Narzędzia wspomagające projektowanie

CACSD – Computer-Aided Control System Design

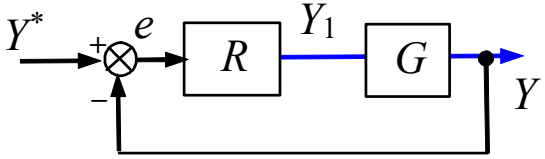
Matlab (Control System Toolbox)

1. Blok PID + PID Tuner

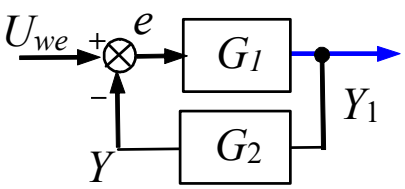
2. Proste funkcje

- feedback
- pidtune
- step, stepinfo

```
Go = R *G
Gz  = feedback(Go,1)
```




```
wyY1 = feedback(G1, G2)
wyY   = wyY1* G2
```



3. SISO Design

Zadania informatyczne

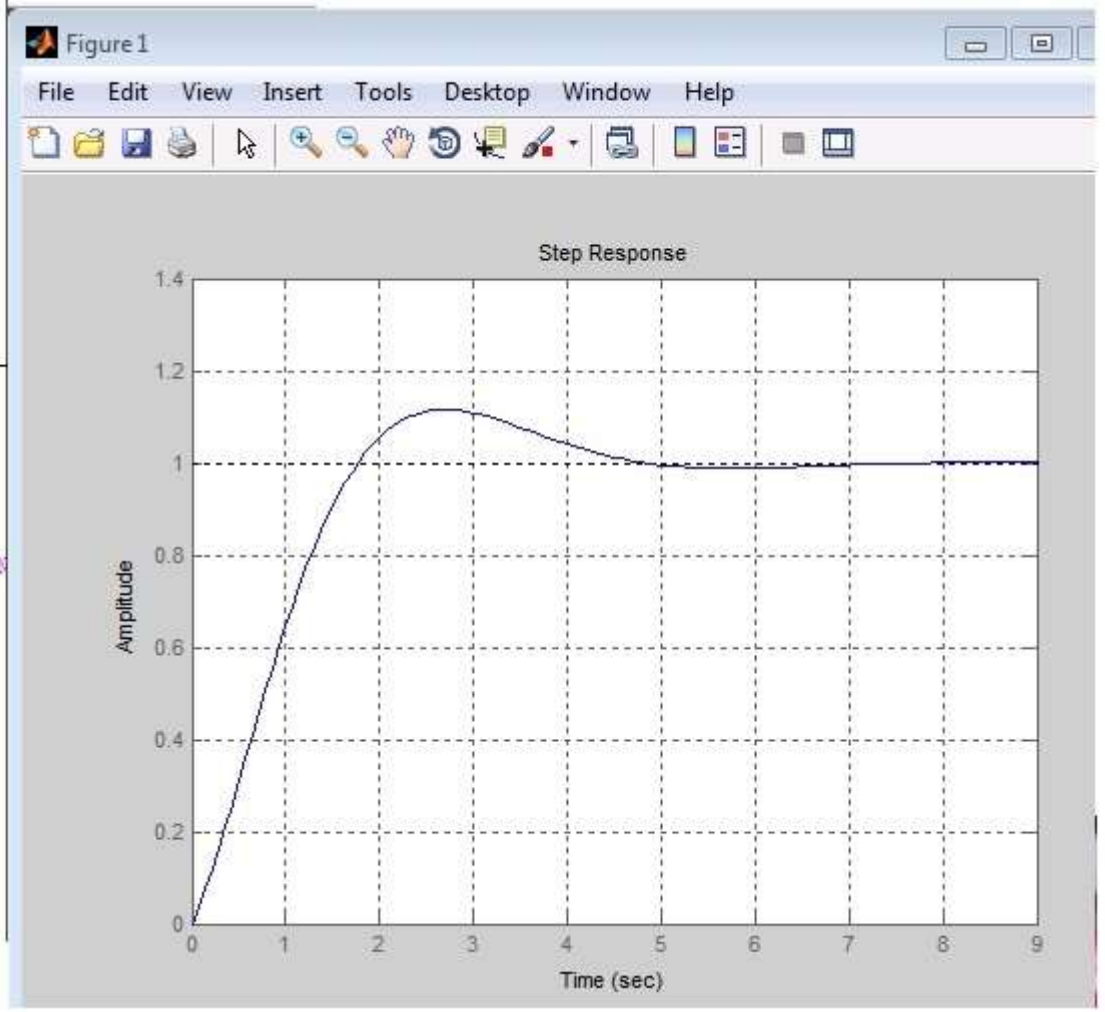
1. Identyfikacja modelu
   na podstawie odpowiedzi skokowej
2. Analiza przebiegu (parametry)
3. Wskaźniki jakości
4. Realizacja algorytmu PID

Przykład 1: obiekt = 1/(s+1), regulator = Kp + Ki/s

funkcja pidtune

```
1 -     close all, clear all
2 -     s=tf('s');
3 -     K1=1; T1=1;
4 -     G1=K1/(T1*s+1);
5
6 -     [C,info]=pidtune(G1,'pi')
7 -     R=C;                                %1.sposob
8       %Kp=C.Kp; Ki=C.Ki; R=Kp+Ki/s;       %2.sposob
9 -     Gz1=feedback(G1*R,1);
10 -    step(Gz1);
11 -    stepinfo(Gz1), %allmargin(G1*R) i info
```

```
Continuous-time PI controller in parallel form:
                1
Kp + Ki * ---                                    i;
                s
with Kp = 0.47319, Ki = 1.6046
info =                                           (ZN
               Stable: 1
    CrossoverFrequency: 1.1237
          PhaseMargin: 60.0000
ans =
        RiseTime: 1.3003
    SettlingTime: 4.3377
     SettlingMin: 0.9053
     SettlingMax: 1.1158
       Overshoot: 11.5817
      Undershoot: 0
            Peak: 1.1158
        PeakTime: 2.7063
```



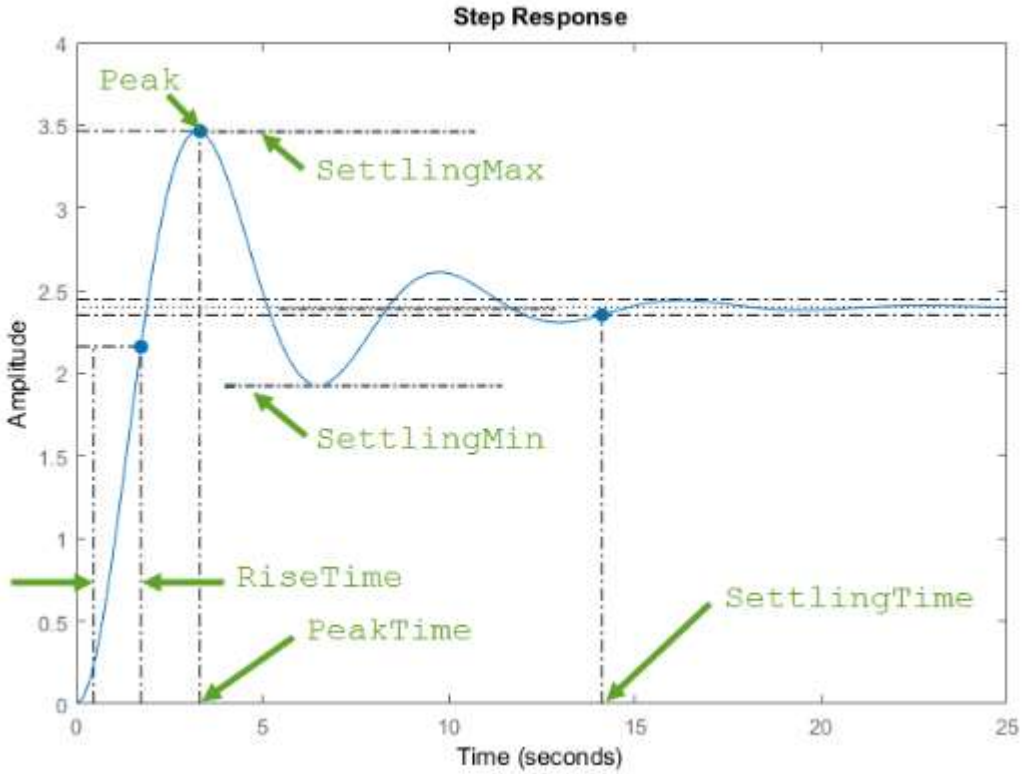
Przykład 1: obiekt = 1/(s+1), regulator = Kp + Ki/s

funkcja pidtune

```
1 -     close all, clear all
2 -     s=tf('s');
3 -     K1=1; T1=1;
4 -     G1=K1/(T1*s+1);
5
6 -     [C,info]=pidtune(G1,'pi')
7 -     R=C;                                %1.sposob
8       %Kp=C.Kp; Ki=C.Ki; R=Kp+Ki/s;       %2.sposob
9 -     Gz1=feedback(G1*R,1);
10 -    step(Gz1);
11 -    stepinfo(Gz1), %allmargin(G1*R) i info
```

```
Continuous-time PI controller in parallel form:
                1
Kp + Ki * ---                                   i;
                s
with Kp = 0.47319, Ki = 1.6046
info =                                          (Z
                    Stable: 1
    CrossoverFrequency: 1.1237
           PhaseMargin: 60.0000
ans =
        RiseTime: 1.3003
    SettlingTime: 4.3377
     SettlingMin: 0.9053
     SettlingMax: 1.1158
       Overshoot: 11.5817
      Undershoot: 0
            Peak: 1.1158
        PeakTime: 2.7063
```

Definicje parametrów



Overshoot = (SettlingMax – xk)/(xk-xp) *100%

Przykład 2: obiekt = 1/(10s+1)*exp(-s), regulator = Kp + Ki/s

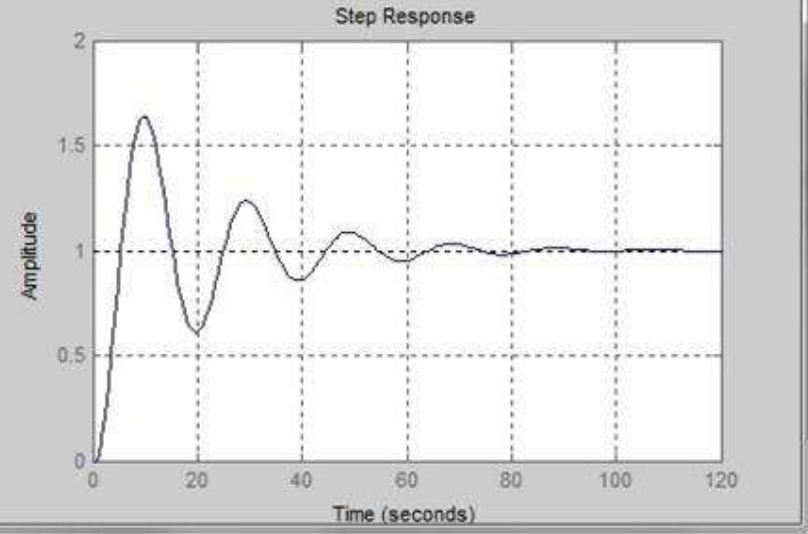
funkcja pidtune

PI_rz10to_txt.m

```
1 -     s=tf('s');
2 -     G=1/(10*s+1)*exp(-s);
3 -     Kp=1; Ki=1;
4 -     R=Kp+Ki/s;
5       [C info]= pidtune(G,'pi')
6 -     R=C;
7 -     Gz=feedback(G*R,1);
8 -     step(Gz)
9 -     stepinfo(Gz)
```

Figure 1

File Edit View Insert Tools Desktop Window Help



Command Window

```
C =

          1
  Kp + Ki * ---
          s

  with Kp = 0.552, Ki = 0.141

Continuous-time PI controller in parallel form.

info =
                Stable: 1
    CrossoverFrequency: 0.1049
           PhaseMargin: 60.0000
ans =
        RiseTime: 3.2878
    SettlingTime: 71.4931
     SettlingMin: 0.6121
     SettlingMax: 1.6382
       Overshoot: 63.8171
      Undershoot: 0
            Peak: 1.6382
        PeakTime: 9.5895

fx >>
```

```
s=tf('s');
G=1/(10*s+1)*exp(-s);
Gp=1/(10*s+1)*pade(exp(-s),1);
Kp=1; Ki=1;
R=Kp+Ki/s;
```

```
%Matlab > R2010b
[C info]= pidtune(G,'pi') R=C;
Gz=feedback(G*R,1);
step(Gz)
stepinfo(Gz)
```

```
%zawsze
[C info]= pidtune(Gp,'pi')  R=C;
Gz=feedback(Gp*R,1);
step(Gz)
stepinfo(Gz)
```

```
          1
Kp + Ki * ---
          s

with Kp = 0.61518, Ki = 0.15101
info =

                Stable: 1
    CrossoverFrequency: 0.1110
           PhaseMargin: 60.0000
ans =

        RiseTime: 3.3021
    SettlingTime: 71.3813
     SettlingMin: 0.6148
     SettlingMax: 1.6368
       Overshoot: 63.6762
      Undershoot: 1.8514
            Peak: 1.6368
        PeakTime: 9.7800
```