MÉRÉSI JEGYZŐKÖNYV (TÁVOKTATÁSI VERZIÓ)

A mérés tárgya:	Rendszer-identifikáció és szabályozás (8. mérés)
A mérést végzi:	Veszelyi Bence Balázs (V3UWB0)
Mérőcsoport:	H12, 41
A mérés időpontja:	2021-03-01
Feladat sorszáma:	32

Neptun kód első karaktere	A feladat sorszáma (index)
0	1
1	2
2	3
3	4
4	5
5	6
6	7
7	8
8	9
9	10
A	11
B	12
C	13
D	14
E	15
F	16
G	17
H	18

Neptun kód első karaktere	A feladat sorszáma (index)
I	19
J	20
K	21
L	22
M	23
N	24
O	25
P	26
Q	27
R	28
S	29
T	30
U	31
V	32
W	33
X	34
Y	35
Z	36

Felhasznált eszközök

Matlab Live Script	rendszeridentifikacio.mlx
Matlab Simulink modellek	identifikaciohoz.slx szimulaciohoz_diszkret.slx szimulaciohoz_folytonos.slx szabalyozashoz.slx

A fájlok letölthetők a tárgy honlapjáról (bejelentkezés után):
https://www.mit.bme.hu/oktatas/targyak/vimiac13/8-meres-rendszer-identifikacio-es-szabalyozas

A normál mérés során használt szakaszt az alábbi videó mutatja be:
https://web.microsoftstream.com/video/e29b08db-13fc-45a4-8ae0-45e3bb34a30e

Mérési feladatok

A számítógépen indítsa el a Matlabot.

A Matlabban állítsa be a megfelelő munkakönyvtárat, és nyissa meg a `rendszeridentifikacio.mlx` Live Scriptet.

Ellenőrizze, hogy a munkakönyvtárban megtalálható-e a méréshez szükséges négy Simulink modell (`identifikaciohoz.slx`, `szimulaciohoz_diszkret.slx`, `szimulaciohoz_folytonos.slx`, `szabalyozashoz.slx`).

1. Ismeretlen lineáris rendszer identifikációja és szabályozása állapottérbeli módszerekkel (kötelező feladat)

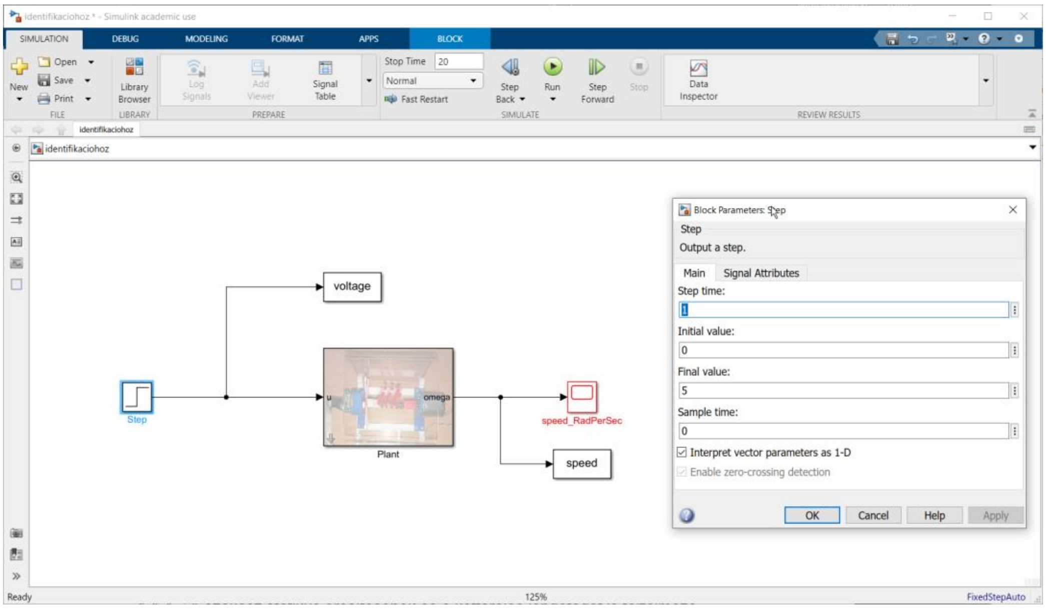
1.1. A szakasz analízise.

A mérés elején nézze meg a szakaszt és a feladatot röviden bemutató videót.

1.1.1. A `rendszeridentifikacio.mlx` Live Script első szekciójának futtatása. (Ebben beállításra kerül a mintavételi idő. Az egyéni feladat sorszámát is itt kell megadni.) Automatikusan megnyílik az `identifikaciohoz.slx` Simulink modell.

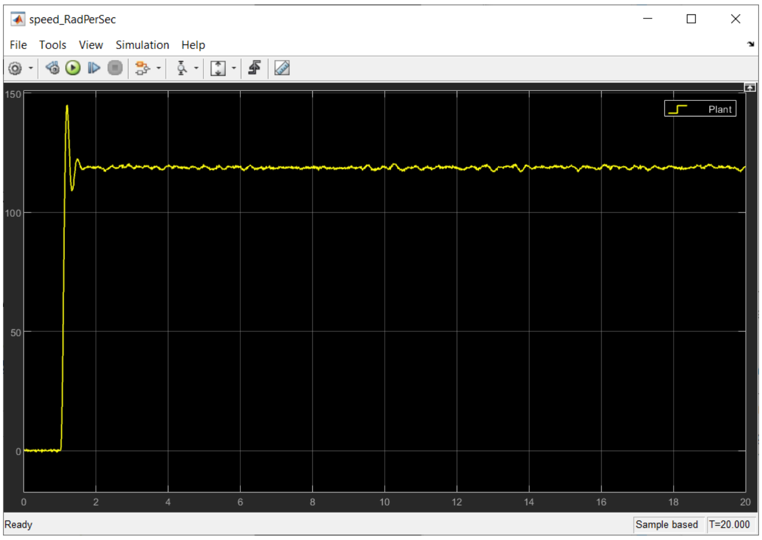
1.1.2. A szakasz bemenetére megfelelő blokk bekötése ugrásválasz felvétele érdekében (Step típusú bemenet). Normal módú futtatás.

A megfelelő fájlok megnyitása és az index beállítását követően, az első ugrásjelet ráadtam a szakasz bemenetére (0→5 V):

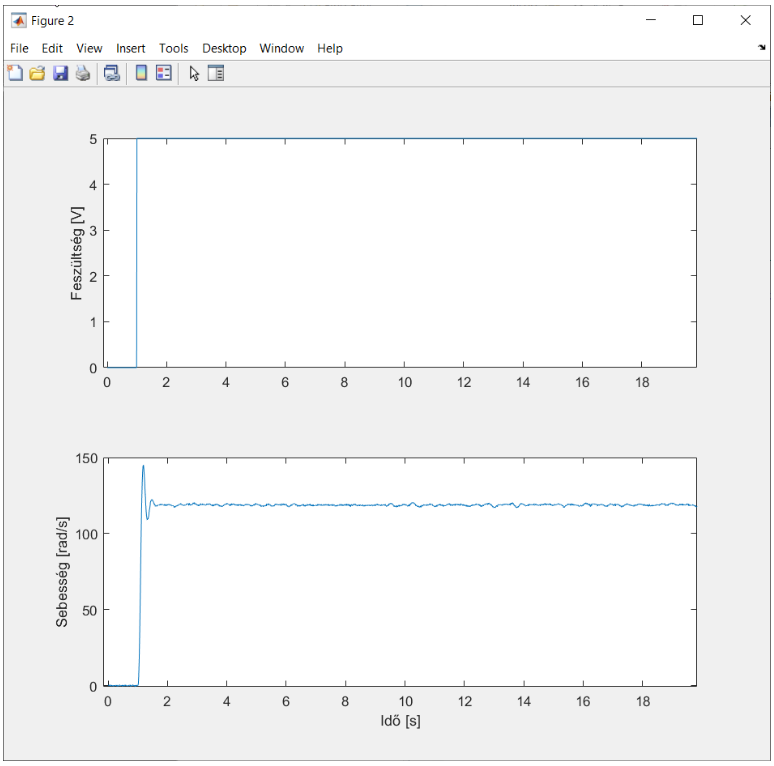


1.1.3. A szakasz ugrásválaszának előállítása: bemeneti feszültség 0-ról 5V-ra állítása, tranziens elmentése. Szakasz kimenetének vizsgálata.

Az eredmény a következő volt:



A második matlab szekció ki is rajzolta:



T_kezdo-nek 1-et, T_veg-nek pedig 1.8-at választottam az ábra alapján.

1.1.4. A szakasz statikus erősítésének és a kéttárolós lengőtagot is tartalmazó szakasz csillapítatlan sajátfrekvenciájának (ω_0), időállandójának ($T_\omega = 1/\omega_0$) és csillapításának (ξ) becslése.

A harmadik matlab szekció által kirajzolt sebesség-iő grafikonok segítségével meghatároztam T_m-et, 0.18 s-re adódott, majd ezen a ponton leolvastam a v_{max}-ot, amely 144.76 rad/s. A v_∞-re pedig 119-et becsültem.

A becsült paraméterekből következtetve a Δv=0.2165, ebből kiszámolható ζ és T_m segítségével w_0. Matlab segítségével számoltam ki.

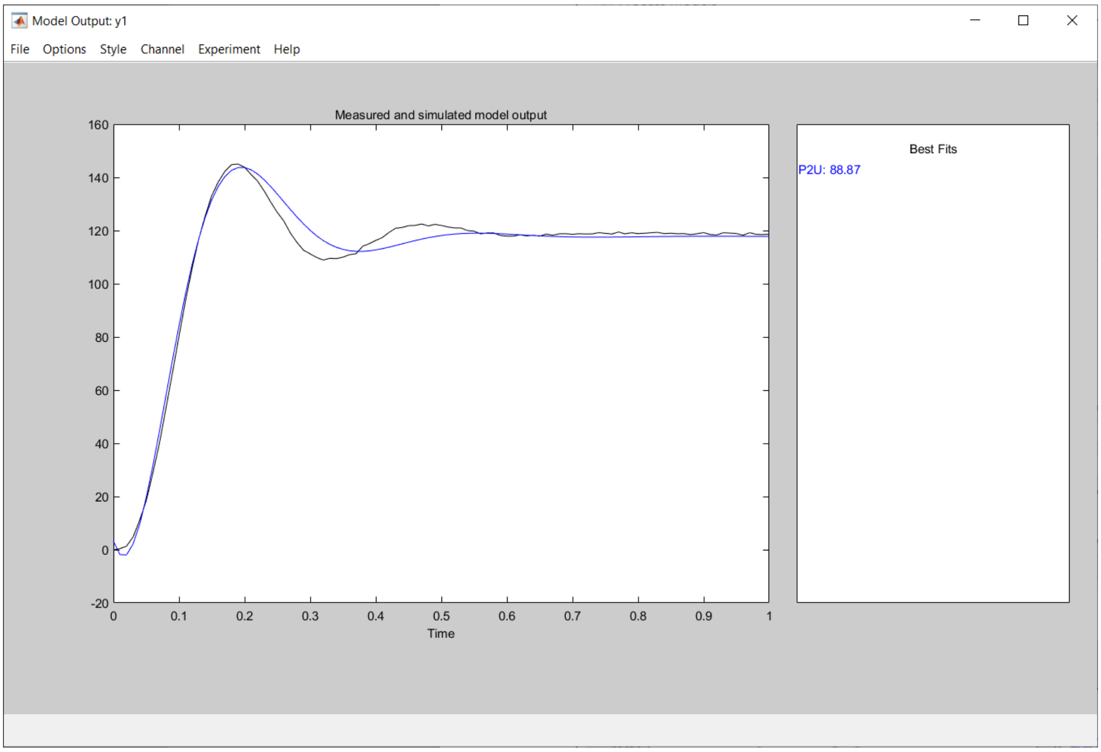
w_0= 19.4136 rad/s

ζ= 0.4379

T_w=0.0515

1.1.5. A becsült paraméterekkel rendelkező káttárolós tag kimenetének összevetése a mért ugrásválasszal. Következtetések levonása.

Az adatok becslése és kiszámolása után megnyitottam a System Identification alkalmazást a modellillesztéshez, majd az adatok felvitele után ezt az eredményt kaptam:



K=23.5588

Az általam becsült adatok közelítőleg helyesek, minimális eltérés van a v_{max} és a T_m értékeim és a valós adatok között. Az illesztés pontossága 88.87%.

1.1.6. A mintavételi idő megválasztásának ellenőrzése. Vegye figyelembe, hogy a mintavételi időnek a Shannon-tétel szerint a felgyorsítandó zárt szabályozási kör minden jeléhez jónak kell lennie.

A w_0 -ból f_0=3.0898 Hz. Ez körülbelül egész szám szorosa az f_s-nek, így teljesül a Shannon-tétel. (f_s=1/T_s , T_s=0.01 s)

1.2. Bemenő jel generálás identifikációhoz.

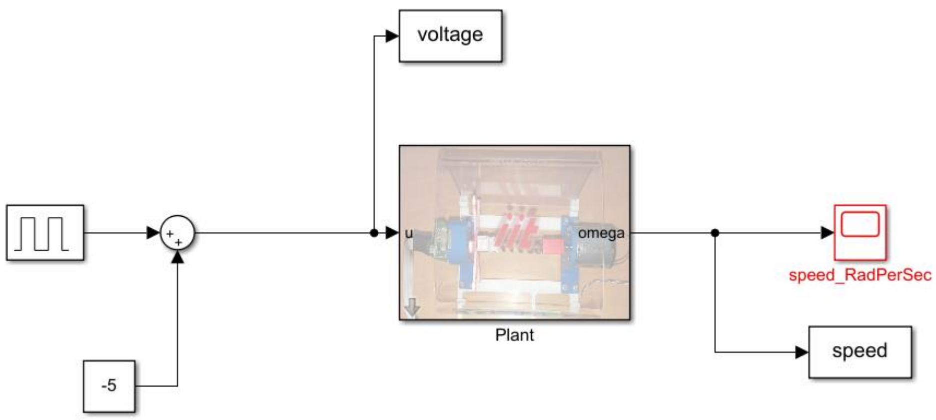
1.2.1. A szakasz becsült jellemzői alapján tervezzen megfelelő bemenő jelet Simulink-ban a szakasz identifikációjához.

1.3. Adatgyűjtés a szakasz identifikációjához

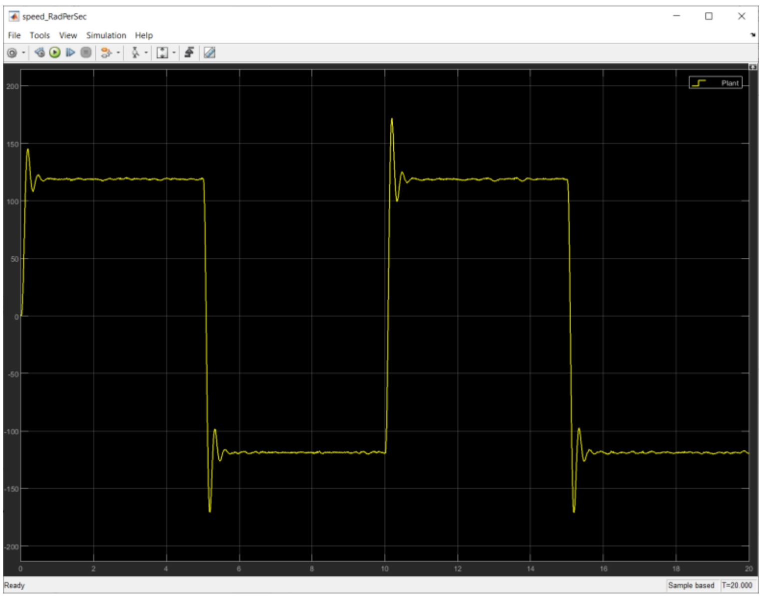
1.3.1. Az `identifikaciohoz.slx` Simulink modell Normal módú futtatása. A blokkok paramétereit a futtatás előtt kell beállítani, a jeleket a futtatás során vagy utána lehet megjeleníteni.

1.3.2. Az adatgyűjtés befejeződése után a bemeneti és kimeneti adatok elérhetők a Matlab munkaterében (`voltage` és `speed` sturktúrák).

A Simulink-ben Pulse Generator-t adtam bemenetre, 10-es amplitúdóval és 50%-os kitöltési tényezővel. Egy -5-ös konstans eltolást is be kellett kötnöm a -5 és +5 V-os bementi feszültség előállításához.

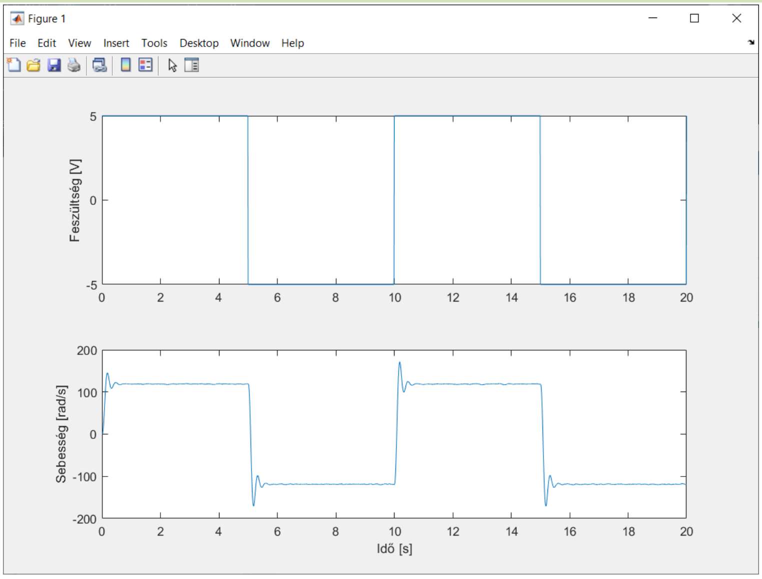


Az eredményül kapott ábra:

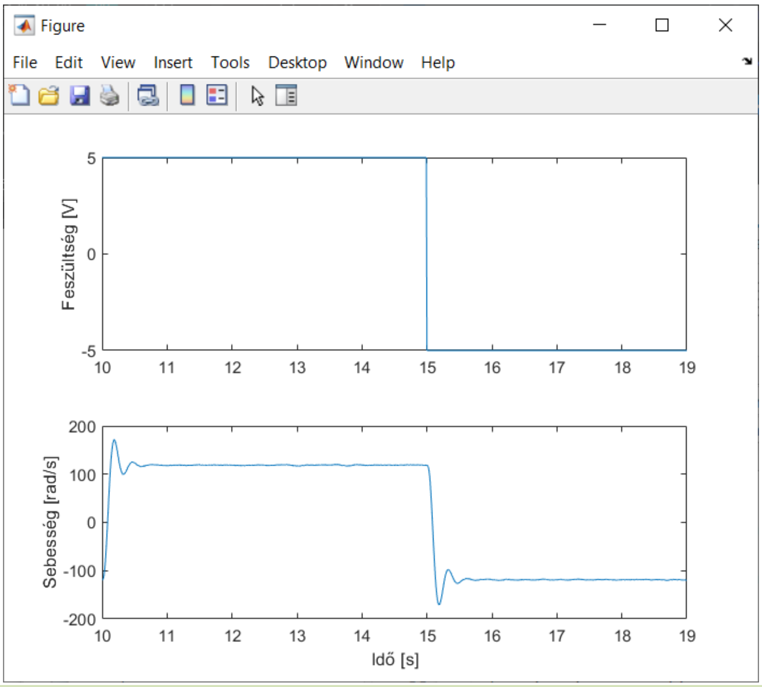


1.4. A szakasz identifikációja.

1.4.1. Az adatgyűjtés során keletkezett adatokat a Live Script megjeleníti. Válassza ki, hogy melyik időtartományhoz tartozó jeleket kívánja felhasználni az identifikációhoz (`T_kezdo`, `T_veg`).



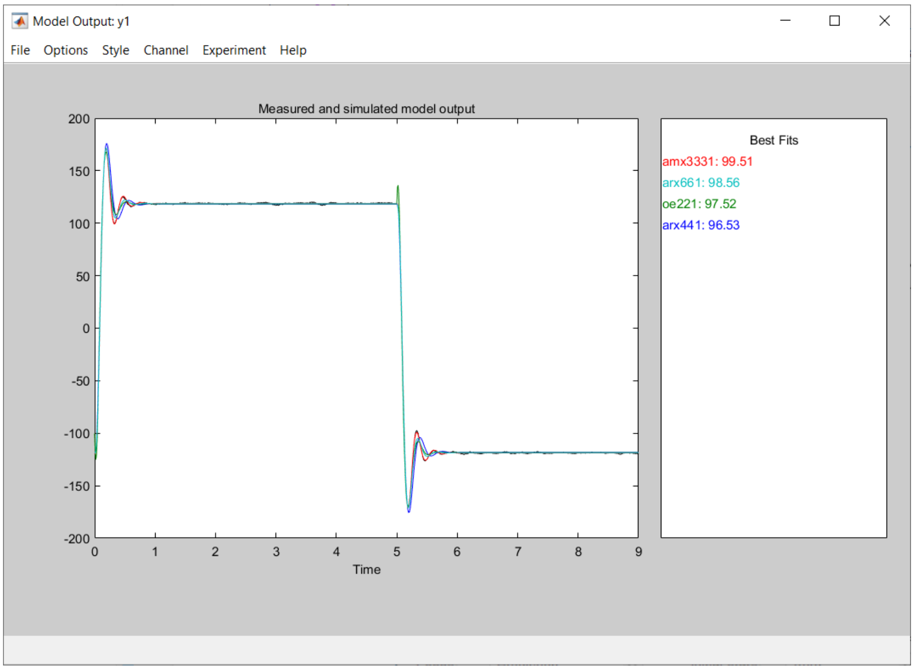
T_kezdo-nek 10-et, T_veg-nek pedig 19-et választottam az ábra alapján.



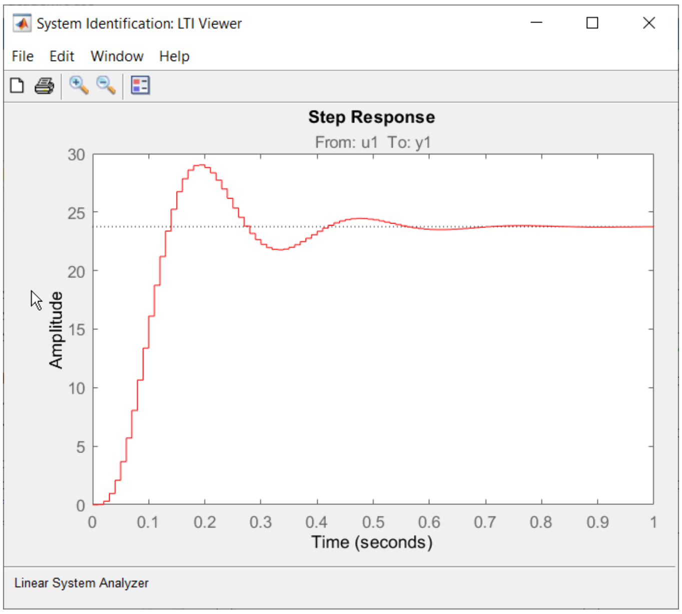
1.4.2. A szabályozott szakasz identifikációja a System Identification app segítségével. Adatok betöltése, polinomiális modellek illesztése (különböző rendszermodellekkel, fokszámokkal), modellek ellenőrzése, összehasonlítása. Ellenőrizendő, hogy az identifikált modellnek van-e folytonosidejű megfelelője. Legjobb, folytonosidejű megfelelővel is rendelkező modell kiválasztása, munkatérbe helyezése.

1.4.3. Az identifikáció eredményeként előálló diszkrét idejű modell folytonosidejű ekvivalensének meghatározása, pólusainak vizsgálata.

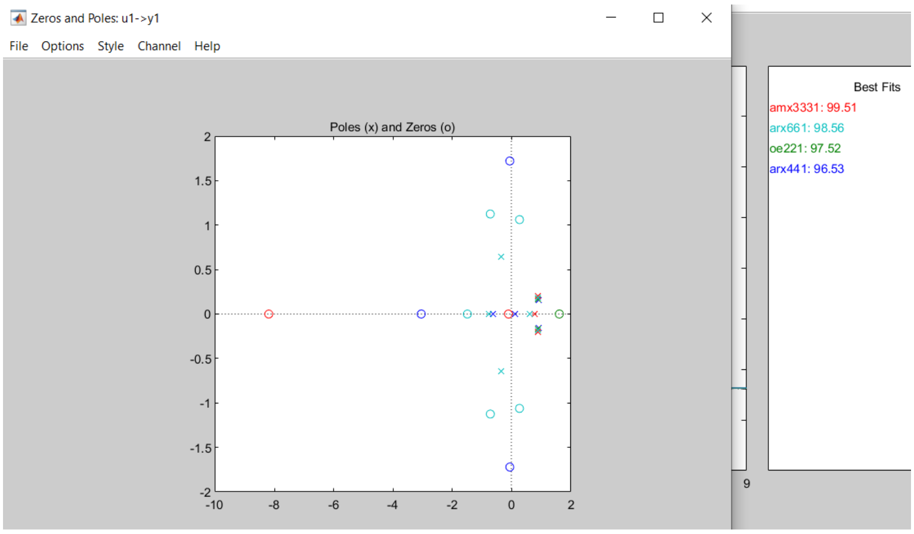
Az identifikációt ARX, ARMAX és OE modellekkel is elvégeztem, a legpontosabb eredményt az ARMAX [3 3 3 1] modell adott, 99.51%-kal.



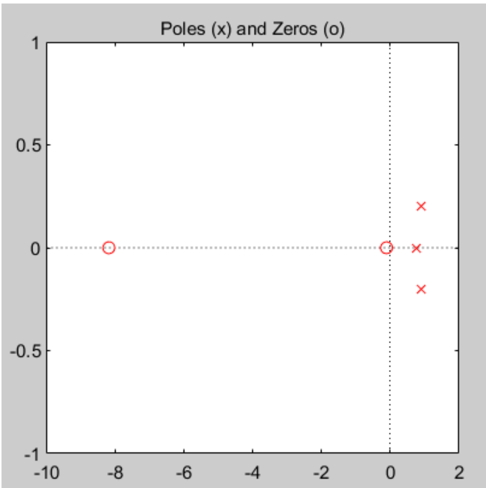
A tranziens viselkedés az ARMAX modellel:



A pólus zérus elrendezés kicsit kusza a 4 féle modell miatt, de látható hogy a pirossal jelzett ARMAX [3 3 3 1], a zöldell jelzett OE [2 2 1] az amelynek nincs negatív pólusa, így megfelelnek a további vizsgálathoz:



A pólus-zérus elrendezésből következtetve az ARMAX [3 3 3 1] modellt választottam a rendszerindetifikációhoz.



-> ARMAX [3 3 3 1] pólus-zérus elrendezése.

1.5. Állapottér-módszeren alapuló diszkrétidejű szabályozótervezés, szimuláció és valósidejű szabályozás.

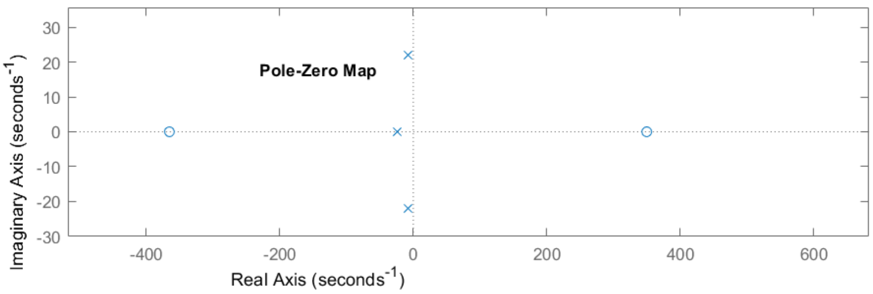
A kapott diszkrét modell:

```
identifikalt_modell =
Discrete-time ARMAX model: A(z)y(t) = B(z)u(t) + C(z)
  A(z) = 1 - 2.605 z^-1 + 2.298 z^-2 - 0.6827 z^-3

  B(z) = 0.02467 z^-1 + 0.2044 z^-2 + 0.01976 z^-3

  C(z) = 1 - 2.239 z^-1 + 1.742 z^-2 - 0.4595 z^-3

Name: amx3331
Sample time: 0.01 seconds

Parameterization:
   Polynomial orders:   na=3   nb=3   nc=3   nk=1
   Number of free coefficients: 9
   Use "polydata", "getpvec", "getcov" for parameters

Status:
Estimated using PEM on time domain data "mydata".
Fit to estimation data: 99.7% (prediction focus)
FPE: 0.133, MSE: 0.1295
```

A folytonos idejű modell:

```
From input "u1" to output "y1":
  -2.36 s^2 - 36.79 s + 3.015e05
  ------------------------------
s^3 + 38.18 s^2 + 873.2 s + 1.269e04
```

A pólus-zérus elrendezésben láthatjuk, hogy a pólusok átkerültek a bal félsíkra:



Terhelésbecslésen alapuló állapotteres szabályozó megtervezése. A hallgató a szabályozó-tervezést a következő lépésekben valósítja meg.

1.5.1. A szakasz irányíthatóságának, megfigyelhetőségének ellenőrzése. Csak akkor tudja megtervezni az állapotteres szabályozót, ha irányítható, megfigyelhető az identifikált szakaszmodell.

A megfelelő függvények (ctrb, obsv) alkalmazását követően a két eredmény mátrix:

```
Mc = 3x3
     0.5000    1.3025    2.2440
          0    1.0000    2.6050
          0         0    0.5000

Mo = 3x3
     0.0493    0.2044    0.0395
     0.5373   -0.0369    0.0337
     1.3258   -0.6006    0.3668
```

Az irányíthatóság és a megfigyelhetőség vizsgálatához a rank függvényt használtam, mindkét esetben 3 jött ki, amely jól láthatóan megegyezik, így a szakasz megfigyelhető és irányítható.

```
% Irányíthatósági mátrix
Mc =  ctrb(allapotter_d)
% Megfigyelhetőségi mátrix
Mo = obsv(allapotter_d)

rank(Mc)
rank(Mo)
```

```
ans = 3
ans = 3
```

1.5.2. A zártkörű viselkedés specifikációjának megválasztása. A tervezéshez a zárt rendszer és a megfigyelő karakterisztikus egyenletének gyökeit folytonos időben (s -tartományban) kell megválasztani és beállítani, majd az identifikáció során használt mintavételi idő mellett automatikusan megtörténik a konverzió diszkrét időbe (z -tartományba). (A Live Script a specifikált sajátértékek alapján automatikusan meghatározza az állapotvisszacsatolás, a terhelésbecslő és az alapjel miatti korrekció mátrixait.)

A zárt kör sajátértékei meghatározása után a terhelésbecslő sajátértékeit is meghatárzotam minf folytonos, mind diszkrét időben:

Zárt körű viselkedés specifikációja folytonos időben

Az előbb látta, hogy hol helyezkednek el a modell sajátértékei. Ezekből kiindulva határozza meg, hogy a zárt körben hova kerüljenek a sajátértékek. Gondolja végig, hogy szeretné-e, hogy a kimenetnek túllövése legyen. (Valós sajátértékek választása esetén nincs túllövés.) A szakasz sajátértékeihez képest milyen irányba kell elmozdítani a sajátértékeket a komplex számsíkon, ha gyorsítani szeretnénk a rendszert? Van-e valamilyen korlátja a gyorsításnak?

Feladat: Zárt kör sajátértékeinek választása folytonos időben.

A zárt körnek ugyanannyi sajátértéke van, mint a szakasznak. Ezt az értéket tartalmazza az n változó. Egy sajátérték lehet többszörös (akár n-szeres) is.

```
s=pole(allapotter_c)

s1 = s(1)
s2 = s(2)
s3 = s(3)
```

```
s = 3×1 complex
   -23.8712 + 0.0000i
    -7.1522 +21.9217i
    -7.1522 -21.9217i

s1 = -23.8712
s2 = -7.1522 + 21.9217i
s3 = -7.1522 - 21.9217i
```

Határozza meg az ennek (ezeknek) megfelelő diszkrét idejű pólusokat az $z_i = e^{s_i T_s}$ összefüggés alapján:

```
z1 = exp(s1*Ts)
z2 = exp(s2*Ts)
z3 = exp(s3*Ts)
```

```
z1 = 0.7876
z2 = 0.9087 + 0.2025i
z3 = 0.9087 - 0.2025i
```

A diszkrét idejű pólusokat egy z vektorba kell rendezni, melynek n eleme lesz:

```
z = [z1 z2 z3]
```

```
z = 1×3 complex
     0.7876 + 0.0000i   0.9087 + 0.2025i   0.9087 - 0.2025i
```

A fenti lépéseket most ismételje meg! A különbség csak annyi, hogy most a terhelésbecslő sajátértékeit kell meghatározni. Figyeljen arra, hogy a becslő gyorsabb legyen, mit a zárt kör! (Legalább háromszoros gyorsaságra törekedjen.)

Feladat: Terhelésbecslő sajátértékeinek választása folytonos időben.

```
so1 = 3*s1
so2 = 3*s2
so3 = 3*s3
zo1 = exp(so1*Ts) % sajátérték diszkrét időben
zo2 = exp(so2*Ts)
zo3 = exp(so3*Ts)
```

```
so1 = -71.6137
so2 = -21.4567 + 65.7651i
so3 = -21.4567 - 65.7651i
zo1 = 0.4886
zo2 = 0.6386 + 0.4932i
zo3 = 0.6386 - 0.4932i
```

Most is egy vektorba kell tenni a sajátértékeket. Most viszont a vektornak n+1 eleme lesz, hiszen az állapotokon kívül a terhelést is becsülni kell:

```
zo = [zo1 zo2 zo3 zo1]
```

```
zo = 1×4 complex
     0.4886 + 0.0000i   0.6386 + 0.4932i   0.6386 - 0.4932i   0.4886 + 0.0000i
```

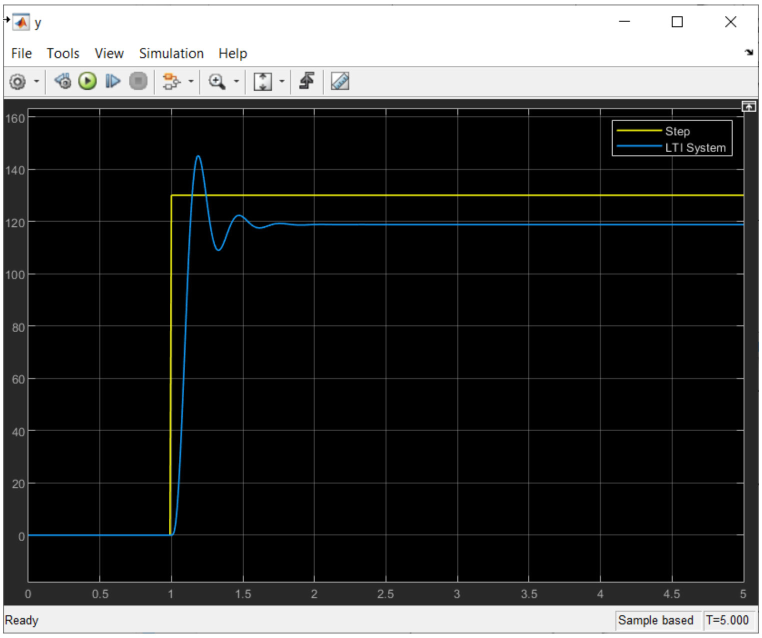
A z_0 vektor helyes megadását követően a szabályozó mátrixai is elkészültek:

```
K = 1×3
10^-14 ×
      0.3553   -0.2665    0.2220

Phih = 4×4
      2.6050   -1.1491    0.6827    0.5000
      2.0000         0         0         0
           0    0.5000         0         0
           0         0         0    1.0000

Gammah = 4×1
      0.5000
           0
           0
           0

Ch = 1×4
      0.0493    0.2044    0.0395         0

Ght = 1×4
      2.6560    2.9968    0.7274    0.3929

Gh = 4×1
      2.6560
      2.9968
      0.7274
      0.3929

Fh = 4×4
      1.1780   -1.0510    0.5932    0.4345
      0.3899    0.1107   -0.1009   -0.0739
     -0.3908    0.5269   -0.0245   -0.0179
     -0.2111    0.0145   -0.0132    0.9903

Hh = 4×1
      0.4345
     -0.0739
     -0.0179
     -0.0097

Nxu = 4×1
      2.0095
      4.0190
      2.0095
      0.0421

Nx = 3×1
      2.0095
      4.0190
      2.0095

Nu = 0.0421
```

1.5.3. A szabályozás szimulálása az identifikált diszkrét idejű modellen. Értékelje a szabályozás hatékonyságát és vizsgálja meg, hogy a szabályozás eleget tesz-e az előírt specifikációknak.

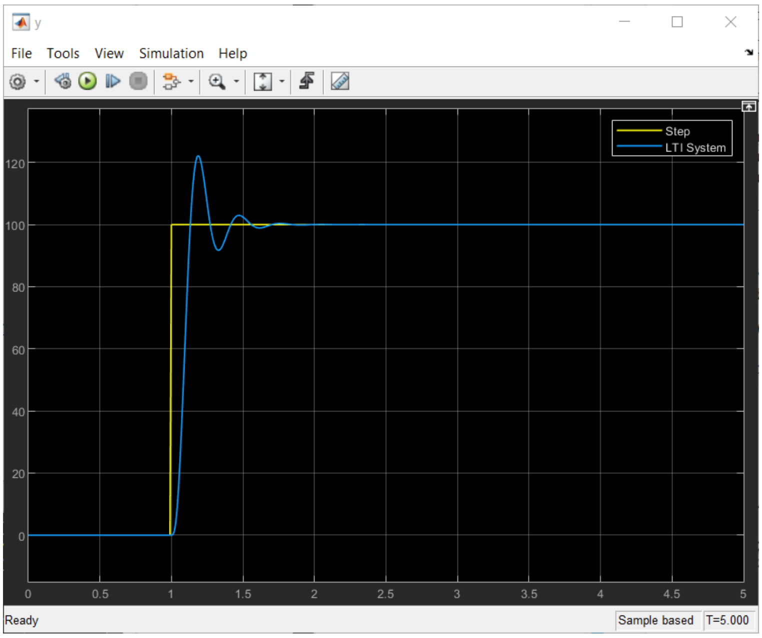
1.5.4. A szabályozás szimulálása az identifikáció eredményéből meghatározott folytonos idejű szakaszmodellel. Vizsgálja meg, hogy a szakasz bemenetén alkalmazott nulladrendű tartószervnek milyen hatása van a zártkörű viselkedésre.

Az első szimulációt követően ezt az eredményt kaptam:



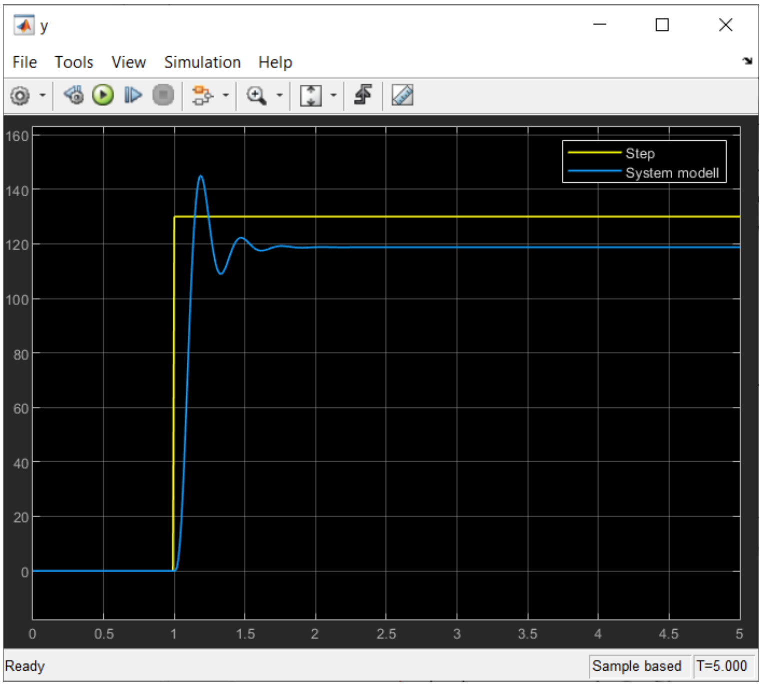
Az ábrán látható, hogy a 130 körüli alapjel túl nagy a modellezett rendszerünknek, így azt célszerű csökkentei.

100-ra lecsökkentve ezt kapjuk:

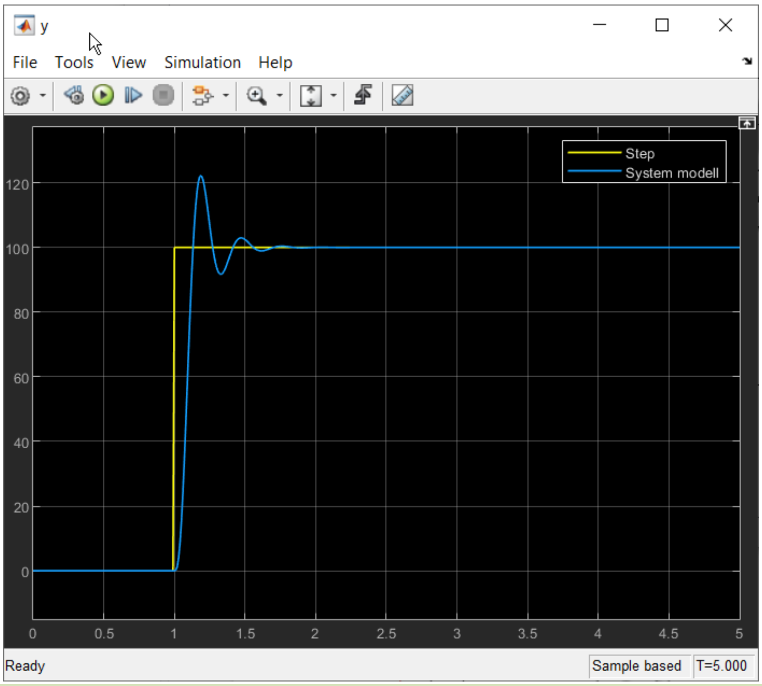


Ezt követően a „a szakasz bemenetén alkalmazott nulladrendű tartószervnek milyen hatása van a zártkörű viselkedésre” következett, amelynél az alapjel 130-as értékénel megint előjött, hogy nem tudja követni a rendszermodell, azonban 100-ra lecsökkentve azt már igen.

130-as alapjelű ábra:

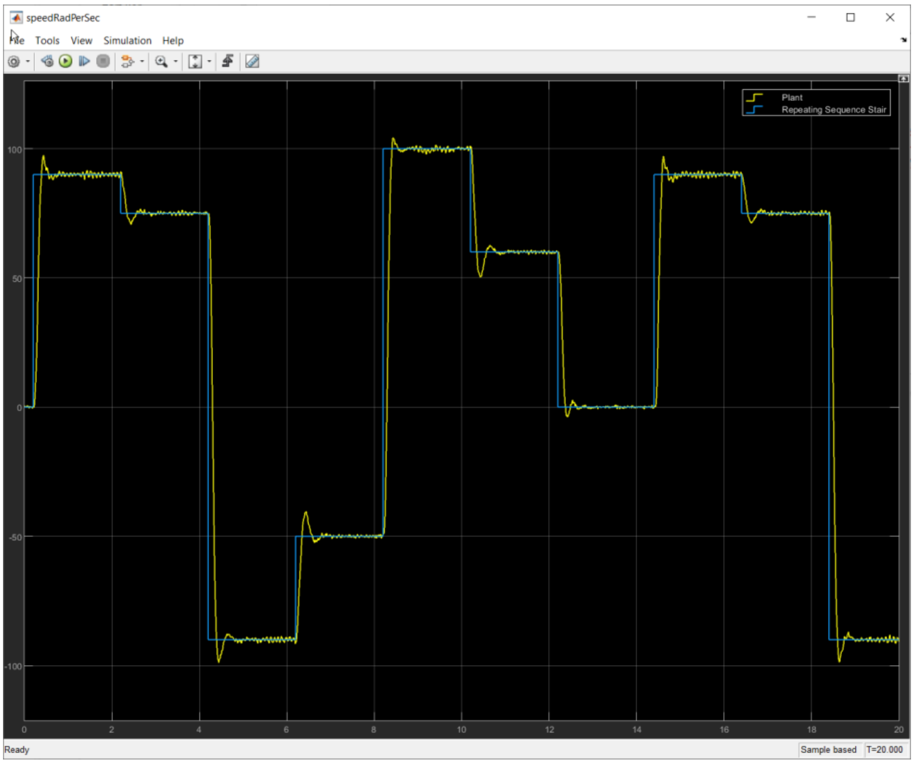


100-as alapjelű ábra:



1.5.5. A `szabalyozashoz.slx` Simulink modell Normal módú futtatásával a zártkörű szabályozás szimulációja a nemlinearitást is tartalmazó szakasz reprezentációval. Vizsgálja meg, mennyire mondható sikeresnek az identifikáción alapuló szabályozótervezés és szimuláció, ha a nemlinearitást is tartalmazó modellel alkalmazzuk a megtervezett szabályozót.

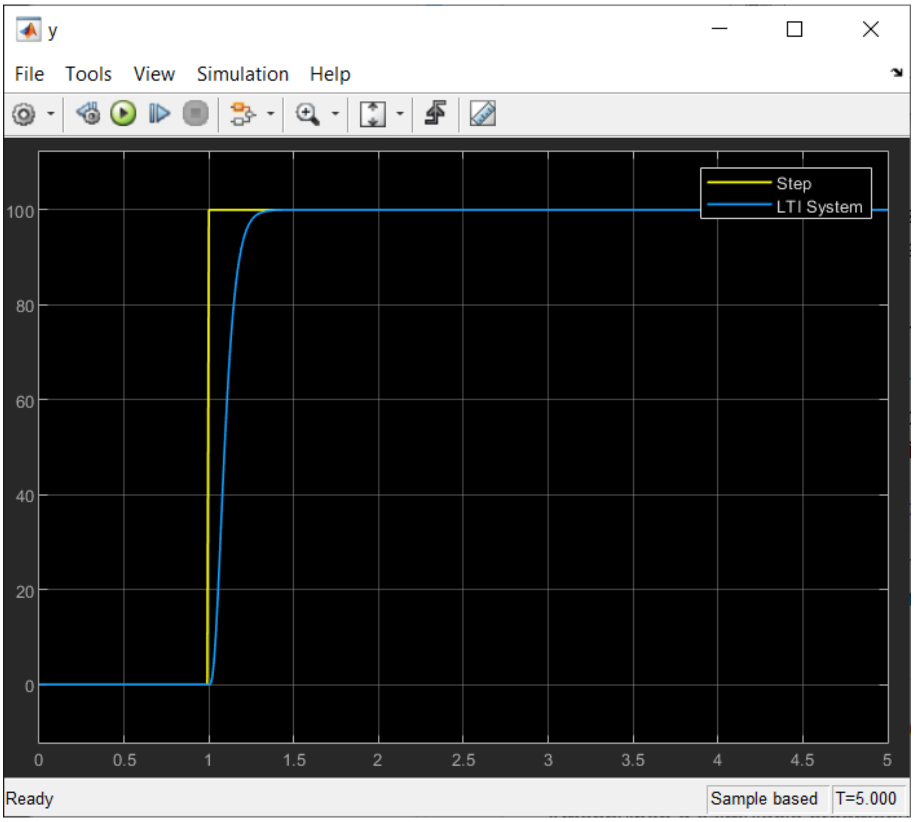
A szimuláció futtatását követen az alábbi eredményt kaptam:



Az alapjel követés kisebb túllengésekkel, de hibátlanul bekövetkezett.

1.5.6. Vizsgálja meg a szabályozás hatékonyságát különböző mértékű gyorsuló pólusáthelyezés és gyorsuló megfigyelő-specifikációk esetén. Eredményezheti-e a gyorsítás a szabályozó telítődését? Mennyire korlátozzák a specifikációk a real-time megvalósíthatóságot?

A pólusok áthelyezésével tudunk növelni a hatékonyságon, például -30-ra áthelyezett mindhárom pólus esetén:



Az áthelyezett pólusokkal kisebb feszültség kilengéseket is elérhetünk, így megvédve eszközeinket az esetleges problémáktól: