LABORATORUL #3
COMPENSAREA PID.

1. Scopul laboratorului.
2. Efectul unor termeni P, I, D asupra unor sisteme în buclă închisă.
3. Utilizarea unei interfețe MATLAB GUI pentru proiectarea PID
4. Procedeul Ziegler-Nichols de proiectare pe baza unor experimente.
5. Conversia în digital, pentru implementarea legii de control pe un microcontroler.
6. Mini-proiect.

1. Scopul laboratorului.

Vom studia efectul diverşilor termeni P, PI, PD, PID asupra sistemelor în buclă închisă. In acest scop vom lua trei exemple de sisteme dinamice asupra cărora vom aplica astfel de legi de control, fără a avea valorile coeficienţilor din legea de reglaj optimizate în prealabil.

Apoi, vom utiliza programul de interfaţă GUI din MATLAB, pentru acordarea regulatoarelor PID considerate ca exemple. Vom utiliza MATLAB şi pentru a imita procesul de acordare experimentală bazat pe metoda Ziegler-Nichols de acordare pe baza sensibilităţii (limita de instabilitate).

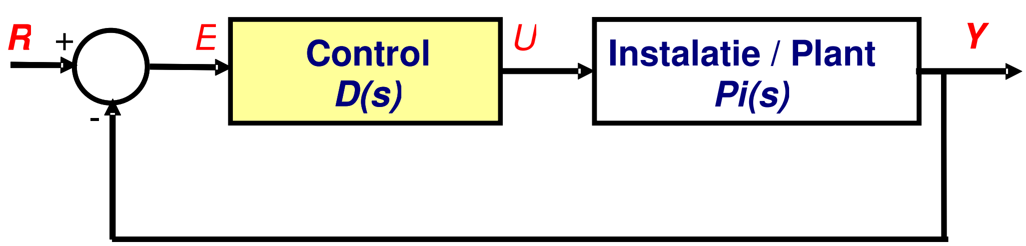
In final, vom utiliza MATLAB pentru obţinerea legii de control digitale ce urmează a fi implementată pe microcontroler. Vom observa ca această conversie nu este unică şi vom încerca sa reţinem efectul fiecărei metode posibile de conversie analog-digital (din model Laplace „***s***" în model cu eşantioane la momente ***t[k]***). Vom vizualiza efectul acestor conversii în SIMULINK, unde vom avea şi o confirmare a programului ce urmează să fie scris într-un microcontroler.

Deși ne vom referi la expresii Laplace pentru legile de control, de fapt proiectarea se face direct în domeniul timp, unde observăm efectul diferitelor valori ale coeficienților proporțional-integrativ-derivativ. Aceasta este deosebit de laboratoarele următoare (metoda locului rădăcinlor, sau metoda proiectării în frecvență), la care considerăm doar reprezentarea în domeniul s.

2. Efectul termenilor P, I, D asupra unor sisteme în buclă închisă.

Vom considera 3 sisteme dinamice, ca exemple pentru câteva clase mai importante de sisteme:

$$P_1(s)=\frac{5}{s+15} \quad P_2(s)=\frac{30}{s^2+2\cdot s+10} \quad P_3(s)=\frac{1}{s^3+2.414\cdot s^2+2.414\cdot s+1}$$



Pentru fiecare astfel de sistem, vom aplica succesiv control cu o lege de compensare P, PI, PD, sau PID. Legea generală de control poate fi scrisă într-o <u>formă standard</u> (clasică):

$$C(s) = K_p + \frac{K_i}{s} + \frac{K_d \cdot s}{T_f \cdot s + 1} = K_p + \frac{K_i}{s} + \frac{K_d \cdot s}{\frac{s}{\beta} + 1}$$

sau într-o <u>formă paralelă</u>:

$$C(s) = K_p \cdot \left(1 + \frac{1}{T_i} \cdot \frac{1}{s} + \frac{T_d \cdot s}{\frac{T_d}{N} \cdot s + 1} \right)$$

Ambele forme sunt echivalente. Exprimarea cu constante de timp (T_I, T_D) are o semnificație istorică și un sens mai practic. T_I poate fi considerată ca o „constantă de timp de resetare” („*reset rate*”), iar T_D ca o „constantă de timp de derivare” („*derivative rate*”). Multi tehnicieni fac setarea coeficienţilor legilor de control din familia PID pe baza acestui concept de câştig şi constantă de timp, care este mai aproape de un sens fizic.

Controlul proporţional este uşor de înţeles împreună cu principiul controlului cu reacţie. Controlul de tip PI are avantajul că elimină eroarea staţionară. Dacă sistemul ar avea eroare staţionară, această eroare ar fi integrată în controlul PI şi ieşirea controlului ar continua sa se schimbe în direcţia scăderii erorii staţionare. In ciuda acestui avantaj, controlul PI încetineşte răspunsul sistemului. Controlul PD are rolul de a raspunde mai rapid la orice schimbare a sistemului. Componenta derivativă acţionează ca un impuls (sau imbold) pentru a accelera operarea sistemului. Din motive practice, nu se poate implementa componenta derivativă direct, şi se consideră o lege P-D echivalentă, de tipul:

$$C(s) = \frac{K_p \cdot \left(\frac{s}{\beta} + 1 \right) + K_d \cdot s}{\left(\frac{s}{\beta} + 1 \right)}$$

unde β este foarte mare (100 în exemplele noastre).

Vom selecta doar un număr limitat de valori pentru fiecare caz, cu scopul evaluării clare a sensului variaţiei fiecărui coeficient. Trebuie să fim capabili să răspundem la întrebări de genul *„Ce se întâmplă dacă creştem kp la un control proporţional pentru un sistem de ordinul întâi?”*

Programul următor are următoarele date de test:

- Control Proporţional
  $Test\_1: K_{p1} = 0.1 \quad K_{i1} = 0 \quad K_{d1} = 0$
  $Test\_2: K_{p2} = 1.0 \quad K_{i2} = 0 \quad K_{d2} = 0$
  $Test\_3: K_{p3} = 10 \quad K_{i3} = 0 \quad K_{d3} = 0$
  $Test\_4: K_{p4} = 100 \quad K_{i3} = 0 \quad K_{d3} = 0$

- Control PI
  $Test\_1: K_{p2} = 1.0 \quad K_{i1} = 0.1 \quad K_{d1} = 0$
  $Test\_2: K_{p2} = 1.0 \quad K_{i2} = 1.0 \quad K_{d2} = 0$
  $Test\_3: K_{p1} = 0.1 \quad K_{i3} = 2.0 \quad K_{d3} = 0$

- Control PD

  $Test\_1: K_{p2} = 1.0 \quad K_{i1} = 0 \quad K_{d1} = 0.1$

  $Test\_2: K_{p2} = 1.0 \quad K_{i2} = 0 \quad K_{d2} = 0.5$

  $Test\_3: K_{p1} = 1.0 \quad K_{i3} = 0 \quad K_{d3} = 2.0$

- Control PID

  $Test\_1 \quad K_p = 3.0 \quad K_i = 1.0 \quad K_d = 2.0$

Acum puteţi lansa programul *L3_Intro.m*.

```
clear;
echo on
%------------------------------------------------------------
% Vom ilustra grafic efectul compensarii de tip P, PI, PD, PID
% Vom considera o structura in bucla inchisa, cu reactie unitara
%                    _________        ___________
%      y            |         |      |           |    error      ref.
%    <------o-----|instal |<-----|   C(s)    |<---------(+)<------
%           |      |________|      |_________|           ^ -
%           |                                            |
%           |____________________________________________|
%
% Compensarea C(s) se proiecteaza astfel incat iesirea sistemului sa aiba
% o valoare apropiata de cea a intrarii.
% Functiile de transfer considerate pentru compensare sunt:
%
%    C1(s) = Kp                              (P)
%    C2(s) = Kp + Ki/s                       (PI)
%    C3(s) = Kp + Kd*s/(s/beta + 1)          (PD)
%    C4(s) = Kp + Ki/s + Kd*s/(s/beta + 1)   (PID)
%
pause   % Apasati orice tasta pentru a continua
%------------------------------------------------------------
%
% Vom exemplifica proiectarea acestor retele de compensare,
% pentru 3 exemple:
%    P1(s) = A/(s+B)
%    P2(s) = C/(S^2 + 2*z*wn*s + wn^2)
%    P3(s) = 1/((s + 1)(s^2 + 1.4142*s + 1))
%
% Proiectarea acestor retele de compensare in sisteme de control, se
% bazeaza pe incercari si depind de simtul sistemelui dezvoltat de inginer
%
pause   % Apasati orice tasta pentru a continua
%------------------------------------------------------------
%
num = [0 5];
den = [1 15];
P1 = tf(num,den)
num = [0 0 90];
den = [1 2 100];
```

```
P2 = tf(num,den)
P3 = (tf(1,[1 1]))*(tf(1,[1 sqrt(2) 1]))
pause   % Apasati orice tasta pentru a continua
%
%----------------------------------------------------------
% Vom atribui diferite valori coeficientilor din legea de compensare
% si vom observa efectul lor in raspunsul la semnal treapta.
%
% Sa retinem parametrii:
% P
Kp1 =  0.1;
Kp2 =  1.0;
Kp3 = 10.0;
Kp4 =100.0;
% I
Ki1 = 0.1;
Ki2 = 1.0;
Ki3 = 2.0;
% D
Kd1 = 0.1;
Kd2 = 0.5;
Kd3 = 2.0;
beta = 100;
pause   % Apasati orice tasta pentru a continua
%----------------------------------------------------------
%
echo off;
subplot(3,1,1);
step(P1);
title('Raspunsul sistemului P1 in bucla deschisa');
subplot(3,1,2);
step(P2);
title('Raspunsul sistemului P2 in bucla deschisa');
subplot(3,1,3);
step(P3);
title('Raspunsul sistemului P3 in bucla deschisa');
echo on;
pause   % Apasati orice tasta pentru a continua
%----------------------------------------------------------
%
% Ne fixam ca obiectiv sa imbunatatim fiecare dintre cele 3 raspunsuri,
% fara a specifica cerinte numerice de performanta.
%
% Ne dorim sa avem raspuns cat mai rapid, cu suprareglarea
% mica, stabilizarea rapida, si fara eroare stationara.
%
pause   % Apasati orice tasta pentru a continua
%----------------------------------------------------------
% CONTROL PROPORTIONAL
% Kp1, Kp2, Kp3 & Kp4.
figure;
echo off;
CL1=feedback(Kp1*P1, 1);
```

```
CL2=feedback(Kp2*P1, 1);
CL3=feedback(Kp3*P1, 1);
CL4=feedback(Kp4*P1, 1);
subplot(3,1,1);
step(CL1); hold on;
step(CL2); hold on;
step(CL3); hold on;
step(CL4); hold on;
legend('kp=0.1','kp=1.0','kp=10', 'kp=100');
title('Raspunsul sistemului P1 in bucla inchisa prin kp');
%
CL1=feedback(Kp1*P2, 1);
CL2=feedback(Kp2*P2, 1);
CL3=feedback(Kp3*P2, 1);
CL4=feedback(Kp4*P2, 1);
subplot(3,1,2);
step(CL1); hold on;
step(CL2); hold on;
step(CL3); hold on;
step(CL4); hold on;
legend('kp=0.1','kp=1.0','kp=10','kp=100' );
title('Raspunsul sistemului P2 in bucla inchisa prin kp');
%
CL1=feedback(Kp1*P3, 1);
CL2=feedback(Kp2*P3, 1);
CL3=feedback(Kp3*P3, 1);
CL4=feedback(Kp4*P3, 1);
subplot(3,1,3);
step(CL1); hold on;
step(CL2); hold on;
step(CL3); hold on;
step(CL4); hold on;
legend('kp=0.1','kp=1.0','kp=10', 'kp=100');
title('Raspunsul sistemului P3 in bucla inchisa prin kp');
echo on;
%
pause   % Apasati orice tasta pentru a continua
%------------------------------------------------------------
% CONTROL PROPORTIONAL-INTEGRAL
% [Kp2 Ki1], [Kp2 Ki2], [Kp1 Ki3].
echo off;
figure;
CC1=tf([Kp2 Ki1],[1 0]);
CC2=tf([Kp2 Ki2],[1 0]);
CC3=tf([Kp1 Ki3],[1 0]);
%
CL1=feedback(CC1*P1, 1);
CL2=feedback(CC2*P1, 1);
CL3=feedback(CC3*P1, 1);
subplot(3,1,1);
step(CL1); hold on;
step(CL2); hold on;
step(CL3); hold on;
```

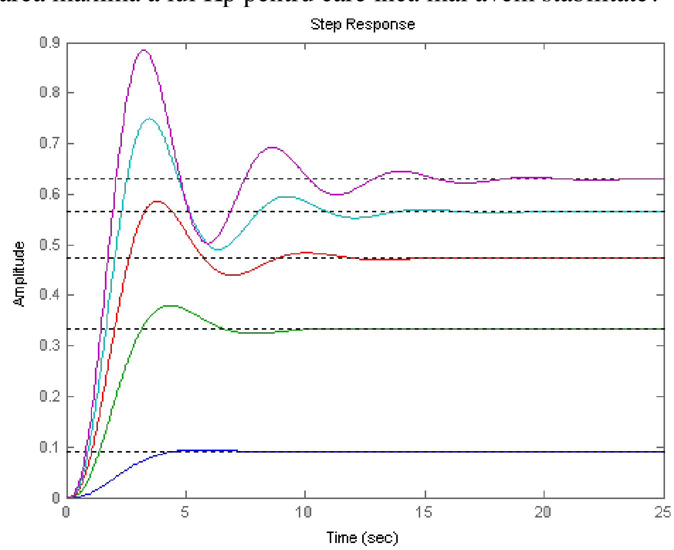
```
axis([0 50 0 2]);
legend('kp=1.0&ki=0.1','kp=1.0&ki=1.0','kp=0.1&ki=2.0');
title('Raspunsul sistemului P1 in bucla inchisa prin PI');
%
CL1=feedback(CC1*P2, 1);
CL2=feedback(CC2*P2, 1);
CL3=feedback(CC3*P2, 1);
subplot(3,1,2);
step(CL1); hold on;
step(CL2); hold on;
step(CL3); hold on;
axis([0 50 0 2]);
legend('kp=1.0&ki=0.1','kp=1.0&ki=1.0','kp=0.1&ki=2.0');
title('Raspunsul sistemului P2 in bucla inchisa prin PI');
%
CL1=feedback(CC1*P3, 1);
CL2=feedback(CC2*P3, 1);
CL3=feedback(CC3*P3, 1);
subplot(3,1,3);
step(CL1); hold on;
step(CL2); hold on;
axis([0 50 0 2]);
legend('kp=1.0&ki=0.1','kp=1.0&ki=1.0');
title('Raspunsul sistemului P3 in bucla inchisa prin PI');
echo on;
%
pause   % Apasati orice tasta pentru a continua
%-------------------------------------------------------------
% CONTROL PROPORTIONAL-DERIVATIV
echo off;
figure;
CC1=Kp2+tf([Kd1,0],[1/beta,1]);
CC2=Kp2+tf([Kd2,0],[1/beta,1]);
CC3=Kp2+tf([Kd3,0],[1/beta,1]);
CC4=5+tf([Kd3,0],[1/beta,1]);
%
CL1=feedback(CC1*P1, 1);
CL2=feedback(CC2*P1, 1);
CL3=feedback(CC3*P1, 1);
CL4=feedback(CC4*P1, 1);
subplot(3,1,1);
step(CL1); hold on;
step(CL2); hold on;
step(CL3); hold on;
step(CL3); hold on;
legend('kp=1.0&kd=0.1','kp=1.0&kd=0.5','kp=1.0&kd=2.0','kp=5.0&kd=2.0');
title('Raspunsul sistemului P1 in bucla inchisa prin PD');
%
CL1=feedback(CC1*P2, 1);
CL2=feedback(CC2*P2, 1);
CL3=feedback(CC3*P2, 1);
CL4=feedback(CC4*P2, 1);
subplot(3,1,2);
```

```
step(CL1); hold on;
step(CL2); hold on;
step(CL3); hold on;
step(CL4); hold on;
legend('kp=1.0&kd=0.1','kp=1.0&kd=0.5','kp=1.0&kd=2.0','kp=5.0&kd=2.0');
title('Raspunsul sistemului P2 in bucla inchisa prin PD');
%
CL1=feedback(CC1*P3, 1);
CL2=feedback(CC2*P3, 1);
CL3=feedback(CC3*P3, 1);
CL3=feedback(CC4*P3, 1);
subplot(3,1,3);
step(CL1); hold on;
step(CL2); hold on;
step(CL3); hold on;
step(CL4); hold on;
legend('kp=1.0&kd=0.1','kp=1.0&kd=0.5','kp=1.0&kd=2.0','kp=5.0&kd=2.0');
title('Raspunsul sistemului P3 in bucla inchisa prin PD');
echo on;
%
pause   % Apasati orice tasta pentru a continua
%-----------------------------------------------------------
% CONTROL PROPORTIONAL-INTEGRATIV-DERIVATIV
echo off;
figure;
Kp=3;
CC1=tf([Kp Ki2],[1 0])+tf([Kd3,0],[1/beta,1]);
%
CL1=feedback(CC1*P1, 1);
subplot(3,1,1);
step(CL1); hold on;
legend('kp=3.0&ki=1.0&kd=2.0');
title('Raspunsul sistemului P1 in bucla inchisa prin PID');
%
CL1=feedback(CC1*P2, 1);
subplot(3,1,2);
step(CL1); hold on;
legend('kp=3.0&ki=1.0&kd=2.0');
title('Raspunsul sistemului P2 in bucla inchisa prin PID');
%
CL1=feedback(CC1*P3, 1);
subplot(3,1,3);
step(CL1); hold on;
legend('kp=3.0&ki=1.0&kd=2.0');
title('Raspunsul sistemului P3 in bucla inchisa prin PID');
%-----------------------------------------------------------
```

Lucru în clasă (rezultatele se includ în raportul de laborator):

Dupa execuţia programului *L3_Intro.m*, răspundeţi la următoarele întrebări.

1. Funcţia P1, control proporţional
   a) Cum se modifică valoarea staţionară după aplicarea unui semnal treaptă, în funcţie de valoarea kp?
   b) Cum putem caracteriza suprareglarea în funcţie de valorile lui kp?
2. Funcţia P2, control proportional
   a) Cum se modifică valoarea staţionară după aplicarea unui semnal treaptă, în funcţie de valoarea kp?
   b) Cum putem caracteriza frecvenţa oscilaţiilor în funcţie de valorile lui kp?
   c) Care este relaţia între suprareglare şi kp?
3. Funcţia P3, control proporţional
   a) Cum puteţi caracteriza răspunsul sistemului în acest caz?
   b) Graficul prezentat în figură este dat pentru 4 valori ale lui Kp. Este posibil ca doar unele să conducă la sistem instabil. Figura următoare s-a obţinut pentru Kp=0.1, 0.5, 0.9, 1.3 şi respectiv 1.7. Această metodă de a utiliza încercări pentru valoarea lui Kp nu pare productivă. Cum altfel aţi putea determina valoarea maximă a lui Kp pentru care încă mai avem stabilitate?



4. Funcţia P1, control proporţional-integral
   a) Cum se modifică valoarea staţionară după aplicarea unui semnal treaptă, în funcţie de valorile kp&ki?
   b) Cum putem caracteriza timpul de creştere şi timpul de răspuns (stabilizare) în comparaţie cu cele două cazuri precedente (buclă deschisă şi control proporţional)?
5. Funcţia P2, control proporţional-integral
   a) Cum se modifică valoarea staţionară după aplicarea unui semnal treaptă, în funcţie de valorile kp&ki?
   b) Cum putem caracteriza timpul de creştere şi timpul de răspuns (stabilizare) în comparaţie cu cele două cazuri precedente (buclă deschisă şi control proporţional)?
   c) Cum se modifică frecvenţa micilor oscilaţii în funcţie de valorile kp&ki? Cine dă această frecvenţă?

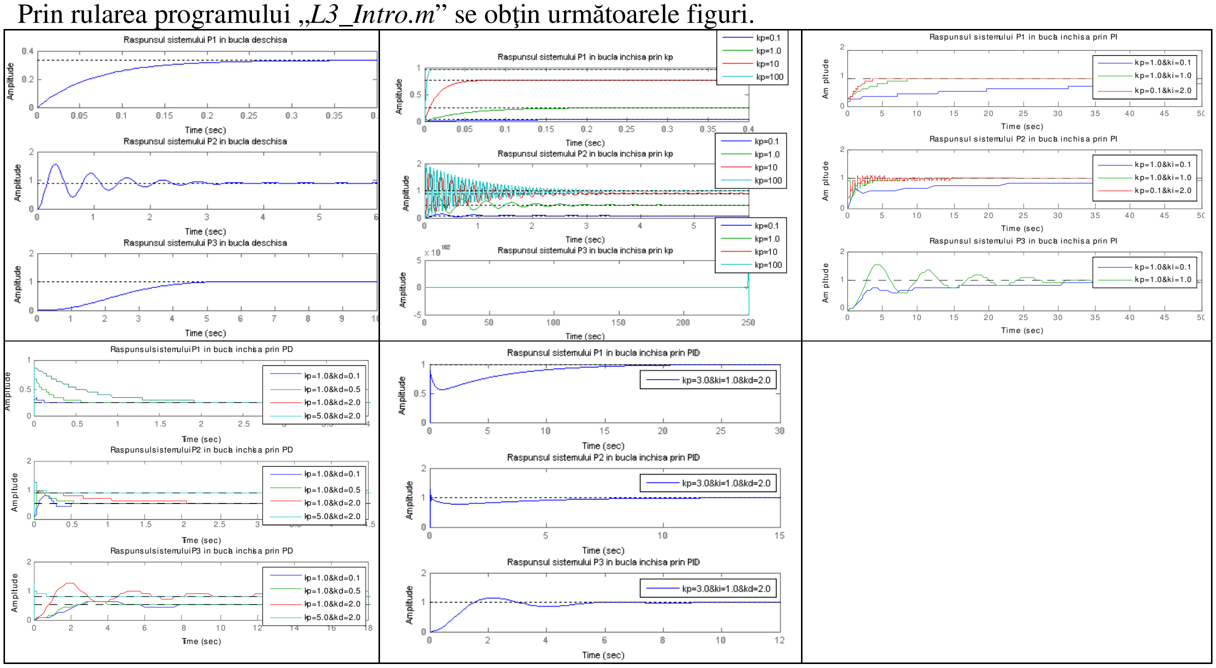
   <u>In concluzie încă mai avem oscilaţii.</u>
6. Funcţia P3, control proporţional-integral

   a) Cum se modifică valoarea staţionară după aplicarea unui semnal treaptă, în funcţie de valorile kp&ki?
   b) Ce credeţi că se va intampla dacă creştem ki în continuare?

   <u>In concluzie încă mai avem posibilitatea de instabilitate.</u>

7. Funcţia P1, control proporţional-derivativ
   a) Cum se modifică valoarea staţionară după aplicarea unui semnal treaptă, în funcţie de valorile kp&kd?
   b) Pentru ce valori ale lui kd, avem un imbold iniţial mai mare (o suprereglare mai mare în răspunsul sistemului la semnal treaptă)?
8. Funcţia P2, control proporţional-derivativ
   a) Cum se modifică valoarea staţionară după aplicarea unui semnal treaptă, în funcţie de valorile kp&kd?
   b) Pentru ce valori ale lui kd, avem un imbold iniţial mai mare (o suprereglare mai mare in răspunsul sistemului la semnal treaptă)?
   c) Ce se întâmplă cu oscilaţiile pe care le-am vazut în cazul sistemului în buclă deschisă şi în cazurile controlului cu lege P sau PI? Pentru ce valori ale lui kd dispar oscilaţiile?
9. Funcţia P3, control proporţional-derivativ
   a) Cum aţi caracteriza efectul componentei derivative în acest caz? Este impulsul iniţial tot aşa de evident ca în cazurile precedente (celelate două funcţii de transfer)?
   b) Pentru ce valori ale lui kd avem răspuns mai rapid?
   c) Observaţi cazul kp=5 care era instabil la controlul simplu proporţional şi devine stabil prin adăugarea componentei kd.
10. Analizaţi toate funcţiile cu control PID, utilizând coeficienţii kp=3, ki=1, kd=2. Bineînţeles că aceeasi lege de control nu poate oferi răspunsuri optimale pentru orice funcţie de transfer în buclă deschisă. Optimizarea controlului pentru fiecare caz în parte este posibilă.
    a) Cum aţi compara aceste răspunsuri la semnal treaptă cu cele ale sistemului în buclă deschisă (sistemul iniţial)?

Prin rularea programului „*L3_Intro.m*” se obţin următoarele figuri.



3. Utilizarea unei interfeţe MATLAB GUI pentru proiectarea PID

De la consola MATLAB introduceţi comanda « *pidtool* », dupa ce în prealabil aţi rulat programul *L3_Intro.m*, pentru a avea funcţiile de transfer P1, P2, şi P3, în mediul de lucru MATLAB. Sau re-scrieţi aceste funcţii în MATLAB.

Încărcaţi pe rând funcţiile P1, P2, şi P3, şi vizualizaţi efectul utilizării diferitelor câştiguri pentru legea de control, cu scopul familiarizării cu acest « *pidtool* ». Încărcarea fiecărei funcţii de transfer se face din colţul stânga-sus, prin functia « *import* ». Selectaţi:

- Design Mode = Basic
- Form = Standard
- Type = P

Selectaţi „*show parameters*” în dreapta ecranului pentru a vedea câştigurile legilor de control PID.

Pentru selecţia graficului urmăriţi tabelul de mai jos. Practic, veţi selecta « *reference tracking* » pentru a vedea răspunsul la semnal treaptă a funcţiei de transfer în buclă închisă. Observaţi că toate aceste rezultate de proiectare sunt ajustate pentru timpul de răspuns ca un criteriu optimal.

Dacă schimbaţi

- Design Mode = Extended

proiectarea se va face prin utilizarea directă a lărgimii de banda (BW) şi a fazei ca cerinţe de proiectare. Vom analiza acest caz în detaliu în săptămâna a 5-a.

Alte semnale ce pot fi analizate sunt:

Response	**Plotted System**
`Reference tracking`	$\frac{C*sys}{1+C*sys}$ (from r to y)
`Controller effort`	$\frac{C}{1+C*sys}$ (from r to u)
`Input disturbance Rejection`	$\frac{sys}{1+C*sys}$ (from d_1 to y)
`Output disturbance Rejection`	$\frac{1}{1+C*sys}$ (from d_2 to y)
`Open-loop`	$C*sys$
`Plant`	sys

Lucru în clasă (rezultatele se includ în raportul de laborator):

11. Încercaţi diferite combinaţii de control pentru cele trei funcţii de transfer. Încercaţi să aveţi în vedere următoarele întrebări:
    a) Este vreun caz în care sistemul bazat pe P1 devine instabil, atunci când este controlat cu control de tip P sau PI? În acest scop încercaţi să scădeţi timpul de creştere. Acest lucru va creşte cerinţele pentru lărgimea de banda şi va determina o creştere a coeficienţilor ***kp*** sau ***ki***. Coeficientul ***ki*** nu este dat direct în legea de control, noi definind termenul integrativ prin ***1/Ti***. Deci ***Ti*** ar putea să scadă pentru creşterea lărgimii de bandă.
    b) În cazul sistemului în buclă deschisă P2, comentaţi asupra stabilităţii sistemului în buclă închisă daca avem control de tip P sau de tip PI.

4. Procedeul Ziegler-Nichols de proiectare pe baza unor experimente.
4.1 Sisteme cu model liniar şi invariabil în timp

Vom exemplifica procedeele de acordare pe bază de experimente Ziegler-Nichols. Aceste procedee se folosesc în cazurile în care avem instalaţia ca hardware, cu modelul nedefinit, sau fără posibilitatea de a face analiza pe calculator. Teoria sistemelor liniare şi invariabile în timp ne sugerează utilizarea acestei metode la sisteme în buclă deschisă ce sunt caracterizate de o funcţie de transfer de ordin mare, atunci când corespondenţa dintre răspunsul în timp şi funcţia de transfer studiată pentru funcţiile de ordinul întâi şi doi nu este direct aplicabilă, iar aceste sisteme de ordin mare pot prezenta instabilitate. Pentru sistemele simple, am putea aplica un semnal treaptă şi identifica funcţia de transfer din forma de undă obţinută în domeniul timp (t_r, M_p, t_s => ζ, ω_n).

In fine, metodele Ziegler-Nichols propun o optimizare capabilă să ofere o reducere a efectului perturbaţiilor, care poate rezulta într-o suprareglare excesivă la urmarirea referinţei de intrare. De obicei, aplicarea directă a metodei conduce la o suprareglare în jurul la 25%. Din acest motiv, în multe cazuri practice, câştigurile derivate prin această metodă sunt micşorate înainte de a fi folosite.

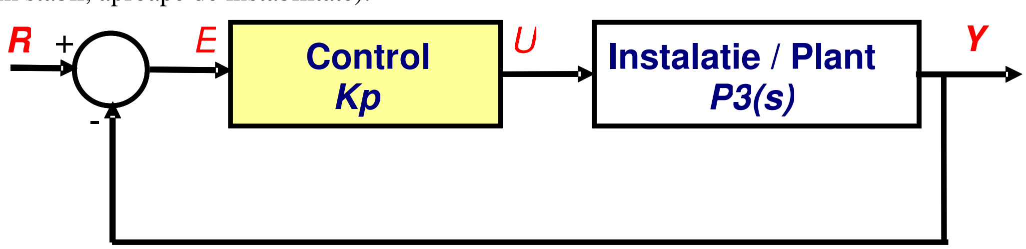
Vom considera în laborator metoda a doua Ziegler-Nichols, denumită ***metoda sensibilitati ultimative***, aplicată funcţiei de transfer P3. Acest procedeu se bazează pe determinarea limitei de stabilitate pentru control proporţional, prin observarea ieşirii sistemului la variaţia parametrilor legii de control.

Vom asuma că, *de fapt*, nu cunoaştem funcţia de transfer, şi vom emula un test experimental.

Scriem secvenţa de program MATLAB:

```
P3 = (tf(1,[1 1]))*(tf(1,[1 sqrt(2) 1]));
for K=1:2:7, CL=feedback(P3,K); stepplot(CL,20); hold on; end
legend('1','3','5','7');
```

Conform metodei Ziegler-Nichols, ar trebui să creştem coeficientul ***kp*** până ce sistemul este la limita între stabilitate şi instabilitate, cu oscilaţii. Oscilaţii în forma de undă a ieşirii apar datorită termenului ***1+kpP3(s)*** de la numitorul funcţiei de transfer în buclă închisă, şi sunt independente de valoarea referinţei constante aplicate la intrare. Va exista o valoare ***kp*** pentru care funcţionarea sistemului va fi la limita de instabilitate, cu o operaţie cu oscilaţii întreţinute în semnalul de ieşire. Chiar dacă procesul de căutare ia mult timp, este de ajuns să găsiţi valorile ***kp*** pentru care oscilaţiile sunt bine conturate (sistem stabil, aproape de instabilitate).



Lucru în clasă (rezultatele se includ în raportul de laborator):

12. Rulaţi programul sugerat şi determinaţi valoarea lui ***kp*** pentru limita aproximativă de instabilitate. Notaţi această valoare cu ***Ku***.
13. Măsuraţi perioada oscilaţiilor din semnalul de ieşire, şi notaţi-o cu ***Pu***.
14. Calculaţi un set de valori de control pe baza tabelului dat (legea extinsă de control).
15. Scrieţi un program MATLAB pentru a vizualiza rezultatele utilizării acestor valori în sistemul în buclă inchisă.

Control	Câstig optimal
P	$K_p = 0.5 \cdot K_u$
PI	$\begin{cases} K_p = 0.45 \cdot K \\ T_I = \dfrac{P_u}{1.2} \end{cases}$
PID	$\begin{cases} K_p = 0.6 \cdot K_u \\ T_I = \dfrac{P_u}{2} \\ T_D = \dfrac{P_u}{8} \end{cases}$

4.2 Observaţii asupra unor sisteme neliniare

Metoda Ziegler-Nichols se poate aplica sistemelor care au o funcţie de transfer ce produce instabilitate pentru anumite valori ale lui ***kp***. In contextul sistemelor liniare şi invariabile în timp, acest lucru este garantat pentru funcţii de transfer cu ***n-m>2*** (unde ***n*** este gradul polinomului de la numitor, iar ***m*** este gradul polinomului de la numărătorul funcţiei de transfer), deci de cel puţin ordinul 3. Alte funcţii de transfer pot de asemeni produce instabilitate.

Majoritatea cazurilor de implementare reală prezintă neliniaritate determinată de limitarea operaţiei elementului de acţionare ce aplică ieşirea legii de control la sistemul controlat sau de operarea instalaţiei la valori mari ale mărimii de acţionare. In astfel de cazuri particulare, există posibilitatea de a observa instabilitatea chiar dacă modelul considerat iniţial era de ordin redus şi stabil în teorie pentru orice ***kp***.

5. Conversia în digital, pentru implementarea legii de control pe un microcontroler

Toate legile de control P, PI, PD, PID considerate anterior, au fost analizate cu modele analogice, prin intermediul funcţiei de transfer Laplace. Ne propunem acum să găsim un echivalent digital pentru fiecare caz şi să parcurgem paşii necesari implementării acestei legi de control pe o structură digitală de tip microcontroler. Vom lucra cu funcţia de transfer P3, cu rezultatele pe care le-am obţinut la primă analiză a funcţiei P3 (kp=2, ki=1, kd=3). Dacă s-ar dori implementarea unui control PI, aceasta s-ar face prin eliminarea termenului D, şi respectarea noilor valori a coeficienţilor corespunzători controlului PI.

Pentru găsirea echivalentului discret al legii de control în format Laplace (***s***), trebuie să selectăm o valoare pentru perioada de eşantionare. O regulă empirică de proiectare spune că perioada semnalului de eşantionare trebuie aleasă astfel încât pe durata timpului de creştere al răspunsului la semnal treaptă (*rising time*), să avem cel puţin 6 eşantioane ale semnalului de intrare (alta sugestie este de a considera 5-10 esantioane). Timpul de creştere se defineşte ca intervalul de timp parcurs între 0.1 şi 0.9 din nivelul final al semnalului de ieşire. In cazuri speciale, prin metode specifice controlului digital, se poate compensa efectul eşantionării, şi permite perioadei de eşantionare să fie doar de 2-3 ori mai mică decât timpul de creştere. Astfel de metode sunt în afara scopului acestui laborator.

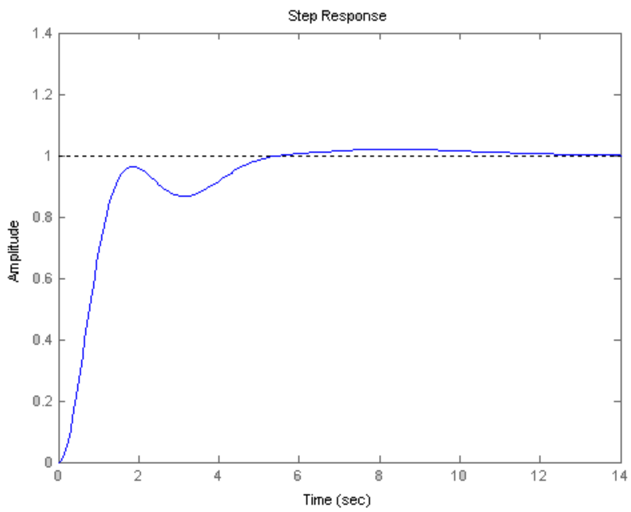
Vom considera aici 6 eşantioane. Din răspunsul la semnal treaptă (obţinut cu programul MATLAB dat mai jos) şi aplicat sistemului în buclă deschisă P3, observăm un timp de creştere (timp de răspuns) de aproximativ 3 secunde. Alegem timpul de eşantionare ts=0.5 sec.

```
clear;
Kp=2 ; Ki=1 ; Kd=3; beta=100;
Ts=0.5;
P3 = (tf(1,[1 1]))*(tf(1,[1 sqrt(2) 1]));
CC=tf([Kp Ki],[1 0])+tf([Kd,0],[1/beta,1]);
CCd=c2d(CC,Ts,'tustin');
CL=feedback(CC*P3, 1);
step(CL);
```

Se foloseşte echivalenţa dată de formula Tustin

$$s \leftrightarrow \frac{2}{T_s} \cdot \frac{z-1}{z+1}$$

Execuţia acestei secvenţe de program MATLAB produce figura următoare:



Funcţia de transfer ***CCd*** a legii de control, exprimată în transformata *z* prin instrucţiunea de conversie c2d, cu metoda Tustin va fi:

```
13.79 z^2 - 22.75 z + 9.923
---------------------------
 z^2 - 0.07692 z - 0.9231
```

iar prin metoda „*matched pole-zeros*"

```
7.153 z^2 - 11.78 z + 5.128
---------------------------
   z^2 - z + 1.929e-022
```

Legea de control va utiliza eşantioanele calculate ale erorii marimii de controlat (e[k-2], e[k-1], e[k]), iar ca ieşire va avea eşantionul următor al mărimii de control (y[k]). Instrucţiunile „***c2d***" din MATLAB oferă o conversie a funcţiei de transfer din Laplace în domeniul *z*. Această conversie trebuie urmată de o scriere a legii de control ca o ecuaţie cu diferenţe finite în eşantioanele ***k***, interpretând *z* ca un operator de întârziere („*shift*"). Legea de control scrisă cu eşantioane la momente ***k*** de timp poate fi scrisă direct în programul microcontrolerului.

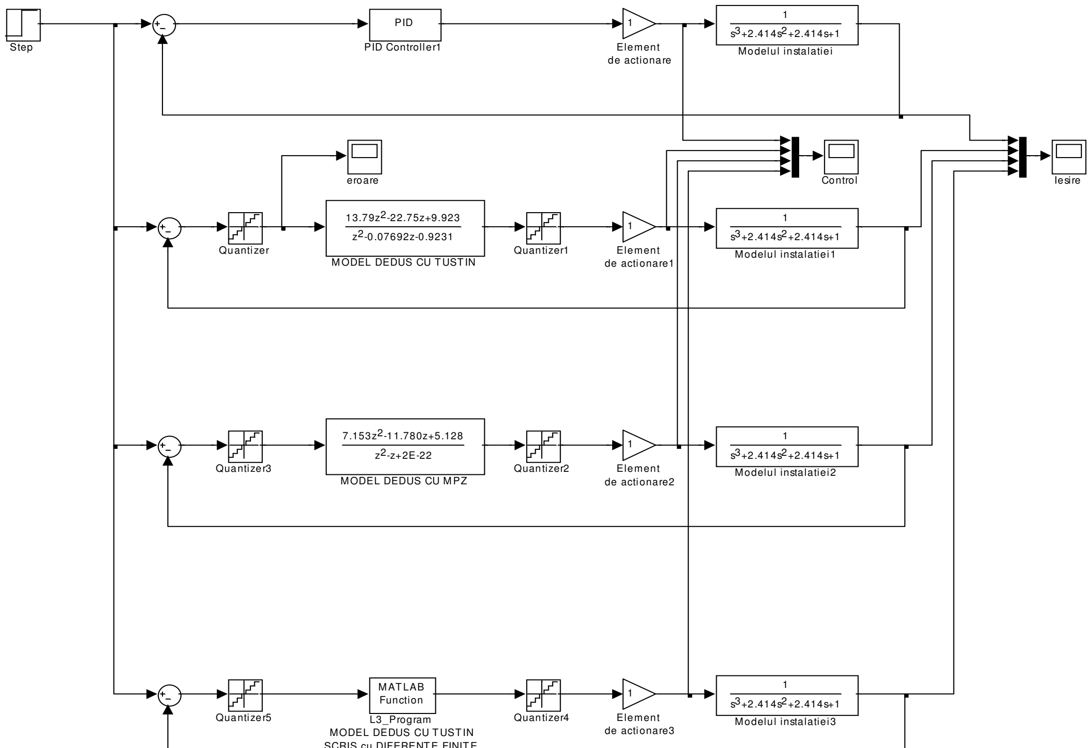
In cazul nostru, obţinem legea de control cu diferenţe finite pentru metoda Tustin:

$$y[k] = 0.07672 \cdot y[k-1] + 0.92310 \cdot y[k-2] + 13.790 \cdot e[k] - 22.750 \cdot e[k-1] + 9.923 \cdot e[k-2]$$

Sau pentru metoda „*matched pole-zeros*”

$$y[k] = y[k-1] + 7.153 \cdot e[k] - 11.780 \cdot e[k-1] + 5.128 \cdot e[k-2]$$

Se observă că cele două legi de control sunt diferite.

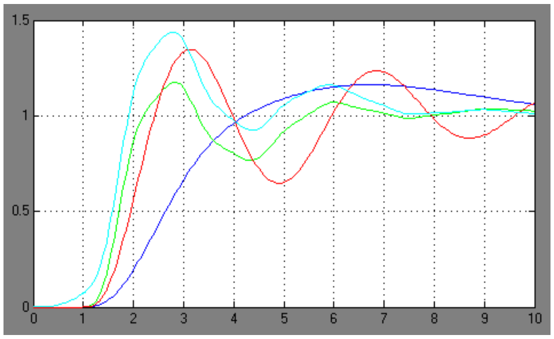


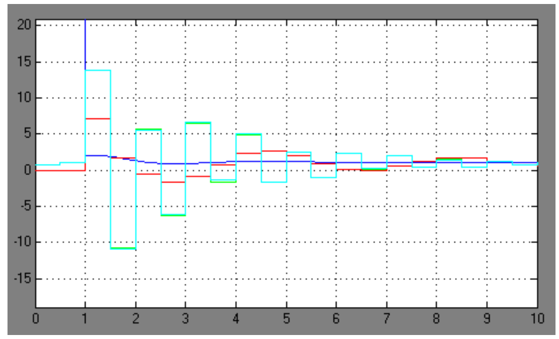
In fine, mediul MATLAB nu oferă direct o posibilitate de simulare a efectului acestei legi de control scrise în eşantioane ***k*** ale mărimilor de intrare-ieşire sau de analiză mixtă în digital şi analog. Din fericire, SIMULINK are capabilitatea de a analiza sisteme digitale (transformata z) şi analogice (transformata Laplace) în aceeaşi simulare. Vom utiliza SIMULINK pentru acest tip de investigaţie.

Vom rula programul SIMULINK, după ce în prealabil am introdus parametrii simulării in mediul MATLAB:

```
L3_Initial;
simulink;
```

Vom încărca modelul „*L3_c2d*” în mediul ***simulink***. Similar, putem rula *L3_c2d* direct la consola MATLAB. Vom obţine rezultatele din figurile următoare:





Pentru a verifica programul scris în limbaj MATLAB sau în limbaj C, ar trebui să folosim un model bazat pe „*s-function*” în MATLAB. În plus, programul implementat pe o platformă de tip microcontroler trebuie să conţină protecţia *anti-wind-up* şi limitarea domeniului de variaţie. Există câteva platforme de control în timp real care se cuplează cu MATLAB-SIMULINK, pentru a rula programul direct pe microcontroler. Oricum, codul verificat cu ecuaţia de control poate fi copiat direct în programul din interiorul microcontrolerului. Acelaşi procedeu este valabil pentru orice lege de compensare vom învăţa în săptămânile viitoare.

Lucru în clasă (rezultatele se includ în raportul de laborator):

16. Identificaţi formele de undă din figurile precedente. Scrieţi în raportul de laborator la ce model din fişierul *simulink (L3_C2D)* corespunde fiecare formă de undă (sau culoare).
17. Repetaţi acest procedeu de scriere a unui program de control digital pentru funcţia P3, considerând un timp de eşantionare de 5 ori mai rapid (Ts=0.10sec). Notaţi în referatul de laborator transformata *z* şi scrieţi legea de control cu diferenţe finite (eşantioanele k).
18. Modificaţi programul MATLAB-SIMULINK pentru noua lege de control şi vizualizaţi funcţionarea prin simulare în MATLAB-SIMULINK.

	Rise Time	Overshoot	Settling Time	Steady State Error
Kp crește	*scade*	*crește*	*crește puțin*	*scade*
Ki crește	*scade puțin*	*crește*	*crește*	*eliminată*
Kd crește	*scade puțin*	*scade*	*scade*	*schimbare*