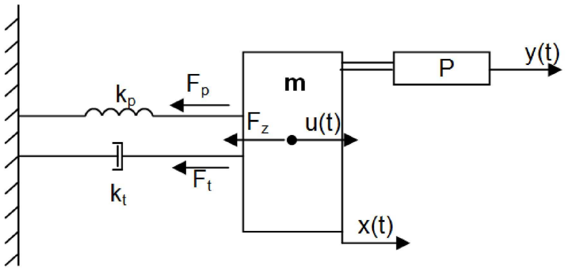
7 Modelovanie a analýza lineárnych dynamických systémov s využitím funkcií Control System Toolbox-u

- Aplikačná knižnica **Control System Toolbox** je zameraná na riešenie úloh analýzy a syntézy lineárnych časovo-invariantných dynamických systémov (Linear-Time-Invariant - LTI)
- Základným predpokladom použitia jednotlivých modulov (funkcií) je znalosť matematických modelov riadených systémov opísaných v stavovom priestore (v časovej oblasti); pomocou prenosových funkcií (v s- oblasti)
- Riešenie úloh pomocou funkcií **Control System Toolbox**-u môžeme riešiť úlohy tvorby modelov rôznymi spôsobmi zápisu a konverzie medzi matematickými modelmi pre systémy SISO v prostredí MATLAB
- **Control System Toolbox** využíva modely v tvare prenosových funkcií alebo systémov diferenciálnych rovníc v stavovom priestore
- Obidva typy modelov môžu byť vyjadrené v spojitom tvare (continuous time) a v diskrétnom tvare (discrete time)

7.1 Spôsoby zadefinovania lineárnych dynamických systémov s využitím funkcií Control Toolboxu

Rôzne spôsoby modelovania LDS budeme ilustrovať na príklade modelu DS – „pružina- tlmič“



$$F_z(t) = m * \frac{dx^2(t)}{dt^2}$$

$$F_t(t) = k_t * \frac{dx(t)}{dt}$$

$$F_p(t) = k_p * x(t)$$

$$y(t) = p * x(t)$$

$$\sum_{i=1}^{n} F_i = 0$$

Na zostavenie matematického modelu dynamického systému využijeme d´Alembertov princíp, výsledkom ktorého je lineárna DR 2.rádu.

$$m.\frac{dx^2(t)}{dt^2} + k_t.\frac{dx(t)}{dt} + k_p.x(t) = u(t)$$

$$m.x^{''}(t) + k_t.x^{'}(t) + k_p.x(t) = u(t)$$

Matematický model LDS prepíšeme do Laplaceovej transformácie

$$(ms^2 + k_t s + k_p)X(s) = U(s)$$

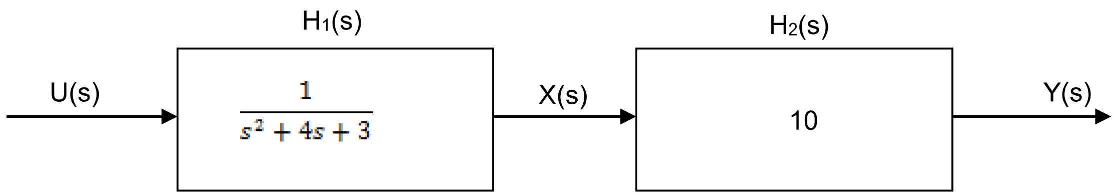
Vytvorenie prenosovej funkcie *$H_1(s)$* :

$$H_1(s) = \frac{X(s)}{U(s)} = \frac{1}{ms^2 + k_t s + k_p}$$

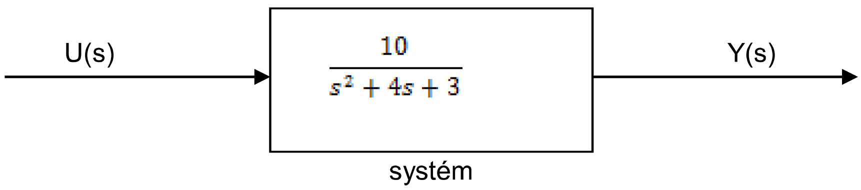
Bloková schéma poukazuje na súvislosť vstupných a výstupných premenných pomocou prenosovej funkcie *$H_1(s)$* a *$H_2(s)$* .

Predpokladajme, že v modeli „pružina-tlmič“ sú zvolené parametre nasledovne :

$m = 1;\ k_t = 3;\ k_p = 4;\ p = 10$



H = $H_1(s).H_2(s)$



Rôzne spôsoby zápisu prenosovej funkcie *H(s):*

a) $H(s) = \frac{b_m s^m + b_{m-1} s^{m-1} + \dots + b_1 s + b_0}{a_n s^n + a_{n-1} s^{n-1} + \dots + a_1 s + a_0}$, $m < n$ - racionálne lomená funkcia

b) $H(s) = K * \frac{(s-z_1)(s-z_2)\dots(s-z_m)}{(s-p_1)(s-p_2)\dots(s-p_n)}$, - póly/nuly (z/p/k)

c) $H(s) = \frac{r_1}{s-p_1} + \frac{r_2}{s-p_2} + \dots + \frac{r_n}{s-p_n} + K$ - reziduálna forma

1) Zadávanie prenosovej funkcie modelu *H(s)* v tvare racionálne / lomenej funkcie (RLF)

- $\boldsymbol{sys = tf(num, den)}$

V simulačnom jazyku MATLAB sa pre vytvorenie prenosovej funkcie v tvare RLF požíva funkciu ***tf***

$$H(s) = \frac{B(s)}{A(s)}$$

kde,

B(s) – num(čitateľ prenosovej funkcie) , *A(s)* – den(menovateľ prenosovej funkcie)

<u>PRÍKLAD 1</u>

```
>> num=[1,2];
>> denum=[1,2,1];
>> sys=tf(num,denum)
```

<u>Výsledok :</u>

Transfer function:

```
s + 2
-------------
s^2 + 2 s + 1
```

2) Zadávanie prenosovej funkcie modelu *H(s)* v tvare *nuly/póly/zosilnenie (z/p/k)*

- $\boldsymbol{sys = zpk(z, p, k)}$

  kde

  z – nuly

  p – póly

  $k = \frac{b_m}{a_n}$ - *zosilnenie*

PRÍKLAD 2

```
>> z=[-2];
>> p=[-1,-1];
>> k=[1];
>> sys=zpk(z,p,k)
```

Výsledok:

```
Zero/pole/gain:
(s+2)
-------
(s+1)^2
```

3) Zadávanie prenosovej funkcie modelu *H(s)* v reziduálnom tvare

- $\boldsymbol{[r, p, k] = residue(num, denum)}$

PRÍKLAD 3

Rozložte na parciálne zlomky obrazový prenos $H(s) = \frac{s+2}{s^2+2s+1}$

```
>> num=[1,2];
>> denum=[1,2,1];
>> [r,p,k]=residue(num,denum)


r =
     1
     1
p =
    -1
    -1
k =
     []
```

$$H(s) = \frac{B(s)}{A(s)} = \frac{1}{s+1} + \frac{1}{s+1}$$

Reprezentácia LDS v stavovom priestore (STATE SPACE MODELS**)** predstavuje významný prístup k modelovaniu tohto druhu systémov.

Majme diferenciálnu rovnicu, ktorá vyjadruje model systému

„pružina – tlmič“ *:* $mx''(t) + k_p x'(t) + k_t x(t) = u(t)$

Vytvorte substitučný kanonický tvar v stavovom priestore, pričom uvažujeme substitúciu : $x(t) = x_1(t)$

$$x_1'(t) = x_2(t)$$

$$x_2'(t) = -\frac{k_t}{m}x_1(t) - \frac{k_p}{m}x_2(t) + \frac{1}{m}u(t)$$

Rovnica pre meraný výstup má tvar

$$y = g(x,u) = 10x_1(t)$$

Konkrétny tvar LDS v stavom priestore pri zadaných parametroch:

$$x_1'(t) = x_2(t)$$
$$x_2'(t) = -3x_1 - 4x_2 + u(t)$$

- Systém DR v substitučnom kanonickom tvare prepíšeme do stavového priestoru, ktorý má tvar*:*

$$x'(t) = \boldsymbol{A}x(t) + \boldsymbol{B}u(t)$$
$$y(t) = \boldsymbol{C}x(t) + \boldsymbol{D}u(t)$$

- Prepis systému pružina-tlmič do stavového popisu

$$\begin{bmatrix} x_1'(t) \\ x_2'(t) \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ -3 & -4 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} 0 \\ 1 \end{bmatrix} u$$

$$y(t) = \begin{bmatrix} 10 & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + [0]u$$

4) Zadávanie dynamického systému pomocou stavového opisu a matíc *A, B, C, D*

Ak je stavový popis DS definovaný *:*

$$x'(t) = \boldsymbol{A}x(t) + \boldsymbol{B}u(t)$$
$$y(t) = \boldsymbol{C}x(t) + \boldsymbol{D}u(t)'$$

potom na zápis systému do stavového priestoru v simulačnom jazyku MATLAB používame funkciu ***„ss“***

- $\boldsymbol{sys} = \boldsymbol{ss}(\boldsymbol{A}, \boldsymbol{B}, \boldsymbol{C}, \boldsymbol{D})$

PRÍKLAD 4

Majme zadané matice systému v stavovom priestore : $\boldsymbol{A} = \begin{bmatrix} -2 & -1 \\ 1 & 0 \end{bmatrix}$, $\boldsymbol{B} = \begin{bmatrix} 1 \\ 0 \end{bmatrix}$, $\boldsymbol{C} = \begin{bmatrix} 1 & 2 \end{bmatrix}$, $\boldsymbol{D} = 0$. Vytvorte v programovom prostredí MATLAB systém v stavovom popise.

```
>> A=[-2 -1;1 0];
>> B=[1;0];
>> C=[1 2];
>> D=[0];
>> sys=ss(A,B,C,D)
```

výsledok:

```
A =
        x1   x2
   x1   -2   -1
   x2    1    0
B =
        u1
   x1    1
   x2    0
C =
        x1   x2
   y1    1    2
```

5) **V ďalšom budeme ilustrovať funkcie Control System Toolbox-u, ktoré umožňujú prepis lineárnych modelov DS z časovej oblasti do s-oblastí s využitím LT*:***

Function		*Význam*
$c2d$	→	$zo\ spojitého\ stavového\ priestoru\ na\ diskrétny\ tvar$
$tf2zp$	→	$prenosová\ funkcia\ H(s) \to H(s)\ v\ tvare\ z_i/p_i$
$zp2tf$	→	$H(s) v\ tvare\ z_i/p_i \to prenosová\ funkcia\ H(s)$
$tf2ss$	→	$prenosová\ funkcia\ H(s) \to stavový\ priestor$
$ss2tf$	→	$stavový\ priestor\ \to prenosová\ funkcia\ H(s)$
$zp2ss$	→	$H(s)\ v\ tvare\ z_i/p_i \to stavový\ priestor$
$ss2zp$	→	$stavový\ priestor\ \to H(s)\ v\ tvare\ z_i/p_i$
$tfdata$	→	$obrazový\ prenos - identifikácia\ dát$
$pzmap$	→	$vypísanie\ pólov\ a\ núl\ v$

Popis funkcií s príkladmi:

- ***c2d function*** **-** konvertuje spojitý popis modelu v stavovom priestore na diskrétny stavový popis modelu

$$[A_d B_d] = c2d(A, B, T_s)$$

kde:

A,B – matice DS v spojitom stavovom popise
T_S – perióda vzorkovania modelu
A_d, B_d – matice diskretného popisu modelu

PRÍKLAD 5

Majme zadaný spojitý systém v stavovom priestore

$$\mathbf{A} = \begin{bmatrix} 0 & 1 \\ -3 & -4 \end{bmatrix} \text{ a } \mathbf{B} = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$$

Vypočítajte matice diskrétného modelu systému ***A_d*, *B_d*** v stavovom priestore pomocou funkcie ***c2d*** systém s periódou vzorkovania 0,1.

```
>> A=[0,1;-3,-4];
>> B=[0; 1];
>> [Ad,Bd]=c2d(A,B,0.1)
```

Hodnoty matíc vypočítame c2d

```
Ad =
    0.9868    0.0820
   -0.2460    0.6588

Bd =
    0.0044
   0.0820
```

Stavový popis diskretného modelu má tvar

$$\begin{bmatrix} x_1(k+1) \\ x_2(k+1) \end{bmatrix} = \begin{bmatrix} 0{,}9868 & 0{,}0820 \\ -0{,}2464 & 0{,}6568 \end{bmatrix} \begin{bmatrix} x_1(k) \\ x_2(k) \end{bmatrix} + \begin{bmatrix} 0{,}0044 \\ 0{,}0820 \end{bmatrix} u(k)$$

- ***tf2zp function*** **-** funkcia konvertuje polynomiálny tvar ***H(s)*** na tvar poly/nuly

$$[z, p, k] = tf2zp(num, den)$$

kde :

num/den – koeficienty čitateľa/menovateľa prenosovej funkcie *H(s)*
z – vektor núl
p – vektor polov
k - zosilnenie

PRÍKLAD 6

Majme zadaný systém v tvare obrazového prenosu *:* $H(s) = \frac{Y(s)}{U(s)} = \frac{10}{s^2+4s+3}$.
Vytvorte systém popísaný prenosovou funkciou v tvare póly/nuly.

Riešenie v jazyku MATLAB:

```
>> num = 10;
>> den=[1 4 3];
>> [z p k] = tf2zp(num,den)
```

```
z =
   Empty matrix: 0-by-1
p =
    -3
    -1
k =
    10
```

- ***zp2tf function -*** prevod ***H(s)*** v tvare póly/nuly na polynomiálny tvar

  $$[num\ den] = zp2tf(z,p,k)$$

  kde :

  z – stĺpcový vektor núl
  p – stĺpcový vektor pólov
  k – statické zosilnenie(pre MIMO systém každý stĺpec je určený pre jeden výstup MIMO systému)
  num/den – vektor koeficientov čitateľa/menovateľa *H(s)*

PRÍKLAD 7

Majme lineárny systém zadaný v tvare prenosovej funkcie *H(s)* póly/nuly. Pričom zosilnenie k = 10, póly systému sú $p_1 = -3$, $p_2 = -1$ a systém nemá nuly.

```
>> z=[];
>> p=[-3 -1];
>> k=10;
>> [num,den]=zp2tf(z,p,k)
```

```
num =
     0     0    10
den =
     1     4     3
```

- ***tf2ss function –*** prenosová funkcia ***H(s)*** v polynomiálnom tvare sa transformuje do podoby stavového popisu linearného dynamického systému

  $$[A,B,C,D] = tf2ss(num,den)$$

  kde:

  num/den – vektory obsahujúce koeficienty pri mocninách s klesajúcim exponentom operátora *s* prenosovej funkcie *H(s)*
  A,B,C,D – matice stavového popisu LDS

PRÍKLAD 8

Majme systém zadaný prenosovou funkciou v polynomiálnom tvare*:* $H(s) = \frac{10}{s^2+4s+3}$.
Vytvorte pomocou funkcii Control System Toolbox-u systém popísaný v stavovom priestore.

Riešenie v jazyku MATLAB:

```
>> num=10;
>> den=[1 4 3];
>> [A,B,C,D]=tf2ss(num,den)
```

```
A =
    -4    -3
     1     0
B =
     1
     0
C =
     0    10
D =
     0
```

Stavový popis systému:

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = \begin{bmatrix} -4 & -3 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} 1 \\ 0 \end{bmatrix} u$$

$$y = \begin{bmatrix} 0 & 10 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + [0]u$$

- ***ss2tf function* -** konvertuje systém v tvare spojitého popisu na polynomiálny tvar prenosovej funkcie ***H(s)***

$$[\boldsymbol{num}, \boldsymbol{den}] = \boldsymbol{ss2tf}(\boldsymbol{A}, \boldsymbol{B}, \boldsymbol{C}, \boldsymbol{D}, \boldsymbol{iu})$$

kde:

A,B,C,D – matice stavového popisu LDS
Iu – pre SISO systém iu = 1
num/den –vektory koeficientov čitateľa/menovateľa prenosovej funkcie ***H(s)***

PRÍKLAD 9

Majme zadaný systém v stavovom priestore:

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ -3 & -4 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} 0 \\ 1 \end{bmatrix} u$$

$$y = \begin{bmatrix} 10 & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + [0]u \; .$$

Vytvorte systém popísaný prenosovou funkciou v polynomiálnym tvarom.

Riešenie v jazyku MATLAB:

```
>> A=[0,1;-3,-4];
>> B=[0 1]';
>> C=[10 0];
>> D=0;
>> iu=1;
>> [num,den]=ss2tf(A,B,C,D,iu)
```

```
num =
     0     0    10
den =
     1     4     3
```

- ***zp2ss function* –** konvertuje prenosovú funkciu ***H(s)*** v tvare póly/nuly do stavového priestoru

$$[\boldsymbol{A}, \boldsymbol{B}, \boldsymbol{C}, \boldsymbol{D}] = \boldsymbol{zp2ss}(\boldsymbol{z}, \boldsymbol{p}, \boldsymbol{k})$$

kde:

z – matica korešpondujúca s rozložením núl, ktorá má jeden stĺpec z MIMO systému(ak uvažujeme SISO systém ***z*** je stlpcový vektor)

p – stĺpcový vektor rozloženia pólov
k - statické zosilnenie
A, B, C, D – matice stavového priestoru

PRÍKLAD 10

Majme systém zadaný prenosovou funkciou v tvare poly/nuly: $H(s) = \frac{10}{(s+3)(s+1)}$.

Pretransformujte tento systém pomocou funkcie *Control System Toolbox*-u na systém popísaný v stavovom priestore.

```
>> z=[];
>> p=[-3,-1];
>> k=10;
>> [A,B,C,D]=zp2ss(z,p, k)
```

```
A =
  -4.0000   -1.7321
    1.7321       0
B =
     1
     0
C =
     0     5.7735
D =
     0
```

- ***ss2zp* function -** konvertuje systém v stavovom priestore na prenosovú funkciu ***H(s)*** v tvare poly/nuly

$[\boldsymbol{z}, \boldsymbol{p}, \boldsymbol{k}] = \boldsymbol{ss2zp}(\boldsymbol{A}, \boldsymbol{B}, \boldsymbol{C}, \boldsymbol{D}, \boldsymbol{iu})$

kde:

A, B, C, D – matice dynamického systému opisaného v stavovom priestore, korešpondujúce s „*iu*-tým“ vstupom pre MIMO systém, iu = 1 pre SISO systém
z – vektor núl
p – vektor pólov
k – statické zosilnenie

PRÍKLAD 11

Majme systém „pružina/tlmič“ popísaný v stavovom priestore maticami:

$$A = \begin{bmatrix} 0 & 1 \\ -3 & -4 \end{bmatrix}, \qquad B = \begin{bmatrix} 0 \\ 1 \end{bmatrix}, \qquad C = \begin{bmatrix} 10 & 0 \end{bmatrix} \qquad a \qquad D = 0.$$

S využitím funkcie Control System Toolbox-u transformujte popis tohto systému z časovej oblasti do Laplaceovej transformácie v tvare prenosovej funkcie – poly/nuly.

```
>> A=[0 1; -3 -4];
>> B=[0 1]';
>> C=[10 0];
>> D=[0];
>> iu=1;
>> [z p k]=ss2zp(A,B,C,D,iu)
```

```
z =
   Empty matrix: 0-by-1
p =
    -1
    -3
k =
    10
```

- ***tfdata* function -** zistenie hodnôt parametrov prenosovej funkcie *H(s)*

$[num, den] = tfdata(sys)$

kde:

sys – zadefinovaný systém v tvare prenosovej funkcie *H(s)*
num/den – vektory čitateľa/menovateľa prenosovej funkcie *H(s)* obsahujúce koeficienty (parametre) pri klesajúcich mocninách operátora ***s***

PRÍKLAD 12

Majme systém popísaný v tvare: $H(s) = \frac{s+2}{s^2+2s+1}$.

Pomocou funkcie Control System Toolbox-u zistite hodnoty parametrov prenosovej funkcie.

```
>> sys=tf([1 2], [1,2,1]);          num =    [1x3 double]
>> [num,den]=tfdata(sys)            den =    [1x3 double]
```

- ***pzmap* function -** vypísanie hodnôt pólov a núl v komplexnej rovine

pzmap(sys)

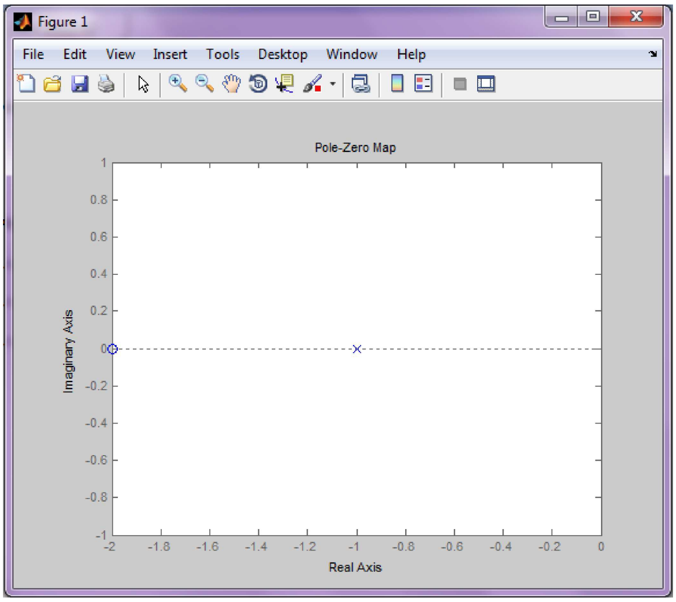
kde:

sys – definovaný LDS, ktorého poly/nuly chceme vykresliť

PRÍKLAD 13

Vykreslite póly a nuly LDS zadaného prenosovou funkciou v polynomiálnom tvare : $H(s) = \frac{s+2}{s^2+2s+1}$.

```
>> num=[1,2];
>> den=[1,2,1];
>> sys=tf(num,den);
>> pzmap(sys)
```



Obrázok 7-1 Znázornenie polov/nul prenosovej funkcie *H(s)* LDS

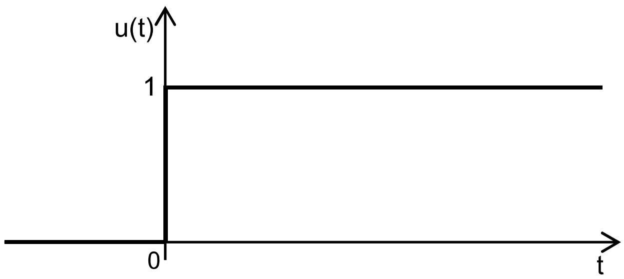
7.2 Analýza lineárnych dynamických systémov (LDS) s využitím funkcií *Control System Toolbox*-u

- **Funkcie pre analýzu lineárnych dynamických systémov v časovej oblasti**

Jednotkový skok a prechodová charakteristika

- prechodová charakteristika je odpoveď dynamického systému na jednotkový skok pri nulových počiatočných podmienkach.
- analytické vyjadrenie prechodovej charakteristiky sa nazýva prechodová funkcia
- jednotkový skok (Heavisidov skok) sa označuje ako $1(t)$ a jeho matematické vyjadrenie:

$$1(t) < \begin{matrix} 0\ ak\ t < 0 \\ 1\ ak\ t \geq 0 \end{matrix}$$



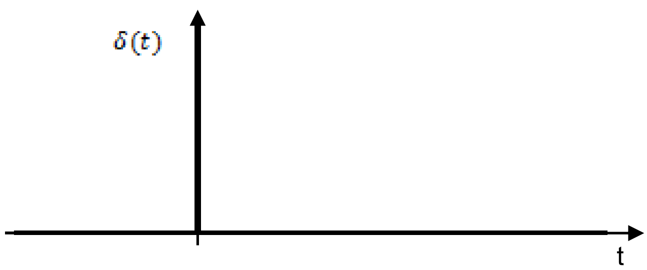
Vstupný budiaci signál u(t) = 1(t) (jednotkový skok)

- funkcia $\boldsymbol{step(system)}$ - umožňuje vypočítať odozvu LDS

Váhová funkcia a impulzná charakteristika

- Impulzná charakteristika je odozva na výstupe z LDS na Diracov impulz (jednotkový impulz).
- Diracov impulz je definovaný ako derivácia jednotkového skoku,
- váhová funkcia je analytické vyjadrenie odozvy nenabudeného sústavu na Diracov impulz,
- jednotkový impulz označujeme $\delta(t\ )$ a má nasledujúce matematické vyjadrenie:

$$u(t) = \delta(t) < \begin{matrix} 0\ ak\ t > 0, t < 0 \\ nie\ je\ definovan\acute{e}\ ak\ t = 0 \end{matrix}$$



Vstupný budiaci signál $u(t) = \delta(t)$ (Diracov impulz)

- funkcia $\boldsymbol{impulse(syst\acute{e}m)}$ - umožňuje vypočítať odozvu LDS na vstupný signál $u(t) = \delta(t)$

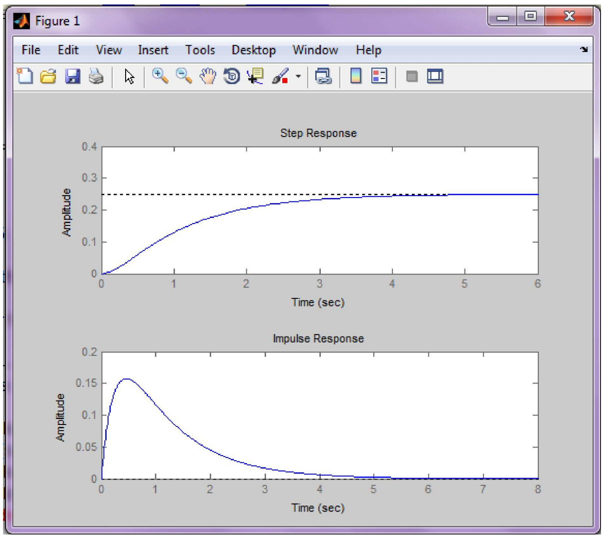
PRÍKLAD
Majme obrazový prenos LDS prenosovou funkciou $F(s) = \frac{1}{s^2+5s+4}$.

Vypočítajte a vykreslite v simulačnom jazyku MATLAB a pomocou Control System Toolbox-u prechodovú a impulznú charakteristiku systému.

```
>> num =1; % definícia čitateľa obrazového prenosu
>> den = [1 5 4]; % definícia menovateľa obrazového prenosu
>> sys=tf(num,den) % vytvorenie systému pomocou num, den

Transfer function:
  1
-------------
s^2 + 5 s + 4

>> subplot(211)
>> step(sys) % vykreslenie prechodovej charakteristiky
>> subplot(212)
>> impulse(sys) % vykreslenie impulznej časovej charakteristiky
```



Obrázok 7-2 Prechodová a impulzná charakteristika LDS ako odozva na u(t) = 1(t)/ $\delta(t)$

- **Funkcie pre analýzu lineárnych dynamických systémov vo frekvenčnej oblasti**

Nyquistova charakteristika
Nyquistova charakteristika je znázornenie komplexných hodnôt frekvenčného prenosu v Gaussovej rovine komplexných čísel, pričom sa vychádza z algebraického tvaru frekvenčného prenosu (funkcie):

$$F(i\omega) = Re[F(i\omega)] + i\,Im[F(i\omega)]$$

- funkcia $\boldsymbol{nyquist(system)}$ – umožňuje vypočítať a vykresliť Nyquistovú charakteristiku

Nicholsova charakteristika
Nicholsova charakteristika je znázornenie prirodzeného logaritmu frekvenčného prenosu v Gaussovej rovine komplexných čísel. Vychádza z exponenciálneho tvaru frekvenčného prenosu:

$$\ln F(i\omega) = \ln|F(i\omega)|e^{i\varphi(\omega)} = \ln|F(i\omega)| + i\varphi(\omega)$$

- funkcia $\boldsymbol{nyquist(system)}$ – umožňuje vypočítať a vykresliť Nicholsovu charakteristiku

<u>Bodeho charakteristika</u>
Bodeho frekvenčné charakteristiky sú parametrickým vyjadrením Nicholsovej frekvenčnej charakteristiky v závislosti od logω (amplitudová charakteristika, fázová charakteristika)

Funkcie ***bode(system)*** umožňuje vypočítať a vykresliť Bodeho charakteristiky vykresliť pomocou

<u>PRÍKLAD</u>
Majme definovaný obrazový prenos v tvare

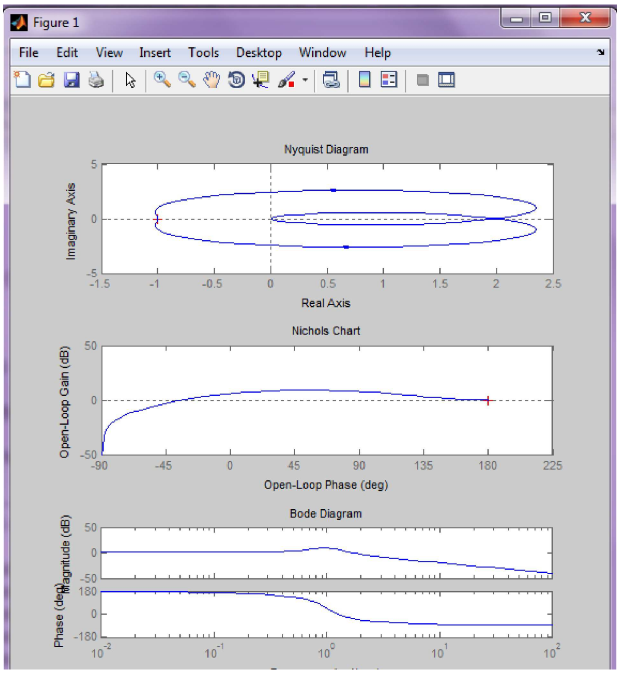
$$F(s) = \frac{s-1}{s^2+0.5s+1},$$

pomocou funkcií *Control System Toolbox*-u. Vykreslite Nyquistovu, Nicholsovu a Bodeho frekvenčné charakteristiky.

```
>> num= [1 -1];  % vytvorenie num - čitateľ obrazového prenosu
>> den = [1 0.5 1]; % vytvorenie denum - menovateľ obrazového prenosu
>> sys=tf(num,den) % vytvorenie obrazového prenosu pomocou num,den

Transfer function:
 s - 1
---------------
s^2 + 0.5 s + 1

>> subplot(311)
>> nyquist(sys) %nyquistová frekvenčná charakteristika
>> subplot(312)
>> nichols(sys) %nicholsová frekvenčná charakteristika
>> subplot(313)
>> bode(sys) % bodeho frekvenčné charakteristiky
```

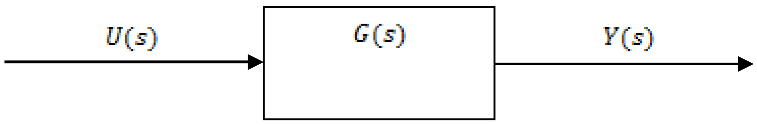


Obrázok 7-3 Frekvenčné charakteristiky LDS (Nyquist, Nichols, Bode)

7.3 Úprava blokových schém regulačných obvodov v prostredí MATLAB

Cieľom cvičenia je naučiť sa aplikovať príkazy pre úpravu blokových schém pre rôzne typy regulačných obvodov.

Základomblokovej schémy riadiaceho systému je blokové zobrazenie všeobecného lineárneho systému so vstupom U(s) a výstupom Y(s):



ktorý je popísaný prenosovou funkciou v tvare

$$G(s) = \frac{Y(s)}{U(s)}$$

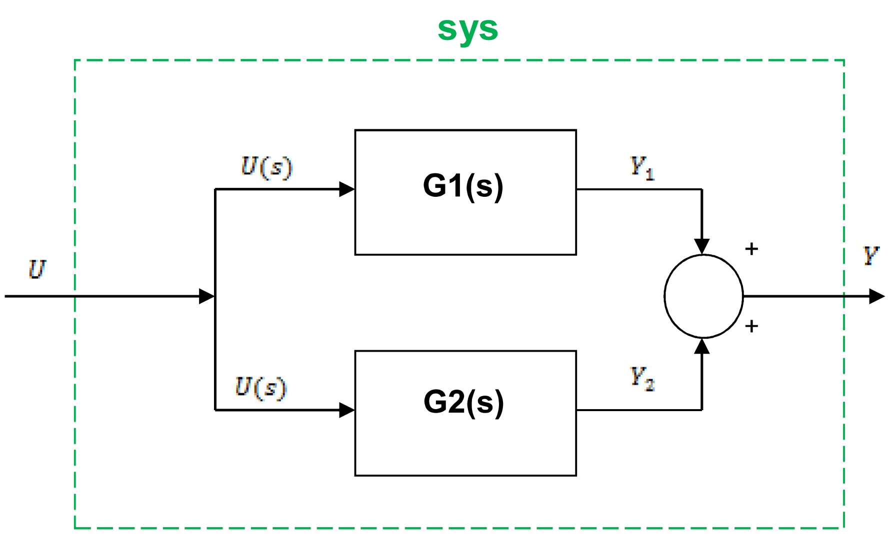
- **Redukcie blokových schém v simulačnom jazyku MATLAB**

Control Toolbox obsahuje príkazy na redukciu blokových schém na jednoduchšie schémy. Tieto príkazy (*cloop, feedback, series, paralel, connect, blkbuild)* vedú k zjednotenému opisu komplexnejšieho systému.

- **Paralelné zapojenie:**

function parallel - generuje celkový prenos dvoch systémov radených paralelne

```
sys = parallel (sys1, sys2)
```



Obrázok 7-4 Paralelné zapojenie prenosov

Z obrázka pre výstup systému vyplýva:

$$Y = Y_1 + Y_2 = G_1 U + G_2 U = U(G_1 + G_2)$$

Výsledný prenos dostávame v tvare:

$$\frac{Y(s)}{U(s)} = G_1 + G_2$$

PRÍKLAD 1

Nech je daný systém pozostávajúci z dvoch paralelne zapojených prenosových funkcií:
$H_1(s) = \frac{1}{s+2}$; $H_2(s) = \frac{s+3}{s+10}$;

Vypočítajte výsledný prenos $H(s)$ s využitím funkcie ***parallel***.

```
sys1 = tf([1],[1 2]);
sys2 = tf([1 3],[1 10]);
[numh,denh] = parallel(sys1,sys2);
sys=parallel(sys1,sys2);
printsys(numh,denh,'s')
```

Výsledok získaný v programovom prostredí MATLAB:

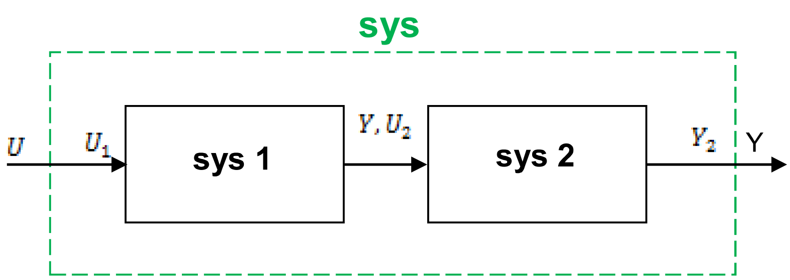
```
Transfer function:

s^2 + 6 s + 16
---------------
s^2 + 12 s + 20
```

- **Sériové zapojenie:**

  $function\ series$ - generuje celkový prenos dvoch systémov zapojených do série

  `sys = series (sys1, sys2)`



Obrázok 7-5 Sériové zapojenie prenosov

Z definície prenosov jednotlivých systémov je zrejmé, že:

$$\frac{Y_1}{U_1} = G_1 \Rightarrow Y_1 = G_1 U_1$$

$$\frac{Y_2}{U_2} = G_2 \Rightarrow Y_2 = G_2 U_2$$

odkiaľ

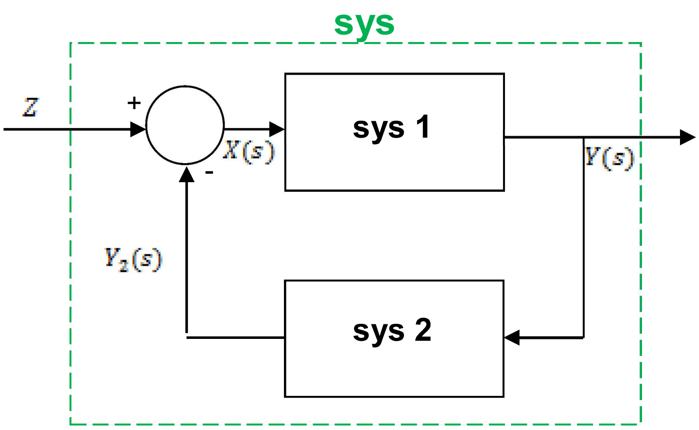
$$U_2 = Y_1 \Rightarrow Y_2 = G_2 G_1 U_1$$

Výsledný prenos má tvar:

$$\frac{Y_2}{U_1} = G_2 G_1$$

- **Spätnoväzobné zapojenie:**

```
sys = feedback (sys1, sys2)
```



Obrázok 7-6 Spätnoväzobná riadiaca štruktúra – prenos na poruchu

Takto volaná funkcia reprezentuje v teórii automatického riadenia výpočet prenosu na poruchu (kde sys1 = $G_S(s)$ – LDS, sys2 = $G_R(s)$ – zvolený prenos regulátora):

$$\frac{Y(s)}{Z(s)} = \frac{Y(s)}{Y_2(s) + X(s)} = \frac{G_s(s)X(s)}{X(s)\,G_s(s)G_R(s) + X(s)} = \frac{\boldsymbol{G_s(s)}}{\boldsymbol{G_s(s)G_R(s) + 1}}$$

PRÍKLAD 2
Napíšte program pre výpočet výsledného prenosu spätnoväzobného usporiadania, ak sys1 = $G_0(s)$, sys2 = H(s) a platí

$$G_0 = \frac{s+1}{(s+3)(s+5)} \text{ a } H(s) = \frac{s+6}{s+10}$$

v spätnej väzbe.

Funkcia $feedback$ vypočíta výsledný prenos spätnoväzobného usporiadania podľa vzorca:

$$G_{CL}(s) = \frac{Y(s)}{Z(s)} = \frac{G_0(s)}{(1+G_0(s)*H(s))};$$

ktorý v teórii riadenia odpovedá prenosu URO na poruchu (Z(s))

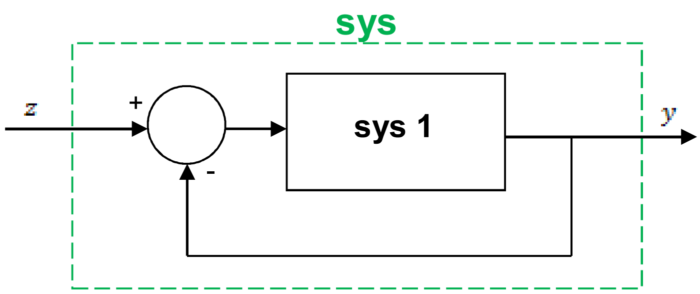
```
numgo = [1 1]; dengo = conv([1 3],[1 5]);
sys1=tf(numgo,dengo);
sys2=tf([1 6],[1 10]); %prenos H(s) v spatnej vazbe
sys=feedback(sys1,sys2); % vysledny prenos URO
printsys(sys,'s'); % nejde potrebujem num, den
```

Poznámky k funkcii *feedback*:

- funkcia ***feedback*** defaultne uvažuje zapojenie so *zápornou spätnou väzbou*
- vyjadrenie *kladnej spätnej väzby* v programovom prostredí MATLAB v Control System Toolboxe vykonáme pomocou príkazu:

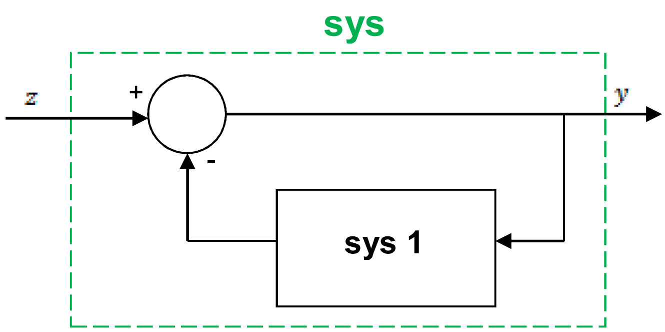
$$sys = feedback(sys1, -sys2)$$

⇨ ak v niektorej vetve nie je zapojený systém s daným prenosom, uvažujeme prenos rovný 1



$sys = feedback(sys1, 1)$

Obrázok 7-7 Prenos nachádzajúci sa v priamej vetve

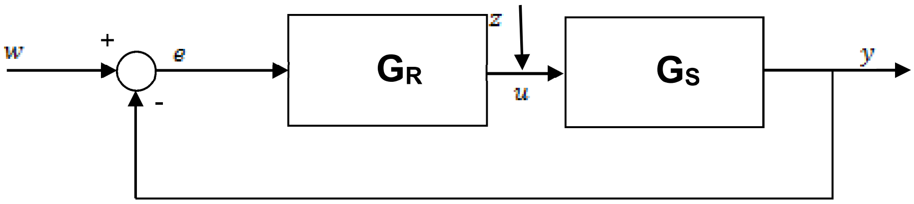


$sys = feedback(1, sys1)$

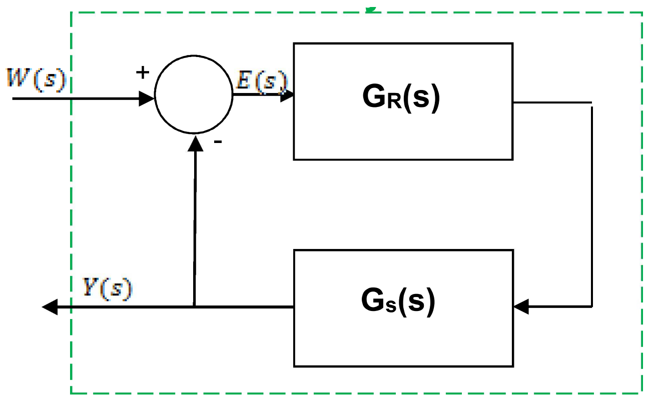
Obrázok 7-8 Prenos nachádzajúci sa v spätnej väzbe

⇨ ak je sys1 v priamej vetve tvorený sériovým zapojením prenosov $G_S(s)$ a $G_R(s)$, kde $G_S(s)$ – LDS, $G_R(s)$ – zvolený prenos regulátora, výsledný prenos vypočítame volaním funkcie feedback v tvare

```
sys = feedback(series(Gr,Gs),1)
```



Obrázok 7-9 Spätnoväzobná riadiaca štruktúra – prenos na požadovanú hodnotu



Obrázok 7-10 Spätnoväzobná riadiaca štruktúra – prenos na požadovanú hodnotu – alternatívne zobrazenie

čo reprezentuje v teórii automatického riadenia výpočet prenosu na požadovanú hodnotu:

$$\frac{Y(s)}{W(s)} = \frac{Y(s)}{E(s) + Y(s)} = \frac{E(s)\,G_R(s)G_s(s)}{E(s) + E(s)G_R(s)G_s(s)} =$$
$$= \frac{G_R(s)G_s(s)}{1 + G_R(s)G_s(s)}$$

<u>PRÍKLAD 3</u>

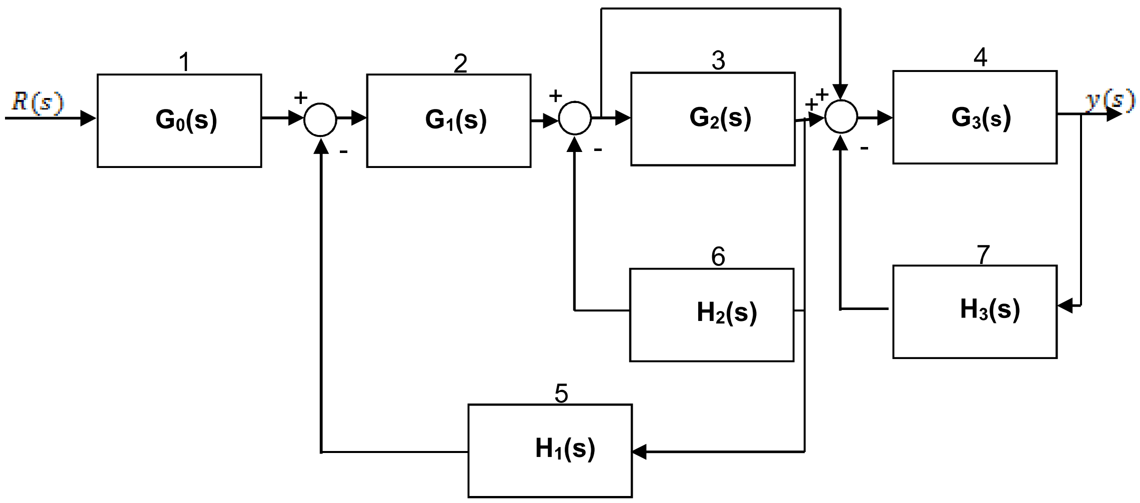
Uvažujeme bloky $G_0(s), H(s)$ z predošleho príkladu, ale zaradené v sérií v priamej vetve. Vytvorte prenosovú funkciu $G_{C3}(s)$ zloženú z blokov $G_0(s), H(s)$ v priamej vetve, kt. su uzatvorené do štruktúry s jednotkovou SV

```
% redukcia blokových schém
ngo=[1 1]; dgo=conv([1 3],[1 5]);
sys1=tf(ngo,dgo); sys2=tf([16],[1 10]);
sys12=series(sys1,sys2);
sys=feedback(sys12,1); % G_0H=G_0*H
printsys(sys,'s'); % G_c3 = G_0H/1+G_oH = Y/W
```

Riešenie komplikovaného systému zahŕňa niekoľko krokov. Nasledujúce príkazy obsahuje programové prostredie MATLAB v Control Toolbox. Tieto príkazy si ukážeme na nasledujúcom probléme.

<u>PRÍKLAD 4</u>

Zaoberajme sa štruktúrou na Obr. 8 . Úlohou je vypočítať výsledný prenos tejto štruktúry.



Obrázok 7-11 Bloková schéma zložitého systému

Bloková štruktúra zložitého systému $\frac{Y(s)}{R(s)}$ =?

Predpokladajme, že:

$$G_0(s) = 1;\ G_1(s) = \frac{1}{s+1};\ G_2(s) = \frac{1}{s+2};\ G_2(s) = \frac{1}{s+3};\ H_1(s) = 4\ ;\ H_1(s) = 8;\ H_1(s) = 12.$$

Použijeme príkazy $bbuild$, $connect$ pre vytvorenie m-file-u buidsys.m

- ⇨ Každému systému priradíme číslo (viď obr.)
- ⇨ Nakoľko pracujeme so 7 prenosovými funkciami, vložíme ich do programového priestoru MATLAB ako polynóm n_i, d_i alebo pomocou funkcie „*tf*".

```
% redukcia zložitej blokovej schémy
n1=1; d1=1; n2=1; d2= [1 1]; n3=1; d3=[1 2]; n4=1; d4=[1 3];
n5=4; d5=1; n6=8; d6=1; n7=12; d7=1;
nblocks = 7;        % počet subsystémov
blkbuild    % používa premennú nblocks na výstavbu systému, vykoná sa to v
stavovom priestore použitím "tf2ss";
```

```
% vytvára jeden blokovo-diagonálny stavový model použitím
% funkcie "append"
q=[2 1 -5 0 0      % vytvoríme maticu q
    3 2 -6 0 0      % identifikuje prepojenia medzi subsystémami
    4 2 -6 3 -7     % každý riadok odpovedá samostatnému subsystému
    5 3 0 0 0        % Prvé číslo je pridelene subsystému
    6 3 0 0 0        % ostatne čísla určujú, ktoré bloky majú svoj výstup
    7 4 0 0 0];      % pripojený na vstup tohto systému.

input = 1;  % určený vstupný blok štruktúry
output = 4;      % určený výstupný blok štruktúry
[Ad,Bd,Cd,Dd]=connect(A,B,C,D,q,input, output)
      % redukcia systému po vykonaní prepojení na 1st.m.
        [num,den]=ss2tf(Ad,Bd,Cd,Dd);
printfsys(num,den,'s')
```

Poznámka

1. *nblocks* – špecifikácia 7 subsystémov
2. *blkbuild*– funkcia konvertuje vsetky opisy prenosovými funkciami na modeli v stavovom priestore →***tf2ss***→ a tvorí jeden z blokovo diagonálny stavový model obsahujúci A,B,C,D opakovaným použitím ***append***.
3. Vytvorenie matice q → definuje prepojenia medzi systémami. Každý riadok q odpovedá jednému subsystému. Prvé číslo je číslo subsystému, ostatné čísla určujú, ktoré bloky majú výstupy pripojené na vstup subsystému. (1.riadok ~ subsystém 2 ($G_1(s)$) :$G_0(s)$~1, $H_1(s)$~ -5
4. Funkcia ***connect*** vykonáva prepojenia a redukuje celý systém na jeden stavový model.

Testovanie:

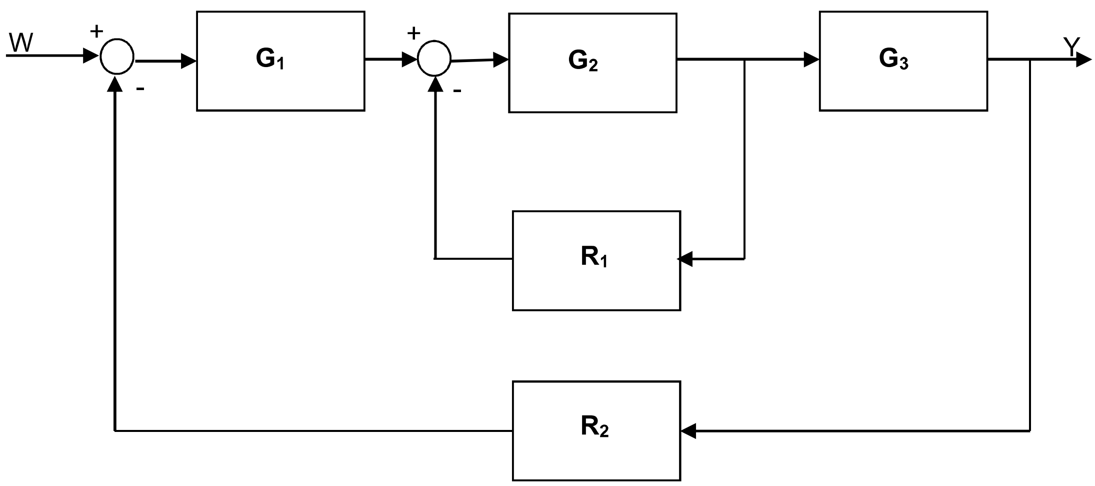
$$\frac{num}{den} = \frac{-7.105e - 15s^2 + 1s + 3}{s^3 + 26s^2 + 179s + 210} \quad \cong 0$$

Kontrola porovnať polohy pólov a núl pre opis stavovým modelom a opisom prenosovou funkciou ~ musia byť identické!

Funkcia ***minreal***→ funkcia na získanie minimálnej formy
Programové prostredie MATLAB pri operáciách spájania nevykompenzuje spoločné korene činiteľa a menovateľa. Prenosovej funkcie → odporúča sa použiť príkaz funkcie minreal!

PRÍKLAD 5

Pre blokovú schému regulačného obvodu na Obr. 9 vypočítajte výslednú prenosovú funkciu $G_{\frac{Y}{W}}(s)$. Pri spájaní jednotlivých blokov použite príkazy prebraté skôr.



Obrázok 7-12 Bloková schéma zložitého systému

$$G_1(s) = \frac{0,5}{2s};\ G_2(s) = \frac{0,1}{s(0,15s+0,5)};\ G_3(s) = \frac{10}{1,2s+1};\ R_1(s) = 2;\ R_2(s) = \frac{s+0,15}{10s}$$

Na základe pravidiel blokovej algebry:

$$G_{\frac{Y}{W}}(s) = \frac{Y(s)}{W(s)} = \frac{G_1(s) * G_2(s) * G_3(s) * R_1(s) * R_2(s)}{1 + G_2(s) * R_1(s) + G_1(s) * G_2(s) * G_3(s) * R_2(s) * R_1(s)}$$

```
g1s=tf(0.5,[2 0]) % inicializácia G1(s)
g2s = tf(0.1,[0.5 0.5 0])%inicializácia G2(s)
g3s=tf(10, [1.2 1]) % inicializácia G3(s)
r1s=tf(2); r2s=tf([1 0.15],[10 0])% inicializácia R1 a R2
numgyw=g1s*g2s*g3s*r1s*r2s% čitateľ G_Y/W -> URO
dengyw=1+g2s*r1s+g1s*g2s*g3s*r2s*r1s % menovateľ G_Y/W
gyw=numgyw/dengyw % celková prenosová funkcia URO
gywpn =zpk(gyw) % vyjadrenie pomocou pólov a núl
gywm=minreal(gyw) % min. realizácia pr. funkcie URO
```

zjednodušený postup → najprv spočítame vnútornú slučku a následne potom celú funkciu pomocou funkcie feedback

```
spv1=feedback(g2s,r1s) % vnútorná SV G2(s) a R1(s)
```

transfer function: $\frac{0,1}{0,15s^2+0,5s+0,2}$

```
spv2=feedback(g1s*spv1*g3s,r2s) % 2. SV tvorená G1(s) ,SPV1, G3(s) a R2(s)
```

transfer function: $\frac{5s}{36s^5+15s^4+4,8s^3+4s^2+0,5s+0,075}$;

Prevod do formy „póly-nuly-zosilnenie“: ***zpk(spv2)***

Z/P/Gain: $\frac{1,3889s}{(s+2,859)(s+0,096)(s+0,297)(s^2+0,04s+0,02)}$

Zjednodušenie prenosovej funkcie: ***minreal(spv2)***

transfer function: $\frac{1,389s}{s^5+4,16s^4+4,11s^3+1,11s^2+0,13s+0,02}$

riešenie cez $\boldsymbol{connect}$→ q

Použitím prebratých príkazov nájdite výslednú prenosovú funkciu ak: regulovaný proces je opísaný

$G_p(s) = \frac{5}{(40s+1)^2}e^{-40s}$ a regulátor PID je vyjadrený prenosovou funkciou

$$G_R(s) = 8\left(1 + \frac{1}{80s} + 20s\right) = \frac{160s^2+80s+1}{10s}$$

7.4 Zadanie č. 4. : Modelovanie a analýza modelu fyzikálneho systému jednosmerný motor v prostredí MATLAB s využitím funkcií Control Toolboxu

ZADANIE:
Uvažujte fyzikálno- matematický model dynamického systému, ktorý je popísaný lineárnou diferenciálnou rovnicou (LDR) 2. a vyššieho rádu.

ÚLOHA: *Navrhnite m-file v simulačnom jazyku Matlab, ktorý umožní:*

1. *Zadefinovanie prenosovej funkcie opisujúcej LDS v s-oblasti (v polynomiálnom tvare, v tvare póly/nuly) a v stavovom priestore pomocou matíc A,B,C,D.*
2. *Konverziu modelov zo stavového priestoru do tvaru prenosovej funkcie a naopak.*
3. *Analýzu LTI DS v časovej (prechodová charakteristika, impulzná charakteristika, odozva na ľubovoľný vstupný signál) a frekvenčnej oblasti (Nyquistová, Nicholsová a Bodeho charakteristiky).*
4. *Vyhodnotenie stability uvažovaného LTI dynamického systému na základe získaných odoziev na rôzne typy vstupných signálov.*

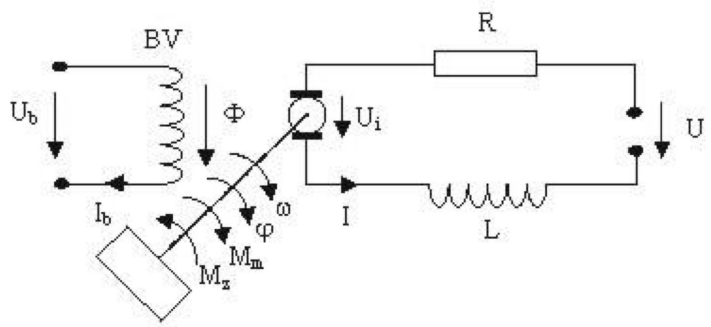
Parametre:

$B\ [N.m.s^{-1}]$ – *koeficient viskózneho trenia*

R [Ω] – *odpor*

L [H] – *indukcia*

$C_u\ [N.m\ /\ Amp]$ – *momentová konštantamotora*



Pôsobiace momenty:

$J\ [kg.m^2]$ – *moment zotrvačnosti*

$M_m\ [Nm]$– *krútiaci moment*

$M_{dyn}\ [kg.m^2s^{-2}]$ – *dynamický moment*

Vstup:
$U\ [V]$– *zdroj napätia*
Výstup:
$\omega[s^{-1}]$- *uhlová rýchlosť hriadeľa*
$\varphi\ [rad]$ - *uhlová poloha hriadeľa*

Na motore sa vytvára indukované napätie: $U_i = C_u * \omega$

Získame dve rovnice. Prvá je diferenciálna rovnica vyplývajúca z 2.Kirchhoffovho zákona :

$$L\frac{dI(t)}{dt} + RI(t) + U_i(t) = U(t)$$

druhá rovnica popisuje mechanické deje – pohyb (vyplýva z 2.Newtonovho zákona):

$$M_m(t) - M_z(t) = M_{dyn}(t).$$

Pre krútiaci moment hriadeľa približne platí:

$$M_m(t) = C_u I(t)$$

Záťažový moment (typický tvar) :
$M_z(\omega) = B * \omega$

Dynamický moment je daný rovnicou:
$M_{dyn}(t) = J\frac{d\omega}{dt} = J\frac{d^2\varphi}{dt^2}$

Do 2. rovnice dosadíme rovnice pre $M_m(t)$, $M_z(t)$, $M_{dyn}(t)$ z ktorých následne dostaneme:

$$M_m(t) = M_{dyn}(t) + M_z(t) = J\frac{d\omega}{dt} + B * \omega = C_u I(t)$$

Z tejto rovnice si vyjadríme prúd $I(t)$:

$$I(t) = \frac{J}{C_u}\frac{d\omega}{dt} + \frac{B}{C_u} * \omega$$

Pre dosadenie do 1.diferenciálnej rovnice potrebujeme aj prvú deriváciu prúdu:

$$\frac{dI(t)}{dt} = \frac{J}{C_u}\frac{d^2\omega}{dt^2} + \frac{B}{C_u} * \frac{d\omega}{dt}$$

Teraz dosadíme rovnice do prvej diferenciálnej rovnice, pričom za indukované napätie na motore dosadíme $U_i = C_u * \omega$:

$$L * \left(\frac{J}{C_u}\frac{d^2\omega}{dt^2} + \frac{B}{C_u} * \frac{d\omega}{dt}\right) + R * \left(\frac{J}{C_u}\frac{d\omega}{dt} + \frac{B}{C_u} * \omega\right) + C_u * \omega = U(t)$$

$$\frac{L * J}{C_u} * \frac{d^2\omega}{dt^2} + \left(\frac{L * B}{C_u} + \frac{R * J}{C_u}\right) * \frac{d\omega}{dt} + \left(\frac{R * B}{C_u} + C_u\right) * \omega = U(t)$$

resp.
$$L * J * \frac{d^2\omega}{dt^2} + (L * B + R * J) * \frac{d\omega}{dt} + \left(R * B + {C_u}^2\right) * \omega = U(t) * C_u$$

Použitím Laplaceovej transformácie získame diferenciálnu rovnicu v oblasti laplaceových obrazov:

$$L * J * s^2 * Y(s) + (L * B + R * J) * s * Y(s) + \left(R * B + {C_u}^2\right) * Y(s) = C_u * U(s)$$

Z ktorej tejto rovnice získam prenosovú funkciu:

$$G(s) = \frac{C_u * U(s)}{s^2(JL) + s(BL + JR) + ({C_u}^2 + BR)}$$

Pre zápis v programovom prostredí MATLAB s použitím Control Math Toolbox:

$$num = C_u U(t)$$

$$denum = JL + (BL + JR) + ({C_u}^2 + BR)$$

<u>Prepis do stavového popisu v maticovom tvare:</u>

Stavový popis tvoria dve rovnice v maticovom tvare, kde prvá stavová rovnica je diferenciálna,predstavujúca rovnicu stavu:

$x'(t) = Ax(t) + Bu(t),$

kde $x(t)$ je stavový vektor,$u(t)$ je vektor vstupov, A je matica vnútorných väzieb systému a B je matica väzieb systému na vstup. Druhá stavová rovnica je algebraická rovnica predstavujúca rovnicu výstupu:

$y(t) = Cx(t) + Du(t),$

kde$y(t)$ je vektor výstupov, $u(t)$ je vektor vstupov,C je matica väzieb výstupu na stav a D matica väzieb vstupu na výstup.

Stavový vektor:

$$\underline{x} = \begin{bmatrix} I \\ \omega \end{bmatrix}$$

Treba vyjadriť prvú deriváciu oboch stavových veličín :

Dosadením do $M_m - M_z = M_{dyn}$ rovnice :

$$M_m(t) = C_u I$$
$$M_z(\omega) = B * \omega$$
$$M_{dyn}(t) = J\frac{d\omega}{dt}$$

Dostaneme :

$$J\frac{d\omega}{dt} = C_u I - B * \omega$$

z ktorej si jednoducho vyjadrime prvú deriváciu uhlovej rýchlosti:

$$\frac{d\omega}{dt} = \frac{C_u}{J}I - \frac{B}{J} * \omega$$

A z rovnice

$$L\frac{dI}{dt} + RI + U_i = U$$

kde za indukované napätie na motore dosadíme :

$$U_i = C_u * \omega$$

jednoducho vyjadríme prvú deriváciu prúdu :

$$\frac{dI}{dt} = \frac{U}{L} - \frac{R}{L}I - \frac{C_u}{L} * \omega$$

$$\begin{bmatrix} \dot{I} \\ \dot{\omega} \end{bmatrix} = \begin{bmatrix} -\frac{R}{L} & -\frac{C_u}{L} \\ \frac{C_u}{J} & -\frac{B}{J} \end{bmatrix} \begin{bmatrix} I \\ \omega \end{bmatrix} + \begin{bmatrix} \frac{1}{L} \\ 0 \end{bmatrix} * U$$

$$y = \omega = \begin{bmatrix} 0 & 1 \end{bmatrix} \begin{bmatrix} I \\ \omega \end{bmatrix}$$

<u>Hodnoty parametrov pre modelovanie a analýzu jednosmerného motora v programovom prostredí MATLAB:</u>

$J = 0{,}01\ kg.m^2, B = 0{,}1\ N.m.s$, $R = 1\ \Omega, L = 0{,}5H,\ C_u = 0{,}01\ ^{N.m}/_{Amp},\ U = 1V$

Riešenie úloh modelovania a analýzy v programovom prostredí MATLAB s využitím vlastnej aplikácie s menu rozhraním

```
voľba=menu('','definovanie prenosovej funkcie','konverzia','časové charakteristiky',' frekvenčné charakteristiky','stabilita','koniec');
% tvorba menu a jeho obsahu, prvý parameter je názov, ostatné možnosti (tlačidlá) menu

switch volba
case 1,
        vstup
case 2,
        konverzia
case 3,
        casova
case 4,
        frekvencna
case 5,
        stabilita
case 6,
end
```

Obrázok 7-13 Grafické MENU vytvorené pomocou funkcie menu

Definovanie vstupov v simulačnom jazyku MATLAB:

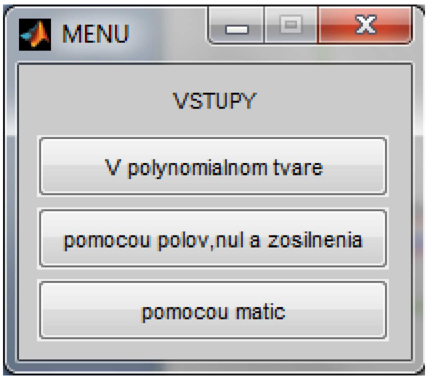
```
function [sys]=vstup()

global sys;
global a b c d;

voľba = menu('VSTUPY','V polynomiálnom tvare ','pomocou pólov ,núl a zosilnenia ','pomocou matíc');

switch volba
case 1,
        num=input('Zadaj citatel ... num = ');
        den=input('zadaj menovatela... den = ');
        sys=tf(num,den) % tvorba systému v polynomiálnom tvare
    case 2,
        z=input('zadaj nuly : ');
```

```
        p=input('zadaj poly : ');
        k=input('zadaj zosilnenie : ');
        sys=zpk(z,p,k) % tvorba systému pomocou pólov a núl
case 3,
        a=input('zadaj maticu A: ');
        b=input('zadaj maticu B: ');
        c=input('zadaj maticu C: ');
        d=input('zadaj maticu D: ');
        sys=ss(a,b,c,d) % tvorba systému pomocou matíc stavového popisu
end
hlavny % skok späť do menu
return
```



Obrázok 7-14 Výber možnosti zadania systému

```
Zadanie systému v polynomiálnom tvare:
Zadaj čitateľa ... num = 0.01
Zadaj menovateľa... den = [0.005 0.06 10.01]
 Transfer function:
          0.01
--------------------------
0.005 s^2 + 0.06 s + 10.01
```

Ďalšie príkazy ktoré je potrebné použiť pri riešení tohto zadania v Control Toolbox:

Funkcie pre konverziu:

```
[num,den]=tfdata(sys,'v')
[num,den]=ss2tf(a,b,c,d,1)
[a,b,c,d]=tf2ss(num,den)
[z,p,k]=ss2zp(a,b,c,d)
[z,p,k]=tf2zp(num,den)
[num,den]=zp2tf(z,p,k)
[a,b,c,d]=zp2ss(z,p,k)
```

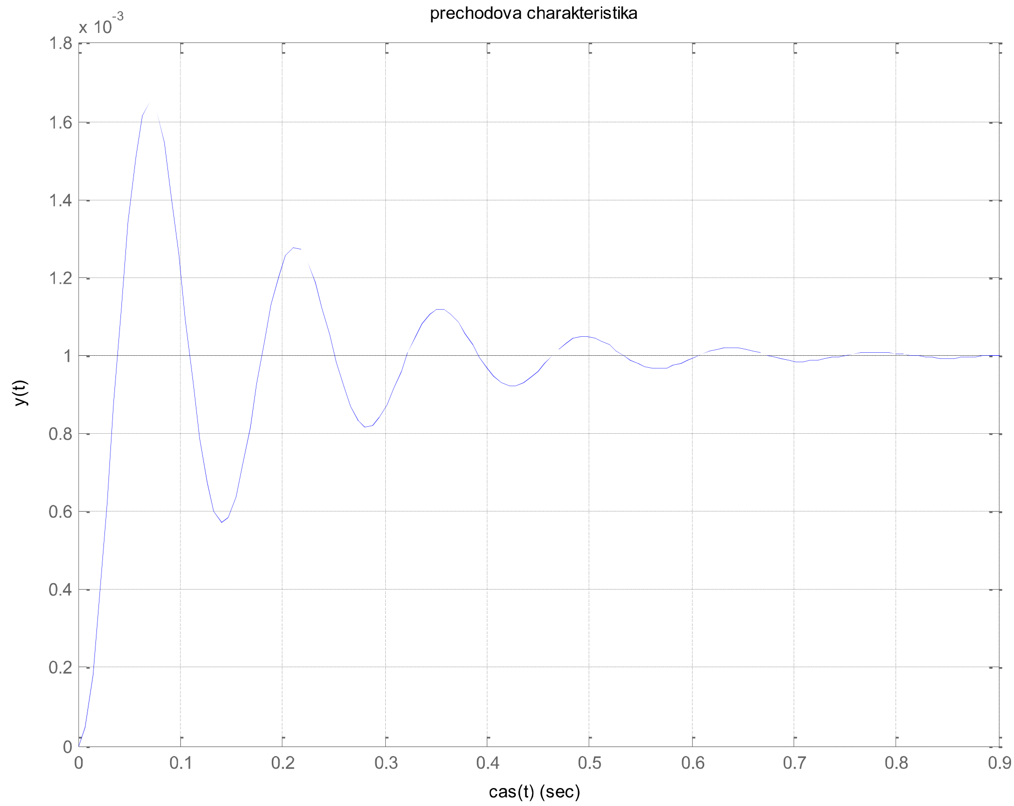
Časové charakteristiky:

```
step(sys);
impulse(sys)
```

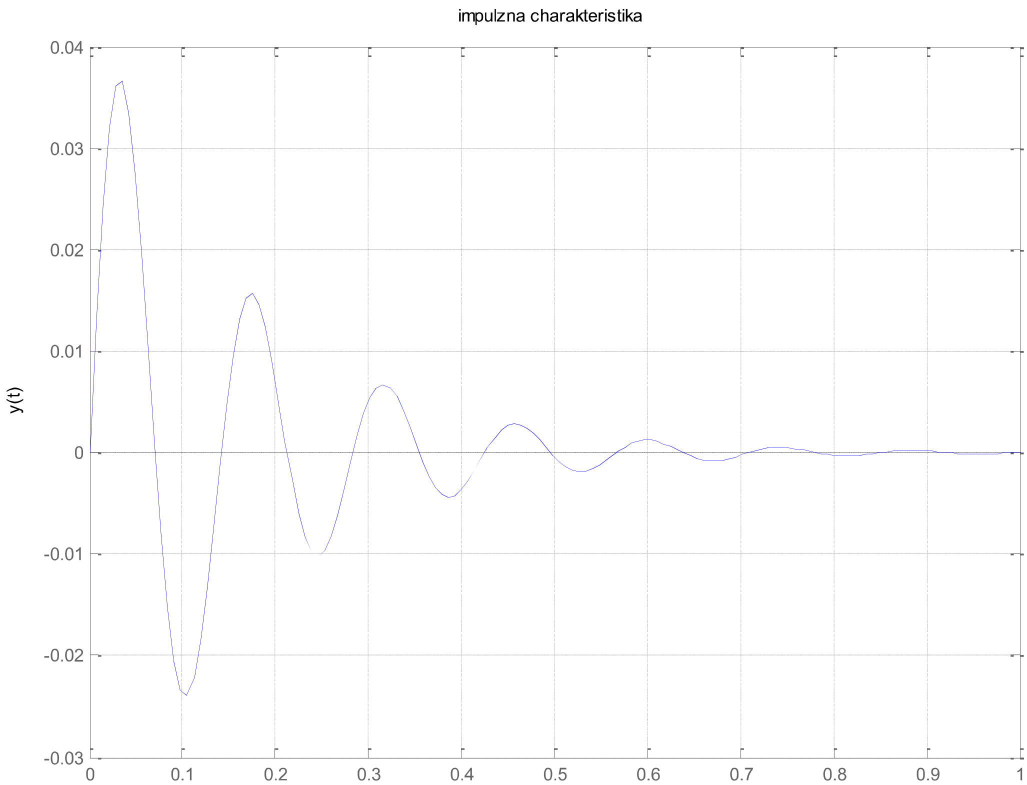
Frekvenčné charakteristiky:

```
nyquist(sys);
bode(sys);
nichols(sys);
```

Časové charakteristiky modelu otáčok motora:

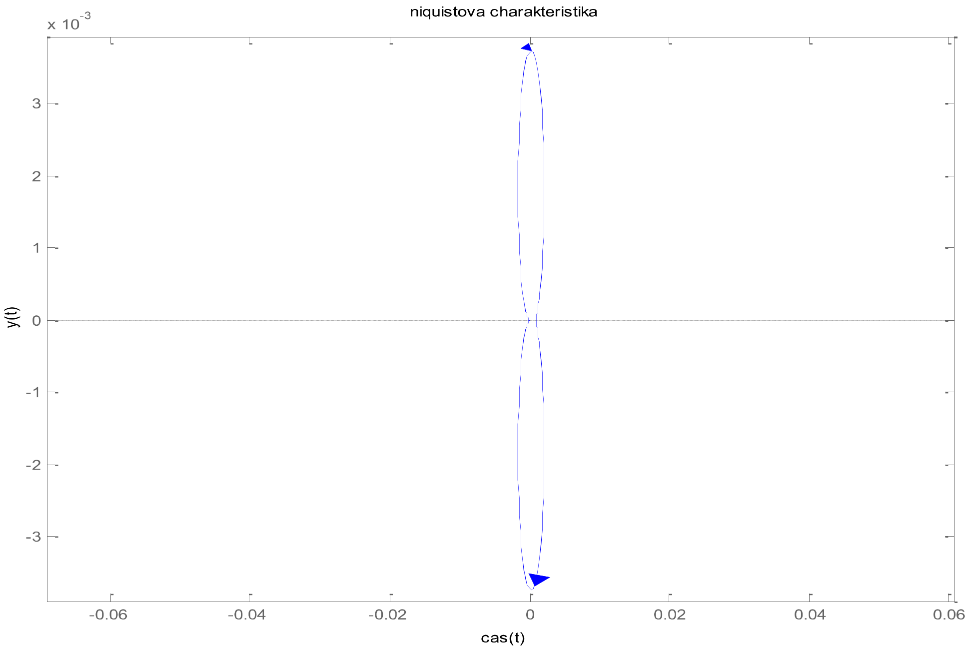


Obrázok 7-15 Prechodová charakteristika modelu otáčok motora

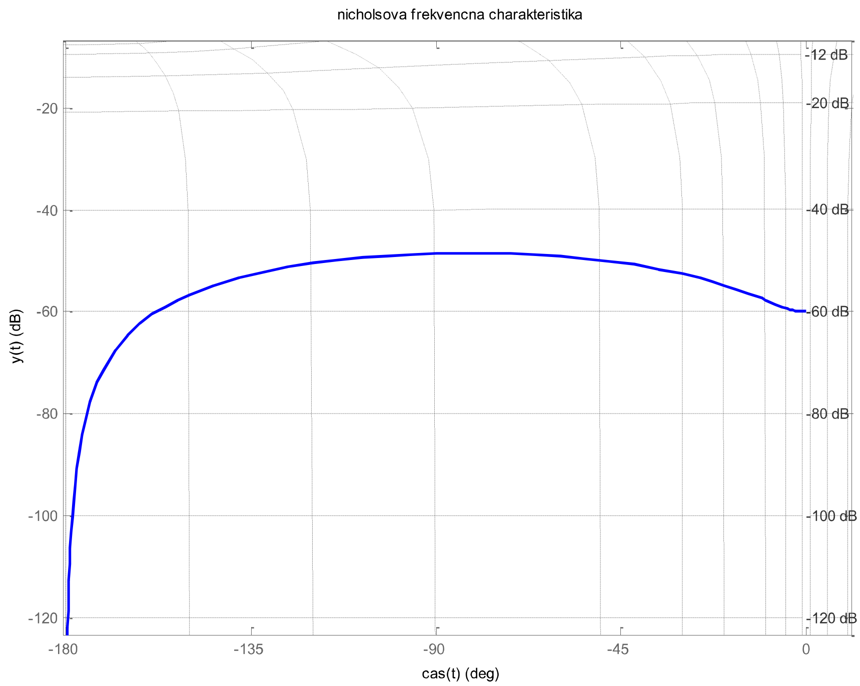


Obrázok 7-16impulzná charakteristika modelu otáčok motora

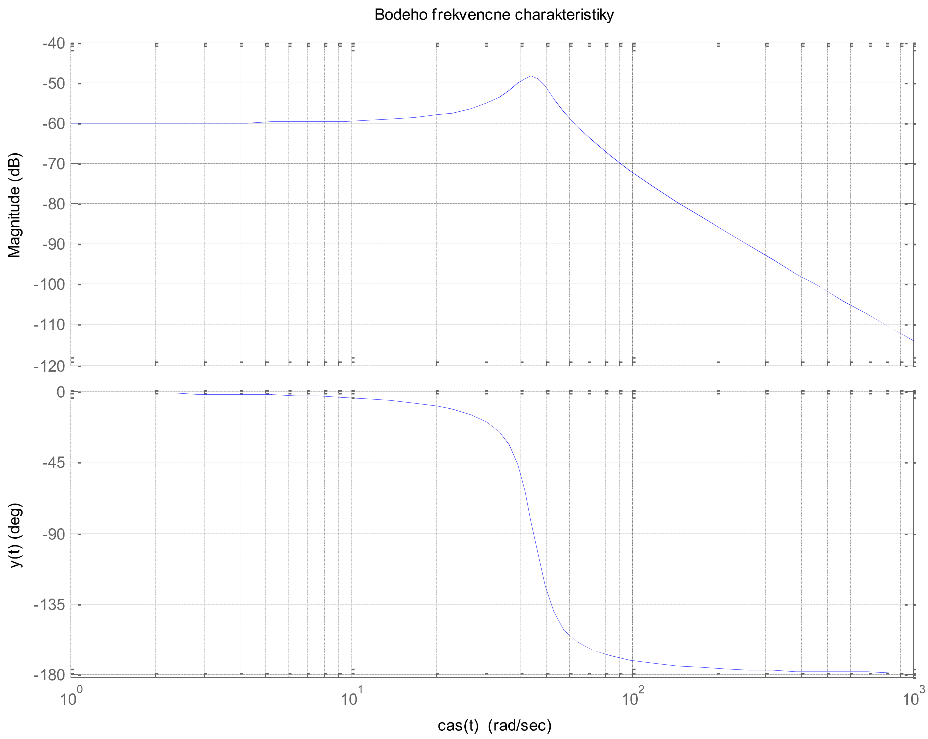
Frekvenčné charakteristiky modelu otáčok motora:



Obrázok 7-17 Nyquistova charakteristika



Obrázok 7-18 Nicholsova charakteristika



Obrázok 7-19 Bodeho frekvenčné charakteristiky

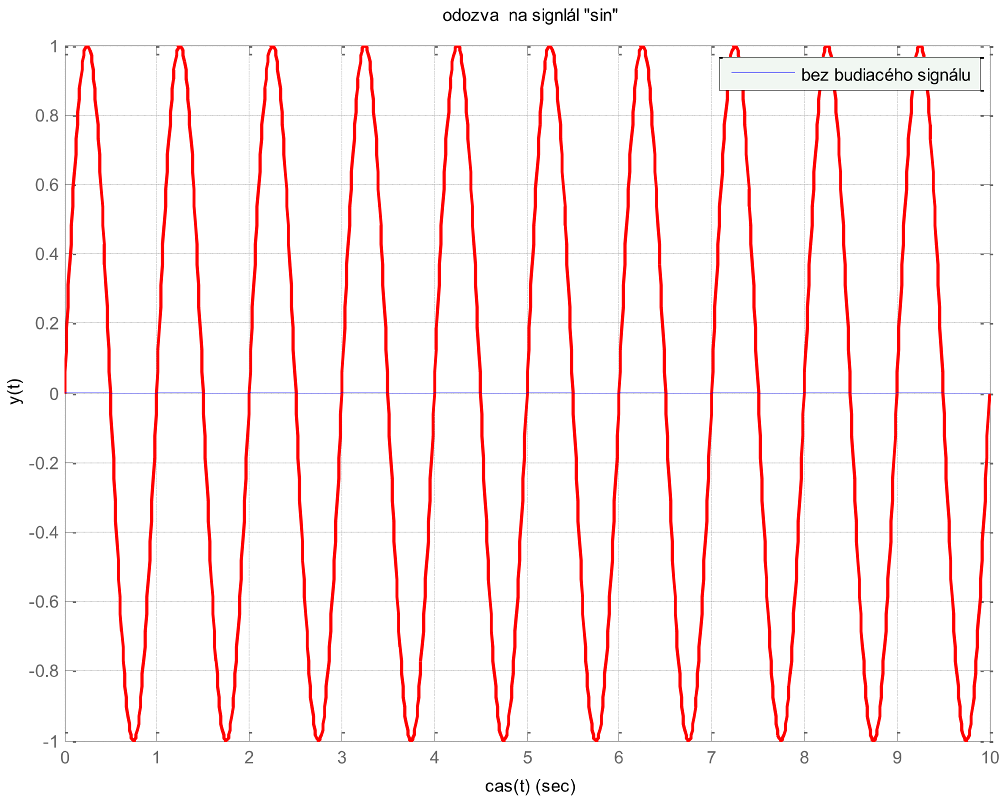
Odozva na ľubovoľný signál:

```
function []=odozva()
global sys ;

type=input('Zadaj typ signalu napr. sin, pulse, square: ');
Ton=input('zadaj periodu vzorkovania : ');
Tf=input('zadaj celkovu dobu simulacie : ');
Ts=input('Zadaj vzorkovaci cas Ts : ');

[u,t]=gensig(type,Ton,Tf,Ts)
lsim(sys,u,t)
grid;
title('odozva na lubovolny vstupny signal');
xlabel('cas(t)'); ylabel('y(t)');

hlavny
return
```



Obrázok 7-20 Odozva na sínusový signál

Vyhodnotenie stability systému:

```
%vyhodnotenie stability
function []=stabilita()
global sys;
[num,den]=tfdata(sys,'v');

r=roots(den)
max=size(r);
test=1;

for a= 1:max(:,1)
if r(a) > 0
        test=0;
end
end

if test==0
    disp('nestabilny')
else disp('stabilny alebo na hranici')
end
hlavny
return
```