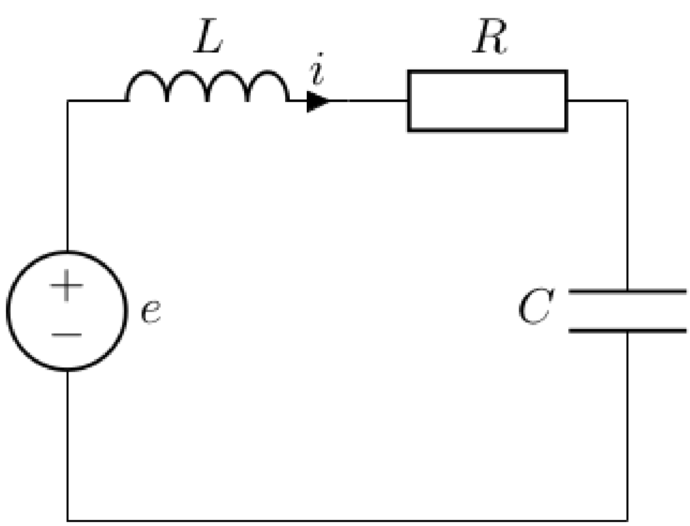
Podstawy Sterowania Optymalnego
Labolatorium 4
Regulator PID - Optymalizacja układów liniowych.

Prowadzący: mgr inż. Krzysztof Hałas
Wykonał: Ryszard Napierała

8 grudnia 2021

Zadanie 5



Rysunek 1: Układ RLC

1. **W języku Python zaimportować biblioteki *numpy, scipy.signal, scipy.itegrate.solve_ivp, scipy.integrate.odeint*, a następnie *matplotlib.pyplot***

```
import numpy as np
from scipy import signal
import matplotlib.pyplot as plt
```

2. **Wprowadzić zmienne regulatora PID $K_p = 2$, $T_I = 1$, $T_D = 4$ oraz parametry obiektu z rysunku 1.**

```
Kp = 2
Ti = 1
Td = 0.4

lti = signal.lti([], [-3]*3, 5)
Go = signal.TransferFunction(lti)
A = np.array([
    [0, 1, 0],
    [0, 0, 1],
    [-1, -3, -3]
])
B = np.array([[1], [0], [0]])
C = np.array([[5]])
D = np.array([[0]])
```

Output:

```
num = [5.]
den = [ 1.   9. 27. 27.]
```

3. **Wyznaczyć i zaimplementować transmitancję układu zamkniętego. Wykorzystać w tym celu funkcję *signal.TransferFunction(num,den)*.**

$$G_{PID} = K_p(1 + \frac{1}{sT_I} + sT_D) = \frac{K_pT_Ds^2 + K_ps + \frac{K_p}{T_I}}{s}$$

$$G_o = \frac{5}{s^3 + 9s^2 + 27s + 27}$$

$$G_u = G_{PID} * G_o$$

$$G_u = \frac{5(K_pT_Ds^2 + K_ps + \frac{K_p}{T_I})}{s(s^3 + 9s^2 + 27s + 27)} = \frac{5K_pT_Ds^2 + 5K_ps + \frac{5K_p}{T_I}}{s^4 + 9s^3 + 27s^2 + 27s}$$

$$G_z = \frac{G_u}{1 + G_u}$$

$$G_z = \frac{\frac{5K_pT_Ds^2 + 5K_ps + \frac{5K_p}{T_I}}{s^4+9s^3+27s^2+27s}}{1 + \frac{5K_pT_Ds^2 + 5K_ps + \frac{5K_p}{T_I}}{s^4+9s^3+27s^2+27s}} = \frac{5K_pT_Ds^2 + 5K_ps + \frac{5K_p}{T_I}}{s^4 + 9s^3 + 27s^2 + 27s + 5K_pT_Ds^2 + 5K_ps + \frac{5K_p}{T_I}}$$

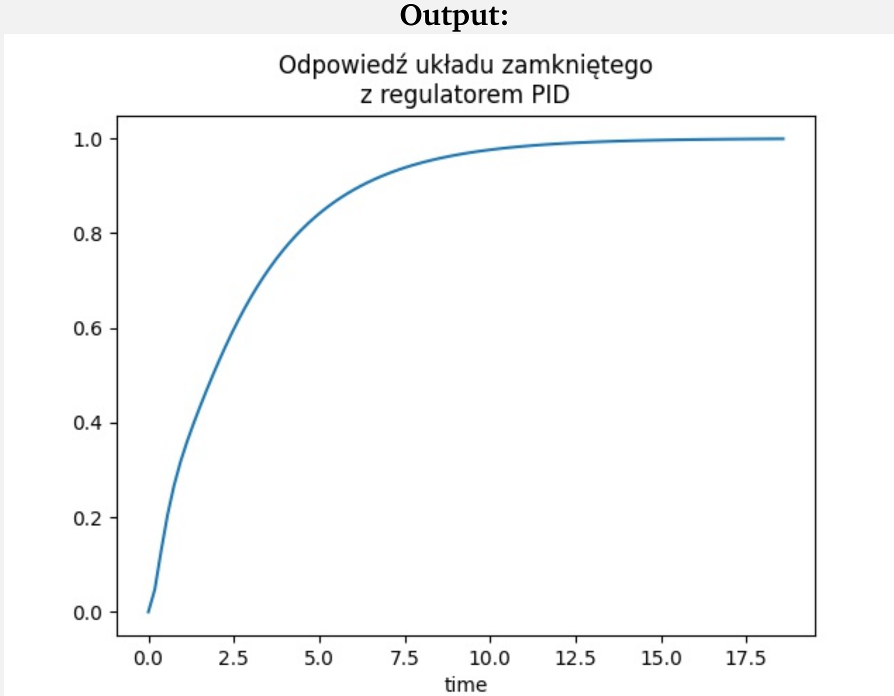
$$G_z = \frac{5K_pT_Ds^2 + 5K_ps + \frac{5K_p}{T_I}}{s^4 + 9s^3 + (5K_pT_D + 27)s^2 + (5K_p + 27)s + \frac{5K_p}{T_I}}$$

```python
def closed_loop_with_PID(Kp: float, Td: float, Ti: float) -> signal.TransferFunction:
    G = signal.TransferFunction(
        [5*Kp*Td, 5*Kp, 5*Kp/Ti],
        [1, 9, 5*Kp*Td+27, 5*Kp+27, 5*Kp/Ti]
    )
    return G

Gz = closed_loop_with_PID(Kp, Td, Ti)
```

4. **Wyznaczyć odpowiedź skokową układu (można w tym celu wykorzystać polecenie *signal.step()*) oraz utworzyć jej wykres, przykładowo poleceniem *matplotlib.pyplot***

```python
x, y = signal.step(Gz)
plt.plot(x, y)
plt.xlabel('time')
plt.title('Odpowiedź układu zamkniętego\nz regulatorem PID')
plt.show()
plt.close()
```

Output:



Czy układ z regulatorem PID z losowo dobranymi wartościami parametrów jest stabilny?
Układ z regulatorem PID z losowo dobranymi wartościami parametrów nie koniecznie będzie stabilny.

5. **Wyznaczyć równania stanu na podstawie transaminacji. Wykorzystać w tym celu funkcje *signal.tf2ss(num, den)*.**

```
sysz = Gz.to_ss()
print('A =', sysz.A)
print('B =', sysz.B)
print('C =', sysz.C)
print('D =', sysz.D)
```

Output:

```
A = [[  −9.  −31.  −37.  −10.]
    [   1.    0.    0.    0.]
    [   0.    1.    0.    0.]
    [   0.    0.    1.    0.]]
B = [[1.]
    [0.]
    [0.]
    [0.]]
C = [[  0.   4.  10.  10.]]
D = [[0.]]
```

Zadanie 6

1. **Zapisać system w postaci fazowych zmiennych stanu wykorzystując polecenie *signal.StateSpace(A,B,C,D)* Wyznaczyć i wykreślić odpowiedź skokową układu**

```
x, y = signal.step(Go.to_ss(), T=t)
```

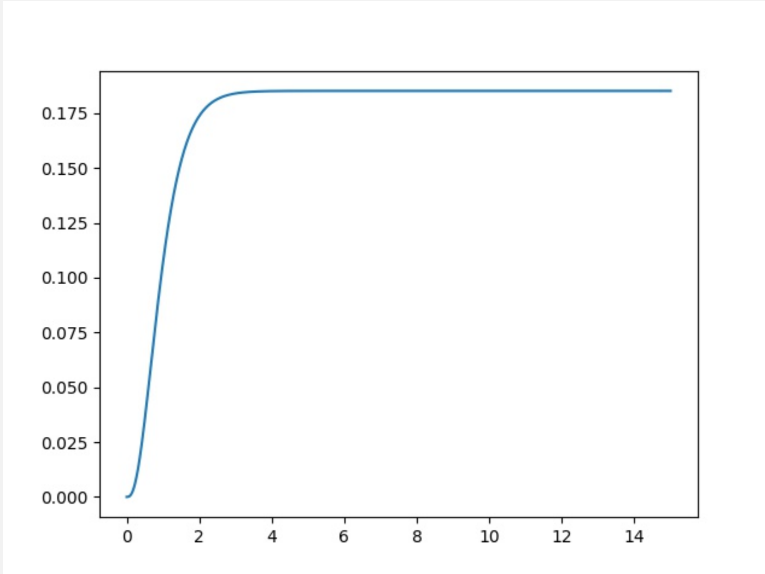
2. **Utworzyć tablicę wartości czasu $t \in (0, 15)$.**

```
t = np.linspace(0, 15, 1000)
```

3. **Wykreślić rozwiązanie równania dla pobudzenia skokiem jednostkowym, wykorzystujące polecenie *matplotlib.pyplot***

```
plt.plot(x, y)
plt.show()
plt.close()
```

Output:

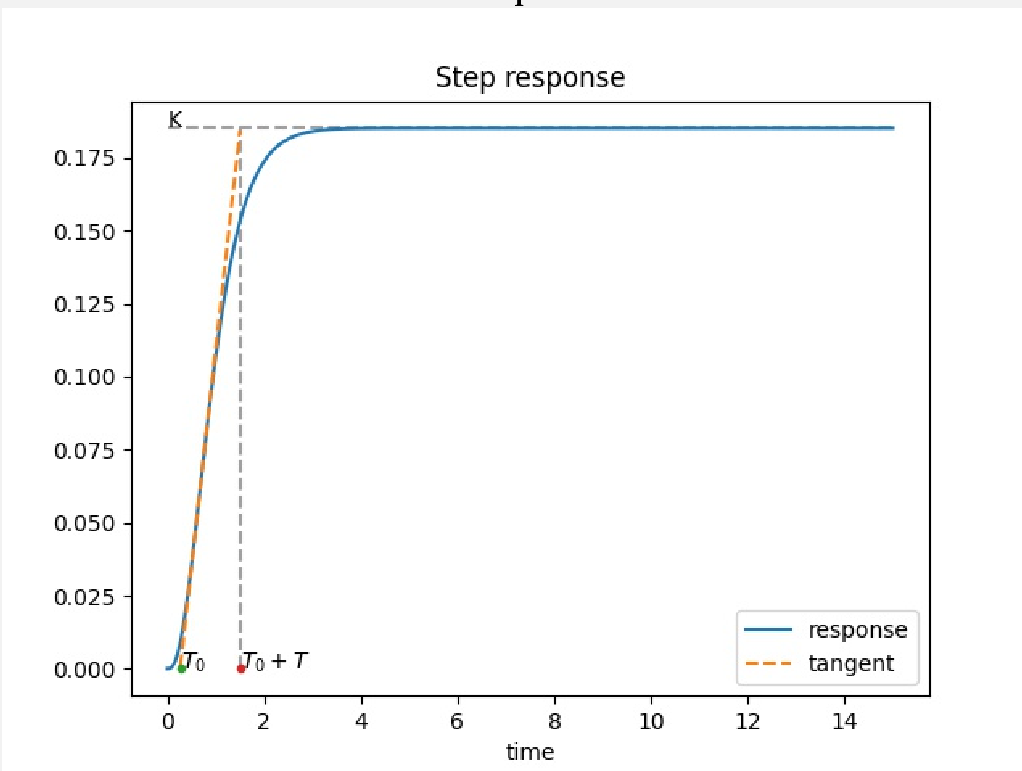


4. **Wyznaczyć nastawy K_p, T_i, T_d zgodnie z równaniem.**

```
# Wyznaczenie wartości K
K = np.max(y)

# Obliczanie punktu przegięcia
dy = y[1:] - y[:-1]
ddy = dy[1:] - dy[:-1]
ddy_zero = np.isclose(ddy, np.zeros_like(ddy), atol=1e-5)
x_ddy_zero_lok = np.where(ddy_zero == True)[0][0]
print(x_ddy_zero_lok)
x_inflect = x[x_ddy_zero_lok: x_ddy_zero_lok + 2]
y_inflect = y[x_ddy_zero_lok: x_ddy_zero_lok + 2]

# wyznaczanie stycznej do wykresu w punkcie przegięcia
tangent = np.poly1d(np.polyfit(x_inflect, y_inflect, 1))
y_tangent = tangent(x)

# Wyznaczenie wartości T0
T0_where = np.where(y_tangent >= 0)[0][0]
T0 = x[T0_where]

# Wyznaczenie wartości T
T_where = np.where(y_tangent <= K)[0][-1]
T_x = x[T_where]
T = T_x - T0

plt.plot(x, y)
plt.plot(x[T0_where:T_where+1], y_tangent[T0_where:T_where+1], '--')
plt.plot(T0, 0, '.')
plt.plot(T_x, 0, '.')
```

```
plt.hlines(K, x[0], x[-1], '#A0A0A0', '--')
plt.vlines(T_x, y[0], y[-1], '#A0A0A0', '--')
plt.annotate('K', (0, K))
plt.annotate('$T_0$', (T0, 0))
plt.annotate('$T_0+T$', (T_x, 0))
plt.title('Step response')
plt.xlabel('time')
plt.legend(['response', 'tangent'])
plt.show()
plt.close()
```

Output:



Output:

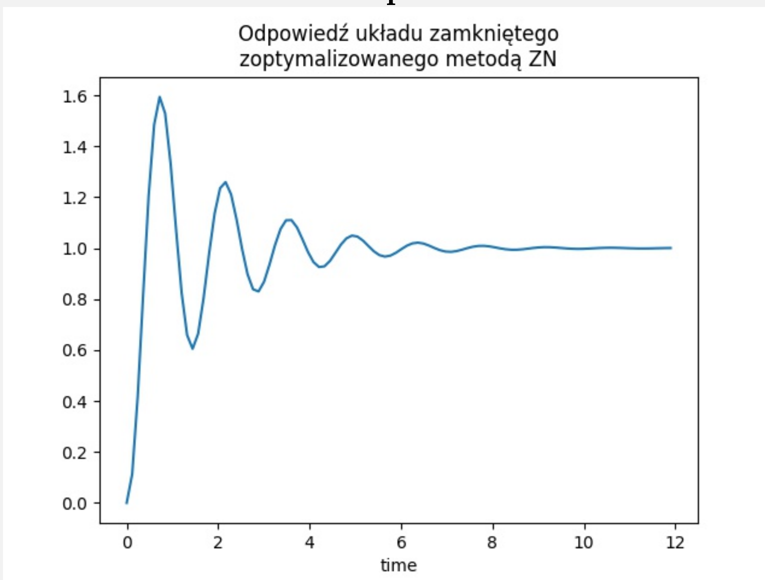
```
T0/T =  0.21951219512195122
Kp =  29.520000000000408
Ti =  0.5405405405405406
Td =  0.10810810810810811
```

5. **Zbadać odpowiedź skokową układu zamkniętego (zoptymalizowanego metodą ZN).**

```
syszn = closed_loop_with_PID(Kp, Td, Ti).to_ss()
x, y = signal.step(syszn)
plt.plot(x, y)
```

```python
plt.xlabel('time')
plt.title('Odpowiedź układu zamkniętego\nzoptymalizowanego metodą ZN')
plt.show()
plt.close()
```

Output:



Jakie są właściwości odpowiedzi skokowej układu zamkniętego dla tak dobranych nastaw regulatora?
Duże przeregulowanie i długo gasnące drgania.

6. **Wyznaczyć wartość kryterium całkowego IAE.**

```python
e = np.ones_like(y) - y
print('I_IAE = ', np.sum(e))


class Experiment:

    def __init__(self, Kp: float, Td: float, Ti: float) -> None:
        self.Kp = Kp
        self.Td = Td
        self.Ti = Ti
        x, y = signal.step(self.sys)
        self.e = self.error(x, y)

    @property
    def sys(self) -> signal.StateSpace:
        return closed_loop_with_PID(
            self.Kp, self.Td, self.Ti
        ).to_ss()

    def error(self, x: np.ndarray, y: np.ndarray) -> float:
```

```
        return np.sum(np.abs(np.ones_like(y) - y))

    def __str__(self) -> str:
        return f"Kp={self.Kp:.3f}, Ti={self.Ti:.3f}, Td={self.Td:.3f}, e={self.e:.3f}


experiments = []

for Kp in np.linspace(0.001, 50, 20):
    for Ti in np.linspace(0.001, 5, 20):
        for Td in np.linspace(0.001, 5, 20):
            experiments.append(Experiment(Kp, Td, Ti))

sorted_exp = sorted(experiments, key=lambda x: x.e)

print('Wartości dla najniższego znalezionego experymentalnie\
kryterium IAE to: ', sorted_exp[0])
plt.plot(*signal.step(sorted_exp[0].sys))
plt.title('Odpowiedź skokowa dla experymentalnie\n\
znalezionych wartości PID minimalizując IAE')
plt.show()
plt.close()
```

Output:

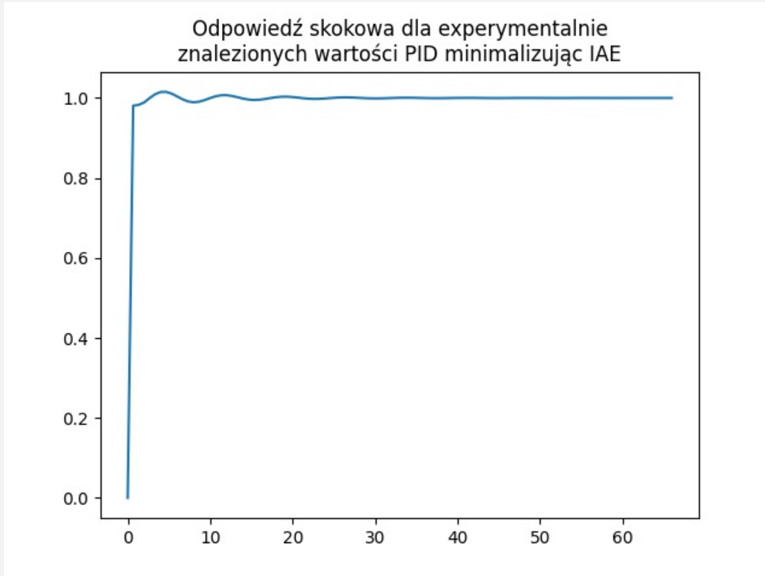
```
I_IAE =  1.323539452930474
```

Czy można ustawić inne wartości parametrów regulatora PID tak, aby wartość kryterium IAE była mniejsza?
Tak.
Spróbować również ręcznie metodą prób i błędów znaleźć nastawy które pozwalają na lepszą regulację oraz sprawdzić za pomocą kryterium IAT.

Output:

```
Wartosci dla najnizszego znalezionego experymentalnie kryterium IAE to:
Kp=50.000, Ti=0.264, Td=5.000, e=1.217.
```

Output:



7. **Przeprowadzić strojenie regulatora PID drugą metodą Zieglera-Nicholsa.**

```python
# Poszukiwanie K krytycznego
slope = np.inf
Kkr = 0
for k in np.linspace(30, 50, 1000):
    ss = closed_loop_with_PID(k, 0, np.inf).to_ss()
    x, y = signal.step(ss)
    peaks_pos = signal.find_peaks(y)[0]
    peaks = y[peaks_pos]
    sl = np.sum(np.abs(peaks[1:] - peaks[:-1]))
    if sl < slope:
        slope = sl
        Kkr = k
print('K krytyczne = ', Kkr)
sysKkr = closed_loop_with_PID(Kkr, 0, np.inf).to_ss()

# Wyznaczenie okresu oscylacji
x, y = signal.step(sysKkr)
peaks_pos = signal.find_peaks(y)[0]
Tosc = x[peaks_pos[3]] - x[peaks_pos[2]]
print('Tosc = ', Tosc)

# Wyznaczenie nastaw regulatora PID
Kp = 0.6*Kkr
Ti = 0.5*Tosc
Td = Tosc/8
print(f'Wyznaczone parametry drugą metodą \
Zieglera - Nicholsa to: Kp={Kp:.3f}, Ti={Ti:.3f}, Td={Td:.3f}')
```

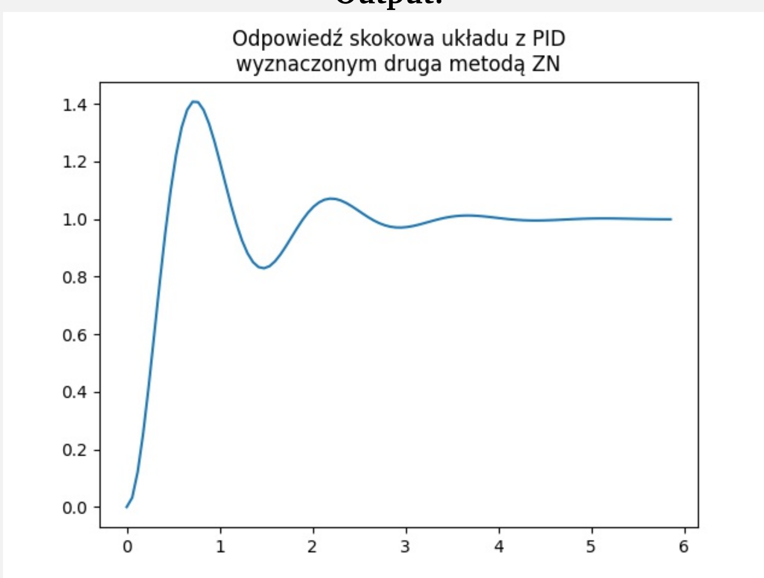
Output:

```
K krytyczne =  43.293293293293296
Tosc =  1.202020202020202
Wyznaczone parametry druga metoda Zieglera−Nicholsa to:
Kp=25.976, Ti=0.601, Td=0.150
```

8. **Wykreślić odpowiedź skokową układu zamkniętego.**

```python
syszn2 = closed_loop_with_PID(Kp, Td, Ti).to_ss()
x, y = signal.step(syszn2)
plt.plot(x, y)
plt.title('Odpowiedź skokowa układu z PID\nwyznaczonym druga metodą ZN')
plt.show()
plt.close()
```

Output:



9. **Porównać odpowiedzi skokowe dla dwóch zestawów nastaw: z pierwszej i drugiej metody Zieglera-Nicholsa.**
   Jakie różnice są zauważalne dla obu uzyskanych przebiegów? Odpowiedź dla pierwszej metody ma większe przeregulowanie oraz dłużej się stabilizuje niż dla drugiej metody

10. **Dla rozważanego układ, dobrać parametry regulatora PID tak, aby zminimalizować kry- terium ITSE.**
    Czy ten sam zestaw parametrów regulatora PID odpowiada minimalnym warto- ściom wszystkich rozważanych kryteriów całkowych? Spróbować również ręcznie metodą prób i błędów znaleźć nastawy które pozwalają na lepszą regulację oraz sprawdzić za pomocą kryterium ITSE

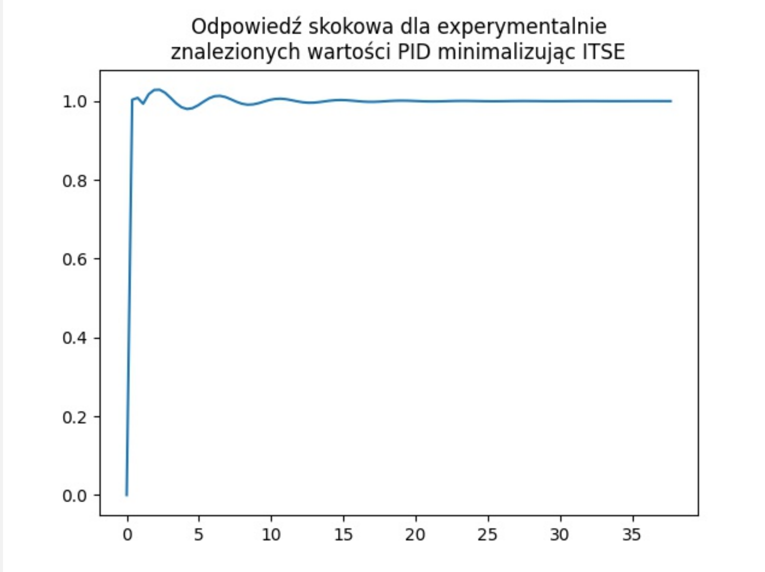
```python
class Experiment2(Experiment):

```

```python
    def error(self, x: np.ndarray, y: np.ndarray) -> float:
        return np.sum(x*((np.ones_like(y) - y)**2))


zn2_itse = Experiment2(Kp, Td, Ti)
print('Nastawy PID oraz wartość kryterium ITSE \
dla drugiej metody Zieglera-Nicholsa to: ', zn2_itse)

experiments = []
for Kp in np.linspace(15, 35, 20):
    for Ti in np.linspace(0.001, 3, 20):
        for Td in np.linspace(0.001, 3, 20):
            experiments.append(Experiment2(Kp, Ti, Td))

sorted_exp = sorted(experiments, key=lambda x: x.e)
print('Znalezione experymentalnie nastawy PID minimalizujące\
ITSE to: ', sorted_exp[0])
x, y = signal.step(sorted_exp[0].sys)
plt.plot(x, y)
plt.title('Odpowiedź skokowa dla experymentalnie\
\nznalezionych wartości PID minimalizując ITSE')
plt.show()
plt.close()
```

Output:

```
Nastawy PID oraz wartosc kryterium ITSE dla drugiej metody
Zieglera-Nicholsa to: Kp=25.976, Ti=0.601, Td=0.150, e=1.512.
Znalezione experymentalnie nastawy PID
minimalizujace ITSE to: Kp=35.000, Ti=0.159, Td=2.684, e=0.018.
```

Output:



Zadanie 7

1. **Dokonać regulacji podonie jak w przpadku pierwszej metody ZN**

```
print(f'R = {T/T0:.3f}')


class Experiment3(Experiment):

    def error(self, x: np.ndarray, y: np.ndarray) -> float:
        return np.sum((np.ones_like(y)-y)**2)


exp_a = Experiment3(0.3*T/(K*T0), 0.5*T0, T)
exp_20 = Experiment3( 0.7*T/(K*T0), 4.7*T0, 1.4*T)
```

Output:

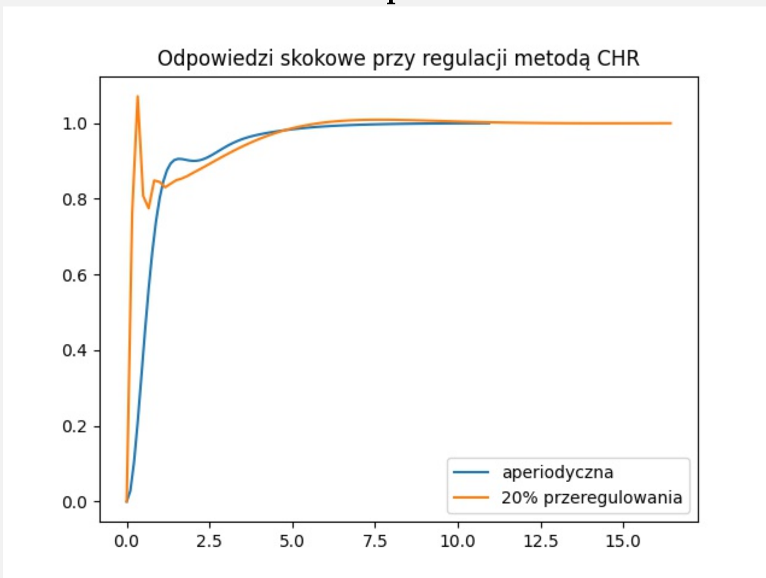
R = 4.556

2. **Wykreślić odpowiedź skokową układu zamkniętego z regulatorem PID.**

```
res_a = signal.step(exp_a.sys)
res_20 = signal.step(exp_20.sys)
plt.plot(*res_a, *res_20)
plt.title('Odpowiedzi skokowe przy regulacji metodą CHR')
plt.legend(['aperiodyczna', '20% przeregulowania'])
```

```
plt.show()
plt.close()
```

Output:



3. **Na podstawie kryterium ISE (9) sprawdzić jakość regulacji**

```
print('Wartości dla regulatora CHR aperiodycznego, \
oraz błąd ISE to: ', exp_a)
print('Wartości dla regulatora CHR z przeregulowaniem 20%, \
oraz błąd ISE to: ', exp_20)

experiments = []
for Kp in np.linspace(2, 25, 20):
    for Ti in np.linspace(0.5, 3, 20):
        for Td in np.linspace(0.001, 3, 20):
            experiments.append(Experiment3(Kp, Ti, Td))

sorted_exp = sorted(experiments, key=lambda x: x.e)
print('Znalezione experymentalnie nastawy PID \
minimalizujące ISE to: ', sorted_exp[0])
x, y = signal.step(sorted_exp[0].sys)
plt.plot(x, y)
plt.title('Odpowiedź skokowa dla experymentalnie\
\nznalezionych wartości PID minimalizując ISE')
plt.show()
plt.close()
```

Output:

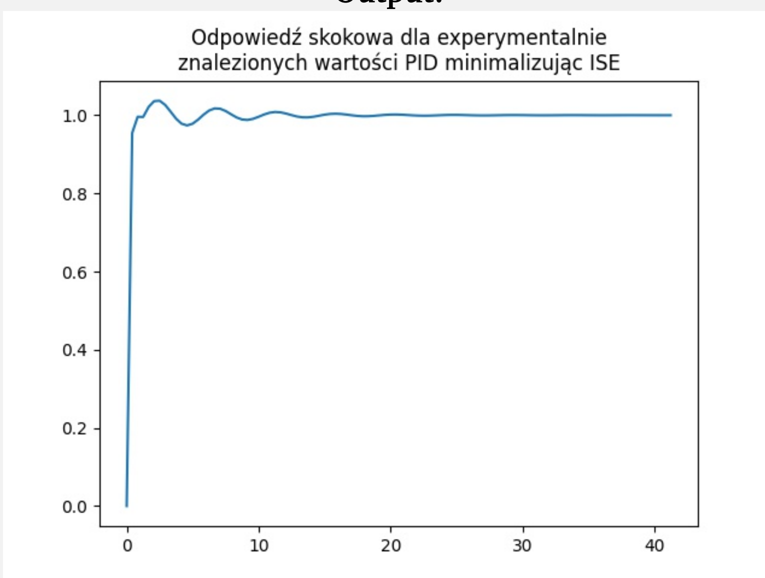
```
Wartosci dla regulatora CHR aperiodycznego, oraz blad ISE to:
Kp=7.380, Ti=1.231, Td=0.135, e=4.737.
Wartosci dla regulatora CHR z przeregulowaniem 20%,
oraz blad ISE to:  Kp=17.220, Ti=1.724, Td=1.270, e=1.426.
```

Czy można ustawić inne wartości parametrów regulatora PID tak, aby wartość kryterium ISE była mniejsza?
Tak
Spróbować również ręcznie metodą prób i błędów znaleźć nastawy którepozwalają na lepszą regulację oraz sprawdzić za pomocą kry- terium ISE

Output:

```
Znalezione experymentalnie nastawy PID minimalizujace ISE to:
Kp=25.000, Ti=0.159, Td=3.000, e=1.009.
```

Output:



4. **Porównać odpowiedzi skokowe dla trzech metod strojenia.**
Wybrać jedno z kryteriów optymalności i sprawdzić która z metod sprawdza się najlepiej dla przedstawionego regulatora
To która metoda najlepiej się sprawdza jest zależne od przyjętych kryteriów, jakie ma spełniać układ regulacji. Jednakże metoda CHR wydaje mi się być najlepsza.