MŰEGYETEM 1782

Budapesti Műszaki és Gazdaságtudományi Egyetem
Irányítástechnika és Informatika Tanszék

Irányításelmélet - (BMEVIIIM016)
Házi Feladat

Név: Szőke Kálmán Benjamin
Neptun: SLZ0UE
Dátum: 2014. december 7.

Tartalomjegyzék

Ábrák jegyzéke

1. I. feladat

1.1. Szakasz identifikáció

A feladatot leíró folytonos átviteli függvény a következő alakban keresendő:

$$W(s) = \frac{A_p}{(1 + 2\xi\tau\ s + \tau^2 s^2) \cdot (1 + s\alpha\tau)}$$

A feladat diszkrét átviteli függvénye:

$$D(z) = \frac{b_1 + b_2 z^{-1} + b_3 z^{-2}}{1 + a_1 z^{-1} + a_2 z^{-2} + a_3 z^{-3}} z^{-1}$$

Az előírt feltételekből adódó fokszám feltételek a következőek lesznek:
($nA = 3$ mert a nevező polinomja harmadfokú, holtidő fokszáma $nK = 1$)

$$nA = 3$$

$$nK = 1$$

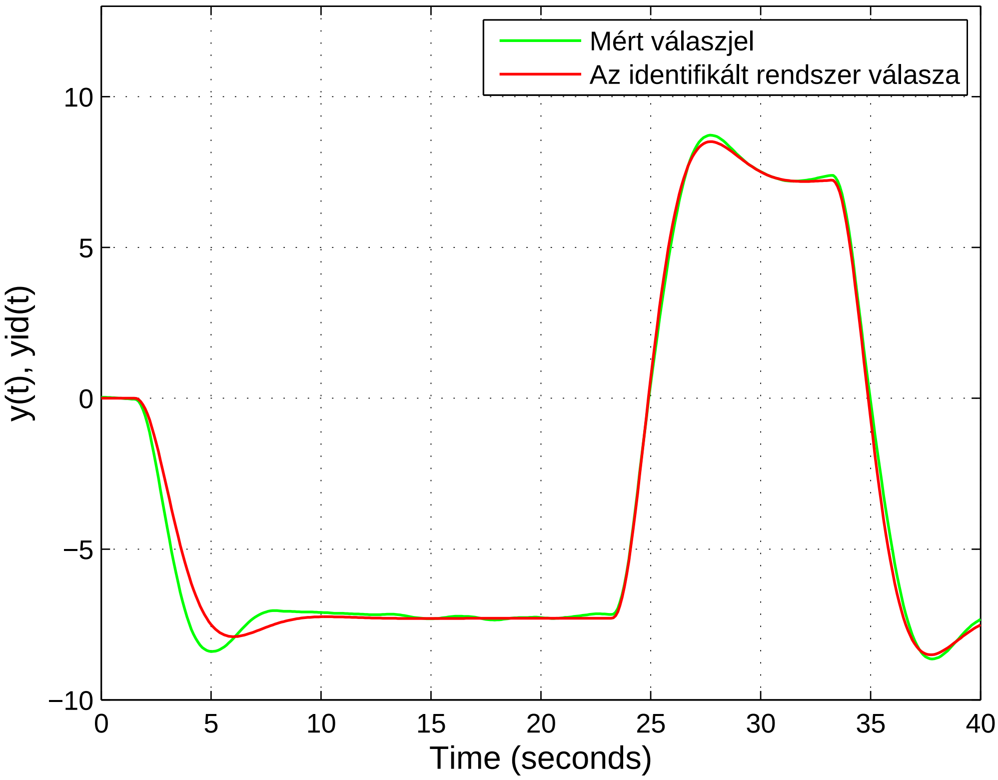
$$nB = nA - 1 + nK$$

$$nC = nA$$

```
load slz0ue meres_u meres_y;
% Ts az a mintavételi idő, amit az identifikáció és a
% szabályozó megtervezése során kell használnia
Ts=0.1;

%% Identifikáció
z=iddata(meres_y,meres_u,Ts);
nA=3;   % nevezo fokszama
nk=1;   % holtido fokszama
nB=nA-1+nk;
nC=nA;
tharmax=armax(z,[nA nB nC nk]); % th struktura
[A,B,C]=th2poly(tharmax);
Darmax=tf(B,A,Ts);
t=(0:Ts:length(meres_u)*Ts-Ts);
yid=lsim(Darmax,meres_u,t);

figure(1);
plot(t,meres_y,'g',t,yid,'r');
legend('Mért válaszjel','Az identifikált rendszer válasza');
xlabel('Time (seconds)');
ylabel('y(t), yid(t)');
```

Az identifikált mintavételes rendszer pólusainak átszámítása folytonos időbe:

$$z_i = e^{s_i T_s} \rightarrow s_i = \frac{\ln z_i}{T_s}$$



1. ábra. A mért válaszjel és az identifikált rendszer válasza.

Az így kapott pólusok a következők:

$$s_{1,2} = -0.575 \pm j0.743$$

$$s_3 = -3.03$$

```
zpoles=roots(A);                    % Diszkret ideju polusok szamitasa
spoles=log(zpoles)/Ts;              % A folytonos ideju polusok szamitasa
%Tcs=-1./spoles;                    % Az idþállandok
spoly=poly(spoles);
D0=polyval(B,1)/polyval(A,1);       % A diszkret ideju atvitel allandosult allapotban
Ap=D0                               % Allandosult allapotban az erosites
Warmax=tf(Ap*spoly(end),spoly);     % Folytonos atvit
zpkW=zpk(Warmax);                   % Folytonos atvit zpkban
Tau=sqrt(1/(zpkW.p{1}(3)* zpkW.p{1}(2)))
alpha=(1/(-1*zpkW.p{1}(1)))/Tau
omega_0=1/Tau;
Sigma_e=-max(real(zpkW.p{1}));
xi=Sigma_e/omega_0
```

Az átviteli függvény folytonos idejű paramétereire a következő értékek adódtak:

$$A_p = 1.8243$$

$$\xi = 0.6122$$
$$\tau = 1.0645$$
$$\alpha = 0.3101$$

Így az átviteli függvény a megadott formában a következő:

$$W(s) = \frac{4.8772}{(0.8825 + 1.15s + s^2) \cdot (s + 3.03)}$$

1.2. Két szabadságfokú szabályozó tervezése

Zárt körben a két szabadságfokú szabályozó struktúrája a következő egyenlettel írható le:

$$\frac{\frac{TB}{AR}}{1 + \frac{SB}{AR}} = \frac{TB}{AR + SB} = \frac{B_m}{A_m} = \frac{B_m A_0}{A_m A_0}$$

A rendszerre a feladatban előírt paraméterek a következők voltak:

$$\xi = 0.07$$
$$\omega_0 = \frac{1}{\tau} = 0.9394$$

```
%% Szabalyozotervezes
xi_spec=0.7;
w0=1/Tau;
sdom1=-w0*xi_spec+1j*w0*sqrt(1-xi_spec^2);
sdom2=conj(sdom1);
scinf=-3*abs(real(sdom1));
soinf=-5*abs(real(sdom1));
lint=1;                    % Integratorok szama
DWarmax=c2d(Warmax, Ts, 'zoh');
zdom1=exp(sdom1*Ts);
zdom2=exp(sdom2*Ts);
zcinf=exp(scinf*Ts);
zoinf=exp(soinf*Ts);
```

A számítások elvégzése után a szabályzó pólusainak értéke a következőkre adódtak:

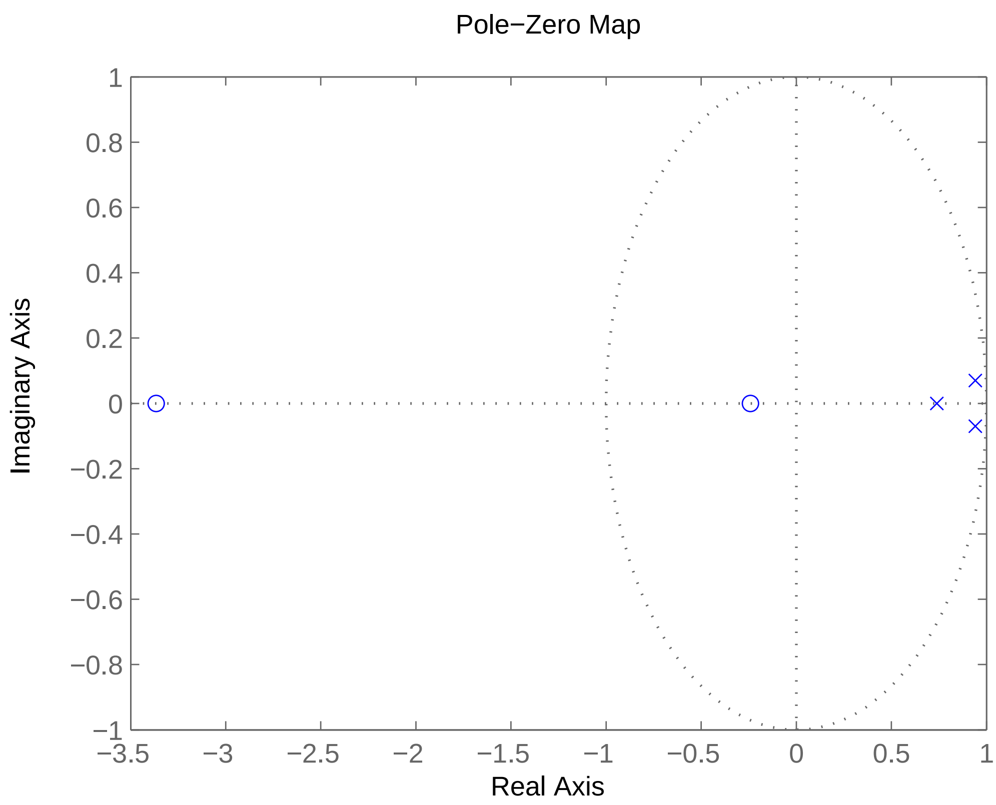
$$s_{1,2} = -0.6576 \pm j0.6709$$
$$s_{c\infty} = -3\omega_0\xi = -1.9727$$
$$s_{0\infty} = -5\omega_0\xi = -3.2879$$

Az ismertetett egyenletben az egyenlőség az R, S és T polinomok helyes megválasztásával teljesíthető. A B polinomot azonban szét kell választanunk két részre. Az egyik rész a kiejthető, a másik pedig a nem kiejthető pólusokat fogja tartalmazni. Így B faktorizációja a következő:

$$B = B^+ B^-$$

Az instabil és tisztán valós részű zérusok nem ejthetők ki a rendszerből, ezért meg kell vizsgálnunk a rendszerünk zérusait.

```
B=DWarmax.num{1};   % B kinyerese
B=B(2:end);         % Vezeto nullak elhagyas
Broots=roots(B);
Bminus=B;           % B faktorizacio, Bminus a kiejtheto zerusok
Bplus=1;            % B faktorizacio, Bplus a nem kiejtheto zerusok
grB=length(B)-1;    % B fokszama, az egyutthatoi szamanal 1-el kisebb
```



2. ábra. Zérus-Pólus térkép

Látható, hogy a rendszer zérusai tisztán valósak, ezért nem ejthetőek ki! Ezért B polinom minden pólusa a nem kiejthető részbe B^{-}-ba fog kerülni és így $B^{+} = 1$ lesz, hogy B zérusai ne változzanak és a szorzat értéke ne változzon. Így létrehoztuk B megfelelő faktorizációját. Ezután az előadáson levezetett megfontolások alapján, eleget téve a fokszámfeltételeknek felírható egy Diophantoszi egyenletrendszer, amelyből kiadódik $R(z)$, $S(z)$ és $T(z)$.

```
A=DWarmax.den{1};             % A kinyerese
grA=length(A)-1;              % A fokszama, az egyutthatoi szamanal 1-el kisebb
grBminus=length(Bminus)-1;    % Bminus fokszama, az egyutthatoi szamanal 1-el kisebb

%Fokszam feltetelek
grAm=1+grBminus;              % Mivel grBminus>1, ezert nincs plusz tag
grS=grA+lint-1;
```

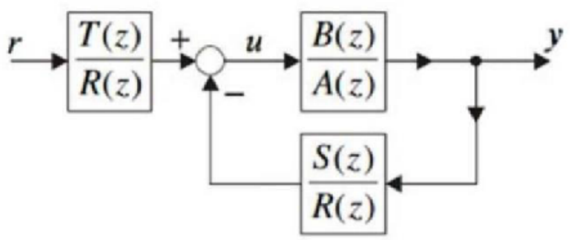
```
grA0=grA+lint-1;              % Nincs plusz tag
grR1prime=grBminus;

%Am es Ao polinomok kiszamitasa
Am=poly([zdom1 zdom2 zcinf]);    % Am gyokei a dominans poluspar es zcinf
A0=poly(ones(1,grA0)*zoinf);     % Ao gyokei zoinf, grAo-szorosa

% Az egyenletrendszer dioA*x=dioC alaku
polyint=[1 -1];               % (1-z)^lint
Atilde=conv(polyint,A);
dioA=[ toeplitz([Atilde 0],[Atilde(1) 0]) toeplitz([Bminus 0 0 0],[Bminus(1) 0 0 0]) ];
Ctilde=conv(Am,A0)-[Atilde 0 0];
dioC=Ctilde(2:end)';
sol=inv(dioA)*dioC;           % Az egyenletrendszer megoldasa

% R1prime monic, igy polinomjanak legmagasabb fokszamu egyutthatoja 1,
R1prime=[1 sol(1:2)'];
% csak utana kovetkeznek a kiszamitott egyutthatok
R=conv(R1prime,polyint);      % R=R1prime*(z-1), mivel lint=1 és Bplus=1
% S polinom egyutthatoi a megoldasvektor tovabbi elemei
S=sol(3:end);
S=S';

% A T polinom számítása
Bmprime=polyval(Am,1)/polyval(Bminus,1);
T=Bmprime*A0;
```

Így a szabályozó polinomjai a következők:

$$R(z) = z^3 - 2.2303z^2 + 1.6405z - 0.4102$$

$$S(z) = 4.1754z^3 - 10.9550z^2 + 9.5407z - 2.7529$$

$$T(z) = 0.3725z^3 - 0.8043z^2 + 0.5789z - 0.1389$$

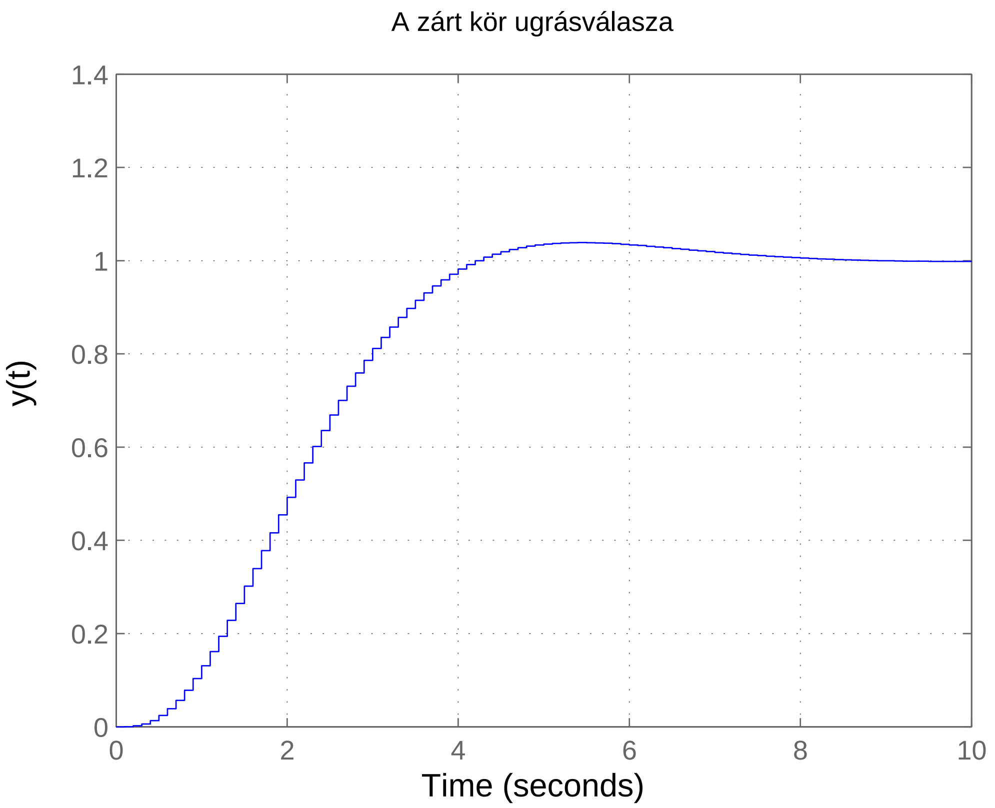


3. ábra. RST-szabályozó

Ezután előállítható a tervezett szabályozó a képen látható formában.

```
% A szabalyozo tagok atviteli fuggvenyei
d_ff=tf(T,R,Ts);                  % Elorecsatolo ag
d_fb=tf(S,R,Ts);                  % Visszacsatol ag
```

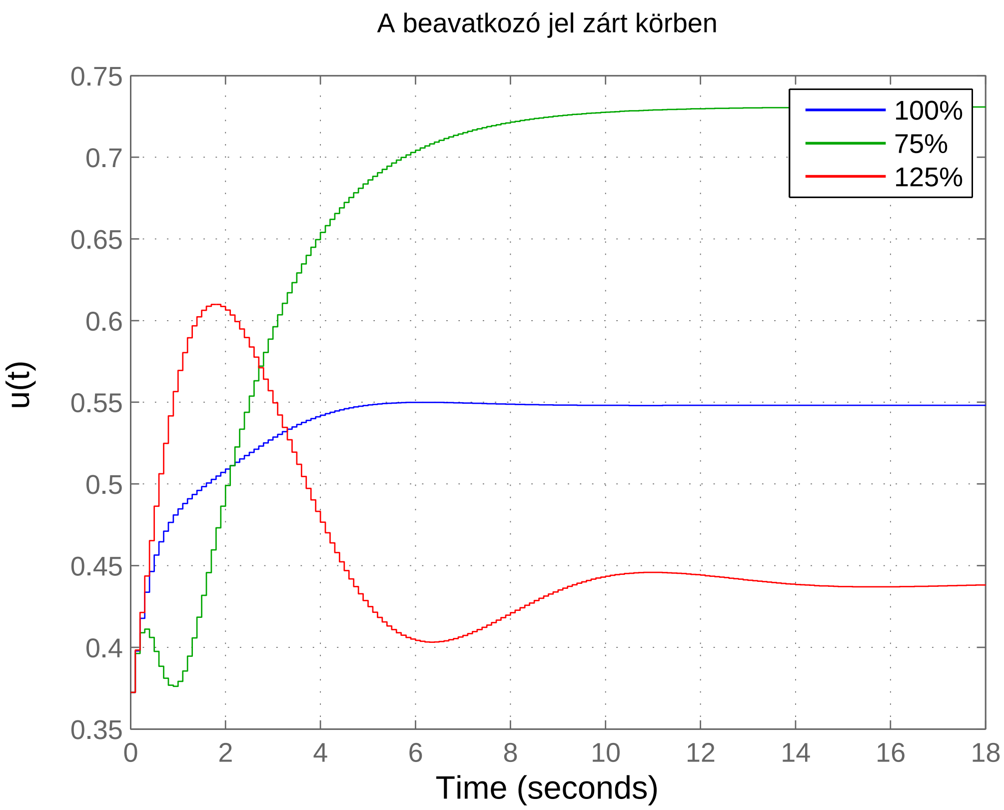
```
dyr=d_ff*feedback(DWarmax,d_fb,-1); % A zart kor r->y atviteli fuggvenye

figure(3);
step(dyr);
title('A zárt kör ugrásválasza');
xlabel('Time');
ylabel('y(t)');
```



4. ábra. Zárt kör ugrásválasza

1.3. Robusztusság vizsgálata

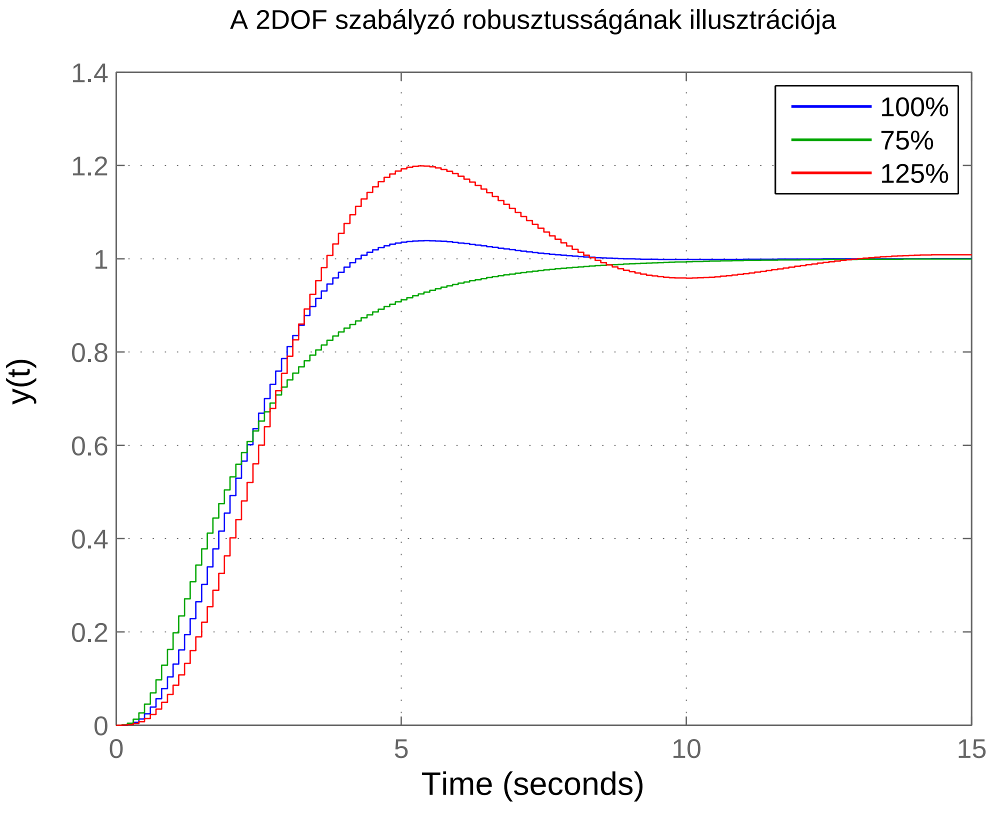
A folytonos idejű szakasz paramétereinek ±25%-kal való megváltoztatásának segítségével ellenőrizzük a szabályozó robusztusságát. A vizsgálathoz egységugrás jelet alkalmazunk és megfigyeljük a rendszer tranziens viselkedését.



5. ábra. Beavatkozó jel zárt körben

```
% A szakasz minden parameteret 75%-ra csokkentjuk
Ap75=Ap*0.75;
xi75=xi*0.75;
Tau75=Tau*0.75;
alpha75=alpha*0.75;
DEN75=conv([Tau75^2 2*xi75*Tau75 1],[alpha75*Tau75 1]);
wp75=tf(Ap75,DEN75);
% A szakasz minden parameteret 125%-ra noveljuk
Ap125=Ap*1.25;
xi125=xi*1.25;
Tau125=Tau*1.25;
alpha125=alpha*1.25;
DEN125=conv([Tau125^2 2*xi125*Tau125 1],[alpha125*Tau125 1]);
wp125=tf(Ap125,DEN125);
% Atteres diszkret idore
DWarmax75=c2d(wp75,Ts,'zoh');
```

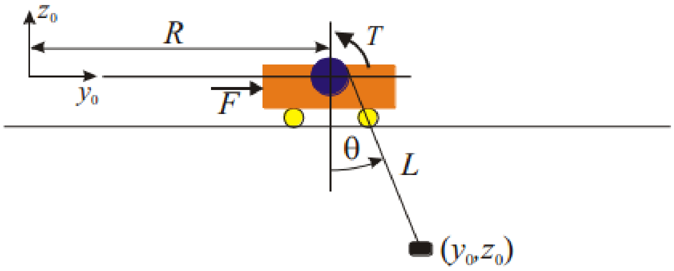
```
DWarmax125=c2d(wp125,Ts,'zoh');
% Robusztussag vizsgalata
dur=d_ff*feedback(tf(1,1,Ts),DWarmax*d_fb,-1);
dur75=d_ff*feedback(tf(1,1,Ts),DWarmax75*d_fb,-1);
dur125=d_ff*feedback(tf(1,1,Ts),DWarmax125*d_fb,-1);

figure(4);
step(dur, dur75, dur125);
legend('100%', '75%', '125%');
title('A beavatkozó jel zárt körben');
xlabel('Time');
ylabel('u(t)');

% Zart kor osszeallitasa
dyr75=d_ff*feedback(DWarmax75,d_fb,-1);
dyr125=d_ff*feedback(DWarmax125,d_fb,-1);

figure(5);
step(dyr,dyr75,dyr125);
legend('100%', '75%', '125%');
title('A 2DOF szabályzó robusztusságának illusztrációja');
```



6. ábra. Robusztusság vizsgálata

2. II. Feladat

2.1. Linearizált állapotegyenlet meghatározása



7. ábra. Rakodó daru

$$\begin{bmatrix} m+M & mL_0 & 0 \\ m & mL_0 & 0 \\ 0 & 0 & m+\frac{J}{\rho^2} \end{bmatrix} \cdot \begin{pmatrix} \ddot{r} \\ \ddot{\vartheta} \\ \ddot{l} \end{pmatrix} = \begin{pmatrix} f \\ -mg\vartheta \\ -\frac{\tau}{\rho} \end{pmatrix}$$

A legmagasabb deriváltakat kifejezve a baloldali mátrixot invertálva és átrendezve a jobboldalra az egyes deriváltak a következő értékeket kapják:

$$\ddot{r} = \frac{f+mg\vartheta}{M}; \qquad \ddot{l} = -\frac{\tau}{m\rho+\frac{J}{\rho}}; \qquad \ddot{\vartheta} = \frac{-Mg\vartheta - f - mg\vartheta}{ML_0}$$

Így az állapotváltozók (x), gerjesztés (u) és a kimenet (y) a következőképpen választható:

$$x = \begin{pmatrix} r \\ \vartheta \\ \dot{r} \\ \dot{\vartheta} \\ l \\ \dot{l} \end{pmatrix}; \qquad u = \begin{pmatrix} f \\ \tau \end{pmatrix}; \qquad y = \begin{pmatrix} r \\ l \end{pmatrix}$$

A rendszer állapottér modellel felírva:

$$\underline{\dot{x}} = \underline{\underline{A}}\,\underline{x} + \underline{\underline{B}}\,\underline{u}$$

$$\underline{y} = \underline{\underline{C}}\,\underline{x} + \underline{\underline{D}}\,\underline{u}$$

$$\begin{pmatrix} \dot{r} \\ \dot{\vartheta} \\ \ddot{r} \\ \ddot{\vartheta} \\ \dot{l} \\ \ddot{l} \end{pmatrix} = \begin{bmatrix} 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & \frac{mg}{M} & 0 & 0 & 0 & 0 \\ 0 & -\frac{(M+m)g}{ML_0} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \cdot \begin{pmatrix} r \\ \vartheta \\ \dot{r} \\ \dot{\vartheta} \\ l \\ \dot{l} \end{pmatrix} + \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ \frac{1}{M} & 0 \\ -\frac{1}{ML_0} & 0 \\ 0 & 0 \\ 0 & -\frac{\rho}{m\rho^2+J} \end{bmatrix} \cdot \begin{pmatrix} f \\ \tau \end{pmatrix}$$

$$\underline{y} = \begin{pmatrix} r \\ l \end{pmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix} \cdot \begin{pmatrix} r \\ \vartheta \\ \dot{r} \\ \dot{\vartheta} \\ l \\ \dot{l} \end{pmatrix} + \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix} \cdot \begin{pmatrix} f \\ \tau \end{pmatrix}$$

A rendszeren jól látható, hogy két különálló részre bontható, mint ahogy arra a feladatkiírás is felhívja a figyelmünket. Ezek a rendszerek külön-külön is szabályozhatóak. A szétcsatolt rendszerek állapotterei a következők lesznek.

(r, f) rendszer állapottere:

$$\begin{pmatrix} \dot{r} \\ \dot{\vartheta} \\ \ddot{r} \\ \ddot{\vartheta} \end{pmatrix} = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & \frac{mg}{M} & 0 & 0 \\ 0 & -\frac{(M+m)g}{ML_0} & 0 & 0 \end{bmatrix} \cdot \begin{pmatrix} r \\ \vartheta \\ \dot{r} \\ \dot{\vartheta} \end{pmatrix} + \begin{bmatrix} 0 \\ 0 \\ \frac{1}{M} \\ -\frac{1}{ML_0} \end{bmatrix} \cdot f$$

$$y = r = \begin{bmatrix} 1 & 0 & 0 & 0 \end{bmatrix} \cdot \begin{pmatrix} r \\ \vartheta \\ \dot{r} \\ \dot{\vartheta} \end{pmatrix} + 0 \cdot f$$

(l, τ) rendszer állapottere:

$$\begin{pmatrix} \dot{l} \\ \ddot{l} \end{pmatrix} = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} \cdot \begin{pmatrix} l \\ \dot{l} \end{pmatrix} + \begin{bmatrix} 0 \\ -\frac{\rho}{m\rho^2+J} \end{bmatrix} \cdot \tau$$

$$y = l = \begin{bmatrix} 1 & 0 \end{bmatrix} \cdot \begin{pmatrix} l \\ \dot{l} \end{pmatrix} + 0 \cdot \tau$$

Az egyes rendszerek átviteli függvényei meghatározhatók a következő formula segítségével:

$$W(s) = \underline{\underline{C}} \left(s\underline{\underline{I}} - \underline{\underline{A}} \right)^{-1} \underline{\underline{B}} + \underline{\underline{D}}$$

A Matlab szimbolikus mátrix műveletei segítségével a meghatározott átviteli függvények a következők:

$$W_{rf}(s) = \frac{r(s)}{f(s)} = \frac{L_0 s^2 + g}{ML_0 s^4 + g(m+M)s^2} = \frac{0.5882 s^2 + 5.771}{s^4 + 32.89 s^2}$$

$$W_{l\tau}(s) = \frac{l(s)}{\tau(s)} = \frac{\rho}{(m\rho^2 + J)s^2} = \frac{-6.098}{s^2}$$

A két részre bontott $W_{rf}(s)$ és $W_{l\tau}(s)$ rendszer pólusai és zérusai a következők:
(A második rendszer, $W_{l\tau}(s)$ nem rendelkezik zérusokkal!)

$$s_{1,2}^{rf} = 0.00 \pm j5.7352$$

$$z_{1,2}^{rf} = 0.00 \pm j3.1321$$

$$s_1^{l\tau} = 0.00$$

```
%% A szakasz modelljének meghatározása
% A szakasz Wrf és Wltau átviteli függvényeit az alábbi változókban tárolja
% Wrf   - a szakasz Wrf átviteli függvénye
% Wltau - a szakasz Wltau átviteli függvény
syms m M J rho g L0 s

Asym=[0 0 1 0 0 0;
      0 0 0  1 0 0;
      0 m*g/M 0 0 0 0;
      0 -g*(m+M)/L0/M 0 0 0 0;
      0 0 0 0 0 1;
      0 0 0 0 0 0];

Bsym=[0 0 ;
      0 0;
      1/M 0;
      -1/L0/M 0;
      0 0;
      0 -1/(m*rho+J/rho)];

Csym=[1 0 0 0 0 0;
      0 0 0 0 1 0];

Dsym=[0 0;
      0 0];

I=eye(6);
Wsym=Csym*inv((s*I-Asym))*Bsym+Dsym;
Wrfsym=Wsym(1,1)
Wltausym=Wsym(2,2)
clear m M J rho g L0 s
```

2.2. Szabályozó tervezés

A szabályozó tervezését külön végeztem el a két rendszerre. A megadott paraméterek alapján a rendszer domináns pólusait kiszámítva a következő eredményt kaptam:

$$s_{1,2} = 0.00 \pm j5.7352$$

```
%% Szabalyozotervezes
load slz0ue m M J rho L0 y0;
g=9.81;

A=[0 0 1 0 0 0;
   0 0 0  1 0 0;
   0 m*g/M 0 0 0 0;
   0 -g*(m+M)/L0/M 0 0 0 0;
```

```
   0 0 0 0 0 1;
   0 0 0 0 0 0];

B=[0 0 ;
   0 0;
   1/M 0;
   -1/L0/M 0;
   0 0;
   0 -1/(m*rho+J/rho)];

C=[1 0 0 0 0 0;
   0 0 0 0 1 0];

D=[0 0;
   0 0];

A1=A(1:4,1:4);
A2=A(5:6,5:6);
C1=C(1,1:4);
C2=C(2,5:6);
D1=D(1,1);
D2=D(2,2);
B1=B(1:4,1);
B2=B(5:6,2);

[sys1.num, sys1.den]=ss2tf(A1, B1, C1, D1);
[sys2.num, sys2.den]=ss2tf(A2, B2, C2, D2);
Wrf=tf(sys1.num,sys1.den)
Wltau=tf(sys2.num, sys2.den)

pol1=roots(sys1.den)
pol2=roots(sys2.den)
zer1=roots(sys1.num)
zer2=roots(sys2.num)
```

2.2.1. Állapot-visszacsatolás

Az előírt pólusok alapján az Ackermann formula segítségével az első rendszer visszacsatoló ágának erősítése kiszámítható.

$$K = [0 \ \ldots \ 0 \ 1] \, M_c^{-1} \cdot \varphi_c$$

$$K_1 = [8.4662, \ 1.4067, \ 10.8689, \ -6.2925 \,]$$

A tervezési folyamat eleje a második rendszerre is hasonlóan történik. Az előírt pólusok alapján az Ackermann formula segítségével a második rendszer visszacsatoló ágának erősítése szintén kiszámítható. Az így kapott visszacsatoló ág erősítése a második rendszerre a következő:

$$K_2 = [-0.5731, \ -0.4292, \ ]$$

```
%Allapot visszacsatolas
Mc=ctrb(A,B);
rank(Mc)

sdom1=-w0*xi+1j*w0*sqrt(1-xi^2);
sdom2=conj(sdom1);
phic1=[sdom1 sdom2 scinf scinf];  % A zart kor gyokei
phic2=[sdom1 sdom2];

K1=acker(A1,B1,phic1)
K2=acker(A2,B2,phic2)
```

2.2.2. Alapjel korrekció

Alapjel miatti korrekció kiszámításához az alábbi mátrix egyenletet szükséges megoldani:

$$\begin{pmatrix} N_x \\ N_u \end{pmatrix} = \begin{bmatrix} A & B \\ C & 0 \end{bmatrix}^{-1} \begin{pmatrix} 0_{n\times m} \\ I_m \end{pmatrix}$$

Az így kapott N_x és N_u a következő az első rendszer esetén:

$$N_x = \begin{pmatrix} 1 \\ 0 \\ 0 \\ 0 \end{pmatrix}$$

$$N_u = 0$$

A második rendszer esetén az alapjel miatti korrekció kiszámításához ismételjük az előző eljárást. A kapott N_x és N_u a második rendszerre:

$$N_x = \begin{pmatrix} 1 \\ 0 \end{pmatrix}$$

$$N_u = 0$$

```
%Alapjel korrekcio
NxNu1=inv([A1 B1; C1 0])*[zeros(4,1);1];    % n=4, m=1
Nx1=NxNu1(1:4)
Nu1=NxNu1(5)
NxNu2=inv([A2 B2; C2 0])*[zeros(2,1);1];    % n=2, m=1
Nx2=NxNu2(1:2)
Nu2=NxNu1(3)
```

2.2.3. Állapotmegfigyelő és terhelésbecslő

A terhelésbecslőt is tartalmazó állapotmegfigyelő alapgondolata, hogy a szakasz bemenetére ható zavarást bevesszük a rendszer állapotvektorába és előírjuk, hogy ez a zavar a szakasz bemenetén konstans és $\dot{x}_d = 0$. Ekkor a bővített rendszer állapotegyenlete a következő lesz:

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

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

Az x_d alrendszer nem irányítható (a beérkező zavarást nem tudjuk befolyásolni). Ezért az előzőekben, az állapot-visszacsatolást és az alapjel figyelembevételéhez Nx , Nu értékét az eredeti (A, B, C) rendszerhez kellet megtervezni. A továbbiakban viszont az állapot-megfigyelőt a bővített ($\tilde{A}$, $\tilde{B}$, $\tilde{C}$) rendszerhez kell megtervezni.

$$\dot{\tilde{x}} = \tilde{A}\tilde{x} + \tilde{B}u$$
$$y = \tilde{C}\tilde{x}$$

Tehát a megfigyelőnket a kiterjesztett $\tilde{A}$, $\tilde{B}$ és $\tilde{C}$ mátrixokhoz tervezzük meg, mely így alkalmas lesz a zavar becslésére. A megfigyelő tagjait az alábbi megfontolások alapján választjuk meg:

$$\tilde{F} = \tilde{A} - \tilde{G}\tilde{C}$$
$$\tilde{H} = \tilde{B}$$
$$\tilde{G} = \tilde{K}^T$$

A dualitás elve alapján a megfigyelő tervezése visszavezethető egy fiktív rendszer állapot-visszacsatolásának megvalósítására.

$$\left(\tilde{A}, \tilde{C}\right) \longleftrightarrow \left(\tilde{A}^T, \tilde{C}^T\right)_{II} \xrightarrow[\tilde{M}_{c,II}]{\tilde{\varphi}_o(s)} \tilde{K}_{II} \longrightarrow \tilde{G} = \tilde{K}_{II}^T$$

Az első rendszerhez eredményként a következő mátrixok adódtak:

$$\tilde{F}_1 = \begin{bmatrix} -32,7152 & 0 & 1 & 0 & 0 \\ -21,7761 & 0 & 0 & 1 & 0 \\ -395,2222 & 23,0824 & 0 & 0 & 0,5882 \\ 166,1739 & -32,8924 & 0 & 0 & -0,5882 \\ -2078,1641 & 0 & 0 & 0 & 0 \end{bmatrix}$$

$$\tilde{H}_1 = \begin{bmatrix} 0 \\ 0 \\ 0,5882 \\ -0,5882 \\ 0 \end{bmatrix}$$

$$\tilde{G}_1 = \begin{bmatrix} 32,7152 \\ 21,7761 \\ 395,2222 \\ -166,1739 \\ 2078,1641 \end{bmatrix}$$

A megfigyelő tagjait a második rendszer esetén is az előző megfontolások alapján választjuk meg. Így a kapott mátrixok a következő alakot öltik a második rendszer esetén:

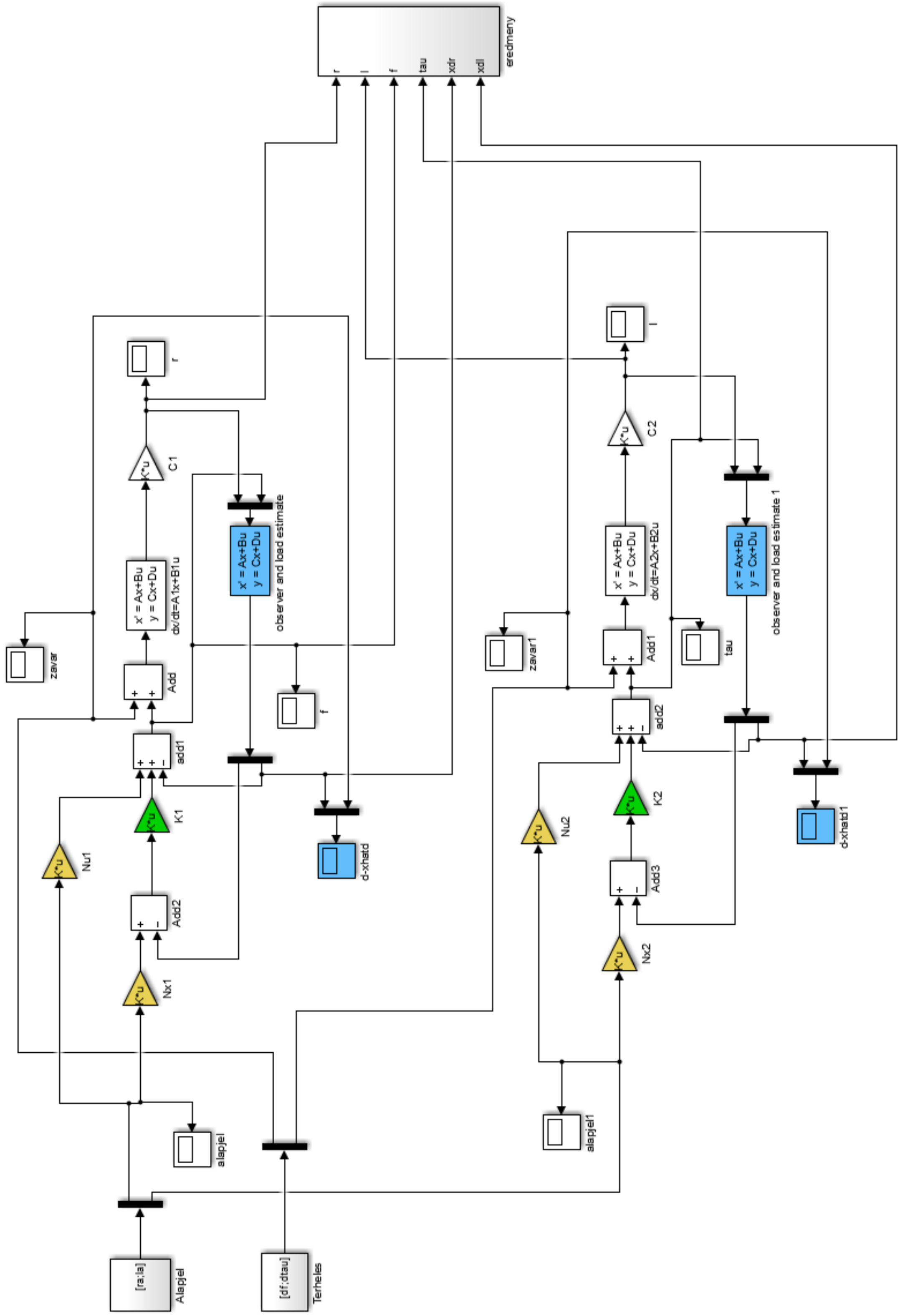
$$\tilde{F}_2 = \begin{bmatrix} -19,6291 & 1 & 0 \\ -128,4344 & 0 & -6,0976 \\ 45,9392 & 0 & 0 \end{bmatrix}$$

$$\tilde{H}_2 = \begin{bmatrix} 0 \\ -6,0976 \\ 0 \end{bmatrix}$$

$$\tilde{G}_2 = \begin{bmatrix} 19,6291 \\ 128,4344 \\ -45,9392 \end{bmatrix}$$

```
%Terheles becsles
Atilde1=[A1 B1; zeros(1,5)];
Btilde1=[B1; 0];
Ctilde1=[C1 0];
% Zajjal terhelt rendszer kar.polija
phiotilde1=[ones(1,4)*soinf soinf];
% Allapot megfigyelo a zajos rendszerhez
Gtilde1=acker(Atilde1',Ctilde1',phiotilde1)';
Ftilde1=Atilde1-Gtilde1*Ctilde1;
Htilde1=Btilde1;

Atilde2=[A2 B2; zeros(1,3)];
Btilde2=[B2; 0];
Ctilde2=[C2 0];
% Zajjal terhelt rendszer kar.polija
phiotilde2=[ones(1,2)*soinf soinf];
% Allapot megfigyelo a zajos rendszerhez
Gtilde2=acker(Atilde2',Ctilde2',phiotilde2)';
Ftilde2=Atilde2-Gtilde2*Ctilde2;
Htilde2=Btilde2;

Ftilde=blkdiag(Ftilde1, Ftilde2);
Gtilde=blkdiag(Gtilde1, Gtilde2);
Htilde=blkdiag(Htilde1, Htilde2);
Nu=blkdiag(Nu1, Nu2);
Nx=blkdiag(Nx1, Nx2);
K=blkdiag(K1, K2);
```

2.3. Szimulációs blokkvázlat



8. ábra. Simulink modell

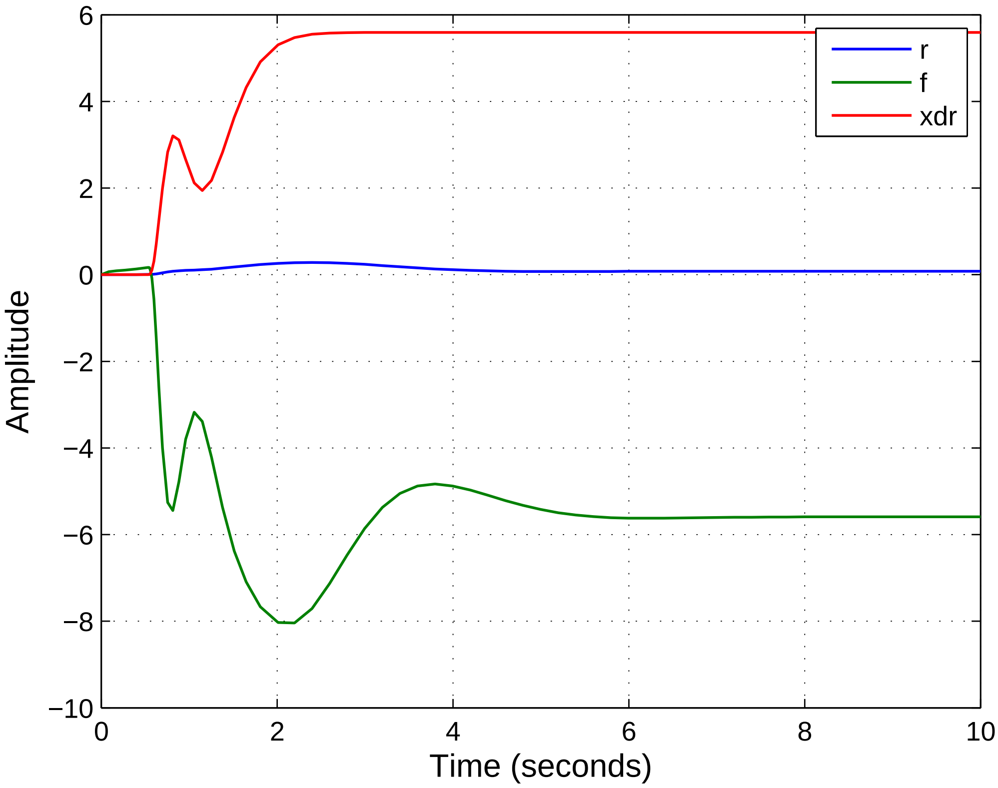
2.4. Tranziens viselkedésének vizsgálata

Ahhoz hogy a zavarás hatása megfelelőképpen látszódjon a rendszeren, a zavarás időpontját a kezdeti tranziensektől távolabb választottam meg, valamint a zavar normalizálást kisebbre állítottam.

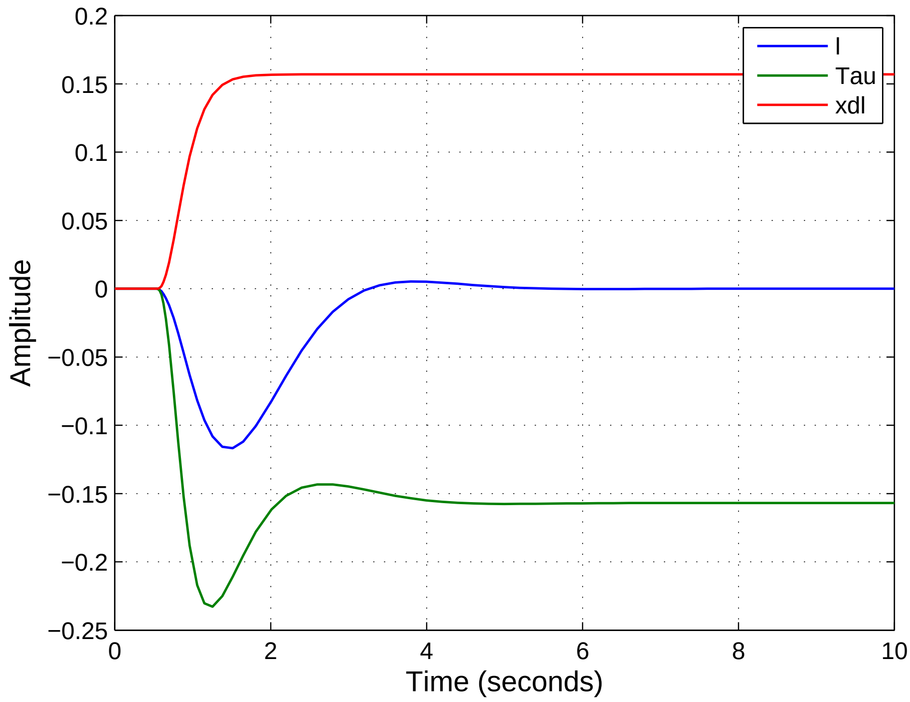
$$T_{d,start} = 4$$

$$d_{norm} = 0.01$$

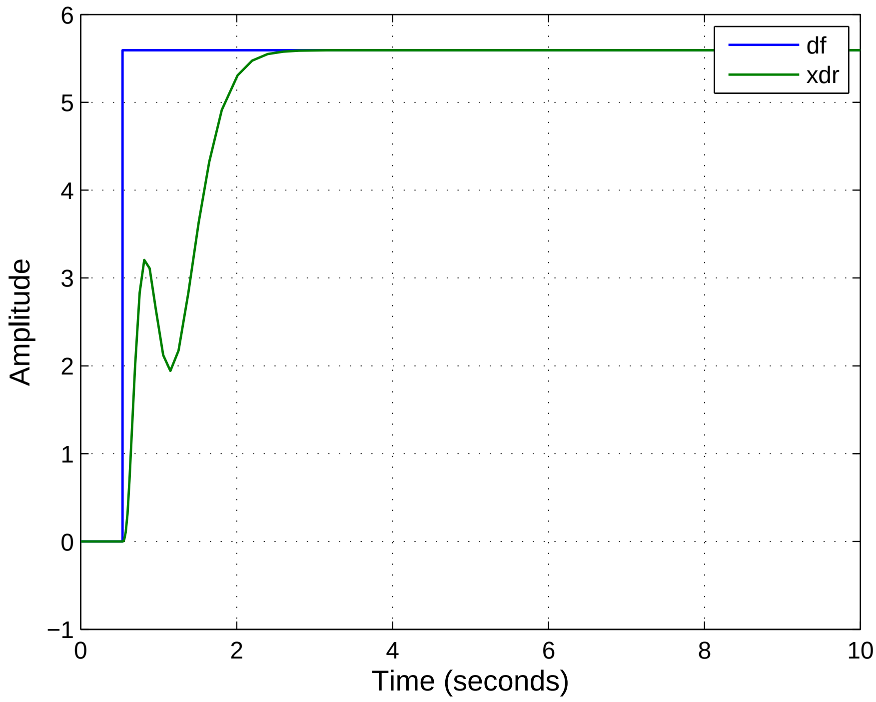
Az eredeti és a módosított adatokkal kapott tranziens viselkedések a következő négy ábrán látható az 1-es és 2-es rendszerre:



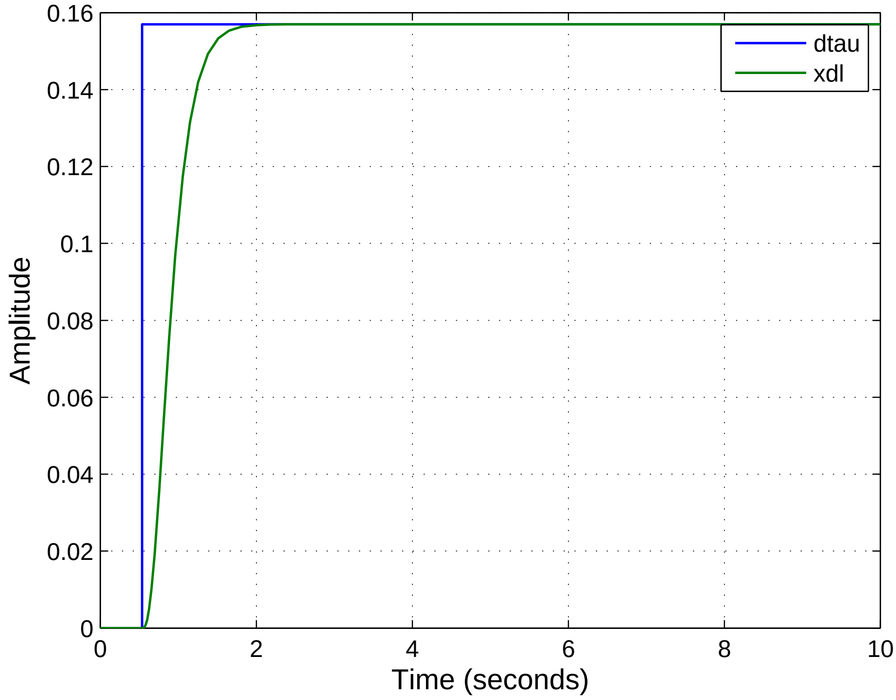
9. ábra. 1. rendszer tranziens viselkedése, $T_{d,start} = 2/a$, $d_{norm} = 1$



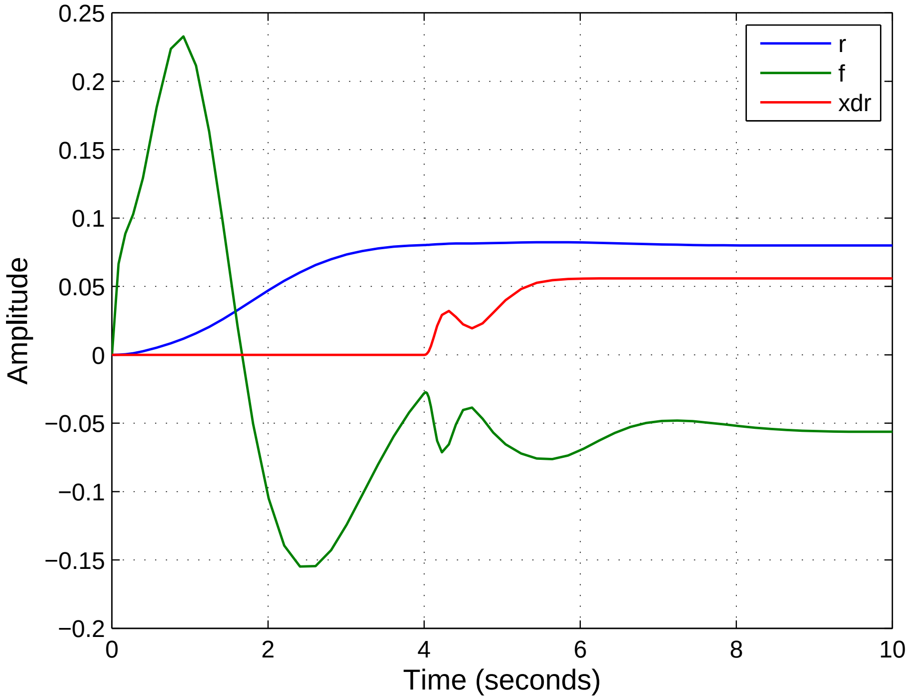
10. ábra. 2. rendszer tranziens viselkedése, $T_{d,start} = 2/a$, $d_{norm} = 1$



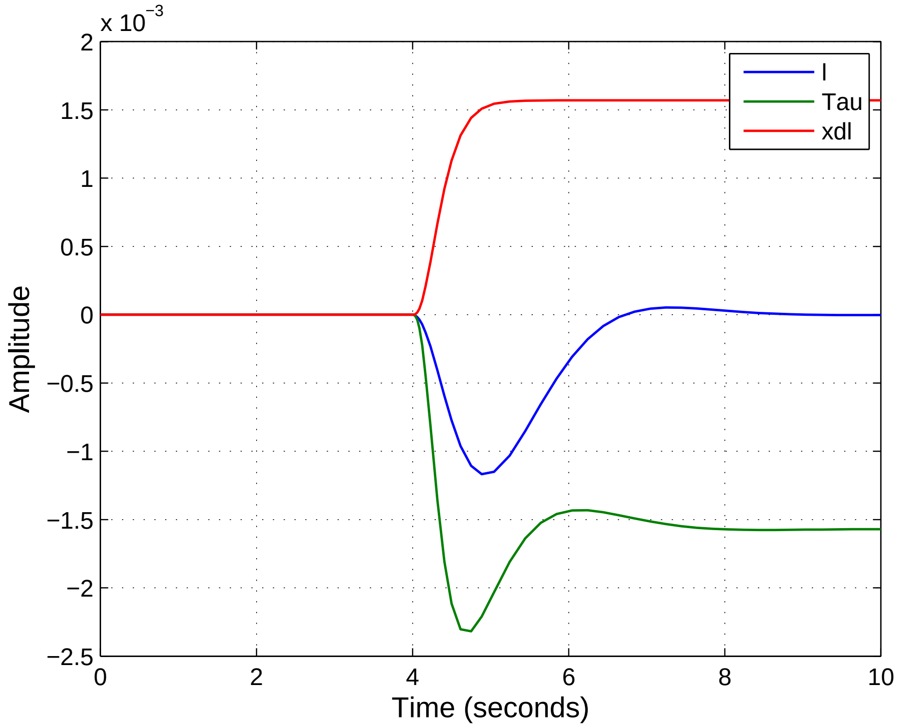
11. ábra. 1. rendszer, df, xdr, $T_{d,start} = 2/a$, $d_{norm} = 1$



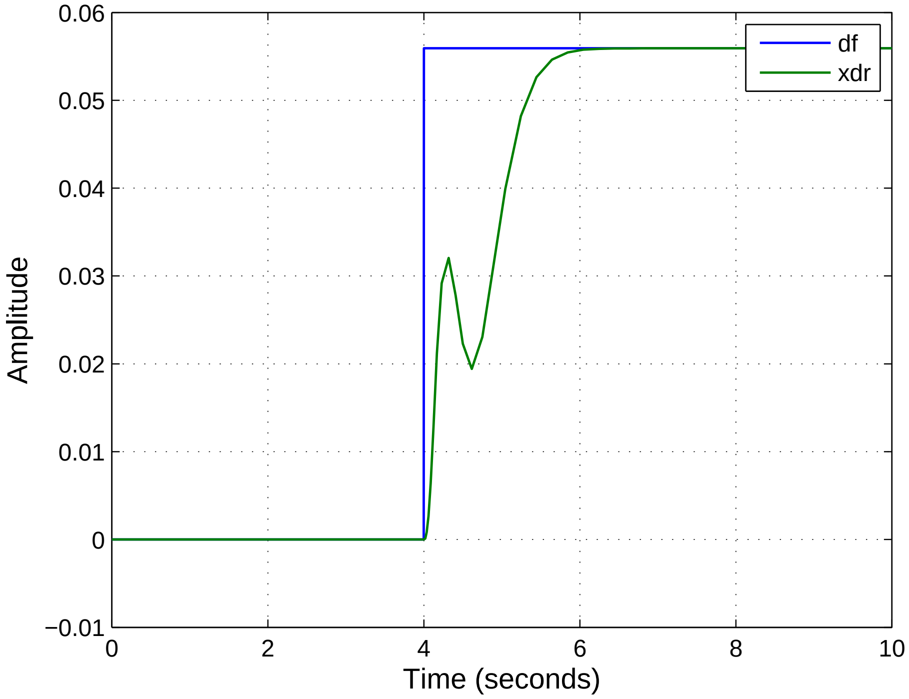
12. ábra. 2. rendszer, dtau, xdl, $T_{d,start} = 2/a$, $d_{norm} = 1$



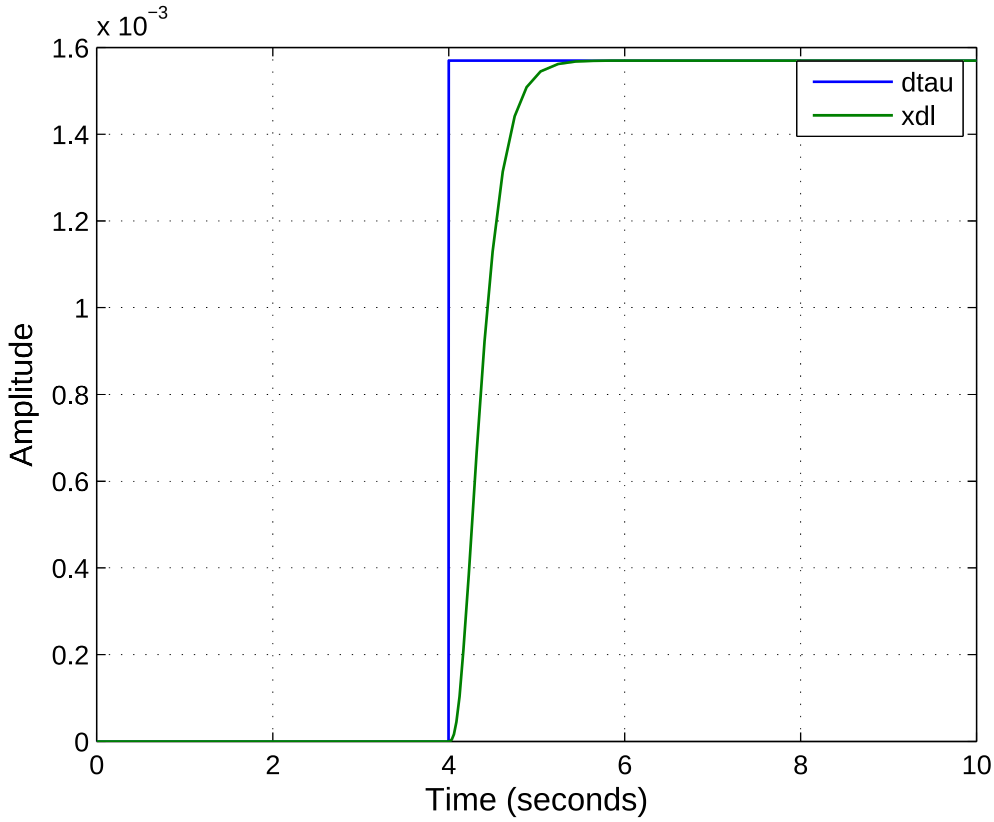
13. ábra. 1. rendszer tranziens viselkedése, $T_{d,start} = 4$, $d_{norm} = 0.01$



14. ábra. 2. rendszer tranziens viselkedése, $T_{d,start} = 4$, $d_{norm} = 0.01$



15. ábra. 1. rendszer, df, xdr, $T_{d,start} = 4$, $d_{norm} = 0.01$



16. ábra. 2. rendszer, dtau, xdl, $T_{d,start} = 4$, $d_{norm} = 0.01$

3. Melléklet

1. feladat teljes Matlab kódja

```
%% Mérési adatok betöltése
% A következő parancssorban a <neptun> helyére írja be saját Neptun-kódját,
% pl. load abc123 meres_u meres_y;
% A parancs a <neptun>.mat fájlból betölti a mérési regisztrátumot,
% a mérés során használt bemenő jelet a meres_u, a mért kimenetet
% a meres_y változóba tölti
close all
clear all
load slz0ue meres_u meres_y;
% Ts az a mintavételi idő, amit az identifikáció és a
% szabályozó megtervezése során kell használnia
Ts=0.1;

%% Identifikáció
% Az identifikáció végeredményét a következő változókban tárolja:
% Ap - a szakasz átviteli függvényének Ap paramétere
% xi - a szakasz átviteli függvényének xi paramétere
% Tau - a szakasz átviteli függvényének Tau paramétere
% alpha - a szakasz átviteli függvényének alpha paramétere
z=iddata(meres_y,meres_u,Ts);
nA=3;    % nevezo fokszama
nk=1;    % holtido fokszama
nB=nA-1+nk;
nC=nA;
tharmax=armax(z,[nA nB nC nk]); % th struktura
[A,B,C]=th2poly(tharmax);
Darmax=tf(B,A,Ts);
t=(0:Ts:length(meres_u)*Ts-Ts);
yid=lsim(Darmax,meres_u,t);

figure(1);
plot(t,meres_y,'g',t,yid,'r');
legend('Mért válaszjel','Az identifikált rendszer válasza');
xlabel('Time (seconds)');
ylabel('y(t), yid(t)');

zpoles=roots(A);          % Diszkret ideju polusok szamitasa
spoles=log(zpoles)/Ts;    % A folytonos ideju polusok szamitasa
%Tcs=-1./spoles;          % Az időállandok
spoly=poly(spoles);
D0=polyval(B,1)/polyval(A,1); % A diszkret ideju atvitel allandosult allapotban
Ap=D0                         % Allandosult allapotban az erosites
Warmax=tf(Ap*spoly(end),spoly); % Folytonos atvit
zpkW=zpk(Warmax);               % Folytonos atvit zpkban
Tau=sqrt(1/(zpkW.p{1}(3)* zpkW.p{1}(2)))
alpha=(1/(-1*zpkW.p{1}(1)))/Tau
%Ap=Warmax.num{1}(4)/Warmax.den{1}(4)
omega_0=1/Tau;
%Sigma_e=-((zpkW.p{1}(3)+ zpkW.p{1}(2))/2)
```

```
Sigma_e=-max(real(zpkW.p{1}));
xi=Sigma_e/omega_0
%xi=-(((zpkW.p{1}(3)+ zpkW.p{1}(2))/2)/(zpkW.p{1}(3)* zpkW.p{1}(2)))/Tau


%% Szabályozótervezés
% A szabályozó polinomjait a következő változókban tárolja:
% R - a szabályozó R(z) polinomja
% S - a szabályozó S(z) polinomja
% T - a szabályozó T(z) polinomja
xi_spec=0.7;
w0=1/Tau;
sdom1=-w0*xi_spec+1j*w0*sqrt(1-xi_spec^2);
sdom2=conj(sdom1);
scinf=-3*abs(real(sdom1));
soinf=-5*abs(real(sdom1));
lint=1;                       % Integratorok szama
DWarmax=c2d(Warmax, Ts, 'zoh');
zdom1=exp(sdom1*Ts);
zdom2=exp(sdom2*Ts);
zcinf=exp(scinf*Ts);
zoinf=exp(soinf*Ts);

figure(2);
pzmap(DWarmax);

B=DWarmax.num{1};            % B kinyerese
B=B(2:end);                  % Vezeto nullak elhagyas
Broots=roots(B);
Bminus=B;                    % B faktorizacio, Bminus a kiejtheto zerusok
Bplus=1;                     % B faktorizacio, Bplus a nem kiejtheto zerusok
grB=length(B)-1;             % B fokszama, az egyutthatoi szamanal 1-el kisebb

A=DWarmax.den{1};            % A kinyerese
grA=length(A)-1;             % A fokszama, az egyutthatoi szamanal 1-el kisebb
grBminus=length(Bminus)-1;   % Bminus fokszama, az egyutthatoi szamanal 1-el kisebb

%Fokszam feltetelek
grAm=1+grBminus;             % Mivel grBminus>1, ezert nincs plusz tag
grS=grA+lint-1;
grA0=grA+lint-1;             % Nincs plusz tag
grR1prime=grBminus;

%Am es Ao polinomok kiszamitasa
Am=poly([zdom1 zdom2 zcinf]); % Am gyokei a dominans poluspar es zcinf
A0=poly(ones(1,grA0)*zoinf);  % Ao gyokei zoinf, grAo-szorosa

% Az egyenletrendszer dioA*x=dioC alaku
polyint=[1 -1];                 % (1-z)^lint
Atilde=conv(polyint,A);
dioA=[ toeplitz([Atilde 0],[Atilde(1) 0]) toeplitz([Bminus 0 0 0],[Bminus(1) 0 0 0]) ];
Ctilde=conv(Am,A0)-[Atilde 0 0];
dioC=Ctilde(2:end)';
sol=inv(dioA)*dioC;             % Az egyenletrendszer megoldasa
```

```
% R1prime monic, igy polinomjanak legmagasabb fokszamu egyutthatoja 1
R1prime=[1 sol(1:2)'];
% csak utana kovetkeznek a kiszamitott egyutthatok
R=conv(R1prime,polyint); % R=R1prime*(z-1), mivel lint=1 és Bplus=1
% S polinom egyutthatoi a megoldasvektor tovabbi elemei
S=sol(3:end);
S=S';

% A T polinom számítása
Bmprime=polyval(Am,1)/polyval(Bminus,1);
T=Bmprime*A0;

% A szabalyozo tagok atviteli fuggvenyei
d_ff=tf(T,R,Ts);                   % Elorecsatolo ag
d_fb=tf(S,R,Ts);                   % Visszacsatol ag
dyr=d_ff*feedback(DWarmax,d_fb,-1); % A zart kor r->y atviteli fuggvenye

figure(3);
step(dyr);
title('A zárt kör ugrásválasza');
xlabel('Time');
ylabel('y(t)');

% A szakasz minden parameteret 75%-ra csokkentjuk
Ap75=Ap*0.75;
xi75=xi*0.75;
Tau75=Tau*0.75;
alpha75=alpha*0.75;
DEN75=conv([Tau75^2 2*xi75*Tau75 1],[alpha75*Tau75 1]);
wp75=tf(Ap75,DEN75);
% A szakasz minden parameteret 125%-ra noveljuk
Ap125=Ap*1.25;
xi125=xi*1.25;
Tau125=Tau*1.25;
alpha125=alpha*1.25;
DEN125=conv([Tau125^2 2*xi125*Tau125 1],[alpha125*Tau125 1]);
wp125=tf(Ap125,DEN125);
% Atteres diszkret idore
DWarmax75=c2d(wp75,Ts,'zoh');
DWarmax125=c2d(wp125,Ts,'zoh');
% Robusztussag vizsgalata
dur=d_ff*feedback(tf(1,1,Ts),DWarmax*d_fb,-1);
dur75=d_ff*feedback(tf(1,1,Ts),DWarmax75*d_fb,-1);
dur125=d_ff*feedback(tf(1,1,Ts),DWarmax125*d_fb,-1);

figure(4);
step(dur, dur75, dur125);
legend('100%', '75%', '125%');
title('A beavatkozó jel zárt körben');
xlabel('Time');
ylabel('u(t)');

% Zart kor osszeallitasa
dyr75=d_ff*feedback(DWarmax75,d_fb,-1);
dyr125=d_ff*feedback(DWarmax125,d_fb,-1);
```

```
figure(5);
step(dyr,dyr75,dyr125);
legend('100%', '75%', '125%');
title('A 2DOF szabályzó robusztusságának illusztrációja');
```

2. feladat teljes Matlab kódja

```
%% Paraméterek betöltése
% A következő parancssorban a <neptun> helyére írja be saját Neptun-kódját,
% pl. load abc123 meres_u meres_y;
% A parancs a <neptun>.mat fájlból a következő változókba tölti a megfelelő
% paramétereket:
% m   - a mozgatott teher tömege
% M   - a kocsi tömege
% J   - a sodronyt tekercselő motor és a dob tehetetlenségi nyomatéka
% rho - a dob sugara
% L0  - a sodrony hossza a munkapontban
% y0  - a kocsi pozíciója a munkapontban
close all
clear all
load slz0ue m M J rho L0 y0;
g=9.81;
%% Tervezési előírások
% A feladatkiírás szerinti tervezési előírások az alábbiak, a kiírásnak
% megfelelően nagy beavatkozó jelek esetén ezeket módosíthatja.
% Figyelem! A módosításokat itt hajtsa végre, a változóneveket ne cserélje
% le!
% Ügyeljen a dnorm normalizáló tényező és a Tdstart paraméter helyes
% megválasztására!
a=sqrt((m+M)*g/(m*L0));
w0=a/2;
xi=0.7;
scinf=-a;
soinf=-2*a;
Ta=1/w0;
Tdstart=2/a;
%Tdstart=4;
dnorm=1;
%dnorm=0.01;

%% A szakasz modelljének meghatározása
% A szakasz Wrf és Wltau átviteli függvényeit az alábbi változókban tárolja
% Wrf   - a szakasz Wrf átviteli függvénye
% Wltau - a szakasz Wltau átviteli függvény
syms m M J rho g L0 s

Asym=[0 0 1 0 0 0;
      0 0 0  1 0 0;
      0 m*g/M 0 0 0 0;
      0 -g*(m+M)/L0/M 0 0 0 0;
      0 0 0 0 0 1;
      0 0 0 0 0 0];
```

```
Bsym=[0 0 ;
      0 0;
      1/M 0;
      -1/L0/M 0;
      0 0;
      0 -1/(m*rho+J/rho)];

Csym=[1 0 0 0 0 0;
      0 0 0 0 1 0];

Dsym=[0 0;
      0 0];

I=eye(6);
Wsym=Csym*inv((s*I-Asym))*Bsym+Dsym;
Wrfsym=Wsym(1,1)
Wltausym=Wsym(2,2)
clear m M J rho g L0 s


%% Szabalyozotervezes
load slz0ue m M J rho L0 y0;
g=9.81;

A=[0 0 1 0 0 0;
   0 0 0  1 0 0;
   0 m*g/M 0 0 0 0;
   0 -g*(m+M)/L0/M 0 0 0 0;
   0 0 0 0 0 1;
   0 0 0 0 0 0];

B=[0 0 ;
   0 0;
   1/M 0;
   -1/L0/M 0;
   0 0;
   0 -1/(m*rho+J/rho)];

C=[1 0 0 0 0 0;
   0 0 0 0 1 0];

D=[0 0;
   0 0];

A1=A(1:4,1:4);
A2=A(5:6,5:6);
C1=C(1,1:4);
C2=C(2,5:6);
D1=D(1,1);
D2=D(2,2);
B1=B(1:4,1);
B2=B(5:6,2);

[sys1.num, sys1.den]=ss2tf(A1, B1, C1, D1);
[sys2.num, sys2.den]=ss2tf(A2, B2, C2, D2);
```

```
Wrf=tf(sys1.num,sys1.den)
Wltau=tf(sys2.num, sys2.den)

pol1=roots(sys1.den)
pol2=roots(sys2.den)
zer1=roots(sys1.num)
zer2=roots(sys2.num)

%Allapot visszacsatolas
Mc=ctrb(A,B);
rank(Mc)

sdom1=-w0*xi+1j*w0*sqrt(1-xi^2);
sdom2=conj(sdom1);
phic1=[sdom1 sdom2 scinf scinf];  % A zart kor gyokei
phic2=[sdom1 sdom2];

K1=acker(A1,B1,phic1)
K2=acker(A2,B2,phic2)

%Alapjel korrekcio
NxNu1=inv([A1 B1; C1 0])*[zeros(4,1);1];  % n=4, m=1
Nx1=NxNu1(1:4)
Nu1=NxNu1(5)
NxNu2=inv([A2 B2; C2 0])*[zeros(2,1);1];  % n=2, m=1
Nx2=NxNu2(1:2)
Nu2=NxNu1(3)

%Terheles becsles
Atilde1=[A1 B1; zeros(1,5)];
Btilde1=[B1; 0];
Ctilde1=[C1 0];
% Zajjal terhelt rendszer kar.polija
phiotilde1=[ones(1,4)*soinf soinf];
% Allapot megfigyelo a zajos rendszerhez
Gtilde1=acker(Atilde1',Ctilde1',phiotilde1)';
Ftilde1=Atilde1-Gtilde1*Ctilde1;
Htilde1=Btilde1;

Atilde2=[A2 B2; zeros(1,3)];
Btilde2=[B2; 0];
Ctilde2=[C2 0];
% Zajjal terhelt rendszer kar.polija
phiotilde2=[ones(1,2)*soinf soinf];
% Allapot megfigyelo a zajos rendszerhez
Gtilde2=acker(Atilde2',Ctilde2',phiotilde2)';
Ftilde2=Atilde2-Gtilde2*Ctilde2;
Htilde2=Btilde2;

Ftilde=blkdiag(Ftilde1, Ftilde2);
Gtilde=blkdiag(Gtilde1, Gtilde2);
Htilde=blkdiag(Htilde1, Htilde2);
Nu=blkdiag(Nu1, Nu2);
Nx=blkdiag(Nx1, Nx2);
K=blkdiag(K1, K2);
```

```
%% Szimuláció
% A szimulációt a mellékelt sim.mdl modell alapján elkészített
% <neptun>sim.mdl néven (pl. abc123sim.mdl) mentett Simulink–modellel
% végezze! A modellben található alapjel– és terhelés–generáló blokkok a
% fenti paramétereket (y0,Tdstart,Ta,m,M,g,rho) használják, így a szimuláció
% elϕtt mindig futtassa le ezt a scriptet!
```