

**Wydział Elektroniki i Technik Informacyjnych
Politechnika Warszawska**

**Projektowanie układów sterowania
(projekt grupowy)**

Sprawozdanie z projektu nr 4, zadanie nr 6

Mateusz Koroś, Ksawery Pasikowski, Mateusz Morusiewicz

Warszawa, 2017

Spis treści

1. Zad. 1	2
2. Zad. 2	3
3. Zad. 3	5
3.1. PID	5
3.2. DMC	6
4. Zad. 4	9
5. Zad. 5	13
6. Zad. 6	16
7. Zad. 7	19

1. Zad. 1

Poprawność punktu pracy została udowodniona poprzez sprawdzenie, czy obiekt, będący w punkcie pracy, pozostanie w nim, jeśli wartości sterowania pozostanie taka sama. Zostało to wykonane za pomocą komendy:

```
y_ust = symulacja_obiektu6y(0, 0, 0, 0)
```

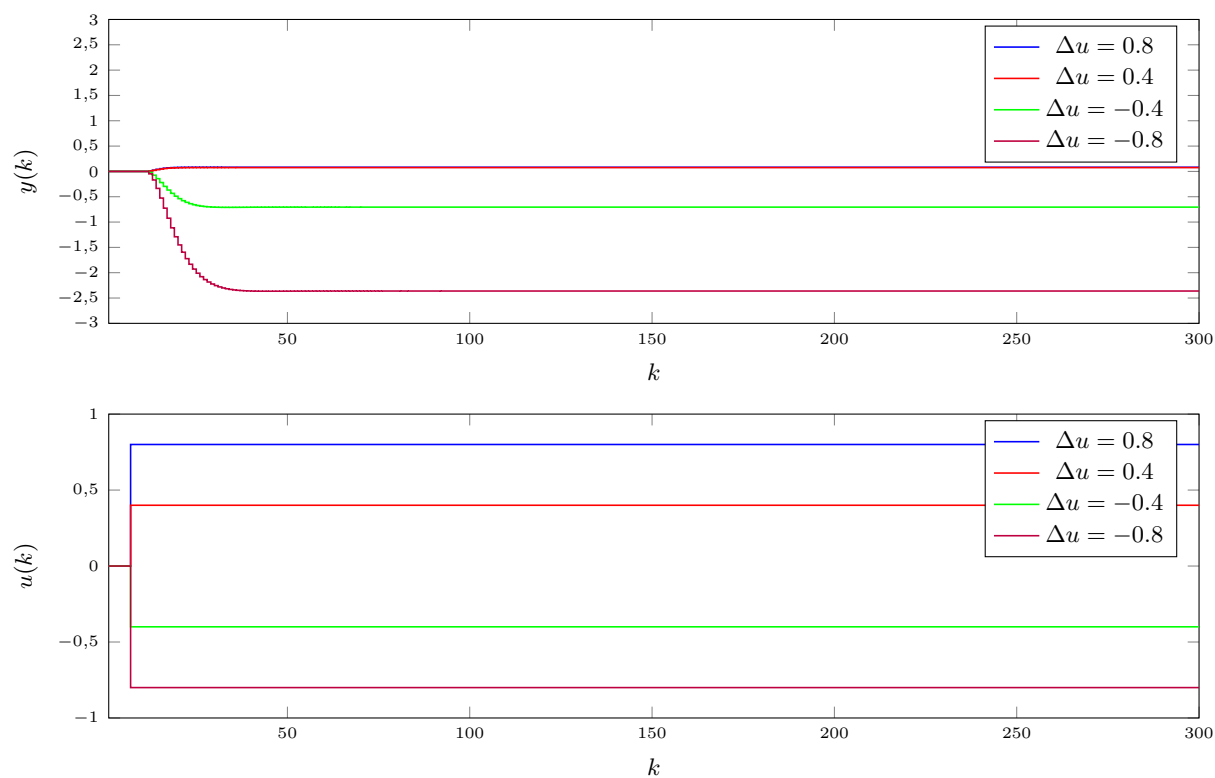
Co dało wynik $[0,0]$, co dowodzi, że punktem pracy rzeczywiście jest punkt $u = y = 0$.

2. Zad. 2

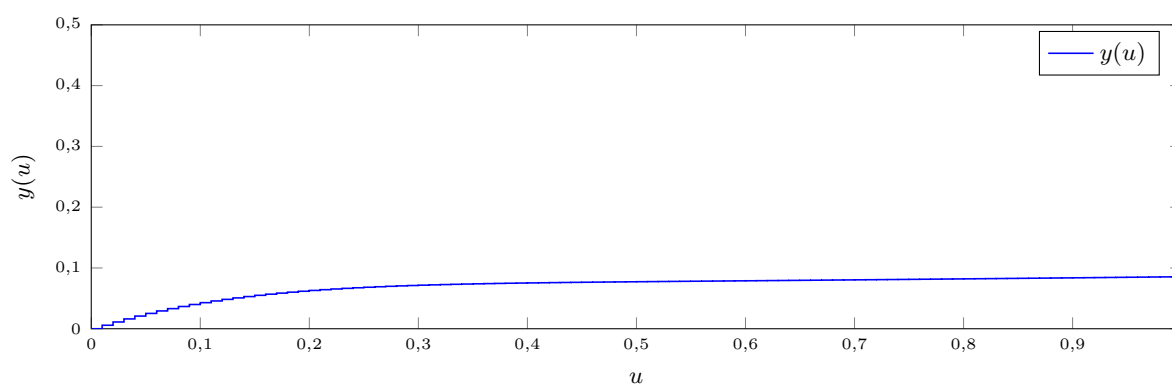
Sterowanie zostało wzbudzone do wartości:

- 0,4
- 0,8
- -0,4
- -0,8

Odpowiedzi skokowe widać na wykresie 2.1. Z charakterystyki 2.2 widać, że właściwości statyczne procesu nie są liniowe, dynamiczne również.



Rys. 2.1. Odpowiedzi skokowe



Rys. 2.2. Charakterystyka statyczna

3. Zad. 3

3.1. PID

```
function [ y, u, E, yzad ] = policzPID( Kp_, Ti_, Td_, Kk_)
U_min = -1;
U_max = 1;

Kp = Kp_;
Ti = Ti_;
Td = Td_;

Kk = Kk_;
Tp = 0.5;

r2 = (Kp * Td) / Tp ;
r1 = Kp * ( (Tp/(2*Ti)) - 2*(Td/Tp) - 1 ) ;
r0 = Kp * ( 1 + Tp/(2*Ti) + Td/Tp ) ;

%warunki poczatkowe
u(1:11) = 0 ;
y(1:11) = 0 ;
e(1:11) = 0 ;
index = 1;
yzads = [-1 -2.5 -1 0.06];
yzad = yzads(index);
yzadVec(1:Kk) = yzad;
% u(1:11) = 0.34 ;
% y(1:11) = 0.073 ;
% e(1:11) = 0 ;
% index = 1;
% yzads = [0.084];
% yzad = yzads(index);
% yzadVec(1:Kk) = yzad;

% glowna petla symulacji
for k = 7 : Kk
if mod(k,200) == 0
index = index + 1;
if index > length(yzads)
index = length(yzads);
end
yzad = yzads(index);
end
```

```

yzadVec(k) = yzad;

y(k) = symulacja_obiektu6y(u(k-5), u(k-6), y(k-1), y(k-2));

e(k) = yzad - y(k) ;

u(k) = r2 * e(k-2) + r1 * e(k-1) + r0 * e(k) + u(k-1) ;

if u(k) > U_max
u(k) = U_max;
elseif u(k) < U_min
u(k) = U_min;
end

end

E = (yzadVec - y) * (yzadVec - y)';

yzad = zeros(1, Kk);
yzad(1, :) = yzadVec;

end

```

3.2. DMC

```

function [ Y, U, E, yzadVec ] = policzDMC( D_, N_, Nu_, lambda_, Kk_)

D=D_;
N=N_;
Nu=Nu_;
lambda=lambda_;

% testowanie i dobieranie parametrów z zadania 6 i 7
s = ...
    load('wykresy_pliki/zad6/odpowiedzi/wyjście_skok_-1_-0.75.txt');
s = s(:, 2);

Upp=0;
Ypp=0;
Umin=-1;
Umax=1;

% testowanie i dobieranie parametrów z zadania 6 i 7
index = 1;
yzads = [-2.264];
yzad = yzads(index);
yzadVec(1:Kk_) = yzad;
Yzad = yzadVec - Ypp;

```

```

% index = 1;
% yzads = [-1 -2.5 -1 0];
% yzad = yzads(index);
% yzadVec(1:Kk_) = yzad;
% Yzad = yzadVec - Ypp;
%inicjalizacja sta?ych
kk=Kk_;
M=zeros(N,Nu);
for i=1:N
for j=1:Nu
if (i>=j)
M(i,j)=s(i-j+1);
end;
end;
end;

MP=zeros(N,D-1);
for i=1:N
for j=1:D-1
if i+j<=D
MP(i,j)=s(i+j)-s(j);
else
MP(i,j)=s(D)-s(j);
end;
end;
end;

% Obliczanie parametr?w regulatora

I=eye(Nu);
K=((M'*M+lambda*I)^-1)*M';
ku=K(1,:)*MP;
ke=sum(K(1,:));

% U(1:kk)=Upp;
% Y(1:kk)=Ypp;

% testowanie i dobieranie parametr?w z zadania 6 i 7
U(1:kk)=-0.96;
Y(1:kk)=-3.187;

e=zeros(1,kk);

u=U-Upp;
y=Y-Ypp;
umax = Umax - Upp;
umin = Umin - Upp;

deltaup=zeros(1,D-1);

```



```
for k=7:kk
if mod(k,200) == 0
index = index + 1;
if index > length(yzads)
index = length(yzads);
end
yzad = yzads(index);
end
yzadVec(k) = yzad;
Yzad(k) = yzadVec(k) - Ypp;

%symulacja obiektu
Y(k)= symulacja_obiektu6y(U(k-5), U(k-6), Y(k-1), Y(k-2));
y(k) = Y(k) - Ypp;
%uchyb regulacji
e(k)=Yzad(k) - y(k);

% Prawo regulacji
deltauk=ke*e(k)-ku*deltaup';

for n=D-1:-1:2
deltaup(n)=deltaup(n-1);
end
deltaup(1)=deltauk;
u(k)=u(k-1)+deltaup(1);

if u(k)>umax
u(k)=umax;
elseif u(k)<umin
u(k)=umin;
end
U(k)=u(k)+Upp;

end

%obliczenie b??du
E=0;
for k=1:kk
E=E+((Yzad(k)-Y(k))^2);
end

end
```

4. Zad. 4

Regulatory oceniane były na podstawie wykresów oraz wartości wskaźnika jakości:

$$E = \sum_{k=1}^{k_{konc}} \sum_{m=1}^2 (y_m^{zad}(k) - y(k))^2$$

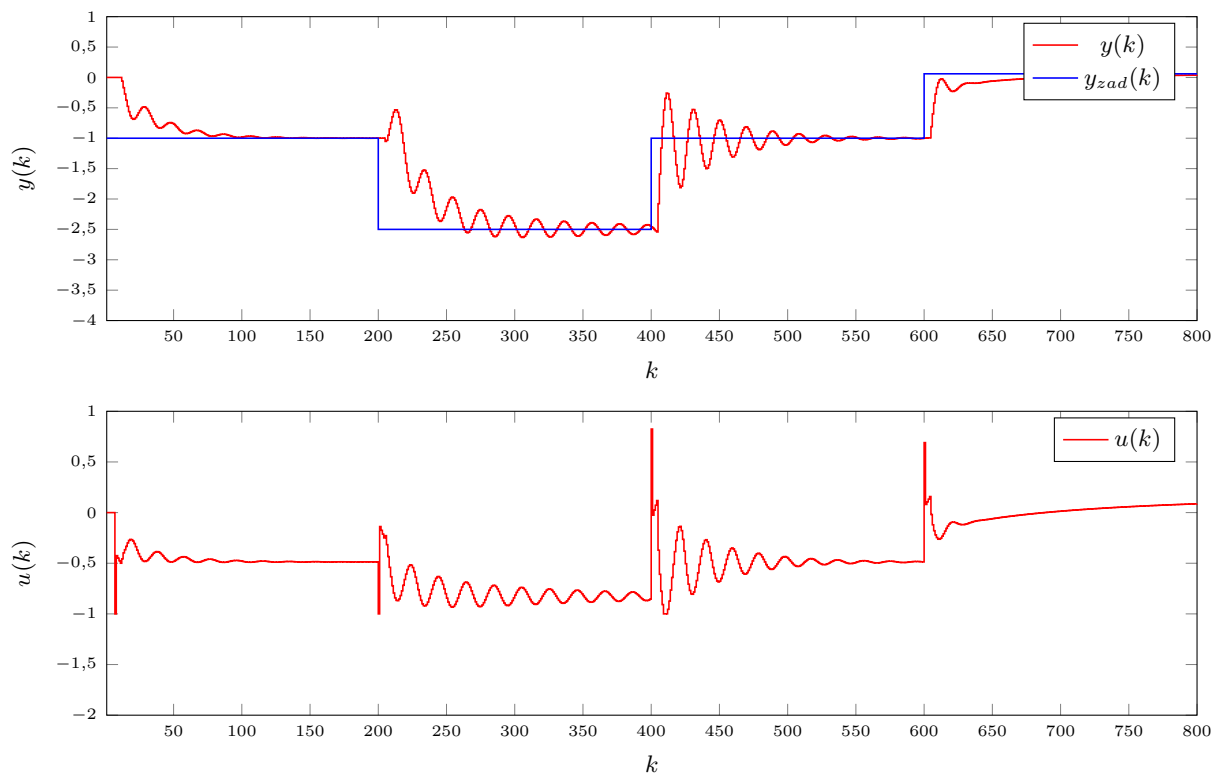
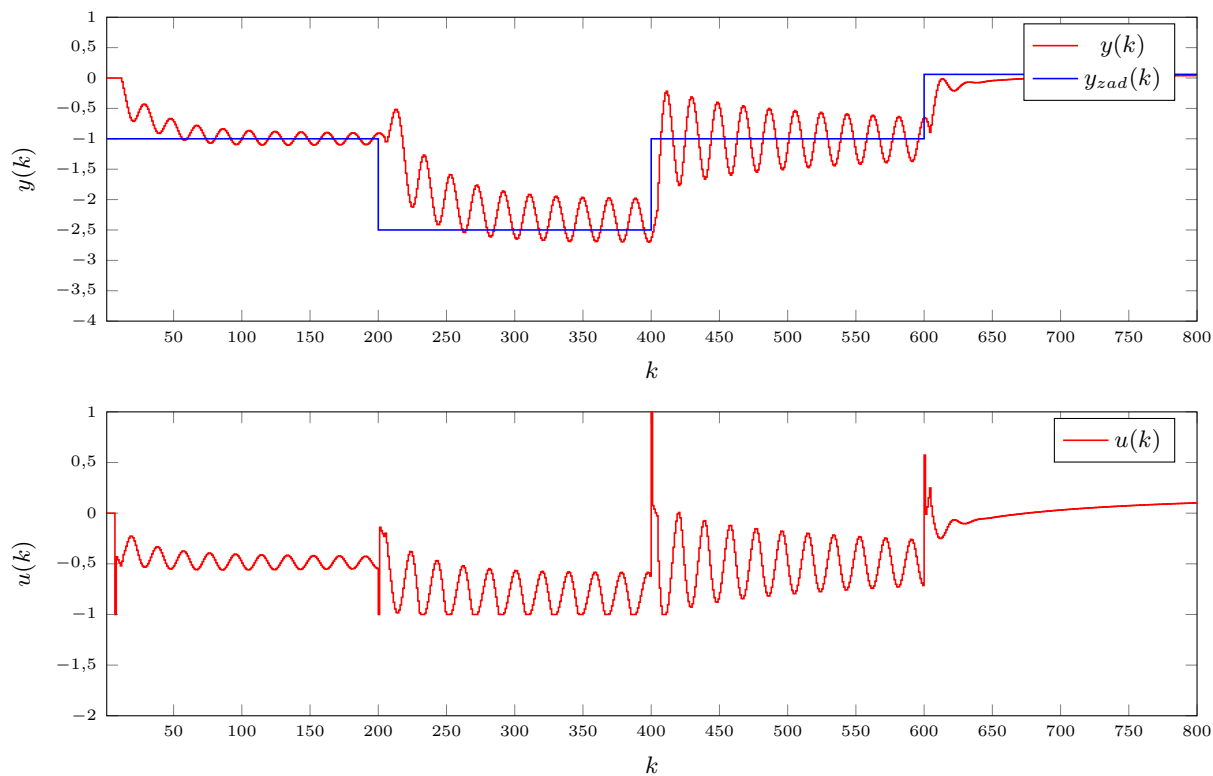
Parametry zaimplementowanych regulatorów PID oraz wartości błędów:

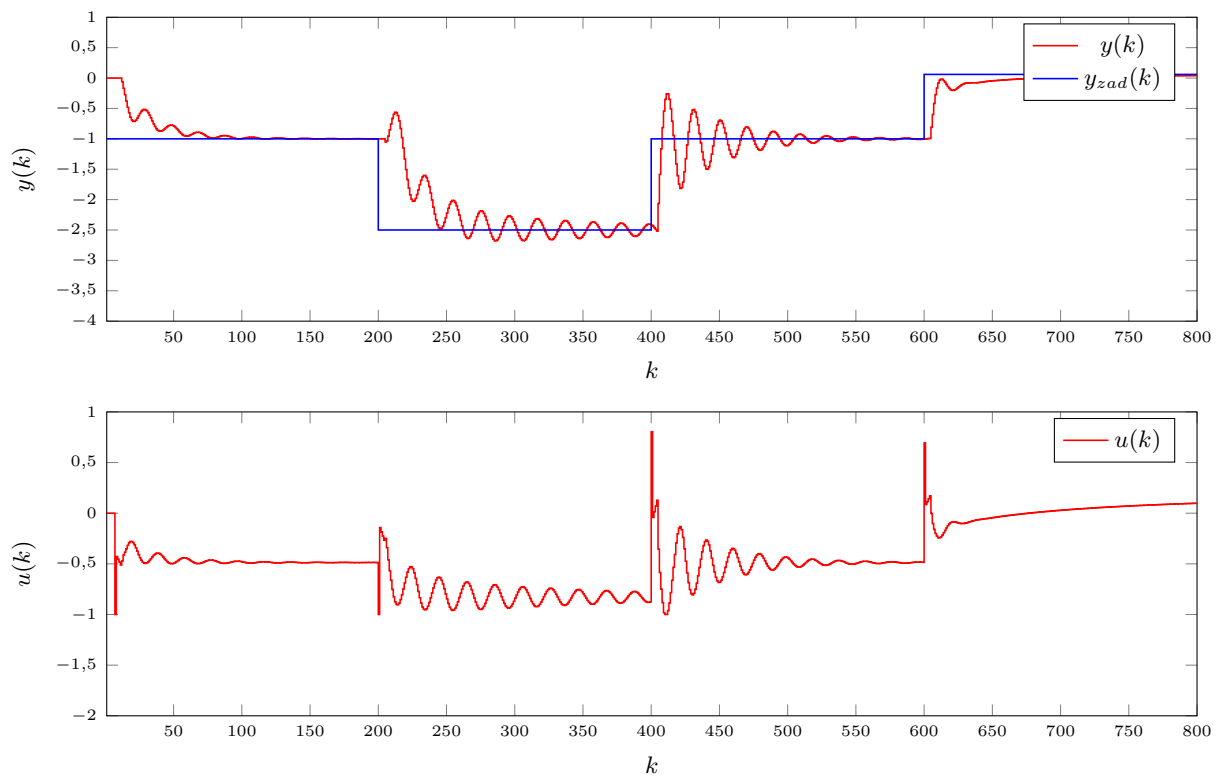
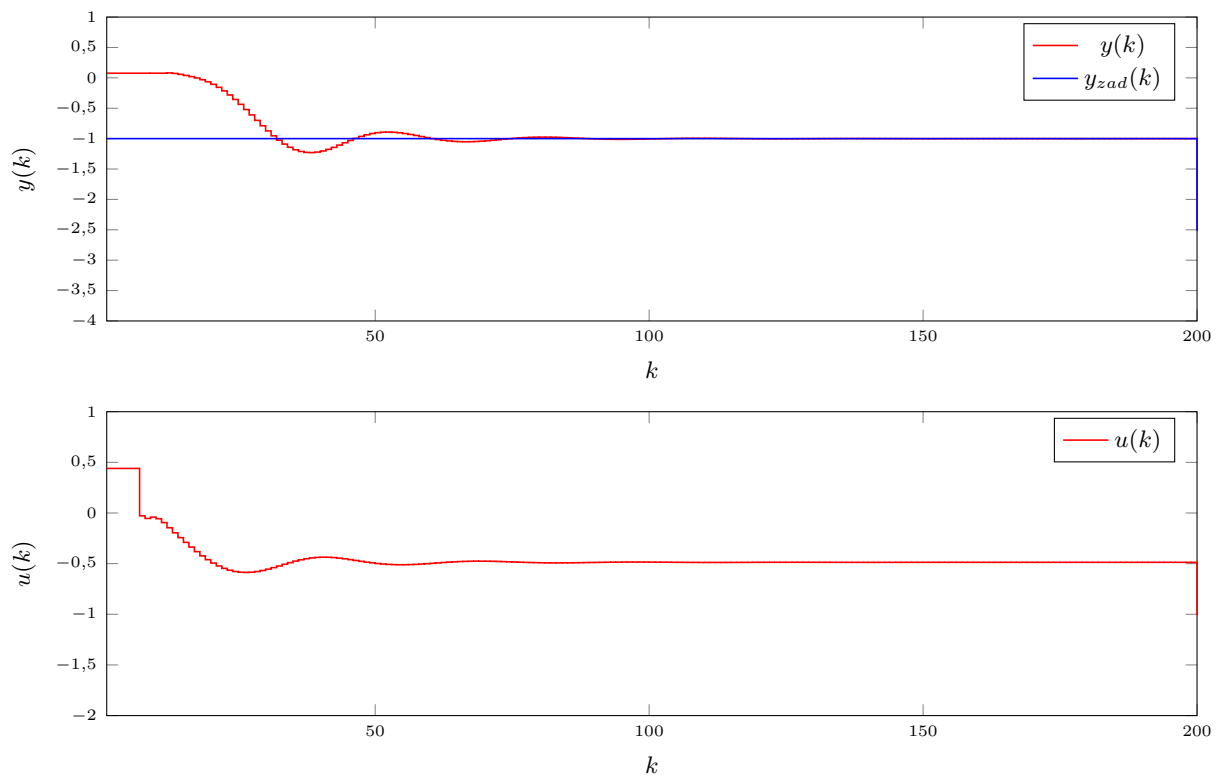
- $K_p = 0,6 \quad T_i = 10 \quad T_d = 0,5 \Rightarrow E = 1,548\,078 \cdot 10^2$ (wykres 4.1)
- $K_p = 0,5 \quad T_i = 10 \quad T_d = 0,6 \Rightarrow E = 1,218\,494 \cdot 10^2$ (wykres 4.2)
- $K_p = 0,5 \quad T_i = 9 \quad T_d = 0,6 \Rightarrow E = 1,155\,311 \cdot 10^2$ (wykres 4.3)

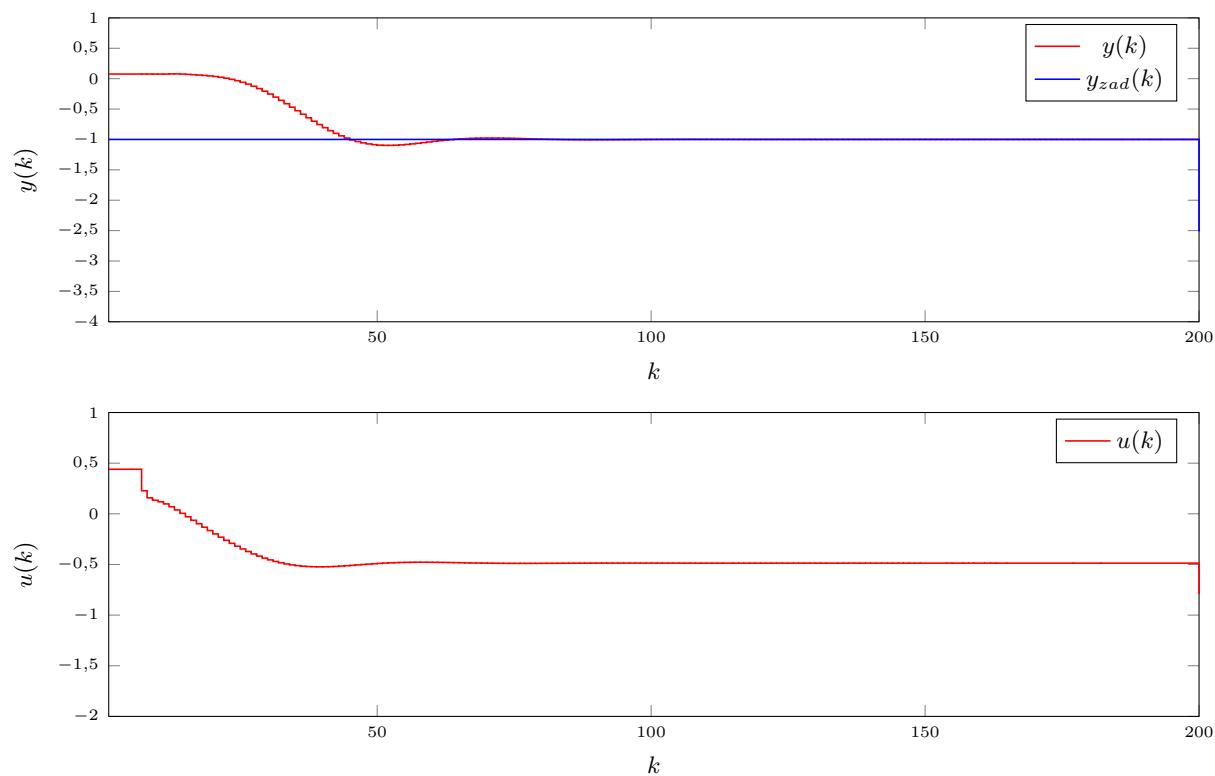
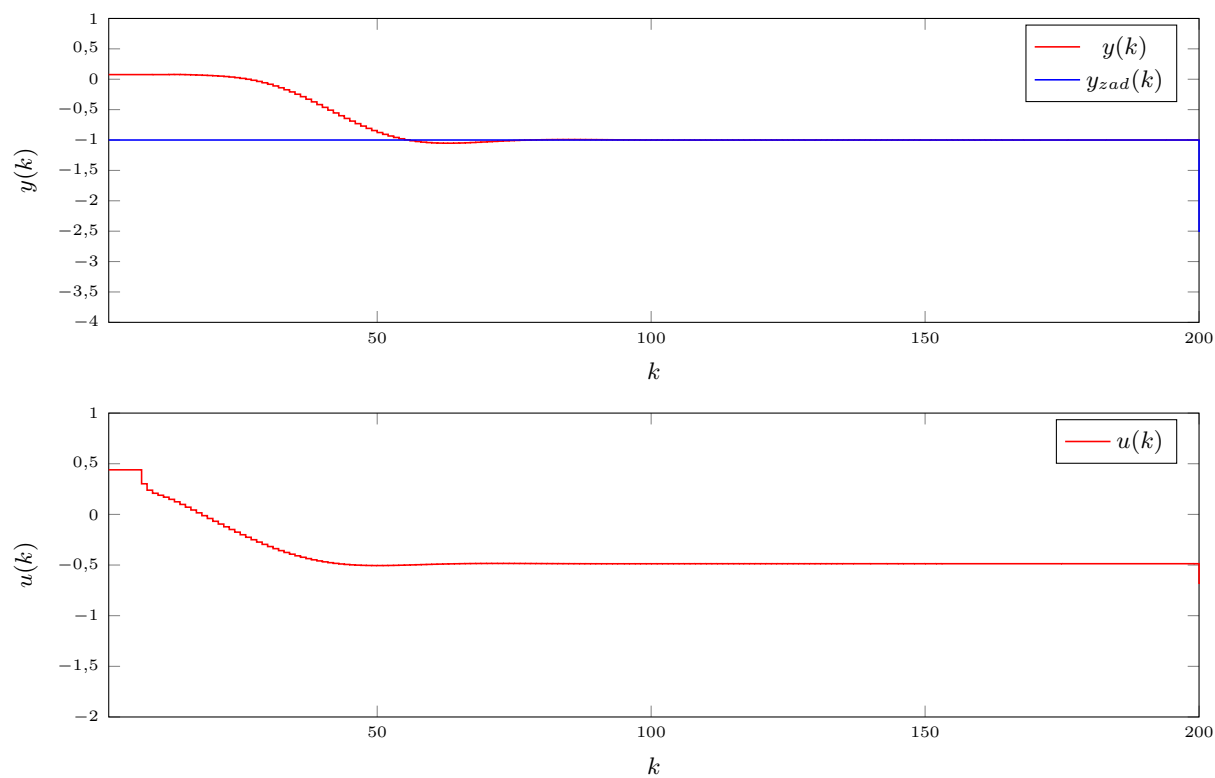
Parametry zaimplementowanych regulatorów DMC oraz wartości błędów:

- $\lambda = 1 \Rightarrow E = 7,523\,405 \cdot 10^1$ (wykres 4.4)
- $\lambda = 10 \Rightarrow E = 8,822\,105 \cdot 10^1$ (wykres 4.5)
- $\lambda = 30 \Rightarrow E = 9,933\,839 \cdot 10^1$ (wykres 4.6)

Wszystkie regulatory DMC bardzo dobrze poradziły sobie z osiągnięciem wartości zadanej, w szczególności ten o $\lambda = 30$, który, pomimo największego błędu, najszybciej dochodził do wartości zadanej i posiadał najmniejsze przeregulowanie. Tego samego nie można powiedzieć o regulatorach PID. Nawet najlepszy z nich (o parametrach $K_p = 0,6 \quad T_i = 10 \quad T_d = 0,5$) mimo osiągnięcia wartości zadanej wcześniej wpadał w znaczne oscylacje. Jest to spowodowane nieliniową charakterystyką obiektu i może być poprawione poprzez zastosowanie regulatora rozmytego.

Rys. 4.1. Regulator PID o parametrach: $K_p = 0,6$ $T_i = 10$ $T_d = 0,5$ Rys. 4.2. Regulator PID o parametrach: $K_p = 0,5$ $T_i = 10$ $T_d = 0,6$

Rys. 4.3. Regulator PID o parametrach: $K_p = 0,5$ $T_i = 9$ $T_d = 0,6$ Rys. 4.4. Regulator DMC o parametrze $\lambda = 1$

Rys. 4.5. Regulator DMC o parametrze $\lambda = 10$ Rys. 4.6. Regulator DMC o parametrze $\lambda = 30$

5. Zad. 5

Zaimplementowany został rozmyty regulator PID, składający się z 2, 3, 4 i 5 regulatorów. Parametry regulatorów rozmytych w zależności od parametrów składowych oraz błędy podczas symulacji są podane poniżej:

— 2 regulatory:

$$K_{p1} = 0,11 \quad T_{i1} = 3,5 \quad T_{d1} = 1,5$$

$$K_{p2} = 8 \quad T_{i2} = 1,5 \quad T_{d2} = 0,1$$

$$E = 7,465\,864 \cdot 10^1 \text{ (wykres 5.1)}$$

— 3 regulatory:

$$K_{p1} = 0,11 \quad T_{i1} = 3,5 \quad T_{d1} = 1,5$$

$$K_{p2} = 0,3 \quad T_{i2} = 7 \quad T_{d2} = 0,25$$

$$K_{p3} = 10 \quad T_{i3} = 1,5 \quad T_{d3} = 0,1$$

$$E = 1,045\,992 \cdot 10^2 \text{ (wykres 5.2)}$$

— 4 regulatory:

$$K_{p1} = 0,11 \quad T_{i1} = 3,5 \quad T_{d1} = 1,5$$

$$K_{p2} = 0,4 \quad T_{i2} = 3,2 \quad T_{d2} = 1$$

$$K_{p3} = 1,5 \quad T_{i3} = 3 \quad T_{d3} = 0,8$$

$$K_{p4} = 10 \quad T_{i4} = 1,5 \quad T_{d4} = 0,1$$

$$E = 1,028\,271 \cdot 10^2 \text{ (wykres 5.3)}$$

— 5 regulatorów:

$$K_{p1} = 0,11 \quad T_{i1} = 3,5 \quad T_{d1} = 1,5$$

$$K_{p2} = 0,4 \quad T_{i2} = 3,2 \quad T_{d2} = 1$$

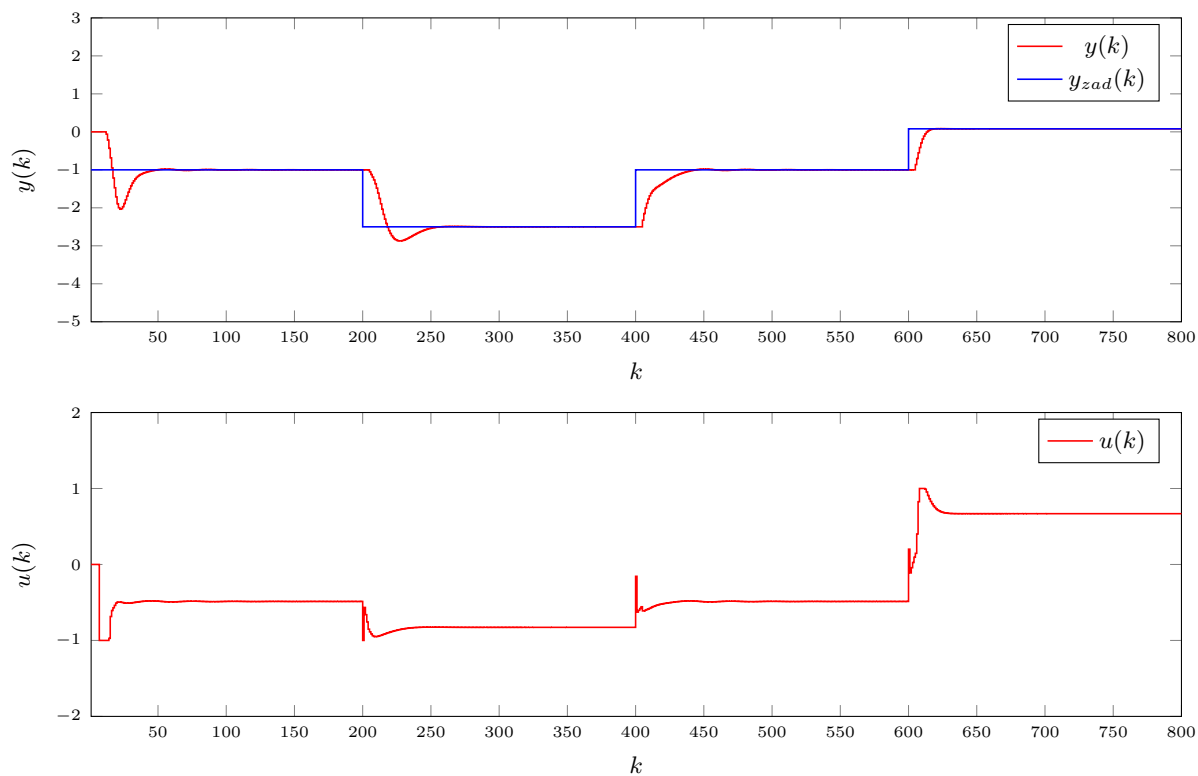
$$K_{p3} = 1,5 \quad T_{i3} = 3 \quad T_{d3} = 0,8$$

$$K_{p4} = 3 \quad T_{i4} = 2,5 \quad T_{d4} = 0,6$$

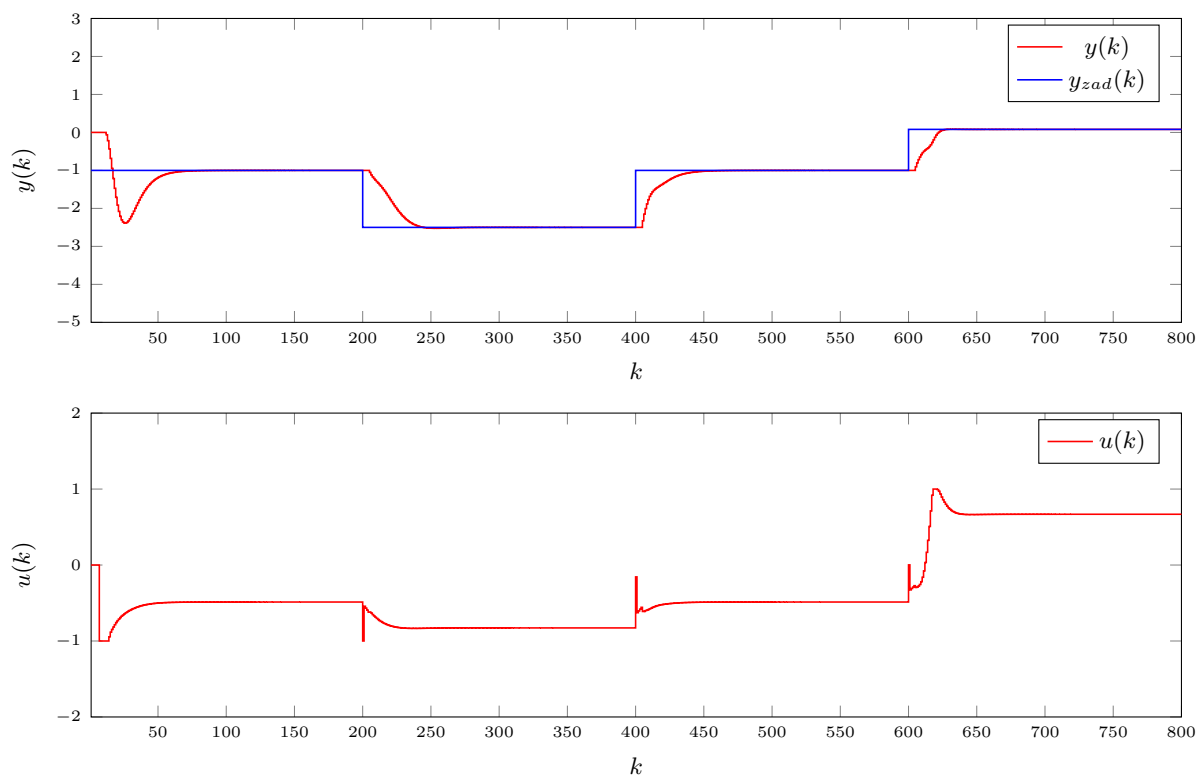
$$K_{p5} = 10 \quad T_{i5} = 1,5 \quad T_{d5} = 0,1$$

$$E = 9,825\,637 \cdot 10^1 \text{ (wykres 5.4)}$$

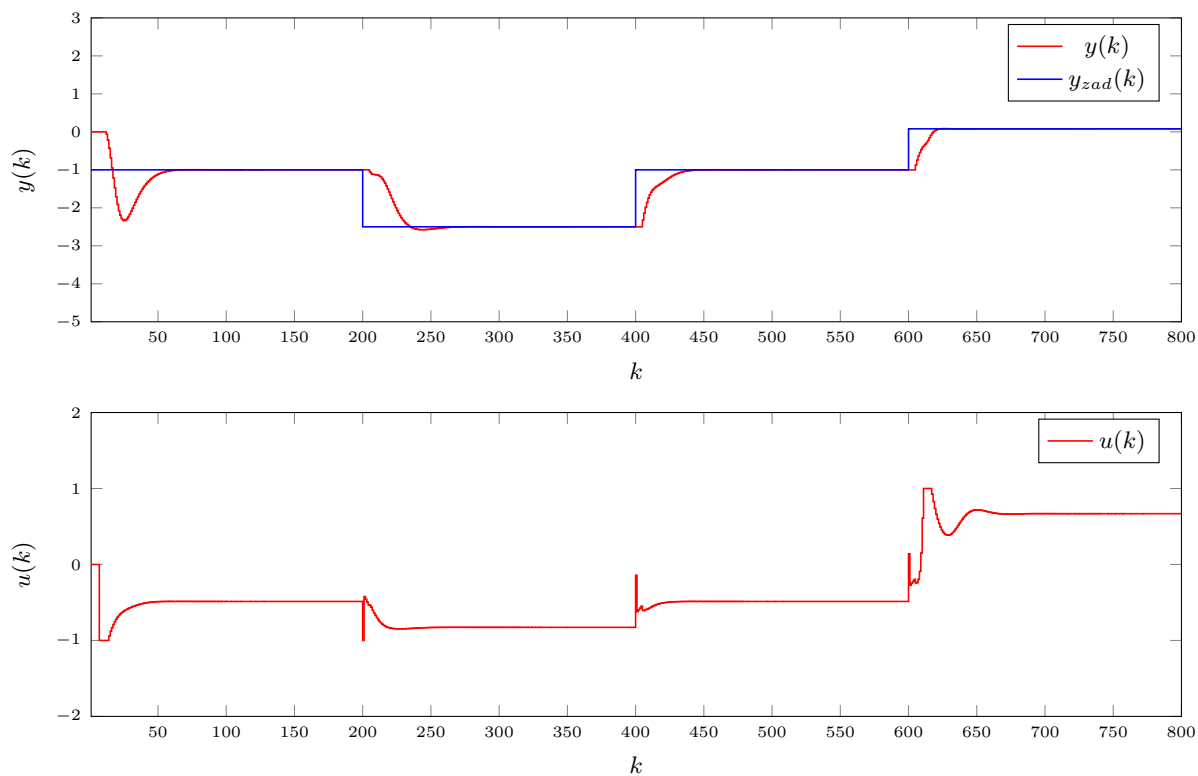
Rozmyty regulator PID okazał się dużo lepszy od tradycyjnego. Wszystkie regulatory osiągały wartość zadaną w bardzo krótkim czasie. Wynika z tego, że w obiektach nieliniowych dużo korzystniej jest zastosować rozmyty regulator PID zamiast tradycyjnego.



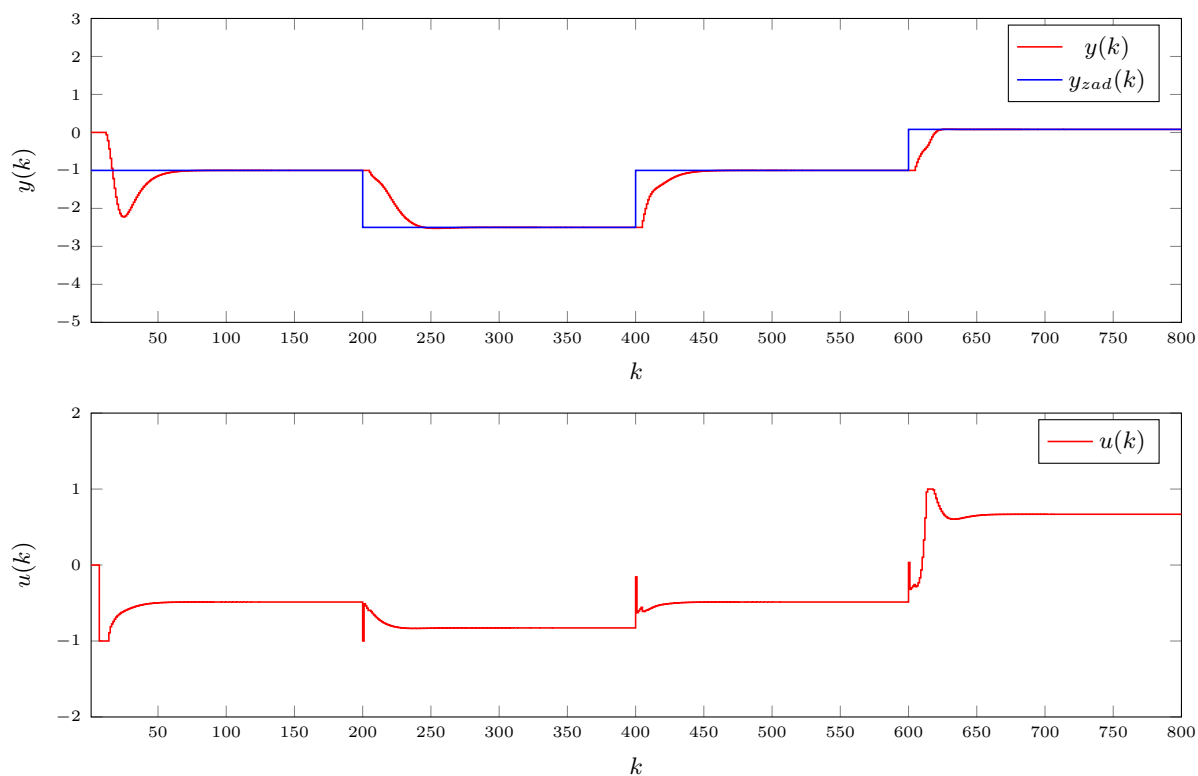
Rys. 5.1. Rozmyty regulator PID, składający się z dwóch regulatorów składowych



Rys. 5.2. Rozmyty regulator PID, składający się z trzech regulatorów składowych



Rys. 5.3. Rozmyty regulator PID, składający się z czterech regulatorów składowych

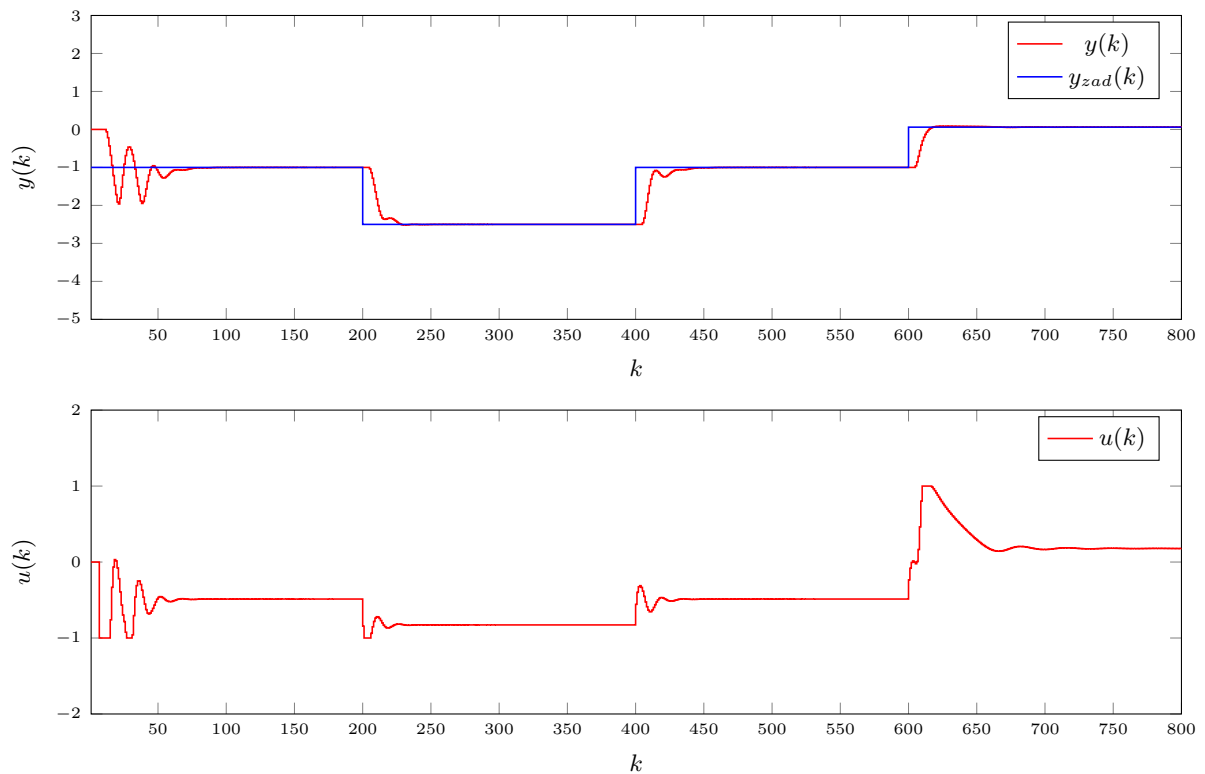


Rys. 5.4. Rozmyty regulator PID, składający się z pięciu regulatorów składowych

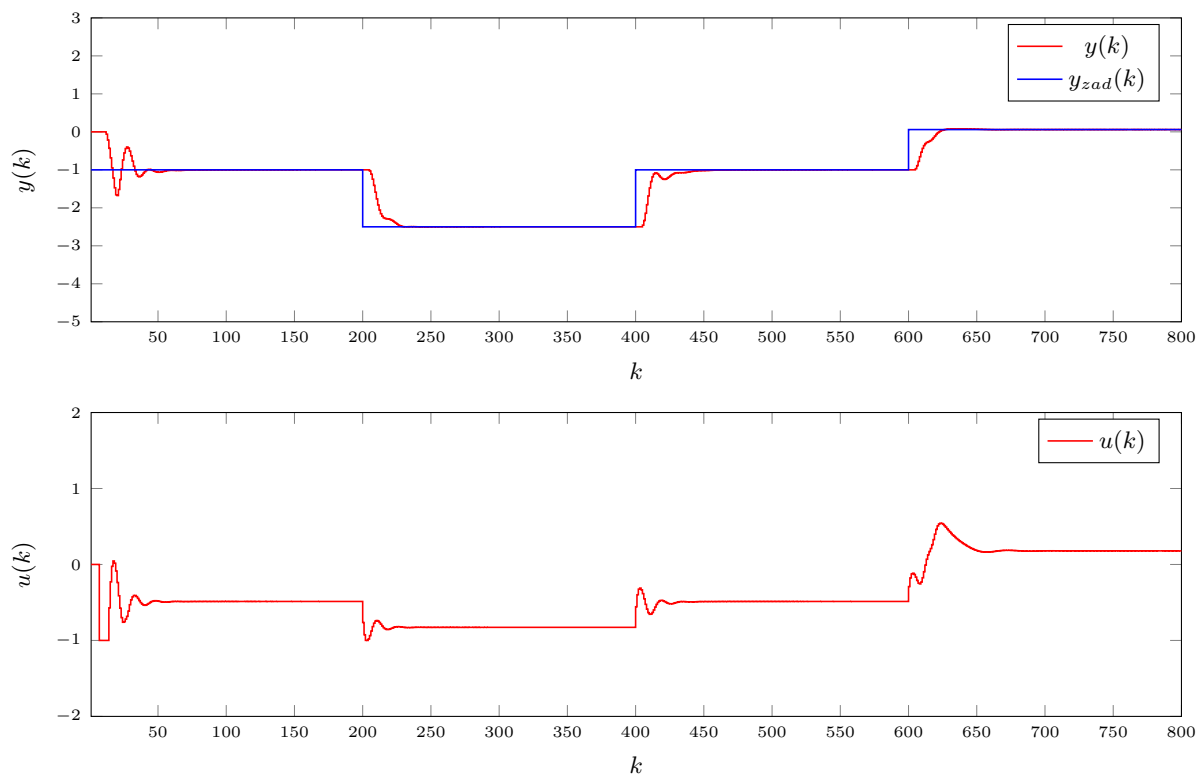
6. Zad. 6

Rozmyty regulator DMC, podobnie jak PID, składał się z 2, 3, 4 i 5 regulatorów. Parametry regulatorów rozmytych w zależności od parametrów składowych oraz błędy podczas symulacji są podane poniżej:

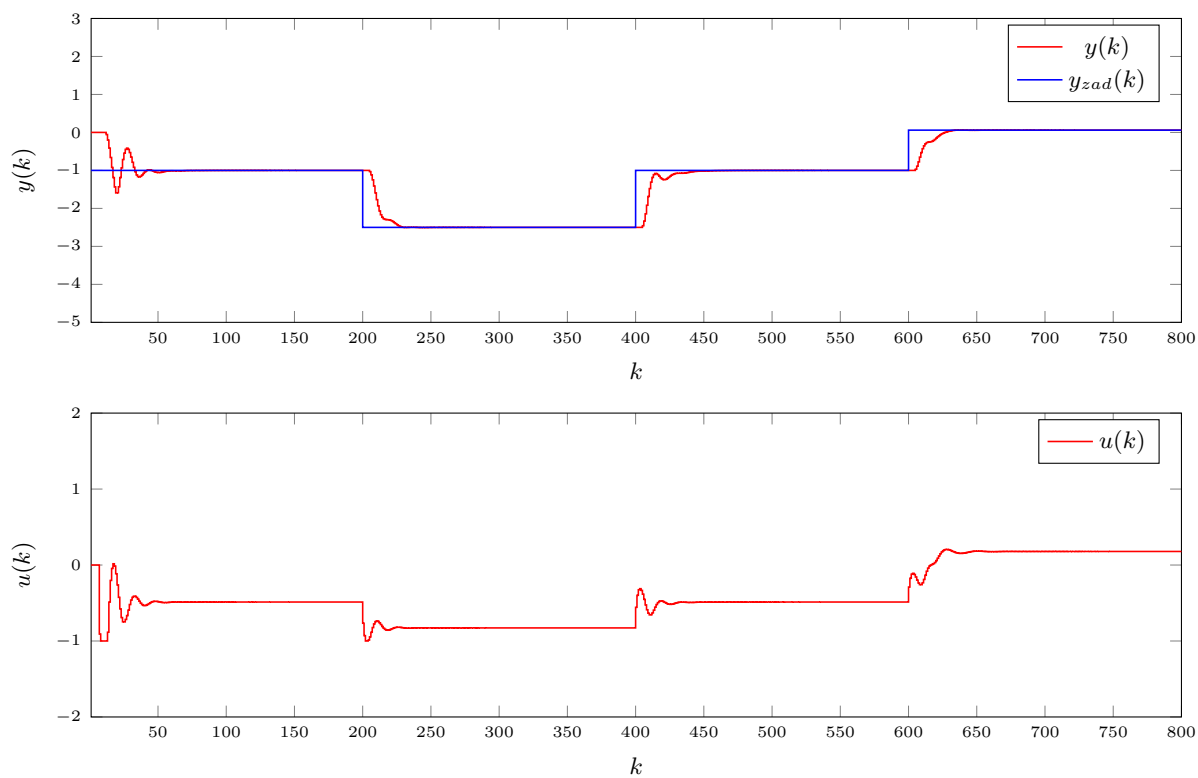
- 2 regulatory: $\Rightarrow \lambda_1 = 20 \ \lambda_2 = 0,7 \ E = 6,983\,574 \cdot 10^1$ (wykres 6.1)
- 3 regulatory: $\Rightarrow \lambda_1 = 20 \ \lambda_2 = 10 \ \lambda_3 = 0,7 \ E = 6,578\,143 \cdot 10^1$ (wykres 6.2)
- 4 regulatory: $\Rightarrow \lambda_1 = 20 \ \lambda_2 = 16 \ \lambda_3 = 9 \ \lambda_4 = 0,7 \ E = 6,545\,036 \cdot 10^1$ (wykres 6.3)
- 5 regulatorów: $\Rightarrow \lambda_1 = 20 \ \lambda_2 = 16 \ \lambda_3 = 10 \ \lambda_4 = 5 \ \lambda_5 = 0,7 \ E = 6,474\,870 \cdot 10^1$ (wykres 6.4)



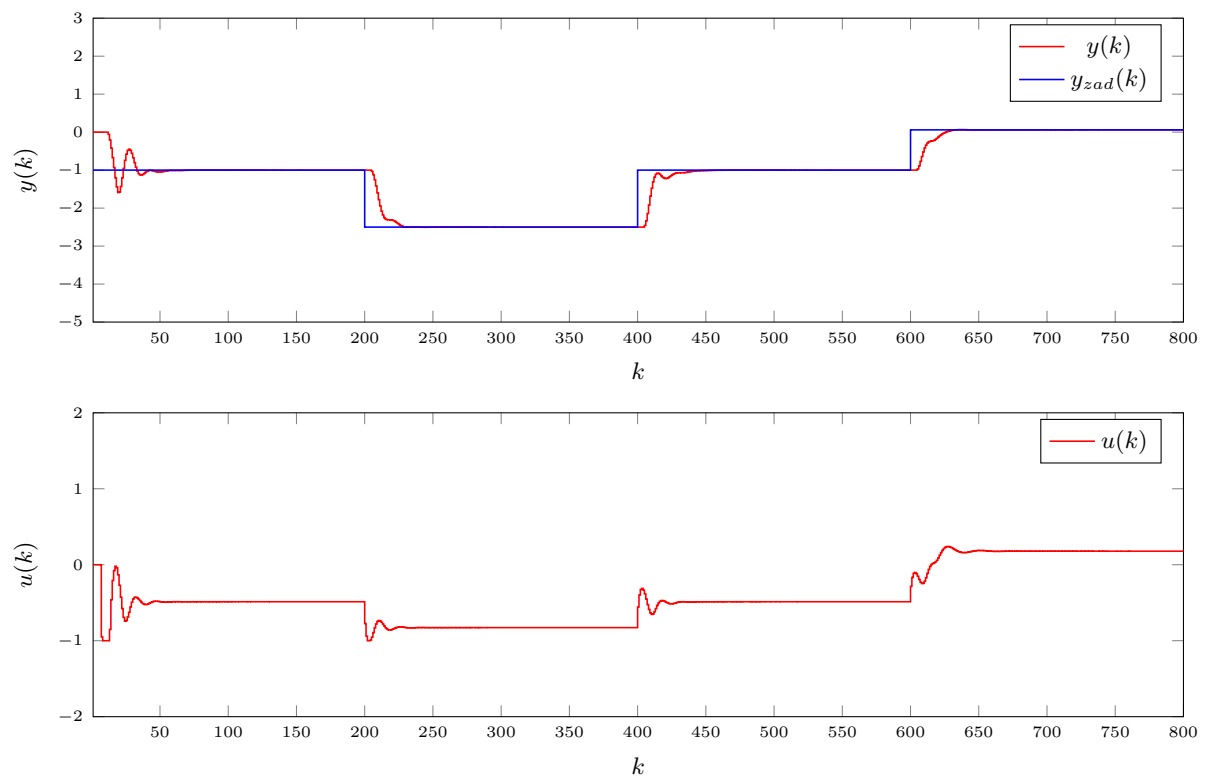
Rys. 6.1. Rozmyty regulator DMC, składający się z dwóch regulatorów składowych



Rys. 6.2. Rozmyty regulator DMC, składający się z trzech regulatorów składowych



Rys. 6.3. Rozmyty regulator DMC, składający się z czterech regulatorów składowych



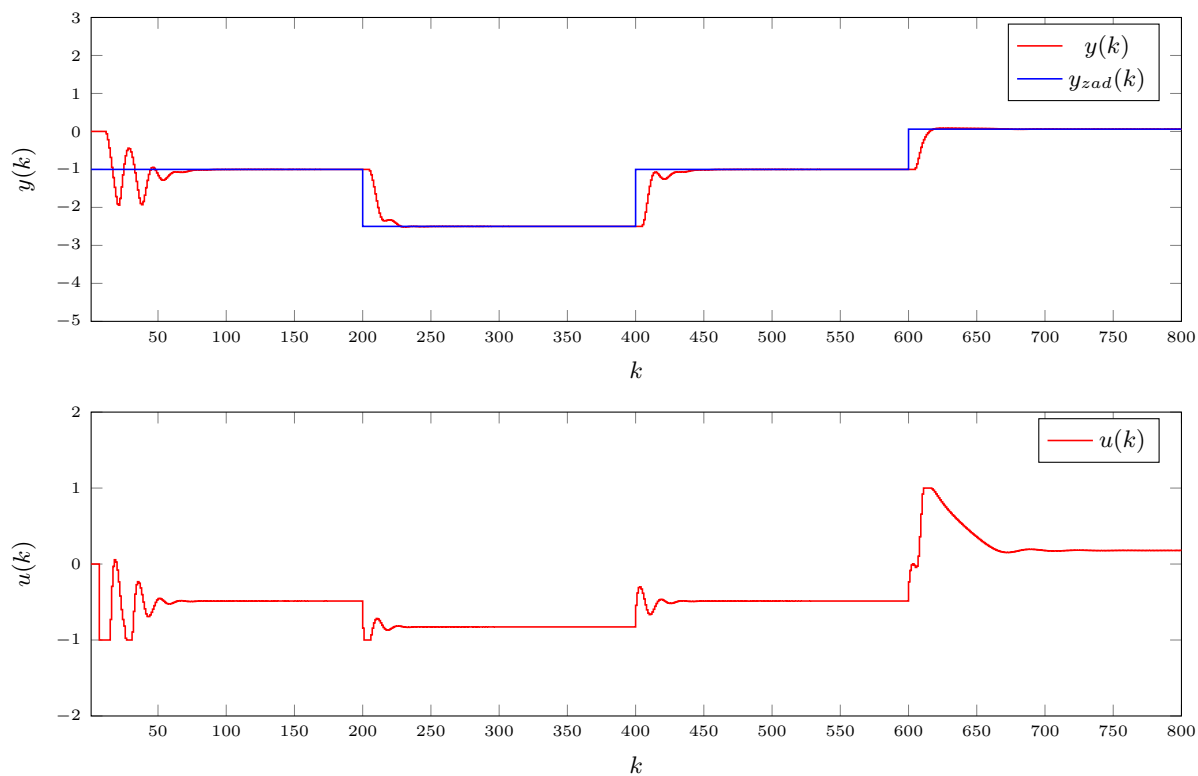
Rys. 6.4. Rozmyty regulator DMC, składający się z pięciu regulatorów składowych

7. Zad. 7

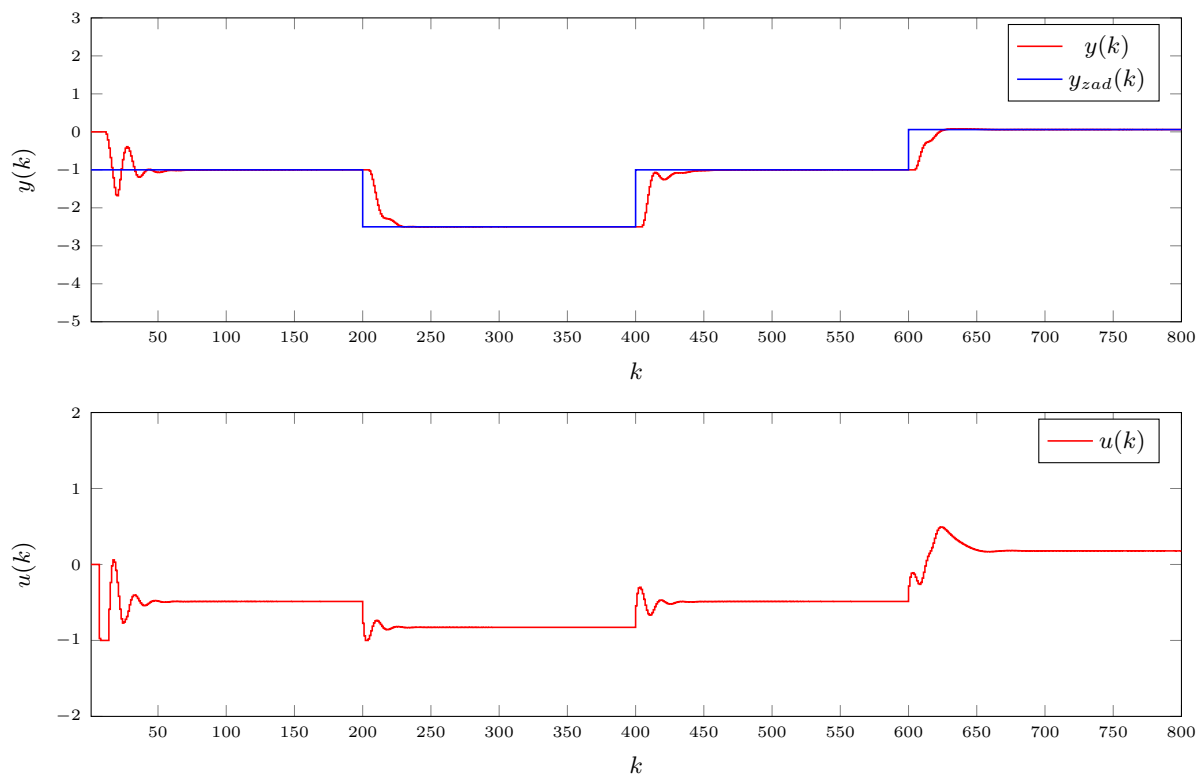
Sprawdzone zostały również regulatory DMC o innych wartościach parametru λ . Parametry regulatorów rozmytych w zależności od parametrów składowych oraz błędy podczas symulacji są podane poniżej:

- 2 regulatory: $\Rightarrow \lambda_1 = 18 \ \lambda_2 = 0,9 \quad E = 6,937\,087 \cdot 10^1$ (wykres 7.1)
- 3 regulatory: $\Rightarrow \lambda_1 = 18 \ \lambda_2 = 12 \ \lambda_3 = 0,9 \quad E = 6,548\,492 \cdot 10^1$ (wykres 7.2)
- 4 regulatory: $\Rightarrow \lambda_1 = 17 \ \lambda_2 = 16 \ \lambda_3 = 9 \ \lambda_4 = 1 \quad E = 6,503\,162 \cdot 10^1$ (wykres 7.3)
- 5 regulatorów: $\Rightarrow \lambda_1 = 16 \ \lambda_2 = 16 \ \lambda_3 = 10 \ \lambda_4 = 5 \ \lambda_5 = 1,5 \quad E = 6,437\,018 \cdot 10^1$ (wykres 7.4)

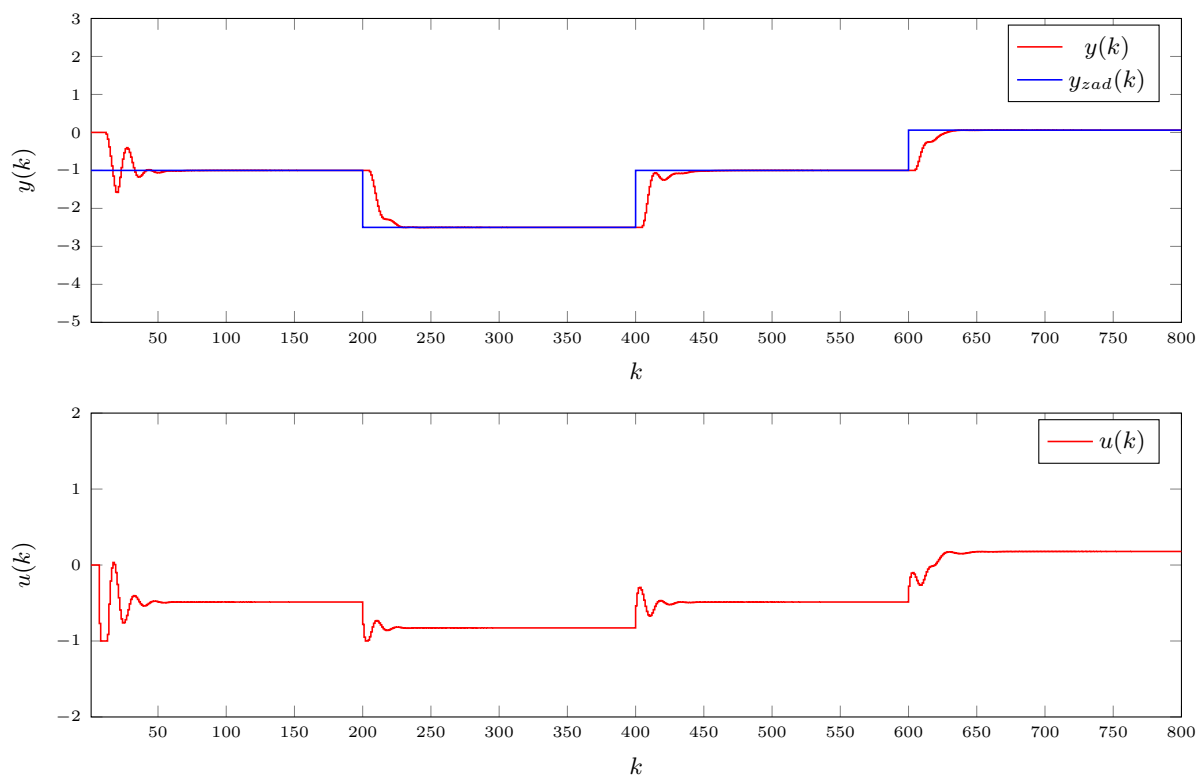
Jak widać, zmiana parametrów $\lambda_1, \lambda_2, \lambda_3, \lambda_4$ oraz λ_5 poprawiła, choć nieznacznie, wartość wskaźnika jakości regulacji.



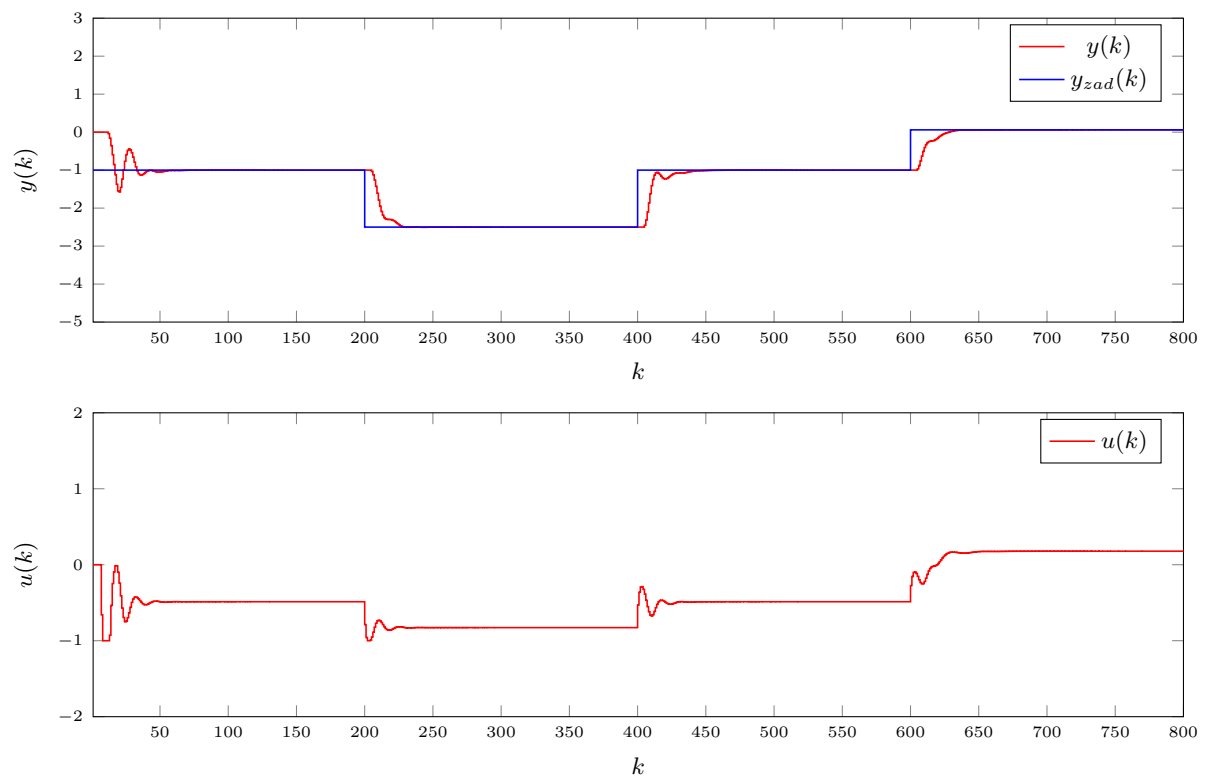
Rys. 7.1. Poprawiony ozmyty regulator DMC, składający się z dwóch regulatorów składowych



Rys. 7.2. Poprawiony ozmyty regulator DMC, składający się z trzech regulatorów składowych



Rys. 7.3. Poprawiony ozmyty regulator DMC, składający się z czterech regulatorów składowych



Rys. 7.4. Poprawiony rozmyty regulator DMC, składający się z pięciu regulatorów składowych