

Sprawozdanie – WEAIIB, AiR	
Podstawy Automatyki 2	
Ćwiczenie 9: Dostrajanie regulatora PID	
Czwartek, 14:30	Data wykonania: 25.05.2023
Roman Nowak	Data zaliczenia:
	Ocena:

Celem ćwiczenia jest zapoznanie się z praktycznymi metodami doboru nastaw regulatorów PID. Podczas ćwiczenia wykorzystamy następujące metody doboru nastaw:

- metoda Zieglera – Nicholasa w wersji „klasycznej” i przekątnikowej (metoda Astroma - Hagglunda),

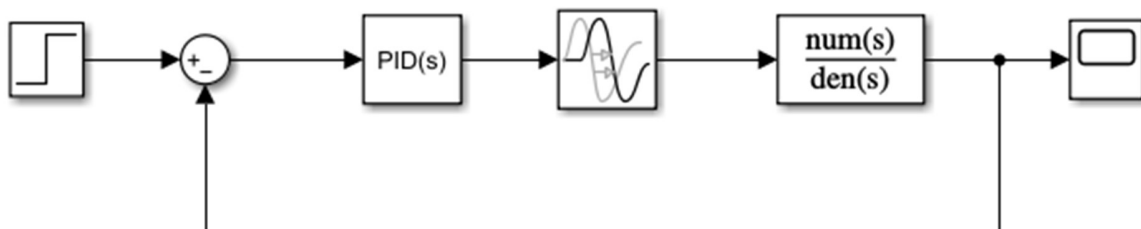
Parametry PID dobieramy według poniższych wzorów

**regulator P:**  $k = 0.5 k_{kr}$ ,

**regulator PI:**  $k = 0.45 k_{kr}$ ,  $T_i = 0.85 T_{osc}$ ,

**regulator PID:**  $k = 0.6 k_{kr}$ ,  $T_i = 0.5 T_{osc}$ ,  $T_d = 0.12 T_{osc}$ .

- metody oparte o parametry odpowiedzi skokowej obiektu,
- autotuning regulatora dostępny w środowisku SIMULINK.



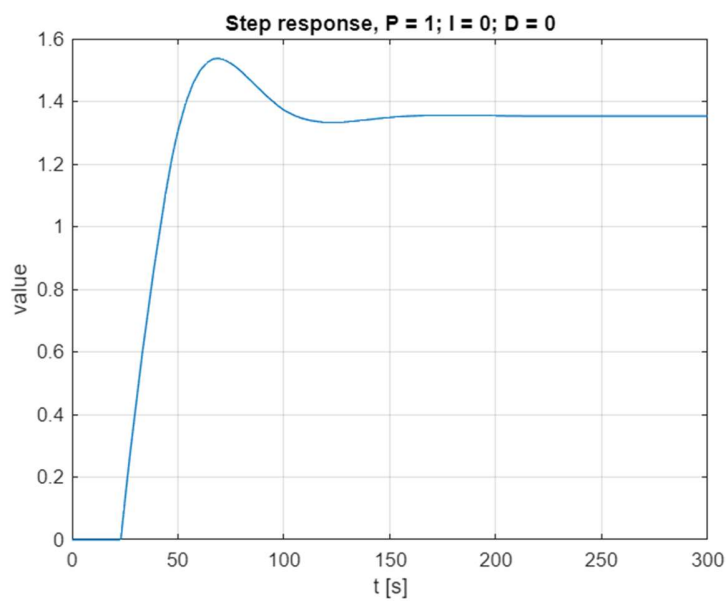
Rys. Schemat układu regulacji w Simulinku.

```

clear;
k = 1.18; tau = 22; T = 45; r = 2.5;

P = 1; I = 0; D = 0;
out = sim("PID.slx");
figure;
plot(out.stepResponse.time, out.stepResponse.signals.values);
title("Step response, P = 1; I = 0; D = 0");
xlabel("t [s]");
ylabel("value");
grid on;

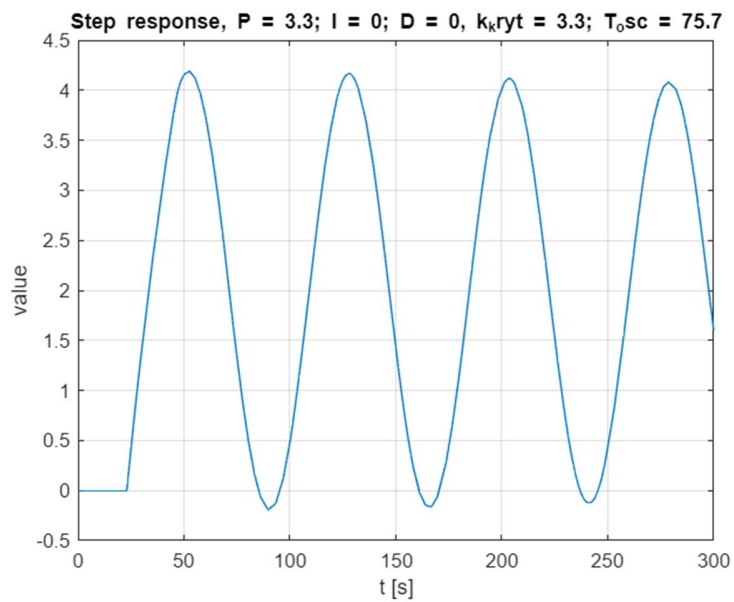
```



```

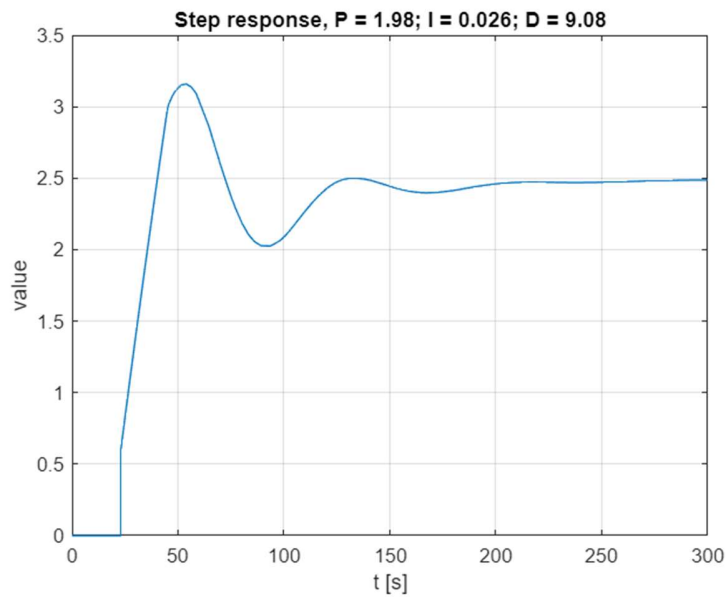
P = 3.3; I = 0; D = 0;
out = sim("PID.slx");
figure;
plot(out.stepResponse.time, out.stepResponse.signals.values);
title("Step response, P = 3.3; I = 0; D = 0, k_kryt = 3.3; T_osc = 75.7");
xlabel("t [s]");
ylabel("value");
grid on;

```



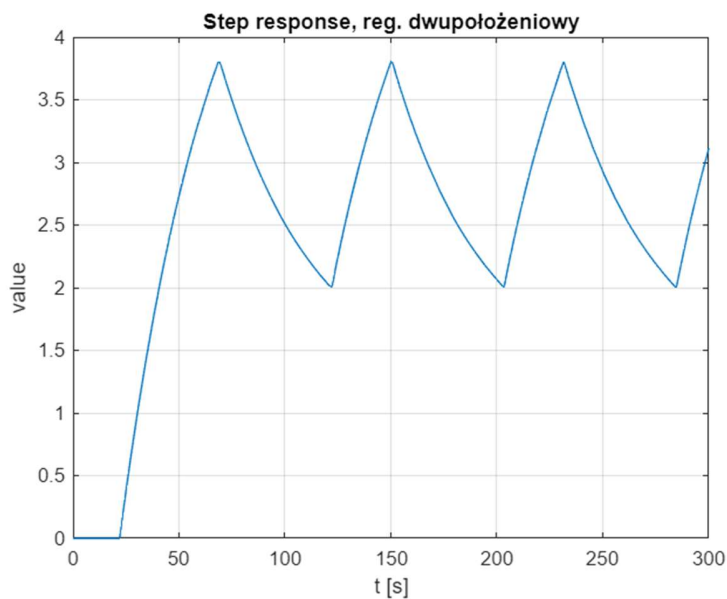
Metoda Zieglera – Nicholasa

```
% [t, y] = ginput(2);
T_osc = 75.7;
P = 0.6 * 3.3; I = 1 / (0.5 * T_osc); D = 0.12 * T_osc;
out = sim("PID.slx");
figure;
plot(out.stepResponse.time, out.stepResponse.signals.values);
title("Step response, P = 1.98; I = 0.026; D = 9.08");
xlabel("t [s]");
ylabel("value");
grid on;
```



Metoda Astroma – Hagglunda – na podstawie odpowiedzi skokowej układu regulacji z regulatorem dwupołożeniowym

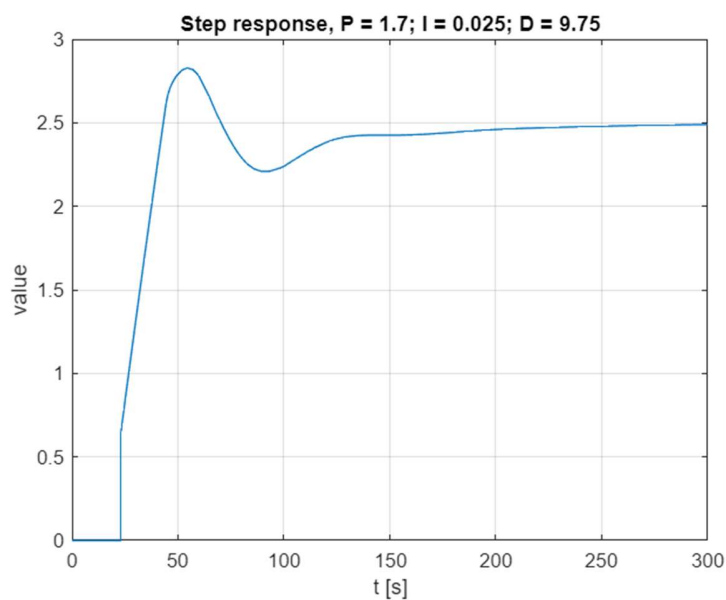
```
out = sim("PID.slx");
figure;
plot(out.twoPolStepResponse.time, out.twoPolStepResponse.signals.values);
title("Step response, reg. dwupołożeniowy");
xlabel("t [s]");
ylabel("value");
grid on;
```



```

u = 2;
%[t, y] = ginput(2);
A = 1.7944 / 2;
T_osc = 81.2155;
k_kr = 4 * u / (pi * A);
P = 0.6 * k_kr; I = 1 / (0.5 * T_osc); D = 0.12 * T_osc;
out = sim("PID.slx");
figure;
plot(out.stepResponse.time, out.stepResponse.signals.values);
title("Step response, P = 1.7; I = 0.025; D = 9.75");
xlabel("t [s]");
ylabel("value");
grid on;

```

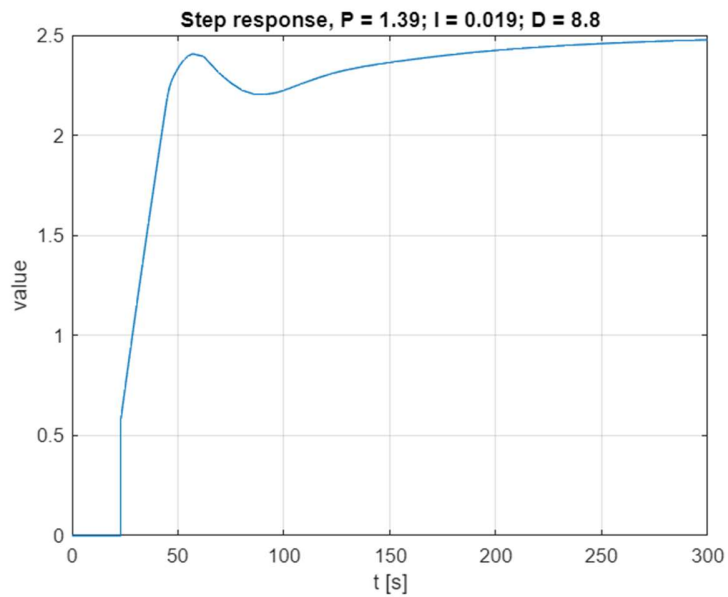


Założenie przeregulowania 20%

```

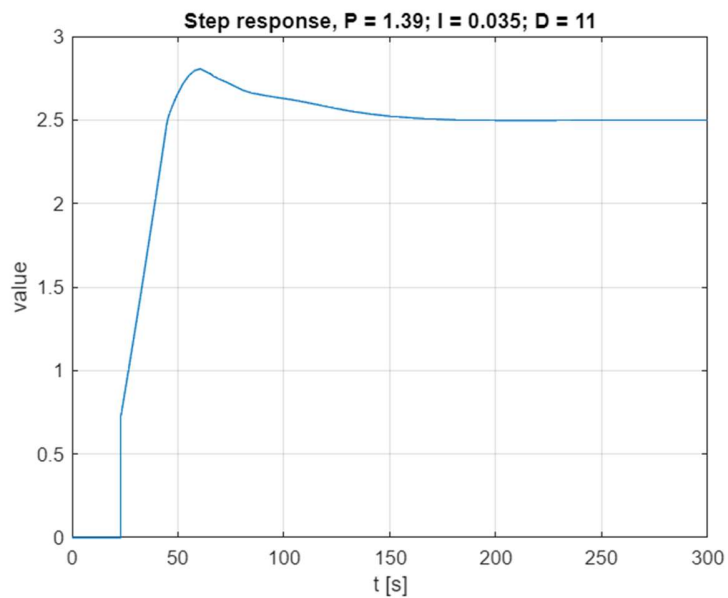
P = k_kr * (tau/T); I = 1 / (2.4 * tau); D = 0.4 * tau;
out = sim("PID.slx");
figure;
plot(out.stepResponse.time, out.stepResponse.signals.values);
title("Step response, P = 1.39; I = 0.019; D = 8.8");
xlabel("t [s]");
ylabel("value");
grid on;

```



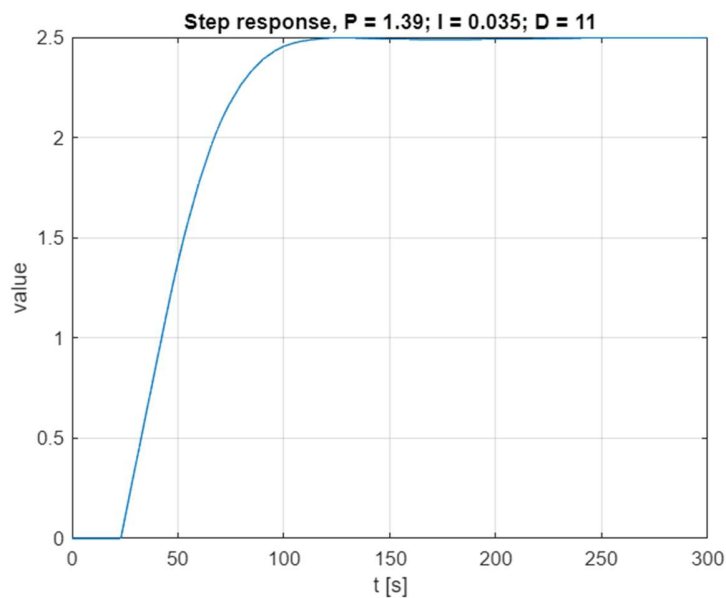
Założenie minimum z całki kwadratu uchybu

```
P = k_kr * (tau/T); I = 1 / (1.3 * tau); D = 0.5 * tau;
out = sim("PID.slx");
figure;
plot(out.stepResponse.time, out.stepResponse.signals.values);
title("Step response, P = 1.39; I = 0.035; D = 11");
xlabel("t [s]");
ylabel("value");
grid on;
```



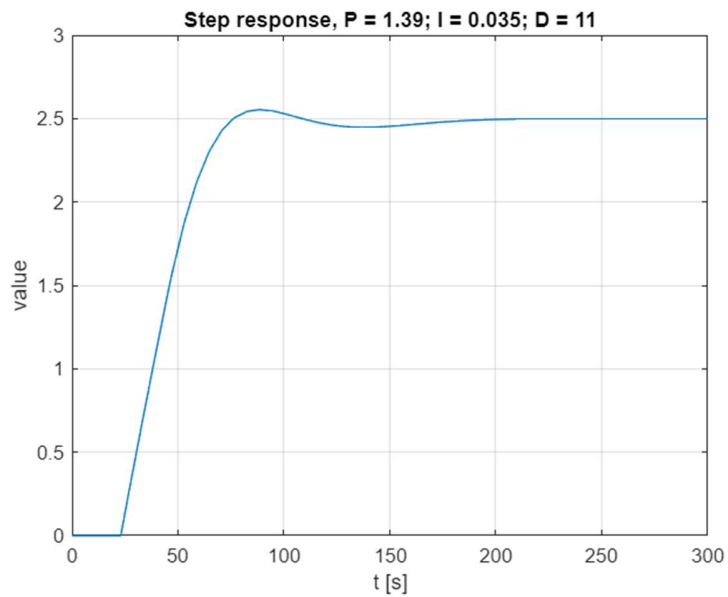
## Strojenie automatyczne - Eksperyment 1 - brak przeregulowania

```
out = sim("PID.slx");  
figure;  
plot(out.stepResponse.time, out.stepResponse.signals.values);  
title("Step response, P = 1.39; I = 0.035; D = 11");  
xlabel("t [s]");  
ylabel("value");  
grid on;
```



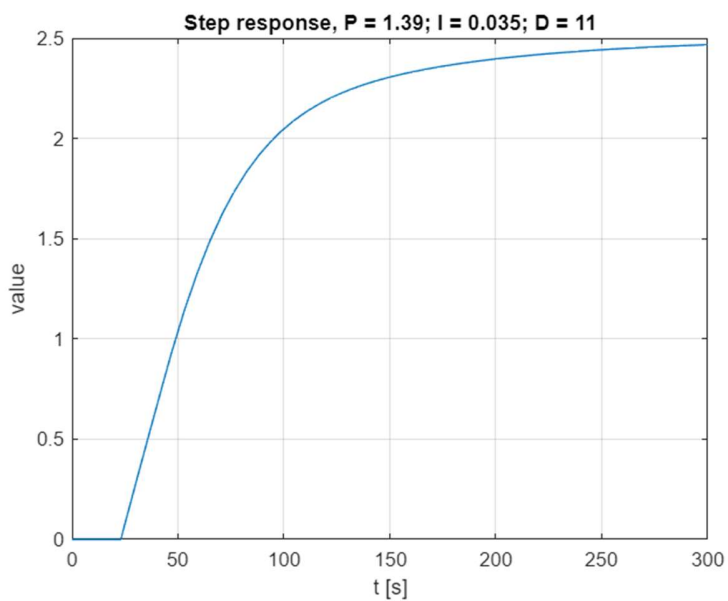
## Strojenie automatyczne - Eksperyment 2 - niewielkie przeregulowanie

```
out = sim("PID.slx");  
figure;  
plot(out.stepResponse.time, out.stepResponse.signals.values);  
title("Step response, P = 1.39; I = 0.035; D = 11");  
xlabel("t [s]");  
ylabel("value");  
grid on;
```



Strojenie automatyczne - Eksperyment 3 - wolne wystierowanie, brak przeregulowania

```
out = sim("PID.slx");
figure;
plot(out.stepResponse.time, out.stepResponse.signals.values);
title("Step response, P = 1.39; I = 0.035; D = 11");
xlabel("t [s]");
ylabel("value");
grid on;
```



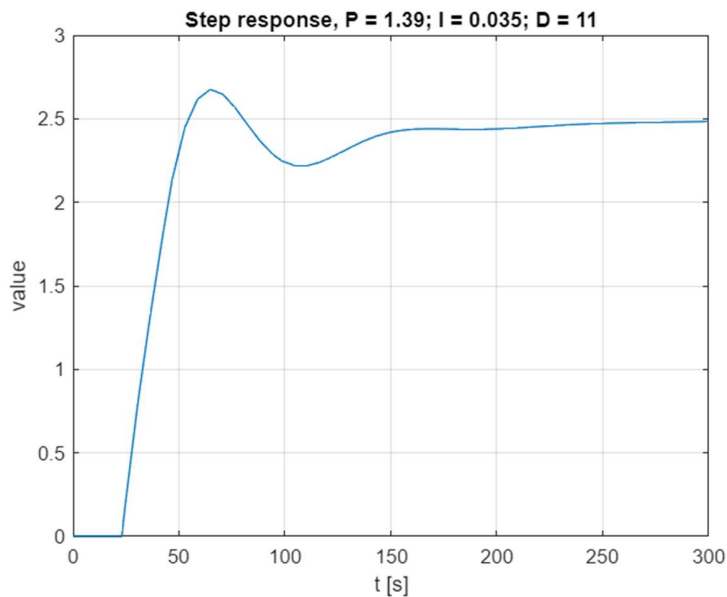
Strojenie automatyczne - Eksperyment 4 - bardzo szybkie wystierowanie



```

out = sim("PID.slx");
figure;
plot(out.stepResponse.time, out.stepResponse.signals.values);
title("Step response, P = 1.39; I = 0.035; D = 11");
xlabel("t [s]");
ylabel("value");
grid on;

```

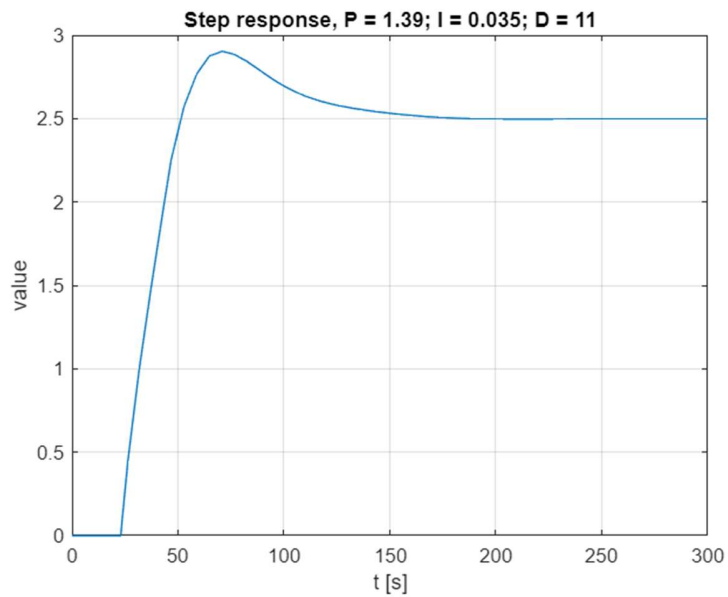


Strojenie automatyczne - Eksperyment 5 - bardzo szybkie wysterowanie, brak oscylacji

```

out = sim("PID.slx");
figure;
plot(out.stepResponse.time, out.stepResponse.signals.values);
title("Step response, P = 1.39; I = 0.035; D = 11");
xlabel("t [s]");
ylabel("value");
grid on;

```



## Wnioski

- Metody, w których parametry regulatora PID dobieramy według wzorów, są proste w użyciu i pozwalają na dobranie parametrów, zapewniających zadowalającą pracę układu.
- Dobieranie parametrów regulatora za pomocą autotuningu, dostępnego w Simulinku jest wygodne i pozwala na dokładne dopasowanie pracy układu do wymagań.