Laboratorium Podstaw Automatyki **Ćwiczenie 9 – Dostrajanie regulatorów PID**		
Nazwisko Imię	**Grupa**	**Data i godzina zajęć**
Szczypek Jakub	Grupa 5a	23.05.2022r. godz.17.00 Poniedziałek

1. Cel ćwiczenia

Celem ćwiczenia było zapoznanie się z praktycznymi, przemysłowymi metodami doboru nastaw regulatorów PID. Podczas ćwiczenia wykorzystane zostały następujące metody doboru nastaw:

- metoda Zieglera – Nicholsa w wersji „klasycznej" i przekaźnikowej (metoda Astroma - Hagglunda),
- metody oparte o parametry odpowiedzi skokowej obiektu,
- autotuning regulatora dostępny w środowisku SIMULINK.

2. Wstęp teoretyczny

Metoda Zieglera-Nicholsa

Pozwala ona na dobór nastaw regulatora bez wcześniejszej znajomości modelu obiektu. Niezbędne jednak jest przeprowadzenie eksperymentu na rzeczywistym, zamkniętym układzie regulacji. Do wyznaczenia wartości wykorzystuje się następujące wzory:

regulator P: $k = 0.5\, k_{kr}$,
regulator PI: $k = 0.45\, k_{kr}$, $T_i = 0.85 T_{osc}$,
regulator PID: $k = 0.6\, k_{kr}$, $T_i = 0.5 T_{osc}$, $T_d = 0.12 T_{osc}$.

Metoda Astroma-Haggunda

Pozwala na dobór nastaw regulatora zgodnie ze wzorami wykorzystywanymi w metodzie Zieglera–Nicholsa wzmocnienie krytyczne obliczane jest zgodnie z następującym wzorem

$$k_{kr} = \frac{4u}{\pi A}$$

gdzie:

u–$amplituda\ sterowania\ przekaźnika\ II\ położeniowego$

A–$amplituda\ oscylacji\ wielkości\ regulowanej$

Oparte o parametry odpowiedzi skokowej obiektu

Zakłada się, że obiekt regulacji jest opisany transmitancją zastępczą z opóźnieniem, przy czym parametry tej transmitancji są identyfikowane na podstawie znajomości odpowiedzi skokowej obiektu. Mając zadaną transmitancję obiektu wyznacza się nastawy regulatora w oparciu o gotowe wzory:

- przy założeniu przeregulowania 20% oraz minimalnego czasu regulacji:

$$k\,k_r\,(\tau/T) = 0.95$$
$$T_i = 2.4\,\tau$$
$$T_d = 0.4\,\tau$$

- przy założeniu minimum z całki kwadratu uchybu:

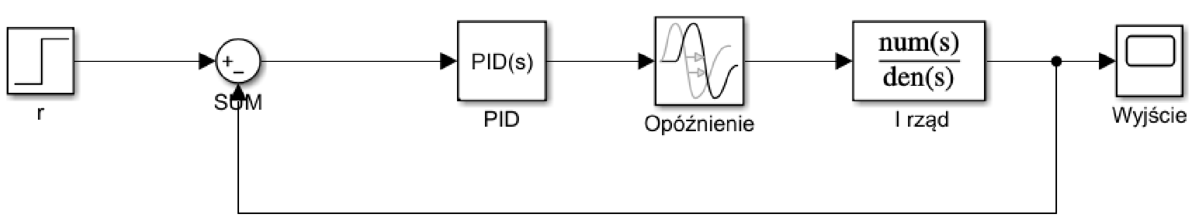
$$k\,k_r\,(\tau/T) = 1.4$$
$$T_i = 1.3\,\tau$$
$$T_d = 0.5\,\tau$$

Metoda wbudowana ,,Autotune”

Metoda dostępna w środowisku Simulink, automatycznie dobiera nastawy regulatora w optymalny sposób

3. Przebieg ćwiczenia

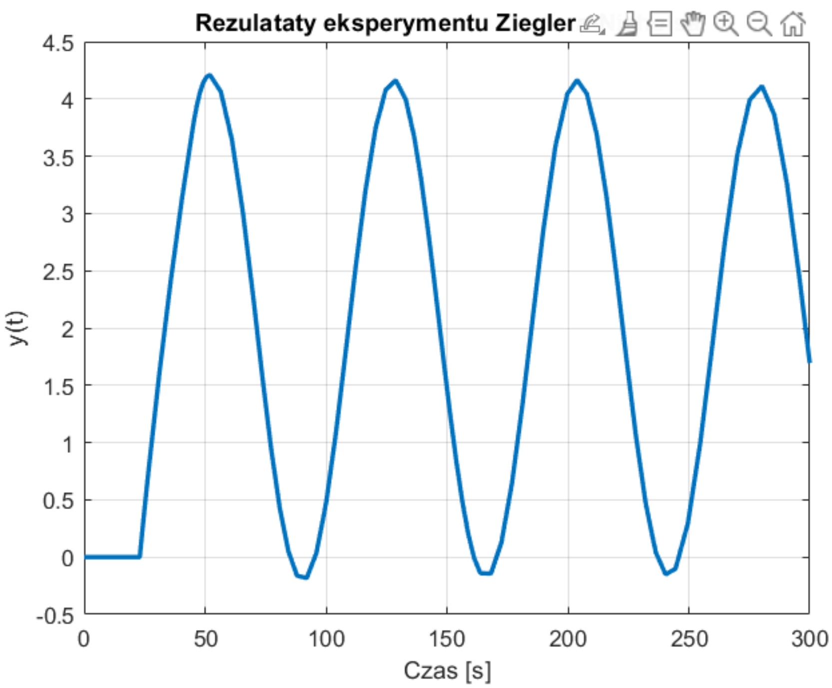
Zbudowano model obiektu inercyjnego I rzędu z opóźnieniem w Simulinku:



Następnie napisano poniższy skrypt do rozwiązania zadania metodą Zieglera-Nicholsa

```
k = 1.18;
tau = 22;
T = 45;
r = 2.5;
kr = 3.3;
ki = 0;
```

```
kd = 0;

out = sim("model.slx");
plot(out.y.time, out.y.signals.values, "LineWidth", 2)
title("Rezulataty eksperymentu Zieglera-Nicholsa")
ylabel("y(t)")
xlabel("Czas [s]")
grid on

[time, y] = ginput(2)
```




```
time = 2×1
   52.2099
  127.9006
y = 2×1
    4.1787
    4.1437
```

```
Tosc = time(2) - time(1)
```

```
Tosc = 75.6906
```

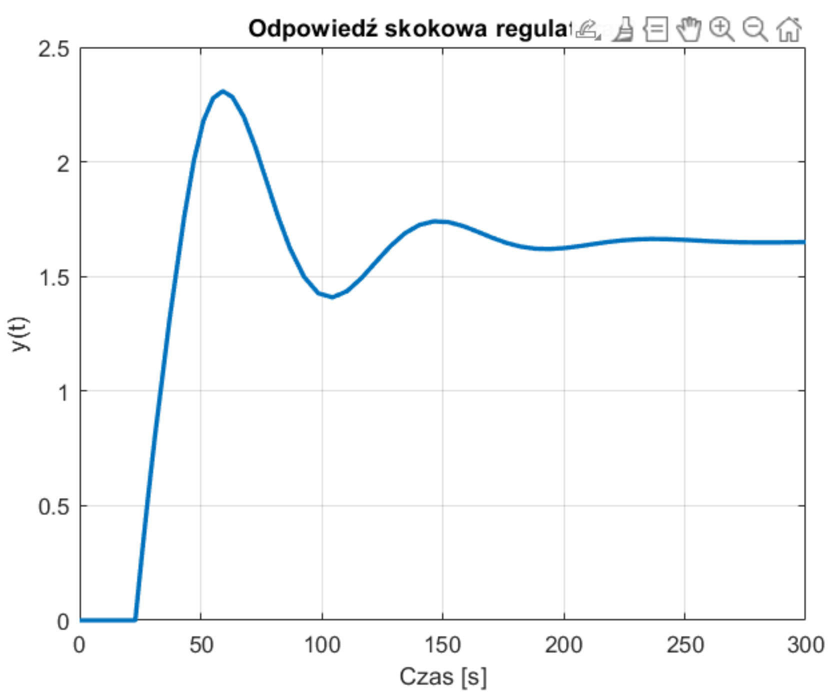
Regulator P

```
k = 1.18;
```

```
tau = 22;
T = 45;
r = 2.5;
kr = 3.3;
kp = 0.5 * kr
```

```
kp = 1.6500
```

```
Ti = 0;
Td = 0;
ki = 0;
kd = 0;

out = sim("model1.slx");
plot(out.y.time, out.y.signals.values, "LineWidth", 2)
title("Odpowiedź skokowa regulatora P")
ylabel("y(t)")
xlabel("Czas [s]")
grid on
```



Regulator PI

```
k = 1.18;
tau = 22;
T = 45;
```

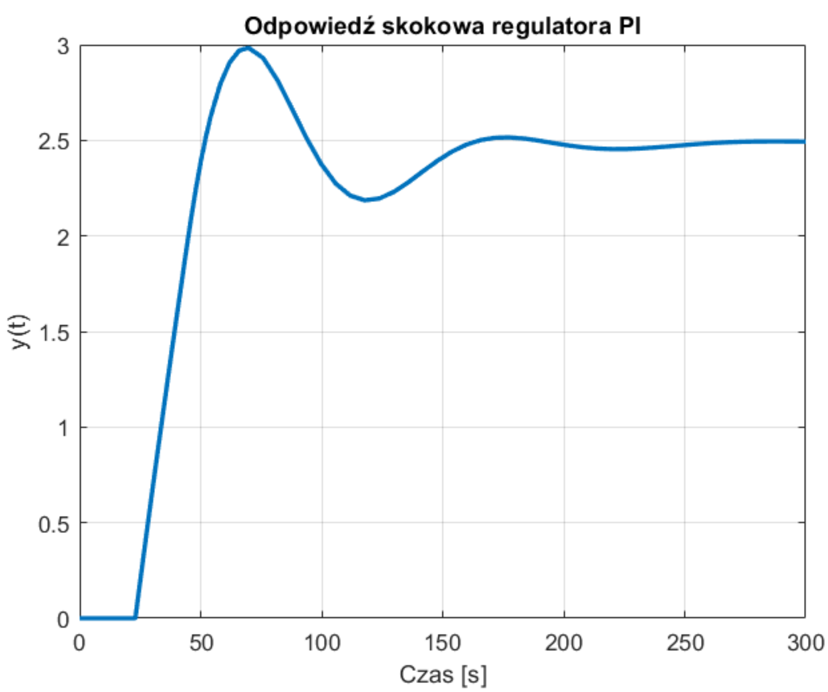
```
r = 2.5;
kr = 3.3;
kp = 0.45 * kr
```

kp = 1.4850

```
Ti = 0.85 * Tosc
```

Ti = 64.3370

```
Td = 0;
ki = kp/Ti;
kd = 0;

out = sim("model1.slx");
plot(out.y.time, out.y.signals.values, "LineWidth", 2)
title("Odpowiedź skokowa regulatora PI")
ylabel("y(t)")
xlabel("Czas [s]")
grid on
```



Regulator PID

```
k = 1.18;
tau = 22;
```

```
T = 45;
r = 2.5;
kr = 3.3;
kp = 0.6 * kr
```

kp = 1.9800

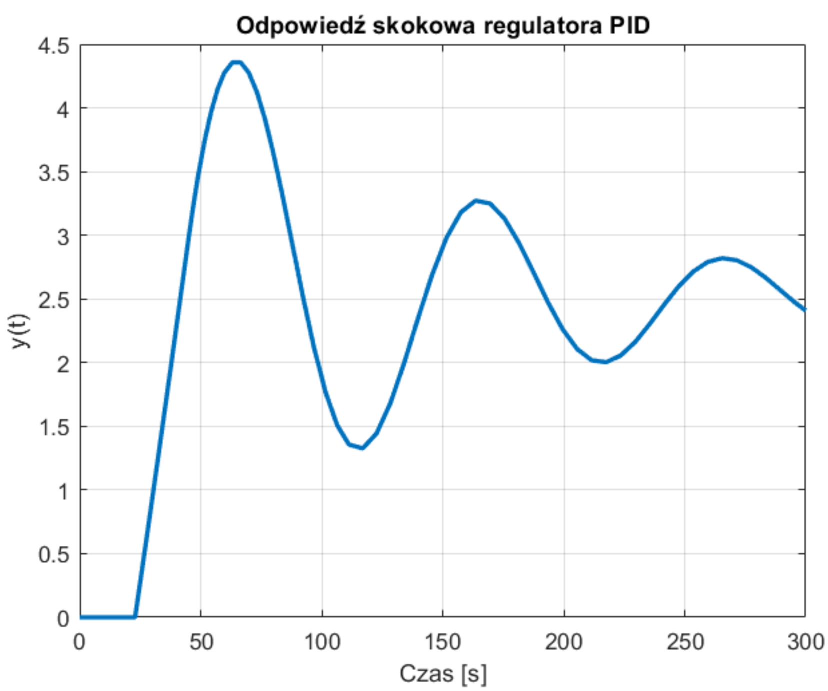
```
Ti = 0.5 * Tosc
```

Ti = 37.8453

```
Td = 0.12 * Tosc
```

Td = 9.0829

```
ki = kp/Ti;
kd = kp/Td;

out = sim("model1.slx");
plot(out.y.time, out.y.signals.values, "LineWidth", 2)
title("Odpowiedź skokowa regulatora PID")
ylabel("y(t)")
xlabel("Czas [s]")
grid on
```



Metoda Astroma-Hagglunda

```
k = 1.18;
tau = 22;
T = 45;
r = 2.5;
kr = 3.359;
kp = 0.6 * kr
```

```
kp = 2.0154
```

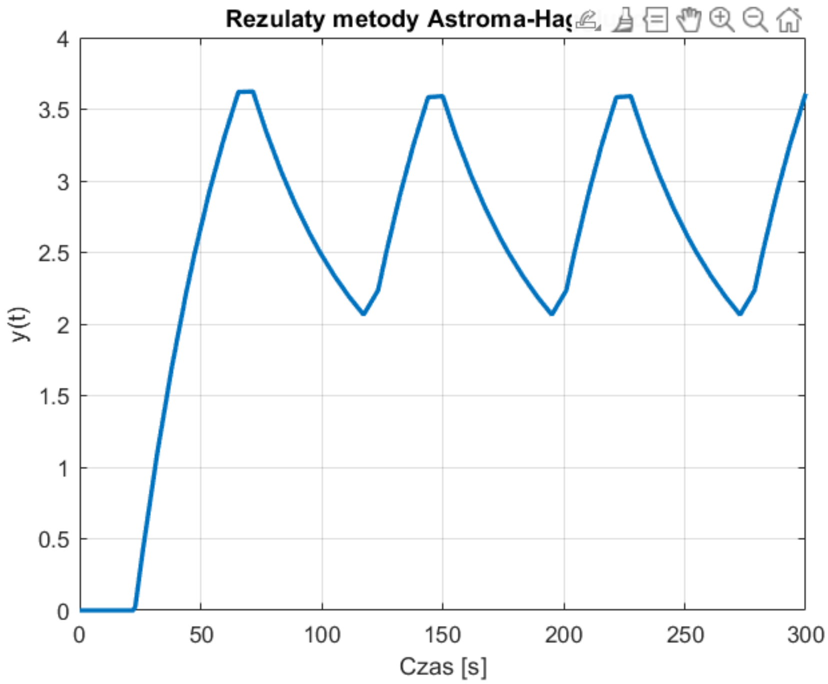
```
Ti = 0.5 * Tosc
```

```
Ti = 38.6740
```

```
Td = 0.12 * Tosc
```

```
Td = 9.2818
```

```
ki = kp/Ti;
kd = kp/Td;

out = sim("model2.slx");
plot(out.y.time, out.y.signals.values, "LineWidth", 2)
title("Rezulaty metody Astroma-Hagglunda")
ylabel("y(t)")
xlabel("Czas [s]")
grid on

[time, y] = ginput(2)
```




```
time = 2×1
   69.3370
  147.2376
y = 2×1
    3.6121
    3.5841
```

```
Tosc = time(2) - time(1)
```

```
Tosc = 77.9006
```

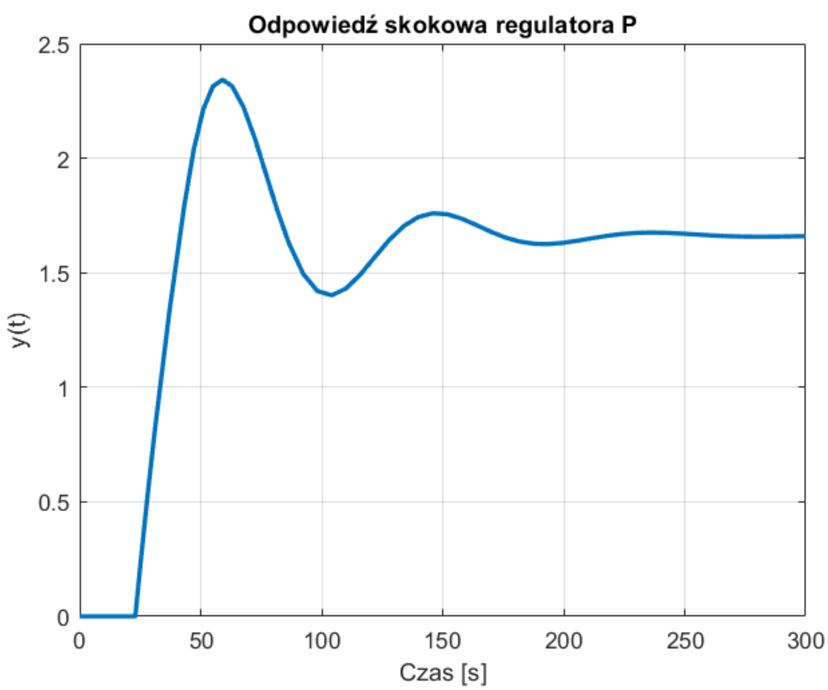
Regulator P

```
k = 1.18;
tau = 22;
T = 45;
r = 2.5;
kr = 3.359;
kp = 0.5 * kr
```

```
kp = 1.6795
```

```
Ti = 0;
```

```
Td = 0;
ki = 0;
kd = 0;

out = sim("model1.slx");
plot(out.y.time, out.y.signals.values, "LineWidth", 2)
title("Odpowiedź skokowa regulatora P")
ylabel("y(t)")
xlabel("Czas [s]")
grid on
```



Regulator PI

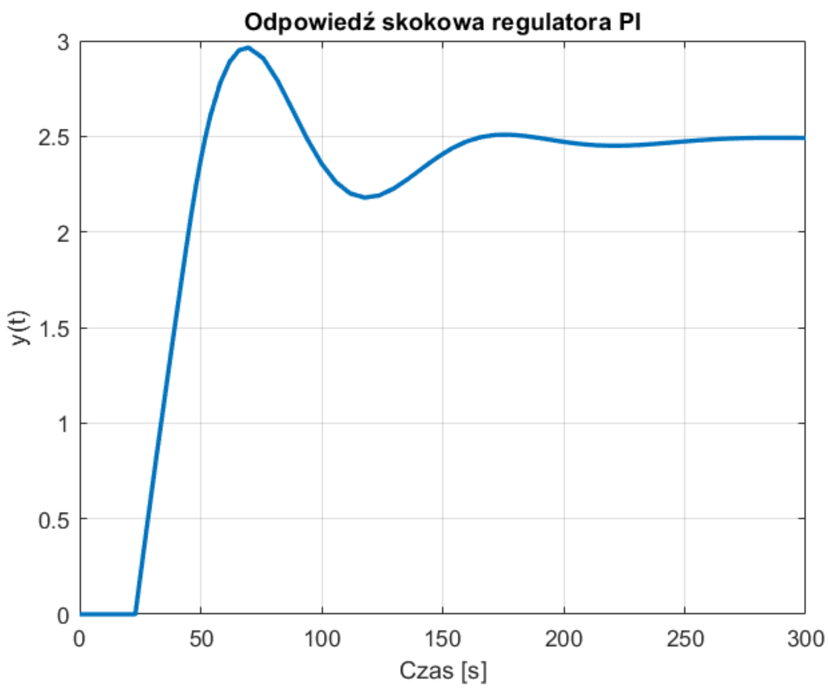
```
k = 1.18;
tau = 22;
T = 45;
r = 2.5;
kr = 3.359;
kp = 0.45 * kr
```

kp = 1.4850

```
Ti = 0.85 * Tosc
```

Ti = 65.7459

```
Td = 0;
ki = kp/Ti;
kd = 0;

out = sim("model1.slx");
plot(out.y.time, out.y.signals.values, "LineWidth", 2)
title("Odpowiedź skokowa regulatora PI")
ylabel("y(t)")
xlabel("Czas [s]")
grid on
```



Regulator PID

```
k = 1.18;
tau = 22;
T = 45;
r = 2.5;
kr = 3.359;
kp = 0.6 * kr
```

```
kp = 2.0154
```

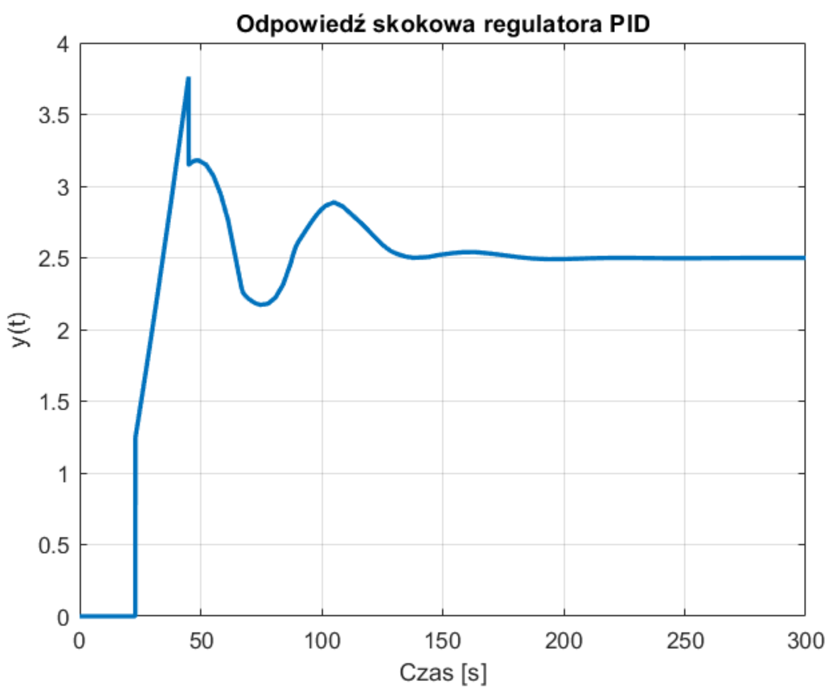
```
Ti = 0.5 * Tosc
```

Ti = 38.9503

```
Td = 0.12 * Tosc
```

Td = 9.3481

```
ki = kp/Ti;
kd = kp*Td;

out = sim("model1.slx");
plot(out.y.time, out.y.signals.values, "LineWidth", 2)
title("Odpowiedź skokowa regulatora PID")
ylabel("y(t)")
xlabel("Czas [s]")
grid on
```



Dostrajanie regulatora PID na podstawie parametrów transmitancji zastępczej

```
k = 1.18;
tau = 22;
T = 45;
r = 2.5;
kr = 3.359;
kp = (0.95*T) / (tau*k);
```

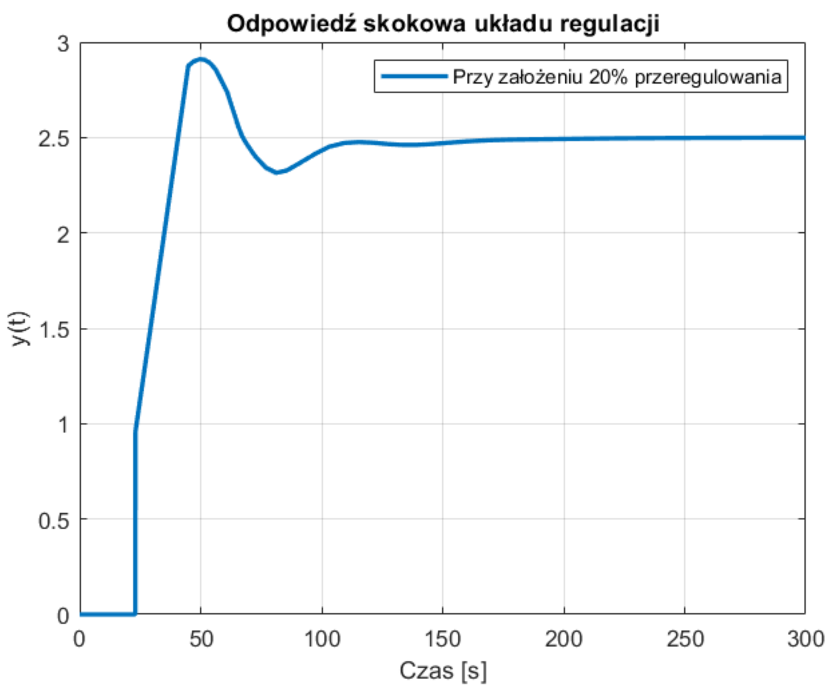
```
Ti = 0.5 * Tosc
```

Ti = 38.9503

```
Td = 0.12 * Tosc
```

Td = 9.3481

```
ki = kp/(2.4 * tau);
kd = kp*0.4*tau;

out = sim("model1.slx");
plot(out.y.time, out.y.signals.values, "LineWidth", 2)
title("Odpowiedź skokowa układu regulacji")
legend("Przy założeniu 20% przeregulowania")
ylabel("y(t)")
xlabel("Czas [s]")
grid on
```




```
k = 1.18;
tau = 22;
T = 45;
r = 2.5;
kr = 3.359;
kp = (1.4*T) / (tau*k);
Ti = 0.5 * Tosc
```

```
Ti = 38.9503
```

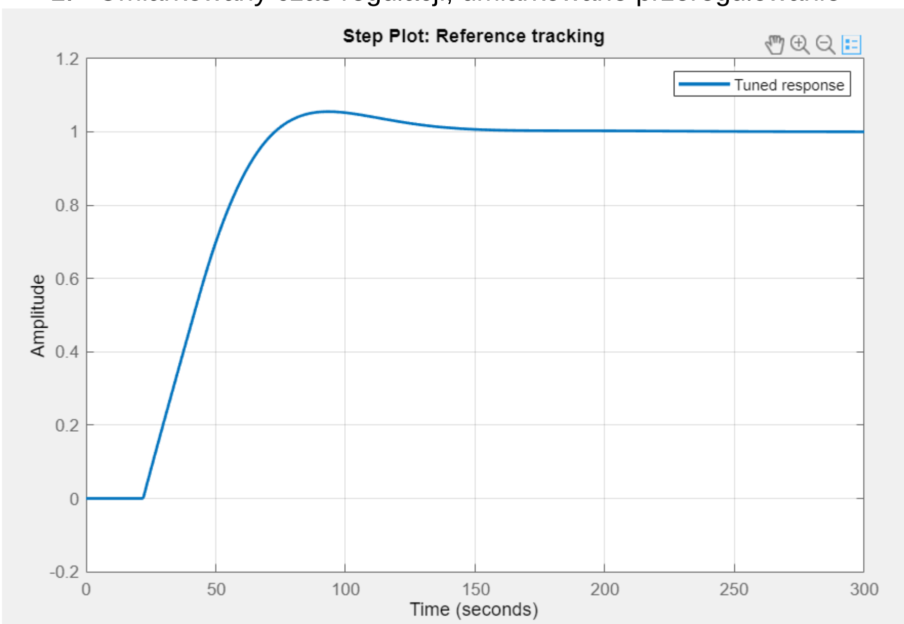
```
Td = 0.12 * Tosc
```

```
Td = 9.3481
```

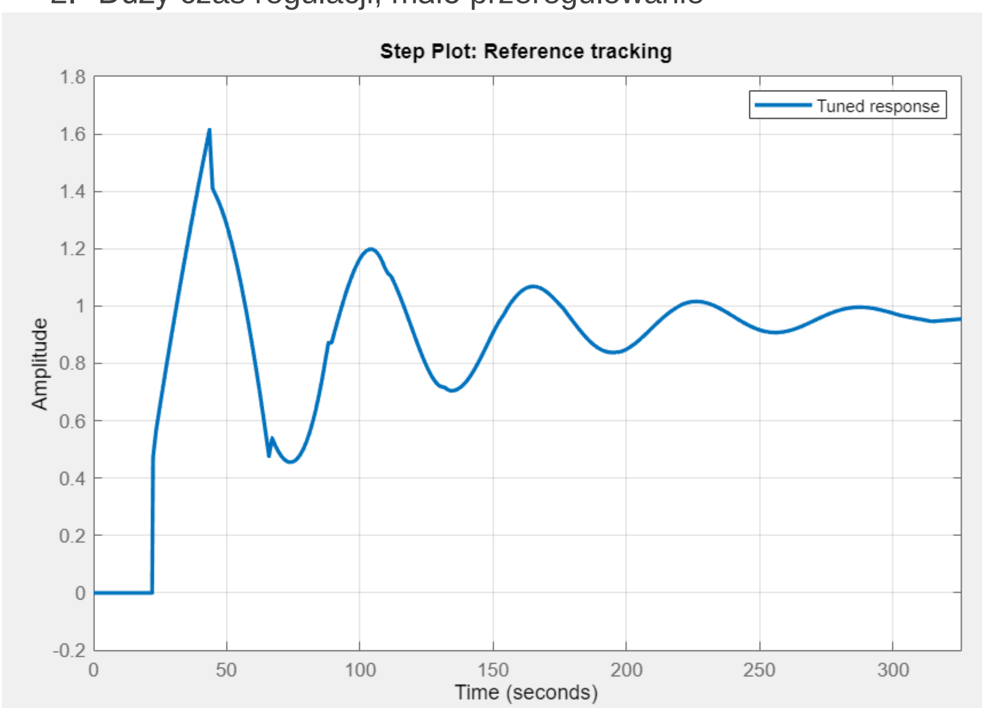
```
ki = kp/(1.3 * tau);
kd = kp*0.5*tau;

out = sim("model1.slx");
plot(out.y.time, out.y.signals.values, "LineWidth", 2)
title("Odpowiedź skokowa układu regulacji")
legend("Przy założeniu minimum z całki kwadratu uchybu")
ylabel("y(t)")
xlabel("Czas [s]")
grid on
```

Użycie funkcji „Autotune” dostępnej w środowisku SIMULINK

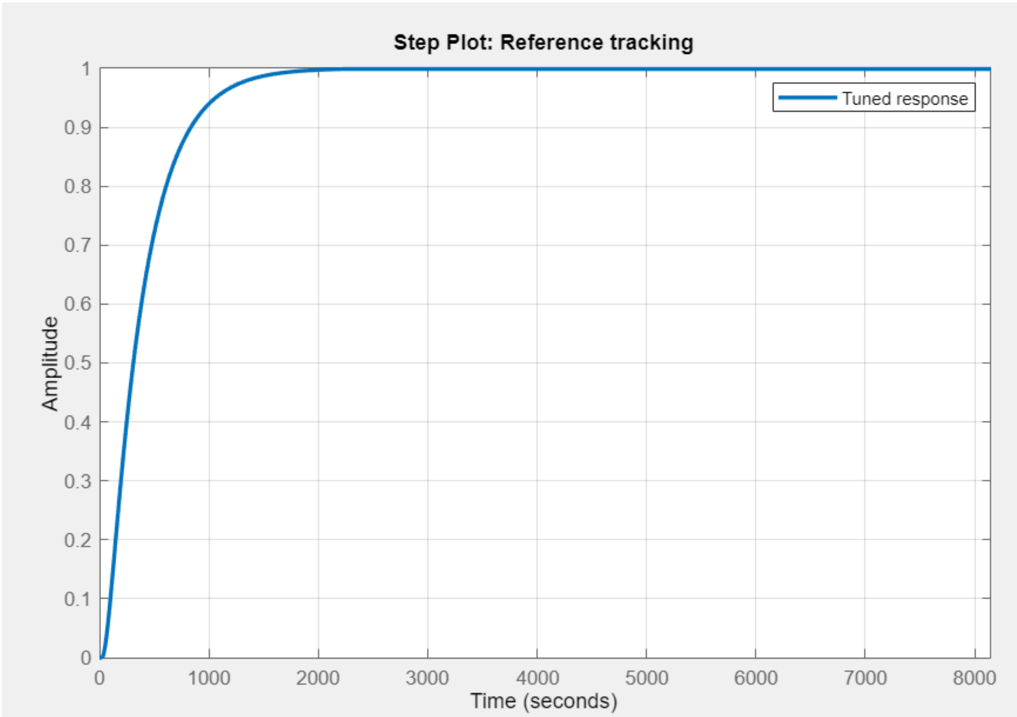
1. Umiarkowany czas regulacji, umiarkowane przeregulowanie



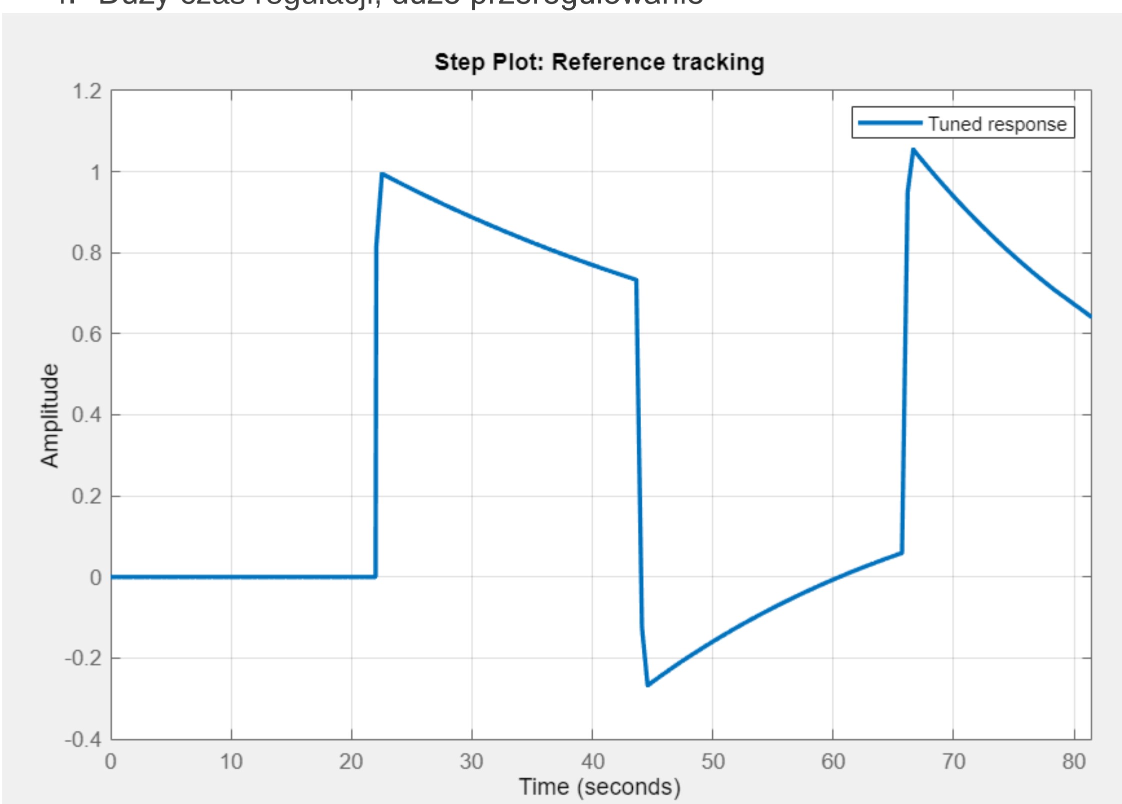
2. Duży czas regulacji, małe przeregulowanie



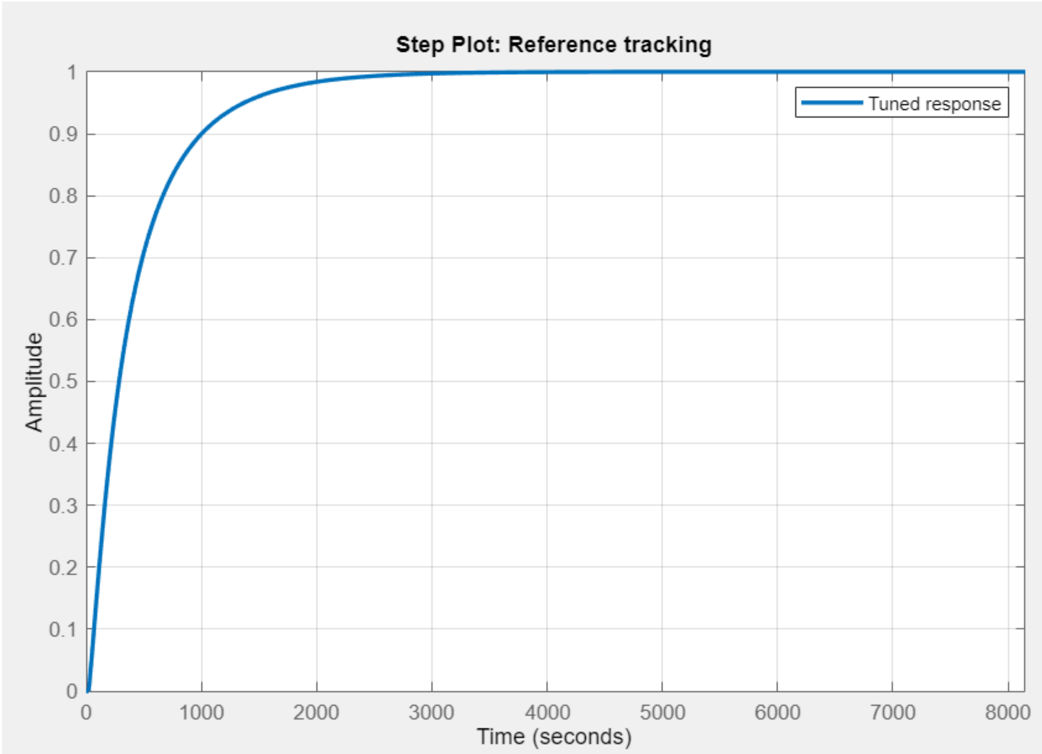
3. Mały czas regulacji, małe przeregulowanie



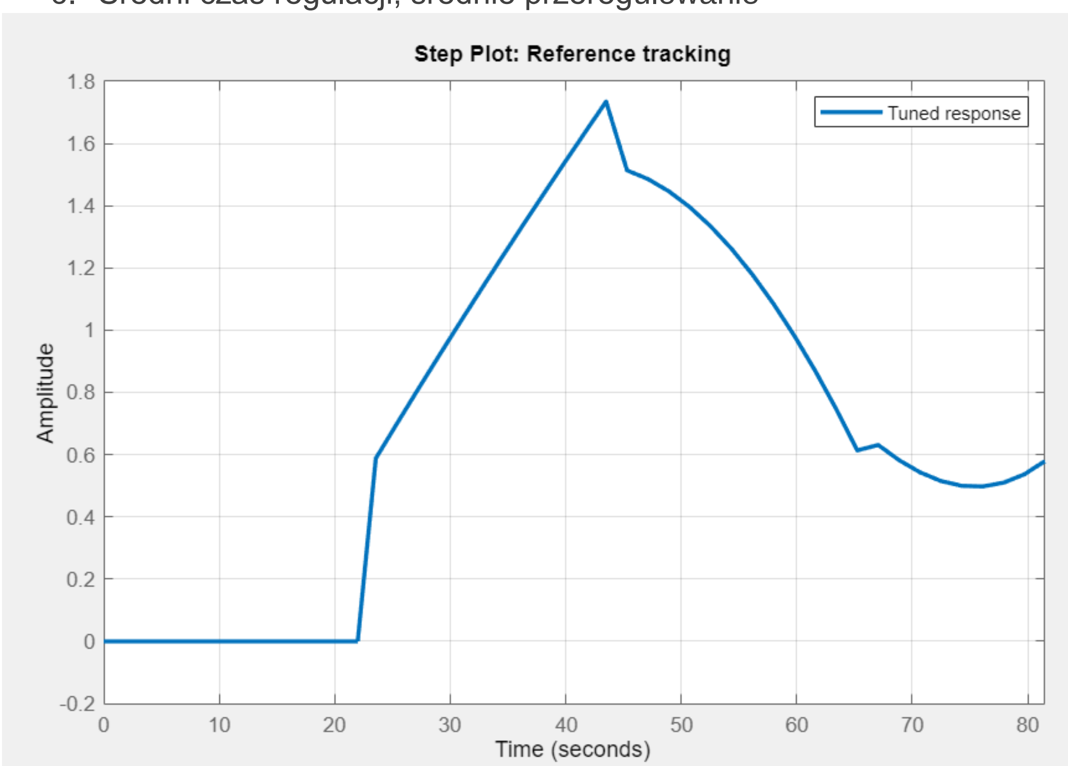
4. Duży czas regulacji, duże przeregulowanie



5. Mały czas regulacji, duże przeregulowanie



6. Średni czas regulacji, średnie przeregulowanie



4. Wnioski

- Podczas wykonywania tego ćwiczenia poznałem zastosowanie nowej funkcji Matlaba - „ginput( )”, która okazała się bardzo przydatna do odczytywania i zapisania np. wartości okresu czy amplitudy sygnału z wykresu.
- Wyniki z dostrajania metodą Zieglera-Nicholsa i Astroma-Haggunda są niemal identyczne. Dla modelu transmitancji zastępczej dla regulatora PID otrzymano szybsze ustanie oscylacji, ustabilizowanie przebiegu oraz zmniejszyło największą możliwą amplitudę.
- Metoda Autotune dała bardzo dobre efekty. Widoczne jest bardzo małe przeregulowanie i dobry czas regulacji. Ponadto można wybrać charakter odpowiedzi. Jest także bardzo przyjemna w użyciu i najszybsza w realizacji.
- Najlepszy efekt udało się uzyskać dostrajając regulator PID na podstawie parametrów transmitancji zastępczej, przy założeniu przeregulowania 20% oraz minimalnego czasu regulacji. Widoczne jest tylko jedno większe przeregulowanie na samym początku, ale potem układ szybko się stabilizuje i osiąga wartość zadaną.