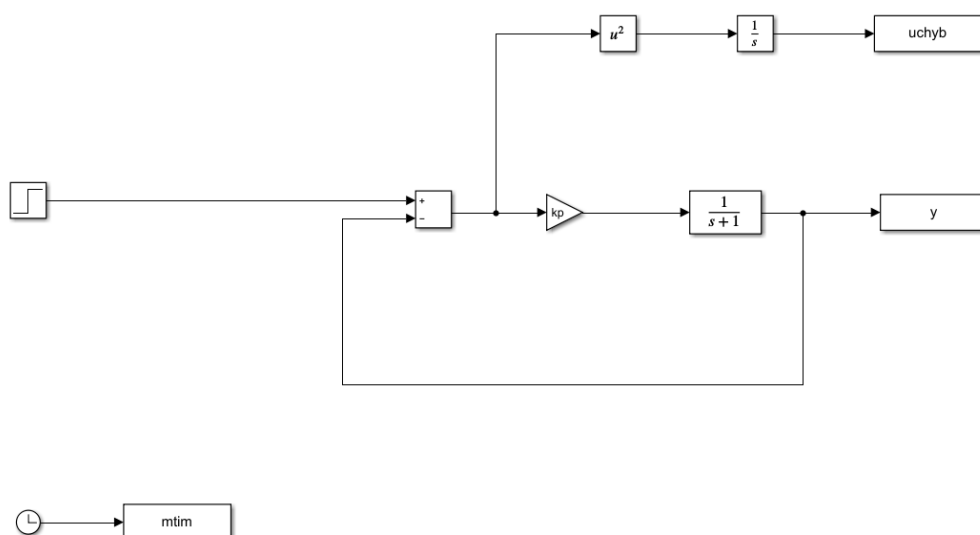


Laboratorium: Układy o złożonej strukturze połączeń, układy automatycznej regulacji

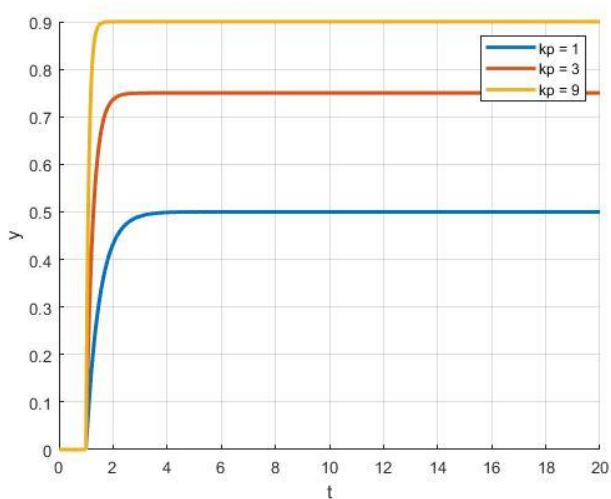
1. Badanie odpowiedzi układu automatycznej regulacji na podanie skoku jednostkowego, dla różnych parametrów wzmocnienia regulatora P

1.1. Schemat badanego układu

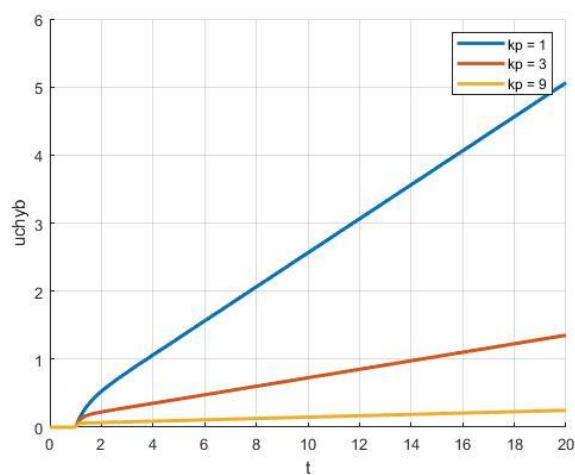


1. Schemat badanego układu

1.2. Odpowiedź układu



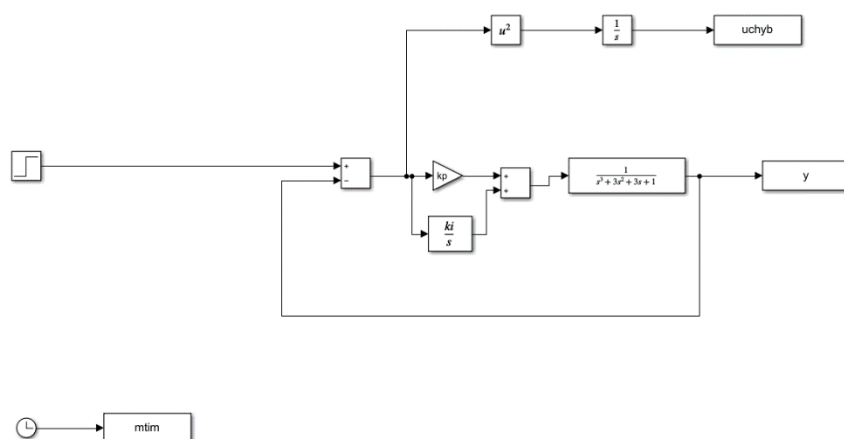
3. Odpowiedź układu



2. Całka kwadratu uchybu układu

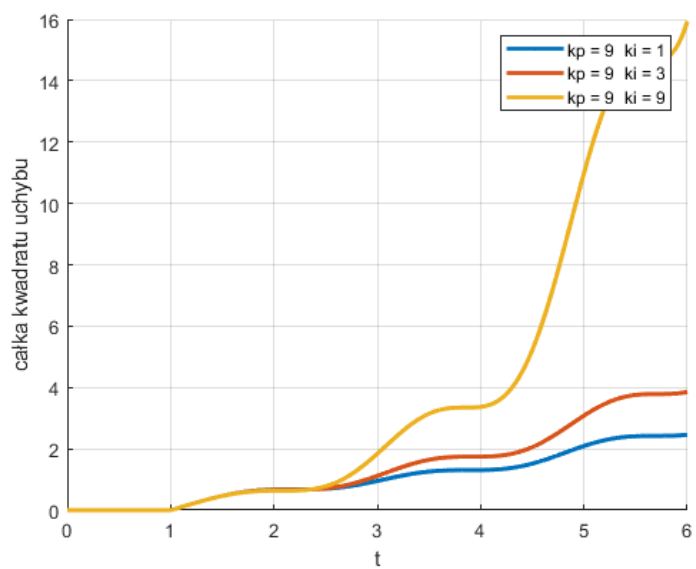
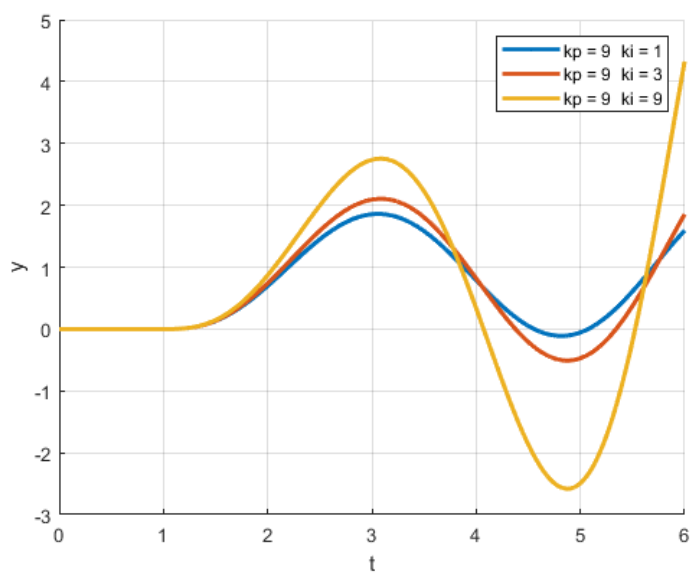
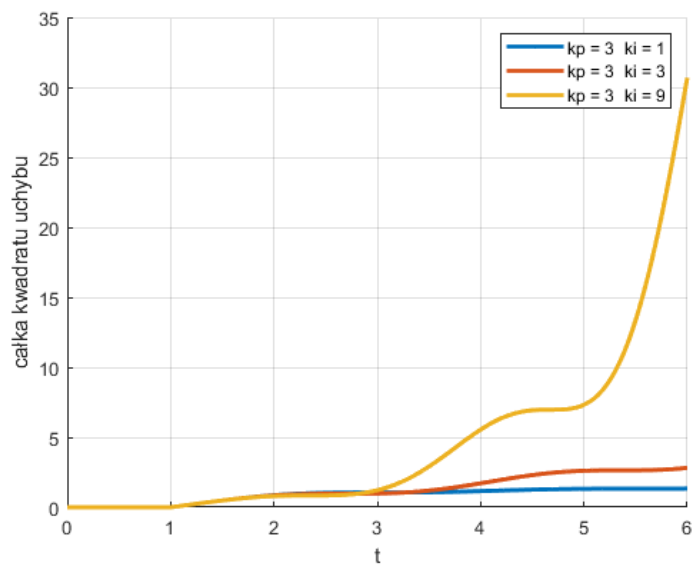
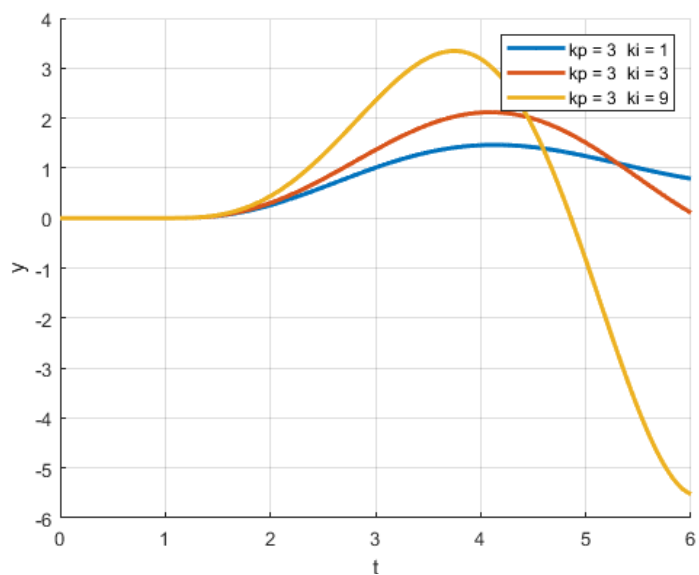
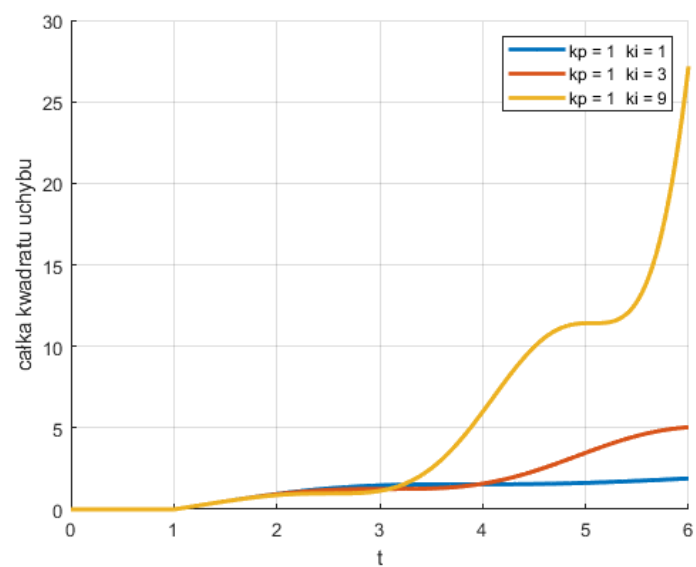
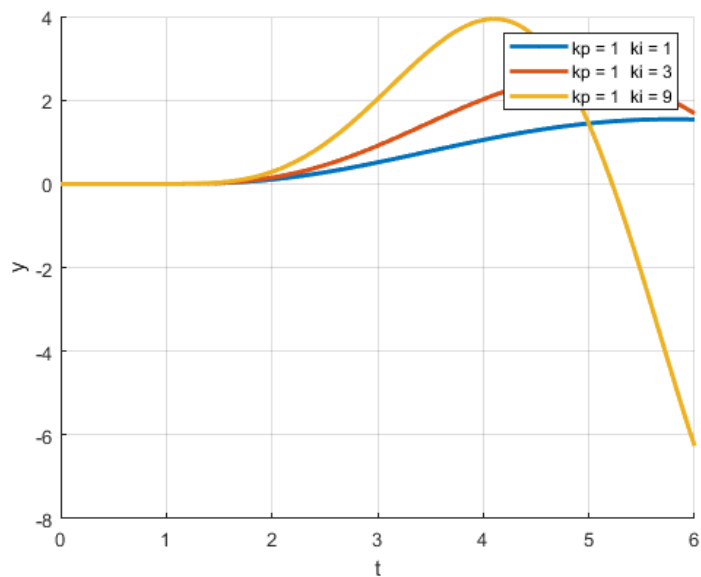
2. Badanie odpowiedzi układu automatycznej regulacji na podanie skoku jednostkowego, dla różnych parametrów regulatora PI

2.1. Schemat badanego układu



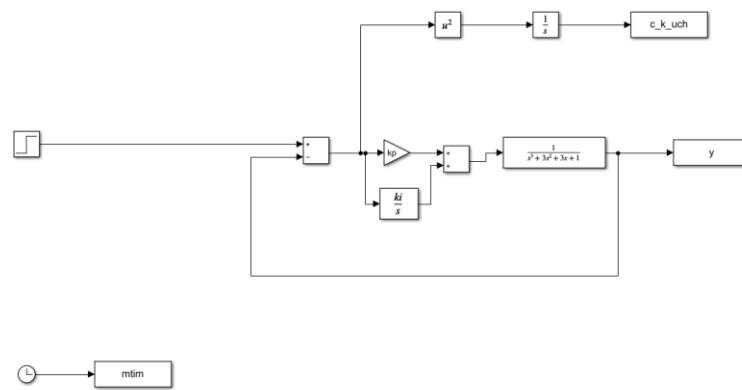
4. Schemat badanego układu

2.2. Odpowiedź układu



3. Szukanie najlepszych parametrów regulatora PI

3.1. Schemat układu



3.2. Program do znajdowania najlepszych parametrów

```
clear all;
close all;

Ttt = 100;

step_ki = 0.25;
step_kp = step_ki;

min_kp = 0.1;
min_ki = 0.0;

max_kp = 8;
max_ki = 5;

t_kp = min_kp:step_kp:max_kp;
t_ki = min_ki:step_ki:max_ki;

m_uchyb = zeros(length(t_kp),length(t_ki));
%m_uchyb = zeros((max_ki/step_ki),(max_kp/step_kp));

best_c_k_uch = 99999999;
best_kp = 0;
best_ki = 0;

for i=1:length(t_kp)
    for j=1:length(t_ki)
        kp = t_kp(i);
        ki = t_ki(j);

        sim('simu3_31.slx');

        %warunek stabilności obliczony z kryterium Hurwitza
        if (kp < 8)
            if (ki < ((-kp*kp+7*kp+8)/9))
                m_uchyb(i,j) = max(c_k_uch);
                if m_uchyb(i,j) > 100 %uciecie duzych uchybow
                    m_uchyb(i,j) = 100;
                end
                if m_uchyb(i,j) < best_c_k_uch %sprawdzenie czy calka z kwadratu uchybu jest
mniejsza od ostatniej najlepszej
                    if 1.1 > max(y) %sprawdzenie czy maksymalne przeregulowanie jest mniejsze od 110%
wartosci ustalonej
                        best_c_k_uch = m_uchyb(i,j);
                        best_kp = kp;
                        best_ki = ki;
                        best_copy_y = y;
                    end
                end
            else
                m_uchyb(i,j) = -1;
            end
        else
            m_uchyb(i,j) = -1;
        end
    end

    fprintf('%i z %i\n', i, length(t_kp));
end
surf(t_ki,t_kp,m_uchyb);

%[x,y] = meshgrid(t_ki,t_kp);
%mesh(x,y,m_uchyb);

xlabel('ki'), ylabel('kp'), zlabel('max calka kwadr. uchybu'); %to ma byc na odwrot!
```

3.3. Przeprowadzenie procesu poszukiwania najlepszych parametrów

Przyjęto optymalizację ze względu na minimalizację całki kwadratu uchybu, przy jednoczesnym ograniczeniu przeregulowania do 110% wartości ustalonej

Parametry dla których układ jest niestabilny wyliczono za pomocą metody Hurwitza.

$$K_{UAR} = \frac{k_i + sk_p}{s^4 + 3s^3 + 3s^2 + (k_p + 1)s + k_i}$$

$$H_4 = \begin{bmatrix} 3 & (k_p + 1) & 0 & 0 \\ 1 & 3 & k_i & 0 \\ 0 & 3 & (k_p + 1) & 0 \\ 0 & 1 & 3 & k_i \end{bmatrix}$$

$$8 > k_p > 0$$

$$\frac{(-k_p^2 + 7k_p + 8)}{9} > k_i > 0$$

Całkę z kwadratu uchybu dla systemu niestabilnego przyrównano do -1.

Całkę z kwadratu uchybu o wartościach powyżej 100 przyrównano do 100.

W celu zaoszczędzenia ilości iteracji programu, pierwsze wywołanie przeprowadzono dla

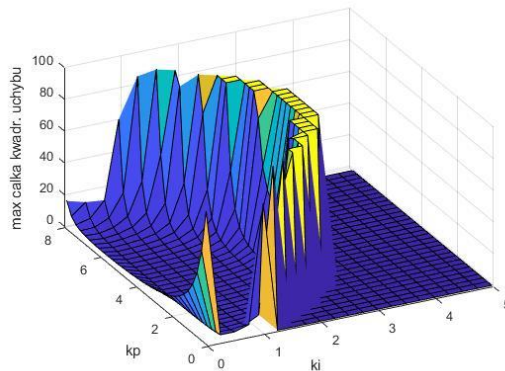
Kp w zakresie od 0.1 do 8, z krokiem 0.25

Ki w zakresie od 0.0 do 5, z krokiem 0.25

Uzyskano w ten sposób przybliżone wartości najlepszych parametrów:

Kp = 2.35

Ki = 0.25



Następnie wykonano symulację na zakresie

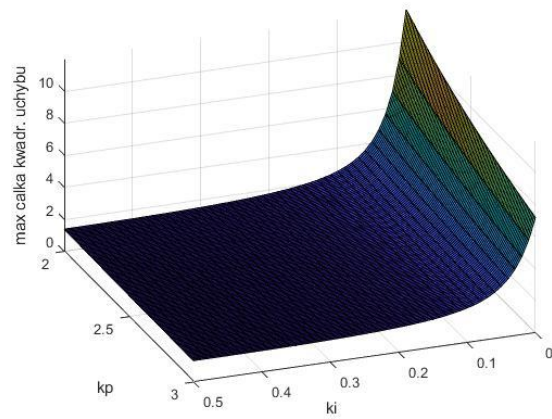
Kp w zakresie od 2 do 3, z krokiem 0.01

Ki w zakresie od 0.0 do 0.5, z krokiem 0.01

Uzyskano w ten sposób przybliżone wartości najlepszych parametrów:

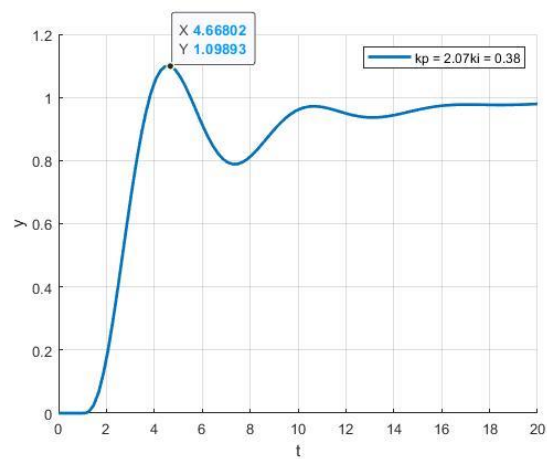
Kp = 2.07

Ki = 0.38



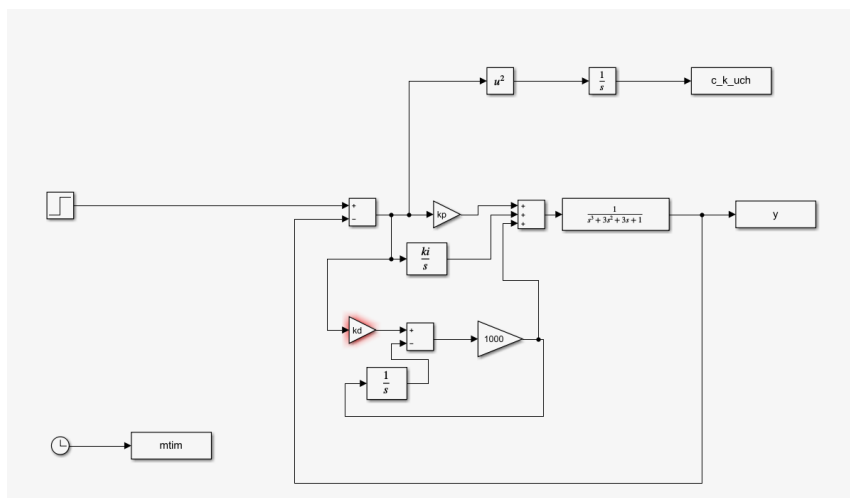
Wartość całki kwadratu uchybu dla czasu 100 wynosi 1.4165

Odpowiedź układu z zaproponowanym regulatorem:



4. Szukanie najlepszych parametrów regulatora PID

4.1. Schemat układu



4.2. Program do znajdowania najlepszych parametrów

```
5. clear all;
6. close all;
7.
8. Ttt = 100;
9.
10. step_ki = 0.5;
11. step_kp = step_ki;
12. step_kd = step_ki;
13.
14. min_kp = 0.1;
15. min_ki = 0.0;
16. min_kd = 0.0;
17.
18. max_kp = 8;
19. max_ki = 5;
20. max_kd = 5;
21.
22. t_kp = min_kp:step_kp:max_kp;
23. t_ki = min_ki:step_ki:max_ki;
24. t_kd = min_kd:step_kd:max_kd;
25.
26. m_uchyb = zeros(length(t_kp),length(t_ki));
27. %m_uchyb = zeros((max_ki/step_ki), (max_kp/step_kp));
28.
29. best_c_k_uch = 99999999;
30. best_kp = 0;
31. best_ki = 0;
32. best_kd = 0;
33.
34. for i=1:length(t_kp)
35.     for j=1:length(t_ki)
36.         for k=1:length(t_kd)
37.             kp = t_kp(i);
38.             ki = t_ki(j);
39.             kd = t_kd(k);
40.
41.             sim('simu3_pid.slx');
42.
43.             %warunek stabilności obliczony z kryterium Hurwitza
44.             if (kd < 8)
45.                 if (ki < ((-kd*kd+7*kd+8)/9))
46.                     m_uchyb(i,j) = max(c_k_uch);
47.                     if m_uchyb(i,j) > 100 %uciecie duzych uchybow
48.                         m_uchyb(i,j) = 100;
49.                     end
50.                     if m_uchyb(i,j) < best_c_k_uch %sprawdzenie czy calka z kwadratu uchybu jest mniejsza
od ostatniej najlepszej
51.                         if 1.1 > max(y) %sprawdzenie czy maksymalne przeregulowanie jest mniejsze od 110%
wartosci ustalonej
52.                             best_c_k_uch = m_uchyb(i,j);
53.                             best_kp = kp;
54.                             best_ki = ki;
55.                             best_kd = kd;
56.                         end
57.                     end
58.                 else
59.                     m_uchyb(i,j) = -1;
60.                 end
61.             else
62.                 m_uchyb(i,j) = -1;
63.             end
64.         end
65.     end
66.     fprintf('%i z %i\n', i, length(t_kp));
67. end
```


4.3. Przeprowadzenie procesu poszukiwania najlepszych parametrów

Przyjęto optymalizację ze względu na minimalizację całki kwadratu uchybu, przy jednoczesnym ograniczeniu przeregulowania do 110% wartości ustalonej

Parametry dla których układ jest niestabilny wyliczono za pomocą metody Hurwitza.

$$K_{UAR} = \frac{k_i + sk_p + s^2 k_d}{s^5 + 3s^4 + 3s^3 + (k_d + 1)s^2 + (k_p)s + k_i}$$

$$H_5 = \begin{bmatrix} 3 & (k_d + 1) & k_i & 0 & 0 \\ 1 & 3 & k_p & 0 & 0 \\ 0 & 3 & (k_d + 1) & k_i & 0 \\ 0 & 1 & 3 & k_p & 0 \\ 0 & 0 & 3 & (k_d + 1) & k_i \end{bmatrix}$$

$$8 > k_d > 0$$

$$\frac{(-k_d^2 + 7k_d + 8)}{9} > k_i > 0$$

$$k_p > 0$$

Całkę z kwadratu uchybu dla systemu niestabilnego przyrównano do -1.

Całkę z kwadratu uchybu o wartościach powyżej 100 przyrównano do 100.

W celu zaoszczędzenia ilości iteracji programu, pierwsze wywołanie przeprowadzono dla

Kp w zakresie od 0.1 do 8, z krokiem 0.5

Ki w zakresie od 0.0 do 5, z krokiem 0.5

Kd w zakresie od 0.0 do 5, z krokiem 0.5

Uzyskano w ten sposób przybliżone wartości najlepszych parametrów:

$$K_p = 2.1$$

$$K_i = 0.5$$

$$K_d = 0.5$$

Minimalna wartość całki uchybu wynosiła 1.1509

Następnie wykonano symulację na zakresie

Kp w zakresie od 2 do 2.5, z krokiem 0.1

Ki w zakresie od 0.2 do 1, z krokiem 0.1

Kd w zakresie od 0.2 do 1, z krokiem 0.1

Uzyskano w ten sposób przybliżone wartości najlepszych parametrów:

$$K_p = 2.3$$

$$K_i = 0.6$$

$$K_d = 0.6$$

Minimalna wartość całki uchybu wynosiła 1.07

Następnie wykonano symulację na zakresie

K_p w zakresie od 2.2 do 2.5, z krokiem 0.01

K_i w zakresie od 0.59 do 0.7, z krokiem 0.01

K_d w zakresie od 0.59 do 0.7, z krokiem 0.01

Uzyskano w ten sposób przybliżone wartości najlepszych parametrów:

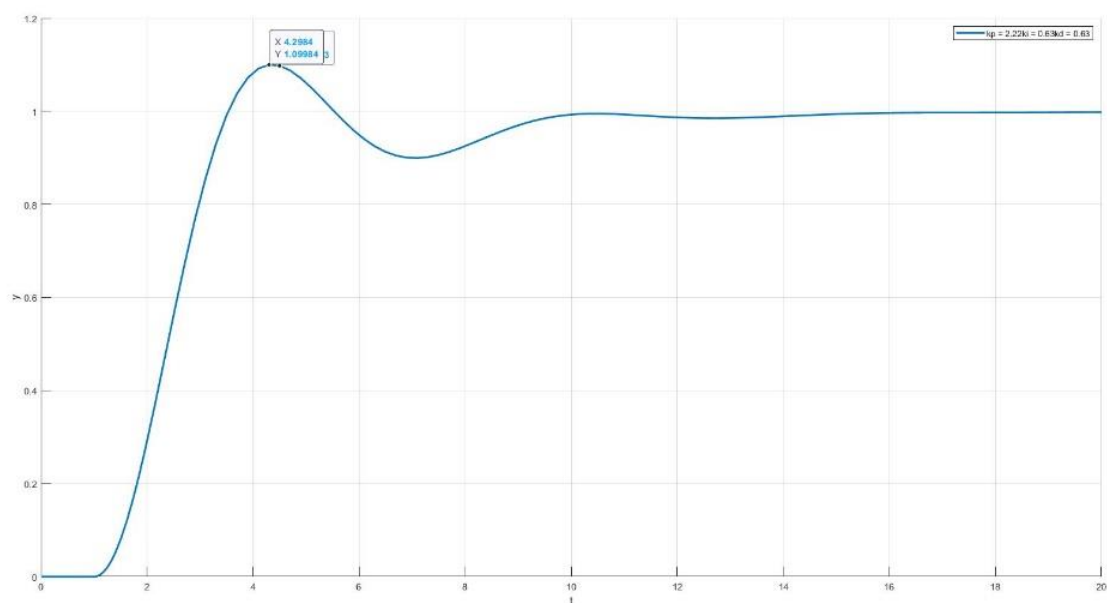
$K_p = 2.22$

$K_i = 0.63$

$K_d = 0.63$

Wartość całki kwadratu uchybu dla czasu 100 wynosi 1.068

Odpowiedź układu z zaproponowanym regulatorem:



5. Wnioski

- Zastosowanie regulatora PID znacząco zmniejsza całkę kwadratu uchybu
- Dzięki zawężeniu szukanego obszaru znacząco przyspieszono obliczenia, choć kosztem potencjalnych przeoczeń lepszych rozwiązań
- Warunek nieprzekroczenia 10% wartości ustalonej w znaczącym stopniu wpłynął na uzyskany wynik, powodując odrzucenie znacznej ilości rezultatów