Wydział Elektroniki i Technik Informacyjnych
Politechnika Warszawska

Projektowanie układów sterowania (projekt grupowy)

Sprawozdanie z projektu i ćwiczenia laboratoryjnego nr 3, zadanie nr 6

Mateusz Koroś, Ksawery Pasikowski, Mateusz Morusiewicz

Warszawa, 2017

Spis treści

1. Zad. 1

Poprawność punktu pracy została udowodniona poprzez sprawdzenie, czy obiekt, będący w punkcie pracy, pozostanie w nim, jeśli wartości sterowania i zakłóceń pozostaną takie same. Zostało to wykonane za pomocą komend:

```
y1_ust = symulacja_obiektu6y1(0, 0, 0, 0, 0, 0)
y2_ust = symulacja_obiektu6y2(0, 0, 0, 0, 0, 0)
```

Co dało wynik [0,0], co dowodzi, że punktem pracy rzeczywiście jest punkt $u_1 = u_2 = y_1 = y_2 = 0$

2. Zad. 2

Oba sterowania zostały wzbudzone do wartości:

— 0,4
— 0,8
— −0,4
— −0,8

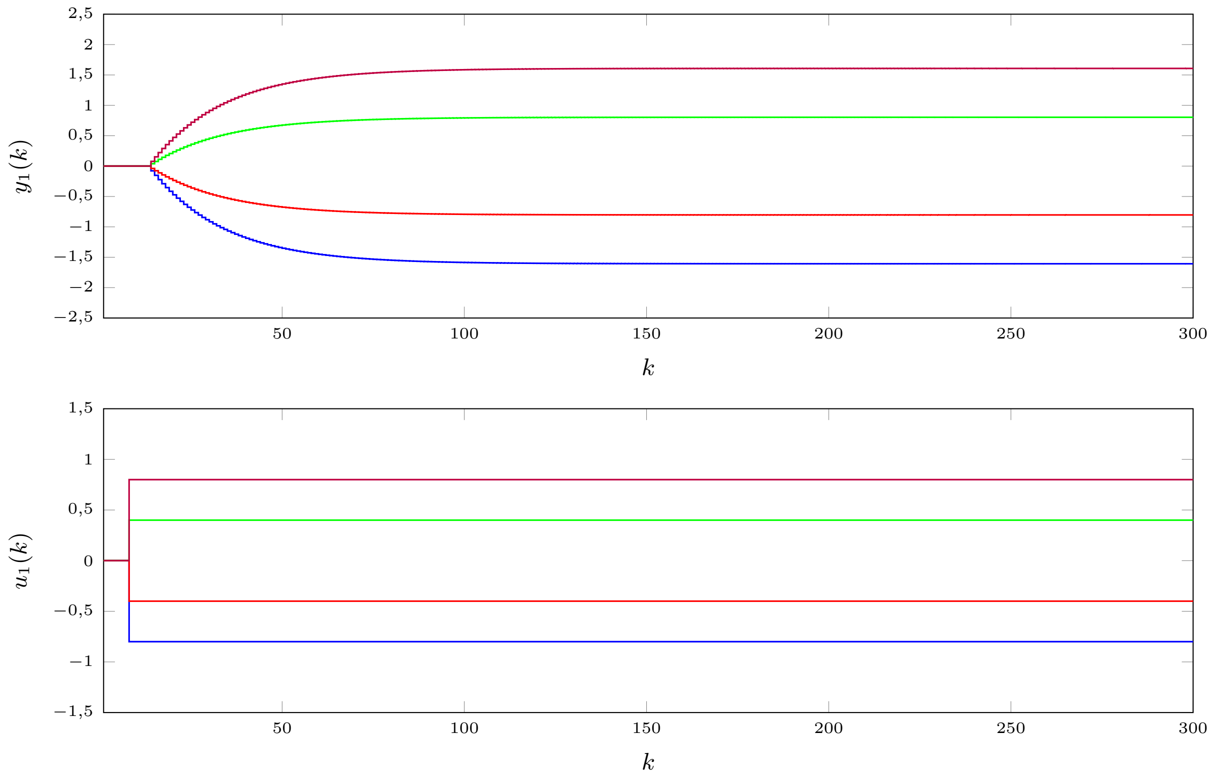
Z charakterystyk widać, że właściwości statyczne procesu są liniowe. Właściwości dynamiczne również, stąd możemy obliczyć wzmocnienia statyczne 4 torów:

$$K_{11} = \frac{\Delta y_1}{\Delta u_1} = 2{,}01$$

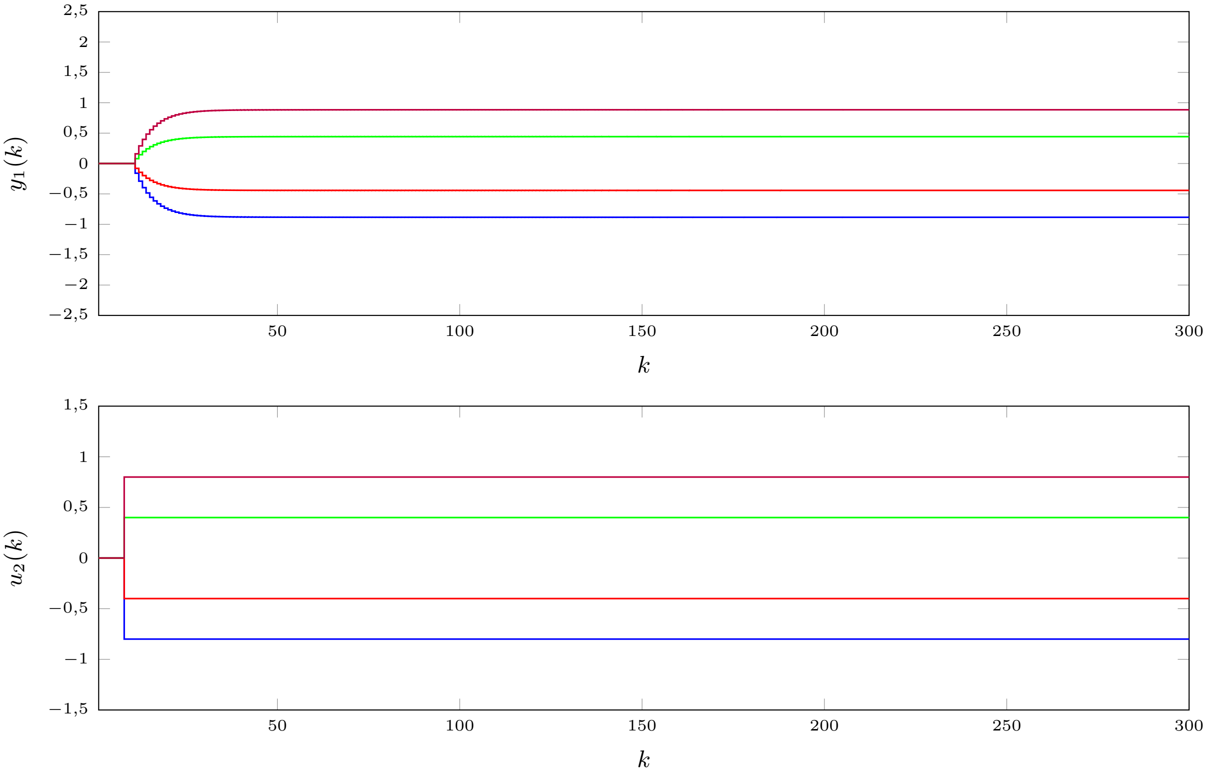
$$K_{12} = \frac{\Delta y_1}{\Delta u_2} = 1{,}11$$

$$K_{21} = \frac{\Delta y_2}{\Delta u_1} = 1{,}11$$

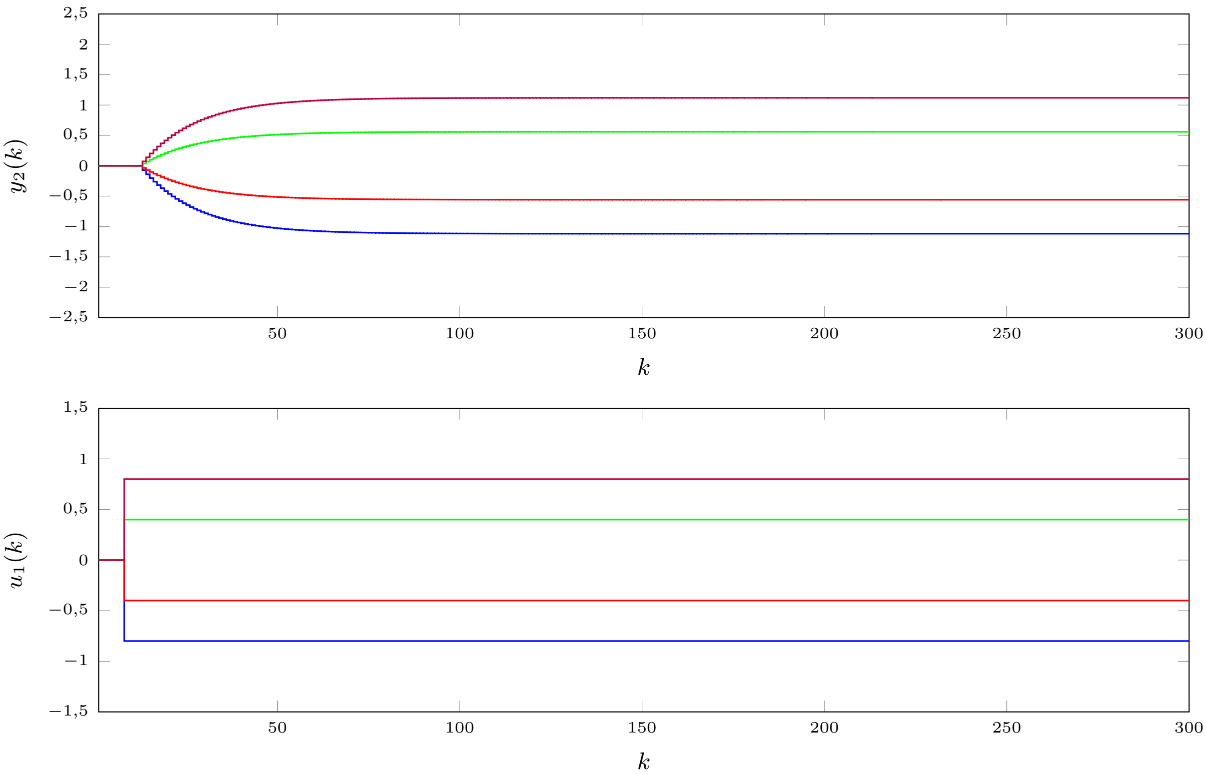
$$K_{22} = \frac{\Delta y_2}{\Delta u_2} = 2{,}50$$



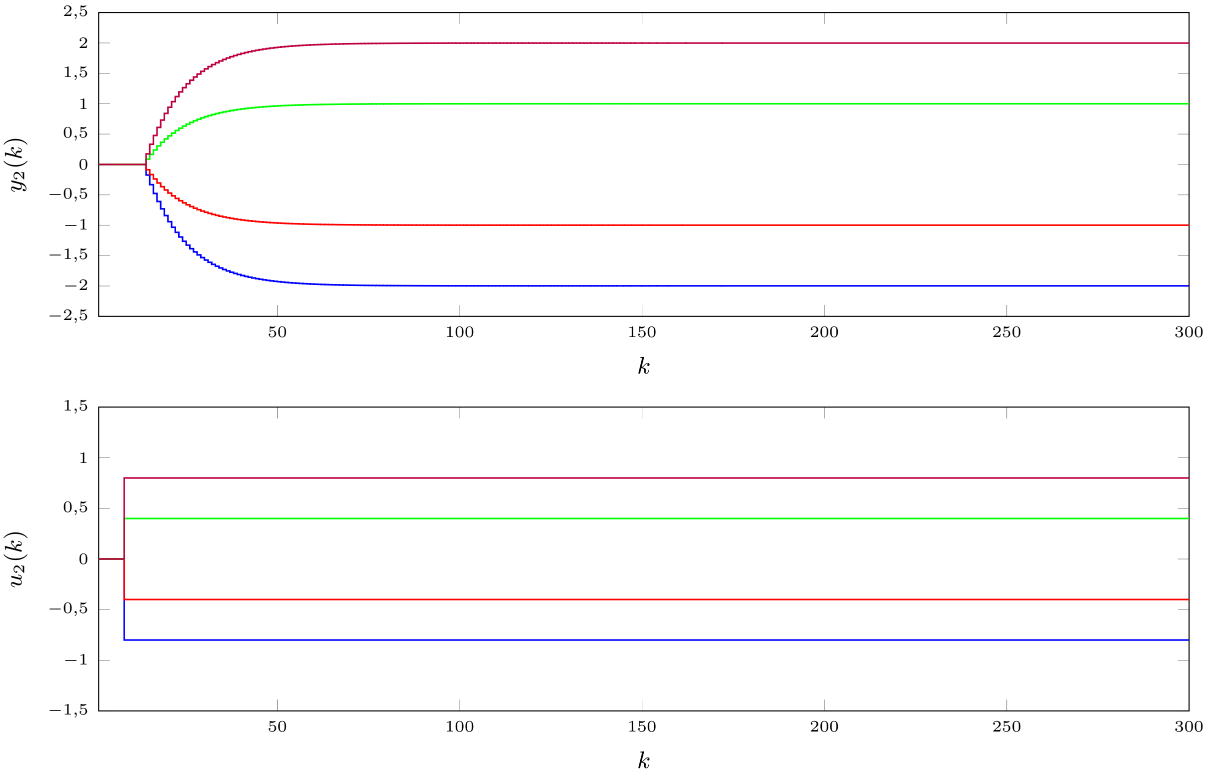
Rys. 2.1. Odpowiedź skokowa y_1 na zmianę wartości u_1



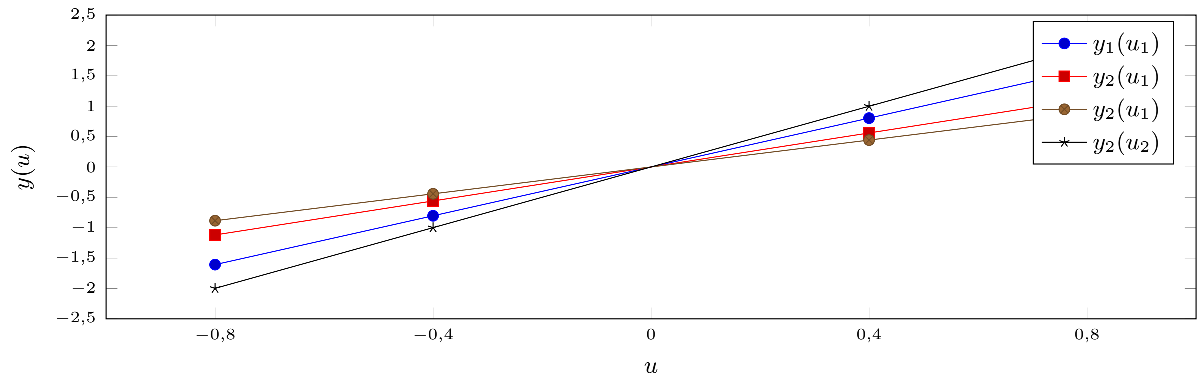
Rys. 2.2. Odpowiedź skokowa y_1 na zmianę wartości u_2



Rys. 2.3. Odpowiedź skokowa y_2 na zmianę wartości u_1

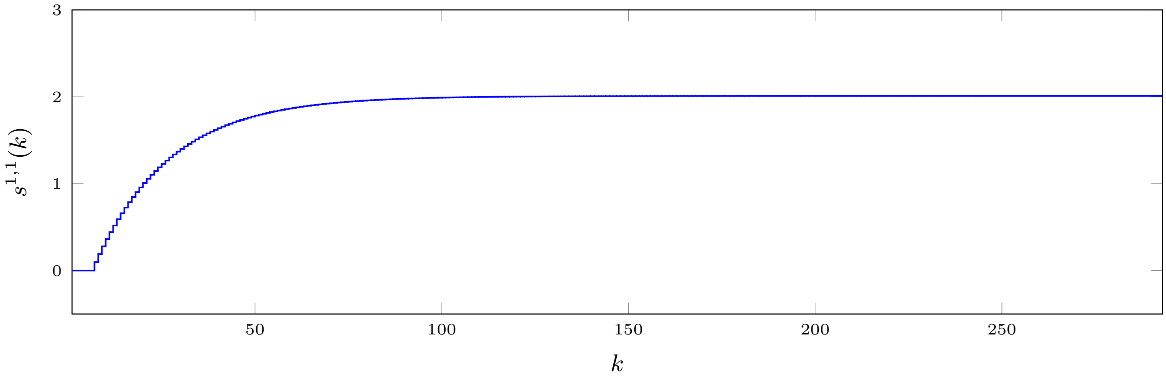


Rys. 2.4. Odpowiedź skokowa y_2 na zmianę wartości u_2

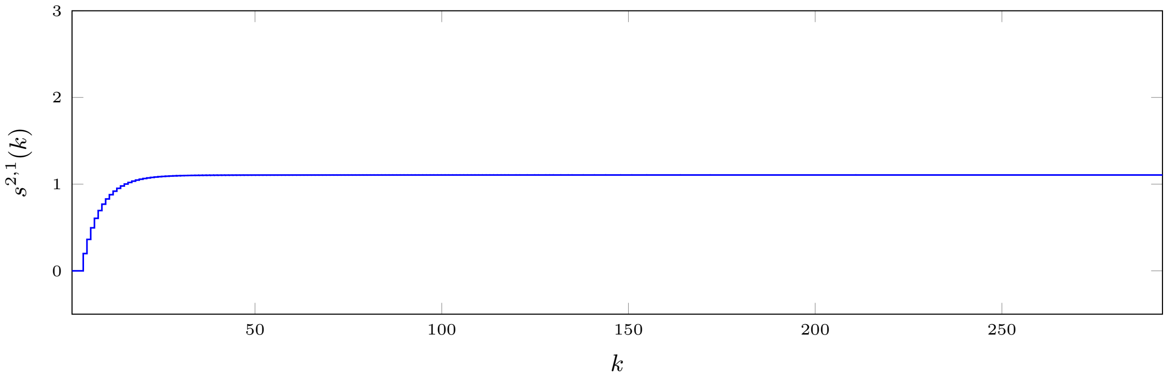


Rys. 2.5. Charakterystyki statyczne procesu

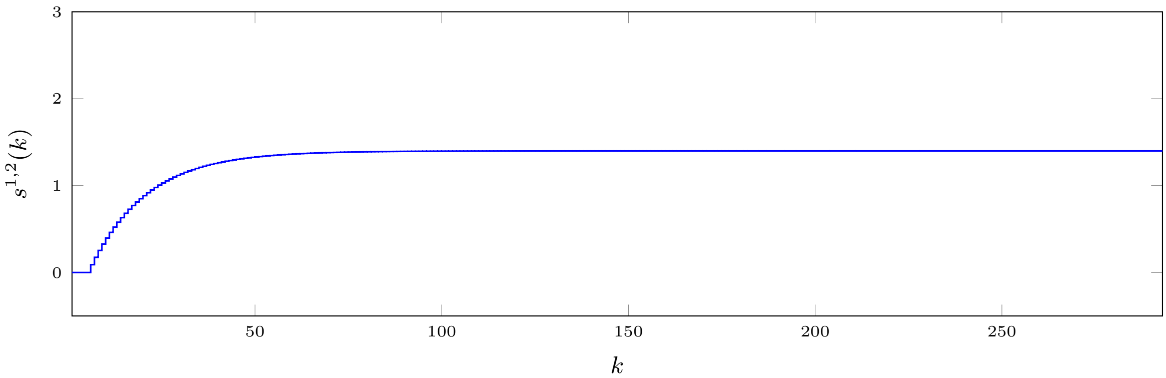
3. Zad. 3



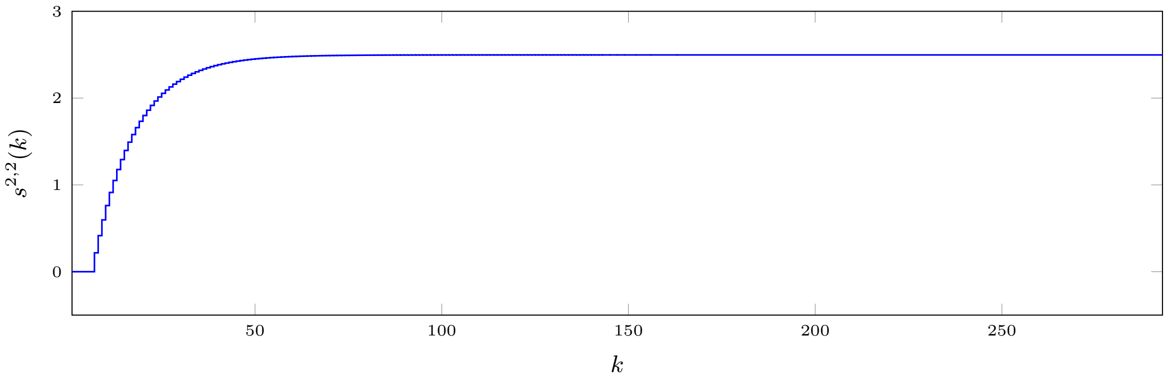
Rys. 3.1. Odpowiedź skokowa używana w algorytmie DMC dla toru $u_1 - y_1$



Rys. 3.2. Odpowiedź skokowa używana w algorytmie DMC dla toru $u_2 - y_1$



Rys. 3.3. Odpowiedź skokowa używana w algorytmie DMC dla toru $u_1 - y_2$



Rys. 3.4. Odpowiedź skokowa używana w algorytmie DMC dla toru $u_2 - y_2$

4. Zad. 4

4.1. PID

```
function [ y, u, E, yzad ] = policzPID( Kp1_, Ti1_, Td1_, ...
Kp2_, Ti2_, Td2_, Kk_, config, nrZad, zakl)
Kp1 = Kp1_;
Ti1 = Ti1_;
Td1 = Td1_;
Kp2 = Kp2_;
Ti2 = Ti2_;
Td2 = Td2_;

Kk = Kk_;
Tp = 0.5;

r2_1 = (Kp1 * Td1) / Tp ;
r1_1 = Kp1 * ( (Tp/(2*Ti1)) - 2*(Td1/Tp) - 1 ) ;
r0_1 = Kp1 * ( 1 + Tp/(2*Ti1) + Td1/Tp ) ;

r2_2 = (Kp2 * Td2) / Tp ;
r1_2 = Kp2 * ( (Tp/(2*Ti2)) - 2*(Td2/Tp) - 1 ) ;
r0_2 = Kp2 * ( 1 + Tp/(2*Ti2) + Td2/Tp ) ;

% warunki poczatkowe
dist1(1 : Kk - 300) = 0;
dist1(Kk - 299 : Kk) = 0.5;

dist2(1 : Kk - 150) = 0;
dist2(Kk - 149 : Kk) = 0.3;

u1(1:11) = 0 ;
y1(1:11) = 0 ;
e1(1:11) = 0 ;
u2(1:11) = 0 ;
y2(1:11) = 0 ;
e2(1:11) = 0 ;
index1 = 1;
index2 = 1;

if nrZad == 5
yzads1 = [1.05 0.8 1.1 0.9];
yzads2 = [0.9 0.8 1.1 1];
else
yzads1 = [1.05 1.05 1.05 1.05];
```

```
yzads2 = [0.9 0.9 0.9 0.9];
end

yzad1 = yzads1(index1);
yzad2 = yzads2(index2);
yzadVec1(1:Kk) = yzad1;
yzadVec2(1:Kk) = yzad2;

% glowna petla symulacji
for k = 12 : Kk
if mod(k,200) == 0
index1 = index1 + 1;
if index1 > length(yzads1)
index1 = length(yzads1);
end
yzad1 = yzads1(index1);
end
yzadVec1(k) = yzad1;

if mod(k,120) == 0
index2 = index2 + 1;
if index2 > length(yzads2)
index2 = length(yzads2);
end
yzad2 = yzads2(index2);
end
yzadVec2(k) = yzad2;
y1(k)=symulacja_obiektu6y1(u1(k-6),u1(k-7),u2(k-3),...
u2(k-4),y1(k-1),y1(k-2));
y2(k)=symulacja_obiektu6y2(u1(k-5),u1(k-6),u2(k-6),...
u2(k-7),y2(k-1),y2(k-2));

if nrZad == 7
y1(k) = y1(k) + dist1(k);
y2(k) = y2(k) + dist2(k);
end

if nrZad == 6
y1(k) = y1(k) + zakl();
y2(k) = y2(k) + zakl();
end


e1(k) = yzad1 - y1(k) ;
e2(k) = yzad2 - y2(k) ;

if config == 1
u1(k) = r2_1 * e1(k-2) + r1_1 * e1(k-1) +...
r0_1 * e1(k) + u1(k-1) ;
u2(k) = r2_2 * e2(k-2) + r1_2 * e2(k-1) +...
r0_2 * e2(k) + u2(k-1) ;
```

```
else
u1(k) = r2_1 * e2(k-2) + r1_1 * e2(k-1) + ...
r0_1 * e2(k) + u1(k-1) ;
u2(k) = r2_2 * e1(k-2) + r1_2 * e1(k-1) + ...
r0_2 * e1(k) + u2(k-1) ;
end

end

E1 = (yzadVec1 - y1) * (yzadVec1 - y1)';
E2 = (yzadVec2 - y2) * (yzadVec2 - y2)';
E = E1 + E2;

yzad = zeros(2, Kk);
yzad(1, :) = yzadVec1;
yzad(2, :) = yzadVec2;

y = zeros(2, Kk);
y(1, :) = y1;
y(2, :) = y2;

u = zeros(2, Kk);
u(1, :) = u1;
u(2, :) = u2;

end
```

4.2. DMC

```
function [ y, u, E, yzad ] = policzDMC(D_, N_, Nu_,...
lambda, Kk_, nrZad, zakl)
N = N_;
Nu = Nu_;
D=D_;

Su1y1 = load('wykresy_pliki/zad3/skok_sterowania/...
odp_skok_y1_u1_ster_0.4.txt');
Su2y1 = load('wykresy_pliki/zad3/skok_sterowania/...
odp_skok_y1_u2_ster_0.4.txt');
Su1y2 = load('wykresy_pliki/zad3/skok_sterowania/...
odp_skok_y2_u1_ster_0.4.txt');
Su2y2 = load('wykresy_pliki/zad3/skok_sterowania/...
odp_skok_y2_u2_ster_0.4.txt');
s11=Su1y1(:,2);
s12=Su2y1(:,2);
s21=Su1y2(:,2);
s22=Su2y2(:,2);

kk=Kk_;
dist1(1 : kk - 300) = 0;
```

```
dist1(kk - 299 : kk) = 0.5;

dist2(1 : kk - 150) = 0;
dist2(kk - 149 : kk) = 0.3;

ny=2;
nu=2;
E1=0;
E2=0;

if nrZad == 5
yzads1 = [1.05 0.8 1.1 0.9];
yzads2 = [0.9 0.8 1.1 1];
else
yzads1 = [1.05 1.05 1.05 1.05];
yzads2 = [0.9 0.9 0.9 0.9];
end

index1 = 1;
index2 = 1;
y=zeros(ny,kk);
yzad=zeros(ny,kk);
yzad1 = yzads1(index1);
yzad2 = yzads2(index2);
yzad(1,Kk_) = yzad1;
yzad(2,Kk_) = yzad2;
u=zeros(nu,kk);
du=zeros(nu,kk);
dUP=cell(D-1,1);
dUP(1:D-1)={zeros(2,1)};
M=cell(N,Nu);

for i=1:N
for j=1:Nu
if (i>=j)
M(i,j)={[s11(i-j+1) s12(i-j+1); s21(i-j+1) s22(i-j+1)]};
else
M(i,j)={zeros(nu,ny)};
end
end
end

MP=cell(N,D-1);
for i=1:N
for j=1:D-1
if i+j<=D
MP(i,j)={[s11(i+j)-s11(j) s12(i+j)-s12(j); ...
s21(i+j)-s21(j) s22(i+j)-s22(j)]};
else
MP(i,j)={[s11(D)-s11(j) s12(D)-s12(j); ...
s21(D)-s21(j) s22(D)-s22(j)]};
```

```
end
end
end

K=(cell2mat(M)'*cell2mat(M)+...
diag(ones(1,Nu*nu)*lambda))^(-1)*cell2mat(M)';
ku=K(1:nu,:)*cell2mat(MP);
ke1=sum(K(1,1:2:(N*ny)));
ke2=sum(K(1,2:2:(N*ny)));
ke3=sum(K(2,1:2:(N*ny)));
ke4=sum(K(2,2:2:(N*ny)));

for k=10:kk

if mod(k, 200) == 0
index1 = index1 + 1;
if index1 > length(yzads1)
index1 = length(yzads1);
end
yzad1 = yzads1(index1);
end
yzad(1,k) = yzad1;

if mod(k, 120) == 0
index2 = index2 + 1;
if index2 > length(yzads2)
index2 = length(yzads2);
end
yzad2 = yzads2(index2);
end
yzad(2,k) = yzad2;


y(1,k)=symulacja_obiektu6y1(u(1,k-6),u(1,k-7),...
u(2,k-3),u(2,k-4),y(1,k-1),y(1,k-2));
y(2,k)=symulacja_obiektu6y2(u(1,k-5),u(1,k-6),...
u(2,k-6),u(2,k-7),y(2,k-1),y(2,k-2));

if nrZad == 6
y(1,k) = y(1,k) + zakl();
y(2,k) = y(2,k) + zakl();
end

if nrZad == 7
y(1,k) = y(1,k) + dist1(k);
y(2,k) = y(2,k) + dist2(k);
end

du(:,k)=[ke1 ke2;ke3 ke4]*(yzad(:,k)-y(:,k))-ku*cell2mat(dUP);

for i=D-1:-1:2
```

```
dUP(i)=dUP(i-1);
end

dUP(1)={du(:,k)};
u(:,k)=u(:,k-1)+du(:,k);

end

for k=1:kk
E1= E1 + ((yzad(1,k) - y(1,k))^2);
E2= E2 + ((yzad(2,k) - y(2,k))^2);
end

E=E1+E2;

end
```

5. Zad. 5

Regulatory oceniane były na podstawie wykresów oraz wartości wskaźnika jakości:

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

Wartości wskaźnika jakości dla regulatora:

— PID

Konfiguracja zwykła(uchyb pierwszego wyjścia oddziałuje na pierwszy sygnał sterujący i analogicznie)

$K_p = 1 \quad T_i = 15 \quad T_d = 0{,}8 \rightarrow E = 39{,}626\,82$
$K_p = 0{,}6 \quad T_i = 10 \quad T_d = 0{,}005 \rightarrow E = 41{,}034\,24$
$K_p = 0{,}8 \quad T_i = 8 \quad T_d = 0{,}007 \rightarrow E = 40{,}590\,62$

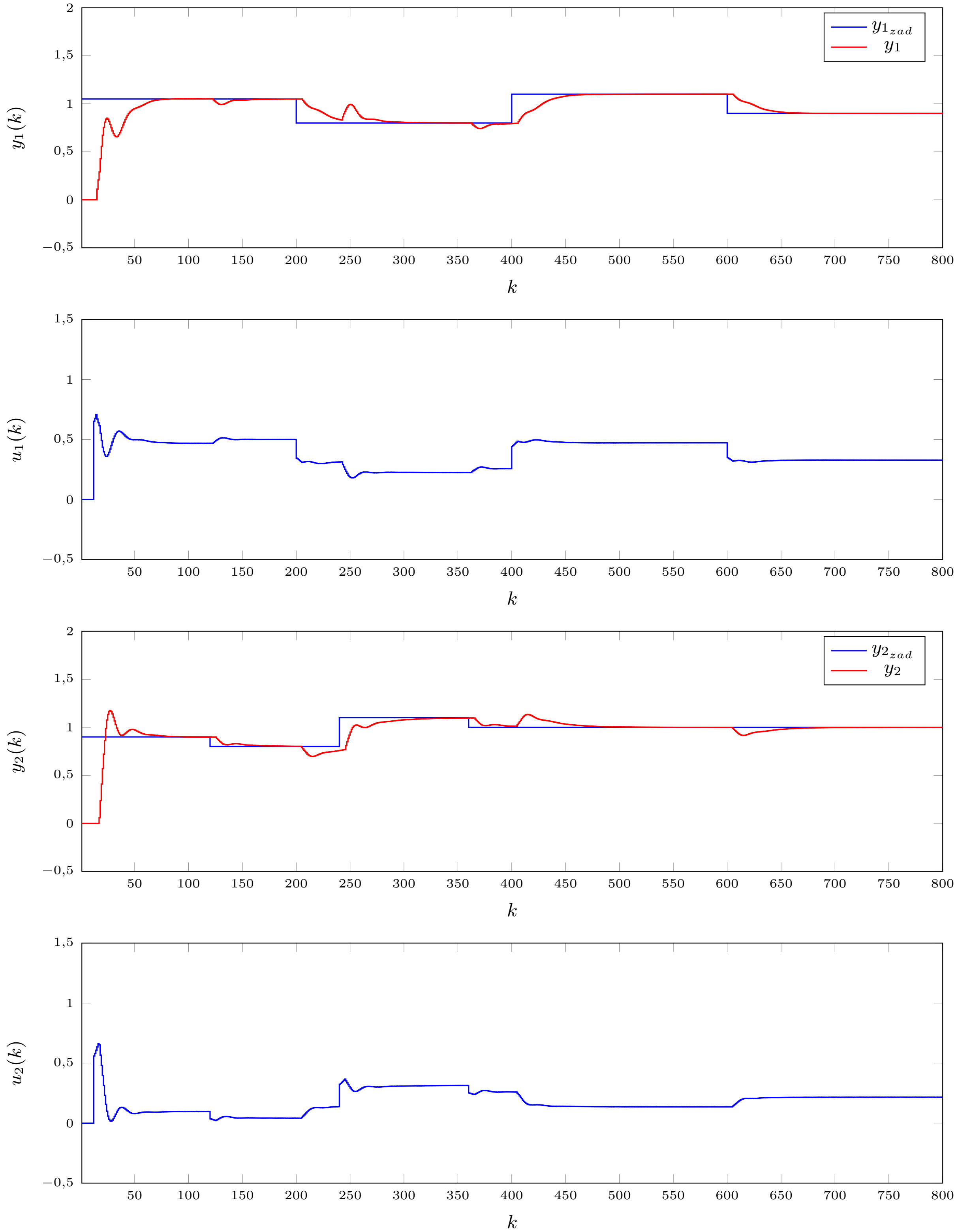
Konfiguracja odwrotna(uchyb pierwszego wyjścia oddziałuje na drugi sygnał sterujący i analogicznie)

$K_p = 1 \quad T_i = 15 \quad T_d = 0{,}8 \rightarrow E = 39{,}626\,82$
$K_p = 0{,}6 \quad T_i = 10 \quad T_d = 0{,}005 \rightarrow E = 41{,}034\,24$
$K_p = 0{,}8 \quad T_i = 8 \quad T_d = 0{,}007 \rightarrow E = 40{,}590\,62$

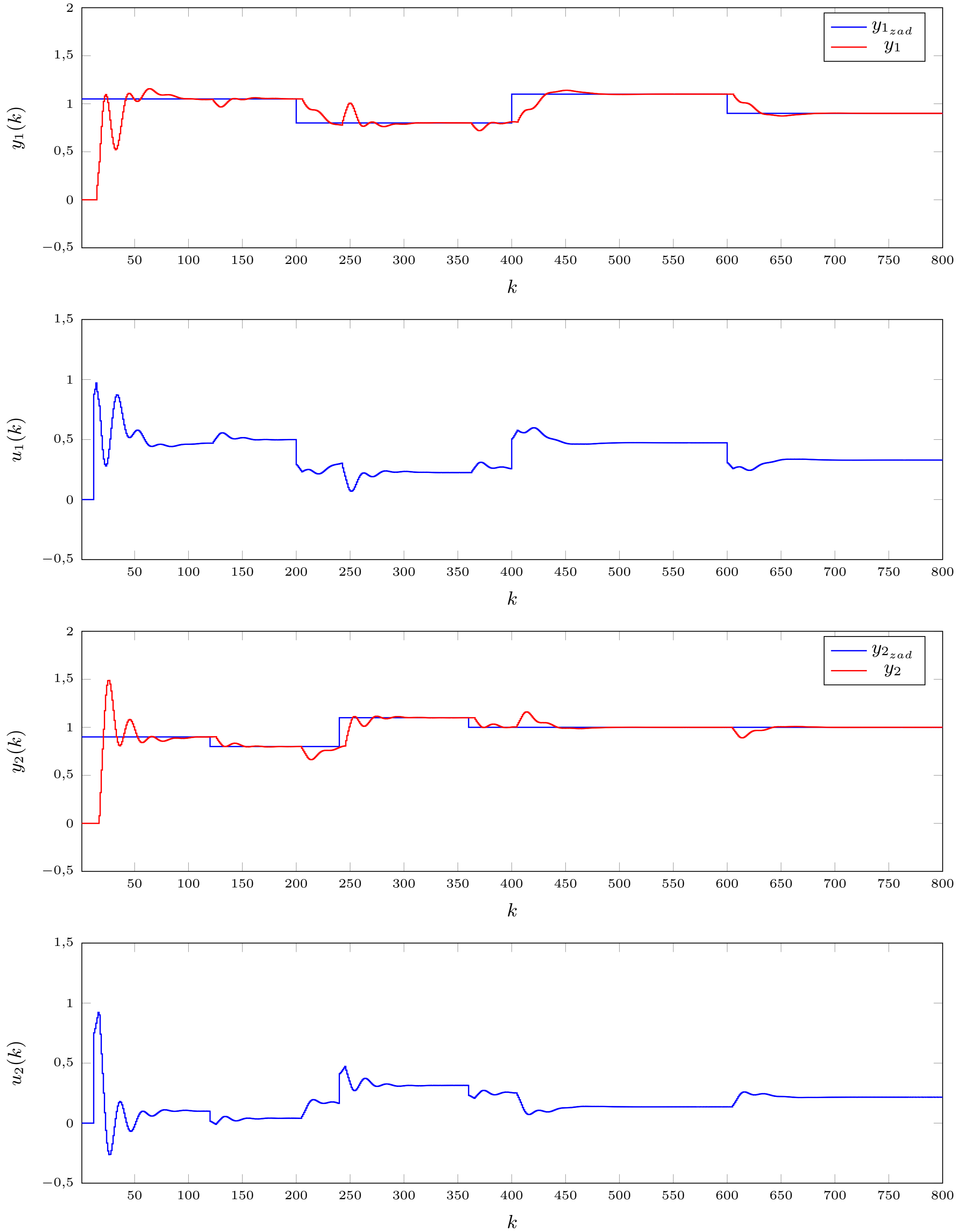
— DMC

$D = N = N_u = 200 \quad \lambda = 1 \rightarrow E = 16{,}189\,37$
$D = N = N_u = 200 \quad \lambda = 5 \rightarrow E = 18{,}647\,37$
$D = N = N_u = 200 \quad \lambda = 10 \rightarrow E = 20{,}140\,27$

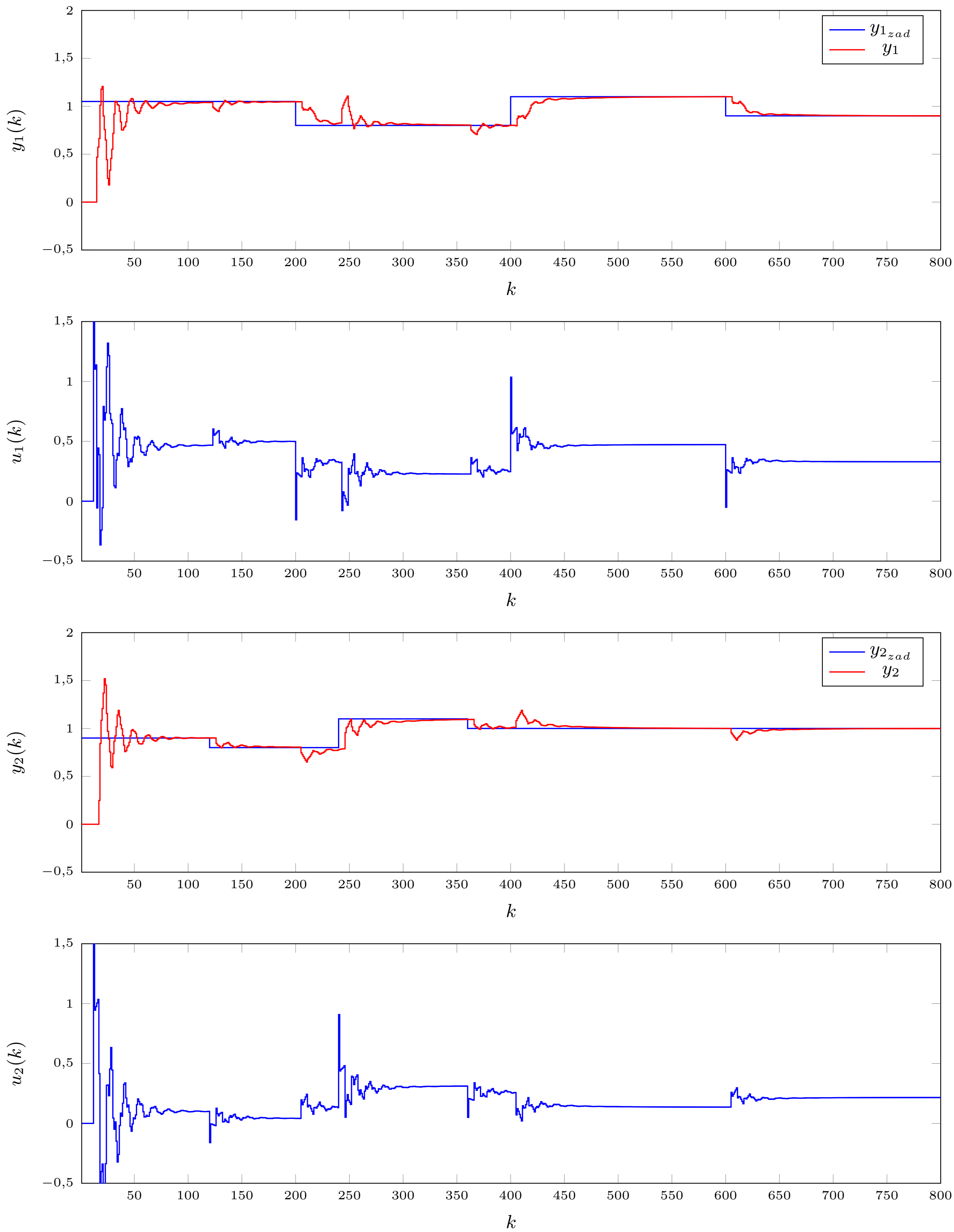
Jak widać najlepszym regulatorem PID okazał się ten o nastawach $K_p = 1 \quad T_i = 15 \quad T_d = 0{,}8$, jednak okazał się on gorszy od każdego z regulatorów DMC. Dowodzi to tego, że regulator DMC lepiej nadaje się do regulacji obiektów wielowymiarowych, zarówno pod względem łatwości implementacji jak i jakości regulacji. Z regulatorów DMC najlepszym jest ten o $\lambda = 1$, o czym świadczy najmniejszy błąd jak i wykres, na którym widać zadowalającą jakość dochodzenia do wartości zadanej przez ten regulator.



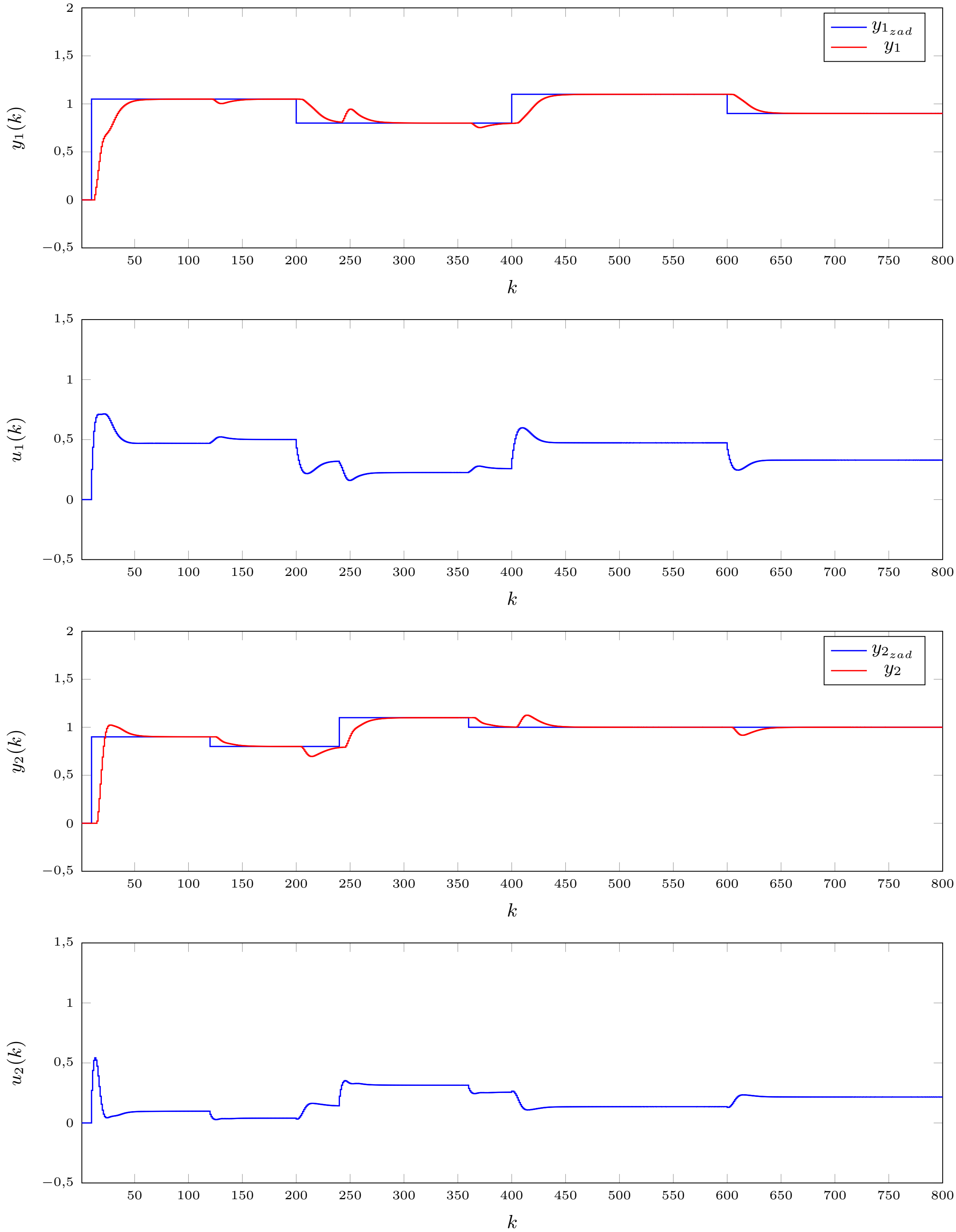
Rys. 5.1. Regulator PID dla nastaw $K_p = 1 \quad T_i = 15 \quad T_d = 0{,}8$



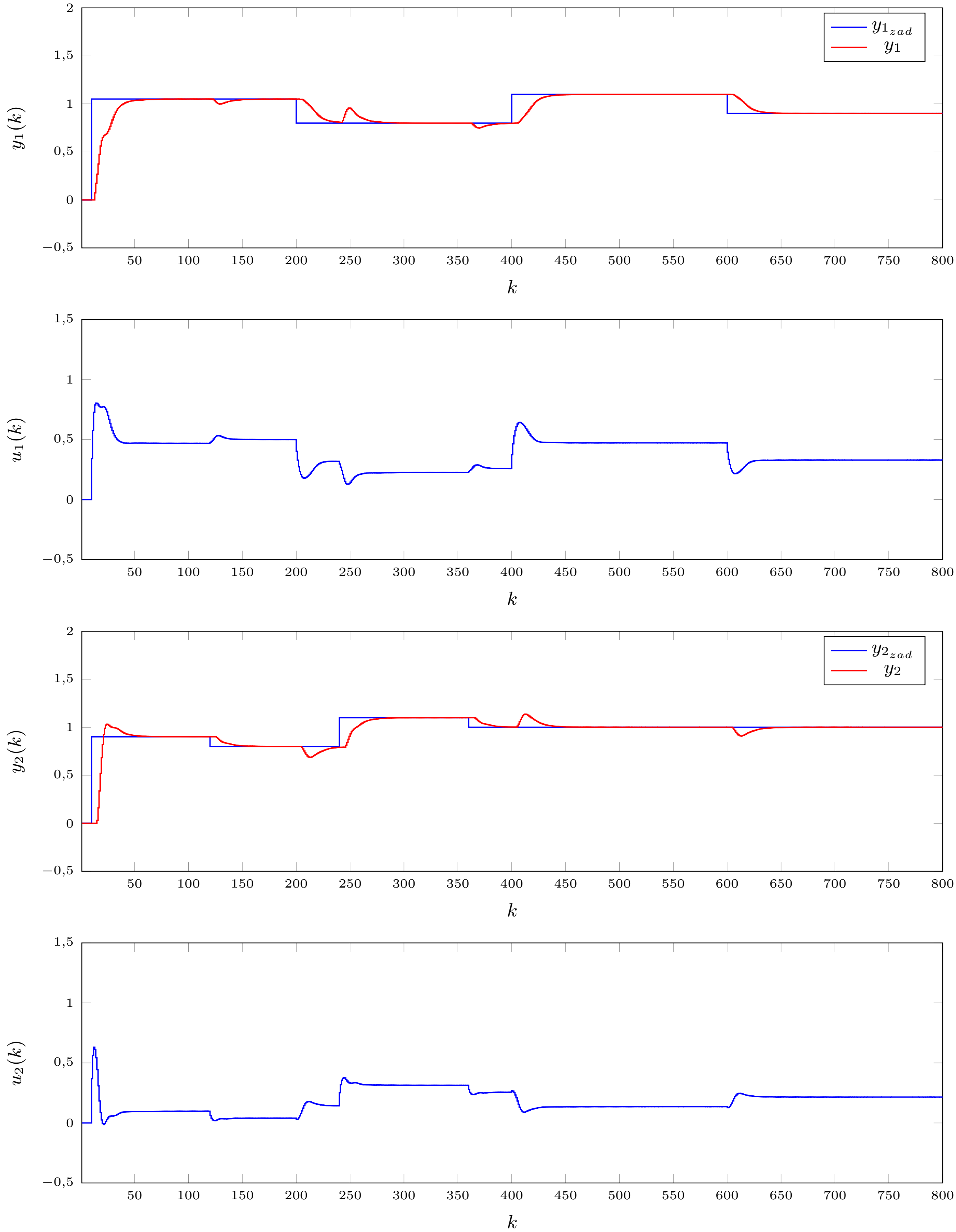
Rys. 5.2. Regulator PID dla nastaw $K_p = 0{,}6 \quad T_i = 10 \quad T_d = 0{,}005$



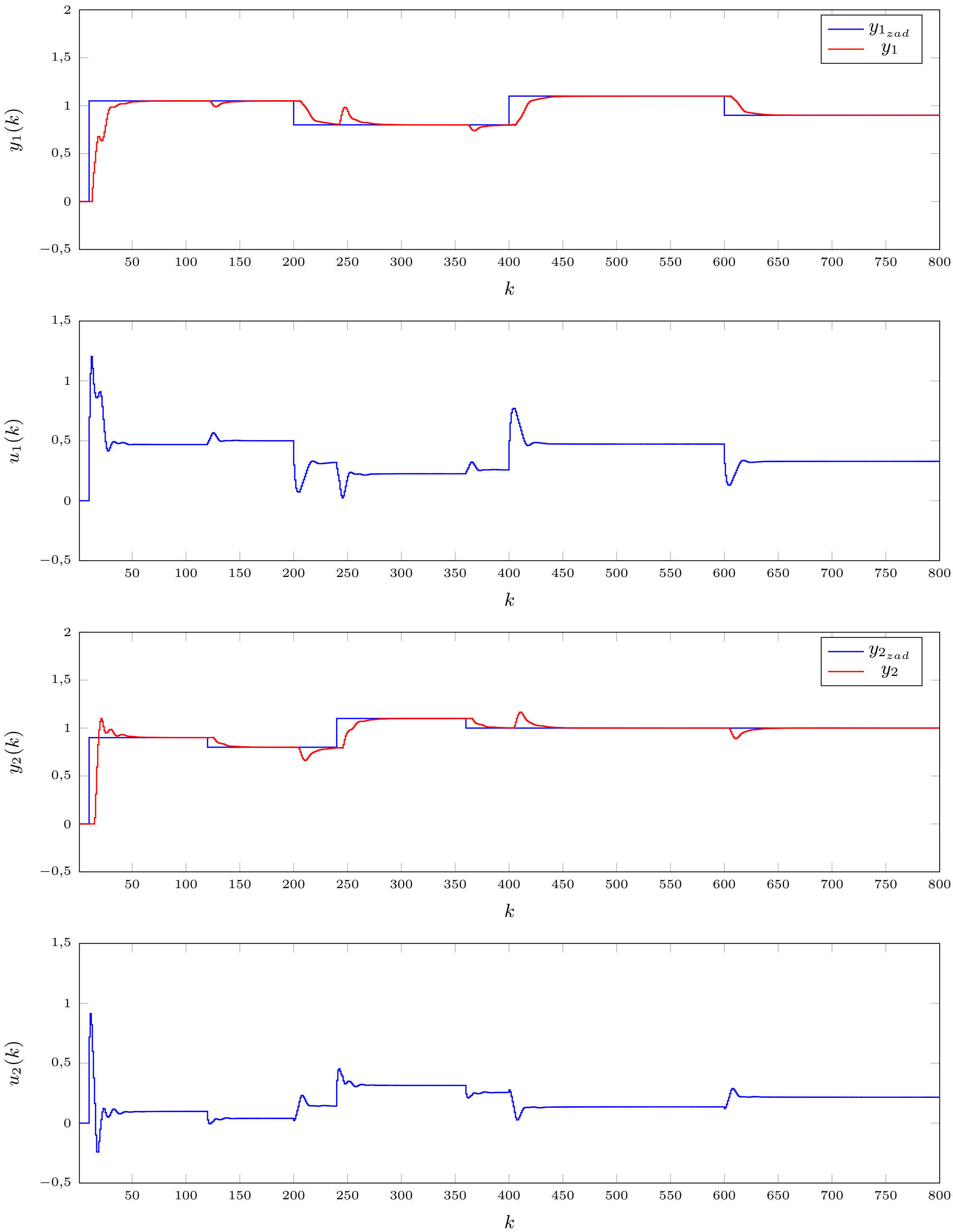
Rys. 5.3. Regulator PID dla nastaw $K_p = 0{,}8 \quad T_i = 8 \quad T_d = 0{,}007$



Rys. 5.4. Regulator DMC dla parametrów $D = N = N_u = 200 \quad \lambda = 1$



Rys. 5.5. Regulator DMC dla parametrów $D = N = N_u = 200 \quad \lambda = 5$



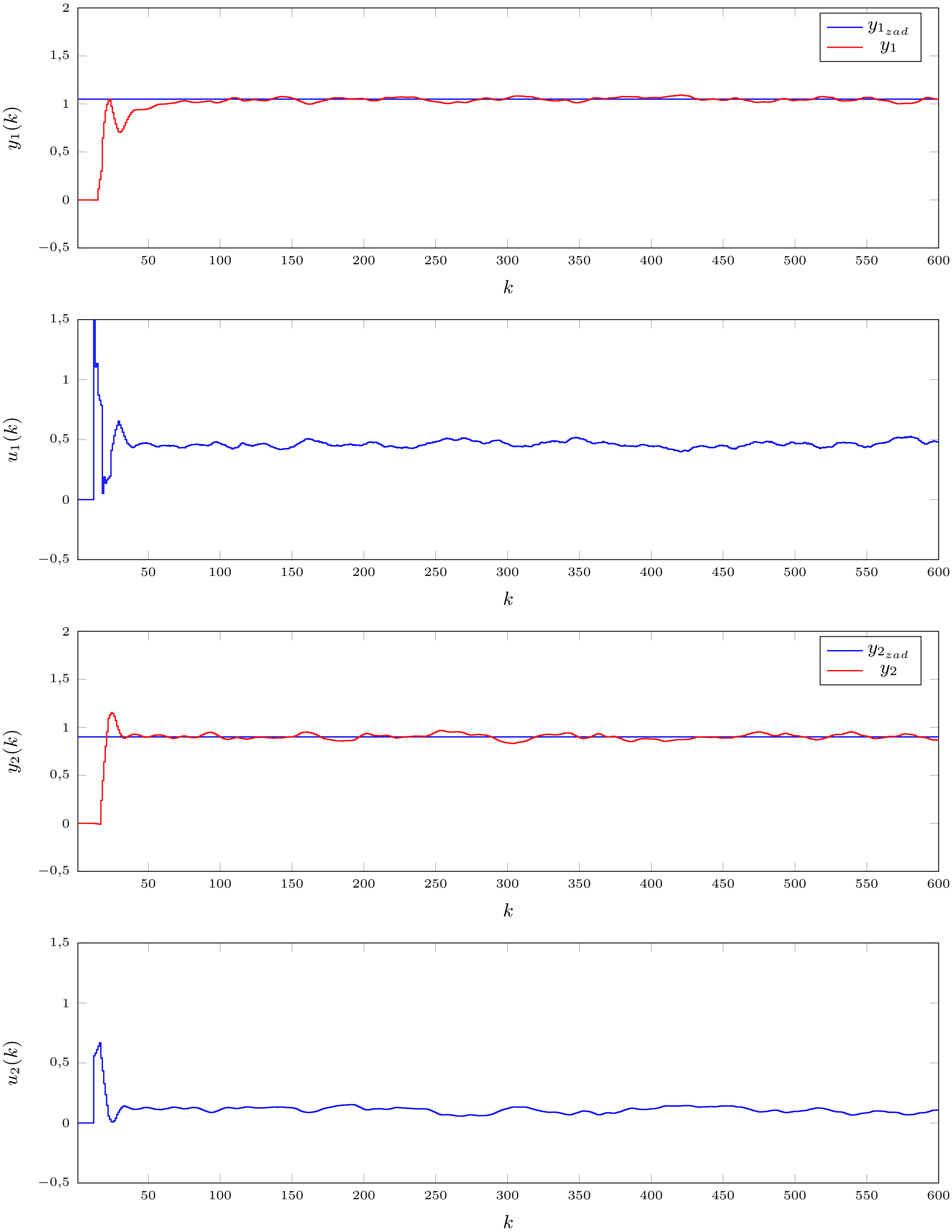
Rys. 5.6. Regulator DMC dla parametrów $D = N = N_u = 200 \quad \lambda = 10$

6. Zad. 6

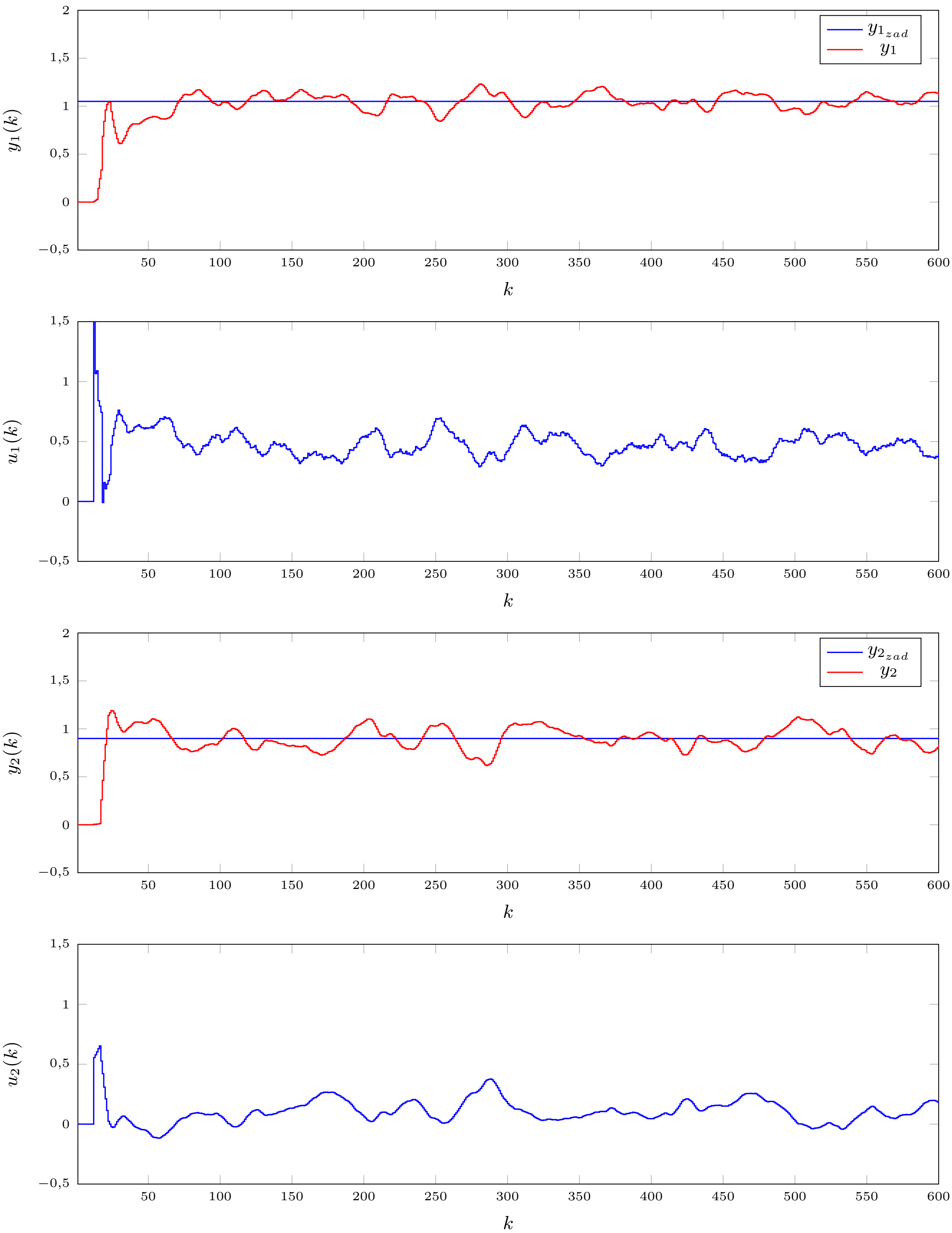
Szumy zostały wygenerowane za pomocą komendy:

```
noises = {@() 0.006*(rand() - 0.5), ...
@() 0.02*(rand() - 0.5), @() 0.06*(rand() - 0.5)};
```

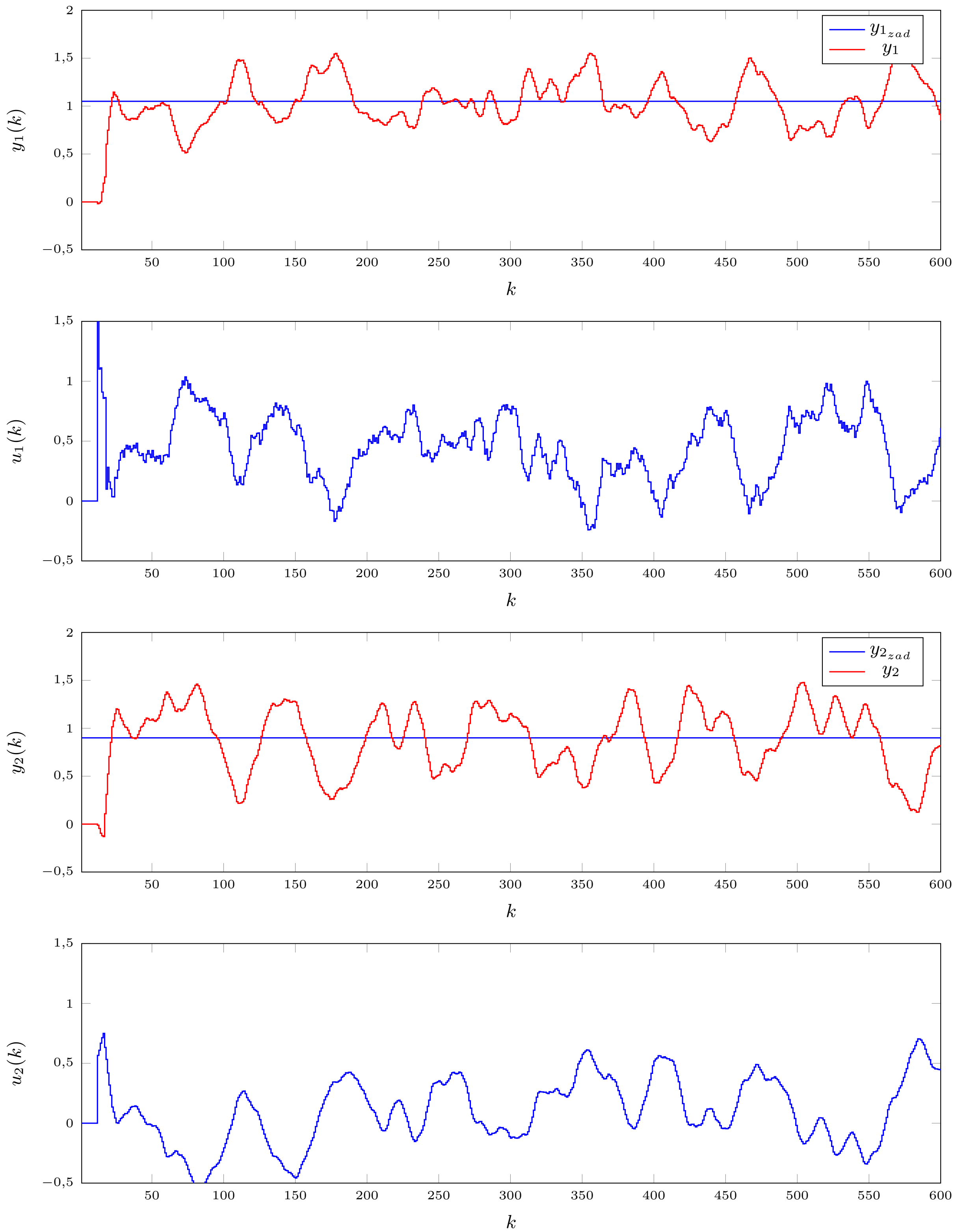
Były one następnie dodawane do mierzonej wartości wyjścia, aby sprawdzić działanie algorytmów PID i DMC przy błędach pomiaru sygnałów. Wyniki widać na poniższych wykresach. Zarówno pierwszy regulator PID jak i pierwszy regulator DMC bardzo dobrze poradziły sobie z zakłóceniami, co można zobaczyć na wykresach 6.1 i 6.4. Inne regulatory miały spore problemy, co świadczy o tym, że szumy pomiarowe mogą poważnie utrudnić regulację, jednak da się ograniczyć utratę jakości regulacji, jeśli stosowany regulator jest wystarczająco odporny.



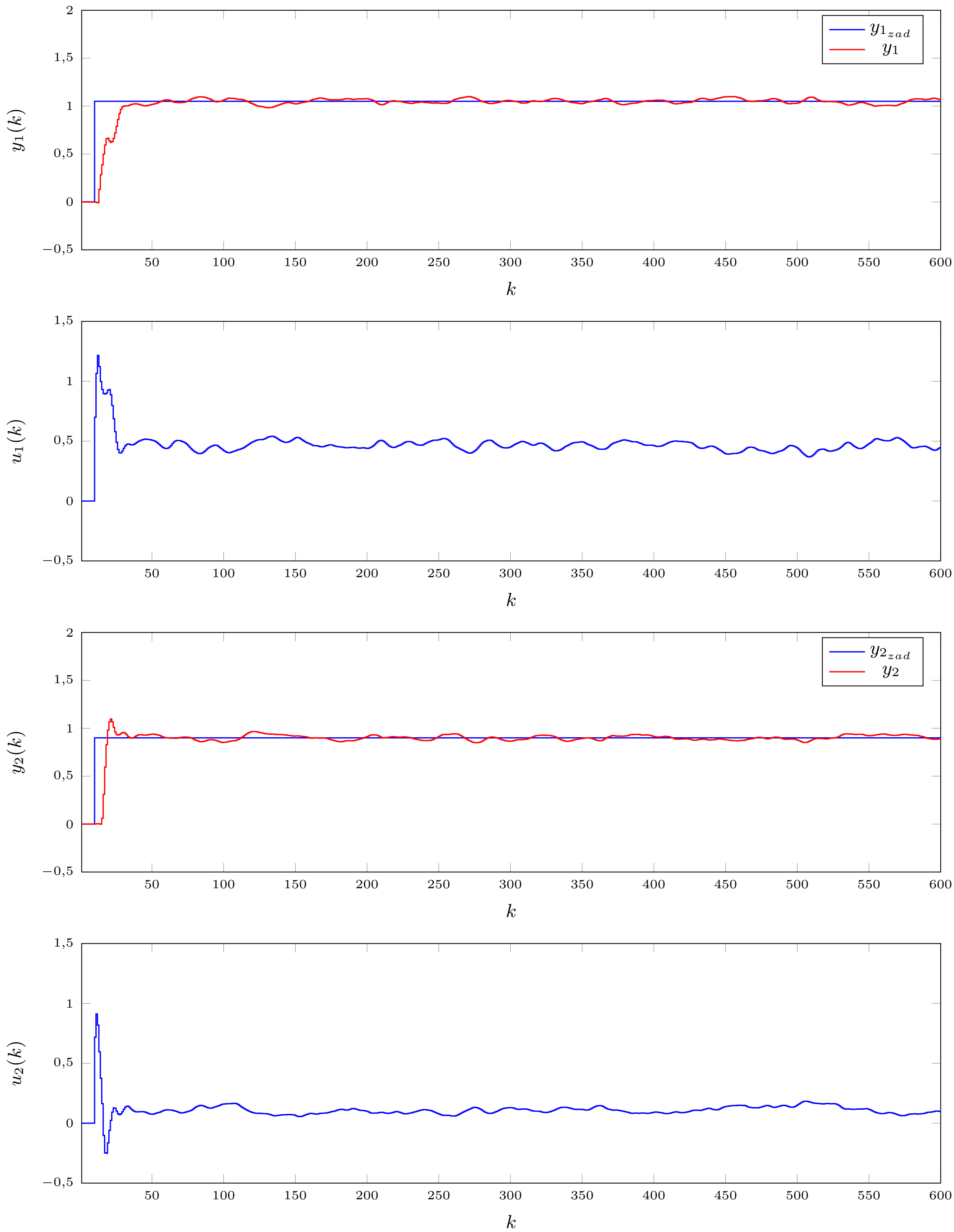
Rys. 6.1. Regulator PID dla nastaw $K_p = 1$ $T_i = 15$ $T_d = 0{,}8$ przy błędach pomiaru sygnałów wyjściowych



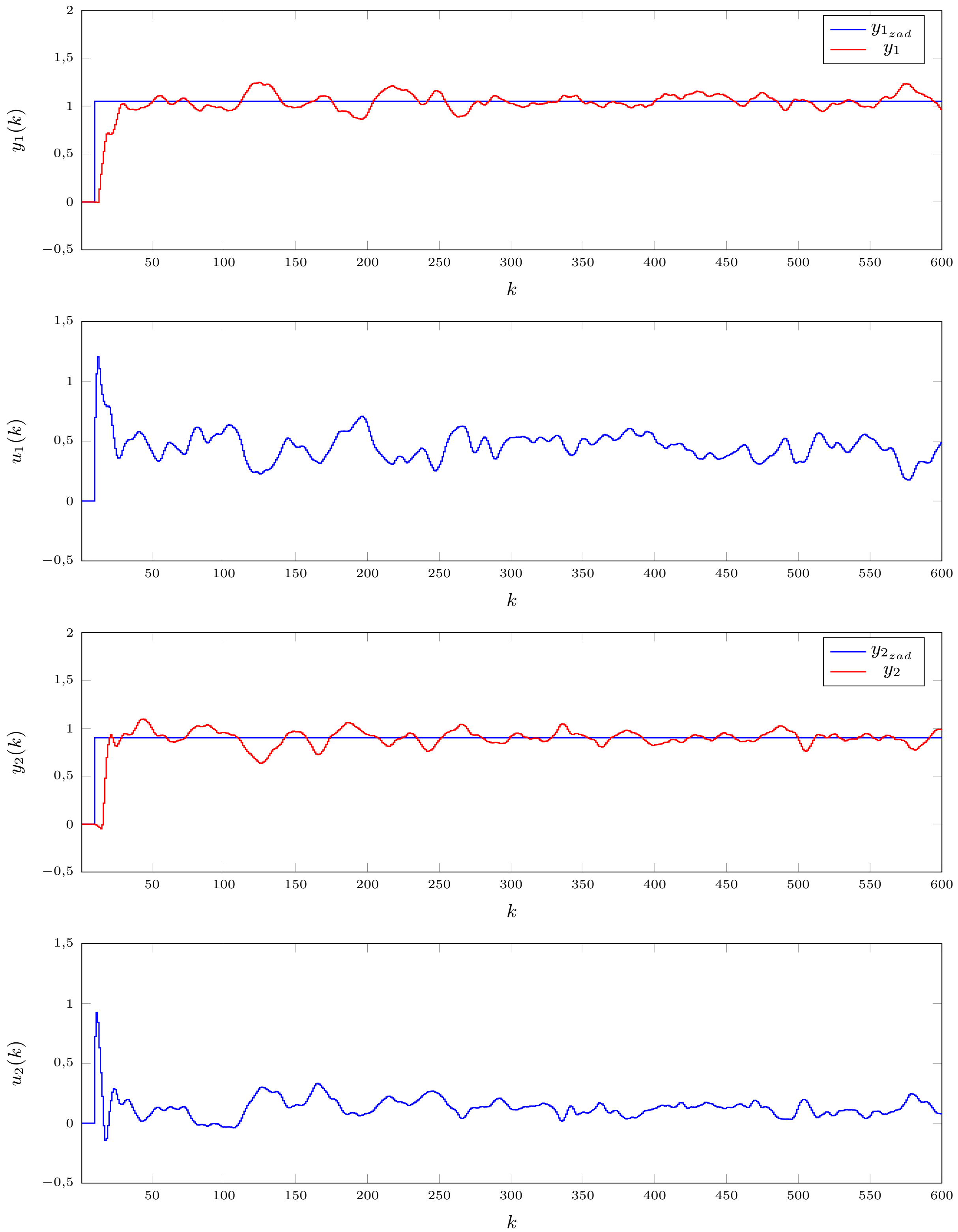
Rys. 6.2. Regulator PID dla nastaw $K_p = 0{,}6$ $T_i = 10$ $T_d = 0{,}005$ przy błędach pomiaru sygnałów wyjściowych



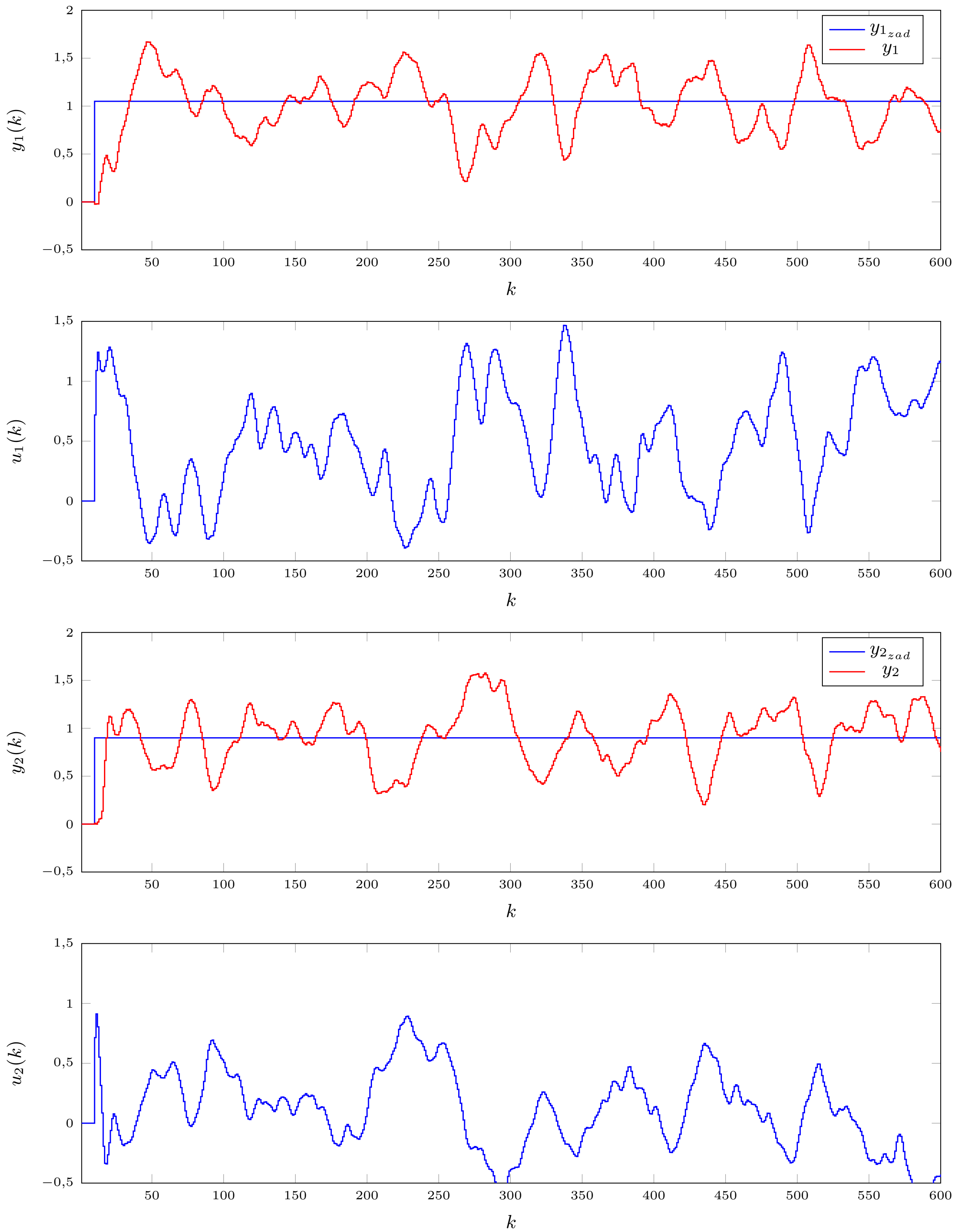
Rys. 6.3. Regulator PID dla nastaw $K_p = 0{,}8$ $T_i = 8$ $T_d = 0{,}007$ przy błędach pomiaru sygnałów wyjściowych



Rys. 6.4. Regulator DMC dla parametrów $D = N = N_u = 200 \quad \lambda = 1$ przy błędach pomiaru sygnałów wyjściowych



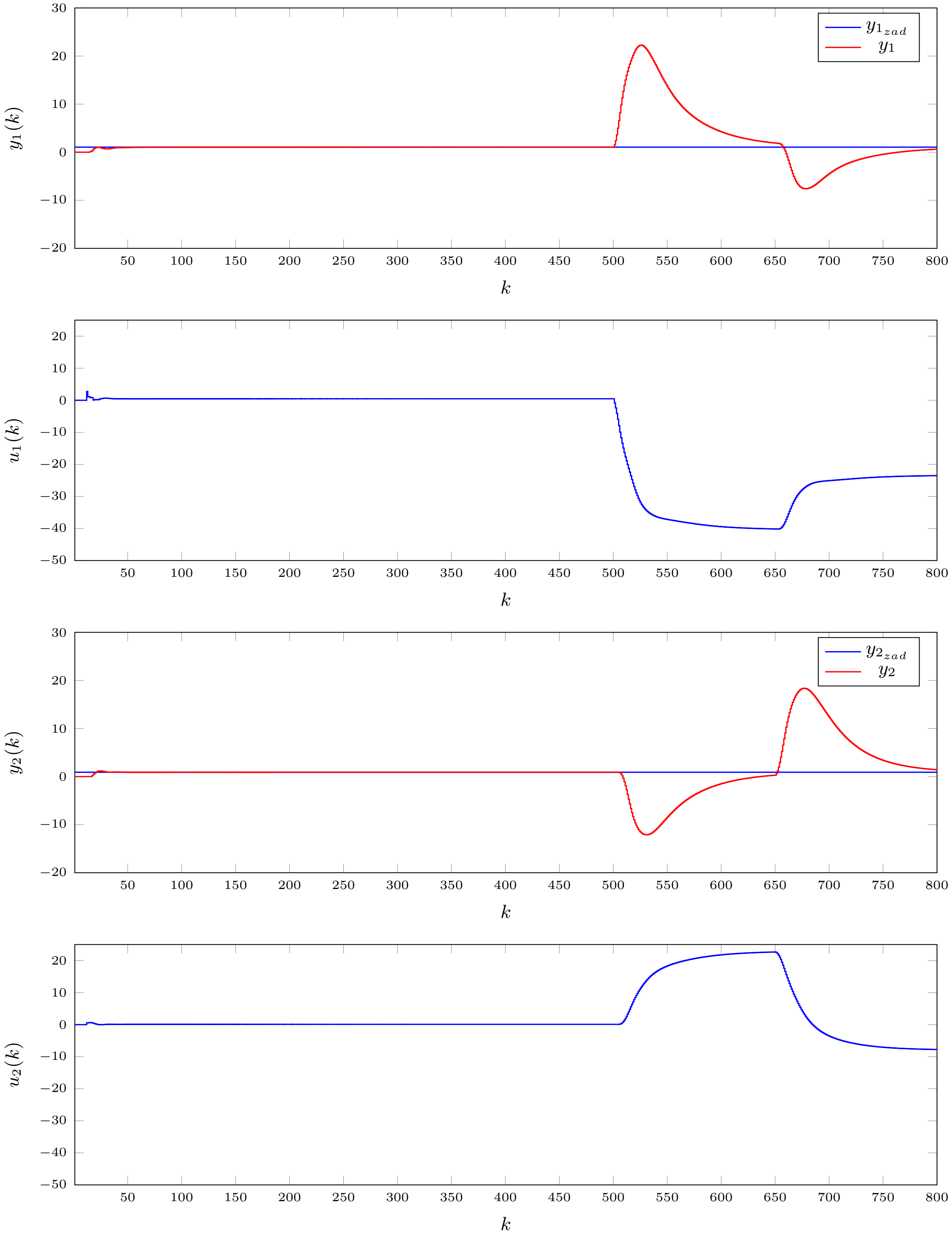
Rys. 6.5. Regulator DMC dla parametrów $D = N = N_u = 200 \quad \lambda = 5$ przy błędach pomiaru sygnałów wyjściowych



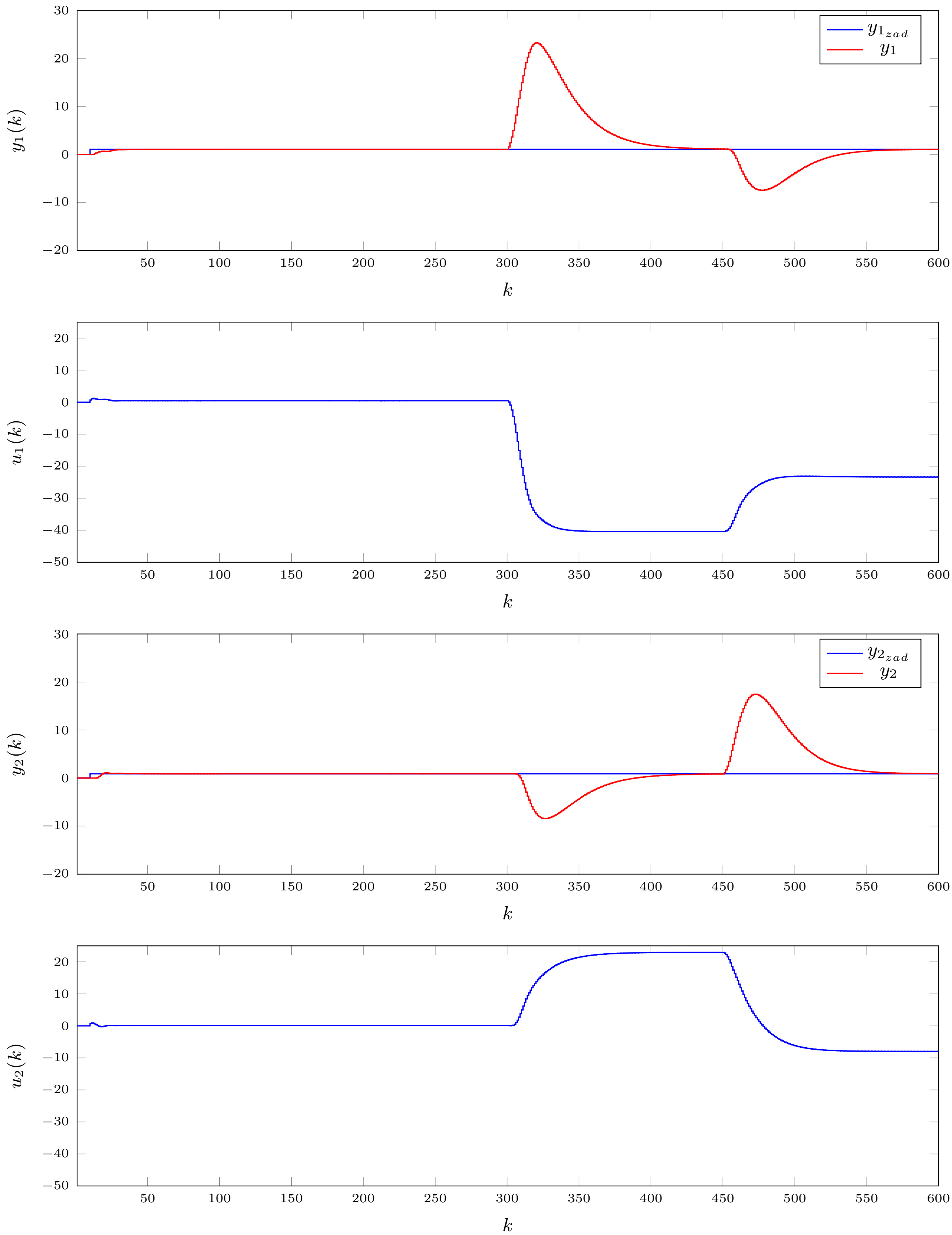
Rys. 6.6. Regulator DMC dla parametrów $D = N = N_u = 200 \quad \lambda = 10$ przy błędach pomiaru sygnałów wyjściowych

7. Zad. 7

Zbadana została odporność na zakłócenia niemierzalne dwóch regulatorów: najlepszego regulatora PID i najlepszego regulatora DMC. Zakłócenia niemierzalne zostały zaimplementowane jako nagły skok mierzonej wartości wyjść. Jak widać na rysunkach 7.1 i 7.2 pierwszy skok zakłócenia (sztuczne zwiększenie mierzonej wartości wyjścia pierwszego o 0,5) spowodował znaczną zmianę wartości i chwilowe zepsucie regulacji. Oba regulatory jednak dały radę wrócić do wartości zadanej aż do momentu drugiego skoku zakłócenia (sztuczne zwiększenie mierzonej wartości wyjścia drugiego o 0,3). Jednak i tym razem oba regulatory wróciły do wartości zadanej przy zakłóceniach utrzymujących się aż do końca czasu regulacji.



Rys. 7.1. Regulator PID dla nastaw $K_p = 1 \quad T_i = 15 \quad T_d = 0{,}8$ przy skokowych niemierzalnych zakłóceniach sygnałów wyjściowych



Rys. 7.2. Regulator DMC dla parametrów $D = N = N_u = 200 \quad \lambda = 1$ przy skokowych niemierzalnych zakłóceniach sygnałów wyjściowych