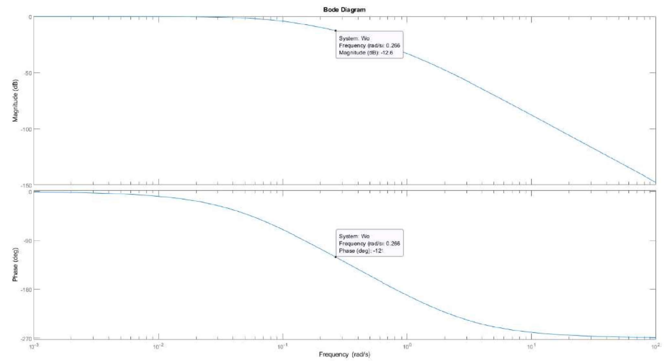
4. Első feladat

4.1. P szabályozó

Az első feladat az volt, hogy adott fázistartalékra kellett készíteni különböző szabályozókat. A P típusúval kezdtem, amire azt lehet mondani, hogy pontatlan és lassú is, viszont a legegyszerűbb szabályozó típus.
A feladatom a megadott időállandók alapján a következő volt:

$$W_{(s)} = \frac{1}{(1+s0.7)*(1+s2.9)*(1+s11.9)} \qquad (2)$$

$\varphi_t = 59^o$ – os fázistartalékra kellett elkészíteni a szabályozót. Ehhez a MATLAB segítségével ki kellett rajzoltatni a függvény BODE diagrammját.

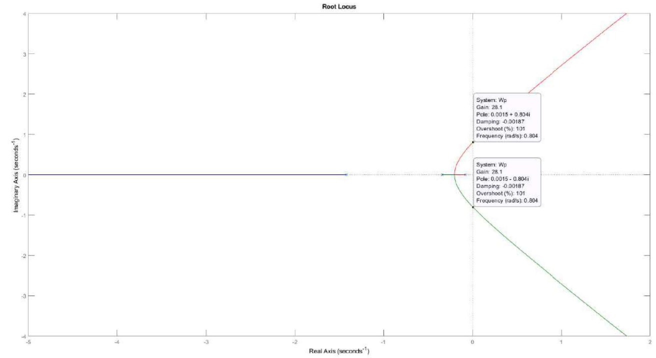


3.ábra. P szabályozó BODE diagramm.

Tehát ha $\varphi_t = 59^o$ – os fázistartalékra kellett megtervezni a szabályozót az azt jelenti, hogy a fázisdiagrammon meg kellett keresni a $-180^{\circ} + 59^{\circ} = -121^{\circ}$ - ot és a hozzá tartozó frekvenciát. Ez ebben az esetben 0.266 $\frac{rad}{sec}$. Ezután ezt a frekvenciát fel kellett vetíteni az amplitúdó diagrammra, ahol meg kellett nézni, hogy ehhez a frekvenciához mekkora erősítés tartozik. A mi esetünkben ez az érték -12.6 dB. Ezt a következő képlettel tudjuk átszámolni lineáris skálára [4,5]:

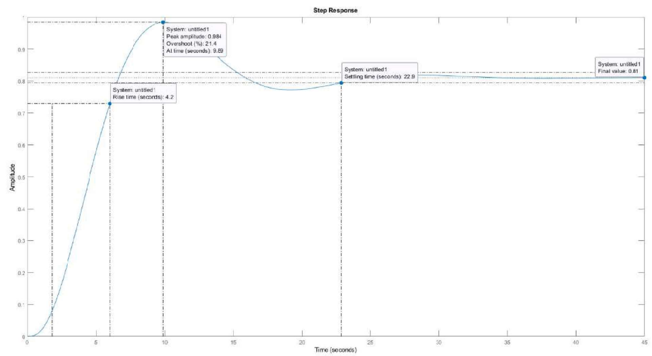
$$K_P = 10^{\frac{0.266}{20}} = 4.2657 \qquad (3)$$

Gyökhelygörbe segítségével meghatároztam, hogy mekkora lehet a maximális Kp érték aminél még nem torzul az ugrásválaszunk és az egész rendszerünk.



4.ábra. P szabályozó gyökhelygörbe.

Itt azt a pontot kellett megkeresni, amikor a függvényünk elmetszi a 0 értéket. A P szabályozónál ez az erősítés érték K_{Pmax} = 28.1. E felett az érték felett nem lesz stabil a szabályozónk. Ezután megvizsgáltam a függvény ugrásválaszát, amelyről sok minden leolvasható [9].



5.ábra. P szabályozó ugrásválasz.

Látható, hogy elég lassú felfutási idővel és nagy túllövéssel rendelkezik és stacionárius állapotban is elég nagy eltérés van az elvárt „1” értéktől [4,5].

4.2. PI szabályozó

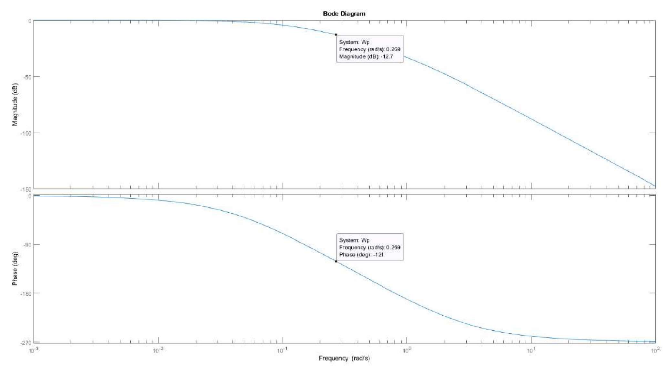
Ennél a szabályozónál egy kis módosításra van szükség ahhoz, hogy póluskiejtést tudjak alkalmazni. Ez annyit jelent, hogy a megadott időállandókból a „leglomhábbat” ki kell ejteni. T_I = 11.9 [6].

$$W_c(s) = K_{PI} * \frac{1+sT_i}{sT_i} \quad (4)$$

Miután elvégeztem a szükséges módosítást ezután a függvényem így nézett ki:

$$W_o = K_{pi} \frac{1}{11.9s(1+0.7s)(1+2.9s)} \quad (5)$$

Ezután ugyanúgy kell eljárnom, mint a P szabályozó esetén. Először tehát megvizsgálom a BODE diagrammot és leolvasom a fázisdiagramról, hogy a -121°- hoz milyen érték tartozik.

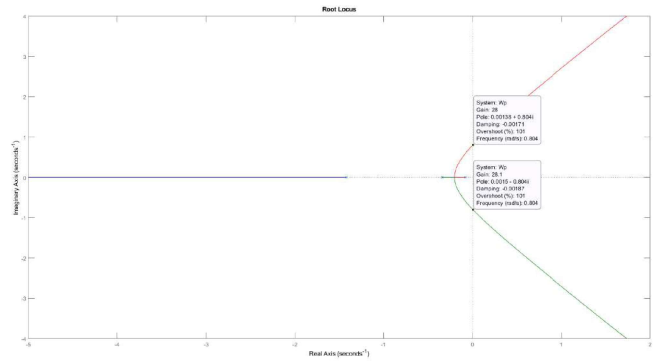


6.ábra. PI BODE diagramm.

Miután megkerestem a -121°- ot, leolvastam a hozzá tartozó frekvenciát, ami 0.269 $\frac{rad}{sec}$. Ezután ezt az értéket felvetítve az amplitúdó karakterisztikára megkerestem, hogy mekkora dB

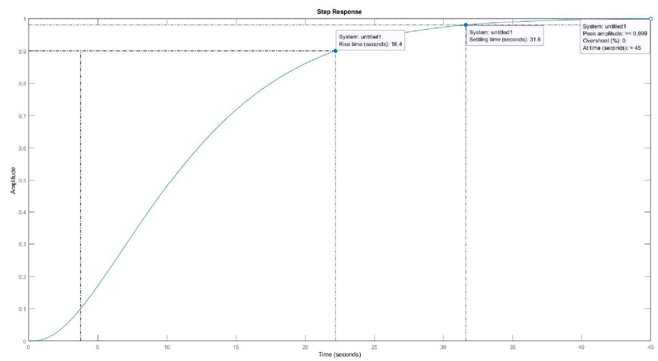
érték tartozik hozzá. Ez az érték – 12.7 dB lett. Az előzőhöz hasonlóan visszaszámoltam K_{PI} értéké, ami 4.3152 lett.

A gyökhelygörbén ismét megvizsgáltam, hogy mekkora az a maximális erősítés érték, aminél még az szabályozónk stabil marad.



7.ábra. PI gyökhelygörbe.

Megkeresve azt a pontot ahol a függvényünk a „0” pontban metszi a tengelyt leolvasható hogy K_{Pmax} = 28.



8.ábra. PI ugrásválasz.

Az előző ugrásválasszal összehasonlítva látható, hogy az integráló tag hatására nagyon pontos lett a szabályozónk, hiszen miután beáll stacionárius állapotba felveszi a kívánt „1” értéket, viszont ennek az az ára, hogy sokkal lassabb felfutási idővel rendelkezik [6].

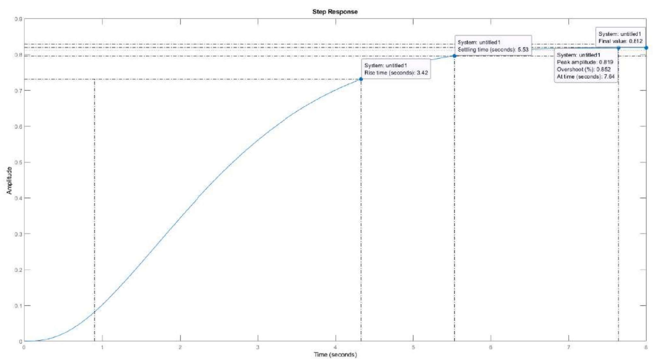
4.3. PD szabályozó

$$W_c(s) = K_{PD} * \frac{1+sT_D}{sT'_D} \qquad (6)$$

Látható hogy a PD szabályozót majdnem ugyan úgy kell megtervezni, mint a PI szabályozót, annyi módosítással, hogy ebben az esetben a leglomhább tagot kell kiejteni. Tehát sT_D értéke 2.9, sT'_D értéke pedig egy dekáddal kisebb vagyis 0.29 lesz. Így a következő függvényt kapjuk meg [7]:

$$W_{(s)} = K_{PD} * \frac{1}{(1+s0.7)*(1+s0.29)*(1+s11.9)} \qquad (7)$$

Innentől kezdve az előzőekhez hasonlóan kell eljárnunk, amit már nem részleteznék külön. A BODE diagramról megkaptam, hogy K_{PD} értéke 4.3152, a gyökhelygörbéről pedig, hogy a maximális erősítés értéke 28. Ezután megvizsgáltam az ugrásválaszt, amely a következőképp nézett ki:



9.ábra. PD ugrásválasz.

Látható, hogy egy differenciáló tag hozzáadásával ez a leggyorsabb szabályozó, ami felveszi a stacionárius értéket, viszont ennek az az ára, hogy ez sem pontos a P szabályozóhoz hasonlóan.

4.4. PID szabályozó

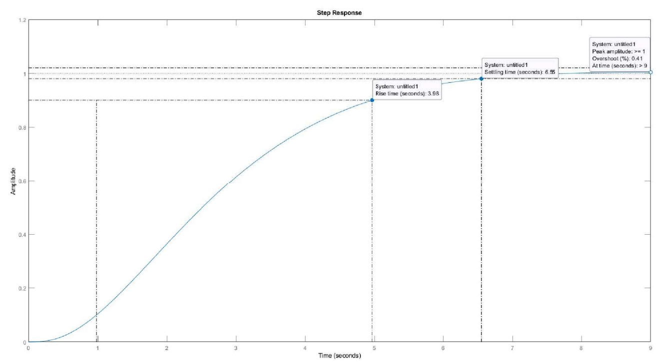
Ez a szabályozó tulajdonképpen az eddigi szabályozók összes tulajdonságával rendelkezik. Ebből az is következik, hogy mindkét módosítást egyszerre kell végrehajtani, hogy a megfelelő függvényt kapjuk. Tehát a leglomhább és a második leglomhább tagot is ki kell ejteni [8].

$$W_c(s) = K_{PID} * \frac{1+sT_i}{sT_i} * \frac{1+sT_D}{sT'_D} \qquad (8)$$

Miután ezt a módosítást elvégeztük a végső függvény a következőképp néz ki:

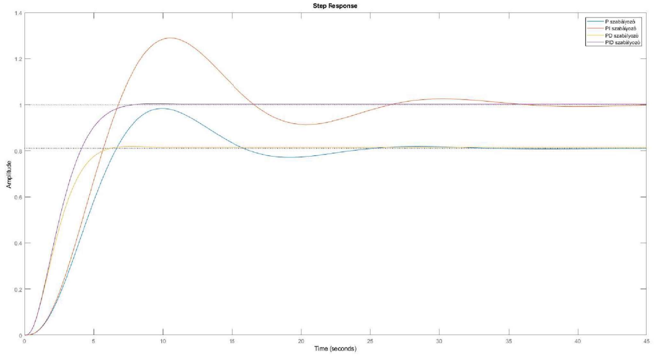
$$W_{(s)} = K_{PID} * \frac{1}{s11.9*(1+s0.7)*(1+s0.29)} \qquad (9)$$

Innentől kezdve az ugyanúgy járok el, mint eddig. Először is a kirajzolt BODE diagrammról leolvasom a megfelelő értékeket és ezekből kiszámolom az erősítés értékét, amit 4.3152 lett. Ezután A gyökhelygörbe segítségével ennél a szabályozónál is meghatározom a maximális erősítés értéket, amelynél a még stabil marad. Ez az érték 28. Az ugrásválaszról látható hogy valóban ez a legjobb szabályozó.



10.ábra. PID ugrásválasz.

Tehát kicsi felfutási idővel rendelkezik, nincs túllövés és stacionárius állapotban pontosan az „1” értékhez simul. Azért, hogy még jobban lehessen szemléltetni a négy szabályozó közti különbséget felrajzoltam egy ábrára az összeset.



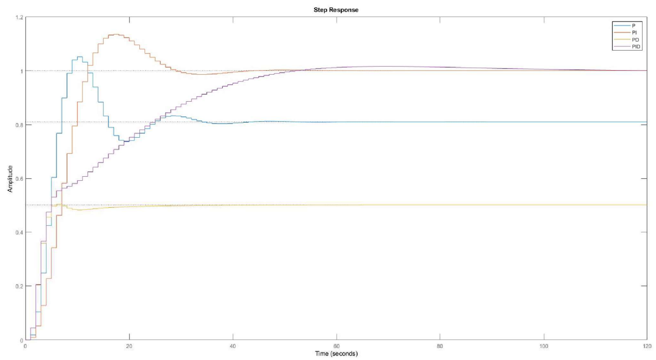
11.ábra. Összes ugrásválasz.

Összefoglalva itt egy ábrán látható, amit eddig megfigyelhettünk. A kék vonallal jelzett ugrásválasz a P szabályozó, ami nem is veszi fel az „1” értéket, de még lassú felfutási idővel és nagy túllövéssel rendelkezik. A narancssárga vonal a PI szabályozót jelöli, aminél látható, hogy valamennyivel gyorsabb és felveszi a kívánt értéket viszont sokkal nagyon túllövéssel rendelkezik. A PD szabályozót a citromsárga vonal jelöli, ami gyors és egyáltalán nem rendelkezik túllövéssel, viszont nem pontos. Végül a lila vonal a PID szabályozót jelöli. Ez a leggyorsabb és legpontosabb típus és természetesen ez a szabályozó sem rendelkezik túllövéssel. Végül egy táblázatban összefoglaltam a hibákat, így számokkal is látható a különbség az egyes típusok között.

	P	PI	PD	PID
Dinamikus hiba	0.215	0.29	0.03	0
Statikus hiba	0.19	0	0.188	0
Túllövés	21.4%	29%	0.852%	0.41%
2%-os beállási idő	22.9s	32.4s	5.53s	6.55s
Felfutási idő	4.2s	4.27s	3.42s	3.98s

5. Második feladat

Ez a feladat arról szólt, hogy diszkrétté kellett tenni az előző feladat jeleit és úgy összehasonlítani őket. Ezt a program segítségével könnyedén megtehettük, ugyanis csak a c2d parancsot kellett beírni a függvény elé és az ugrásválasztan is ugyanezt kellett tenni [3].



12.ábra. Összes ugrásválasz.

Ezen kívül még meg kellett adni a mintavételezési időt Ez azt határozza meg, hogy mennyire legyen „sima” a jel. Minél kisebb mintavételezési időt választunk, annál jobban fog hasonlítani a folytonos jelhez.

	P	PI	PD	PID
Dinamikus hiba	0.29	0.13	0.004	0.02
Statikus hiba	0.19	0	0.5	0
Túllövés	29.7%	13.5%	0.496%	1.55%
2%-os beállási idő	30.5s	26.3s	15.9s	45.4s
Felfutási idő	4.04s	7.58s	2.92s	31.2s

6. Harmadik feladat

Ebben a feladatban is diszkrét idejű szabályozót kellett készíteni, viszont nem folytonos hanem diszkrét idejű modellt feltételezve, szintén adott fázistartalékra, ahol W(z) előállítása W(s) alapján történik a kiválasztott mintavételi periódusidőt felhasználva. A $W_{(S)}$ függvény diszkrét alakját úgy tudtam meghatározni, hogy Command Window-ba beírtam a Wp_d parancsot és kiadta a következő függvényt [3]:

```
>> Wp_d

Wp_d =

  0.004471 z^2 + 0.01163 z + 0.001774
  -----------------------------------
  z^3 - 1.867 z^2 + 1.041 z - 0.1561

Sample time: 1 seconds
Discrete-time transfer function.
```

13.ábra. Wp_d függvény.

W(z) függvény meghatározásához szükség volt a feladatban megadott 3 idő időtagra.

T1 = 0.7s T2 = 2.9s T3 = 11.9s

Ezután a nevezőt teljes szorzattá kellett alakítani, hogy el tudjam végezni az első feladatban alkalmazott póluskiejtést.

$$W_{PI(Z)} = K_{PI} * \frac{Z-Z_I}{Z-1} \quad (10)$$

Ebből Z_I értéke a következőképp határozható meg:

$$Z_I = e^{-\frac{1}{11.9}} = 0.9194 \quad (11)$$

Ezzel a módszerrel a többi tag is kifejezhető:

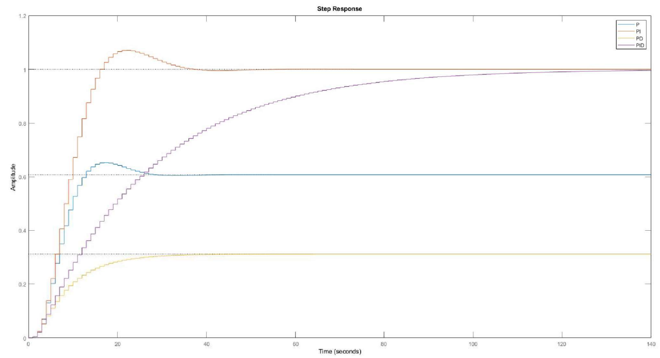
$$W_{PD(Z)} = K_{PD} * \frac{Z-Z_D}{Z} \quad (12)$$

$$Z_D = e^{-\frac{1}{2.9}} = 0.7083 \quad (13)$$

Ezekkel az adatokkal végül így néz ki a $W_{(Z)}$ függvény:

$$W_{(Z)} = \frac{0.004471 * z^2 + 0.01163 * z * 0.1163}{(z-0.9194)*(z-0.7083)*(z-0.2397)} \quad (14)$$

Innentől kezdve a megoldás menete ugyan az mint az első feladatnál. Minden egyes szabályozónál meg kellett vizsgálni a BODE diagrammot és onnan meghatározni az erősítés értékeket.



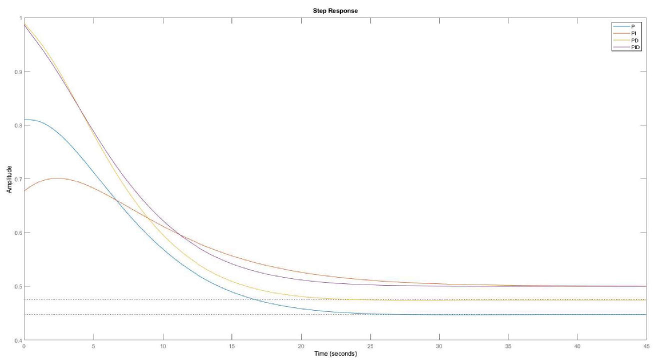
14.ábra. Összes ugrásválasz

Itt látható, hogy valamennyire eltérő értékeket kapunk, mint az első feladatban. Az eltéréseket az alábbi táblázat mutatja be.

	P	PI	PD	PID
Dinamikus hiba	0.074	0	0	0.01
Statikus hiba	0.392	0	0.689	0
Túllövés	7.39%	0%	0%	0%
2%-os beállási idő	24.2s	137s	30.8s	100s
Felfutási idő	7.72s	56.7s	16.5s	55.5s

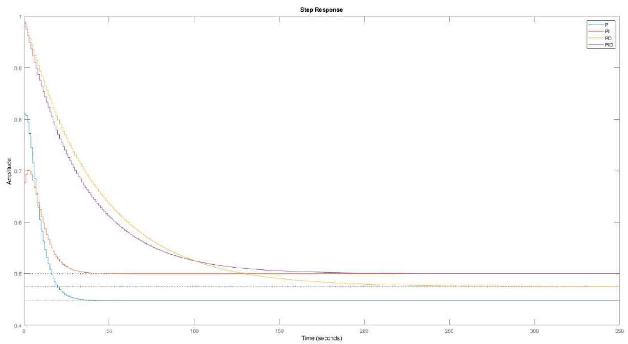
7. Negyedik feladat

Ebben a feladatban módosítani kellett az első és második feladatban kapott eredményeket, úgy hogy a beavatkozó jel ne legyen nagyobb az előírt u(0) = 5.2 értéknél [2].



15.ábra. Folytonos beavatkozó jelek.

Látható az (*ábra*) alapján hogy egyik beavatkozó jel sem haladja meg az előírt 5.2 értéket így egyik szabályozónál sem volt szükség módosításra.



16.ábra. Diszkrét beavatkozó jelek.

A diszkrét beavatkozó jelek sem haladják meg az előírt értéket, így itt sem volt arra szükség, hogy módosítsuk a feladatot.

Forráskód

Első feladat

```
clc;clear all
s = tf('s');
Wp = 1 / ((1+0.7*s)*(1+2.9*s)*(1+11.9*s));
%% 1_feladat
%% P SZABÁLYZÓ
%Kp = 1;
Kp = 4.2657;
Wo = Kp * Wp;
bode(Wp);
rlocus(Wp);
stepplot(feedback(Wo, 1));
%% PI SZABÁLYZÓ
%Kpi = 1;
Kpi = 4.3152;
PI = ((1 + 11.9 * s) / (11.9 * s));
Wo = PI * Wp;
bode(Wp);
rlocus(Wp);
stepplot(feedback(Wo, 1));
%% PD szabályzó
%Kpd = 1;
Kpd = 4.3152;
PD = ((1 + 2.9 * s) / (1 + 0.29 * s));
Wo = Kpd * PD * Wp;
bode(Wp);
rlocus(Wp);
stepplot(feedback(Wo, 1));
%% PID szabályzó
%Kpid = 1
Kpid = 4.3152;
PID = ((1 + 11.9 * s) / (11.9 * s)) * ((1 + 2.9 * s) / (1 + 0.29 * s));
Wo = Kpid * PID * Wp;
bode(Wp);
rlocus(Wp);
stepplot(feedback(Wo, 1));
%% 1. Feladat vége
stepplot(feedback(Kp * Wp, 1), feedback(Kpi * PI * Wp, 1), feedback(Kpd * PD * Wp, 1), feedback(Kpid * PID * Wp, 1));
```

Második feladat

```
%%  2. Feladat
Ts = 1;
Wp_d = (c2d(Wp , Ts));
%%
stepplot(feedback(Kp * Wp_d, 1));
%%
stepplot(feedback((c2d(PI, Ts)) * Kpi * Wp_d, 1));
%%
stepplot(feedback((c2d(PD, Ts)) * Wp_d, 1));
%%
stepplot(feedback((c2d(PID, Ts)) * Wp_d, 1));
%%
stepplot(feedback(Kp * Wp_d, 1), feedback((c2d(PI, Ts)) * Kpi * Wp_d, 1), feedback((c2d(PD, Ts)) * Wp_d, 1), feedback((c2d(PID, Ts)) * Wp_d, 1));
```

Harmadik feladat

```
%% 3 Feladat
clc; clear all
z = tf('z');
Wz = ((0.004471 * z^2) + (0.01163 * z) + (0.01163)) / ((z- 0.9194) * (z - 0.7083) * (z -0.2397));
%% P szabályzó
Kp = 1;
%Kp = 912.0108;
Wo = Kp * Wz;
%%
bode(Wz);
%%
stepplot(feedback(Wz, 1));
%% PI szabályzó
%Kpi = 1;
Kpi = 912;
PI_d = (z - 0.9194) / (z - 1);
Wo = PI_d * Wz;
bode(Wo);
stepplot(feedback(PI_d * Wz, 1));
%% PD szabályzó
%Kpd = 1;
Kpd = 543.25;
PD_d = (z - 0.7083) / z;
Wo = PD_d * Wz;
bode(Wo);
stepplot(feedback(PD_d * Wz, 1));
%% PID szabályzó
%Kpid = 1;
Kpid = 562.34;
PID_d = ((z - 0.9194) / (z - 1)) * ((z - 0.7083) / z);
Wo = PID_d * Wz;
bode(Wo);
stepplot(feedback(PID_d * Wz, 1));
%% 3. Feladat vége
stepplot(feedback(Wz, 1), feedback(PI_d * Wz, 1), feedback(PD_d * Wz, 1), feedback(PID_d * Wz, 1));
```


Negyedik Feladat

```
%% 4. Feladat
% Első feladat kiegészítése
clc;clear all
s = tf('s');
Wp = 1 / ((1+0.7*s)*(1+2.9*s)*(1+11.9*s));
%% P SZABÁLYZÓ
%Kp = 1;
Kp = 4.2657;
Wo = Kp * Wp;
Wc = Kp;
Up = Wc / (1 + Wc * Wp);
stepplot(feedback(Up, 1));
%% PI SZABÁLYOZÓ
%Kpi = 1;
Kpi = 2.0965;
PI = ((1 + 11.9 * s) / (11.9 * s));
Wo = PI * Wp;
Wc = PI * Kpi;
Upi = Wc / (1 + Wc * Wp);
stepplot(feedback(Upi, 1));
%% PD SZABÁLYOZÓ
%Kpd = 1;
Kpd = 9.5499;
PD = ((1 + 2.9 * s) / (1 + 0.29 * s));
Wo = Kpd * PD * Wp;
Wc = PD * Kpd;
Upd = Wc / (1 + Wc * Wp);
stepplot(feedback(Upd, 1));
%% PID SZABÁLYOZÓ
%Kpid = 1
Kpid = 7.4989;
PID = ((1 + 11.9 * s) / (11.9 * s)) * ((1 + 2.9 * s) / (1 + 0.29 * s));
Wo = Kpid * PID * Wp;
Wc = PID * Kpid;
Upid = Wc / (1 + Wc * Wp);
stepplot(feedback(Upid, 1));
%% Beavatkozó jelek
stepplot(feedback(Up, 1), feedback(Upi, 1), feedback(Upd, 1), feedback(Upid, 1));
%% Második feladat kiegészítése
Ts = 1;
Wp_d = (c2d(Wp, Ts));
%% Beavatkozó jelek
stepplot(feedback(c2d(Up, Ts), 1), feedback(c2d(Upi, Ts), 1), feedback(c2d(Upd, Ts), 1), feedback(c2d(Upid, Ts), 1));
```