# 14. Tétel

Szabályozók tervezése (arányos és integráló folyamatra), szabályozók programozása. Folytonos idejű és diszkrét idejű szabályozók, lökésmentes indítás

## Táblázatos Kísérleti módszerek

Pontos szabályozó tervezése a folyamat átviteli függvényének ismeretében történhet 🡪 ez nem mindig áll rendelkezésre 🡪kísérleti módszerekre van szükség

### Ziegler-Nichols módszer

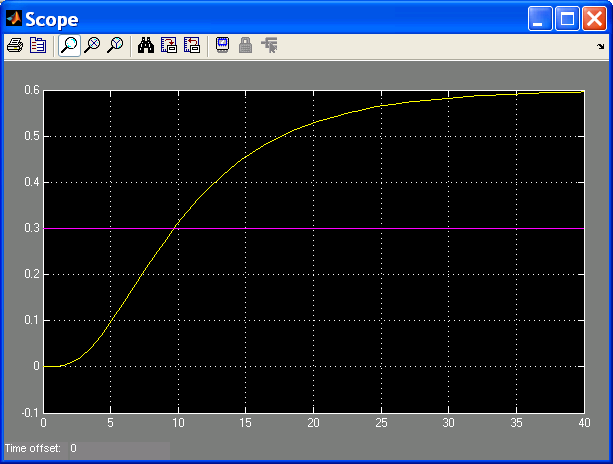
* A szabályozást arányos szabályozóval be kell lengetni
* A hurokerősítés változtatásával meg kell keresni az állandósult lengéshez tartozó kritikus körerősítést
* Meg kell határozni a lengés periódusidejét
* Ezután meghatározhatóak a szabályozó paraméterei táblázat alapján
* Hátrány: a lengések nem egészségesek mindig

### Oppelt módszer

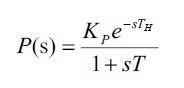
* Ugrás bemenet 🡪 felvesszük az átmeneti függvényt 🡪 grafikusan egytárolós holtidős taggal közelítjük
* Egyenértékű időállandó, látszólagos holtidő, erősítési tényező esetén meghatározhatóak a szabályozó paraméterei
* A módszer csak arányos folyamatokra jó
* A paraméterek táblázatban, amiben tartományok. A tartomány meghatározása után azon belül próbálgatással lehet finomítani (önhangoló szabályozók is így működnek)

## BME-WAY

### Arányos folyamat



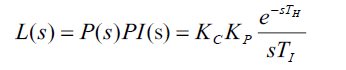
* Nyílt körben felvesszük az egységugrásra adott választ (Oppelthez hasonlóan)
* Grafikuson egytárolós holtidős taggal közelítjük
* Leolvassuk az értékeket (holtidő, időállandó, erősítési tényező)
* Egytárolós holtidős rendszer átviteli függvénye:



* Ehhhez tervezünk egy PI szabályozót, aminek az átviteli függvénye:

_ es)ld 
11s 

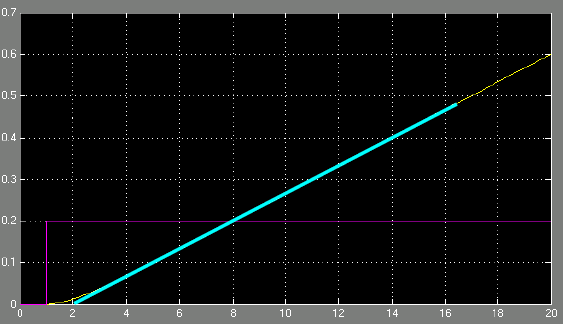
* Legyen Ti = T. Ekkor a felnyitott kör eredő átviteli függvénye:

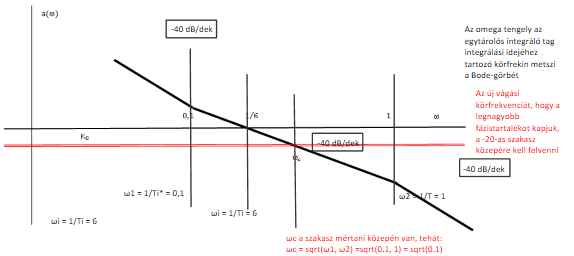


* Egytárolós holtidős taghoz integráló tervezésnél, ha az integráló holtideje kétszerese a tagnak, akkor azzal 60 %-s fázistartalékot kapunk.
* Itt a hotlidő: Ti/KcKp, aminek kell a tag holtidejének kétszeresének lennie. Így Kc = Ti/2ThKp

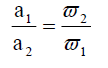
### Integráló folyamat

* A folyamat átmeneti függvényét integráló egytárolós taggal.
* T: egyenes kezdete és a step kezdete közötti idő: 1 s (egy holtidő)
* Ti = egyenes kezdete és a step elérése közötti idő : 6 s
* Ehhez PI-t tervezünk --> integráló egytárolós + PI = II-es szabályozó
* Ennek Bode diagramja, hogy úgy nézzen ki, hogy a két -40 dB/dek szakasz között legyen egy -20-as a PI integrálási idejét a rendszer holtidejei közül a legnagyobbnak a 10-szeresére választjuk
* Ti\* = T \*10 = 10;





* miután felvettük a Bode diagramon az összes szükséges paramétert az alábbi összefüggéssel tudjuk meghatározni az erősítést a -20-as szakasz bármely pontjára:



* A vörös vonal Kc-vel van eltolva. Tehát az eredeti metszéspontban Kc az erősítés, a mostaninál pedig egységnyi.
* Tehát:
  + a1 - Kc
  + a2 - 1
  + omega1 - omegai
  + omegac - sqrt(0.1)
* Majd behelyettesítés után: Kc = 6 \* sqrt(0.1)

### Bode diagram alapján tervezés I-es szabályozáshoz

* Meg kell rajzolni a felnyitott kör Bode-diagramját
* A szabályozó legyen PI
* Így a felnyitott kör:

|  |  |
| --- | --- |
|  | C:\Users\asarpi\AppData\Local\Temp\msohtmlclip1\02\clip_image016.png |

* Ahol a folyamat egy kéttárolós tag:

|  |  |  |  |
| --- | --- | --- | --- |
|  |  |  | 1  (I + S TI) (I + ST2) |

* És legyen a PI egységnyi erősítéssel:

|  |  |
| --- | --- |
|  | PI  STI |

* A stabil szabályozáshoz az új körfrekit a -20-as szakaszon kell felvenni (0db-es tengely lefele eltolása Kc-vel)
* Az új körfrekit a jobboldali törésponthoz tartozó körfreki 60%-ára kell felvenni (pont ekkor 60%-os a fázistartalék)
* A fenti arányossággal számolható az új erősítés:
  + a1 - Kc
  + a2 - 1
  + omega1- 1/T1
  + omega2 - 0.6/T2
* Ebből kijön, h Kc = 0.6 \* T1 / T2

a(o) 
1/Tl 
eredeti 
ú] 
0,6/T2 
1/T2 

* A hurokerősítés a -20-as szakasz 0 dB alatti hosszától függ.
* 45%-os fázistartalék: ha az új freki omega2-ben van (Kc45 = T1/t2)
* Ha a követelmény, hogy a beállás aperiodikus legyen, akkor a zárt kör átviteli függvényéből számolható a kívánt körfreki:

TIT2S2 + TIS + Kc 

* nevező gyökeinek diszkriminánságnak 0-nak kell lennie:

C:\Users\asarpi\AppData\Local\Temp\msohtmlclip1\02\clip_image022.png

* Ekkor:

099 YV O 
z %sz'0 

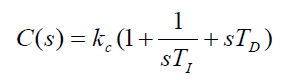
## Szabályozók programozása

### Folytonos idejű szabályozók

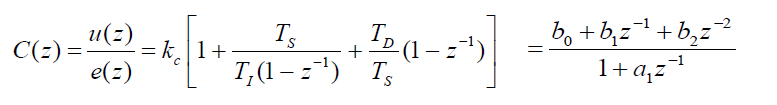
* Ha a mintavételezési frekvencia min. 10x nagyobb, mint a szabályozási kör vágási frekvenciája 🡪 lehet folytonos szabályozót tervezni, de diszkrétként kell realizálni

#### Folytonos PID

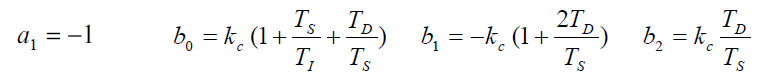
Átviteli fv.:



Átalakítva zpk-ba:



Az alábbi tényezőkkel



A diszkrét alakból a sebesség algoritmus:



Ebből adódik a rekurzív pozíció algoritmus



#### Elintegrálódás kezelése:

A programon belül korlátozás megvalósítása.

Egy PI implementáció (b2 = 0)

ek = rk – yk;

uk = uk1 + b0 \* ek + b1 \*ek1; // PI beavatkozójel

// threshold

if (uk < Umin) {

uk = Umin;

}

if (uk > Umax) {

uk = Umax;

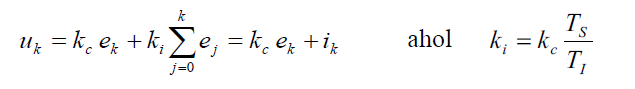
}

// parameter átadás a következő ciklusnak

uk1 = uk;

ek1 = ek;

#### Integráló rekeszes PI szabályozó



Az integrálást elemi módon, egy memória rekeszben való összegzéssel imlementáljuk:

Rekeszes PI implementáció (b2 = 0)

ek = rk – yk; // hibajel

ik = ik + ki \* ek; // szummás rész :

vk = kc \* ek + ik; // vk = kv \* hibajel + szummás rész

// threshold

if (vk < Umin) {

uk = Umin;

}

else if (vk > Umax) {

uk = Umax;

}

else {

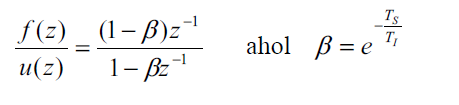
uk = vk;

}

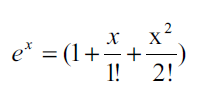
ik = ik + (uk – vk); // integrator visszaállítás

#### FOXBORO szabályozó

A szabályozó átviteli függvénye a diszkrét térben:



Az impelentáció során nem kell a pontos exponenciális függvényt használni. Elég a gyorsabb, Taylor-soros közelítés:



FOXBORO implementáció:

ek = rk – yk;

uk = kc\*ek + fk; // elsőfokú szabályozásszámolása

// threshold

if (uk < Umin) {

uk = Umin;

}

if (uk > Umax) {

uk = Umax;

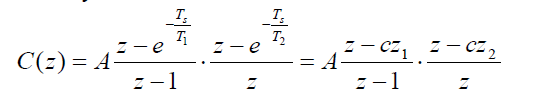
}

// foxboro szabályozás rákötése

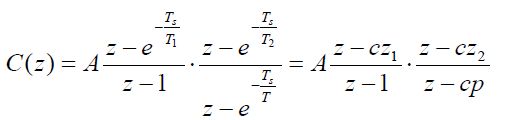
fk = beta \* fk + (1 – beta) \* uk; // fk-t használunk már az egyenletben is

### Diszkrét idejű szabályozók

* ha a mintavételezés körfrekvenciája közel van a felnyitott kör vágási körfrekvenciájához 🡪diszkrét szabályozót kell tervezni.
* diszkrét PID:



* Közelítő diszkrét PID
  + előnyös, mert a differenciáló hatást elnyújtja



* Szabályozó tervezése (alacsonyfrekvenciás közelítés)
  + meghatározzuk a folyamat átviteli fv-ét
  + PI szabályozó cz1 zérusával kiejtjük a folyamat legnagyobb időállandójának megfelelő pp1 pólust
  + PD szabályozó cz2 zérusával kiejtjük a második legnagyobb időállandónak megfelelő pp2 pólust
  + cz1 = pp1, cz2 = pp2
  + használjuk a rekurzív PID-et

## Lökésmentes indítás

* **Ipari szabályozóban:**
  + kézi
  + automatikus
    - az irányító program az initializáció alatt beméri az aktuális irányító jelet (kézi irányítás jelét)
    - az alapjelből és az aktuális ellenőrző jelből kiszámítjuk a legelső alkalommal kiküldendő irányítandó jelet úgy hogy egyenlő legyen a bemért kézi irányító jellel + definiáltalan/határozatlan értékeket inicializáljuk (magyarul: bypasseljük a szabályozót az első néhány ciklusban és egy másik szabályozóval indítjuk a gépet)
    - átkapcsolás automatikus üzemmódba 🡪kiküldeni a visszamérttel megegyező első irányító jelet
    - az átmenetnek akkor kell végbemennie, amikor az már nem okoz lökést
* **Robotokban**
  + nincs kézi-automatikus kapcsoló
  + a betanító pulttal végzet működtetés nem kézi üzem, hanem egy másfajta csuklószabályozás
  + lehető leggyorsabb szabályozás, minimális lökésmentesség
  + amikor a fékek elengednek pontosan akkora nyomatékot kell adni, hogy ne legyen lökés

C:\Users\asarpi\AppData\Local\Temp\msohtmlclip1\02\clip_image012.png

C:\Users\asarpi\AppData\Local\Temp\msohtmlclip1\02\clip_image012.png

C:\Users\asarpi\AppData\Local\Temp\msohtmlclip1\02\clip_image012.png

C:\Users\asarpi\AppData\Local\Temp\msohtmlclip1\02\clip_image025.png