ELEC-C1230 Säätötekniikka/Kotitehtävä 5 Ratkaisut

Tehtävä Prosessia

$$G(s) = \frac{1}{(s+1)(s+2)}$$

säädetään PD-säätäjällä $G_c(s) = K(1+Ts)$. Viritä säätäjän parametrit K ja T siten, että kun referenssi on askelfunktio, niin pysyvä poikkeama on korkeintaan 0.02, asettumisaika (2%) on korkeintaan 0.6 sekuntia, ja ylitys on korkeintaan 10%. Simuloi suljettua järjestelmää ja tarkista, että spesifikaatiot toteutuvat. Viritä PD-säätäjä vertailun vuoksi myös Matlabin komennolla *pidtune*, ja simuloi suljettua järjestelmää. Eroavatko tulokset oleellisesti omasta virityksestäsi? Piirrä vielä avoimen järjestelmän Boden diagrammi (eri virityksillä) ja määritä vaihe- ja vahvistusvarat.

Huom. Tehtävä on ratkaistava hyödyntäen luentokalvojen Luvussa 8 esitettyjä aikatason kriteereitä. Pelkkä kokeileminen säätäjän eri viritysarvoilla ei ole ratkaisu. Huomaa, että tavoitetellut kriteerit eivät välttämättä toteudu tarkalleen.

Huom. Matlabin komentoja: *tf, feedback, step, stepinfo, margin*

Ratkaisu Kun PD-säätäjä kytketään prosessiin, niin avoimen silmukan siirtofunktio on

$$G_{OL} = G_c(s)G(s) = \frac{K(1+Ts)}{(s+1)(s+2)}$$

ja suljetun järjestelmän siirtofunktio on

$$G_{CL} = \frac{G_c(s)G(s)}{1+G_c(s)G(s)} = \frac{K(1+Ts)}{(s+1)(s+2)+K(1+Ts)} = \frac{K(1+Ts)}{s^2+(3+KT)s+2+K}.$$

Suljettu järjestelmä on toista kertalukua, joten kirjoitetaan sen siirtofunktion nimittäjä muotoon $s^2 + 2\zeta\omega_n + \omega_n^2$. Asettamalla s:n potenssien kertoimet yhtä suuriksi saadaan ratkaistua

$$\begin{cases} 2\zeta\omega_n & = (3+KT) \\ \omega_n^2 & = 2+K \end{cases}, \Rightarrow \omega_n = \sqrt{2+K}, \Rightarrow \zeta = \frac{3+KT}{2\omega_n} = \frac{3+KT}{2\sqrt{2+K}}. \tag{1}$$

Jos jätetään huomioimatta, että suljetulla järjestelmällä on myös nolla (kohdassa $s = -1/T$), niin voidaan ratkaista tarvittava vaimennussuhde ζ, jolla ylitys on korkeintaan 10%. Luku 8 s. 12 mukaisesti

$$\begin{aligned}
& 100e^{-\frac{\pi\zeta}{\sqrt{1-\zeta^2}}} \leq 10 \\
\Leftrightarrow\quad & 10e^{-\frac{\pi\zeta}{\sqrt{1-\zeta^2}}} \leq 1 \\
\Leftrightarrow\quad & \ln(10) - \frac{\pi\zeta}{\sqrt{1-\zeta^2}} \leq \ln(1) \\
\Leftrightarrow\quad & \frac{\pi\zeta}{\sqrt{1-\zeta^2}} \geq \ln(10) \\
\Leftrightarrow\quad & \pi\zeta \geq \ln(10)\sqrt{1-\zeta^2} \\
\Leftrightarrow\quad & \pi^2\zeta^2 \geq \ln(10)^2(1-\zeta^2) \\
\Leftrightarrow\quad & \zeta^2 \geq \frac{\ln(10)}{\pi^2+\ln(10)^2},
\end{aligned}$$

josta saadaan $\zeta \geq 0.5912$. Tästä voidaan ratkaista luonnollinen kulmataajuus ω_n asettumisajan perusteella (Luku 8, s. 15)

$$\begin{aligned} 0.6 &\approx \frac{4}{\zeta\omega_n} \\ \Leftrightarrow \quad \omega_n &\approx \frac{4}{0.6\zeta} \approx 11.2774. \end{aligned}$$

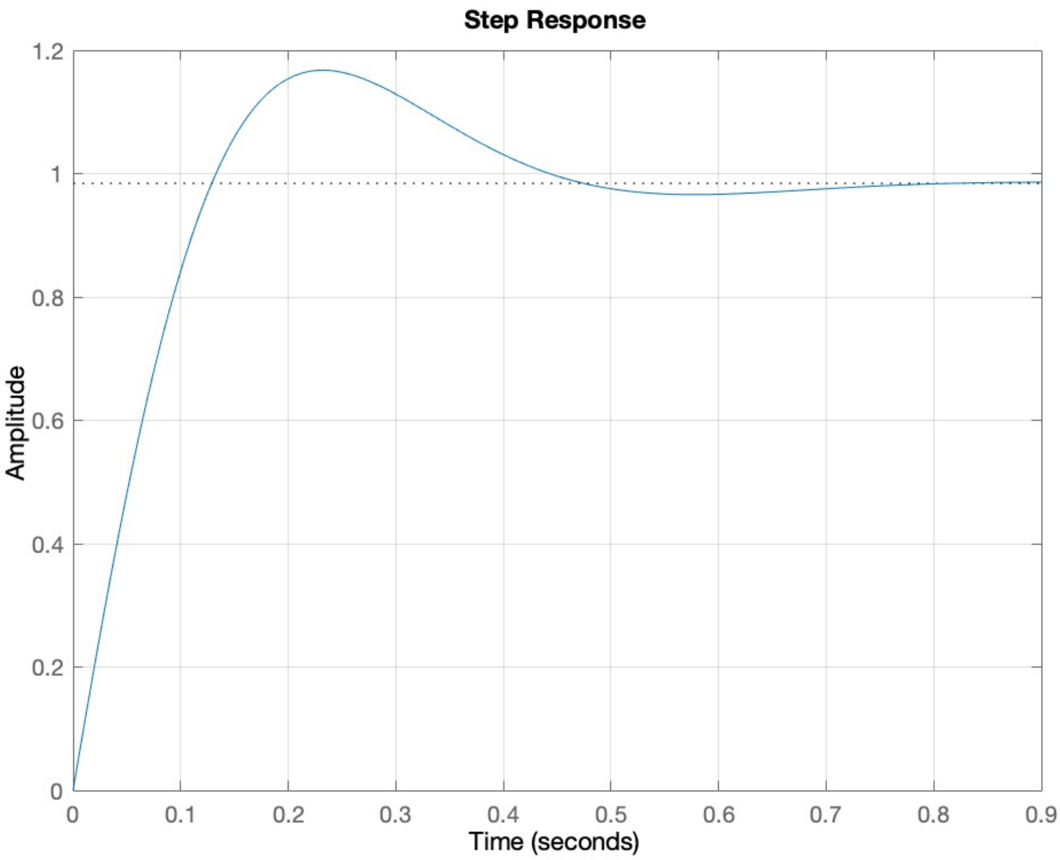
Näistä saadaan arvot säätäjän parametreille K ja T kaavasta (1), eli

$$\begin{aligned} K &= \omega_n^2 - 2 \approx 125.1788 \\ T &= \frac{2\zeta\sqrt{2+K} - 3}{K} \approx 0.0825. \end{aligned}$$

Lasketaan vielä pysyvä poikkeama (Luku 8, s. 21)

$$\frac{1}{1 + G_{OL}(0)} = \frac{1}{1 + \frac{K}{2}} \approx 0.0157 \leq 0.02,$$

eli kaiken pitäisi olla kunnossa. Kuvan 1 askelvasteen perusteella ylitys näyttää kuitenkin liian suurelta, ja Matlabin komennolla *stepinfo* saadaankin, että ylitys on yli 18%, vaikka asettumisaika onkin annetuisssa rajoissa (alle 0.5 sekuntia).



Kuva 1: Askelvaste, kun säätäjä on viritetty huomioimatta suljetun järjestelmän nollaa.

Kokeillaan virittää säätäjä uudelleen siten, että huomioidaan suljetun järjestelmän nolla. Kirjoitetaan suljetun järjestelmän siirtofunktion osoittaja $K(1+Ts)$ tätä varten muotoon (Luku 8, s. 16)

$$\widetilde{K}\omega_n^2\left(\frac{s}{\rho\omega_n}+1\right).$$

Luonnoliseksi kulmataajuudeksi ω_n oli jo aiemmin määritetty $\omega_n = \sqrt{2+K}$, joten parametrille ρ saadaaan lauseke

$$T = \frac{1}{\rho\omega_n} \Rightarrow \rho = \frac{1}{T\omega_n} = \frac{1}{T\sqrt{2+K}},$$

ja $\widetilde{K}$ on suljetun järjestelmän staattinen vahvistus, eli

$$\widetilde{K} = \lim_{s\to 0} G_{CL}(s) = \frac{K}{2+K} = \frac{K}{\omega_n^2}.$$

Aloitetaan virittäminen tällä kertaa siitä, että pysyvä poikkema saa olla korkeintaan 0.02. Tästä saadaan ratkaistua K (Luku 8, s. 21):

$$\frac{1}{1+\frac{K}{2}} \leq 0.02 \Leftrightarrow 1+\frac{K}{2} \geq \frac{1}{0.02} \Rightarrow K \geq 98.$$

Jos asetetaan $K = 98$, niin $\omega_n = \sqrt{2+K} = 10$. Nyt voidaan hyödyntää Luvun 8 sivun 19 taulukkoa, jossa vahvennettujen rajojen sisällä on alue, jolla ylitys on korkeintaan 10% ja asettumisaika (2%) on korkeintaan $6/\omega_n = 6/10 = 0.6$. Huomataan kuitenkin, että

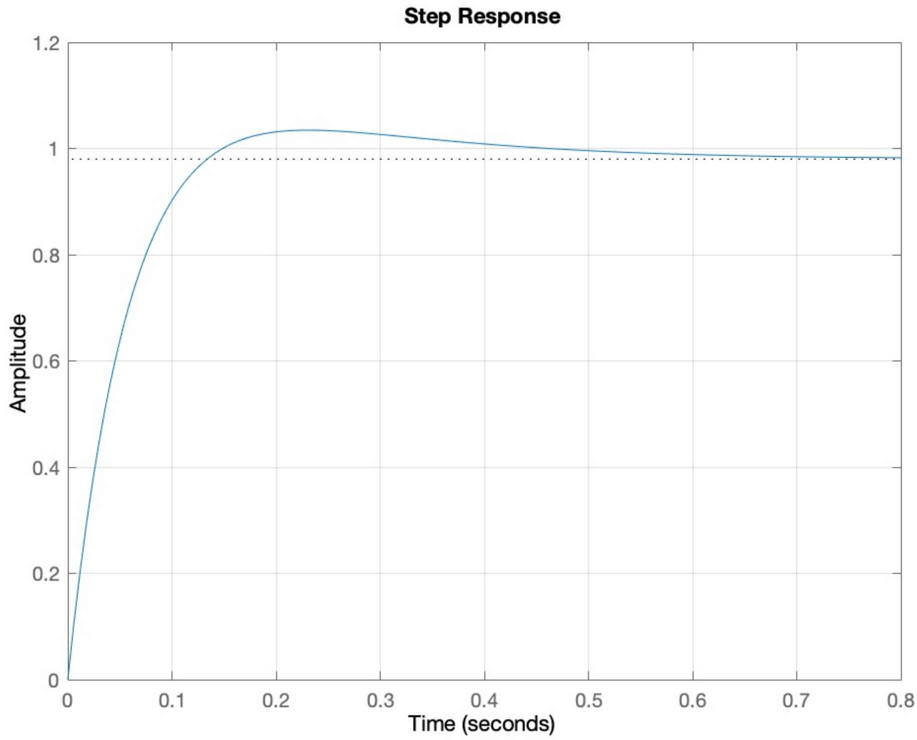
$$\zeta = \frac{3+KT}{2\sqrt{2+K}} \quad \text{ja} \quad \rho = \frac{1}{T\sqrt{2+K}}, \tag{2}$$

eli ζ ja ρ arvoja ei voida asettaa mielivaltaisesti. Taulukon vahvennetulle alueelle päästään kuitenkin esimerksi siten, että valitaan $\zeta = 1.1$, jolloin saadaan ratkaistua

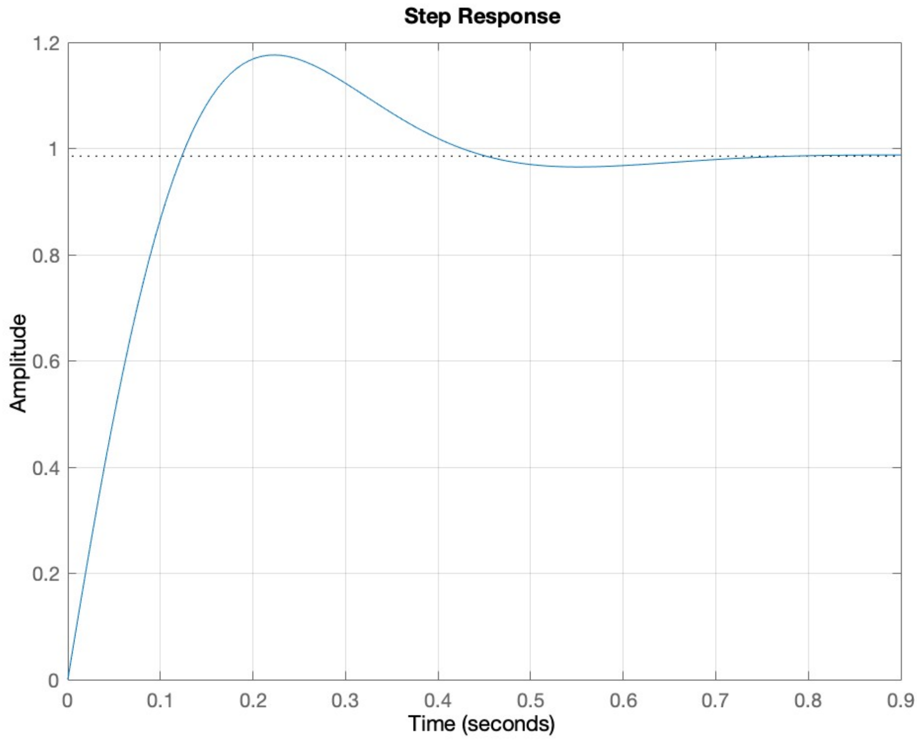
$$T = \frac{2\zeta\sqrt{2+K}-3}{K} \approx 0.1939$$

ja tällöin ρ:n arvoksi tulee $\rho \approx 0.5158$, eli ollaan vahvennetulla alueella, ja spesifikaatioiden pitäisi toteutua. Tämä voidaan vielä tarkistaa simuloimalla (Kuva 2) sekä Matlabin komennolla *stepinfo*, jonka mukaan asettumisaika on alle 0.5 sekuntia ja ylitys alle 6%.

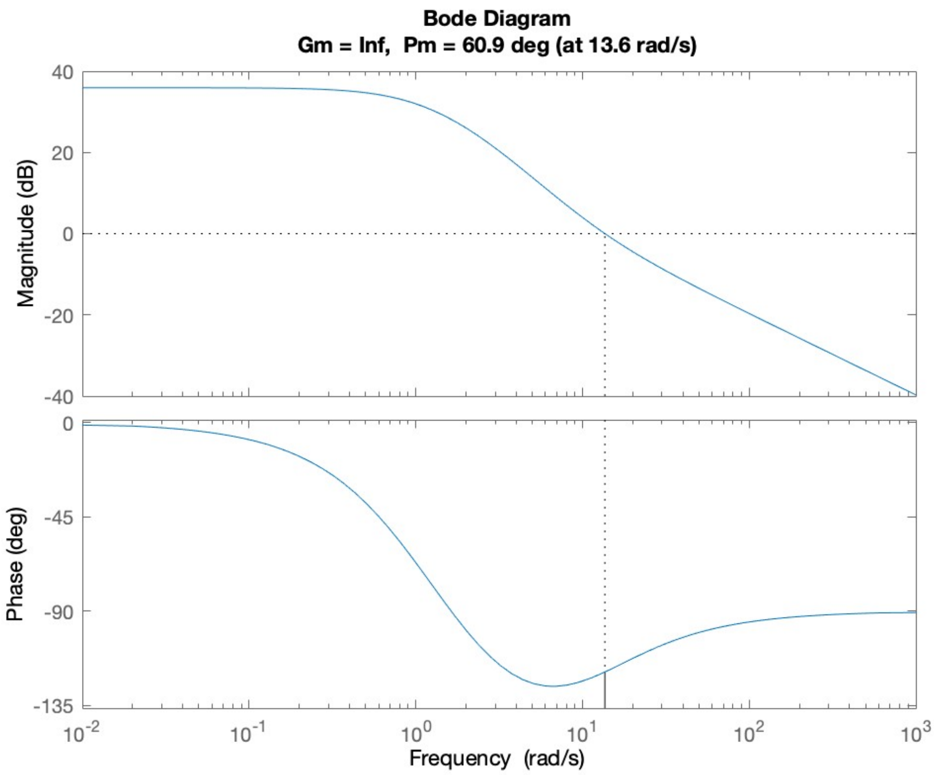
Viritetään vertailun vuoksi PD-säätäjä vielä Matlabin komennolla *pidtune* (Matlab-koodi dokumentin lopussa). Säätäjän parametreiksi saadaan $K \approx 135.7423$ ja $T \approx 0.0783$, jotka ovat suhteellisen lähellä arvoja, jotka saatiin virittämällä säätäjä huomioimatta suljetun järjestelmän nollaa. Askelvasteen simulaatio on esitettynä Kuvassa 3, josta nähdään, että säätimen suorituskyky on myös samaa luokkaa. Matlabin komennolla *stepinfo* itse asiassa nähdään, että *pidtune*-säätimellä ylitys on yli 19% prosenttia, ja myös asettumisaika on pidempi kuin omavirittämillä säätimillä (joskin sittenkin alle 0.6 sekuntia). Pitää toki huomioida, että Matlabin *pidtune* ei viritä säätäjää minkään annettujen kriteerien perusteella, vaan pyrkii yleisesti ottaen kohtuulliseen suorituskykyyn ja robustisuuteen. Tosin Matlabin komennolla *margin* nähdään (Kuvat 4–6), että *pidtune*n virittämän säätäjän vaihevara on näistä kolmesta virityksestä pienin (vahvistusvara on tässä tapauksessa aina ääretön), joten ei *pidtune*n viritys erityisen robustikaan ole.



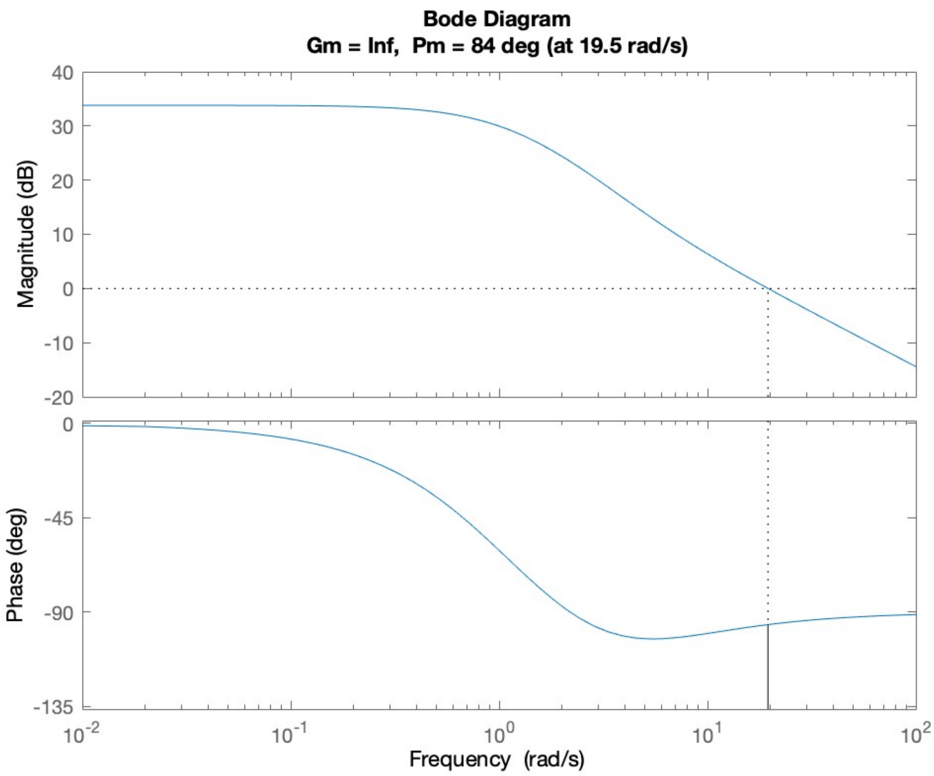
Kuva 2: Askelvaste, kun säätäjä on viritetty suljetun järjestelmän nolla huomioiden.



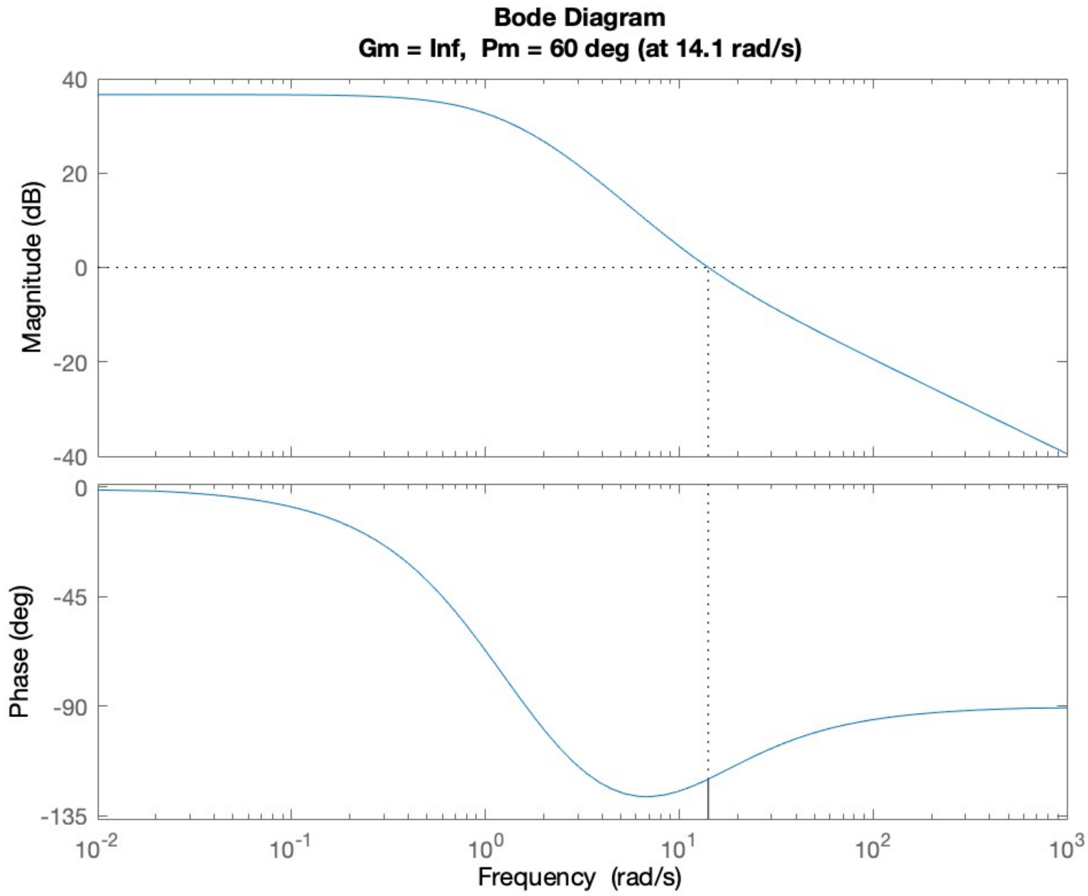
Kuva 3: Askelvaste, kun säätäjä on viritetty Matlabin komennolla pidtune.



Kuva 4: Boden diagrammi, kun säätäjä on viritetty huomioimatta suljetun järjestelmän nollaa.



Kuva 5: Boden diagrammi, kun säätäjä on viritetty suljetun järjestelmän nolla huomioiden.



Kuva 6: Boden diagrammi, kun säätäjä on viritetty Matlabin komennolla pidtune.

```matlab
%% MATLAB-KOODI
% prosessi ja virityksen spesifikaatiot
G = zpk([],[-1 -2],1); % prosessi
tsd = 0.6; % haluttu asettumisaika
poik = 0.02; % sallittu poikkeama
osp = 10; % sallittu ylitys
%% tapa 1 (ei huomioida suljetun jarjestelman nollaa)
% ratkaistaan vaimennus ylityksen perusteella
z10 = log(osp)/sqrt(pi^2+log(osp)^2);
wz = 4/(z10*0.6); % taajuus asettumisajan perusteella
% saatajan parameterit lasketuista arvoista
K1 = wz^2 - 2
T1 = (2*z10*sqrt(2+K1)-3)/K1
pp = 1/(1+K1/2) % tarkistetaan poikkeama (ok)
Gc1 = tf(K1*[T1 1],1); % PD saataja (siirtofunktio)
GCL1 = feedback(Gc1*G,1); % suljettu jarjestelma
step(GCL1) % simuloidaan askelvaste
grid on
stepinfo(GCL1) % askelvasteen info -> liian suuri ylitys (18%)
%% tapa 2 (huomioidaan suljetun jarjestelman nolla)
K2 = 2*(1/poik-1); % K poikkeaman perusteella
zd = 1.1; % haluttu vaimennus
T2 = (2*zd*sqrt(2+K2)-3)/K2 % T vaimennuksen perusteella
rho = 1/(T2*sqrt(2+K2)) % tarkistetaan rho (ok)
Gc2 = tf(K2*[T2 1],1); % PD saataja (siirtofunktio)
GCL2 = feedback(Gc2*G,1); % suljettu jarjestelma
step(GCL2) % simuloidaan askelvaste
grid on
stepinfo(GCL2) % askelvasteen info -> ok
%% Tapa 3 (pidtune)
PD = pidtune(G,'PD') % viritetaan PD
PD.Kd/PD.Kp % T:ta vastaava arvo
GCL3 = feedback(PD*G,1); % suljettu jarjestelma
step(GCL3) % simuloidaan askelvaste
grid on
stepinfo(GCL3) % askelvasteen info -> ylitys yli 19%
%% Vaihe- ja vahvistusvarat Boden diagrammista
margin(G*Gc1)
margin(G*Gc2)
margin(G*PD)
```