

Mérési Jegyzőkönyv

A mérés tárgya: Logikai vezérlők alkalmazástechnikája (11. mérés)

A mérést végzi: Veszelyi Bence Balázs, V3UWB0

Mérőcsoport: H12, 41

A mérés időpontja: 2021-03-08

Mérési feladatok

Válaszolja meg írásban kijelölt házi feladatának kérdéseit!

A:

PLC:

A PLC (Programmable Logic Controller) egy speciálisan gyártási folyamatok irányítására, szabályozására és ellenőrzésére kifejlesztett kisméretű számítógép. Fizikai felépítése, architektúrája és programozási környezete egyedileg fejlesztett, digitális és analóg ki- és bemenetekkel rendelkezik. Nevének legjobb magyar fordítása a programozható logikai vezérlő, általában közepes bonyolultságú folyamatok vezérlésére alkalmazzák, előnyei között sorolható a gyors és akár távoli programozhatóság, a kis méret és magas működési hőmérséklettartomány, a könnyű hibakeresés és az univerzális használat. Hátránya ugyanakkor a villamos energia szükséglet lehet (bár ez többretegű energia ellátórendszerrel szinte teljesen kiküszöbölhető) és ha mechanikus eszközt vezérel, akkor az átalakító tervezése bonyolult feladat lehet (elektromos-mechanikai).

PROFIBUS:

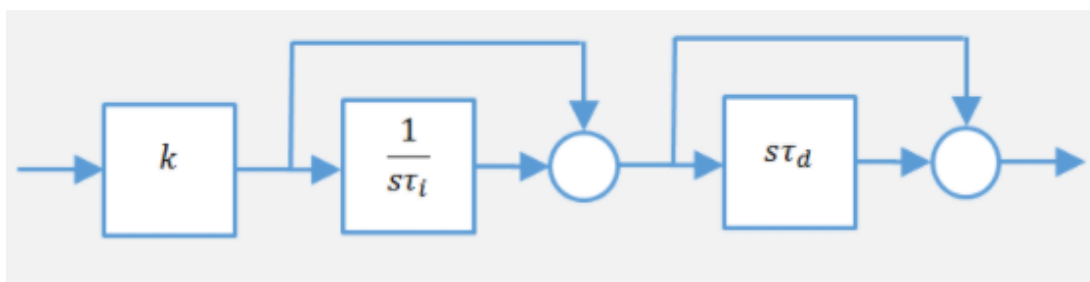
A PROFIBUS (Process Field Bus) egy nyílt ipari kommunikációs rendszer, amelyet 1989-ben fejlesztettek ki. Univerzálisra tervezték, így alkalmazási területe széleskörű. Speciális interfész nélkül is képes segítségével kommunikálni 2 vagy több eszköz, képes a nagysebességű adatátvitelre és nagy, bonyolult kommunikációk megvalósítására is. Felépítése hierarchikus, megkülönböztet master és slave eszközöket, előbbi vezeti a az adatkommunikációt a buszon, utóbbiak a perifériák. Három fő változata van: DP, PA, FMS. Utódja a PROFINET, amely már TCP/IP megoldáson alapul.

B:

PID szabályozók:

Három fő paraméterük van: A_p =Alapsávi erősítés, T_i =Integrálási idő, T_d =Deriválási idő.

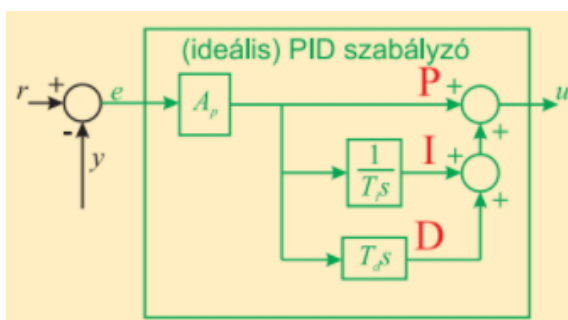
Soros relizációban:



$$\frac{U(s)}{E(s)} = K_P + \frac{K_I}{s} + K_D s$$

, ahol az $U(s)/E(s)$ Laplace transzformált jelek az átviteli függvényt ($W(s)$) adják, illetve $K_P = A_p$, $K_I = 1/T_i$ és $K_D = T_d$.

Párhuzamos realizációban:



Ideális D-hatást tartalmazó PID szabályzó átviteli függvénye.

$$W_{PID_k}(s) = A_p \left(1 + \frac{1}{sT_i} + \frac{sT_d}{sT_c + 1} \right) = \frac{A_p T_i (T_d + T_c) s^2 + (T_i + T_c) s + 1}{T_i s (sT_c + 1)}$$

Ha megelégszünk a közelítő D-hatással is, akkor a: tag $\frac{sT_d}{sT_c + 1}$ egy $T_d \cdot s$ -el helyettesíthető, úgy mint a párhuzamos elrendezés fenti ábráján látható.

Egy termikus modell paramétereinek meghatározása mérési adatok alapján

2. Egy termikus modell paramétereinek meghatározása mérési adatok alapján (Adja meg házi feladatának kódját: C5)

a, Röviden magyarázza el a megoldás során alkalmazott elveket!

A termikus folyamat egytárolós taggal jól modellezhető, amely átviteli függvénye:

$$P(s) = \frac{A}{1+sT} = \frac{k}{s+p}, \text{ ahol } A \text{ az erősítés, } T \text{ pedig az időállandó.}$$

Az első lépése az erősítés meghatározása lesz, ehhez a a fűtési folyamat statikus karakterisztikáját használom:

$$A = \frac{\Delta \vartheta}{\Delta u}$$

, itt Δu a gerjesztés változása, $\Delta \vartheta$ az állandósult állapotbeli hőmérséklet ugrása.

A mérési útmutató alapján T időállandó A ismeretében kiszámolható, ugyanis az ugrásválaszt egytárolós tag esetén:

$$v(t) = A \left(1 - e^{-\frac{t}{T}} \right)$$

$v(t=T)=0.63 \cdot A$, ebből T meghatározható.

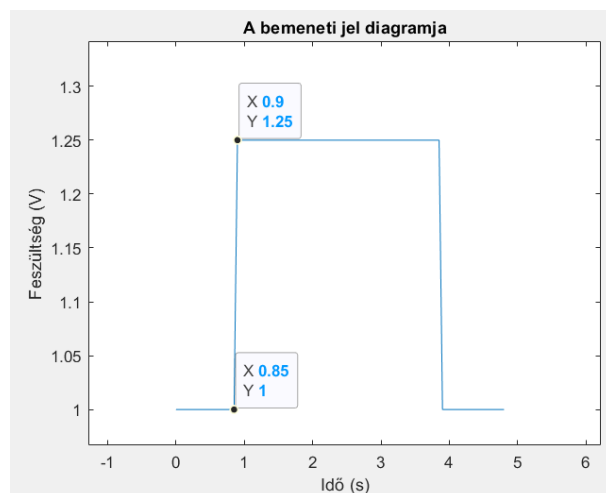
b, A mérési útmutatóban leírtak alkalmazásával határozza meg az egytárolós folyamat paramétereit, amelyen végzett mérések adatait a MeresiAdatxx.m fájlban találja, ahol xx egyben házi feladata kódja is! (Lásd fentebb!) A megoldáshoz csatolja a programkódot és az elkészített ábrákat is! A feladat MATLAB-ban és Excel-ben is megoldható!

Az adatokat MATLAB-ba importálást követően szétbontottam t,u és y adatvektorokra. Ezek segítségével dolgoztam a továbbiakban.

MATLAB kód:

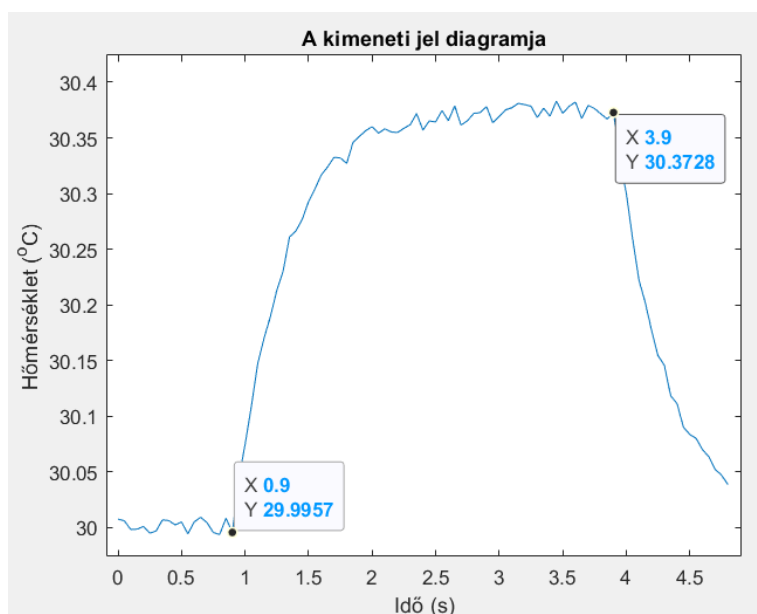
```
t=MeresiAdatC5(:,1);
u=MeresiAdatC5(:,2);
y=MeresiAdatC5(:,3);
figure(1)
plot(t,y);
title('A kimeneti jel diagramja');
xlabel('Idő (t)');
ylabel('Hőmérséklet (^oC)');
figure(2)
plot(t,u);
title('A bemeneti jel diagramja');
xlabel('Idő (t)');
ylabel('Feszültség (V)');
```

A kapott adatok alapján a bemeneti jel:



, mintavételi idő leolvasható: 0.05 s.

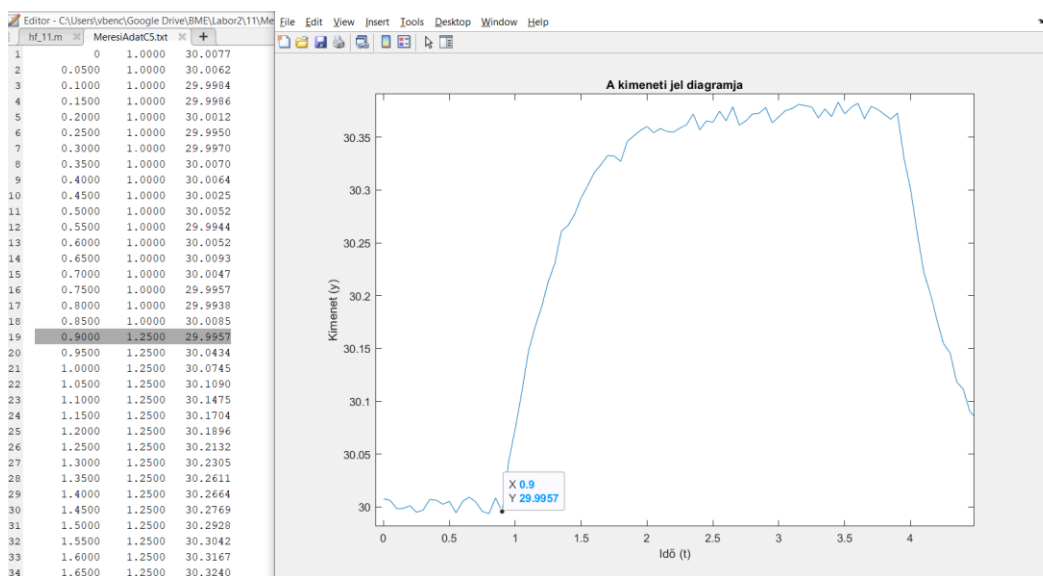
És a kimeneti jel:



Leolvasható, hogy 0.9 másodpercig a hőmérséklet körülbelül 20 °C-n van, majd nőtt 1 másodpercig és ott 30.3728 °C-n volt, majd újra csökkenni kezdett.

c, Adj meg a régi és az új munkaponti értékeket! (Az adatok a MATLAB felületre az Útmutatóban található mintaprogram lefuttatásával vihetők be.) Az adatok megadása során ne felejtkezzen el a mértékegységekről sem!

A régi munkaponti adatok meghatározásához megkerestem az $X=0.9$ -hez tartozó értéket a mérési adattömbben és így az első 19 adatból kiszámítottam a régi munkapontot:



MATLAB kód:

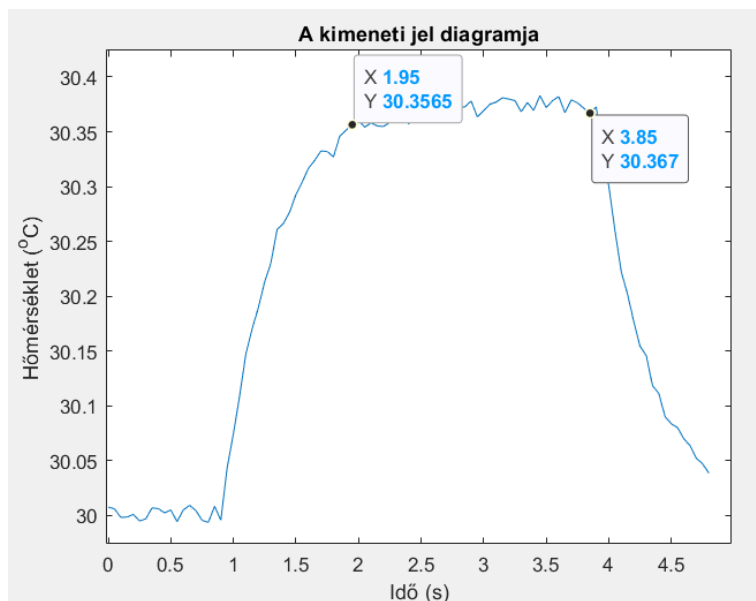
```
y0_regi=mean(y(1:19));
u0_regi=u(1);
```

Eredmény:

$u0_regi = 1 \text{ V}$

$y0_regi = 30.0017 \text{ °C}$

Az új munkapont kiszámítása hasonlóan történt, a lenti ábrán található 2 érték között számolt munkapont (az állandósult állapotot 1.95 másodperctől tekintetem egészen 3.85 másodpercig):



MATLAB kód:

```
y0_uj=mean(y(40:78));
u0_uj=u(20);
```

Eredmény:

u0_regi = 1.25 V

y0_regi = 30.3693 °C

d, Adja meg a meghatározott folyamat átviteli függvényét! Adja meg folyamat időállandóját és erősítését (dimenzióval)!

Az átviteli függvény meghatározása az alábbi kóddal történt:

```
du=u0_uj-u0_regi;
dy=y0_uj-y0_regi;
A=dy/du;
y_63_A=0.63*dy+y0_regi;
T=t(27)-0.9;

P=tf(A, [T 1]);
```

A kapott értékek és az átviteli függvény:

A = 1.4704 °C/V

T = 0.4 s

$$P = \frac{1.47}{0.4 s + 1}$$

Az időállandót az a részben írtak alapján határoztam meg, a $t=1.3$ s időpillanatnál volt (27-ik adatminta) a körülbelül 63%-s erősítés, ebből a felfutás kezdeti időpontját (0.9 s) kivonva kaptam meg.



A későbbi szimulációhoz elkészítettem egy alap ARX-es modell-t is az adathalmazra, így össze tudom hasonlítani a kettőt majd:

```
Ts=t(2)-t(1);
y0_arx=mean(y);
u0_arx=mean(u);

ym=y0_arx-y;
um=u0_arx-u;
data=[ym um];
arx_model=arx(data, [1, 1, 1]);
[num,den]=th2tf(arx_model);
P_arx=d2c(tf(num,den,Ts));
```

$$\text{Az így kapott modell átviteli függvénye: } P_{\text{arx}} = \frac{3.763}{s + 2.503}$$

e, Ellenőrizze a folyamat viselkedését a MATLAB step utasítással! Szimulálja a kapott folytonos folyamat viselkedését a fájlban megadott bemenőjel esetén is (pl.: lsim utasítás)! Ugyanabban az ábrában ábrázolja, majd hasonlítsa össze a mért és a szimulált kimeneti jelet! Excel-ben használja az átmeneti függvény inverz Laplace transzformáltját!

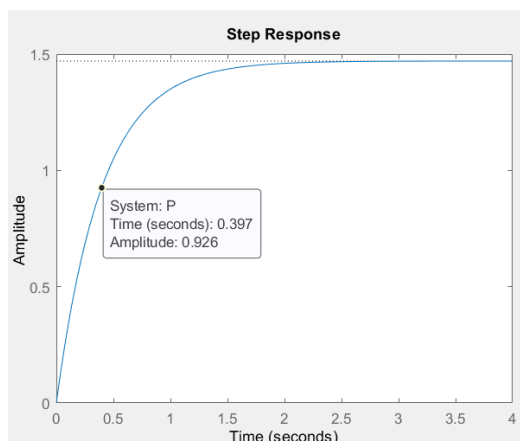
Az ábrázoláshoz és szimulációhoz használt MATLAB kód:

```
figure(1)
step(P);

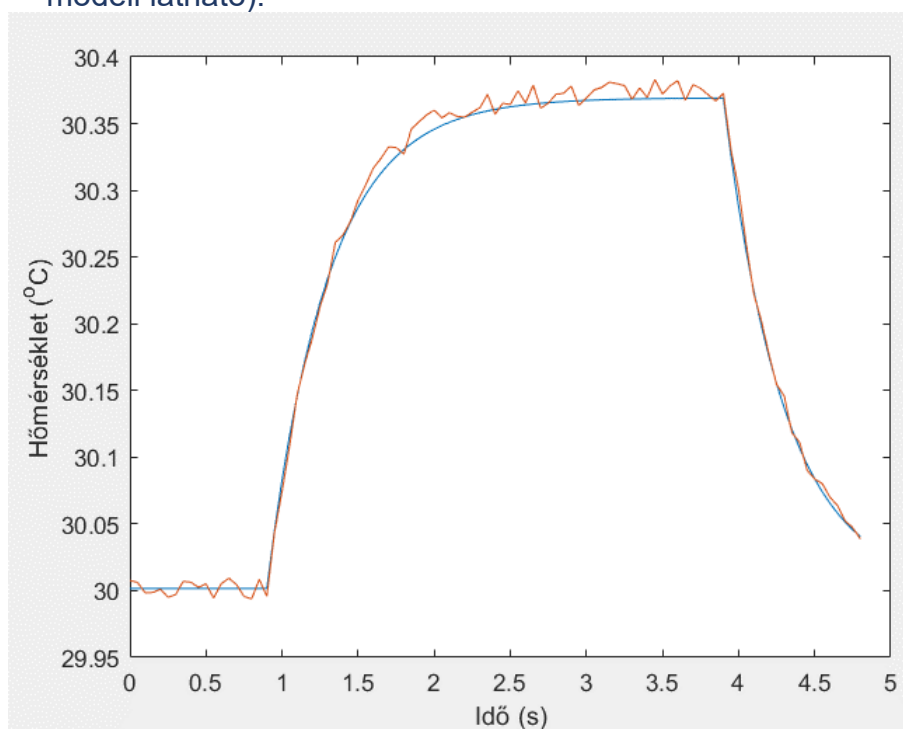
y_sim=lsim(P,u-u0_regi,t)+y0_regi;

figure(2)
plot(t, y_sim, t, y);
xlabel('Idő (t)');
ylabel('Kimenet (y)');
```

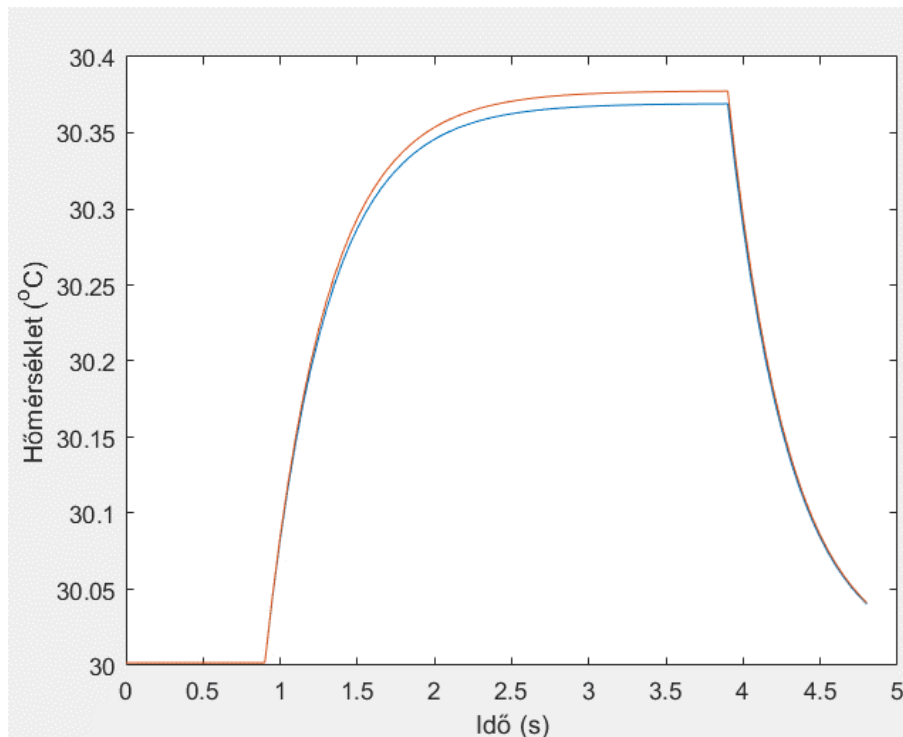
Az ugrásválaszra kapott ábra (bejelölve az amplitúdó 63%-os pontja):



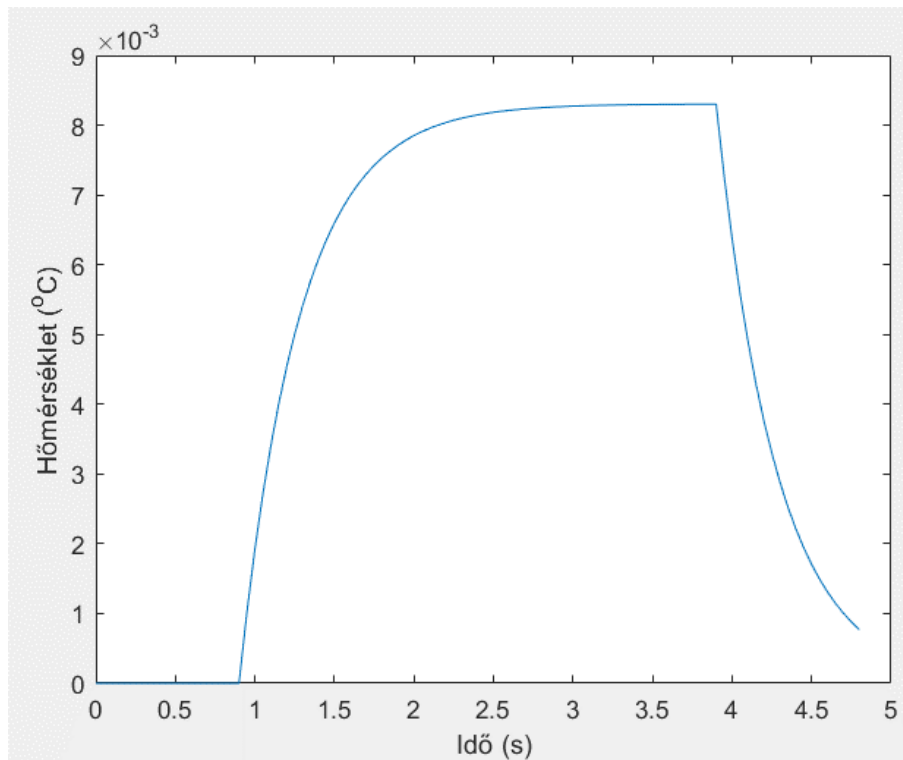
A kimenet viselkedésére kapott szimulációs ábra (késsel az általunk illesztett modell látható):



Jól láthatóan elég pontos modellt sikerült alkotni, nézzük meg mi a helyzet az ARX-es alapmodellünk esetén (az ábra a két szimulált rendszert mutatja, narancssal az ARX modell látható):



Alig van különbség a két modell között, ha a különbségüket rajzoltatjuk ki, akkor láthatjuk hogy a kiemeneteken 10^{-3} nagyságrendű eltérés van az ARX és az általunk számított egytárolós modell között.



A használt plot-ok sorrendben:

```
plot(t, y_sim, t, y);
plot(t, y_sim, t, y_sim_arx);
plot(t, y_sim_arx-y_sim);
```

f, Határozza meg a mért és szimulált jel eltérések átlagát és szórását! Vigyázzon a definíciók pontos implementálására.

Az eredeti és a modellezett kimenet különbségeinek számítására alkalmazott MATLAB kód:

```
dy=y-y_sim;
y_mean=mean(dy);
y_var=sqrt(var(dy));
```

Eredmények:

Átlag = 0.0035

Szórás = 0.006

3. A PLC használatának megismerése a letölthető videó alapján

Nézze meg a https://bmeedu-my.sharepoint.com/:v/g/personal/david_kiss_edu_bme_hu/ER2ehhmv8EFCjR XuV3Qt c y l B u I7Xp2l sy x z l R Q 43 l o Q M 4 g ? e = X j d 7 B D

található videót és válaszoljon a következő kérdésekre!

a, Mi a PLC és mire lehet használni? (1 pont)

A PLC (Programmable Logic Controller) egy speciálisan gyártási folyamatok irányítására, szabályozására és ellenőrzésére kifejlesztett kisméretű számítógép. Fizikai felépítése, architektúrája és programozási környezete egyedileg fejlesztett, digitális és analóg ki- és bemenetekkel rendelkezik. Nevének legjobb magyar fordítása a programozható logikai vezérlő, általában közepes bonyolultságú folyamatok vezérlésére alkalmazzák.

b, Mit jelent az, hogy a PLC programfeldolgozása ciklikus működésű? Ismertesse a PLC ciklikus programvezérlésének a lépéseit! (1 pont)

A programfeldolgozás ciklikus működése azt jelenti, hogy a PLC-n futó operációs rendszer időnként végrehajtja a programciklus lépéseit. Ez az idő 100 ms, ami az alapbeállítás.

PLC programvezérlésének lépései:

1. Ciklus indítása startup blokkal (korábbi változók törlődnek).
2. Ciklusidő-figyelés újraindítása.
3. A kimeneti értékek táblájának kiírása a kimeneti kártyákra.
4. A bemeneti jelek állapotának lekérdezése és a bemeneti értékek táblájának aktualizálása.
5. Az összes engedélyezett felhasználói program végrehajtása a benne lévő utasításokkal (OBI blokkok hívása).

c, Milyen jelszintűek a PLC analóg ki- és bemenetei és ezek milyen funkciót látnak el a mérési elrendezésben? (1 pont)

A PLC analóg ki- és bemenetek 0-10 V-os jelszintűek.

A bemenetek a teljesítménytranszisztor, illetve a hűtőtönk hőmérsékletét mérik, míg a kimenet a fűtőteljesítmény vezérléséért felelős.

d, Mivel mérjük a hűtőtönk hőmérsékletét? Hogyan tudjuk növelni, illetve csökkenteni a hűtőtönk hőmérsékletét? (1 pont)

A hűtőtönk hőmérsékletét egy termikus érzékelővel mérjük, a hőmérsékletet a teljesítménytranszisztor áramának változtatásával növelhetjük, csökkenthetjük a hűtőventilátor sebességének szabályozásával. A termikus folyamat igen lassú, annak érdekében, hogy ezt felgyorsítsuk kényszerhűtést alkalmazunk.

Ha a hőmérséklet 80°C fölé emelkedik, akkor egy SR flipflop letiltja a fűtést.

e, Milyen funkciókat valósít meg ebben a mérési elrendezésben a TIA Portal? Sorolja fel az elérhető funkcióit! (1 pont)

A TIA Portal segítségével programozható a PLC (programfejlesztés, programok betöltése a PLC-be, szimuláció), továbbá itt található a WinCC kezelőfelület,

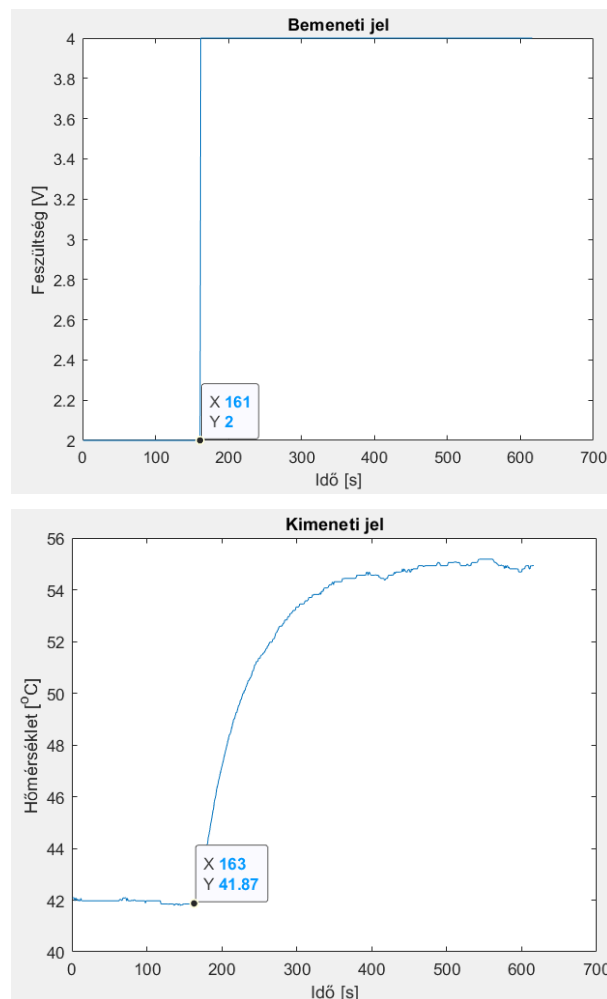
amely HMI-t (Human-Machine Interface) valósítható meg. A mérés során SCL nyelven írunk egy programot az FB10-es blokkba, TIA-n belül.

4. Valós mérési adatok felhasználásával számítsa ki érintő módszer segítségével a 3. feladatban vizsgált termikus folyamat paramétereit!

A folyamat mérési adatait a http://home.mit.bme.hu/~orosz/PLC_tavokt/ könyvtárból töltheti le. Az ön által letöltendő file nevének értelmezése: ha ön a K21 kurzus 3. csoportjának 2. hallgatója, akkor önnek a PLCmeresK32.csv file-t kell letöltenie. (Kurzusok: Hnn, Knn, Snn, Cnn, ahol nn egy kétjegyű decimális szám, csoportok: 0-tól 9-ig, hallgató: 1 vagy 2) Csak a helyesen letöltött adatok eredményeit fogadjuk el. **Ha ön egy másik hallgató adatait használja fel, akkor megoldása nem fogadható el.** Hasonló jelöléseket alkalmazva töltheti le innen az 5.c-hez tartozó feladatát is (A fenti példát alkalmazva a K32.txt file-t).

A mérési adatok felhasználásával határozza meg a termikus rendszer paramétereit!

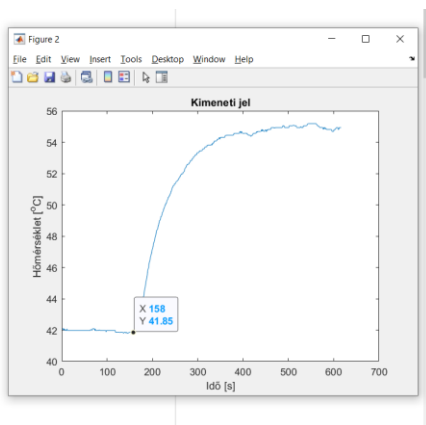
A PLCmeresH41.csv fájlt töltöttem le, az adathalmaz importálását követően szétbontottam a megfelelő adatvektorokra és megjelenítettem a be- és kimeneti jelt:



A régi és új munkapontok meghatározása:

A régi munkaponti adatok meghatározásához megkerestem az X=158-hoz tartozó értéket a mérési adattömbben és így az első 158 adatból kiszámítottam a régi munkapontot:

142	142.00	41.80	2.00	2.00	0.00
143	143.00	41.85	2.00	2.00	0.00
144	144.00	41.84	2.00	2.00	0.00
145	145.00	41.79	2.00	2.00	0.00
146	146.00	41.82	2.00	2.00	0.00
147	147.00	41.80	2.00	2.00	0.00
148	148.00	41.85	2.00	2.00	0.00
149	149.00	41.84	2.00	2.00	0.00
150	150.00	41.85	2.00	2.00	0.00
151	151.00	41.85	2.00	2.00	0.00
152	152.00	41.85	2.00	2.00	0.00
153	153.00	41.85	2.00	2.00	0.00
154	154.00	41.85	2.00	2.00	0.00
155	155.00	41.85	2.00	2.00	0.00
156	156.00	41.85	2.00	2.00	0.00
157	157.00	41.85	2.00	2.00	0.00
158	158.00	41.85	2.00	2.00	0.00
159	159.00	41.85	2.00	2.00	0.00
160	160.00	41.85	2.00	2.00	0.00
161	161.00	41.85	2.00	2.00	0.00
162	162.00	41.88	4.00	4.00	0.00
163	163.00	41.87	4.00	4.00	0.00
164	164.00	41.97	4.00	4.00	0.00
165	165.00	41.97	4.00	4.00	0.00
166	166.00	42.08	4.00	4.00	0.00
167	167.00	42.09	4.00	4.00	0.00
168	168.00	42.22	4.00	4.00	0.00
169	169.00	42.34	4.00	4.00	0.00
170	170.00	42.46	4.00	4.00	0.00
171	171.00	42.59	4.00	4.00	0.00



MATLAB kód:

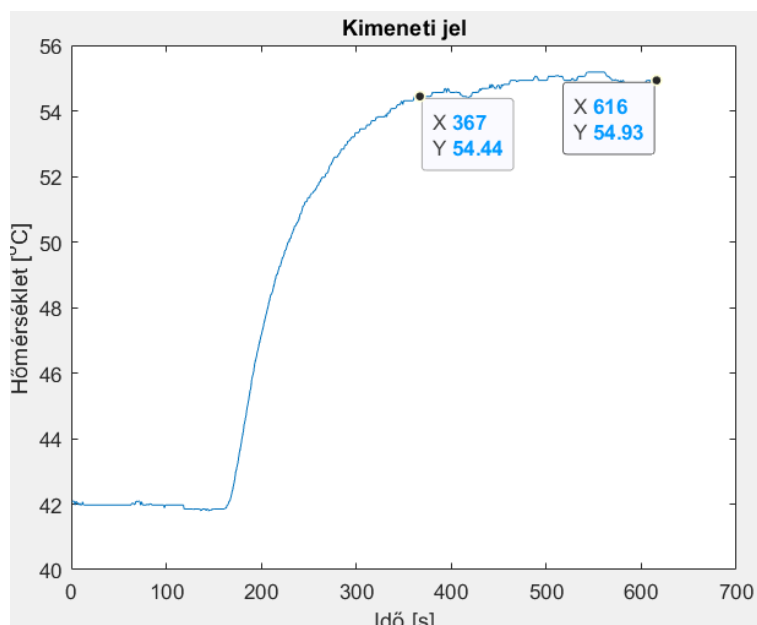
```
u0_regi= u(1);
y0_regi= mean(y(1:158));
```

Az eredmény:

u0_regi = 2 V

y0_regi = 41.948 °C

Az új munkapont kiszámítása hasonlóan történt, a lenti ábrán található 2 érték között számolt munkapont (az állandósult állapotot 367 másodperctől tekintetem egészen 616 másodpercig):



MATLAB kód:

```
u0_uj = u(367);
y0_uj=mean(y(367:616));
```

Az eredmények:

u0_uj = 4 V

y0_uj = 54.8248 °C

a, Írja fel a rendszer átviteli függvényét időállandós és zérus-pólus alakban! Egy közös ábrában hasonlítsa össze a mért és a számított rendszer viselkedését egységugrás esetén! (4 pont)

Az átviteli függvény meghatározása az alábbi kóddal történt:

```
du=u0_uj-u0_regi;
dy=y0_uj-y0_regi;
A=dy/du;
y_63_A=0.63*dy+y0_regi;
T=t(230)-161;

P=tf(A, [T 1]);
```

Az időállandót az a részben írtak alapján határoztam meg, a $t=230$ s időpillanatnál volt a körülbelül 63%-s erősítés, ebből a felfutás kezdeti időpontját (161 s) kivonva kaptam meg.

A kapott értékek és az átviteli függvény:

A = 6.4384 °C/V

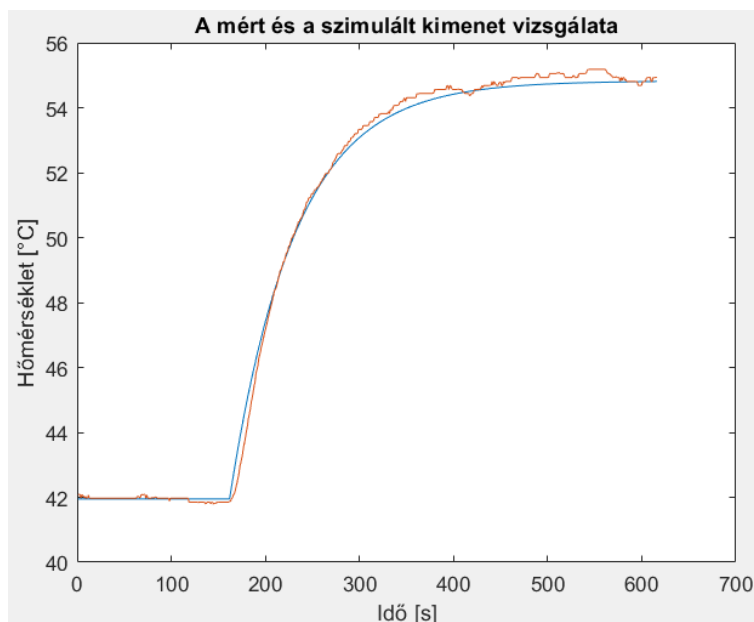
T = 69 s

P = $\frac{6.438}{69s + 1}$

A zérus-pólus alak: $P_{zk} = \frac{0.09331}{(s+0.01449)}$

A mért és szimulált rendszer összehasonlítása és ábrázolása az alábbi kóddal történt:

```
y_sim=lsim(P,u-u0_regi,t)+y0_regi;
plot(t, y_sim, t, y);
title('A mért és a szimulált kimenet vizsgálata');
xlabel('Idő [s]');
ylabel('Hőmérséklet [°C]')
```

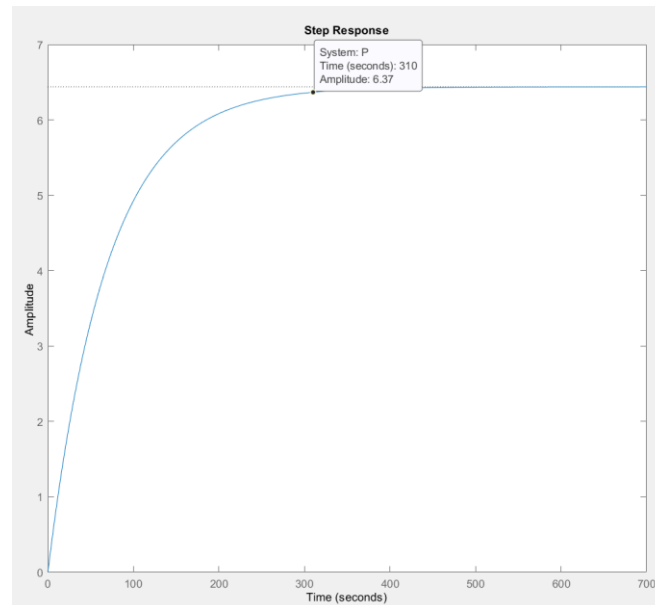


Jól láthatóan elég pontos modellt sikerült alkotni.

b, Az átviteli függvény ismeretében határozza meg, hogy a mérési adatok vajon elegendő mintavételezési pontot tartalmaznak-e a folyamat átviteli függvényének pontos meghatározásához? Legalább hány mintavételezési pontot kell mérni ahhoz, hogy az átviteli függvényt 1%-os pontosságon belül lehessen meghatározni? Válaszát indokolja! (5 pont)

Az ugrásválasz alapján indultam ki, az 1%-os hibahatár az A 99%-os értékénél található, ez a 310-ik mintánál van (ábrán bejelölve). Így ha legalább 310 mennyiségű mintát veszünk, akkor már elég pontos (1%-ob belül) lesz az átviteli függvény közelítése.

0.99* A = 6.37 °C/V



c, A mért folyamat valójában nem egytárolós rendszer. MATLAB segítségével határozza meg az egytárolós modell mellett a kéttárolós modell paramétereit is. Melyik modell közelíti pontosabban a mért adatokat? (1 pont)

A kéttárolós modell paramétereit egy egyszerű (2,1,1-es polinom fokú) ARX modellel készítettem el, az alábbi kóddal:

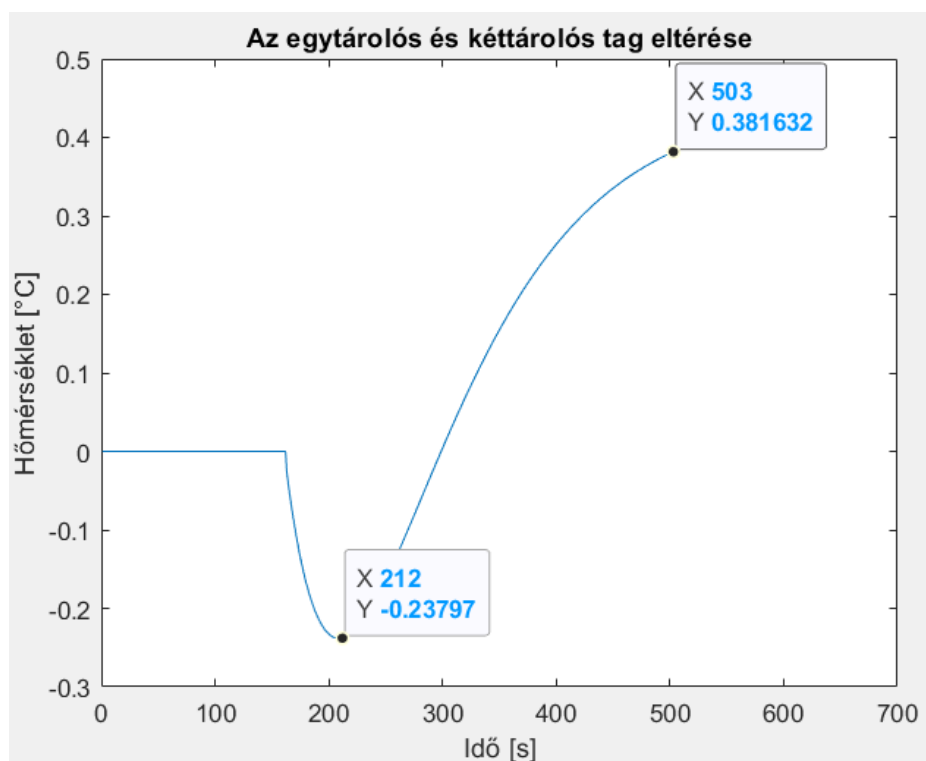
```
Ts=t(2)-t(1);
y0_arx=mean(y);
u0_arx=mean(u);
ym=y0_arx-y;
um=u0_arx-u;
data=[ym um];
arx_model=arx(data, [2, 1, 1]);
[num,den]=th2tf(arx_model);
P_arx=d2c(tf(num,den,Ts));
```

A kapott átviteli függvény:

```
P_arx =

    0.06829 s + 0.2217
-----
s^2 + 2.55 s + 0.03329
```

És az összehasonlításhoz kivontam a szimulált jeleket egymásból, az alábbi ábrán jól látszik, hogy az először létrehozott egytárolós tag és most elkészített a kéttárolós tag között a különbség változik, de az átlaghoz képest is maximum 1%-os (~0.4 °C-os).



Arra a kérdésre, hogy melyik model közelíti jobban a mért adatokat a válasz kicsivel, de a kéttárolós tag. Minden kimeneti pontra megnéztem az eltérések átlagát és a szórását, az eredmények:

Egytárolós eset adatainak szórása: 0.2304

Kéttárolós eset adatainak szórása: 0.219

Egytárolós eset adateltérések átlaga: 0.0693

Kéttárolós eset adateltérések átlaga: 0.0516

5. Az identifikált rendszer megismerése, tanulmányozása

a, Mekkora egyszeri kézi beavatkozó jelet kell beállítani a rendszer bemenetén (U), ha egy általunk megadott hőmérsékleti értékre (T) akarunk beállni? (2 pont)

A rendszer bemenete és kimenete közötti összefüggés:

$$T = A * (U - U_0) + T_0$$

Ebből U kifejezhető és kiszámolható a munkaponti adatokkal, eredmény:

$$U = 2 + \frac{T - 42.09}{6.4384}, \text{ itt } T \text{ az általunk megadott hőmérséklet. Példaként az utolsó}$$

értékre (T=54.93) behelyettesítve adathalmazban található értékkel.

u_kezi_beav =

3.9943

jön ki, ami kerekítve 4, így egyezik az

b, Számítsa ki, hogy a folyamat mennyi idő alatt áll be 1°C hibával! (3 pont)

Az a feladatnál felhasznált egyenletet kiegészítve:

$$T - 1\text{ °C} = A * (U - U_0) * (1 - e^{-\frac{t}{\tau}}) + T_0$$

Ebből t kifejezhető, τ = időállandó (69 s), illetve a többi érték ismeretében. $T = 54.93$ esetén a $t = 174$ s-re adódik.

c, Válaszolja meg és/vagy készítse el azt a PLC programot, amelyet a mérés idejében az MSTeams képernyőről tölthet le. A feladatban szereplő ciklusok megvalósításához nem használhat sem más időzítést, sem pedig (for, while, repeat) ciklus szervező utasításokat, csak azt, ami a modul periódikus meghívásával megoldható. Az időalap a megtekintett videóban szerepel. Kérjük a megoldásában tüntesse fel a feladata szövegét is. (5 pont)

Időalap: 100 ms

Feladat:

Labor2, 11. PLC mérés

5c. feladat: Írjon programot, ami a LED-soron jobbra-balra futó ping pong mintázatot valósít meg.

Ping pong: Mindig egy LED világít az alábbi ábra szerint. A LED soron a magasabb helyiérték jobbra helyezkedik el.



A program minden sorához részletes megjegyzést tegyen, hogy meg lehessen érteni, mit és hogyan akart megírni. Erre azért van szükség, mert nem elérhető a fejlesztőrendszer, így nem tudja sem szintaktikailag, sem szemantikailag ellenőrizni a munkáját.

Kezdeti érték adás:

#Par3 = 0 //LED bitenkénti számát jelenti (pl.: Q0.4, itt #Par1 = 4)

#Par2 = 7 //A visszafutáshoz szükséges bitszámokat jelölő változó

Program:

```
IF #Par3 = 7 THEN           // Ebben az elágazásban indul a visszafele fény sor
"OUTPUT" := Q0.(#Par2)
#Par2 := #Par2 - 1          // #Par2 fokozatos csökkentése
ELSE_IF #Par3 < 7 THEN     // Ez az előre fele futó fénycsík, minden lefutás után növelve
"OUTPUT" := Q0.(#Par3)
#Par3 := #Par3 + 1         // #Par1 fokozatos növelése
END_IF
```