

PID ALGORITMUS ALGORITMIZÁLÁSA IRÁNYÍTÓRENDSZEREK BEN

Móré Ádám 

egyetemi tanársegéd, Miskolci Egyetem, Automatizálási és Infokommunikációs Intézet
H-3515 Miskolc, Miskolc-Egyetemváros, e-mail: autmrdm@uni-miskolc.hu

Trohák Attila 

egyetemi docens, Miskolci Egyetem, Automatizálási és Infokommunikációs Intézet
H-3515 Miskolc, Miskolc-Egyetemváros, e-mail: attila.trohak@uni-miskolc.hu

Absztrakt

A cikk egy alkalmazott analóg PID algoritmus elkészítését mutatja be. Az algoritmus FBD (Funkció Blokk Diagramm) IEC szabványosított nyelven valósul meg. Az FBD programozási nyelvet PLC (Programmable Logic Control) és DCS (Distributed Control system) fejlesztői környezetek is támogatják. A cikk bemutatja, hogy egy matematikai numerikus analóg képletből, hogyan lehet készíteni egy algoritmust, melyet az irányítórendszerben tudunk használni. Különböző alkalmazásokon keresztül foglalkozik a felhasználó által készített algoritmus előnyével.

Kulcsszavak: PID szabályozás, irányítórendszer, irányítórendszerek cseréje, DCS, PLC

Abstract

The paper presents the construction of an applied analog PID algorithm. The algorithm is implemented in FBD (Function Block Diagram) IEC standardized language. The FBD programming language is supported by PLC (Programmable Logic Control) and DCS (Distributed Control System) development environments. The article shows how to make an algorithm from a mathematical numerical analog formula that we can use in the control system. It deals with the benefits of a user-created algorithm for different applications.

Keywords: PID control, control system, control system replacement, DCS, PLC

1. BEVEZETÉS

Az irányítástechnikát két fő csoportra bonthatjuk: vezérlés és szabályozás. A szabályozáson belül két csoportot különböztethetünk meg: értéktartó és érték követő. A vezérlést és a szabályozást az 1. táblázat hasonlítja össze. A táblázatban látható, hogy a szabályozás kevésbé érzékeny a zavarokra. Ez a tulajdonság mind a gyártásautomatizálásban, mind a folyamatiparban fontos. Az irányítórendszerek programjai tartalmazhatnak szabályozási algoritmusokat, melyek lehetnek: P, PI, PID, fuzzy logika és neurális hálózat. Az irányítórendszerek fejlesztőkörnyezetei tartalmaznak előre definiált szabályozó algoritmusokat. Ezeket az algoritmusokat a programozónak a programírásnál már csak paramétereznie kell. A kérdés az, hogy ha a gyártó biztosít a programozó számára tesztelt, megfelelően működő algoritmust, akkor szükséges-e saját algoritmust fejleszteni a programozónak? Bizonyos esetekben igen, ugyanis ezek az algoritmusok a fejlesztőkörnyezet által támogatott kontrollerekre lettek fejlesztve, tehát gyártóspecifikus algoritmusok és eltérő lehet a belső felépítésük, ezáltal a működésük. Léteznek olyan esetek

is, amikor szükséges egy saját algoritmust írni, ilyen például az irányítórendszer korszerűsítése és cseréje. A gyártó ad egy időtartamot, ami alatt az irányítórendszer biztonságosan használható. Azonban, amikor egy irányítórendszer meghibásodik, vagy elöregedik, akkor szükséges a cseréje. Ha egy régi, idejét múlt rendszerről beszélünk, akkor már előfordulhat, hogy nem tudunk olyan irányítórendszert beszerezni, továbbá minden terméknek van egy gyártó által nyújtott támogatása. Ha már megszűnik a támogatás és már nem is lehet beszerezni ilyen eszközt, akkor szükséges az irányítórendszert cserélni. A gyártó támogatja a programmigrálást, ami azt jelenti, hogy korábbi verzió megírt programot, át lehet migrálni az új fejlesztőrendszerbe. Az esetek nagy részében ez sikeres szokott lenni. Azonban, ha a program tartalmaz szabályozásokat, akkor a régebbi paraméterekkel és szoftverstruktúrákkal, működési problémák léphetnek fel. Ez a jelenség rengeteg beüzemelési időt vesz igénybe, romolhat a termék, vagy az üzemeltetés minősége is. Erre a problémára egy általános algoritmus elkészítése javasolt, ami egy PID szabályozáson keresztül szemlélteti az algoritmus elkészítésének a lépéseit, tesztelését és szimulációját. (Bánhidi et al., 1991; Csáki és Bars, 1974; Földi, 2017)

1. táblázat. Vezérlés és szabályozás összehasonlítása

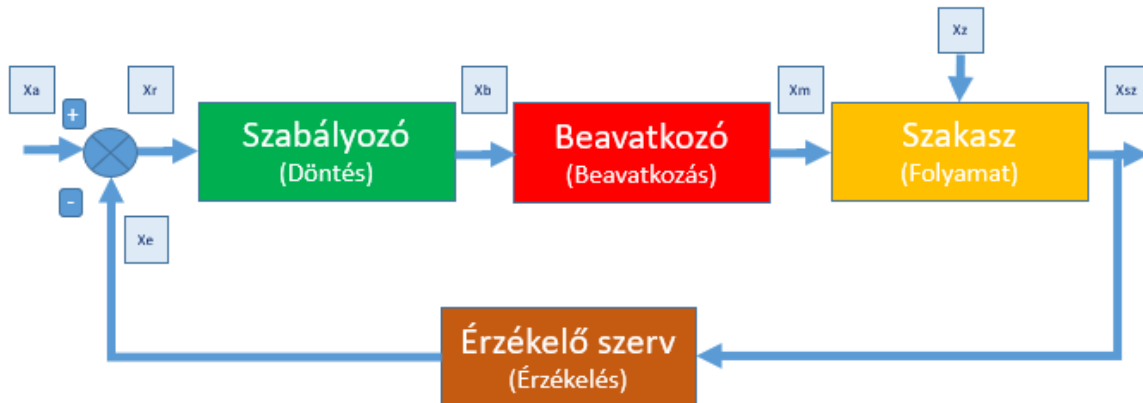
Szemponatok	Szabályozás	Vezérlés
Felépítés	Zárt hatáslánc, több tagból áll	Nyílt hatáslánc, kevesebb tagból áll
Zavarérzékenység	Kevésbé zavarérzékeny a külső zavarokra	Zavarokra jobban érzékeny
Folyamat paraméterérzékenység	Kevésbé érzékeny	Pontosan ismerni kell a bemenet-kimenet közötti kapcsolatot
Stabilitás	Lehetnek stabilitási problémák (túlszabályozás) Egy labilis rendszer visszacsatolással stabilizálható	Nincs stabilitási probléma
Kimenet mérése	A kimenet mérése elengedhetetlen	Nem kell mérni a kimenetet
Kalibrálás	Hosszú időként kell csak újra kalibrálni	Gyakran kell kalibrálni
Költség	Általában magasabb költségek	Kevésbé költséges

2. PID SZABÁLYOZÓ KÖR

A bevezetés tárgyalta, hogy mi a különbség a vezérlés és a szabályozás között, illetve, hogy miért lehet szükségünk egyedi, gyártófüggetlen szabályozási algoritmusra. Most bemutatásra kerül egy soros PID szabályozó kör, amit 1. ábrán ábrázoltam.

A szabályozás művelete során az X_r rendelkezőjel, egy előzetesen megadott X_a alapjelnek, és az X_{sz} szabályozott jellemző pillanatnyi értékétől függő X_e ellenőrző jelnek a különbsége. Az X_r rendelkezőjel (más néven hibajel) – rendszerint egy jelformáló és erősítőszerven végigfutva, az X_b beavatkozási jellel működésre készíti a beavatkozási szervet. A beavatkozási szerv kimenőjele, az X_m módosított jellemző, olyan hatást vált ki a szabályozott berendezésben, amely az X_{sz} szabályozott jellemzőnek az előírt értéktől való eltérését igyekszik megszüntetni. A beavatkozási szerv kimenőjele, az X_m módosított

jellemző, olyan hatást vált ki a szabályozott berendezésben, amely az X_{sz} szabályozott jellemzőnek az előírt értéktől való eltérését igyekszik megszüntetni.



1. ábra. PID szabályozó kör

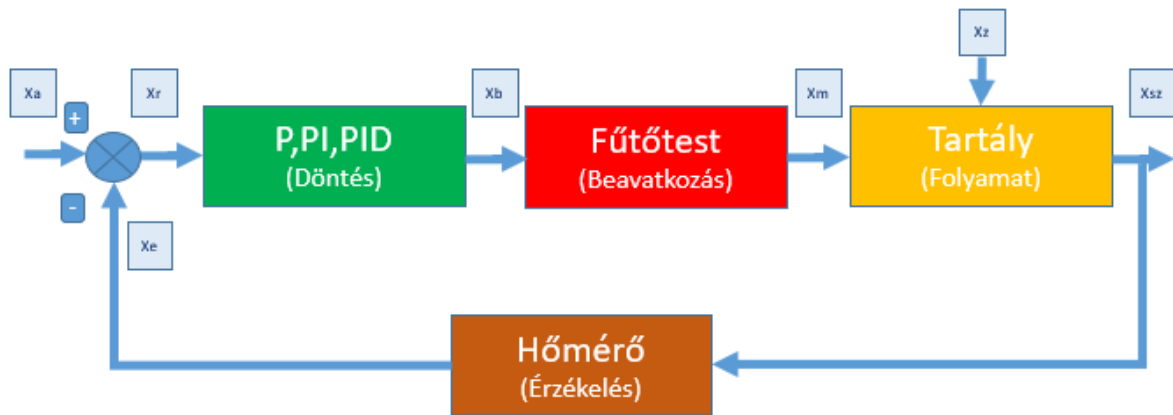
A beavatkozószer kimenőjele, az X_m módosított jellemző, olyan hatást vált ki a szabályozott berendezésben, amely az X_{sz} szabályozott jellemzőnek az előírt értéktől való eltérését igyekszik megszüntetni. A megkívánt, előírt hatás kiváltásával, elérésével a szabályozás művelete befejeződött. A szabályozott berendezésre természetesen hatnak az X_z zavaró jellemzők. Igen gyakran éppen azért alkalmaznak szabályozóberendezéseket, hogy segítségükkel a nem kívánatos zavaró hatásokat kiküszöböljék. A szabályozás művelete tehát úgy befolyásolja a műszaki folyamatot, hogy az X_{sz} szabályozott jellemző megváltozása, visszahat a hatáslánc elejére. Ha eltérés mutatkozik az X_a alapjel és az X_e ellenőrző jel között, létrejön egy ún. szabályozási eltérés, egy X_r rendelkezőjel (hibajel), amely az előzőekben leírt módon fejti ki hatását.

Szabályozásnál a szabályozott jellemző értékét egy ellenőrző szerv (érzékelő) figyeli, és eltérés esetén (azaz abban az esetben, ha a folyamat lefolyása eltér az előírttól) megváltoztatja a rendelkező jelet, és mindaddig korrigál, amíg a folyamat ismét az előírás szerint zajlik. A szabályozás hatáslánca zárt. A szabályozási művelet a negatív visszacsatolás elve alapján valósul meg. (Bánhidi et al., 1991; Csáki és Bars, 1974)

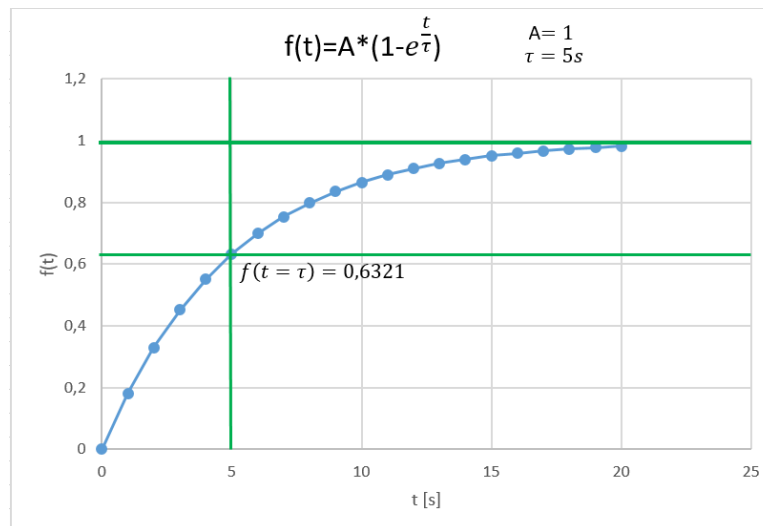
3. P, PI, PID SZABÁLYOZÓ ANALÓG MODELLJE

A szabályozó modellje egy alkalmazási példán kerül bemutatásra, ami egy tartály hőmérséklet szabályozása. A következő 2. ábrán láthatjuk a példa szabályozó körének felépítését. A szabályozási kör elemeinek a funkciója: a tartály, a benne lévő anyag (víz), és maga a tartály a folyamat, ahol a hőmérséklet szabályozását fogjuk megvalósítani. A hőmérő segítségével mérjük a tartályban levő víz hőmérsékletét. A P, PI, PID szabályozó, segítségével fogjuk szabályozni a tartályban levő anyag hőmérsékletét, ahol az alapjel és a szabályozott jellemző közötti érték különbsége fog alapul szolgálni. A fűtőtest, mint beavatkozó fogja tudni módosítani a tartályban levő anyag hőmérsékletét (csak növelni fogja tudni). (Bánhidi et al., 1991; Csáki és Bars, 1974)

A tartály fizikai működését a 3. ábra szemlélteti (az ábrán az időállandó csak példaként $\tau = 5s$). Átmeneti függvényét az egységugrás függvényre adott válasza adja. Ezt a következő függvény adja (3. ábra).



2. ábra. PID hőmérséklet szabályozási kör



3. ábra. A tartály fizikai modellje

Az ábra segítségével lehet majd meghatározni a tartály időállandóját és az „A” erősítést a következő módon:

- A T, azaz a 3. ábrán a τ időállandó megmérése: A tartályban levő víz legyen statikus egyensúlyi állapotban és $20\text{ }^{\circ}\text{C}$ -on. Állandó teljesítményű melegítés hatására, a tartályban levő anyag elkezd melegedni. Valamennyi idő elteltével az új, állandósult állapotban kialakult hőmérsékletet megmérjük, legyen ez a hőmérséklet a T_2 . A melegítés leállítás után megvárjuk míg újra visszahűl a tartályban levő anyag $20\text{ }^{\circ}\text{C}$ -ra.
- Ezek után újból az előbbi teljesítménnyel kezdjük el melegíteni a tartályban levő vizet, de most a melegítés kezdetétől mérjük az időt is. A „T” időállandó értékét, az az idő fogja adni, amikor a tartályban levő anyag hőmérséklete eléri a $0,6321 * T_2$ hőmérsékletet (lásd az előbbi ábrát).

- Az „A” erősítési tényező mérése: Meg kell mérnünk, hogy 1 W hatására hány °C-ot fog emelkedni az anyag hőmérséklete. Tehát a mért hőmérsékletet el kell osztani a melegítési teljesítménnyel: $A = \frac{T_2 - 20}{P}$, ahol a P a melegítési teljesítmény.

A tartályban levő víz mennyiségét a modellünkben egy egytárolós arányos taggal fogjuk közelíteni. A tároló jellegét a tartályban lévő anyag hőkapacitása, az arányos jellegét a hőellenállása adja. A dinamikus viselkedését leíró differenciálegyenlet általános alakja, két fontos állandót is tartalmaz: a „T” időállandót és az „A” erősítési tényezőt.

$$T * \frac{dx_{KI}}{dt} + x_{KI} = A * x_{BE} \quad (1)$$

A differenciálegyenletet az 1. egyenletet Laplace transzformáljuk („s” a Laplace változó), majd megfelelő átrendezéssel megkapjuk az átviteli függvényét. (Ajtonyi és Gyuricza, 2002)

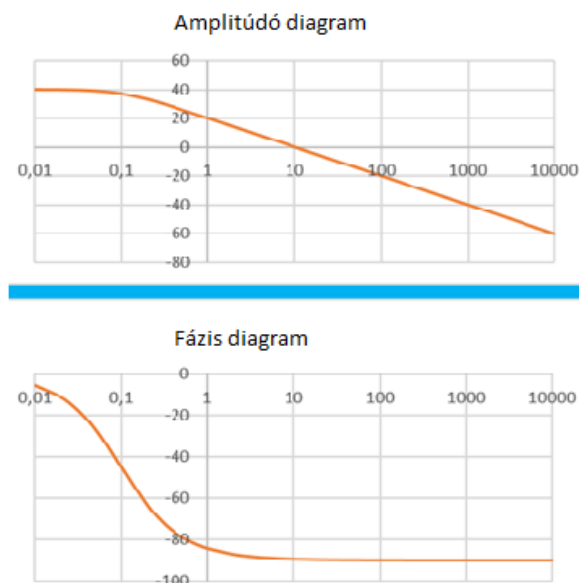
$$T * s * x_{KI}(s) + x_{KI}(s) = A * x_{BE}(s) \quad (2)$$

$$(T * s + 1) * x_{KI}(s) = A * x_{BE}(s) \quad (3)$$

Az átviteli függvény:

$$W(s) = \frac{x_{KI}(s)}{x_{BE}(s)} = \frac{A}{1 + T * s} \quad (4)$$

melynek Bode-diagramját a 4. ábra mutatja (példaként most: A = 100; T = 10 s):



4. ábra. Az egytárolós arányos tag Bode-diagramja

A modellnél egy hőellenállással fogjuk modellezni a tartály falán kiáramló hőt, hővesztéséget mint zavaró tényezőt. Ez a hőellenállás segíteni fog abban, hogy a tartályban levő anyagot „hűteni” is tudjuk, vagyis negatív „beavatkozó jelként” is felfogható. (Ogata, 1997)

4. PID SZABÁLYOZÓ ALGORITMIZÁLÁSA

Most a PID szabályozás következik, amelyet PLC-n kell algoritmizálni, ezért digitális elemekkel kell kialakítani. Ahhoz, hogy ez az eljárás alkalmazható legyen, a PID szabályozás egyenletét, digitálisan fel kell írni, amely a következő:

$$x_i = K_P * e_n + \frac{T}{T_I} \sum_{i=0}^n e_i + \frac{T_D}{T} (e_n - e_{n-1}) \quad (5)$$

Ahol,

x_i : a PID kontroller kimenete az i . időpontban

T : mintavétel idő

K_P : arányos tag erősítési tényezője

T_I : integrálási idő

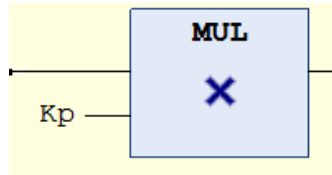
T_D : differenciálási idő

e_i : a hibajel az i . időpontban

e_n : hibajel az adott időpontban (Csáki és Bars, 1974; Ajtonyi és Gyuricza, 2002)

Tehát ezt a képletet kell megvalósítani az algoritmizálás során FBD nyelven, hogy a PLC programban végre tudjuk hajtani az eljárást. A képletet célszerű tagokra bontani, mert így külön tudjuk kezelni a P, I és D tagot. (Dai és Vyatkin, 2009)

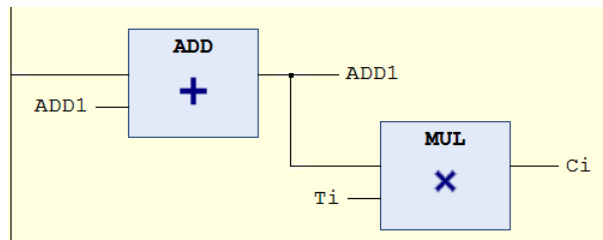
A P tag megvalósítása az 5. képletben a következő: $K_P * e_n$ vagyis ezt a tagot egy szorzás blokkal lehet megvalósítani, amit a következő ábra szemléltet:



5. ábra. P tag megvalósítása

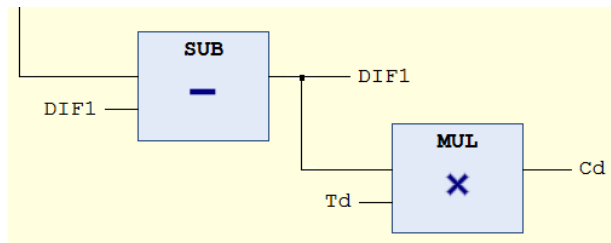
Az 5. ábra mutatja be, hogy hogyan valósult meg a P tag. A szorzó bloknak két bemenete és egy kimenete van. A bemenetre érkezik a különbség képzőből a hibajel (e_n), és a K_P azaz az arányos tag erősítési tényezője. A kimenete pedig a PID függvény összegzőjére megy, mely a későbbiekben egy ábrán is szemléltetve lesz.

Az I tag a fentebb látható 5. képletből a következő: $\frac{T}{T_I} \sum_{i=0}^n e_i$. Tehát ezt az integrálást kell megvalósítani. Ezt úgy lehet megtenni, hogy egy visszacsatolt összeadó blokk végező egy szummázást, majd ezt össze kell szorozni a képletből adódó $\frac{T}{T_I}$ taggal, és a szorzó kimenetét a PID összegzőjére kötni. Tehát a hibajel az összegző bemenetére csatlakozik, amelynek a kimenete visszacsatolt egy másik bemenetére (ADD1), majd egy szorzó blokkba kötve és a szorzó blokk másik bemenetét T_I taggal szorozva. Azt meg kell jegyezni, hogy ez a T_I tag valójában a képletben a $\frac{T}{T_I}$. Mivel a változó nevével nem lehet csak összefüggő karakterekkel elnevezni, ezért rövidítésként T_I -vel neveztük el.



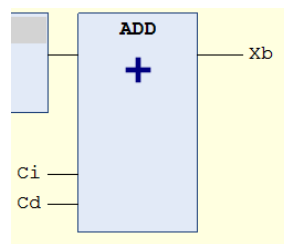
6. ábra. I tag megvalósítása

A D tag az 5. képletben a következő: $\frac{T_D}{T}(e_n - e_{n-1})$. A D tagot egy visszacsatolt kivonóval és egy szorzóval valósítottuk meg a következőképpen:



7. ábra. D tag megvalósítása

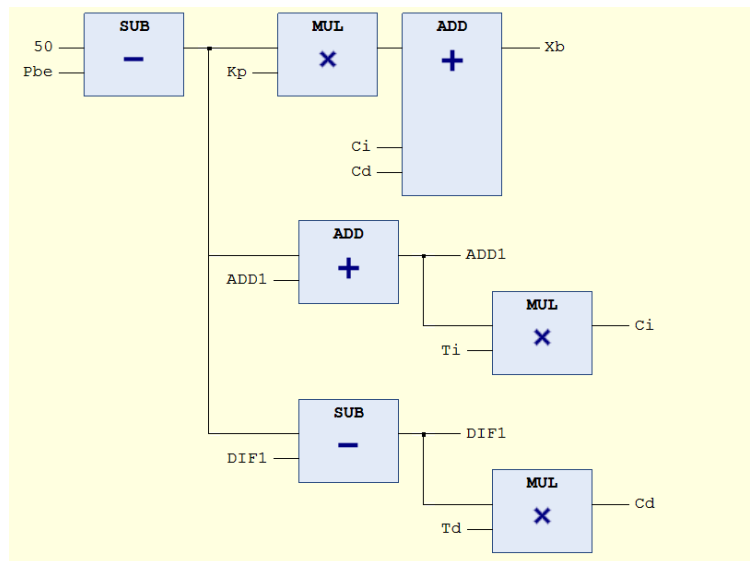
A kivonó egyik bemenetére a hibajel van kötve, a másik bemenetére pedig a DIF1 csatlakozik, azaz a differenciálónknak a kimenetét visszacsatolva a bemenetére. A DIF1 az $e_n - e_{n-1}$ -et értéket csatolja vissza a különbségképzőre, majd a kivonó kimenetét, azaz a DIF1-et rákötöttük a szorzó bemenetére, a másik bemenetére pedig rákötöttük a T_D -t, amely a képletben $\frac{T_D}{T}$. Itt is azért van így elnevezve, mert nincs lehetőség törttel elnevezni. A PID függvény összeadóját a következő ábra szemlélteti:



8. ábra. PID függvény összeadója

Az összegző, egy 3 bemenetes összeadó blokk, ahol az egyik bemenet a P-tag kimenete, a másik bemenet a C_i , azaz az integráló tag kimenete és a C_d pedig a differenciáló tag kimenete. Az X_b a PID szabályozó kimenete. A következő ábra az egész szabályozónkat mutatja be.

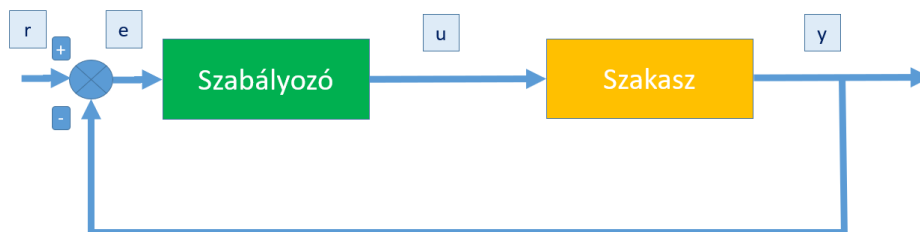
Ilyen alakban valósítottuk meg a PID szabályozót FBD nyelven PLC-re. A 9. ábrán látható még egy tag, amelyről még nem beszéltünk. Ez lesz a különbségképző, amelynek a bemenetére egy alapjelet kötöttünk, továbbá a szabályozókörünk kimenetét itt csatoltuk vissza. Ezt a visszacsatolt P_{be} értéket a következő részben mutatjuk be.



9. ábra. PID szabályozó algoritmizált alakja FBD nyelven

5. SZABÁLYOZÓ KÖR MEGVALÓSÍTÁSA

A szabályozási kör modell megvalósítása alapjaként a következő ábra szolgál. Az előző részben már megvalósítottuk a kör különbségképzőjét és a szabályozóját. A mostani fejezetben a szabályozandó szakaszt fogjuk megvalósítani.



10. ábra. Szabályozó kör egyszerűsített alakja

Ahol,

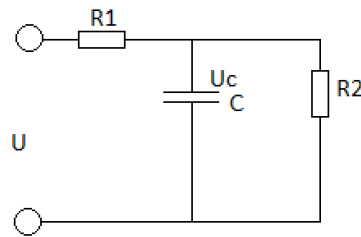
r: alapjel

e: hibajel

u: beavatkozó jel

y: szabályozott jellemző

A szakaszblokkban modellezzük a beavatkozót, a szabályozandó szakaszt és az érzékelést. Az analóg modellezésnél kiderült, hogy a jelenlegi hőmérséklet-szabályozásnál ez a három alkotórész egy domináns, egytárolós arányos taggal modellezhető, amelynek paramétereit a tartály és a tartályban lévő anyag fizikai paramétereire határozzák meg. A modell megalkotásánál figyelembe vettük a tartályban lévő folyadék lehűlését (nem tökéletesen szigetelt tartály). Ezek után a szakaszt megvalósító analóg modell a következő:



11. ábra. Egyszerűsített analóg modellje a szabályozott szakasznak

Ahol,

R_1 : a fűtőtest és tartályban levő víz közötti hővezetési ellenállás,

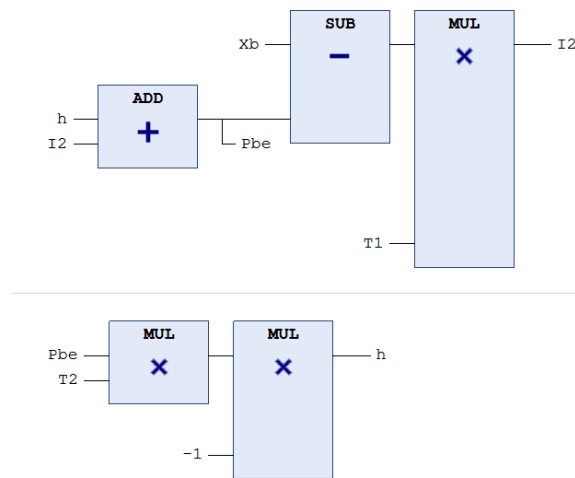
C : a tartályban levő víz hőkapacitása,

R_2 : a hőveszteség,

U_c : a tartályban levő hőmérséklet.

$$\Delta U = \frac{\Delta Q}{c} = \frac{\frac{U - U_c}{R_1} \Delta t}{c} - \frac{U_c}{R_2} \Delta t = (U - U_c) * T_1 - U_c * T_2 \quad (6)$$

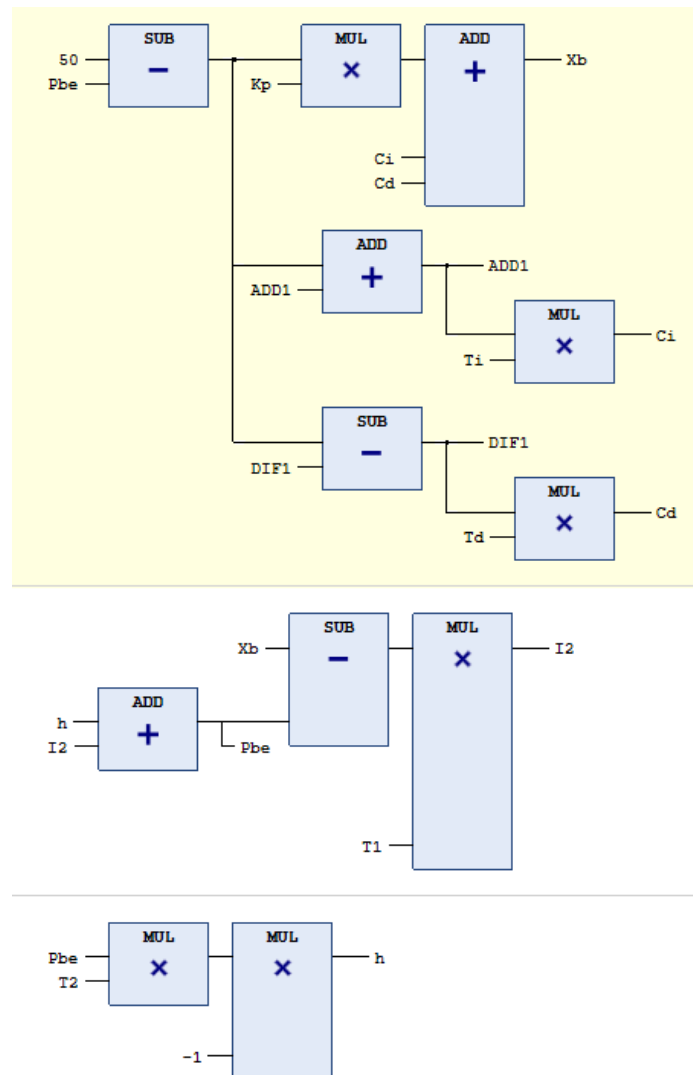
Az ezt megvalósító FBD nyelvű programot a 12. ábra mutatja be:



12. ábra. Szabályozott szakaszt megvalósító modell FBD nyelven

X_b lesz a szabályozó kimenete, vagyis a beavatkozó jel (a 11. ábrán „ U ”). A P_{be} a szabályozott jellemző, amely a 11. ábrán az U_c . Az U_c -t a következő módon állítottuk elő: Először is a szorzóblokkok állítják nekünk elő a $(-U_c * T_2)$ tagot. Ezt úgy állítottuk elő, hogy az első szorzóblokk egyik bemenetére visszacsatoltuk a P_{be} -t, továbbá a T_2 konstanssal szoroztuk meg a képlet szerint. Mivel nekünk a tag elején lévő kivonást is be kellett illeszteni, ezért ennek a szorzóblokknak a kimenetét rákötöttük egy másik szorzóblokkra, amelynél a másik bemenetre egy -1 -et kötöttük. A blokk kimenete adja a h -t, amelyet a P_{be} összegző bemenetére kötöttük, majd az I_2 -t kötöttük az összegző másik bemenetére. A P_{be} -t és az X_b -t

egy különbségképzőre kötöttük, amelynek a kimenetét egy szorzóra kötjük tovább. Ennek a szorzónak a kimenete adja az I_2 -t, amely valójában a 6. képletbe a következő: $(U - U_c) * T_1$. Tehát a „h” a hővesztésből adódó tag, az „ I_2 ” pedig a szabályozóból jövő beavatkozó jel következménye.

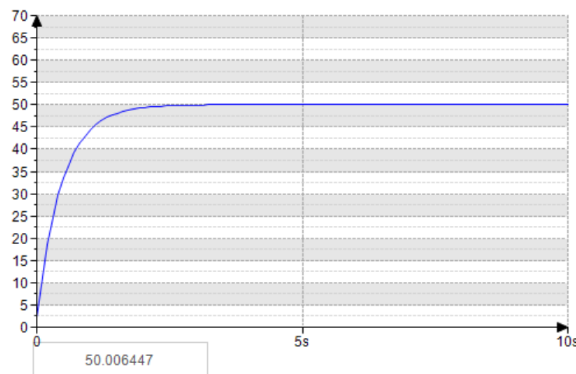


13. ábra. A szabályozási kör FBD nyelven írt modellje

Tehát a következőképpen vettük fel a paramétereket. Itt meg kell jegyeznünk, hogy ezek a paraméterek becslésekből származnak, ugyanis nem volt lehetőségünk fizikailag megvalósítani a szabályozási kört, ezért kellett szimulációval megbecsülni. A kísérletileg beállított paraméterek a következők:

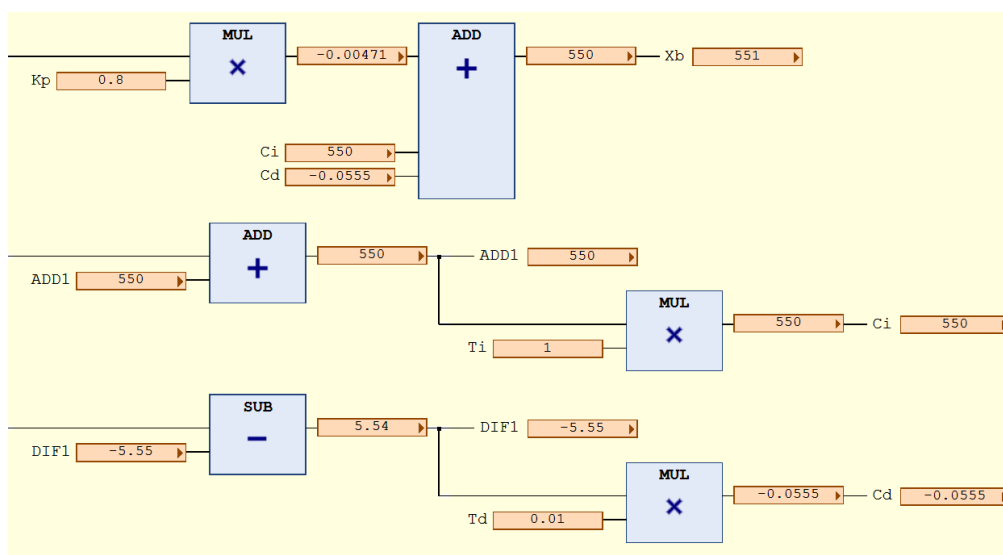
Alapjel = 50 (50 °C-os folyadékot szeretnénk kapni a tartályban). Pbe = 20 (kezdetben 20 °C-os folyadékot feltételezve a tartályban). PID paraméterek: Kp = 0,8, TI = 1 és TD = 0,01. Szabályozott szakaszt jellemző paraméterek: T1 = 0,1 (beavatkozó jel releváns konstans) és T2 = 0,001 (hővesztés releváns konstans).

6. PID ALGORITMUS SZIMULÁCIÓJA ÉS VIZUALIZÁCIÓJA



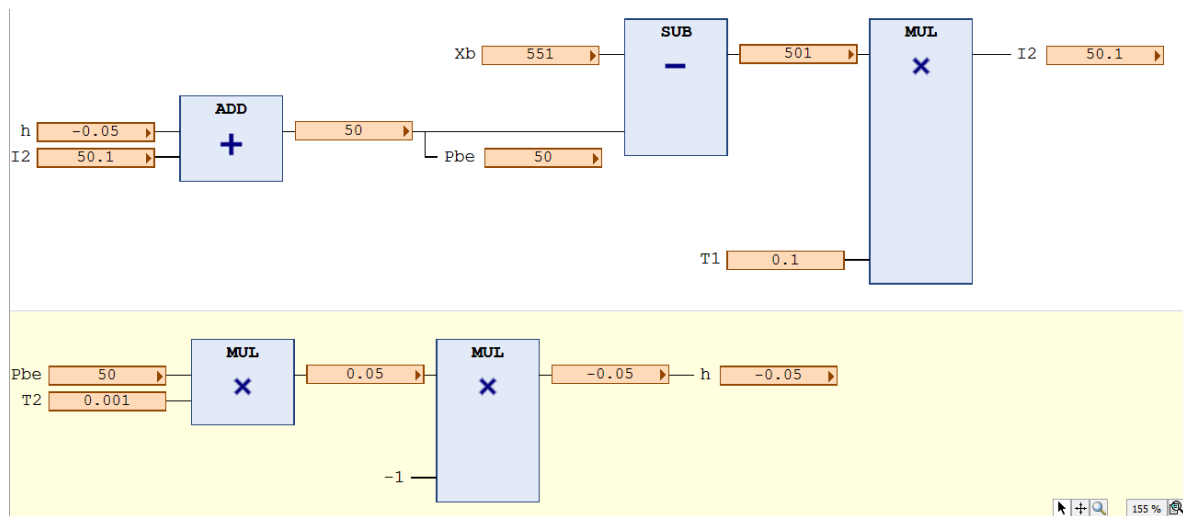
14. ábra. Digitális PID függvénye

A 14. ábrán látható, hogy a szabályozott jellemző, a tartályban lévő folyadék hőmérséklete, hogyan veszi fel a beállított 50 °C-os alapjel értéket. Továbbá látható, hogy a program numerikusan is megadja a pillanatnyi értéket. Tehát az általunk írt PID algoritmus és szabályozó áramkör sikeresen szimulálja a folyadék hőmérsékletének szabályozását. (Ajtonyi és Gyuricza, 2002)



15. ábra. PID szabályozó szimulációja

A 15. ábrán láthatóak a PID szabályozó pillanatnyi értékei egy szimuláció elindítása után. A programban lehetőségünk van a FBD blokkokat egyesével elemeznünk, megfigyelni az értékük változását. A P, I és D tagok működését is nyomon tudjuk követni, valamint a PID szabályozó kimeneti értékének a változását is. A következő 16. ábra a szabályozott szakaszt mutatja be. (Palkin és Suvorov, 2021)



16. ábra. Szabályozott szakasz szimulációja

A 16. ábrán megfigyelhető a szabályozott szakasz fizikai pillanatnyi értéke lépésről lépésre a szimuláció alatt.

7. KONKLÚZIÓ

A kutatómunka során bemutatásra került, hogy hogyan épül fel egy PID szabályozási kör. Egy konkrét példán keresztül bemutattuk a hőmérséklet PID szabályozás algoritmizálását az általunk kidolgozott módszer alkalmazásával. A cikk bemutatta, hogy milyen esetekben érdemes saját szabályozó algoritmust fejleszteni. Az elért eredmények további vizsgálati, összehasonlítási lehetőséget tesznek lehetővé a területen.

IRODALOM

- [1] Bánhidi, L., Oláh, M., Gyurica, I., Kiss, M., Rátkai, L., Szecső, G. (1991). *Automatika*. Nemzeti Tankönyvkiadó, Budapest.
- [2] Csáki, F., Bars, R. (1974). *Automatika 3*. Tankönyvkiadó, Budapest.
- [3] Ajtonyi, I., Gyuricza, I. (2002). *Programozható irányítóberendezések, hálózatok és rendszerek*. Műszaki Könyvkiadó, Budapest.
- [4] Földi, Sz. (2017). *Különböző platformokon megvalósított PID szabályozások vizsgálata, összehasonlító elemzése*. TDK-dolgozat.
- [5] Ogata, K. (1997). *Modern Control Engineering*. Prentice-Hall Inc.
- [6] Palkin, G., Suvorov, I. (2021). Simulation modeling of first rise section of water supply system with installed complex of automatic pump performance control. *Machines*, 9 (63).
<https://doi.org/10.3390/machines9030063>
- [7] Dai, W. W., Vyatkin, V.: *A case study on migration from IEC 61131 PLC to IEC 61499 function block control*, 2009 7th IEEE International Conference on Industrial Informatics.
<https://doi.org/10.1109/INDIN.2009.5195782>