

# Proporcionális hőmérsékletszabályozás

## 1. A gyakorlat célja

Az impulzusszélesség modulált jelek szoftverrel történő generálása. Hőmérséklet szabályozás implementálása P szabályozóval.

## 2. Elméleti bevezető

### 2.1 A proporcionális szabályozás

Az aktuálisan mért hiba függvényében számítjuk ki a beavatkozó jelet. A beavatkozó jel arányos a mért hibával. Minél nagyobb az  $e$  szabályozási hiba (a folyamat  $y$  mért kimenete minél távolabb van az  $r$  alapjeltől), annál nagyobb lesz a kiszámított beavatkozó jel ( $u$ ). Folytonos időben a beavatkozó jel számítása P szabályozó esetén:

$$u(t) = K_P \cdot e(t), \quad K_P > 0 \quad (1)$$

$$e(t) = r(t) - y(t) \quad (2)$$

$K_P$  proporcionális erősítés, a szabályozó erősítése.

A szabályozó átviteli függvénye megegyezik az ideális erősítő átviteli függvényével.

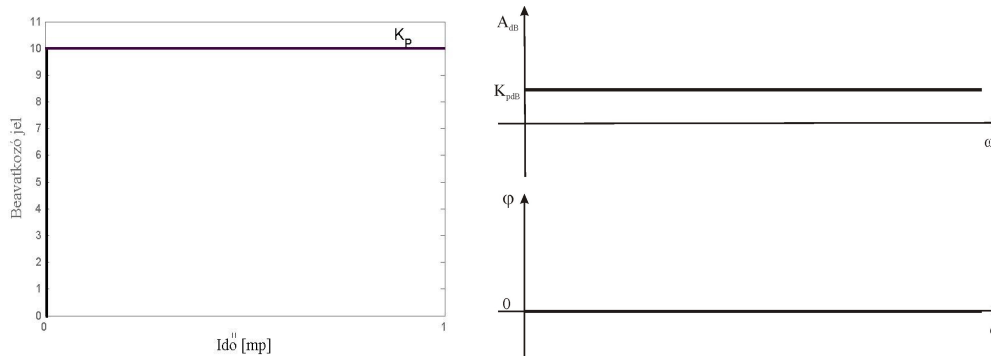
$$u(s) = K_P \cdot e(s)$$

$$H_P(s) = \frac{u(s)}{e(s)} = K_P \quad (3)$$

Mintavételes megvalósításnál a  $k$ -ik mintavételben mért alapjel és kimenet alapján a beavatkozó jel számítása:

$$u_k = K_P \cdot e_k = K_P (r_k - y_k) \quad (4)$$

A szabályozó egységugrásra adott válasza és Bode diagramja az 1. Ábrán látható.



**1 Ábra: Az ideális P szabályozó egységugrásra adott válasza és Bode diagramja**

A Proporcionális - Deriváló kialakítás figyelembe veszi a hiba változását is. A deriváló csatorna a hiba változásából következtet a hiba tendenciájára, jövőbeli alakulására és a szabályozó ezt is figyelembe veszi a beavatkozó jel számításánál.

A beavatkozó jel számításánál a hiba tendenciáját a hiba deriváltjával jellemezzük:

$$u(t) = K_P \left( e(t) + T_d \frac{de}{dt} \right), \quad K_P, T_d > 0 \quad (5)$$

$T_d > 0$  paraméter a deriválási idő.

A mintavételes kialakításnál a deriváló csatorna megvalósításához a hátrtartó differencia közelítést alkalmazhatjuk.

$$\frac{de}{dt} \cong \frac{e_k - e_{k-1}}{T} \quad (6)$$

$T$  a mintavételi periódust jelöli.

A (6) alapján a beavatkozó jel számítása a  $k$ -ik mintavételben:

$$u_k = K_P \cdot \left( e_k + T_d \frac{e_k - e_{k-1}}{T} \right) \quad (7)$$

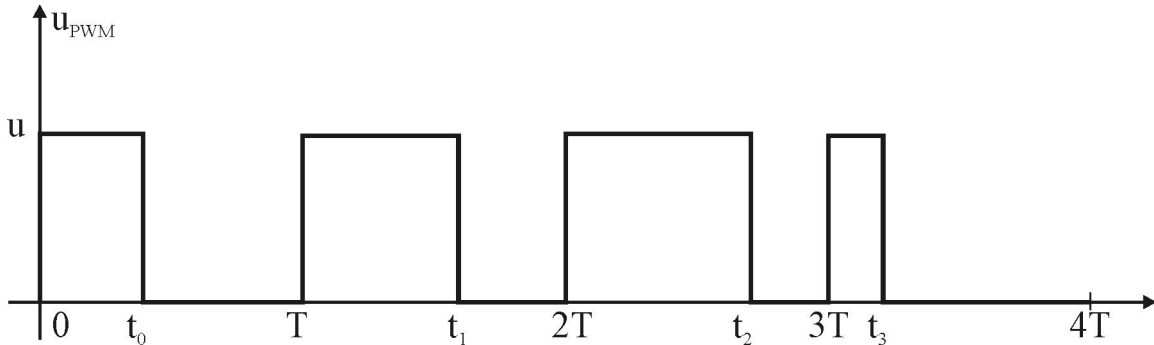
## 2.1 Impulzus szélesség modulált (PWM) jel generálása

Az impulzus szélesség modulált jel két diszkrét állapotot felvevő ( $0 - U_{MAX}$  vagy  $-U_{MAX} - U_{MAX}$ ), konstans periódusú ( $T$ ) jel, amelyben egy periódus alatt az impulzus szélessége változtatható (lásd 2. Ábra).

Határozzuk meg a PWM jel átlagértékét egy  $T$  periódus alatt. Feltételezzük, hogy a PWM jel szélessége (kitöltése)  $0 < t < T$ .

$$\tilde{u}_{PWM} = \frac{1}{T} \int_0^T u_{PWM} d\tau = \frac{1}{T} \left[ \int_0^t u d\tau + \int_t^T 0 d\tau \right] = \frac{1}{T} u \int_0^t d\tau = \frac{u}{T} t$$

Tehát a PWM jel átlagértéke egyenesen arányos a kitöltéssel.



2. Ábra: PWM jel

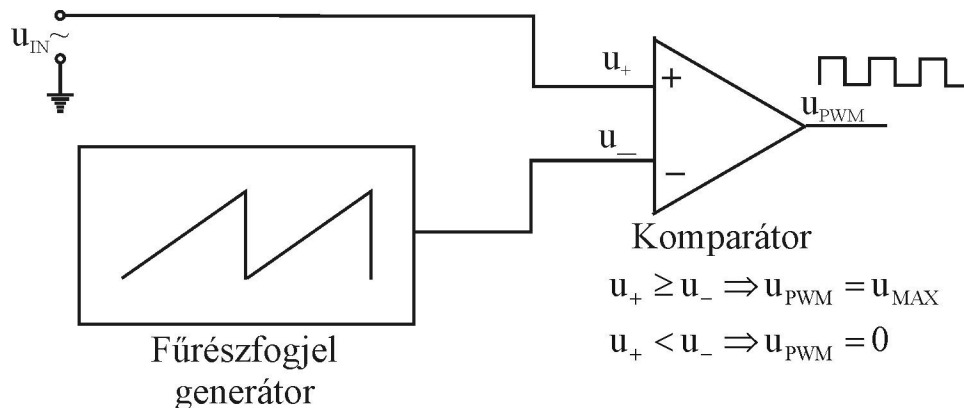
A PWM jel generátorok olyan rendszerek, amelyeknek a bemenete a PWM kitöltésének megfelelő jel, a kimenete pedig a PWM jel.



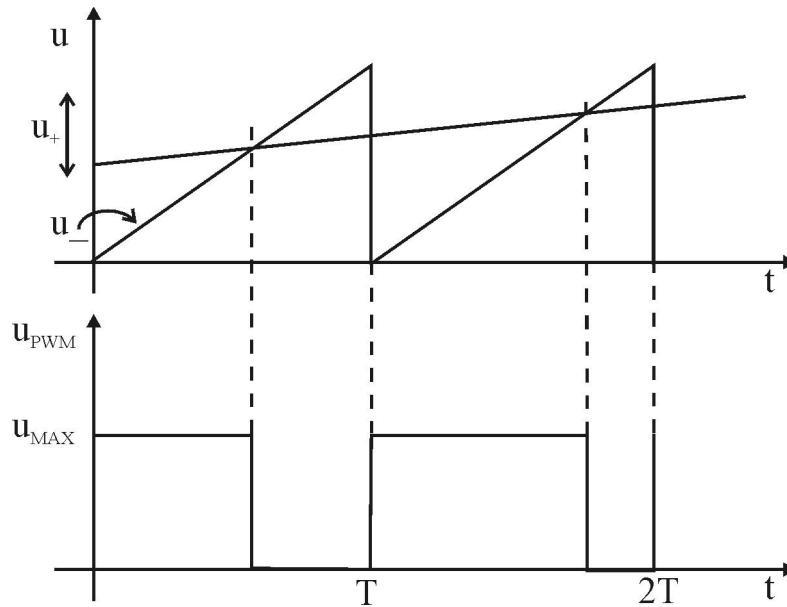
3. Ábra PWM jel generátor

PWM jel generálása megvalósíthatjuk analóg elektronikai eszközökkel, digitális elektronikai eszközökkel (mikrovezérlők perifériáin ez a megoldás az elterjedt) de könnyen le is programozható.

Az *analóg megoldásnál* szükségünk van egy fűrészfog-jel generátorra, egy analóg komparátorra. Ebben az esetben a kitöltés egy  $u_{IN}$  feszültség. Ezt hasonlítjuk össze a fűrészfog-jel generátor. Ha ez a feszültségszint nagyobb, mint a fűrészfogjel értéke, akkor a komparátor kimenete 1, másként a kimenet 0. Tehát a komparátor kimenete egy PWM jel, aminek a kitöltése  $u_{IN}$  feszültséggel lesz arányos.

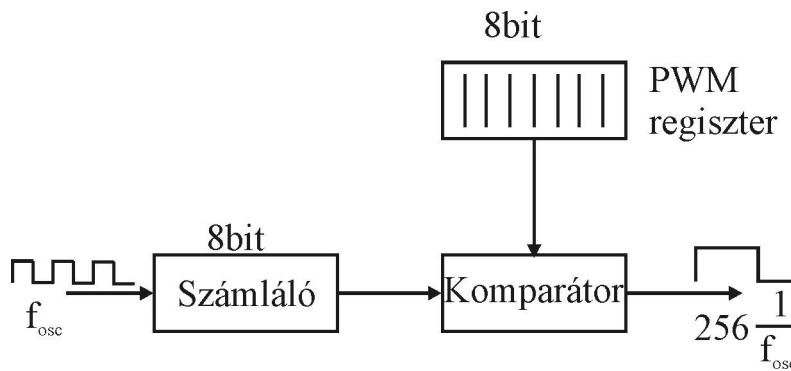


4. Ábra. Analóg PWM jel generátor



5. Ábra A PWM jel generálása

*Digitális áramkörökkel* magvalósított PWM jel esetén a kitöltési tényezőt egy regiszterben tároljuk. Ugyanakkor szükségünk van egy regiszterrel megegyező bithosszúságú számlálóra is. A számláló bemenete egy órajel. Így a számláló értéke folyamatosan nő, majd ha eléri a maximumot, akkor nullázódik majd ismét inkrementálódik. A számláló kimenetét és az órajelet összehasonlítjuk egy digitális komparátorral. A komparátor kimenete PWM jel lesz, aminek a kitöltése egyenlő a regiszterbe beírt értékkel. A PWM jel periódusa: az órajel periódusa szorozva a számláló bithossza.



6. Ábra Digitális PWM jel generálás

*Szoftveres megoldás* esetén egy ciklikus megszakítást alkalmazhatunk. Minden ciklusban növelünk egy változót (*counter*). Tároljuk az  $u$  változóban a PWM kitöltését. Ha a *counter* elért egy maximális értéket (*MAXCOUNTER*), akkor nullázzuk. Ha a *counter* kisebb, mint  $u$  értéke, akkor a DOUT digitális kimeneten egyet, máskülönben nullát küldünk ki.

```

const int MAXCOUNTER=100;
unsigned int counter=0, u;
OnTimer() {
    if (counter==0)
        read (u);
    if(counter<u)
        DOUT=1;
    else
        DOUT=0;
    counter++;

    if (counter == MAXCOUNTER)
        counter=0;
}

```

A PWM jelek segítségével egy digitális kimeneten folytonos módon végezhetünk szabályozást. A PWM jel periódusát úgy kell megválasztani, hogy nagyságrenddel kisebb legyen, mint az irányított folyamat legkisebb időállandója. Így a folyamat bemenetén levő PWM jelnek a hatása a kimeneten csak átlagolva jelenik meg. HA motorvezérlésre szeretnénk használni a PWM jelet, akkor magas jelfrekvenciát kell választani (például 20 kHz), hőmérséklet szabályozás esetén a PWM frekvenciája lehet akár 0.1 – 1 Hz tartományban is. Kemence hőmérséklet szabályozása esetén a PWM jellel egy kapcsolót vezérelhetünk, amely a melegítő ellenállásra vagy rákapcsolja, vagy lekapcsolja a feszültséget.

### 3. A mérés menete

Megoldandó feladatok a *TempControlLab* tervben:

#### 1. A proporcionális szabályozó megvalósítása

A *GetControlSignal* függvényben:

- Lekérdezzük a szabályozási hibát a *GetControlError* függvénnyel
- A (4) összefüggés alapján kiszámoljuk a beavatkozó jelet
- A beavatkozó jelet korlátozzuk a +/-100 tartományra.

#### 2. A PWM jel szoftveres generálása

A *GeneratePWMSignal* függvényben:

- Folyamatosan növelünk egy számláló.
- Ha az értéke eléri a százat (maximális értéket), akkor nullázzuk.
- Beolvassuk a beavatkozó jelet, meghívva a *GetControlSignal* függvényt.
- Ha a beavatkozó jel abszolút értéke nagyobb, mint a számláló értéke akkor az *AUXPORT* 0-s és 1-es bitjén 0 értéket küldünk ki a *cbDOut* függvény segítségével.
- Máskülönb, ha a beavatkozó jel értéke pozitív 1-et küldünk ki a 0s biten (fűtés) és 0-át a 1-es biten. Ha a beavatkozó jel értéke negatív 1-et küldünk ki a 1s biten (hűtés) és 0-át a 0-s biten.

#### 3. Kiértékelés:

- Ábrázoljuk grafikusán a beavatkozó jelet és a hőmérsékletet a *GetTemperatureList* és a *GetControlList* függvényekben.

- Vizsgáljuk meg a hőmérséklet kimenet és a beavatkozó jel változását egymáshoz képest
- Vizsgáljuk meg a hőmérséklet kimenetet különböző  $K_p$  értékekre.

#### ***4. Kérdések és feladatok***

1. Módosítsuk a programban a PWM jel felbontását (számláló maximális értékét) 100-ról 20-ra. Vizsgáljuk a hőmérséklet válasz módosulását.
2. Módosítsuk a P szabályozót PD szabályozásra (a deriválási idő legyen 1-nél kisebb). Milyen változást látunk a szabályozott hőmérsékletjelen?
3. Írjuk át a programot olyan formában, hogy a hűtés kimentetet nem alkalmazzuk. (az AUXPORT második kimenetére mindig zérót küldünk ki). Hogyan változik a rendszer válasza.