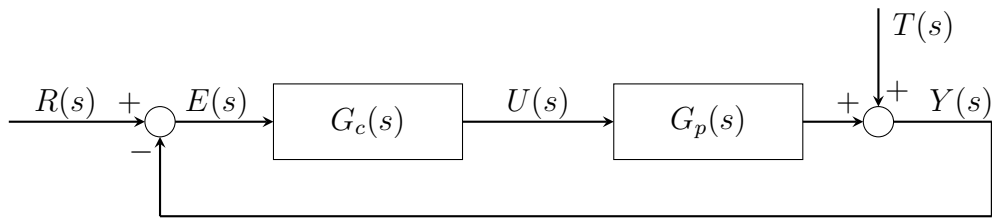


ELEC-C1230 Säättötekniikka/Kotitehtävä 2 Ratkaisut

Tarkastellaan kuvassa esitettyä säätöpiiriä



jossa prosessin siirtofunktio on $G_p(s) = \frac{2}{s^2 + 4s + 5}$. Säättäjänä kokeillaan erikseen P-säättäjää $G_{c1}(s) = K_P$ ja PI-säättäjää $G_{c2}(s) = K_P \left(1 + \frac{1}{T_I s}\right)$. Prosessiin kohdistuu ulkopuolinen häiriö T .

a. Viritä Matlabin *pidtune*-komennon avulla prosessille P-säättäjä. Konstruoi systeemille Simulink-diagrammi ja simuloi, kun

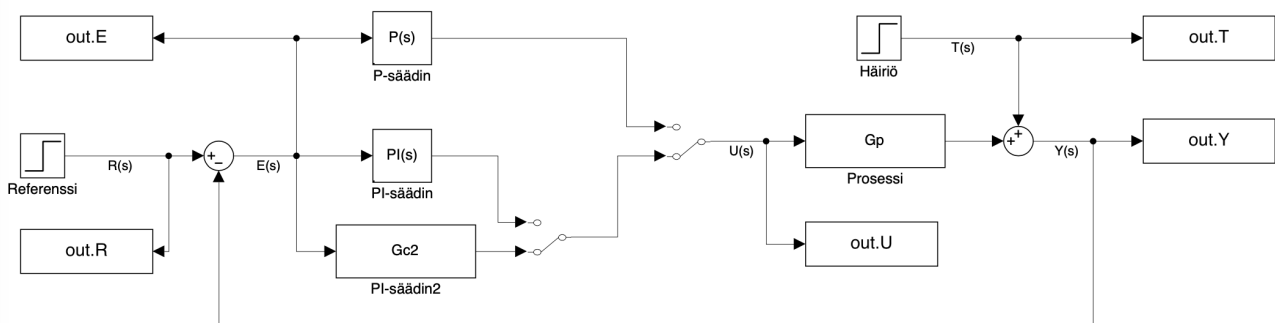
- referenssiin tulee yksikköaskel,
- häiriösuureeseen tulee yksikköaskel.

Esitä kuvat, joissa näkyvät referenssi, häiriösuure, eroisuure, ohjaussuure sekä säädetyn systeemin lähtö.

Käyttämällä Matlabin *pidtune*-komennoa P-säättäjä on viritetty arvoon $K_P \approx 8.492$.

Matlab-koodit löytyvät dokumentin lopusta. Säättöpiirin simulointi on tehty Simulinkissä

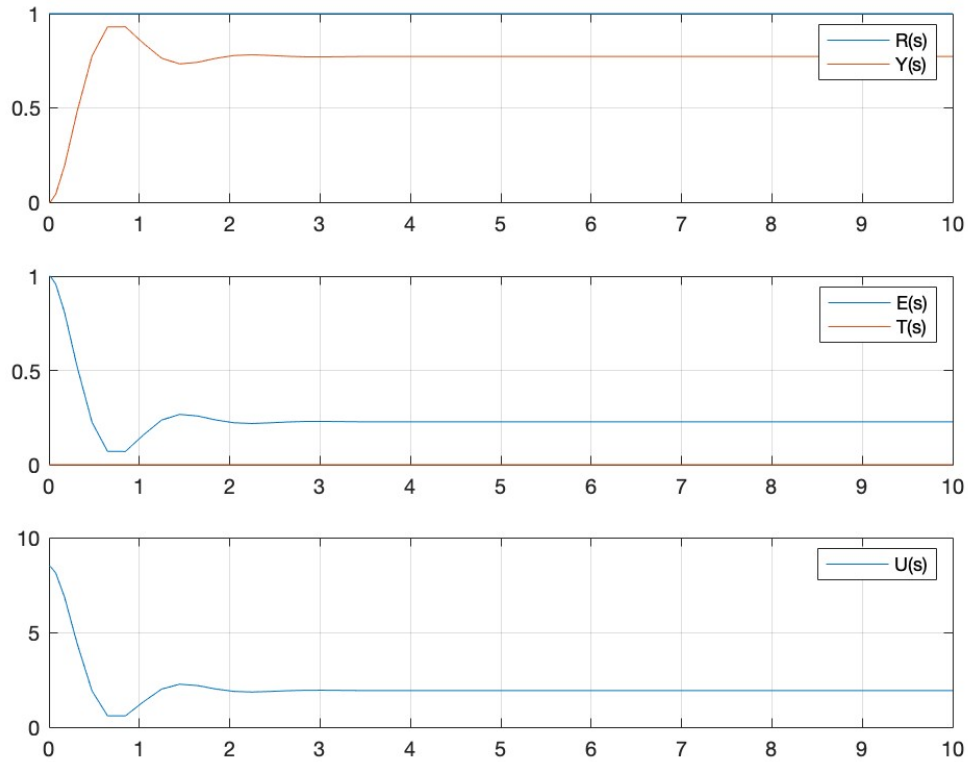
Kuvan 1 mukaisella mallilla. Simulointitulokset P-säädintä käytämällä on esitetty Kuvissa 2 ja 3.



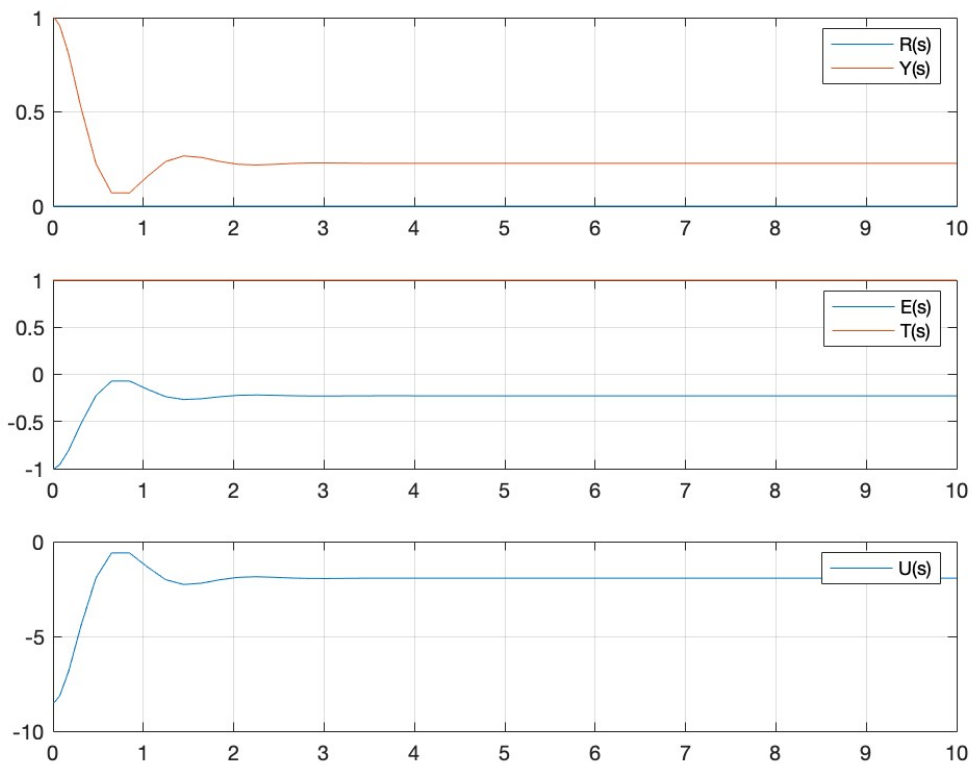
Kuva 1: Simulink-malli. Eri säätimien testaus kytkimiä vaihtamalla.

b. Sama tehtävä kuin a-kohdassa, mutta nyt kokeile PI-säättäjää. Testaa sekä itse siirtofunktiolla toteutamaasi säättäjää että Simulinkin PID-lohkoa. Ovatko tulokset samalla virityksellä identtiset?

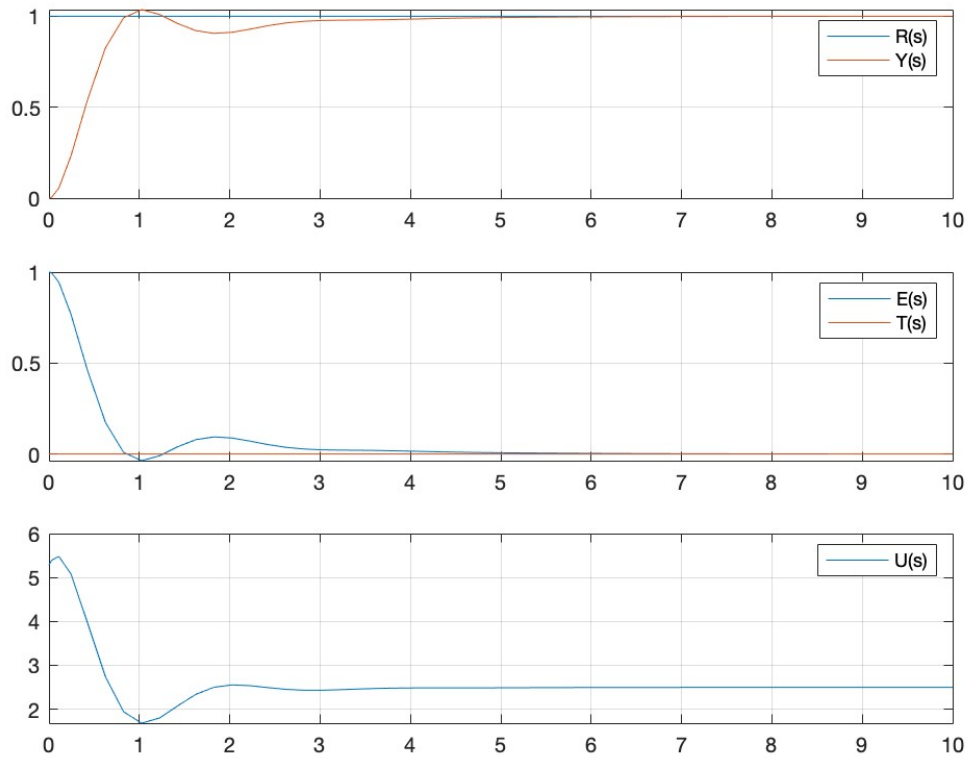
Matlab antaa PI-säättäjän muodossa $K_P + K_I \frac{1}{s}$, jossa *pidtune*-komennolla saatiin $K_P \approx 5.281$ ja $K_I \approx 4.593$. Simuloinnin tulokset on esitetty Kuvissa 4 ja 5. Tulokset ovat samat riippumatta siitä, käytetäänkö Simulinkin PID-lohkoa vai siirtofunktiolla toteutettua säättäjää, joten vain PID-loholla simuloidut kuvat on esitetty.



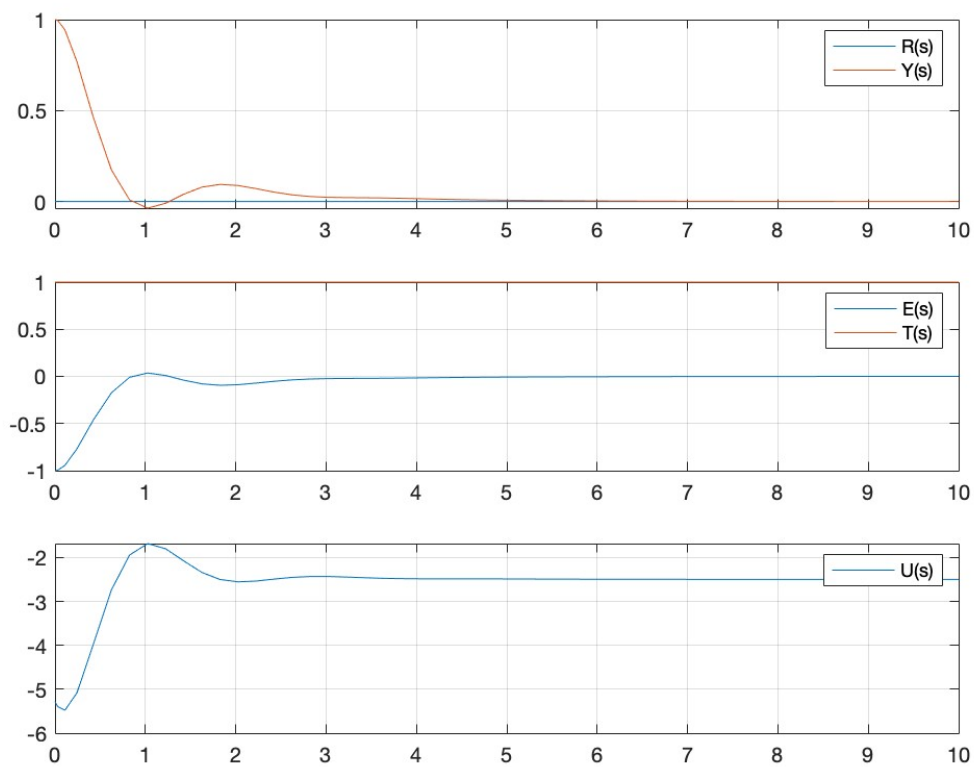
Kuva 2: Systeemin käyttäytyminen, kun prosessiin on kytketty P-säädin, R asettuu arvoon 1 ja $T = 0$.



Kuva 3: Systeemin käyttäytyminen, kun prosessiin on kytketty P-säädin, $R = 0$ ja T asettuu arvoon 1.



Kuva 4: *Systemin käyttäytyminen, kun prosessiin on kytketty PI-säädin, R asettuu arvoon 1 ja $T = 0$.*



Kuva 5: *Systemin käyttäytyminen, kun prosessiin on kytketty PI-säädin, $R = 0$ ja T asettuu arvoon 1.*

- c. Jääkö simulointien perusteella lähtösuureeseen pysyvä poikkeama? Laske (käsini tai Matlabilla) siirtofunktiot referenssistä lähtösuureeseen ja häiriöstä lähtösuureeseen. Verifioi analyttisesti simuloinnin perusteella tekemäsi johtopäätökset.

Kuvien 2–3 ja 4–5 perusteella P-säätimellä lähtöarvoon jää pysyvä poikkeama suhteessa referenssiarvoon, kun taas PI-säätimellä poikkeamaa ei jää. Matlabissa voidaan poimia lähtöarvo simuloinnin lopussa (ks. Matlab-koodi dokumentin lopussa), minkä perusteella P-säätimellä saadut loppuarvot ovat 0.7726 kun referenssi asetuu arvoon $R = 1$ ja 0.2274 kun $R = 0$, eli molemmissa tapauksissa lähtöarvoon jää poikkeama. Kun käytettiin PI-säädintä, simuloinnin loppuarvoiksi saadaan $0.9998 \approx 1$ kun referenssi asetuu arvoon 1 ja $2.3369 \cdot 10^{-4} \approx 0$ kun $R = 0$, eli simulointitarkkuuden rajoissa lähtöarvoon ei jää poikkeamaa.

Tulokset voidaan todeta analyttisesti määrittämällä siirtofunktiot referenssistä lähtösuureeseen ja häiriöstä lähtösuureeseen, ja laskemalla näistä esimerkiksi staattiset vahvistukset. Tarkastellaan ensin siirtofunktiota referenssistä R lähtösuureeseen Y , ja jätetään T huomiotta. Kysessä on takaisinkytkentä, jossa myötähaarassa on sarjaan kytkettynä $G_c(s)$ sekä $G_p(s)$, ja vastahaarassa on vain $Y(s)$, eli vastahaaran siirtofunktio on 1. Näiden peruskytkentöjen lohkokaaviomuunnoksilla saadaan siirtofunktioksi

$$Y(s) = \frac{G_p(s)G_c(s)}{1 + G_p(s)G_c(s)}R(s).$$

Tarkastellaan sitten siirtofunktiota häiriöstä T lähtösuureeseen Y , ja jätetään R huomiotta. Jos aluksi jätetään myös prosessin G_p lähtösuure huomiotta, haettu siirtofunktio olisi yksinkertaisesti $Y(s) = T(s)$. Kun prosessin lähtösuurekin otetaan huomioon, saadaan (positiivinen) takaisinkytkentä, jossa myötähaaran siirtofunktio on äsken päätelty 1 ja vastahaarassa on sarjaan kytkettynä -1 , $G_c(s)$ ja $G_p(s)$, eli vastahaaran siirtofunktio on $-G_p(s)G_c(s)$. Positiivisen takaisinkytkennän siirtofunktioksi saadaan siis

$$Y(s) = \frac{1}{1 + G_p(s)G_c(s)}T(s).$$

Siirtofunktiot voidaan laskea myös Matlabissa (koodi alla), jolloin P-säätäjälle saadaan

$$Y(s) = \frac{16.98}{s^2 + 4s + 21.98}R(s)$$

$$Y(s) = \frac{s^2 + 4s + 5}{s^2 + 4s + 21.98}T(s)$$

ja PI-säätäjälle saadaan

$$Y(s) = \frac{10.56s + 9.185}{s^3 + 4s^2 + 15.56s + 9.185}R(s)$$

$$Y(s) = \frac{s^3 + 4s^2 + 5s}{s^3 + 4s^2 + 15.56s + 9.185}T(s).$$

Näistä voidaan laskea staattiset vahvistukset Matlabissa *dcgain*-komennolla, tai vaihtoehtoisesti käsini laskemalla siirtofunktioiden raja-arvot kun $s \rightarrow 0$, mistä saadaan staattisiksi vahvistuksiksi noin 0.7726, 0.2274, 1.0000 ja 0, eli tässä tapauksessa simuloimalla saadut tulokset vastaavat hyvin analyttisiä tuloksia.

```

%% MATLAB-KOODI
% prosessin siirtofunktio
Gp = tf(2,[1,4,5]);

% a) P-saatimen viritys
P = pidtune(Gp, 'P');
% P-saatimen siirtofunktio:
%  $G_{c1}(s) = K_p$ 
Gc1 = tf(P.Kp);

% b) PI-saatimen viritys
PI = pidtune(Gp, 'PI');
% PI-saatimen siirtofunktio:
%  $G_{c2}(s) = K_p + K_i/s = (K_p*s + K_i)/s$ 
Gc2 = tf([PI.Kp PI.Ki],[1 0]);

% simuloinnit Simulinkissa
out = sim('KT2simulink.slx');
% piirretään tulokset
subplot(3,1,1)
plot(out.tout, out.R.data, 'Displayname','R(s)'); hold on;
plot(out.tout, out.Y.data, 'Displayname','Y(s)'); grid on;
legend
subplot(3,1,2)
plot(out.tout, out.E.data, 'Displayname','E(s)'); hold on;
plot(out.tout, out.T.data, 'Displayname','T(s)'); grid on;
legend
subplot(3,1,3)
plot(out.tout, out.U.data, 'Displayname','U(s)'); grid on;
legend
% loppupoikkeama simulaatiosta
disp(out.Y.data(end))

% c) siirtofuntiot R->Y ja T->Y
RY1 = feedback(series(Gp,Gc1),1);
TY1 = feedback(1, -series(Gp,Gc1),+1);
RY2 = feedback(series(Gp,Gc2),1);
TY2 = feedback(1, -series(Gp,Gc2),+1);
% staattiset vahvistukset
gr1 = dcgain(RY1);
gt1 = dcgain(TY1);
gr2 = dcgain(RY2);
gt2 = dcgain(TY2);
disp([gr1; gt1; gr2; gt2]);

```