1027i$ ,  $0.3679$ .

*Feladat:*

A várható gyorsítás szemléltetésére ábrázoljuk a szakasz és az előírt pólusokkal rendelkező rendszer átmeneti függvényeit. Alkalmazzuk a *step* ill. *dstep* MATLAB utasításokat.

Határozzuk meg a diszkrét állapotegyenletet:

$$\gg [\mathbf{Ad}, \mathbf{Bd}, \mathbf{Cd}, \mathbf{Dd}] = \mathbf{c2dm}(\mathbf{A}, \mathbf{B}, \mathbf{C}, \mathbf{D}, \mathbf{T}_s, 'zoh')$$

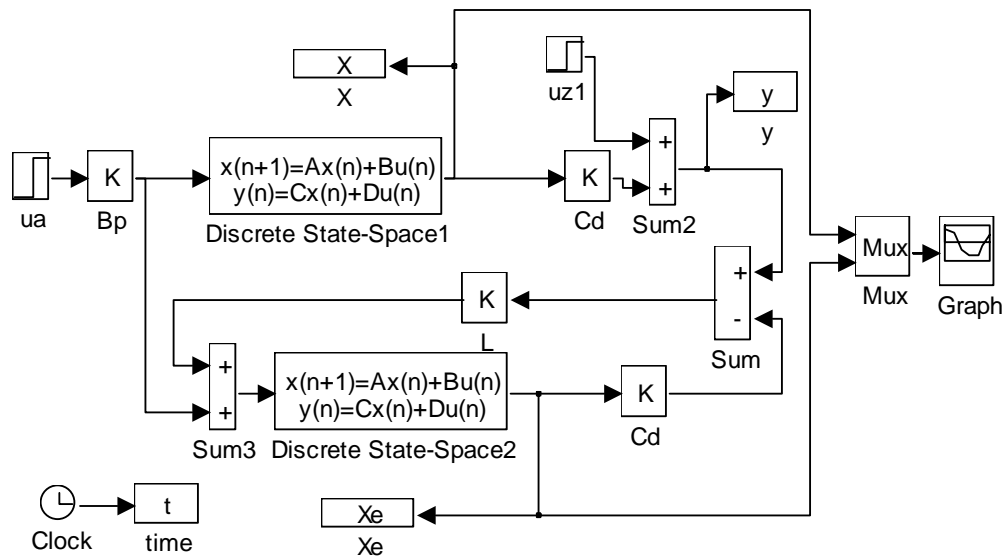
Alkalmazzuk az Ackermann formulát az állapotviszacsatoló vektor meghatározásához.

$$\gg \mathbf{K} = \mathbf{acker}(\mathbf{Ad}, \mathbf{Bd}, \mathbf{S}_d)$$

Az állapotviszacsatolásos kompenzálás Simulink blokk diagramja az alábbi ábrán látható. A kapcsolást kiegészítettük a folytonos szakasz modelljével is, hogy a mintavételei pontok közötti viselkedés is vizsgálható legyen. Vizsgáljuk a szabályozás működését  $[1 \ 1 \ 1]'$  kezdeti feltételek







Ha a szakasz ismert, modelljét megépítjük. A szakaszt és modelljét ugyanazzal a bemenőjellel gerjesztjük. A szakasz és a modell kimenőjeleit összehasonlítva a hibajelet használjuk fel a modell állapotváltozóinak beállítására az  $L$  paramétervektoron keresztül. Akkor várható, hogy a modell állapotváltozóinak értéke gyorsan megközelíti és követi a tényleges állapotváltozók értékét, ha a becslő hálózat jóval gyorsabb a szakasz tranzienseinél. Előírhatjuk a becslő hálózat pólusait, és az Ackermann formula alkalmazásával meghatározhatjuk az  $L$  vektor paramétereit.

**A szakasz legyen a szabályozási példánál is vizsgált háromtárolós arányos szakasz. Legyen mindhárom állapotváltozójának kezdeti értéke 1. Az alapjel és a zavarás értéke legyen zérus.**

A becslő hálózat előírt pólusai legyenek valósak és azonos értékűek, és az állapotváltozókról visszacsatolt hálózat leggyorsabb időállandójánál 3-szor gyorsabb viselkedést írjanak elő (konjugált komplex pólusok esetén tekintsük az abszolút érték reciprokát).

A folytonos zárt rendszer előírt pólusai voltak:

$$\gg \mathbf{S} = [-0.5645 + i \cdot 0.576, -0.5645 - i \cdot 0.576, -5]$$

$$\gg \mathbf{abs}(\mathbf{S}) = [0.8065 \ 0.8065 \ 5]$$

Legyen tehát

$$\gg \mathbf{Se} = [-15 \quad -15 \ -15].$$

Diszkrétben

$$\gg \mathbf{Sed} = \exp(\mathbf{Ts} \cdot \mathbf{Se})$$

A becslő hálózat paramétereit (az  $L$  vektor elemei) az

$$\gg \mathbf{L} = \mathbf{acker}(\mathbf{Ad}', \mathbf{Cd}', \mathbf{Sed})'$$
 utasítással határozhatók meg.

A kapott értékek: [2.0746 3.1280 12.8569]'

A szimuláció az állapotváltozók gyors beállítását mutatja. Ábrázoljuk a tényleges és becslő állapotváltozók lefolyását egy diagramban.

$$\gg \mathbf{plot}(\mathbf{t}, \mathbf{X}, \mathbf{t}, \mathbf{Xe}), \mathbf{grid}$$

Állítsuk a kezdeti feltételek értékét zérusra, és az  $uz1$  zavarás értékét 1-re. A szimulációt lefutattva látható, hogy statikus eltérés lép fel a tényleges és becslő állapotváltozók között. Ennek kiküszöböléséhez a zavarás állapotváltozóival célszerű kibővíteni a szakasz állapotegyenletét és az állapotbecslést a kibővített rendszerre ajánlatos elvégezni (Erre azonban itt nem térünk ki.)

A szabályozást a becslő állapotváltozók visszacsatolásával valósíthatjuk meg.