

Wydział Elektroniki i Technik Informacyjnych  
Politechnika Warszawska

Projektowanie układów sterowania  
(projekt grupowy)

Sprawozdanie z projektu i ćwiczenia laboratoryjnego  
nr 1, zadanie nr 4

Piotr Chachuła, Cezary Dudkiewicz, Piotr Roszkowski

Warszawa, 2019

# Spis treści

## I. Projekt

<b>1. Weryfikacja punktu pracy</b>	3
1.1. Opis postępowania	3
1.2. Wyniki	3
<b>2. Odpowiedzi skokowe</b>	4
2.1. Wyznaczanie odpowiedzi skokowych	4
2.2. Wyznaczanie charakterystyki statycznej procesu	4
2.3. Wzmocnienie statyczne	4
<b>3. Przekształcenie odpowiedzi skokowej</b>	7
<b>4. Regulator PID</b>	8
4.1. Algorytm działania	8
4.2. Ręczne strojenie regulatora PID	9
4.2.1. Wylczenie wstępnych parametrów regulatora	9
4.2.2. Zmniejszenie przeregulowania	12
4.2.3. Dalsze zmniejszenie przeregulowania	15
4.2.4. Przyspieszenie regulatora	18
4.2.5. Człon różniczkujący	21
4.2.6. Końcowe dostrojenie	24
4.3. Automatyczne strojenie regulatora PID	27
4.4. Obserwacje i wnioski	32
<b>5. Regulator DMC</b>	33
5.1. Algorytm działania	33
5.2. Ręczne strojenie regulatora DMC	34
5.3. Automatyczne strojenie regulatora DMC	51
5.4. Obserwacje i wnioski	56

## II. Laboratoria

Część I

**Projekt**

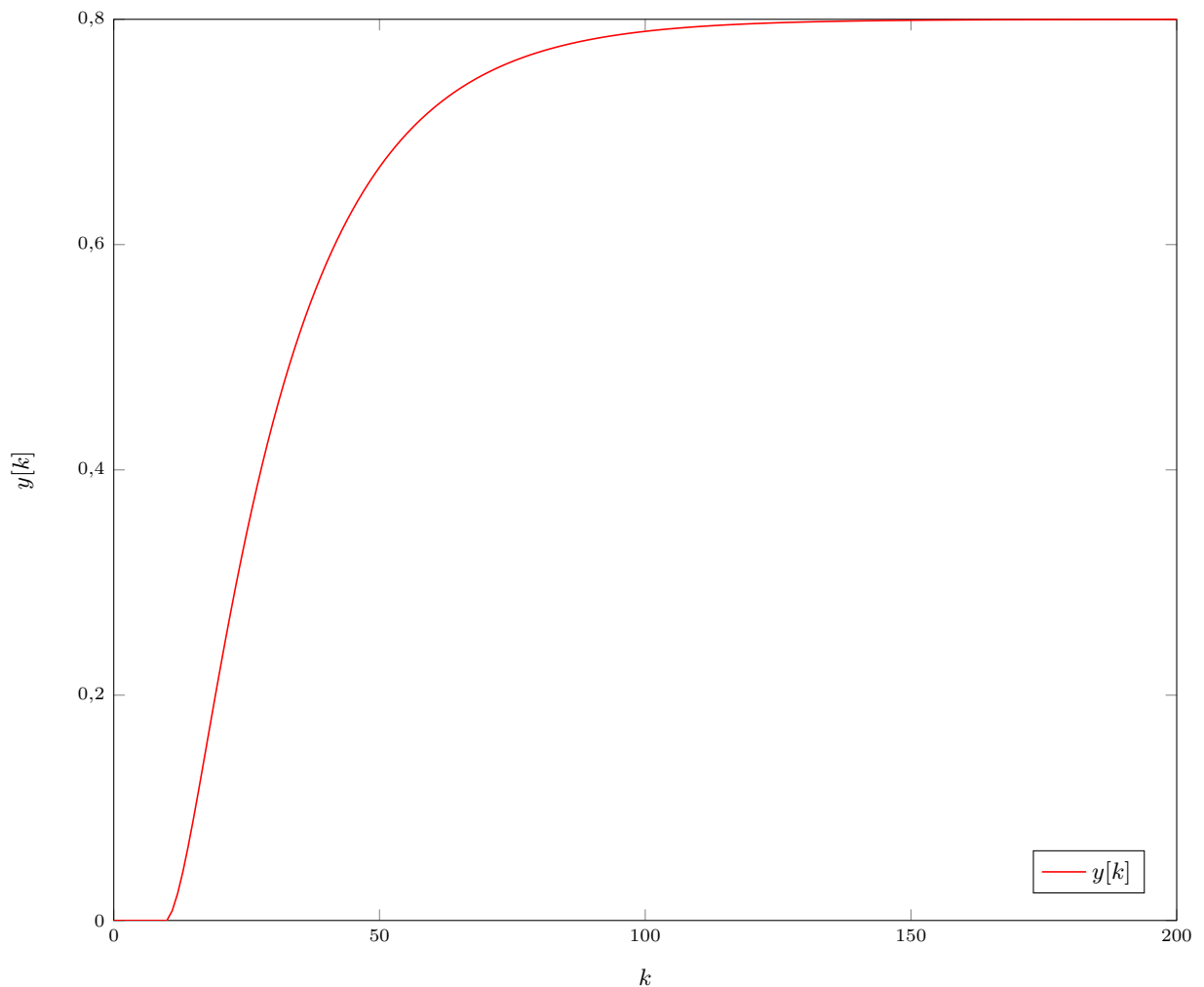
# 1. Weryfikacja punktu pracy

## 1.1. Opis postępowania

W celu sprawdzenia poprawności wartości sygnałów  $U_{pp}$  i  $Y_{pp}$  pobudzono obiekt sterowaniem o wartości  $U_{pp} = 2,0$  i sprawdzaniu czy stabilizuje się on w punkcie pracy  $Y_{pp} = 0,8$ . Do symulacji wyjścia obiektu użyto udostępnionej funkcji `symulacja_obiektu4Y`. Do testów napisano skrypt `PROJ1_1.m`. Wyniki przedstawiono poniżej.

## 1.2. Wyniki

Zgodnie z przewidywaniami wyjście obiektu ustaliło się na wartości  $Y_{pp} = 2,0$ . Punkt pracy ustalony jest więc poprawnie.



Rys. 1.1. Odpowiedź obiektu na sterowaniu  $U_{pp} = 0,8$

## 2. Odpowiedzi skokowe

### 2.1. Wyznaczanie odpowiedzi skokowych

W celu wyznaczenia odpowiedzi skokowej obiekt, znajdujący się w punkcie pracy (tzn.  $U_{pp} = 2,0, Y_{pp} = 0,8$ ) pobudzoną różną zmianą wartości sterowań. Rysunek 2.1 przedstawia odpowiedź obiektu na jego różne wartości.

### 2.2. Wyznaczanie charakterystyki statycznej procesu

Aby wyznaczyć charakterystykę statyczną procesu przeprowadzono analogiczne działania co w rozdziale 1. Tym razem przy użyciu skryptu PROJ1\_2.m dla wielu wartości  $U_{pp}$  wyznaczono odpowiadające im  $Y_{pp}$  oraz z ich pomocą utworzono wykres 2.2. Jak widać charakterystyka statyczna obiektu jest liniowa, a co za tym idzie obiekt jest liniowy.

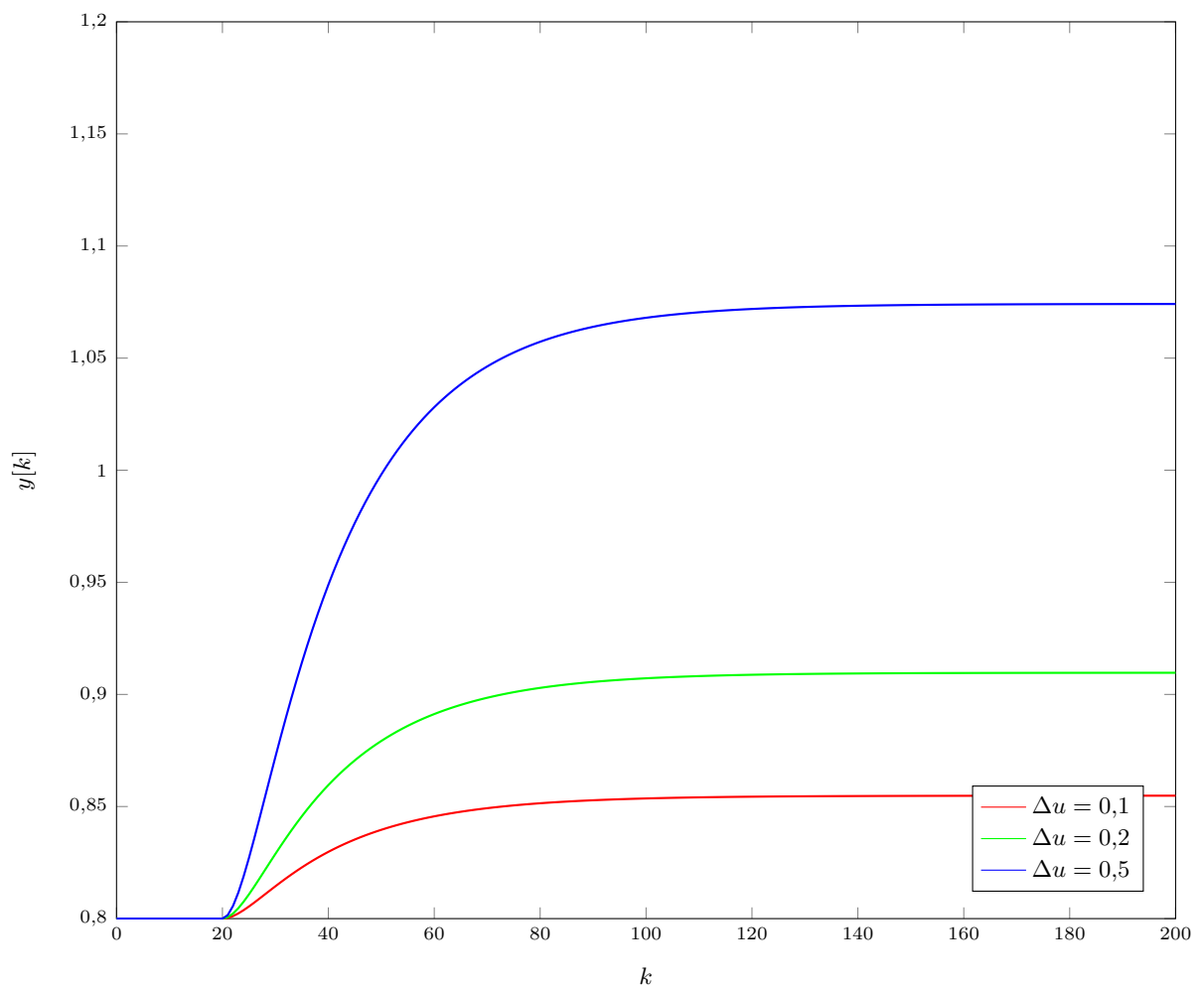
### 2.3. Wzmocnienie statyczne

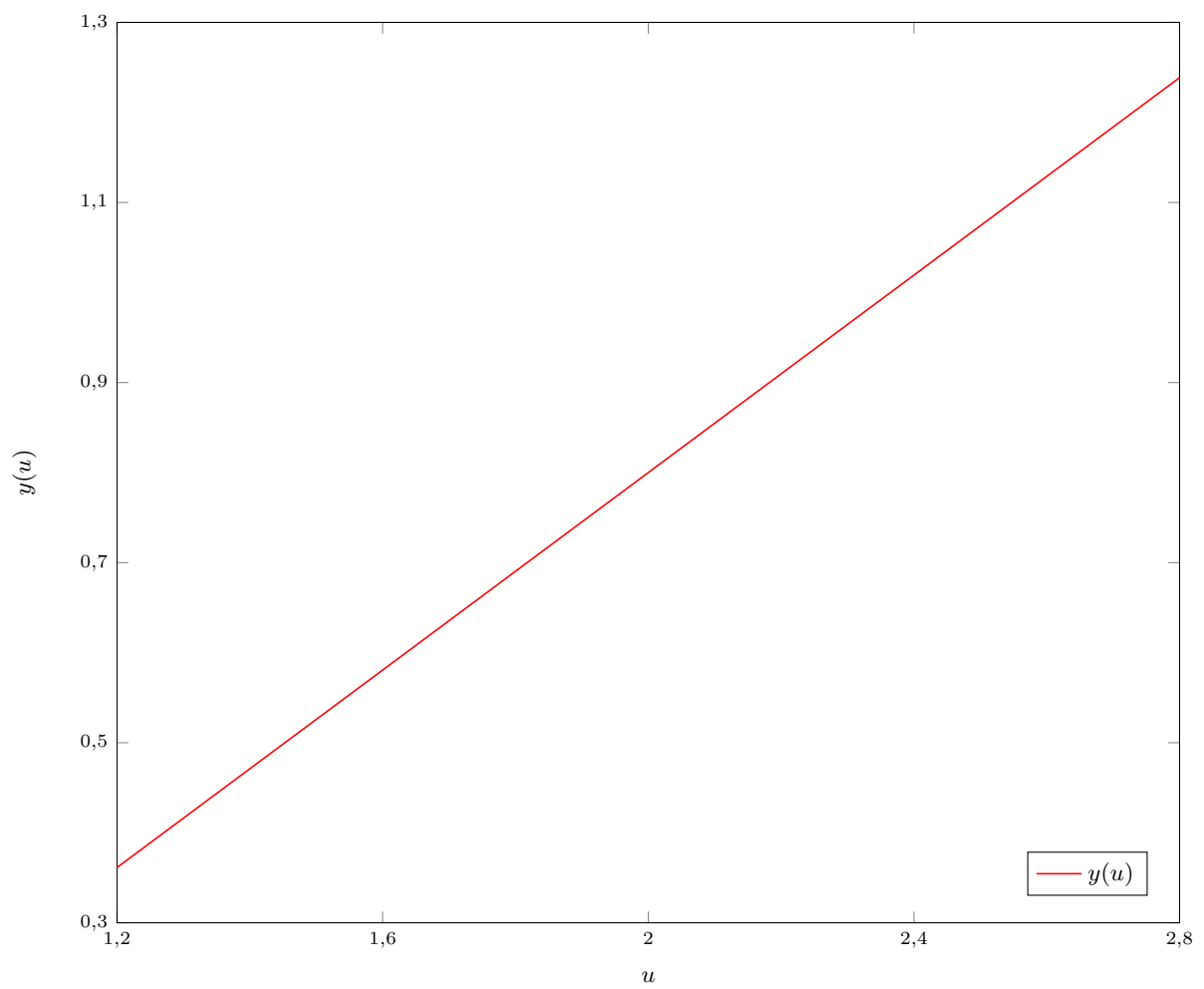
Wzmocnienie statyczne, czyli stosunek pomiędzy zmianą wartości wyjścia i zmianą wartości sterowania w stanie ustalonym. Aby ją wyznaczyć można na przykład znaleźć nachylenie charakterystyki statycznej do osi  $OX$ , czyli np.:

$$K_{\text{stat}} = \frac{y(U_{\text{max}}) - y(U_{\text{min}})}{U_{\text{max}} - U_{\text{min}}} \quad (2.1)$$

W przypadku tak wykreślonej charakterystyki, wzmocnienie statyczne jest równe tangensowi kąta  $\alpha$  pomiędzy prostą a osią  $OX$ .

$$K_{\text{stat}} = \frac{1,239 - 0,361}{2,8 - 1,2} \approx 0,549 \quad (2.2)$$

Rys. 2.1. Odpowiedzi procesu na skokowe zmiany sterowania w momencie  $k=11$

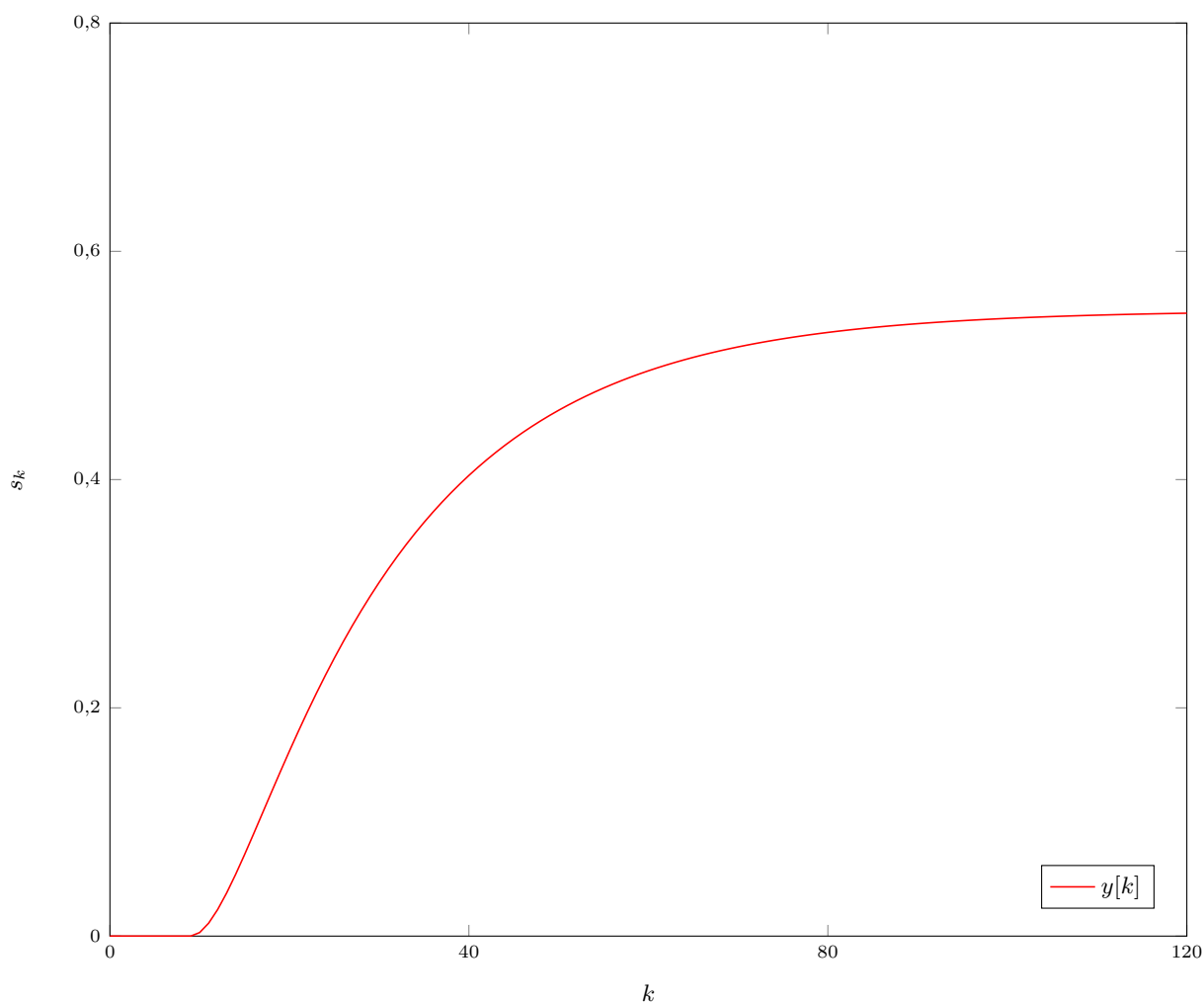
Rys. 2.2. Charakterystyka statyczna  $y(u)$  symulowanego procesu

### 3. Przekształcenie odpowiedzi skokowej

Aby uzyskać znormalizowaną odpowiedź skokową, należy przerzutować ją względem punktu pracy oraz wielkości skoku, a także przesunąć chwilę skoku sterowania do chwili  $k=0$  (z chwili  $k_{skok}$ ). Do tego celu można użyć wzoru:

$$s_i = \frac{s_{i+k_{skok}} - Y_{pp}}{\Delta U} \quad (3.1)$$

Wyznaczono ją przy użyciu skryptu `PROJ1_3.m` (dla odpowiedzi skokowej przy  $\Delta u = 0,5$ ). Następnie przycięto ją do miejsca w którym osiąga 0,995 swojej maksymalnej wartości. Długość tej odpowiedzi jest przyjętym horyzontem dynamiki tego obiektu i jest równy 120. Wynik działania przedstawiony jest na rysunku 3. Odpowiedź ta zostanie użyta do zaprojektowania regulatora DMC.



Rys. 3.1. Postać przekształconej odpowiedzi skokowej symulowanego obiektu ze zmianą sterowania w momencie  $k=0$



## 4. Regulator PID

### 4.1. Algorytm działania

Algorytm działania regulatora oraz implementacja została dobrze udokumentowana w pliku `doPID.m`. Listing jego części algorytmicznej przedstawiony jest poniżej:

```
function [ error ] = doPID( paras )    % Tylko dla auto

% Ustawiamy dlugosc symulacji
sim_len=1200;

% Tylko dla auto
K=paras(1);
Ti=paras(2);
Td=paras(3);

% Czas probkowania
T=1;

% Parametry wygodnego, dyskretnego PIDa
r0=K*(1+T/(2*Ti)+Td/T);
r1=K*(T/(2*Ti)-(2*Td/T)-1);
r2=K*Td/T;

% Inicjalizujemy macierze przechowujace zmienne
Y=zeros(sim_len,1);
U=zeros(sim_len,1);
e=zeros(sim_len,1);
y=zeros(sim_len,1);
u=zeros(sim_len,1);
Yzad=zeros(sim_len,1);
kk=linspace(1,sim_len,sim_len)';

% Ustalamy wartosci przed rozpoczeciem symulacji na wartosci w punktu pracy
Ypp=0.8;
Upp=2.0;
Y(1:11)=Ypp;
U(1:11)=Upp;

% Tworzymy horyzont wartosci zadanej
Yzad(1:29)=0.8;
Yzad(30:sim_len/3-1)=1.0;
Yzad(sim_len/3:2*sim_len/3-1)=0.6;
Yzad(2*sim_len/3:sim_len)=0.7;

%Rzutujemy ograniczenia sterowan wzgledem punktu pracy.
Umin=1.2;
Umax=2.8;
deltaumax=0.25;
umin=Umin-Upp;
umax=Umax-Upp;

for k=12:sim_len
    % Symulujemy wyjscie obiektu
    Y(k)=symulacja_obiektu4Y(U(k-10),U(k-11),Y(k-1),Y(k-2));
    % Rzutujemy wartosc wyjscia wzgledem punktu pracy
    y(k)=Y(k)-Ypp;
    % Liczymy uchyb i uaktualniamy wspolczynnik bledu
    e(k)=Yzad(k)-Y(k);
    error=error+e(k)^2;
    % Liczymy wartosc sterowania
    u_wyliczone=r2*e(k-2)+r1*e(k-1)+r0*e(k)+u(k-1);

    % Rzutowanie ograniczen na wartosc sterowania
    if u_wyliczone<umin
        u_wyliczone=umin;
    elseif u_wyliczone>umax
        u_wyliczone=umax;
    end
end
```

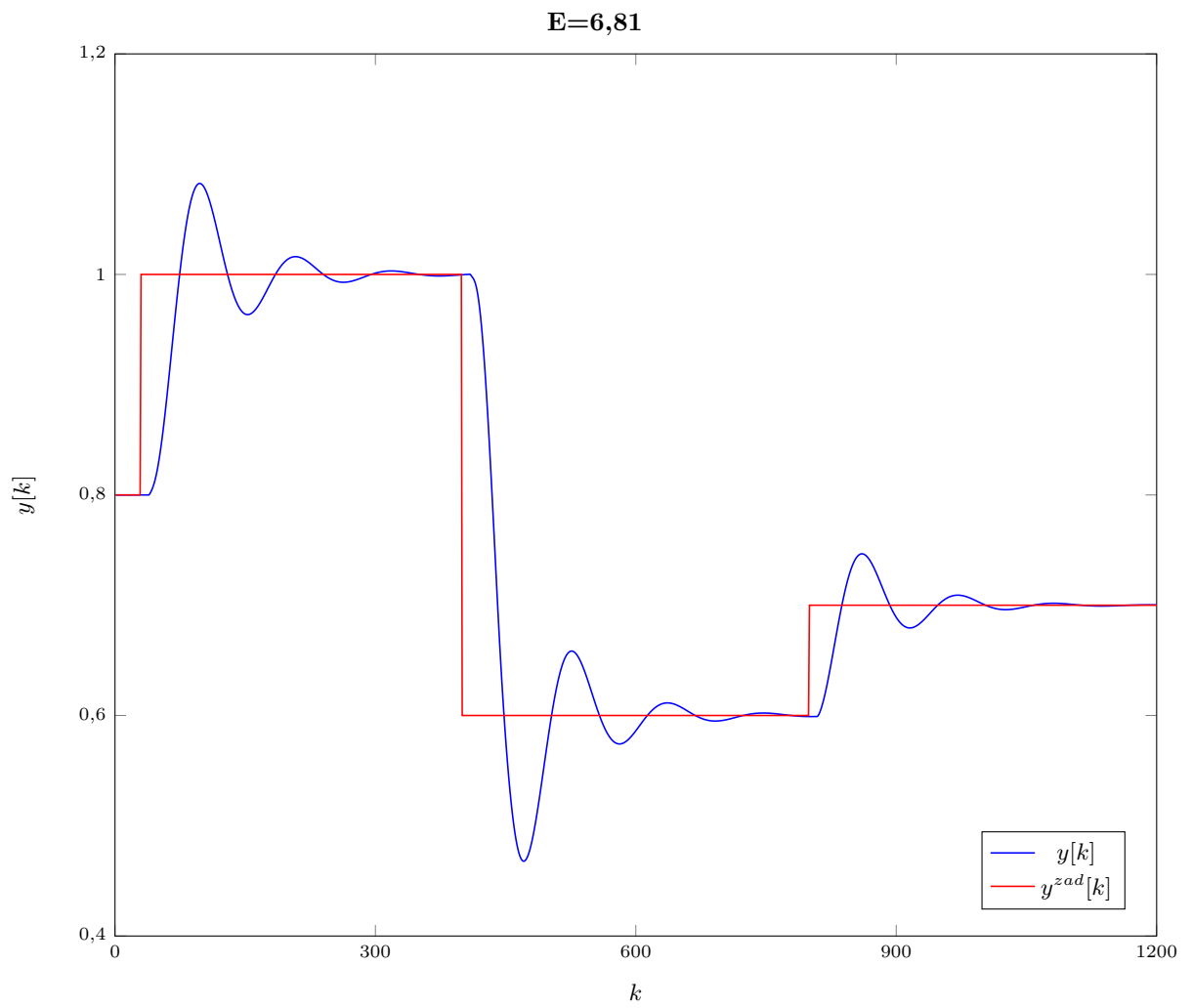
```
% Rzutowanie ograniczen na wartosc zmiany sterowania
if u_wyliczone-u(k-1)<=-deltaumax
    u_wyliczone=u(k-1)-deltaumax;
elseif u_wyliczone-u(k-1)>deltaumax
    u_wyliczone=u(k-1)+deltaumax;
end
% Rzutowanie sterowania wzgledem punktu pracy
u(k)=u_wyliczone;
U(k)=u_wyliczone+Upp;
end
```

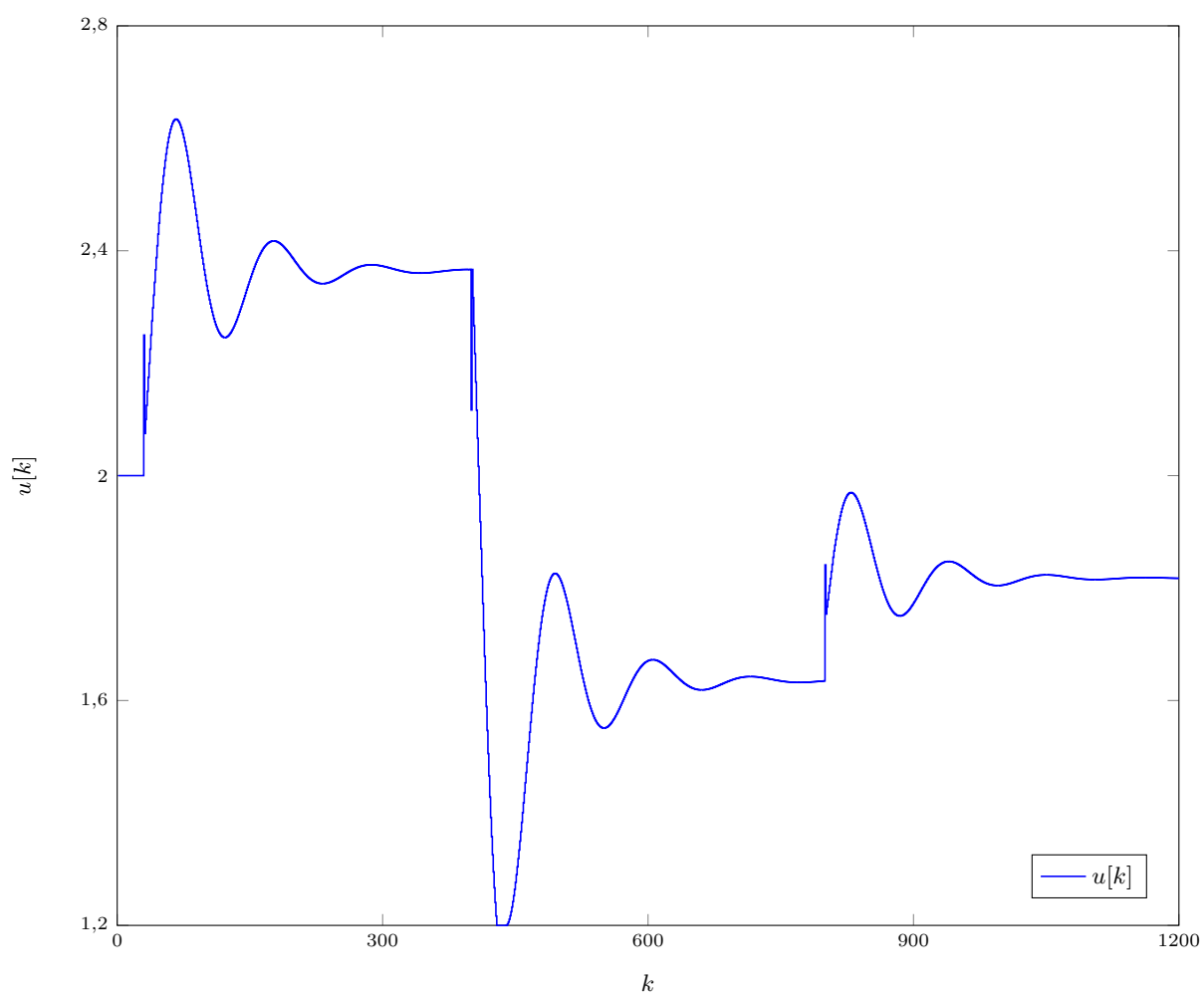
Listing 4.1. Implementacja regulatora PID

## 4.2. Ręczne strojenie regulatora PID

### 4.2.1. Wyliczenie wstępnych parametrów regulatora

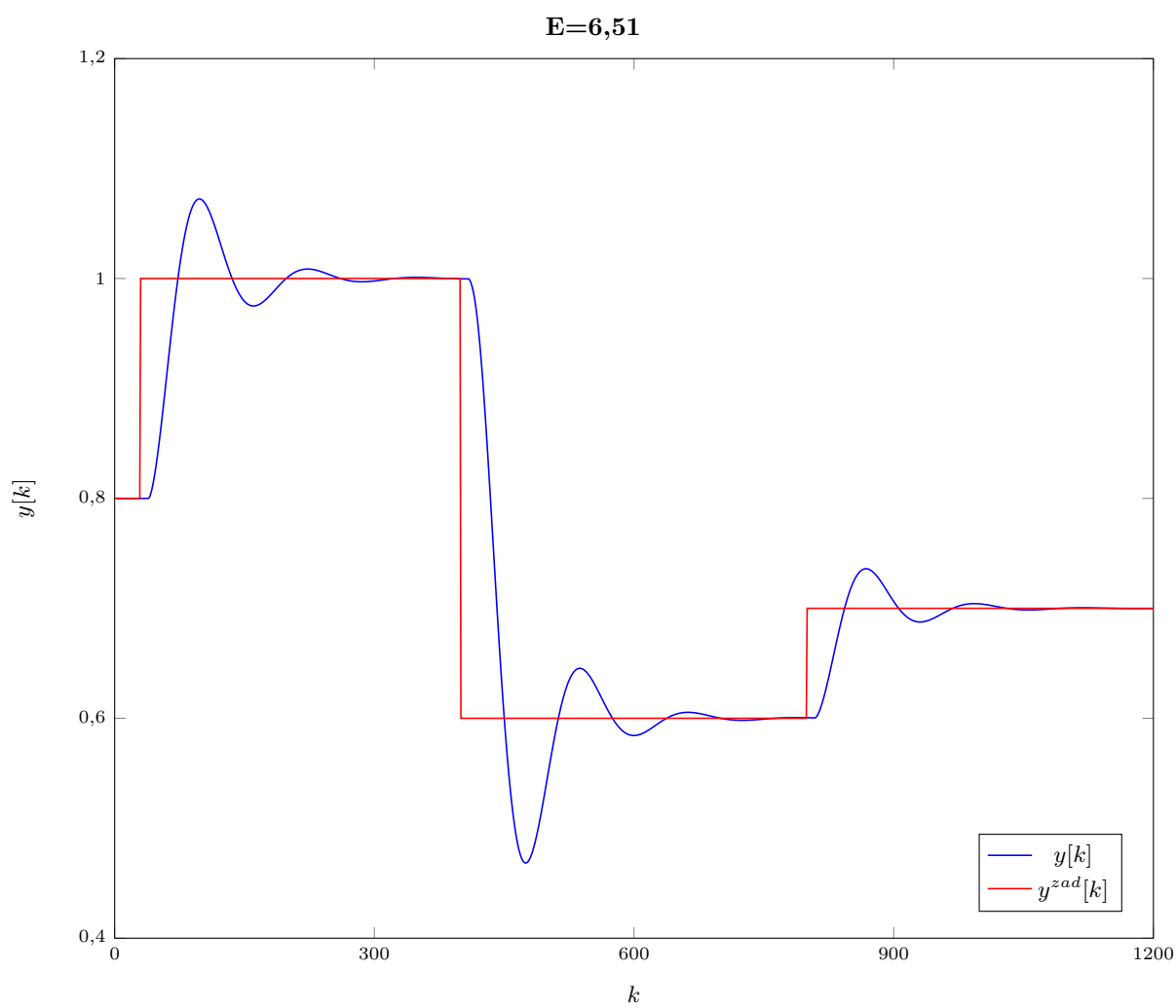
Celem naszej regulacji jest minimalizacja wskaźnika jakości, będącym sumą kwadratów uchybu regulacji w każdym punkcie. W celu dobrania wstępnych parametrów regulatora PID użyto metody eksperymentalnej; przeprowadzono dużą ilość symulacji dla arbitralnych wartości  $K_r$ ,  $T_i$ ,  $T_d$ . Spośród ponad 100 symulacji, wybrano te parametry, których wskaźnik jakości był najlepszy. Stało się to dla regulatora przedstawionego na rysunku 4.1.

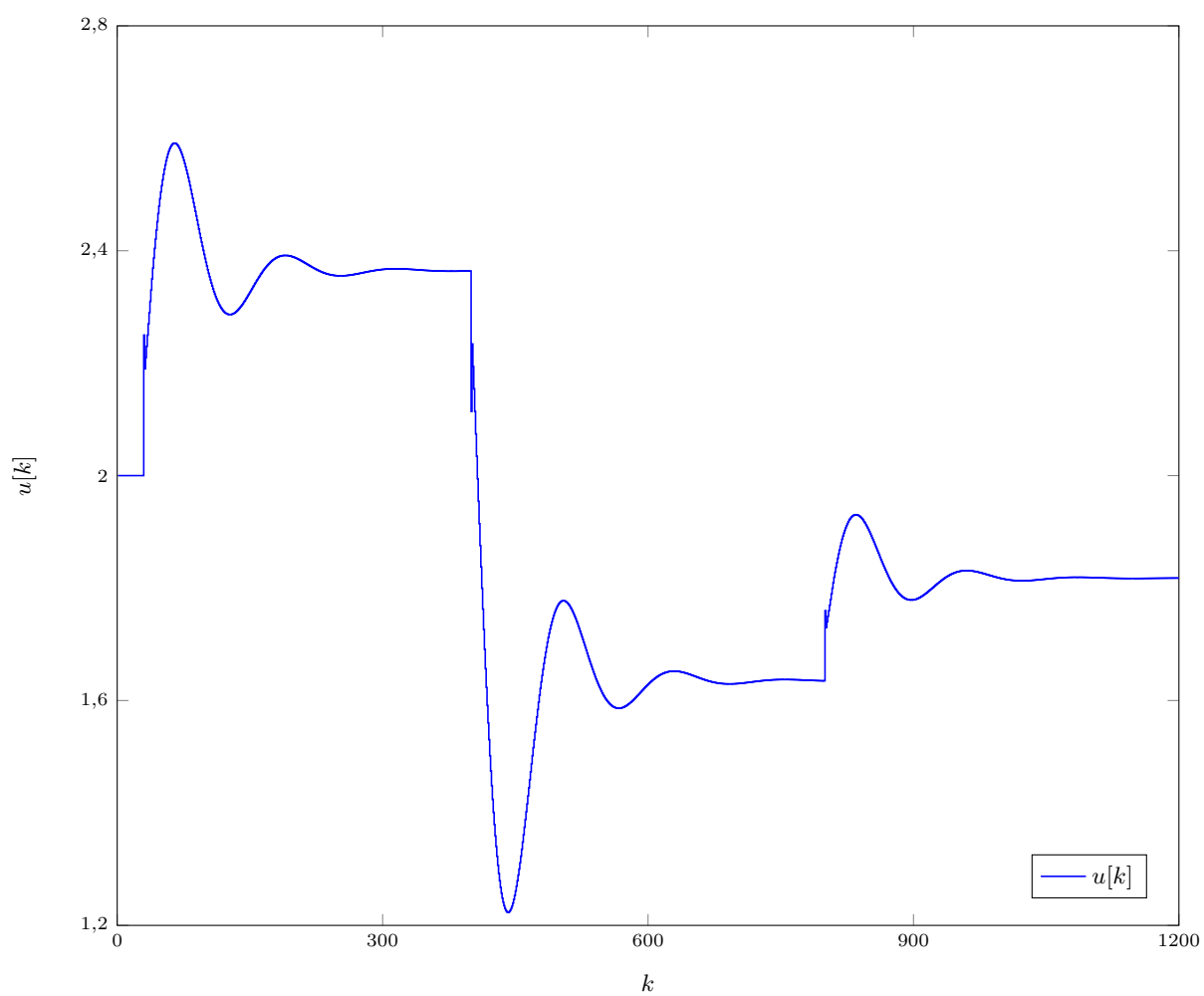
Rys. 4.1. Wyjście dla  $K_r=1$ ,  $T_i=4$ ,  $T_d=0,5$

Rys. 4.2. Sterowanie dla  $K_r=1$ ,  $T_i=4$ ,  $T_d=0,5$

#### 4.2.2. Zmniejszenie przeregulowania

W celu zmniejszenia znacznego przeregulowania występującego w regulatorze zmniejszono jego wzmocnienie do  $K_r=0,8$  oraz dodatkowo zmieniono parametr członu różniczkującego do  $T_d=0,25$ . Wyniki symulacji dla zmienionego regulatora przedstawone zostały na rysunku 4.3.

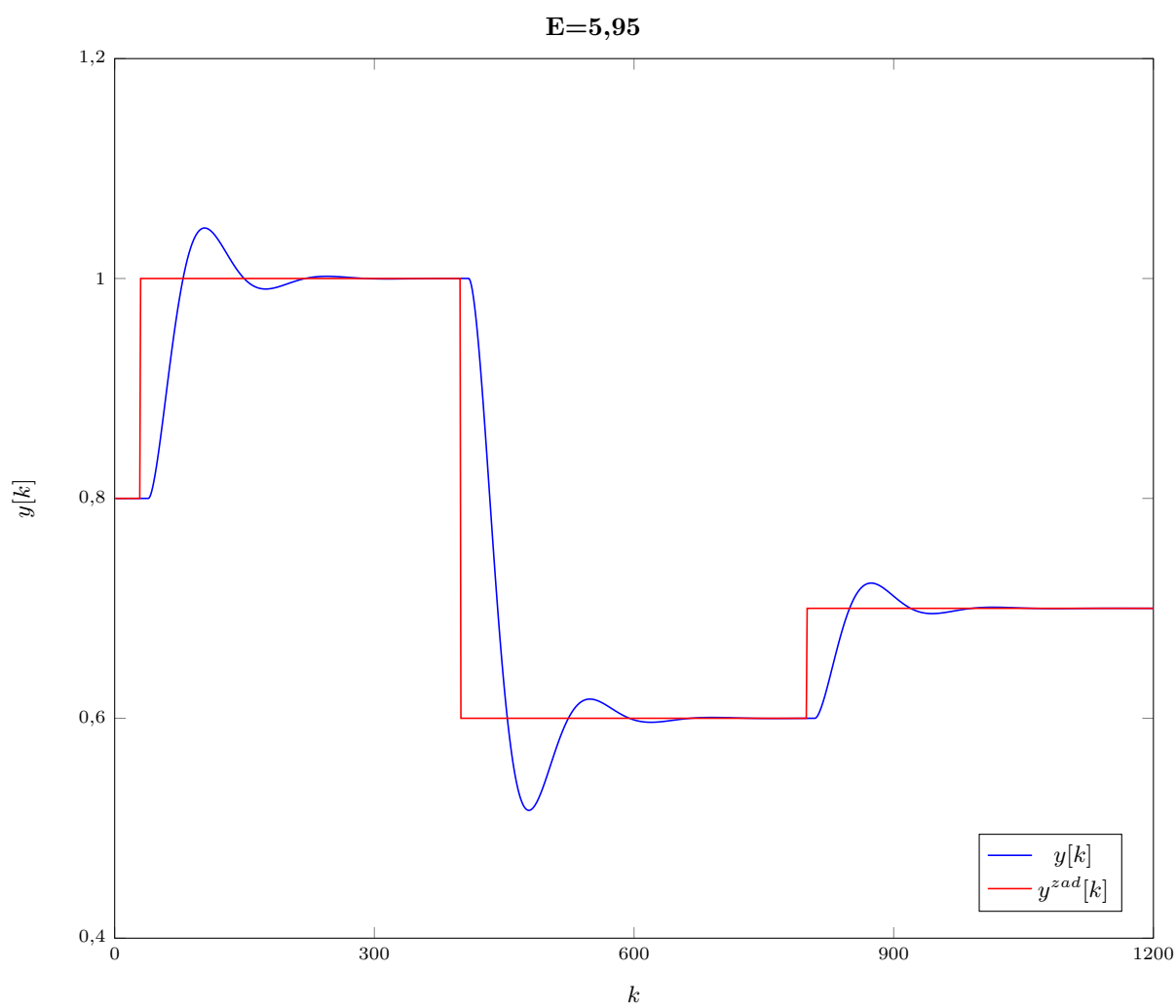
Rys. 4.3. Wyjście dla  $K_r=0,8$ ,  $T_i=4$ ,  $T_d=0,25$

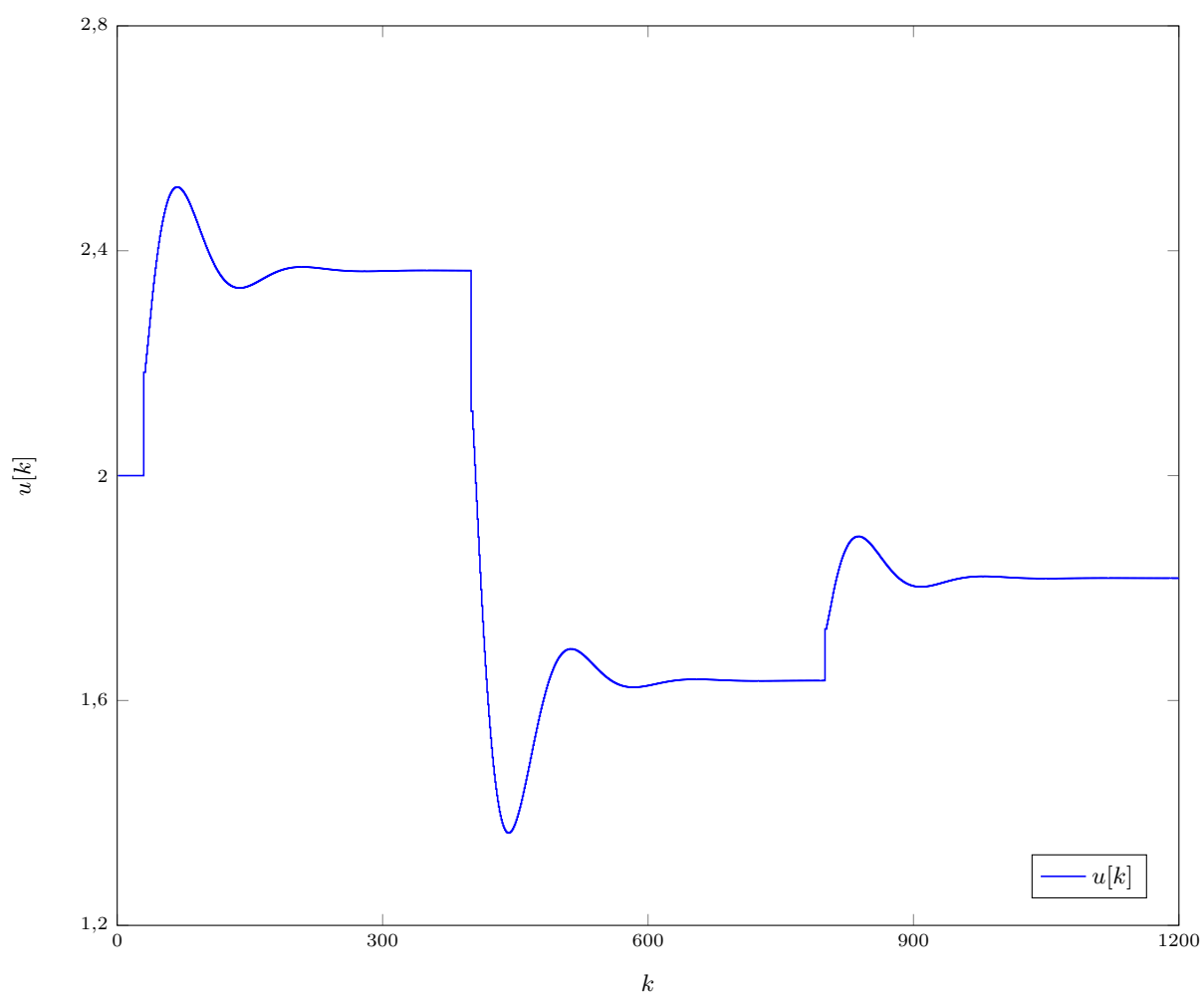
Rys. 4.4. Sterowanie dla  $K_r=0,8$ ,  $T_i=4$ ,  $T_d=0,25$

#### 4.2.3. Dalsze zmniejszenie przeregulowania

Przeregulowanie ciągle występuje w dość znacznym stopniu, dlatego w następnym kroku zmniejszono całkowanie do  $T_i=5$  oraz dodatkowo zmniejszono różniczkowanie do  $T_d=0,05$ . Wyniki symulacji dla zmienionego regulatora przedstawione zostały na rysunku 4.5.

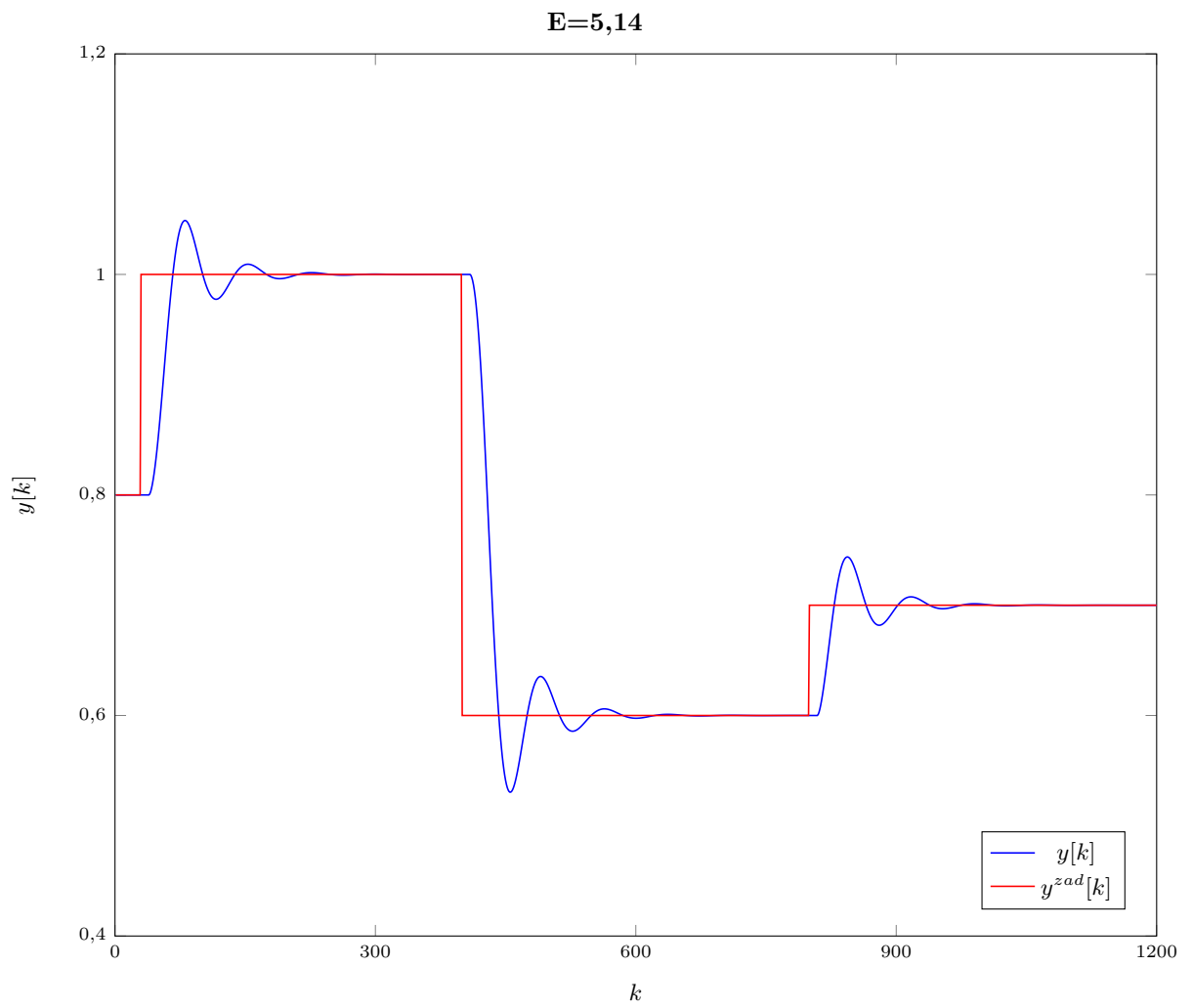


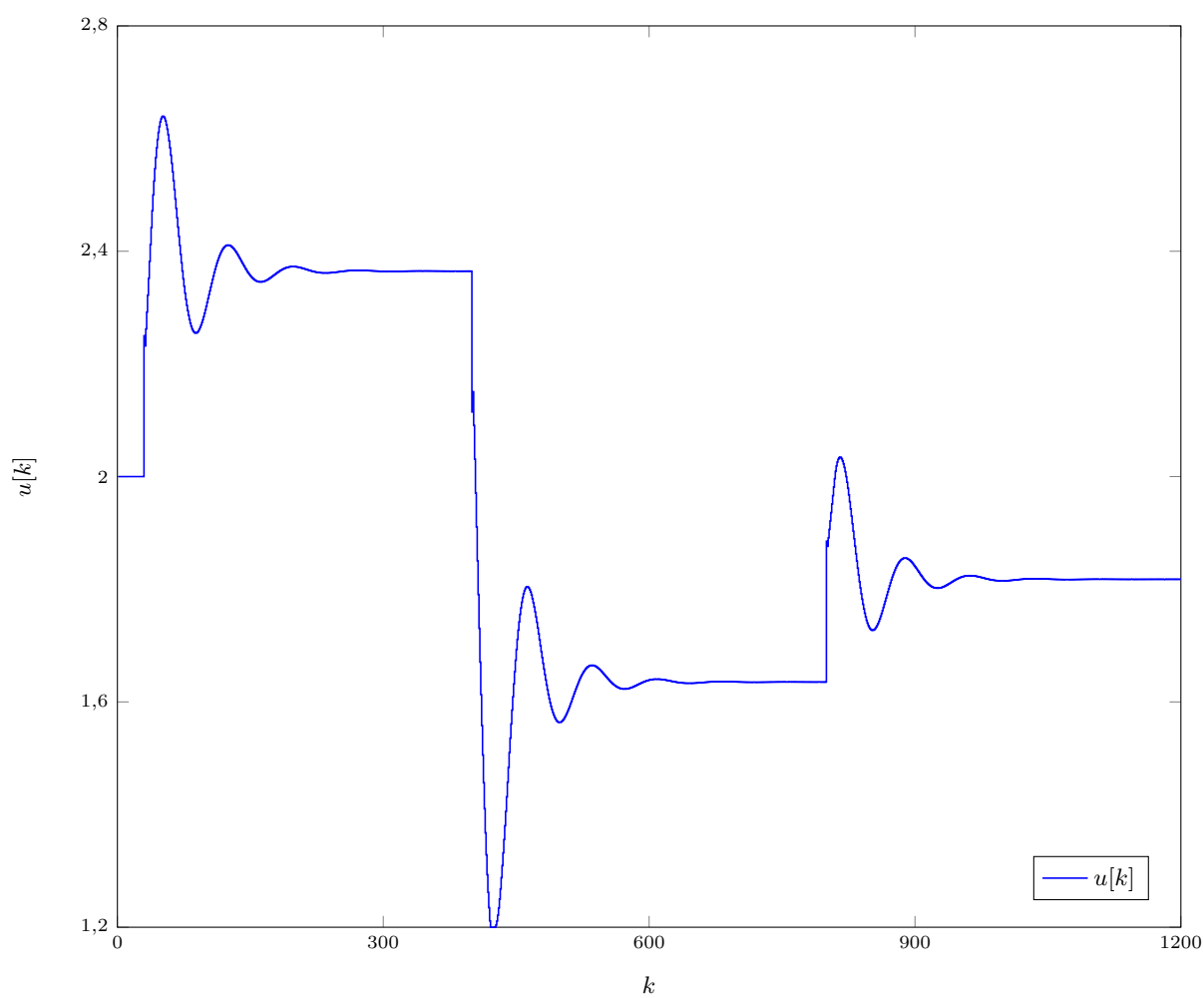
Rys. 4.5. Wyjście dla  $K_r=0,8$ ,  $T_i=5$ ,  $T_d=0,05$

Rys. 4.6. Sterowanie dla  $K_r=0,8$ ,  $T_i=5$ ,  $T_d=0,95$

**4.2.4. Przyspieszenie regulatora**

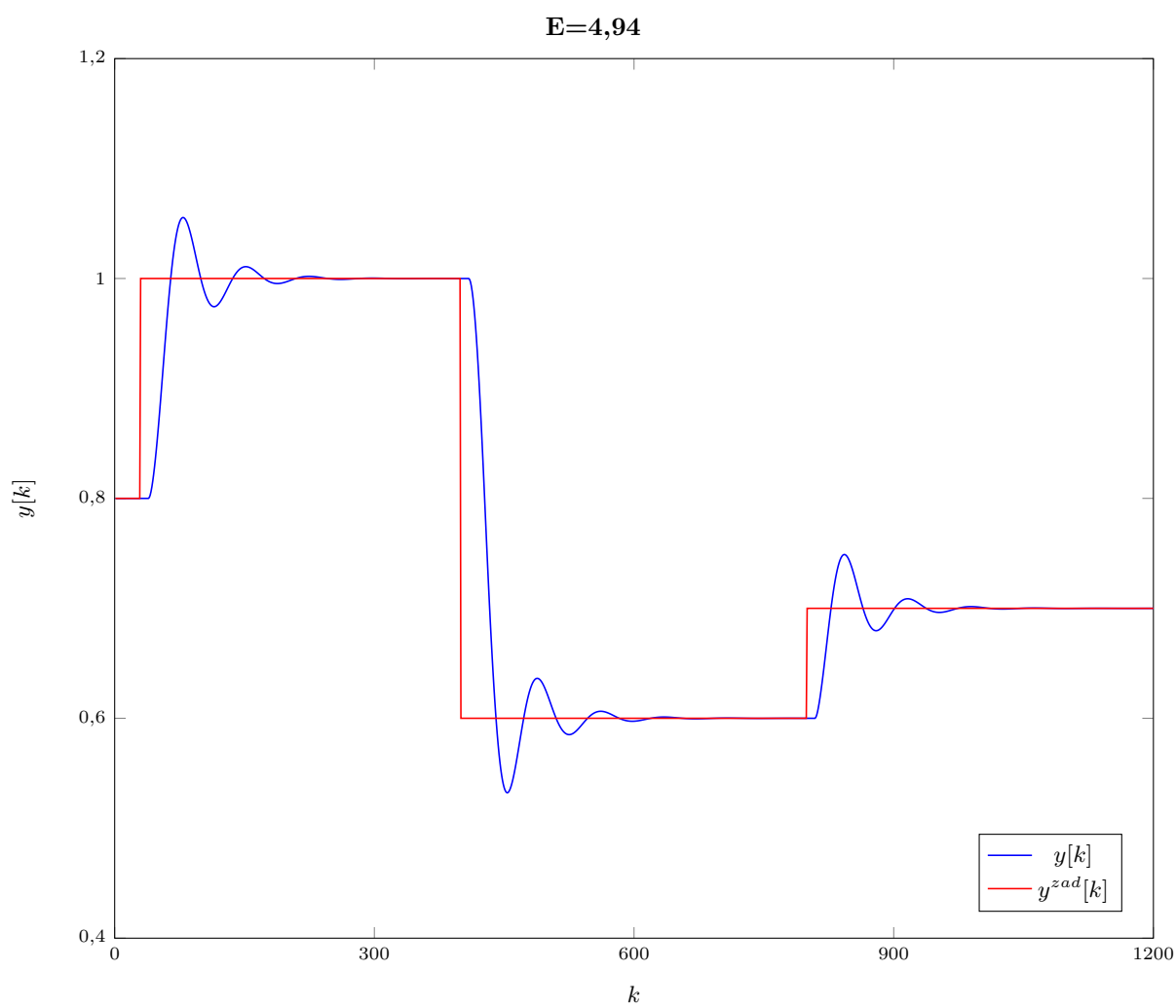
W celu zwiększenia szybkości regulatora zwiększono jego wzmocnienie do  $K_r=2,4$  oraz w ramach kompensacji zmniejszono całkowanie do  $T_i=8$ . Wyniki symulacji dla zmienionego regulatora przedstawione zostały na rysunku 4.7.

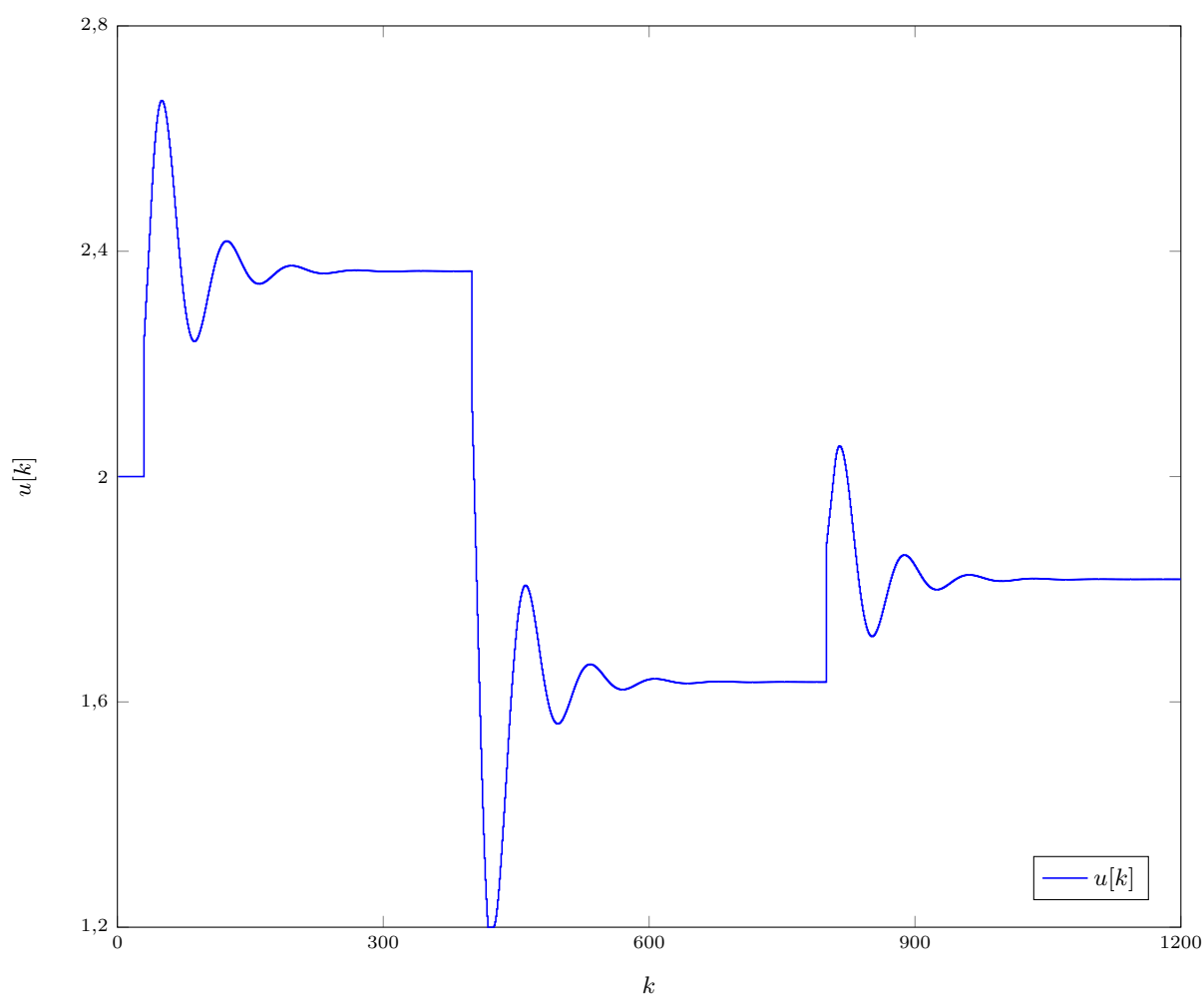
Rys. 4.7. Wyjście dla  $K_r=0,8$ ,  $T_i=5$ ,  $T_d=0,05$

Rys. 4.8. Sterowanie dla  $K_r=0.8$ ,  $T_i=5$ ,  $T_d=0.05$

#### 4.2.5. Człon różniczkujący

Podczas wczesniejszego strojenia zauważono fakt, iż zmniejszenie członu różniczkującego pozytywnie wpływa na jakość regulacji. Postanowiono więc znacznie go zmniejszyć, do wartości  $T_d=0,0005$ . Wyniki symulacji dla zmienionego regulatora przedstawione zostały na rysunku 4.11.

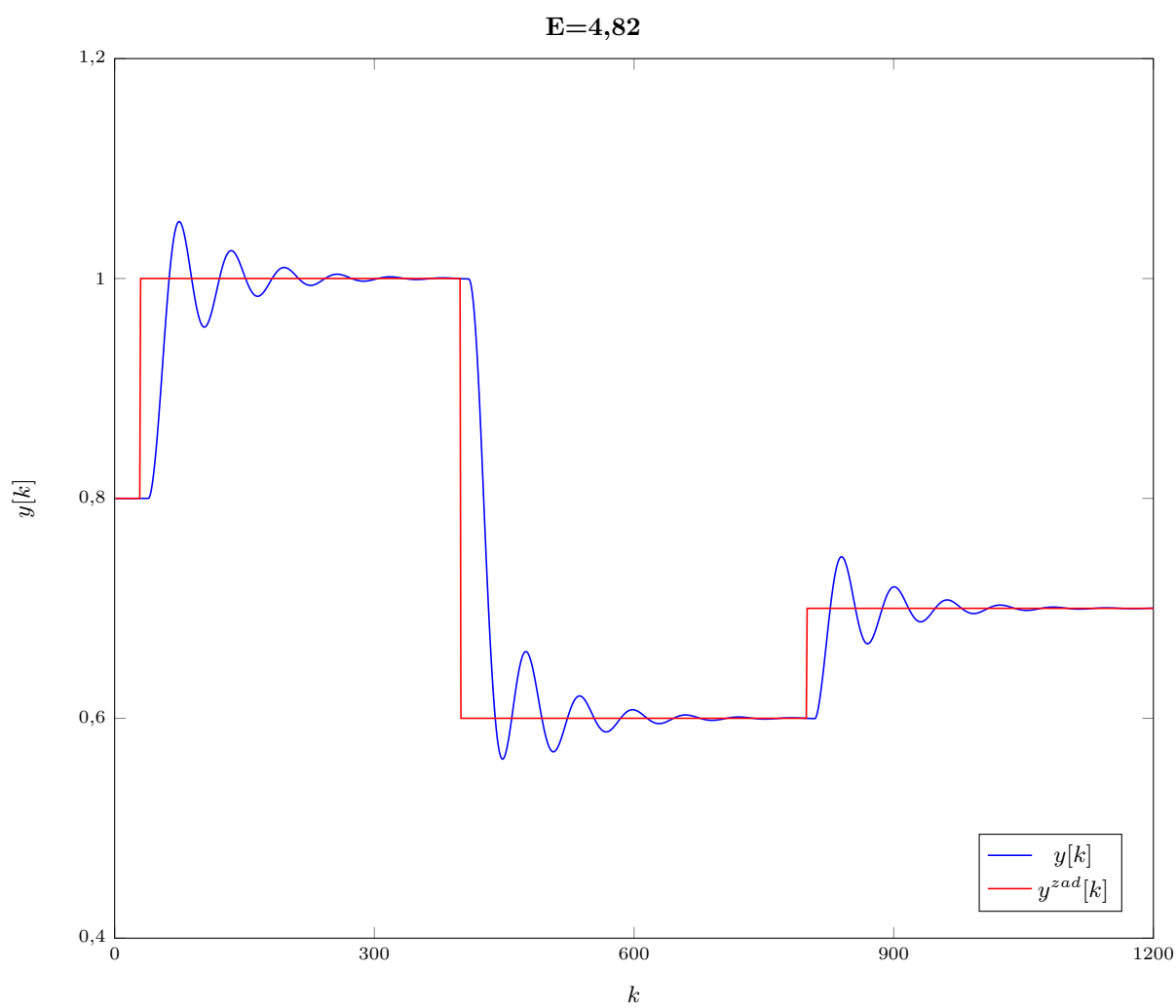
Rys. 4.9. Wyjście dla  $K_r=0,8$ ,  $T_i=5$ ,  $T_d=0,05$

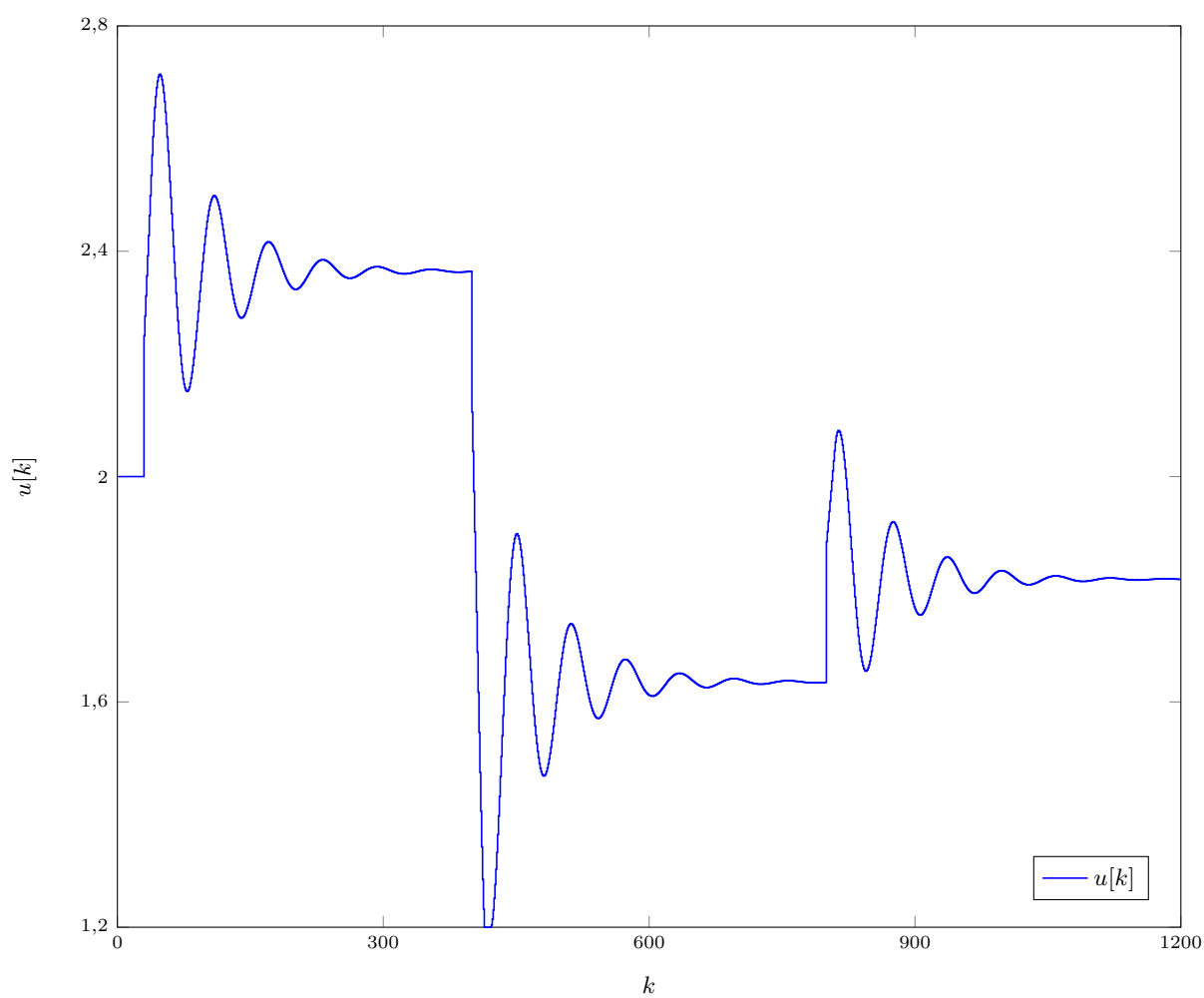
Rys. 4.10. Sterowanie dla  $K_r=0,8$ ,  $T_i=5$ ,  $T_d=0,05$



#### 4.2.6. Końcowe dostrojenie

Analogicznie jak w kroku 4.2.1 przeprowadzono dużą liczbę symulacji dla wielu kombinacji parameterów, nieznacznie oddalonych od obecnych. W ten sposób znaleziono ich optymalną wartość równą:  $K_r=3,3$ ,  $T_i=9,25$ ,  $T_d=0,0005$ . Wyniki symulacji dla takiego regulatora zostały przedstawione na rysunku ??.

Rys. 4.11. Wyjście dla  $K_r=0,8$ ,  $T_i=5$ ,  $T_d=0,05$

Rys. 4.12. Sterowanie dla  $K_r=0,8$ ,  $T_i=5$ ,  $T_d=0,05$

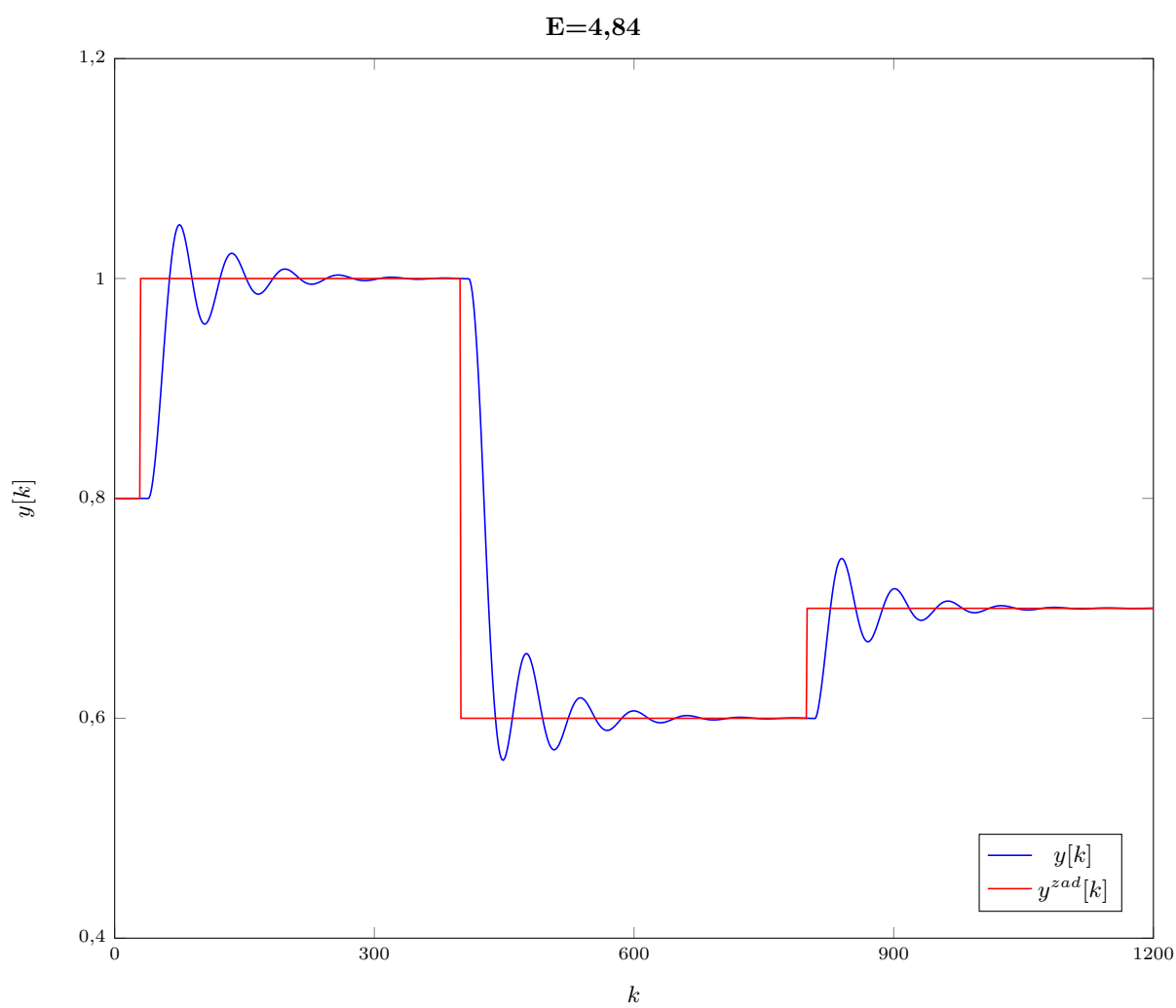
### 4.3. Automatyczne strojenie regulatora PID

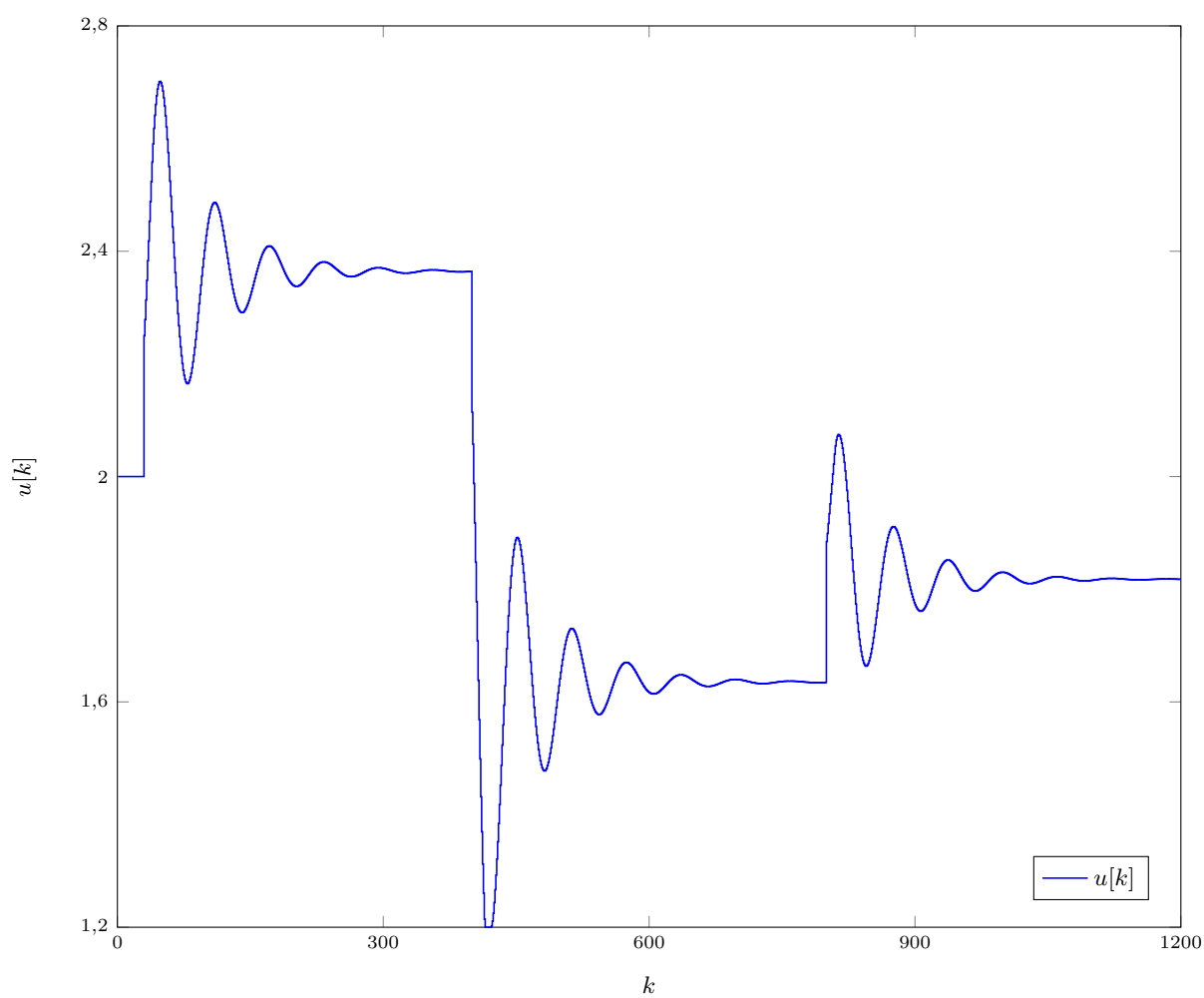
Do automatycznego strojenia regulatora PID użyto już wcześniej używanej funkcji `doPID.m`, której użyto jako funkcji liczącej wartość wskaźnik jakości (funkcja ją zwraca) w matlabowskim algorytmie `fmincon`, której użyto w skrypcie `PIDOptimizer.m`. Przeprowadzono dwie próby: dla małych wartości parametrów początkowych oraz dla dużych wartości kryteriów początkowych, co widać w listingu ?? poniżej.

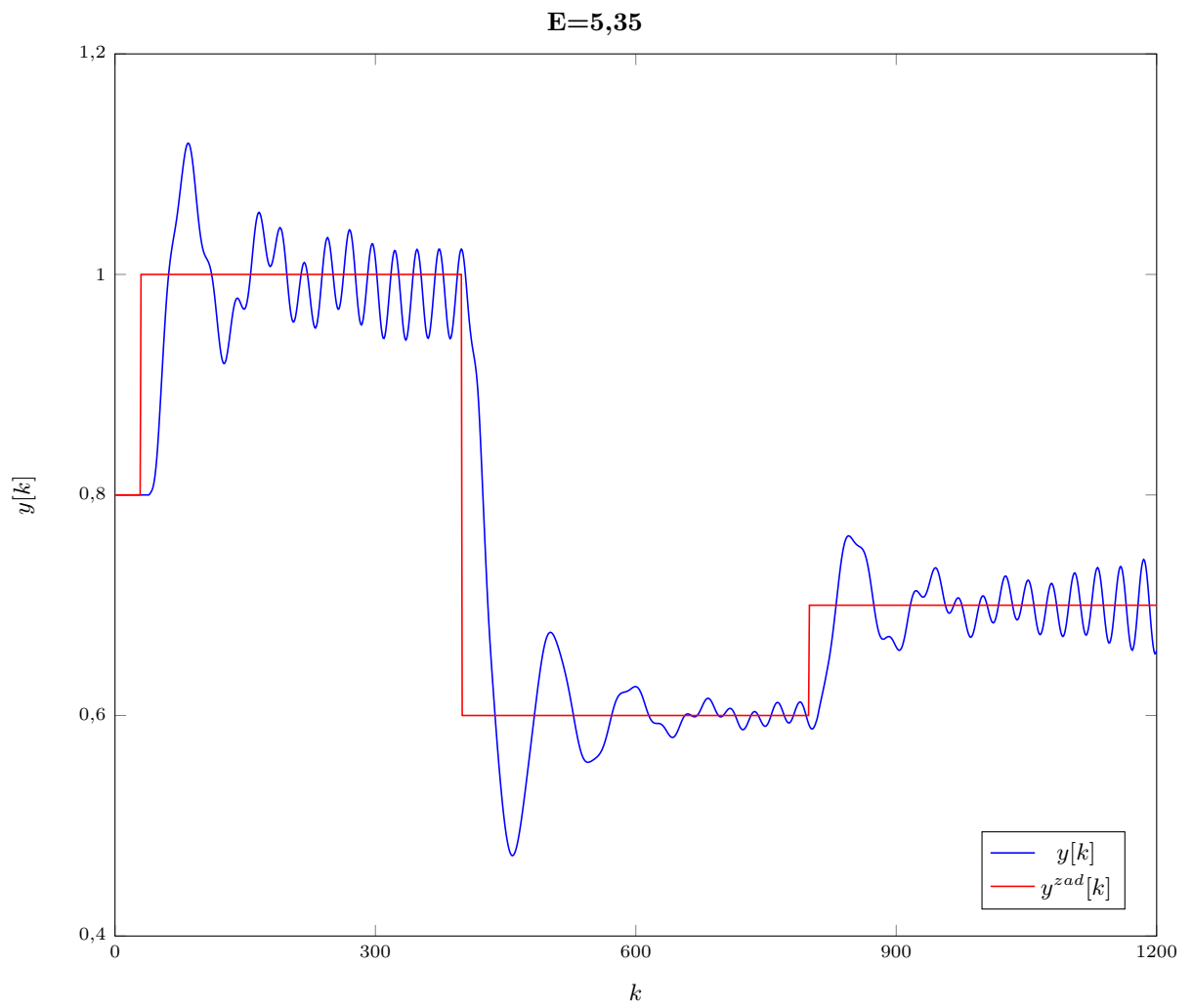
```
x01=[1,1,1];  
x02=[20,20,20];  
lb=[0.01,0.01,0.01];  
ub=[50,50,50];  
paras=fmincon(@doPID,x0,[],[],[],[],lb,ub);  
\label{LIST_1}
```

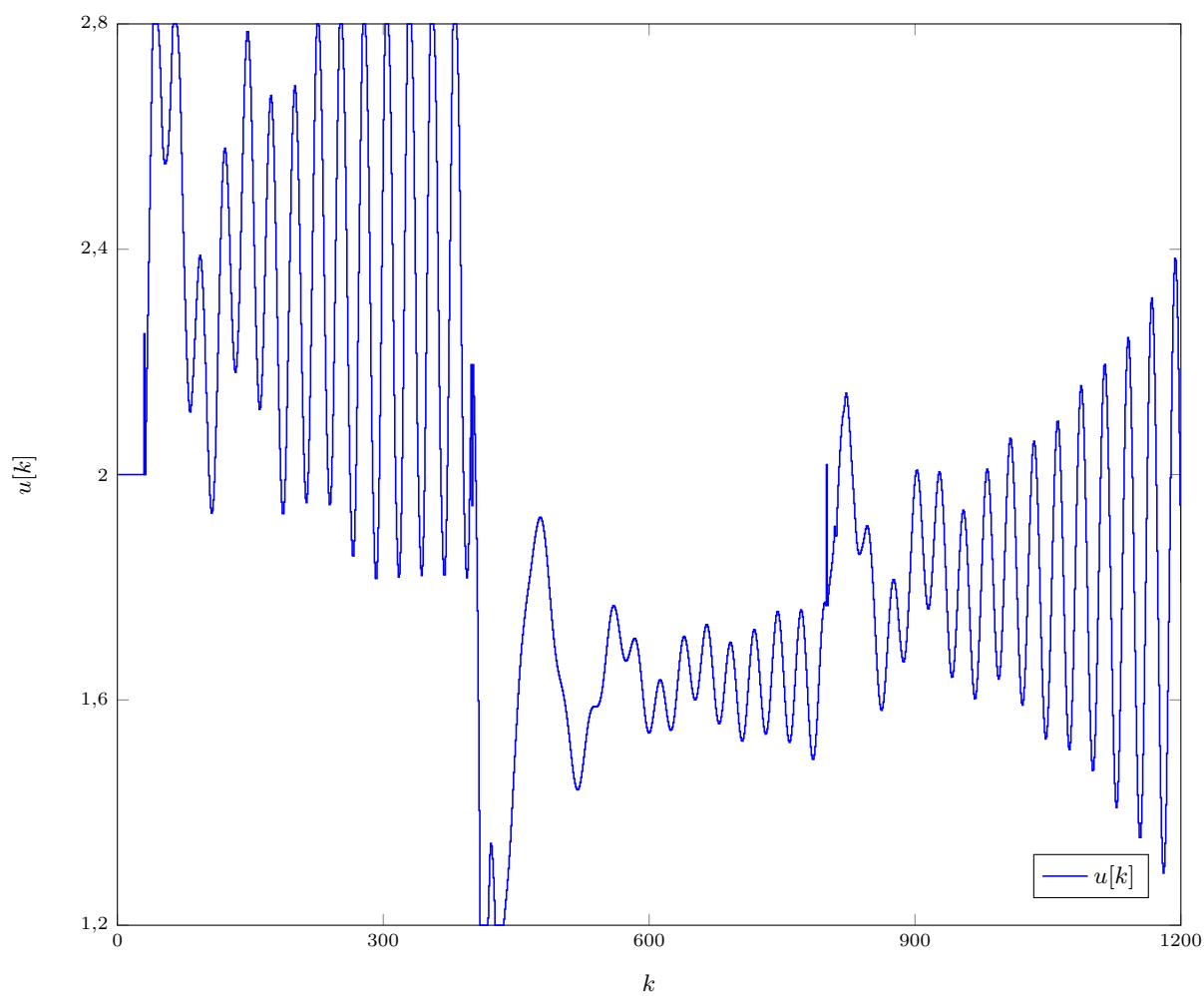
Listing 4.2. Implementacja automatycznego strojenia PID

Wyniki pracy dla pierwszego regulatora przedstawiono na rysunku 5.49, a drugiego na rysunku 5.51.

Rys. 4.13. Wyjście dla  $K_r=3,25$ ,  $T_i=9,3$ ,  $T_d=0,005$

Rys. 4.14. Wyjście dla  $K_r=3,25$ ,  $T_i=9,3$ ,  $T_d=0,005$

Rys. 4.15. Wyjście dla  $K_r=2,74$ ,  $T_i=2,9$ ,  $T_d=10,95$

Rys. 4.16. Sterowanie dla  $K_r=2,74$ ,  $T_i=2,9$ ,  $T_d=10,95$



#### 4.4. Obserwacje i wnioski

Ręczne dostrojenie regulatora PID okazało się odrobinę lepsze, być może z uwagi na bardzo małą niską wartość parametru  $T_d$ , leżącego poza obszarem poszukiwań algorytmu. Z reguły jednak niemożliwe jest tak dobre ręczne nastrojenie, bez użycia znaczącej ilości czasu, automatyczny optymalizator gwarantuje praktycznie najlepsze rozwiązanie, jednak nie jest możliwe jego zastosowanie dla obiektów rzeczywistych poza symulacją, najwyżej co do wstępnej estymacji parametrów przy użyciu zebranego modelu obiektu.

## 5. Regulator DMC

### 5.1. Algorytm działania

Algorytm działania regulatora oraz implementacja została dobrze udokumentowana w pliku `doDMC.m`. Listing jego części algorytmicznej przedstawiony jest poniżej:

```
function [ error ] = doDMC( paras )

% Zaokrąglamy wartości dla algorytmów optymalizacji
N=round(paras(1));
Nu=round(paras(2));
lambda=paras(3);

% Załadowanie odpowiedzi skokowej obiektu
load StepResponse.mat;

% Ucinamy moment 0 żeby łatwiej było operować
stepResp=stepResp(2:end);

% Pobieramy horyzont dynamiki obiektu
D=length(stepResp);

% Inicjalizacja współczynnika jakości
error=0;

% Inicjalizacja macierzy M
M=zeros(N,Nu);
for j=1:Nu
    for i=j:N
        M(i,j)=stepResp(i-j+1);
    end
end

% Inicjalizacja macierzy Mp
Mp=zeros(N,D-1);
for i=1:N
    for j=1:D-1
        if (i+j)<=D
            Mp(i,j)=stepResp(i+j)-stepResp(j);
        else
            Mp(i,j)=stepResp(end)-stepResp(j);
        end
    end
end

% Liczymy macierze K, Ke, Ku
K=(M'*M+lambda*lambda*eye(Nu))^( -1)*M';
Ke=sum(K(1,:));
Ku=K(1,:)*Mp;

% Inicjalizujemy macierze przechowujące zmienne
sim_len=1200;
dUp=zeros(D-1,1);
Y=zeros(sim_len,1);
U=zeros(sim_len,1);
du=zeros(sim_len,1);
e=zeros(sim_len,1);
y=zeros(sim_len,1);
u=zeros(sim_len,1);
Yzad=zeros(sim_len,1);
kk=linspace(1,sim_len,sim_len)';

% Tworzymy horyzont wartości zadanej
Yzad(1:D+11)=0.8;
Yzad(D+12:sim_len/3-1)=1.0;
Yzad(sim_len/3:2*sim_len/3-1)=0.6;
Yzad(2*sim_len/3:sim_len)=0.7;

% Ustalamy wartości przed rozpoczęciem symulacji na wartości w punktu pracy
```

```

Ypp=0.8;
Upp=2.0;
Y(1:D+11)=0.8;
U(1:D+11)=2.0;

% Wprowadzamy ograniczenia
Umin=1.2;
Umax=2.8;
deltaumax=0.25;
deltaumin=-0.25;
umin=Umin-Upp;
umax=Umax-Upp;

% Początek symulacji - zaczynamy w tej chwili w celu uproszczenia
% pozyskiwania wektora dUp
for k=D+12:sim_len
    % Symulujemy wyjście obiektu
    Y(k)=symulacja_obiektu4Y(U(k-10),U(k-11),Y(k-1),Y(k-2));
    % Rzutujemy wartość wyjścia względem punktu pracy
    y(k)=Y(k)-Ypp;
    % Liczymy uchyb i uaktualniamy współczynnik błędu
    e(k)=Yzad(k)-Y(k);
    error=error+e(k)^2;
    % Pozyskujemy wektor dUp z wektora du
    dUp=du(k-D+1:k-1);
    dUp=flip(dUp);
    % Liczymy wartość zmiany sterowania
    du_wyliczone=Ke*e(k)-Ku*dUp;

    % Rzutowanie ograniczeń na wartość sterowania
    if du_wyliczone<deltaumin
        du_wyliczone=deltaumin;
    elseif du_wyliczone>deltaumax
        du_wyliczone=deltaumax;
    end
    % Rzutowanie ograniczeń na wartość zmiany sterowania
    if du_wyliczone+u(k-1)<umin
        du_wyliczone=umin-u(k-1);
    elseif du_wyliczone+u(k-1)>umax
        du_wyliczone=umax-u(k-1);
    end
    du(k)=du_wyliczone;
    % Liczymy wartość sterowania i ją rzutujemy względem punktu pracy
    u(k)=u(k-1)+du(k);
    U(k)=u(k)+Upp;
end

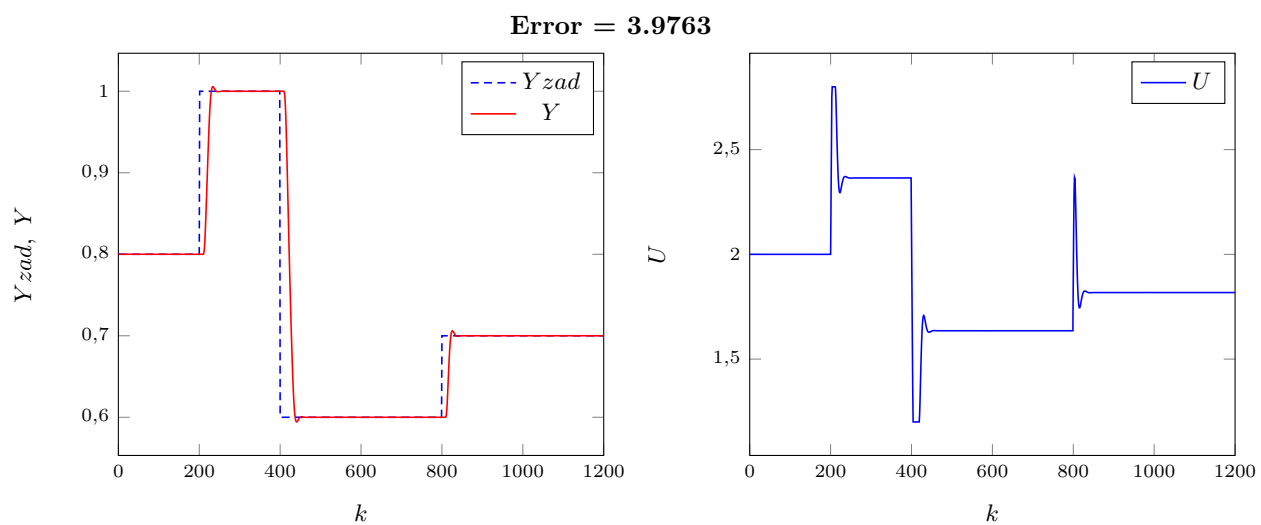
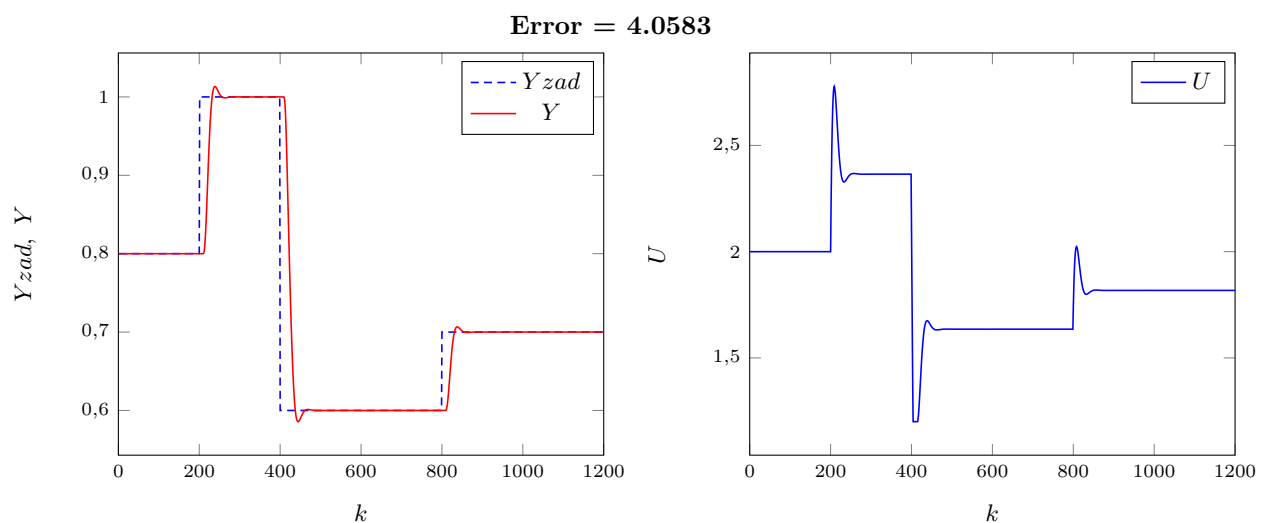
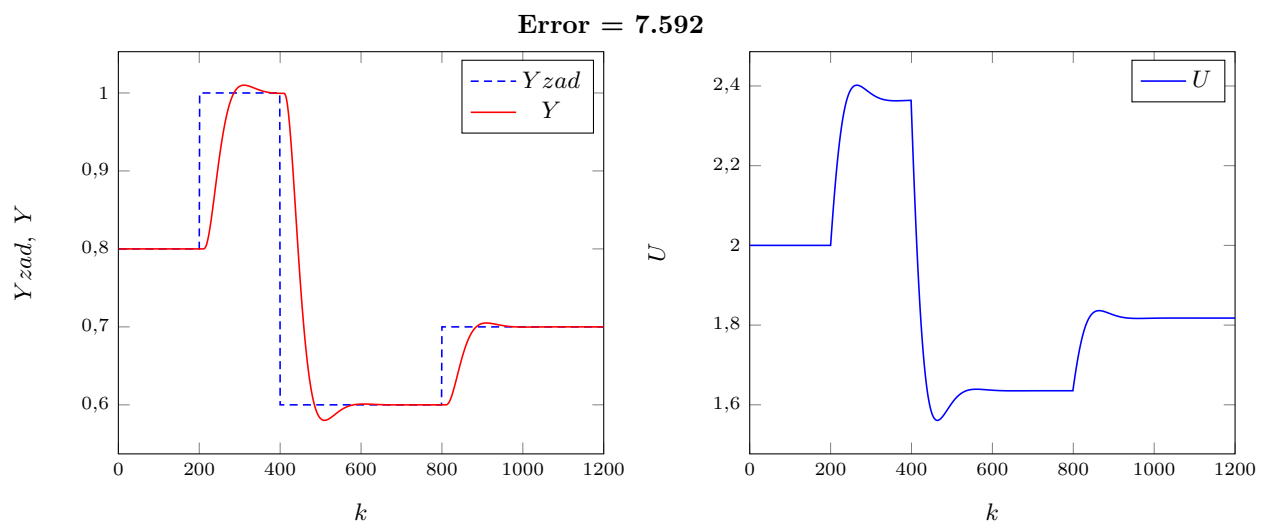
```

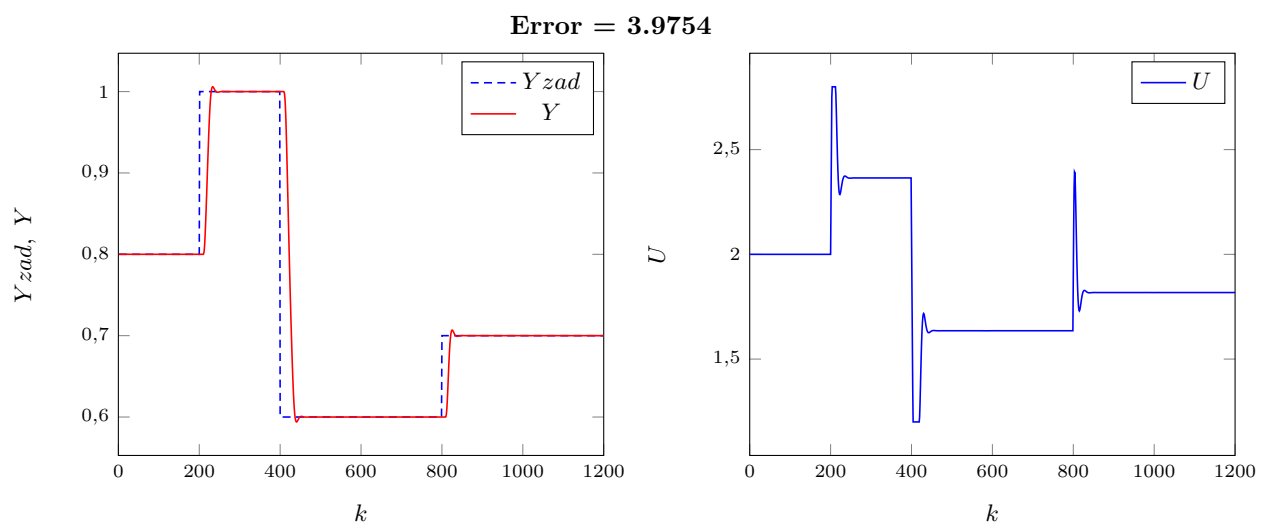
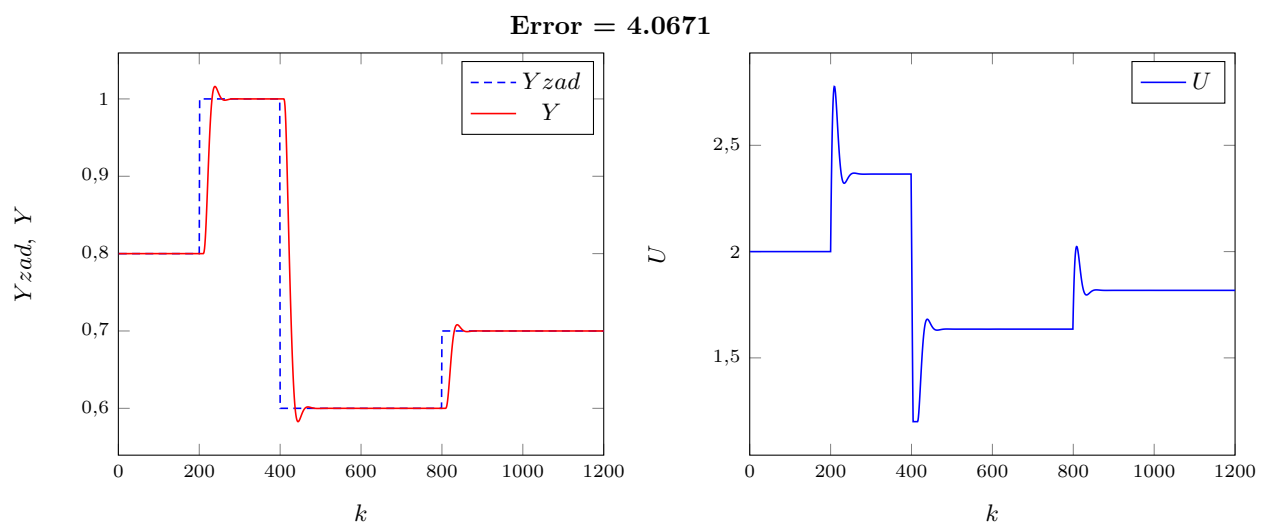
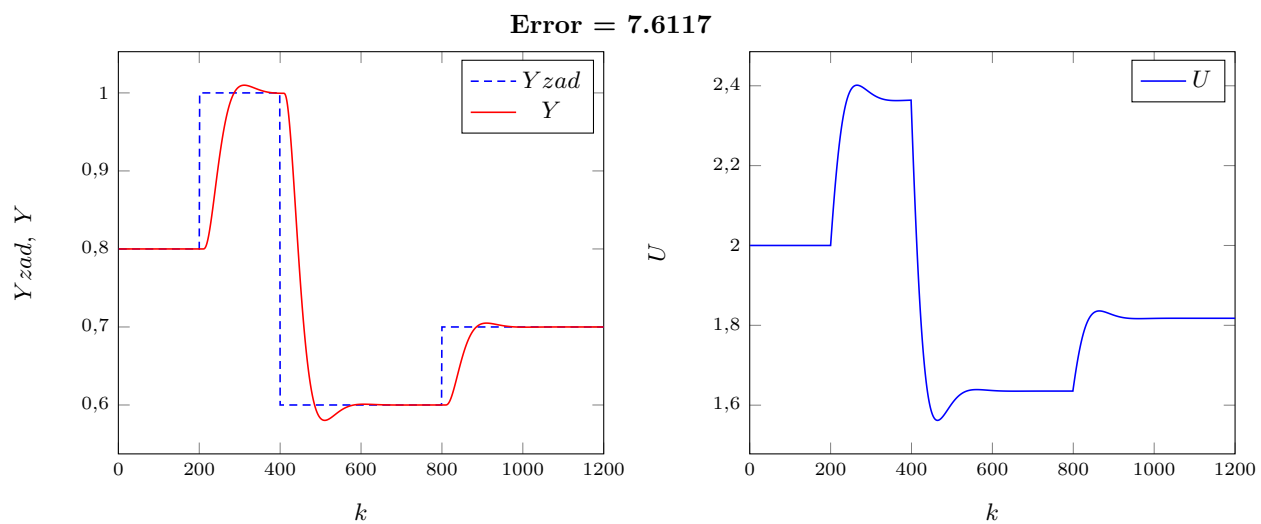
Listing 5.1. Implementacja regulatora DMC

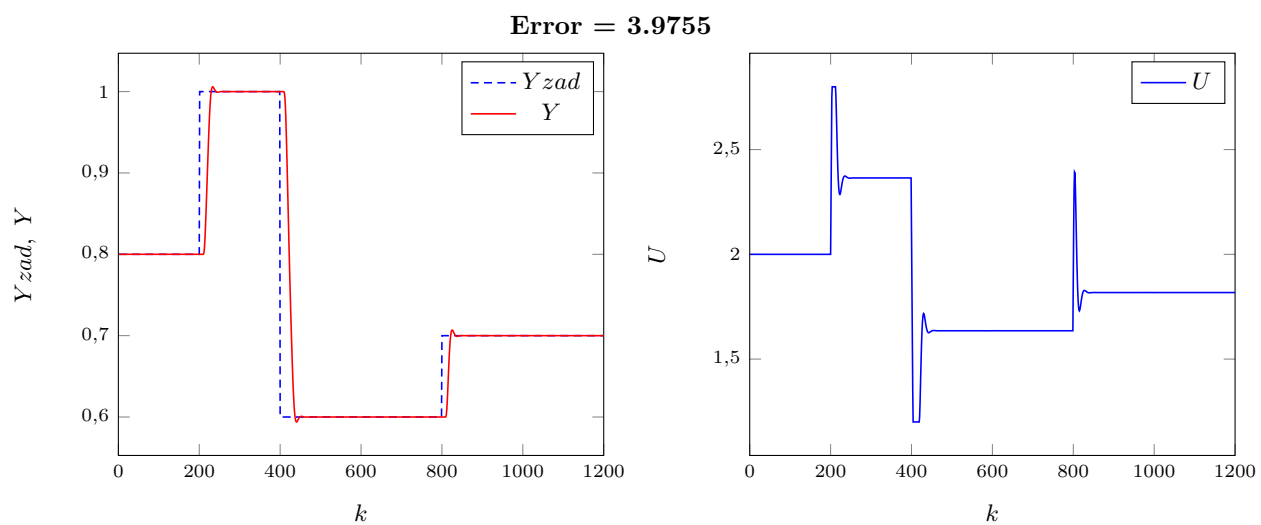
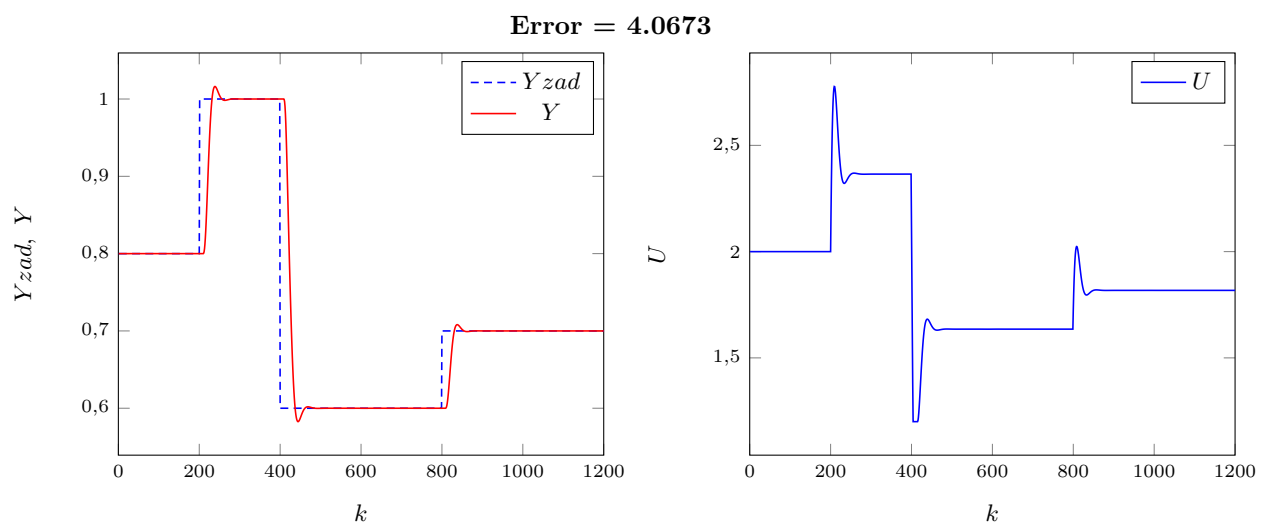
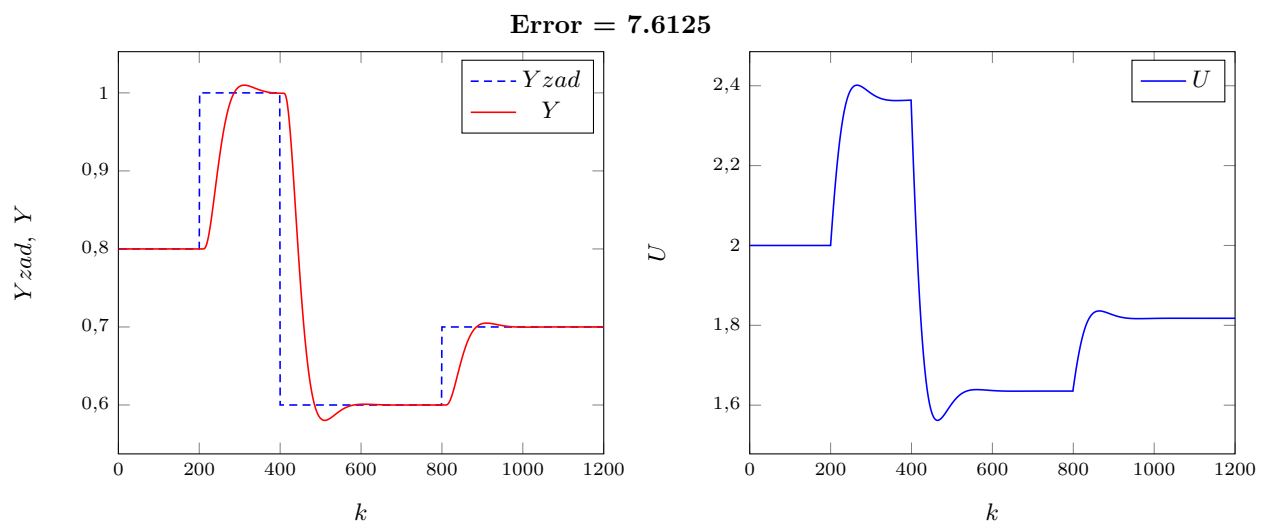
## 5.2. Ręczne strojenie regulatora DMC

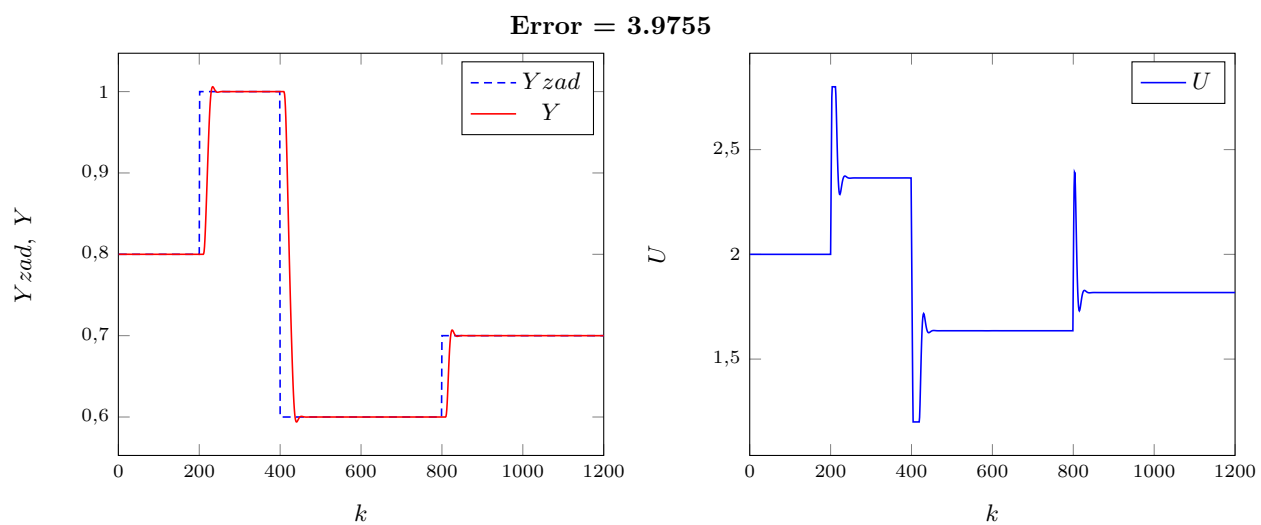
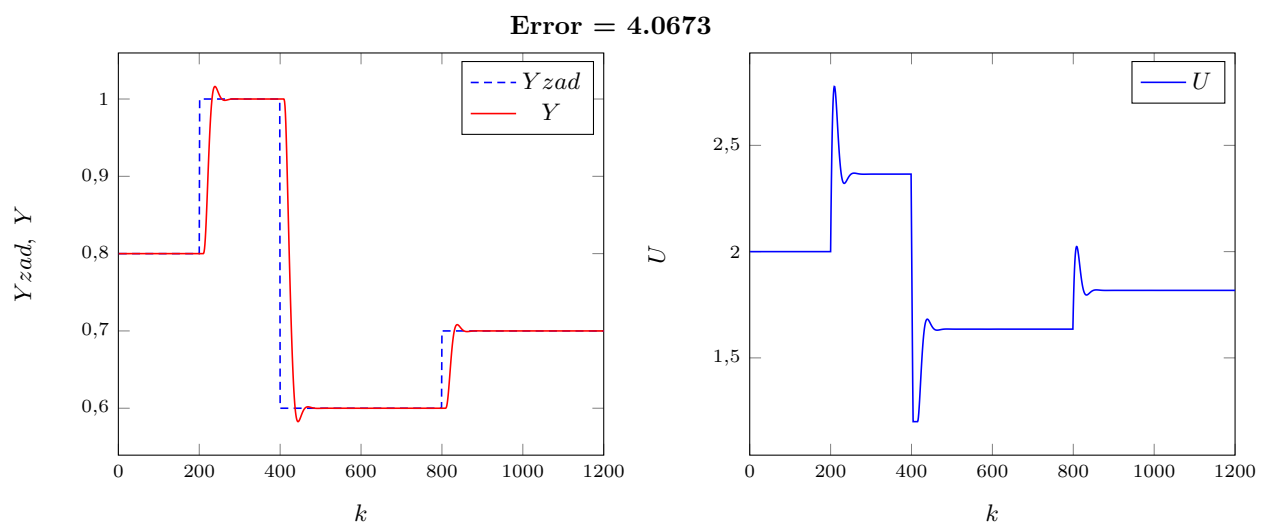
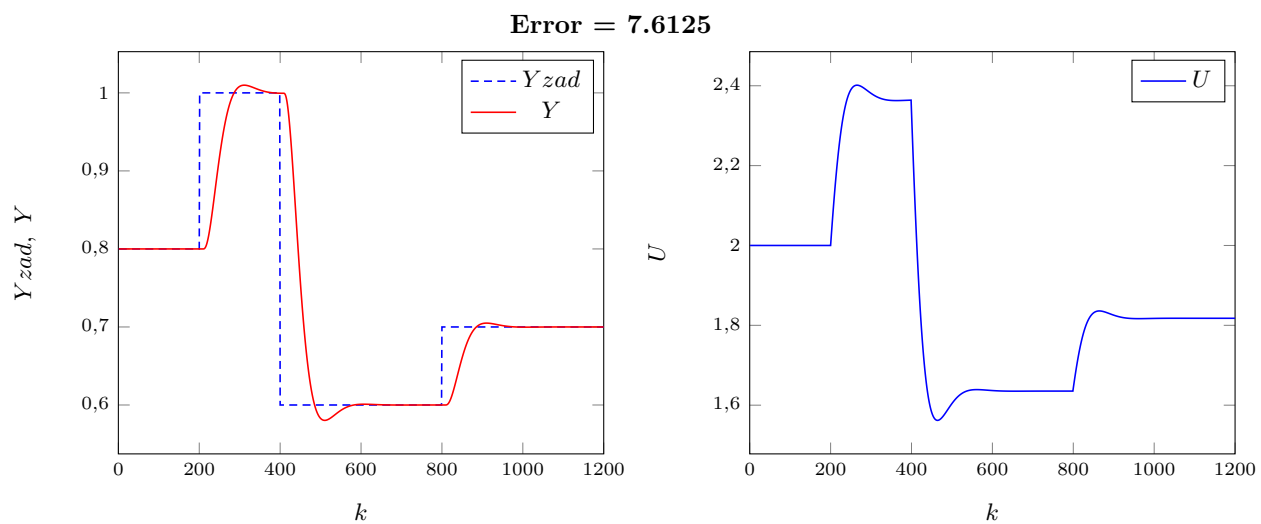
Regulator strojono metodą eksperymentalną. Wybrano kilka arbitralnych wartości parametrów regulatora, pamiętając przy tym, że  $N_u \leq N \leq D$ . Następnie przetestowano ich kombinacje i przeprowadzono symulacje. Wykres przedstawiono poniżej:

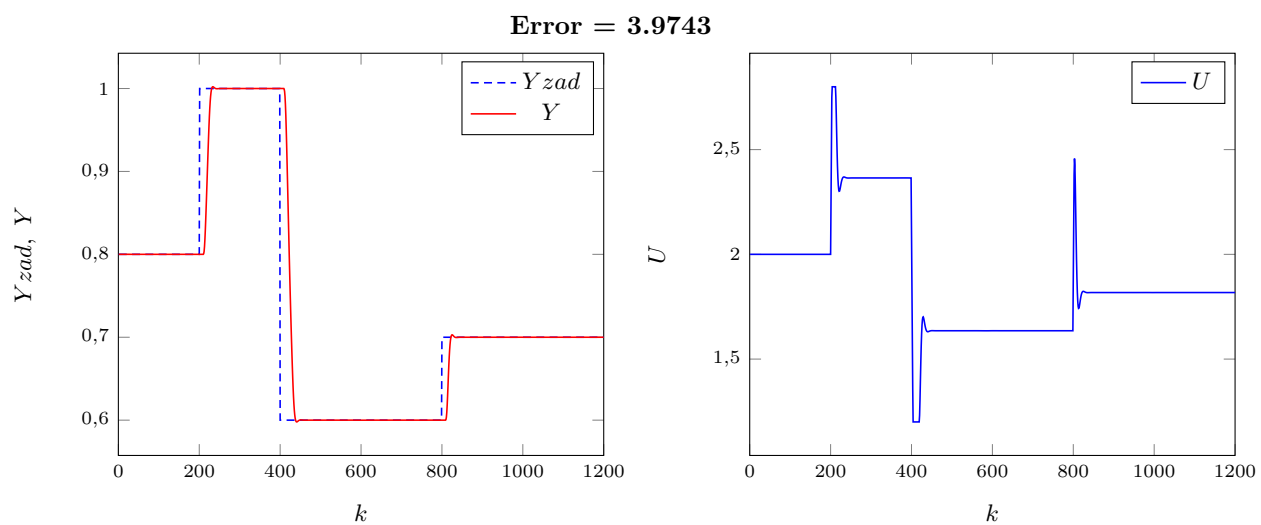
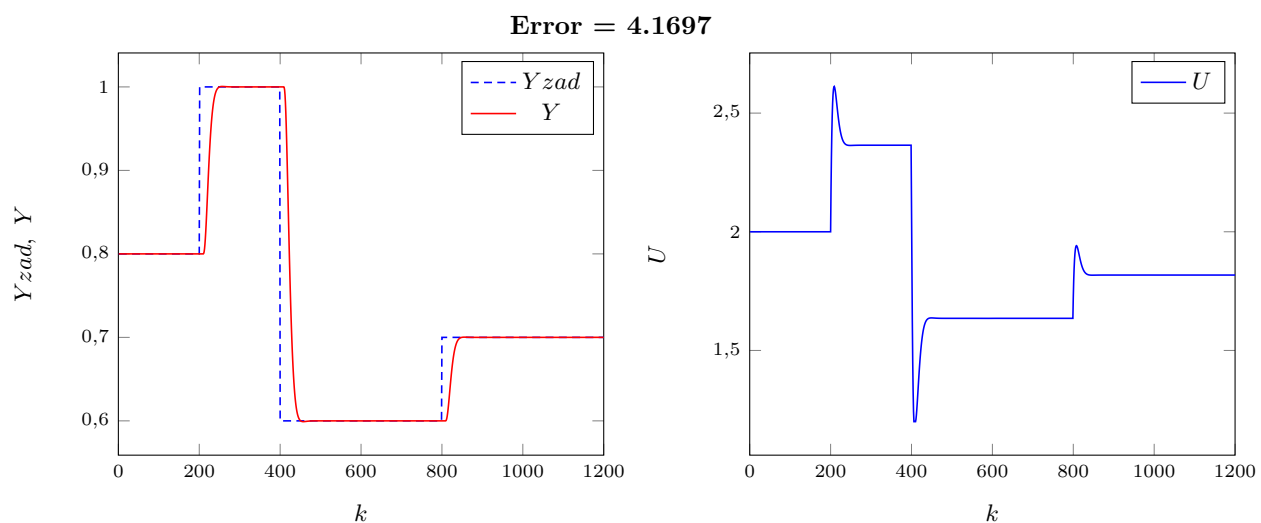
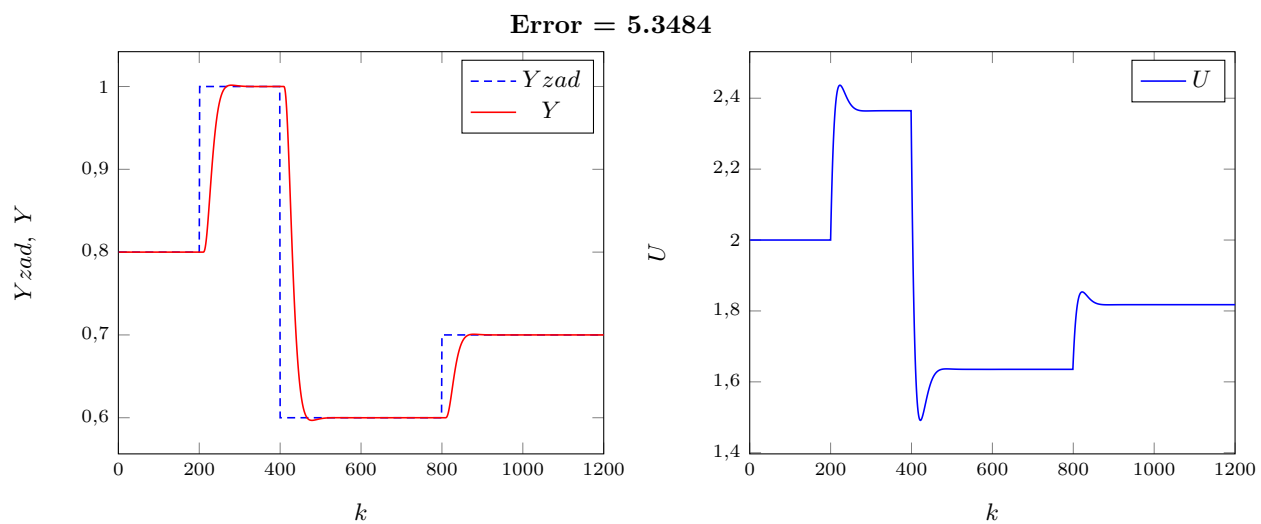
Najlepszym regulatorem okazał się ten o parametrach  $N = 50$ ,  $N_u = 5$ ,  $\lambda = 0.2$ .

Rys. 5.1. Przebieg oraz sterowanie dla parametrów  $N = 25$ ,  $Nu = 5$ ,  $\lambda = 0.2$ Rys. 5.2. Przebieg oraz sterowanie dla parametrów  $N = 25$ ,  $Nu = 5$ ,  $\lambda = 1$ Rys. 5.3. Przebieg oraz sterowanie dla parametrów  $N = 25$ ,  $Nu = 5$ ,  $\lambda = 5$

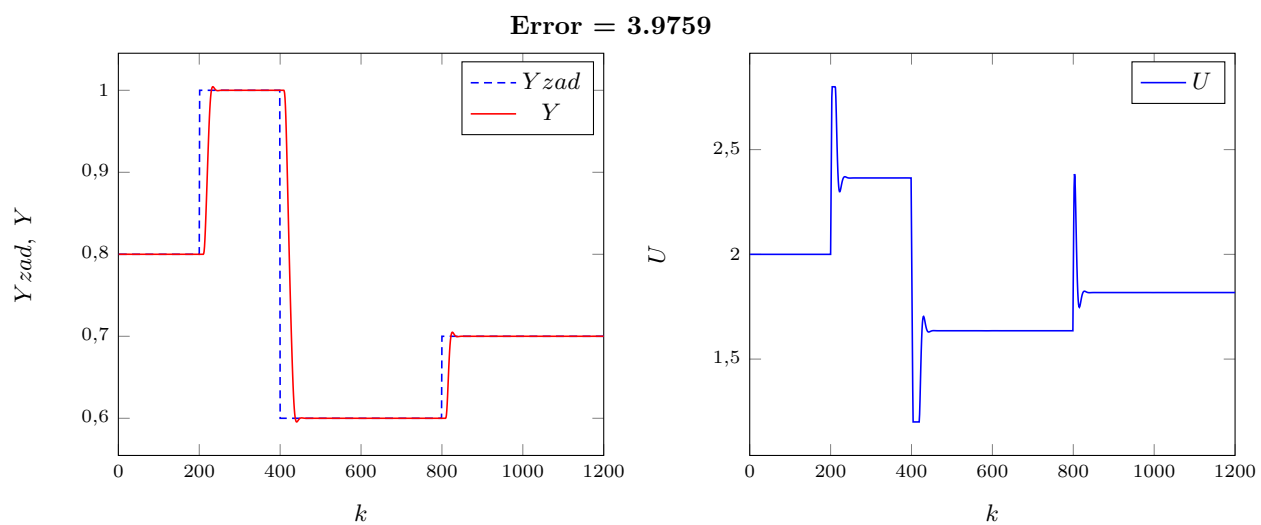
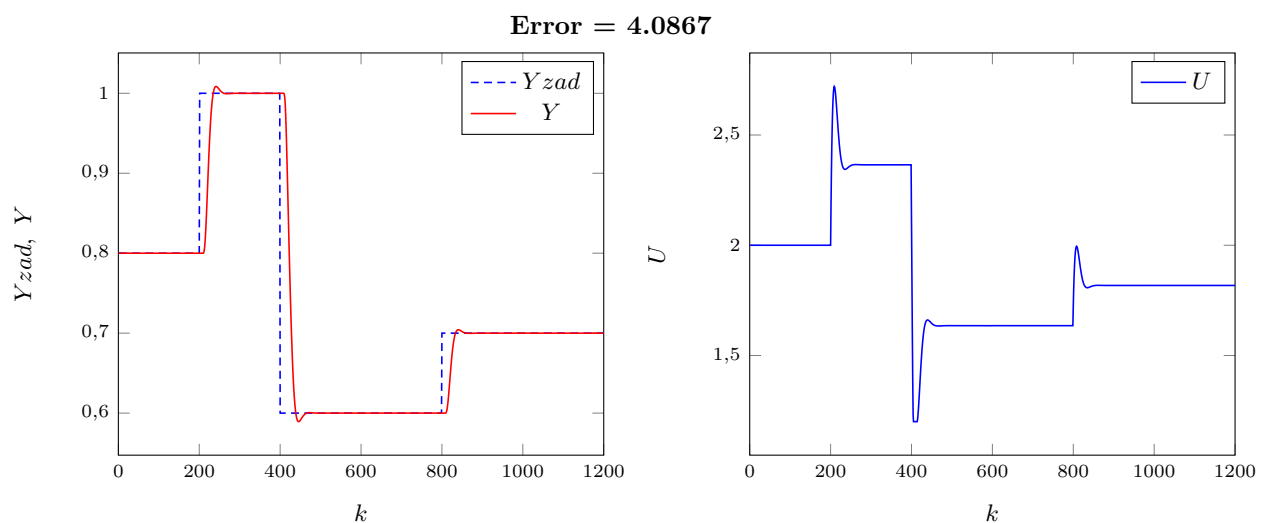
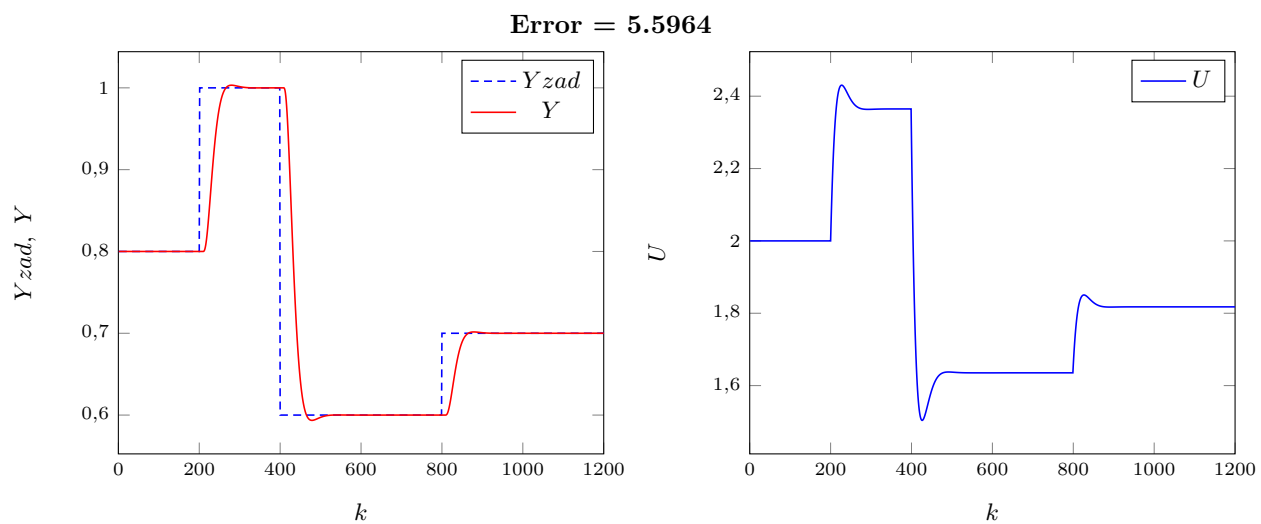
Rys. 5.4. Przebieg oraz sterowanie dla parametrów  $N = 25$ ,  $Nu = 10$ ,  $\lambda = 0.2$ Rys. 5.5. Przebieg oraz sterowanie dla parametrów  $N = 25$ ,  $Nu = 10$ ,  $\lambda = 1$ Rys. 5.6. Przebieg oraz sterowanie dla parametrów  $N = 25$ ,  $Nu = 10$ ,  $\lambda = 5$

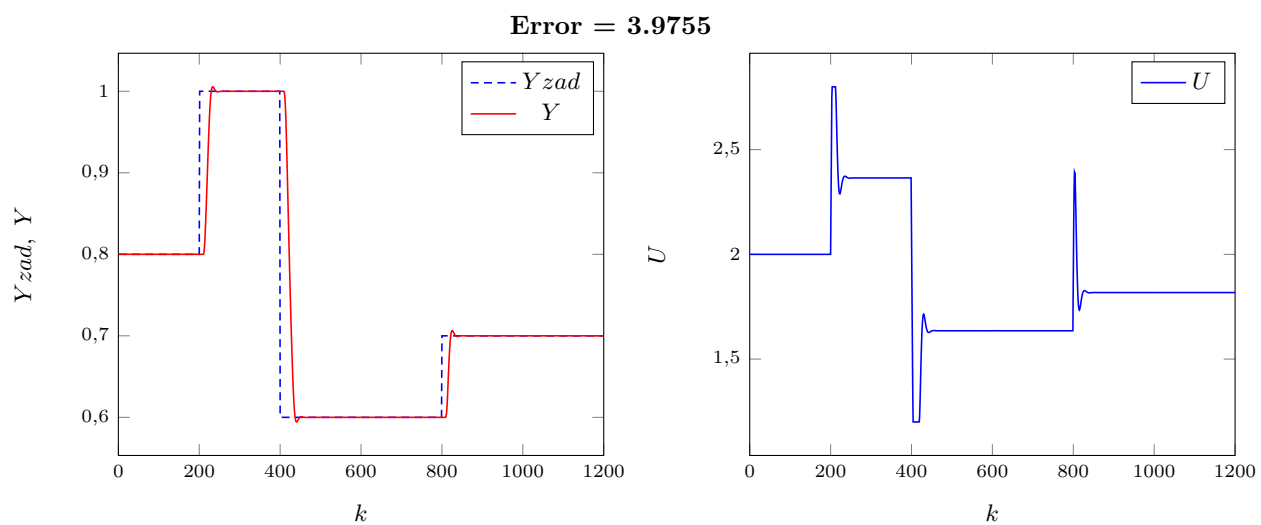
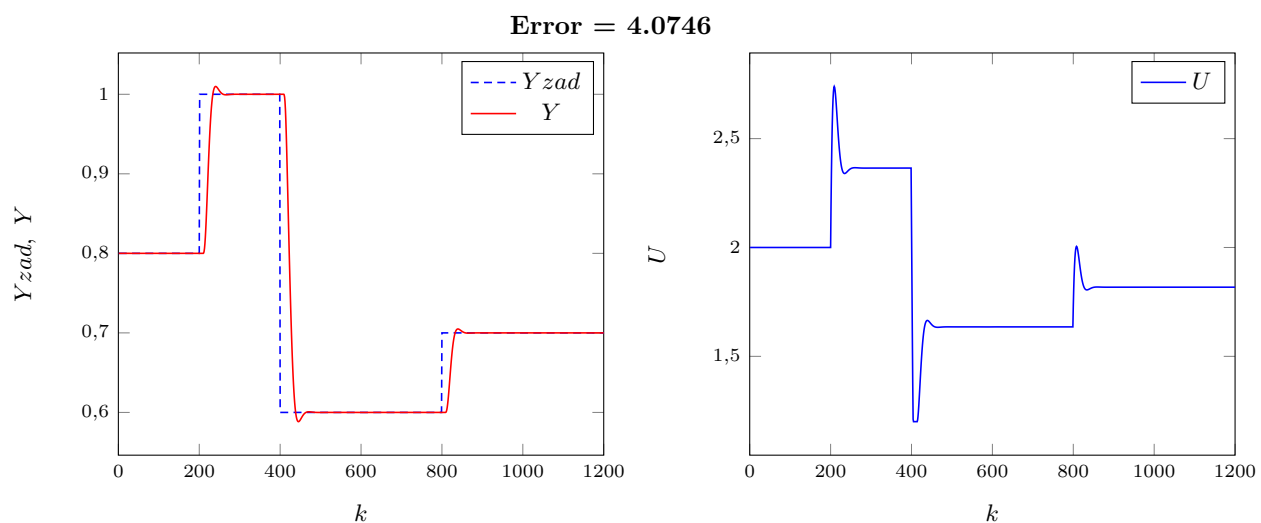
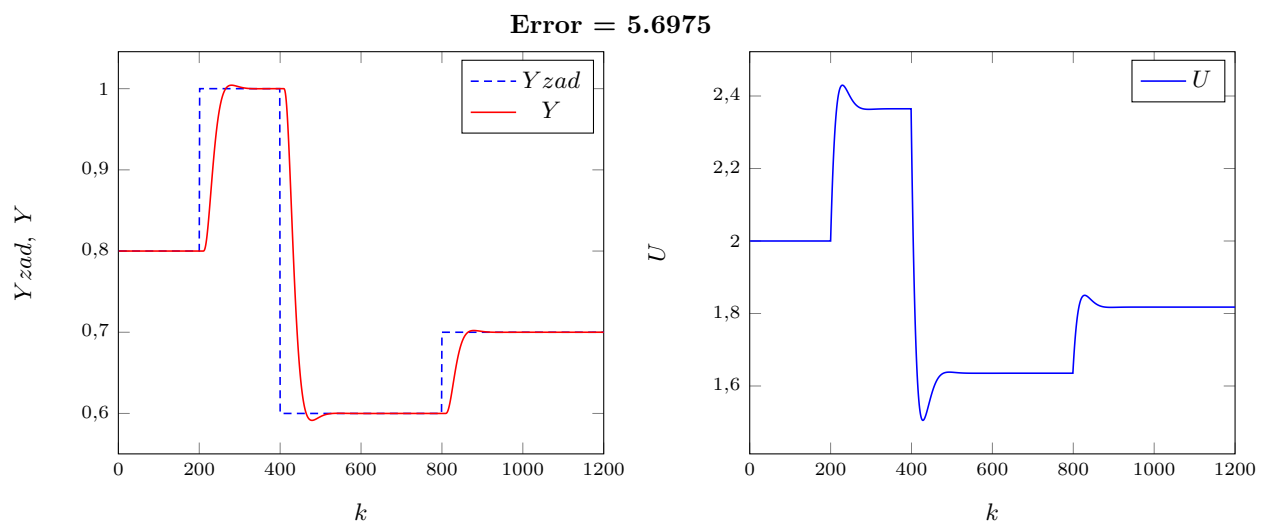
Rys. 5.7. Przebieg oraz sterowanie dla parametrów  $N = 25$ ,  $Nu = 15$ ,  $\lambda = 0.2$ Rys. 5.8. Przebieg oraz sterowanie dla parametrów  $N = 25$ ,  $Nu = 15$ ,  $\lambda = 1$ Rys. 5.9. Przebieg oraz sterowanie dla parametrów  $N = 25$ ,  $Nu = 15$ ,  $\lambda = 5$

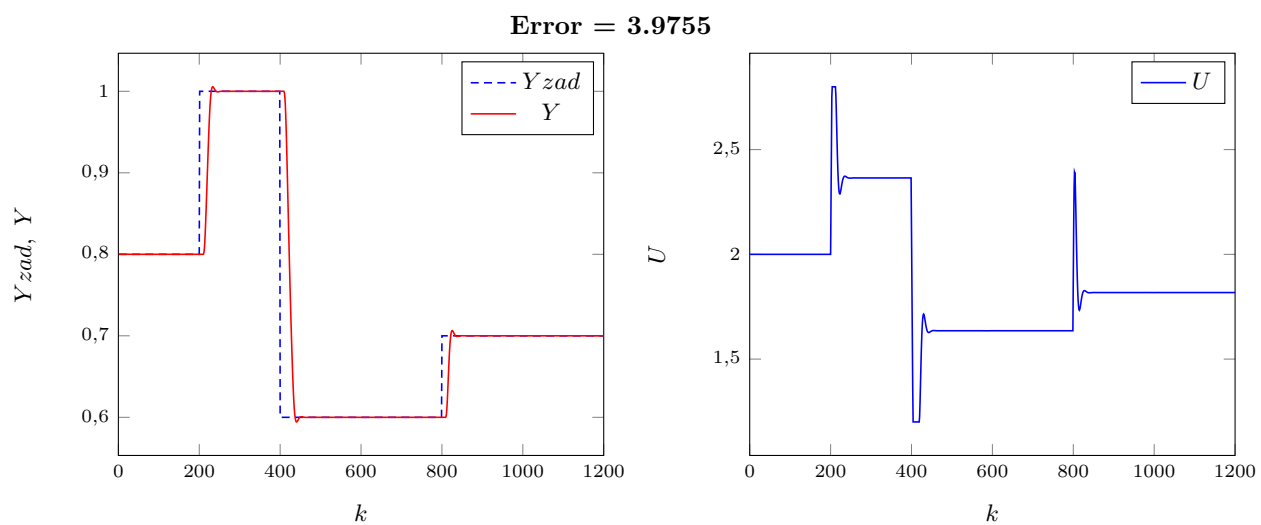
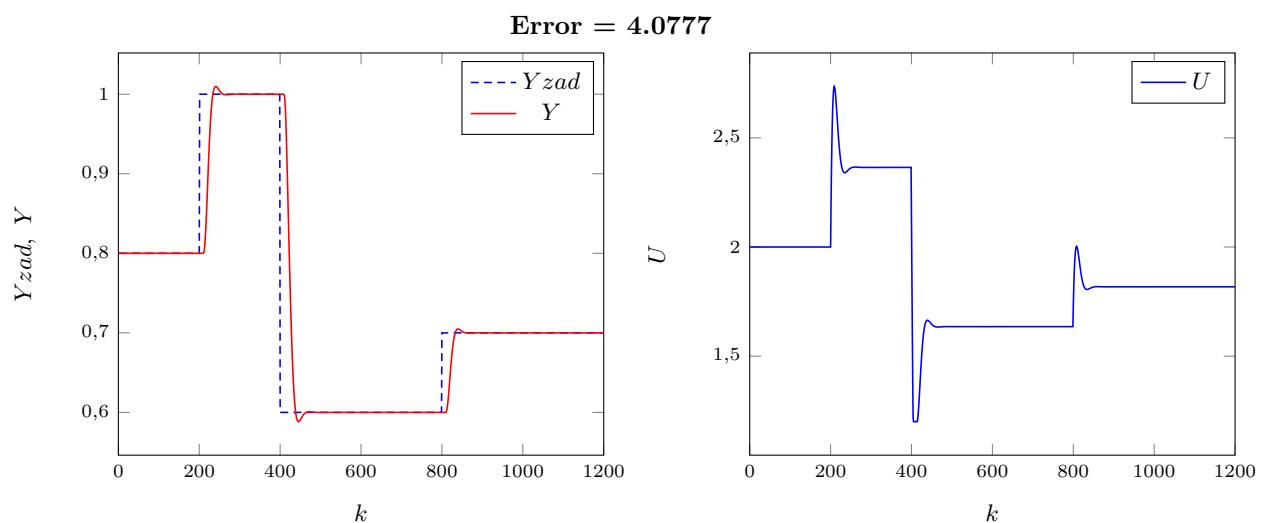
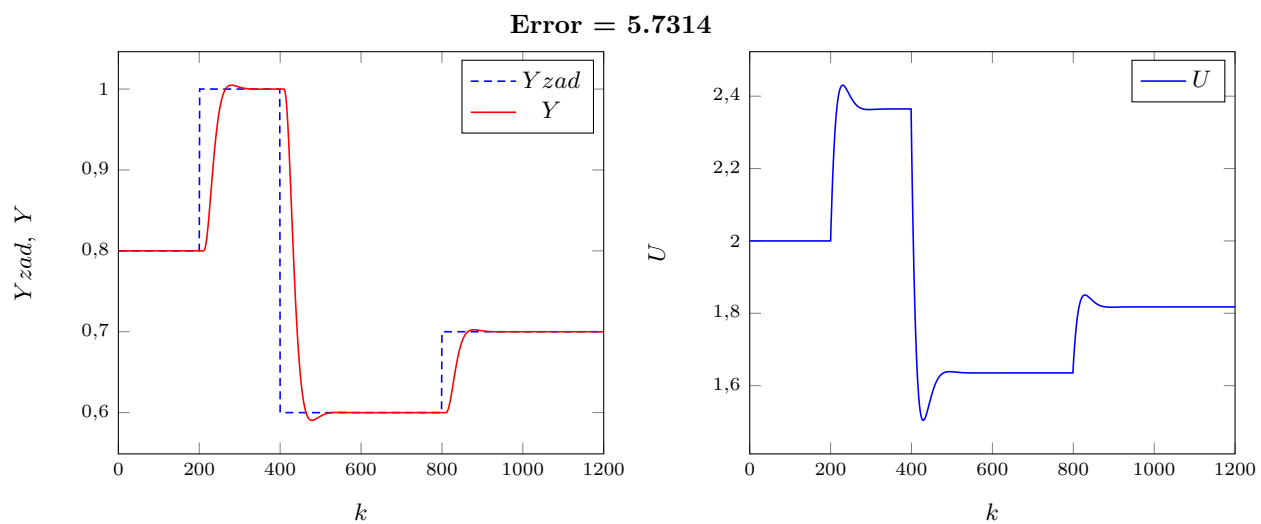
Rys. 5.10. Przebieg oraz sterowanie dla parametrów  $N = 25$ ,  $Nu = 20$ ,  $\lambda = 0.2$ Rys. 5.11. Przebieg oraz sterowanie dla parametrów  $N = 25$ ,  $Nu = 20$ ,  $\lambda = 1$ Rys. 5.12. Przebieg oraz sterowanie dla parametrów  $N = 25$ ,  $Nu = 20$ ,  $\lambda = 5$

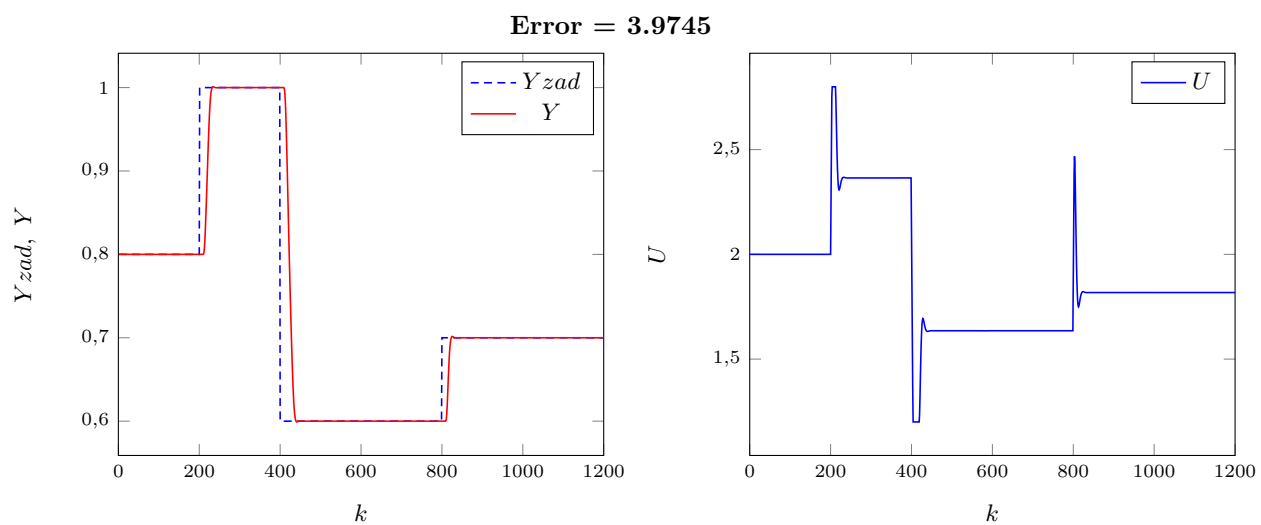
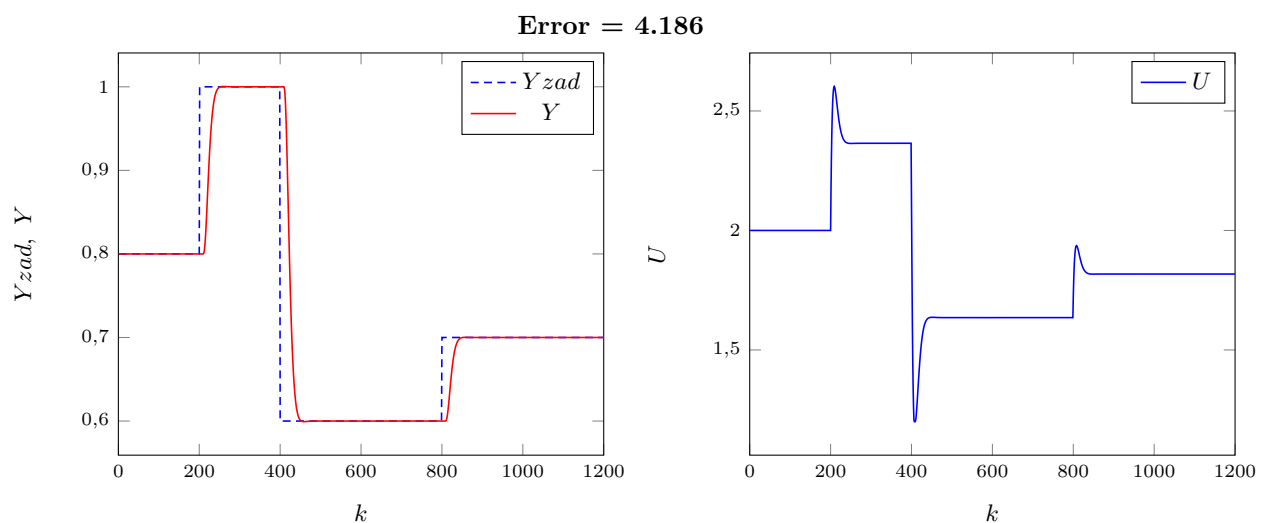
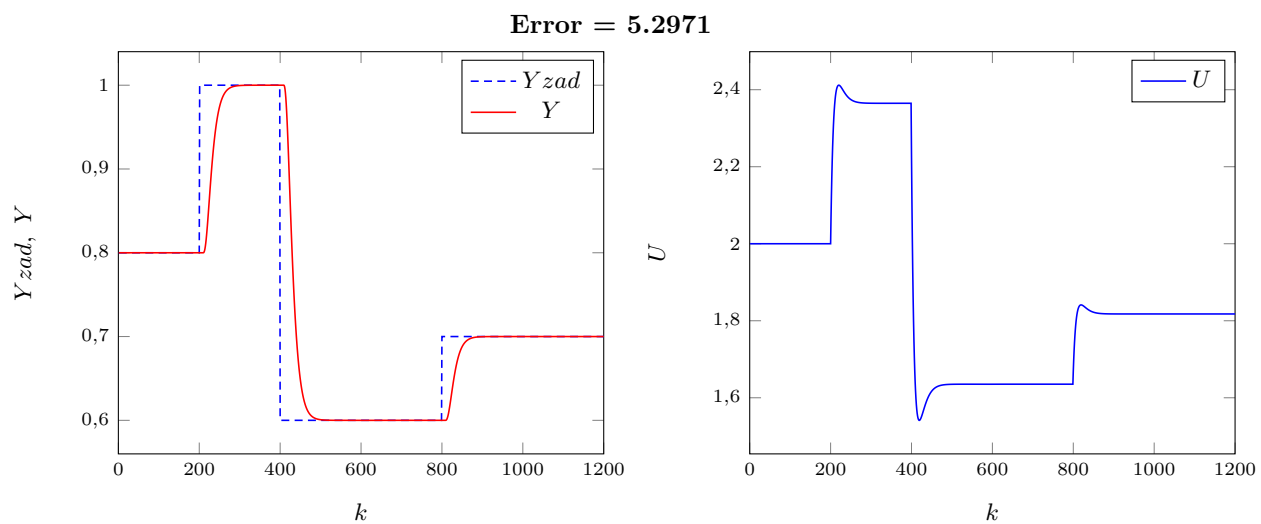
Rys. 5.13. Przebieg oraz sterowanie dla parametrów  $N = 50$ ,  $Nu = 5$ ,  $\lambda = 0.2$ Rys. 5.14. Przebieg oraz sterowanie dla parametrów  $N = 50$ ,  $Nu = 5$ ,  $\lambda = 1$ Rys. 5.15. Przebieg oraz sterowanie dla parametrów  $N = 50$ ,  $Nu = 5$ ,  $\lambda = 5$

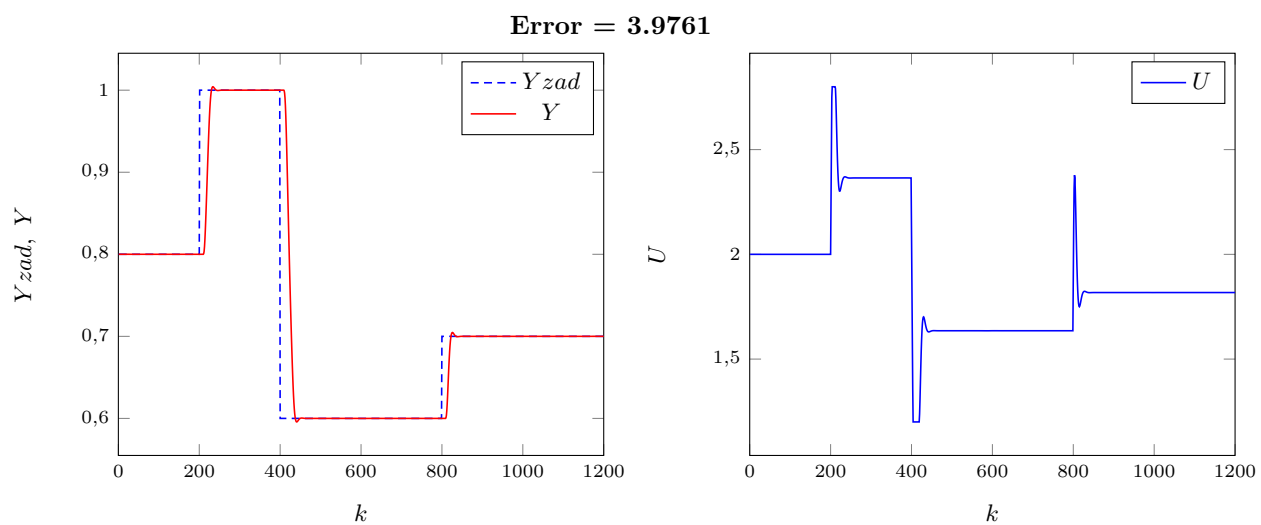
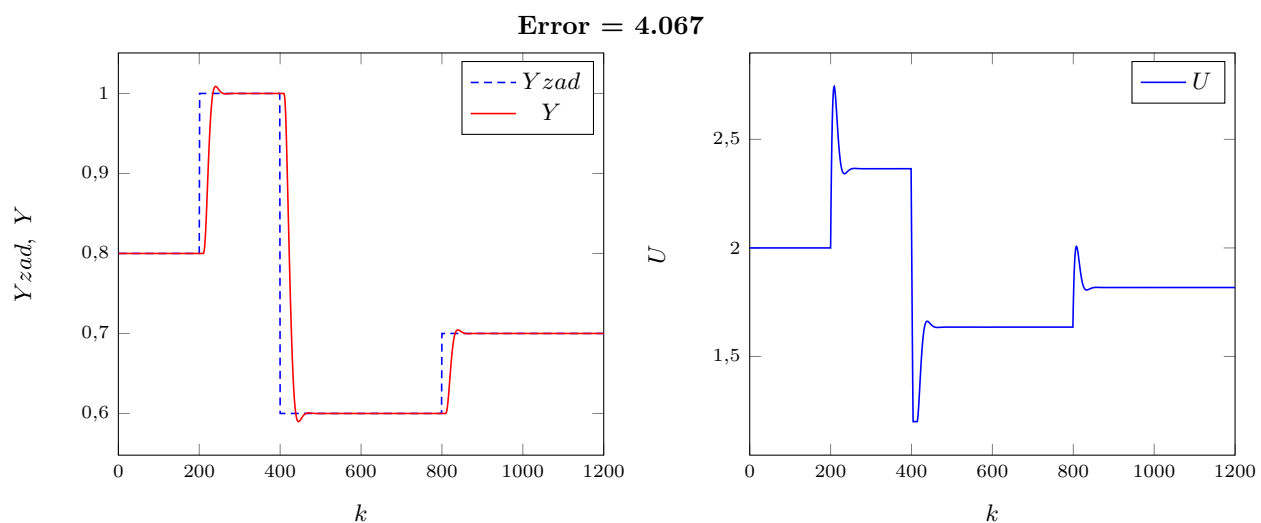
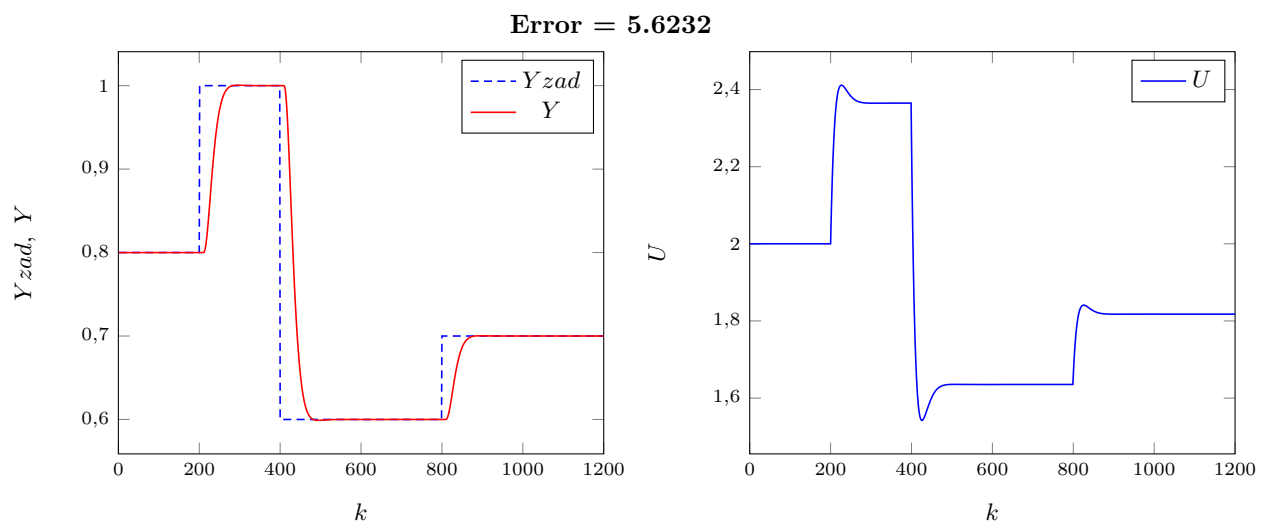


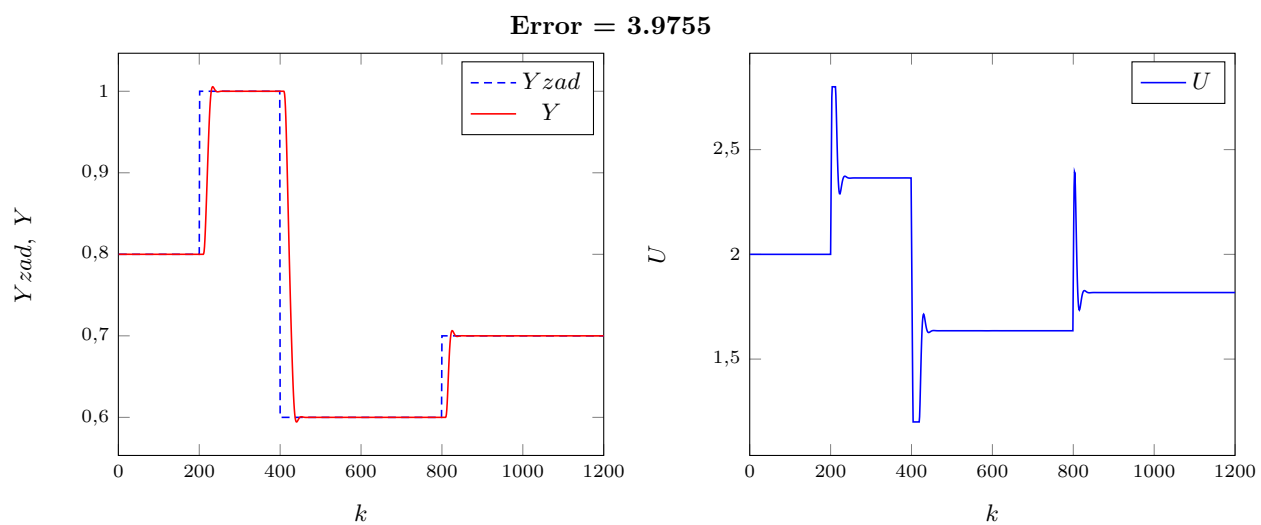
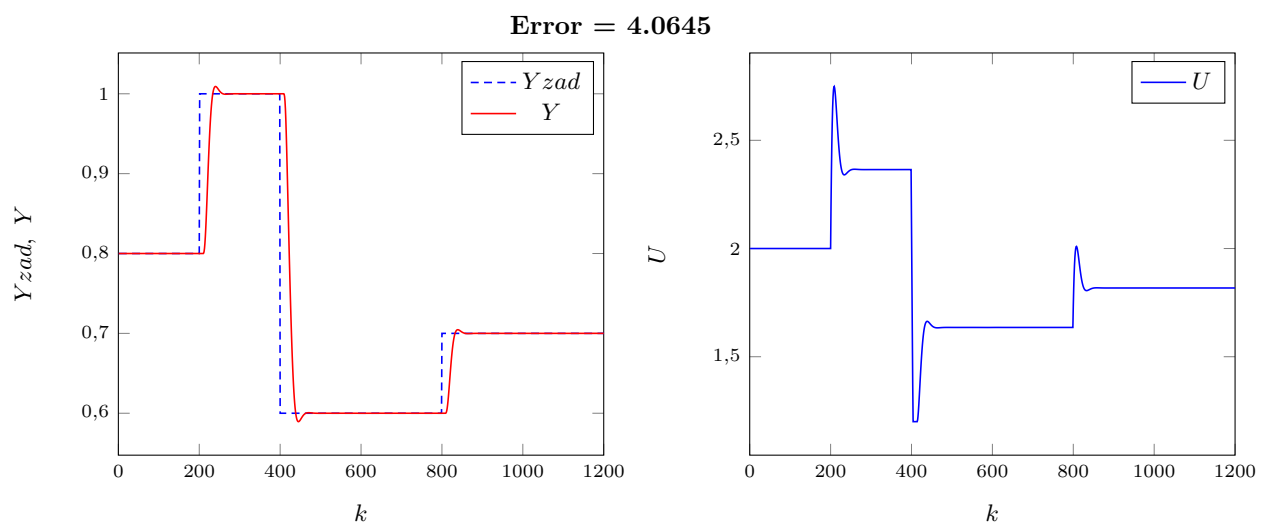
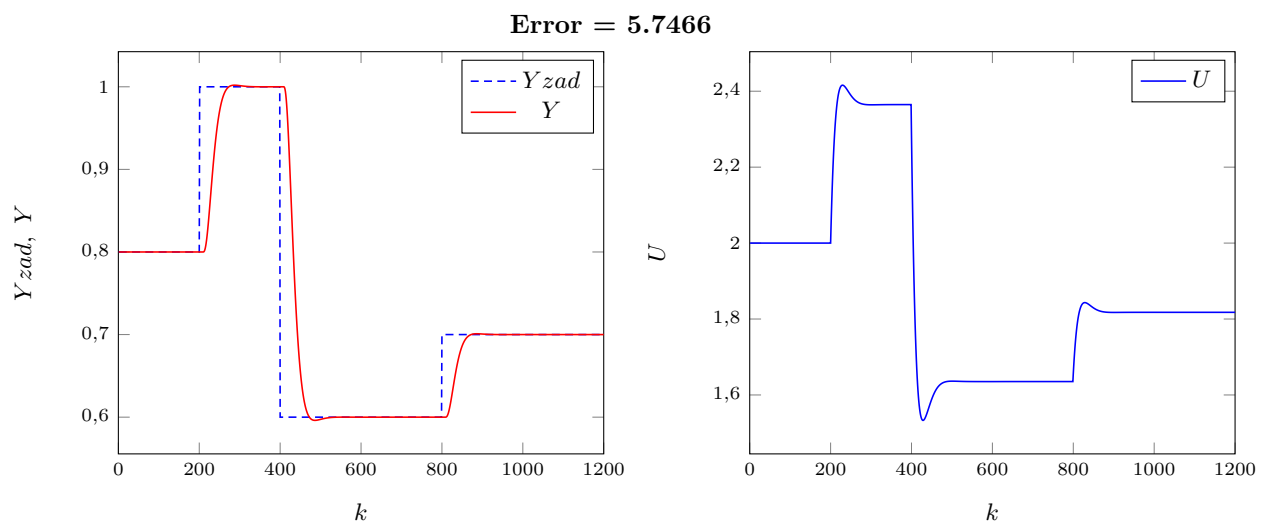
Rys. 5.16. Przebieg oraz sterowanie dla parametrów  $N = 50$ ,  $Nu = 10$ ,  $\lambda = 0.2$ Rys. 5.17. Przebieg oraz sterowanie dla parametrów  $N = 50$ ,  $Nu = 10$ ,  $\lambda = 1$ Rys. 5.18. Przebieg oraz sterowanie dla parametrów  $N = 50$ ,  $Nu = 10$ ,  $\lambda = 5$

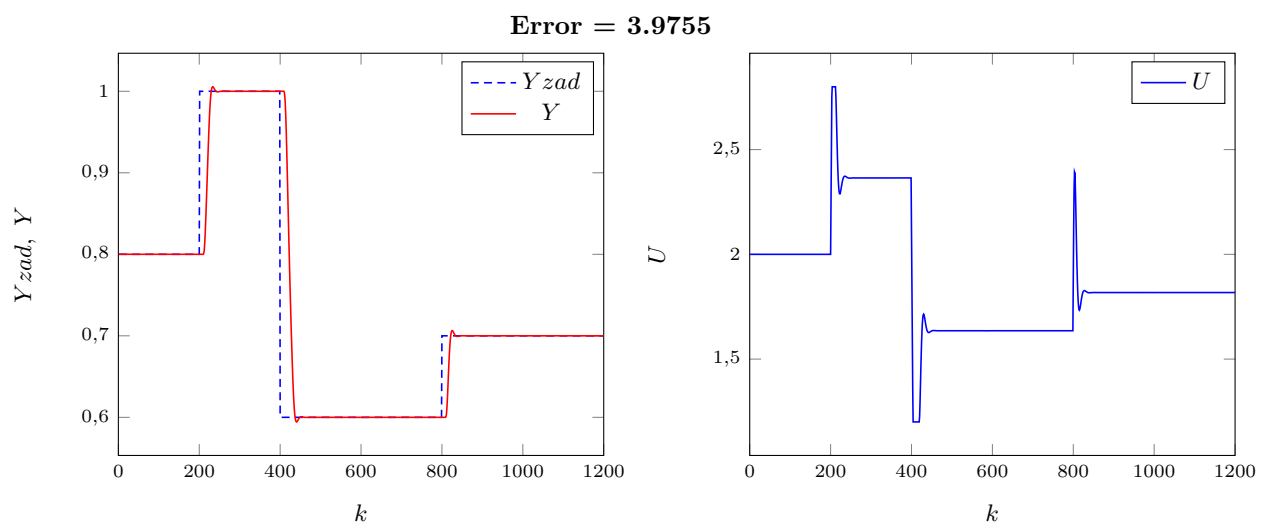
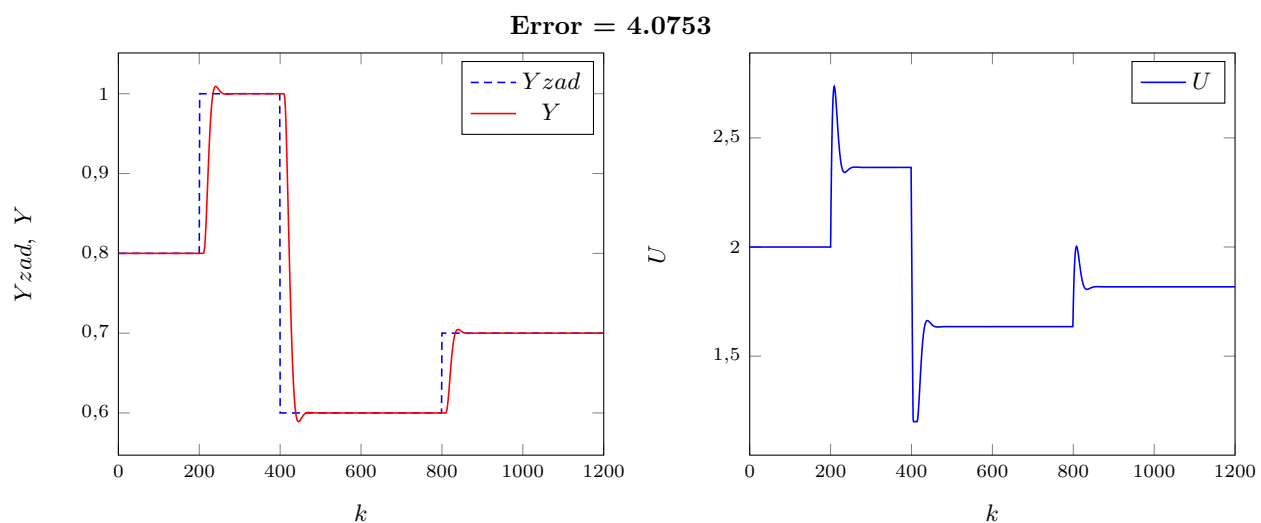
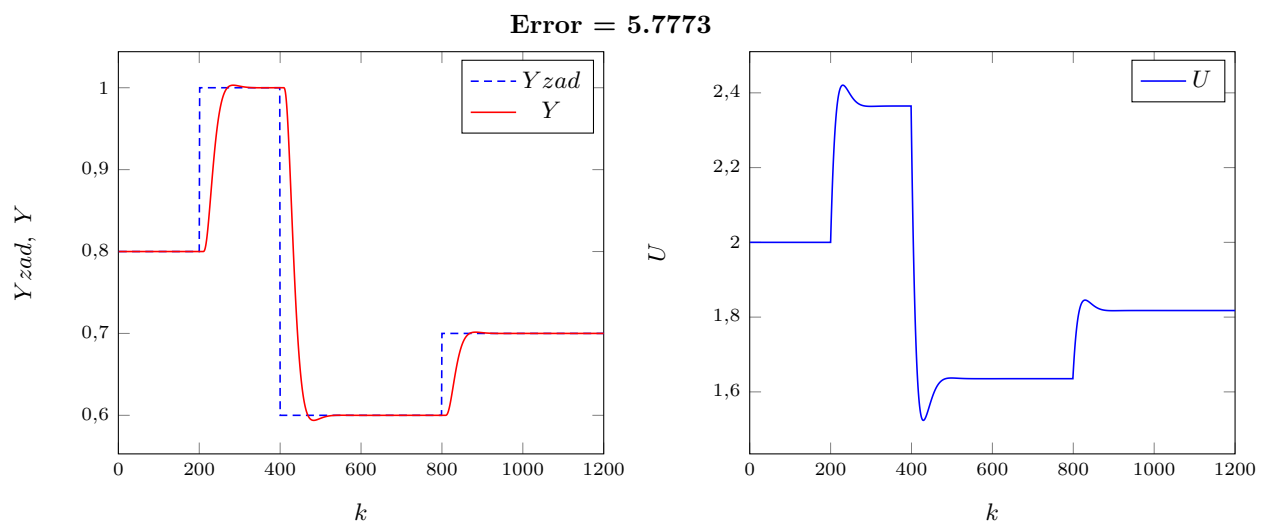
Rys. 5.19. Przebieg oraz sterowanie dla parametrów  $N = 50$ ,  $Nu = 15$ ,  $\lambda = 0.2$ Rys. 5.20. Przebieg oraz sterowanie dla parametrów  $N = 50$ ,  $Nu = 15$ ,  $\lambda = 1$ Rys. 5.21. Przebieg oraz sterowanie dla parametrów  $N = 50$ ,  $Nu = 15$ ,  $\lambda = 5$

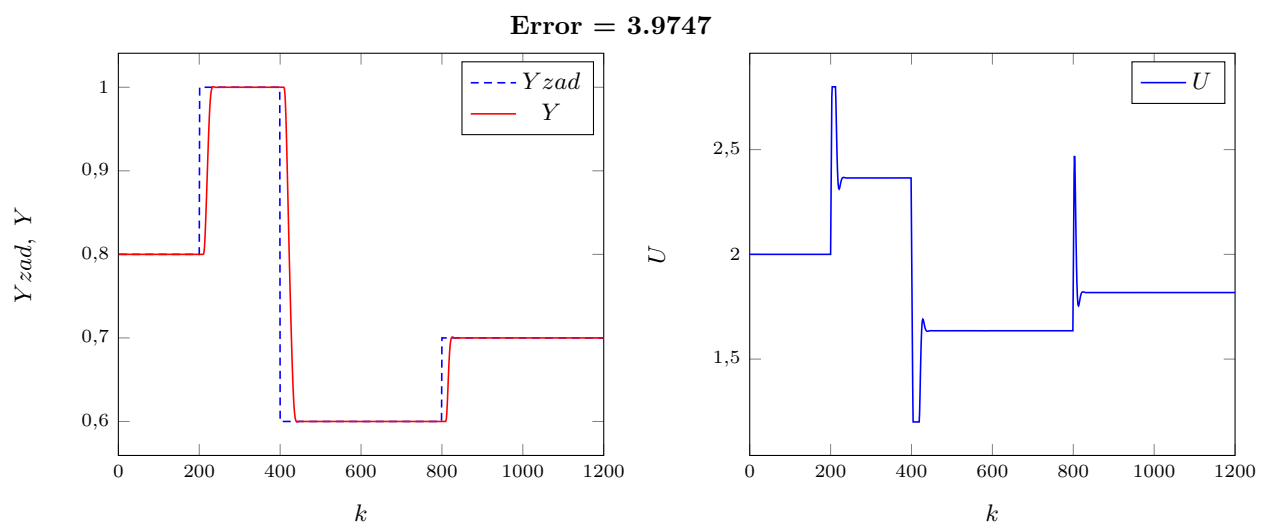
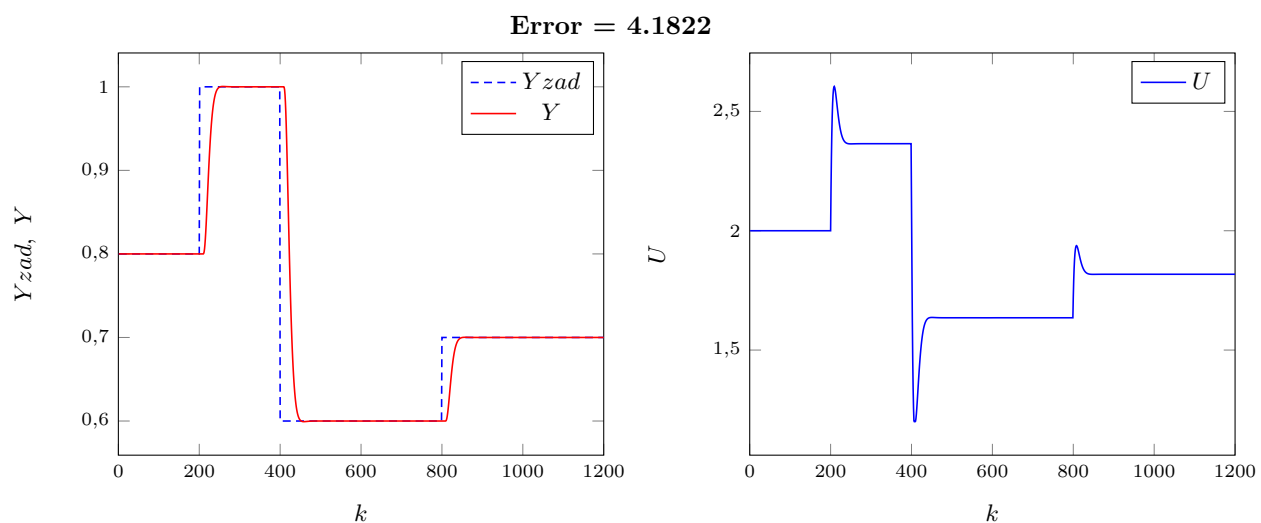
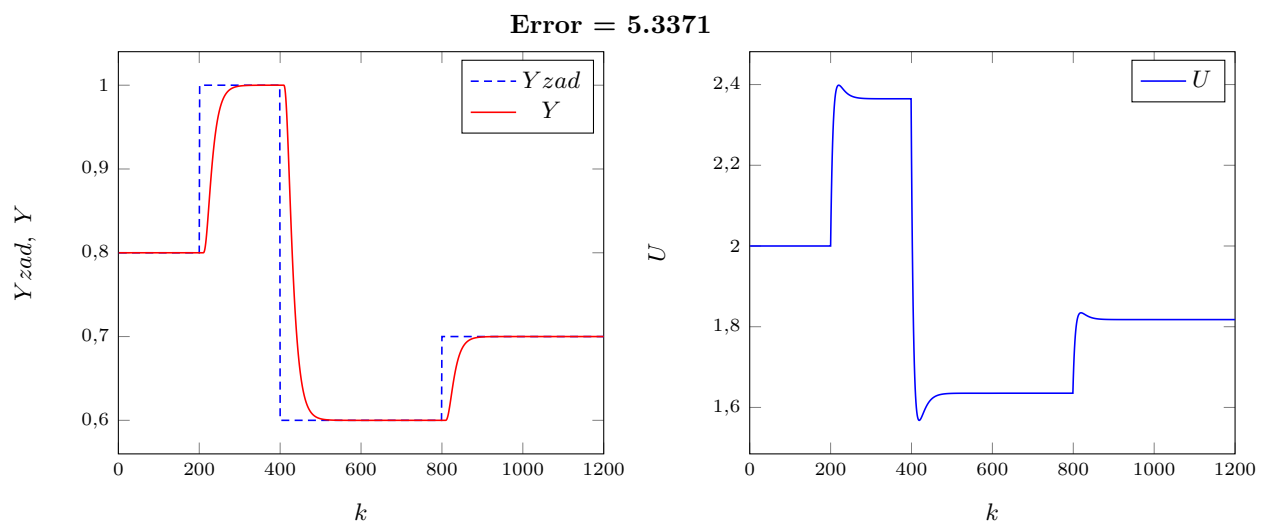
Rys. 5.22. Przebieg oraz sterowanie dla parametrów  $N = 50$ ,  $Nu = 20$ ,  $\lambda = 0.2$ Rys. 5.23. Przebieg oraz sterowanie dla parametrów  $N = 50$ ,  $Nu = 20$ ,  $\lambda = 1$ Rys. 5.24. Przebieg oraz sterowanie dla parametrów  $N = 50$ ,  $Nu = 20$ ,  $\lambda = 5$

Rys. 5.25. Przebieg oraz sterowanie dla parametrów  $N = 75$ ,  $Nu = 5$ ,  $\lambda = 0.2$ Rys. 5.26. Przebieg oraz sterowanie dla parametrów  $N = 75$ ,  $Nu = 5$ ,  $\lambda = 1$ Rys. 5.27. Przebieg oraz sterowanie dla parametrów  $N = 75$ ,  $Nu = 5$ ,  $\lambda = 5$

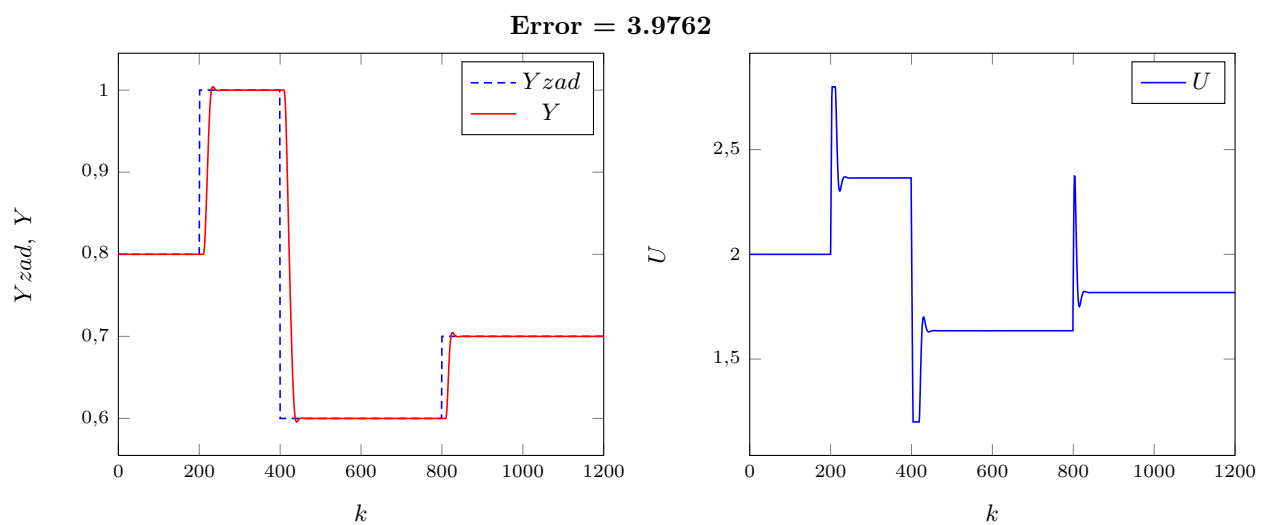
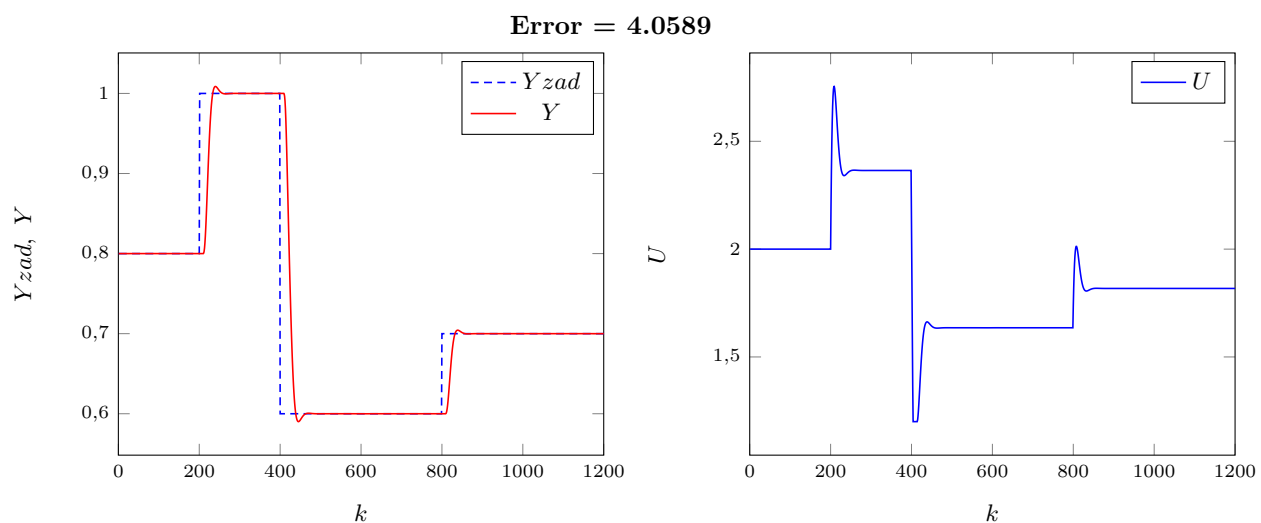
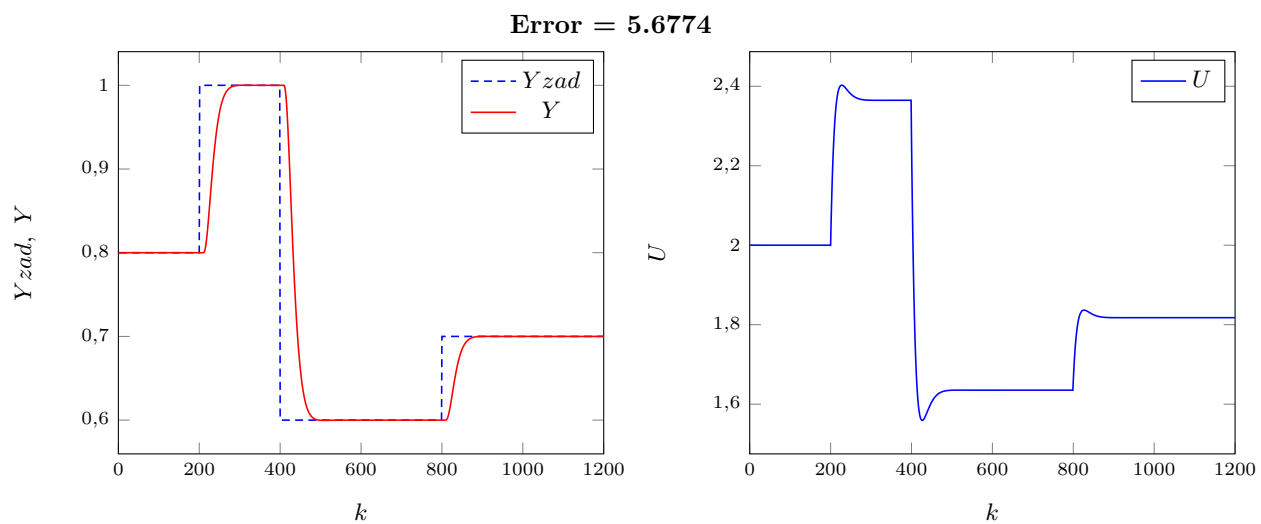
Rys. 5.28. Przebieg oraz sterowanie dla parametrów  $N = 75$ ,  $Nu = 10$ ,  $\lambda = 0.2$ Rys. 5.29. Przebieg oraz sterowanie dla parametrów  $N = 75$ ,  $Nu = 10$ ,  $\lambda = 1$ Rys. 5.30. Przebieg oraz sterowanie dla parametrów  $N = 75$ ,  $Nu = 10$ ,  $\lambda = 5$

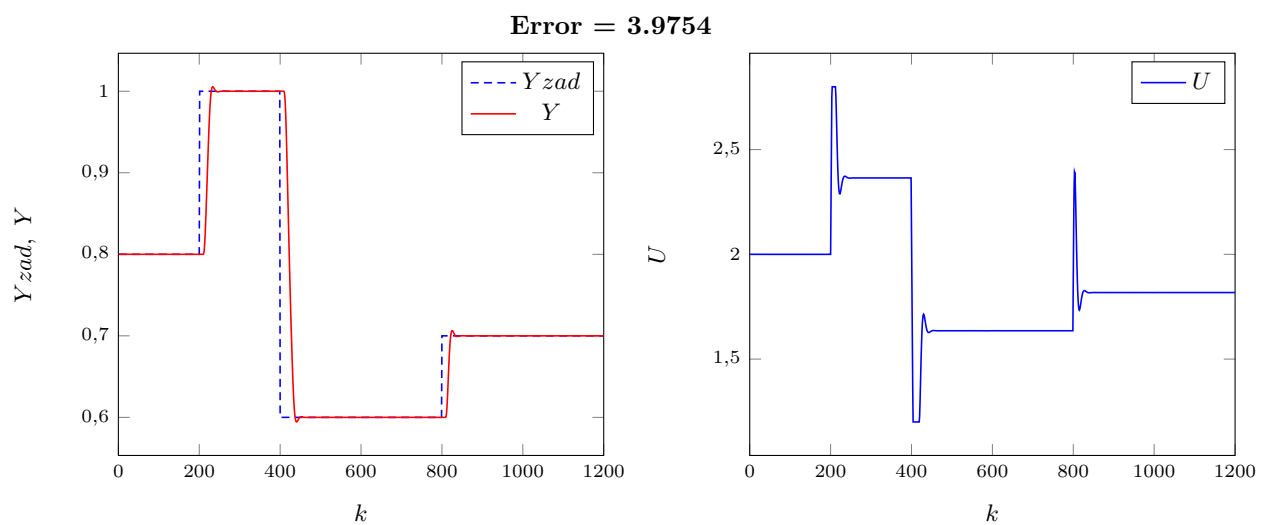
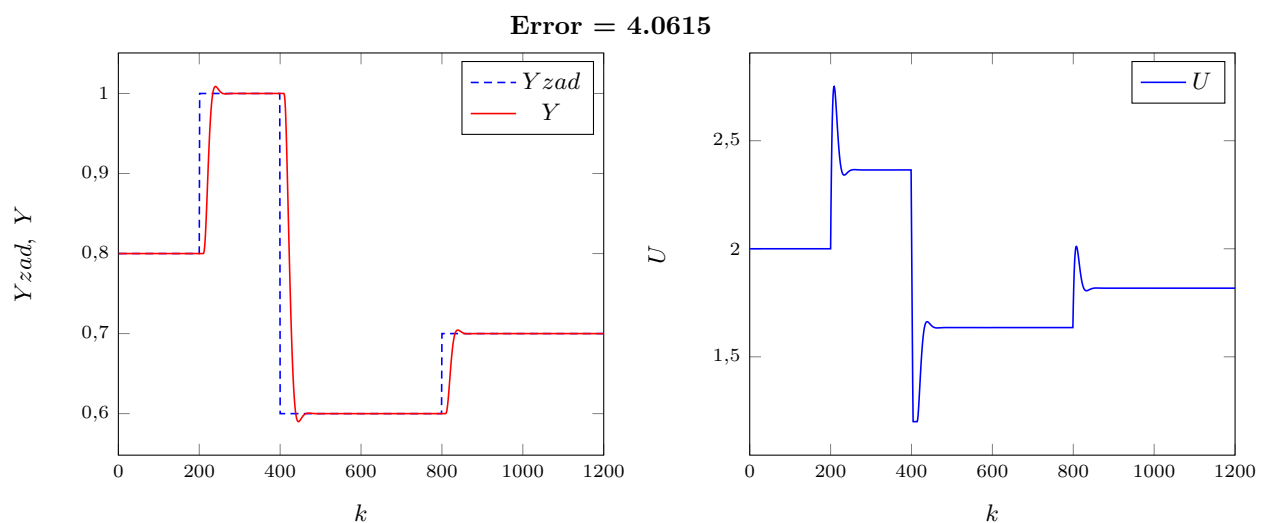
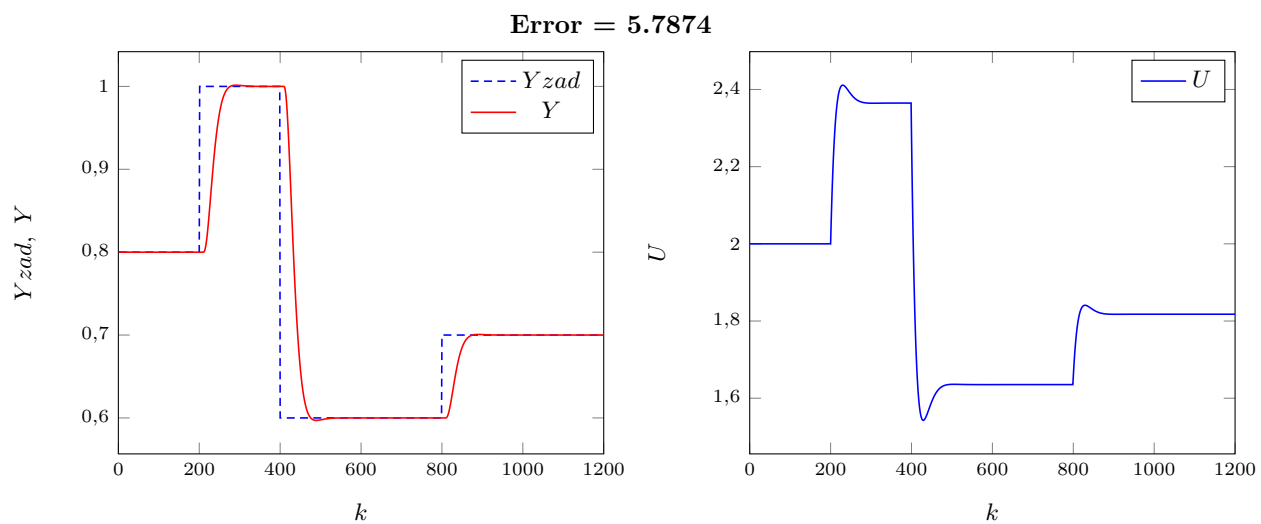
Rys. 5.31. Przebieg oraz sterowanie dla parametrów  $N = 75$ ,  $Nu = 15$ ,  $\lambda = 0.2$ Rys. 5.32. Przebieg oraz sterowanie dla parametrów  $N = 75$ ,  $Nu = 15$ ,  $\lambda = 1$ Rys. 5.33. Przebieg oraz sterowanie dla parametrów  $N = 75$ ,  $Nu = 15$ ,  $\lambda = 5$

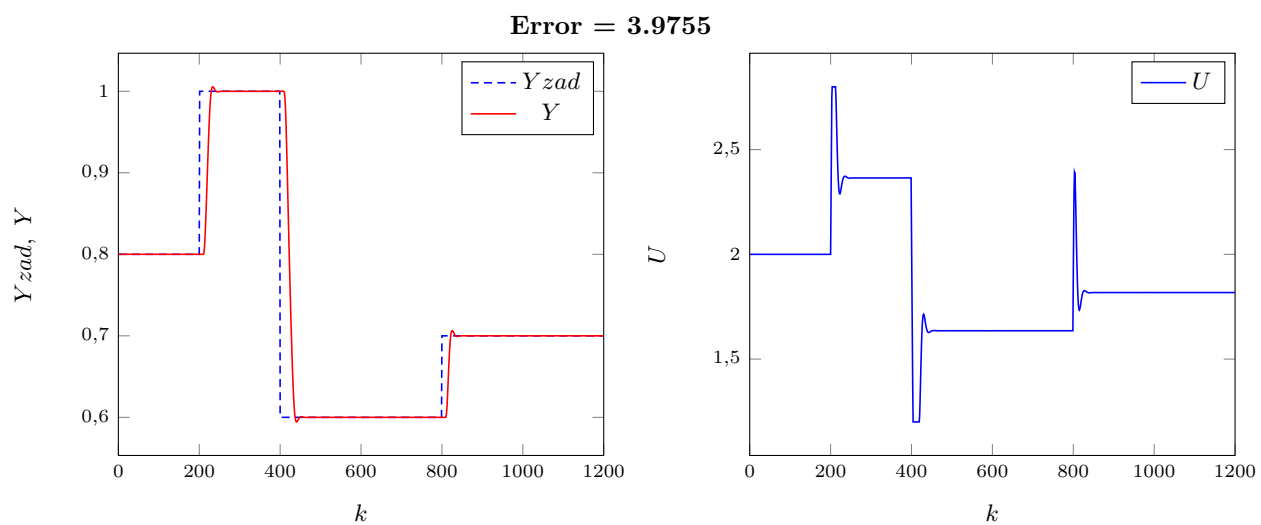
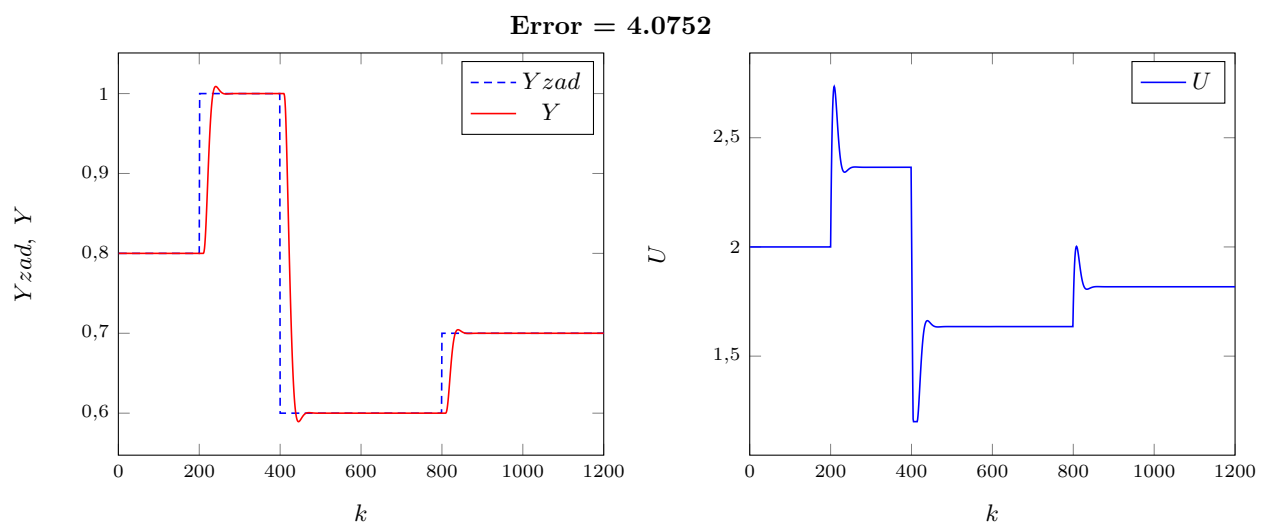
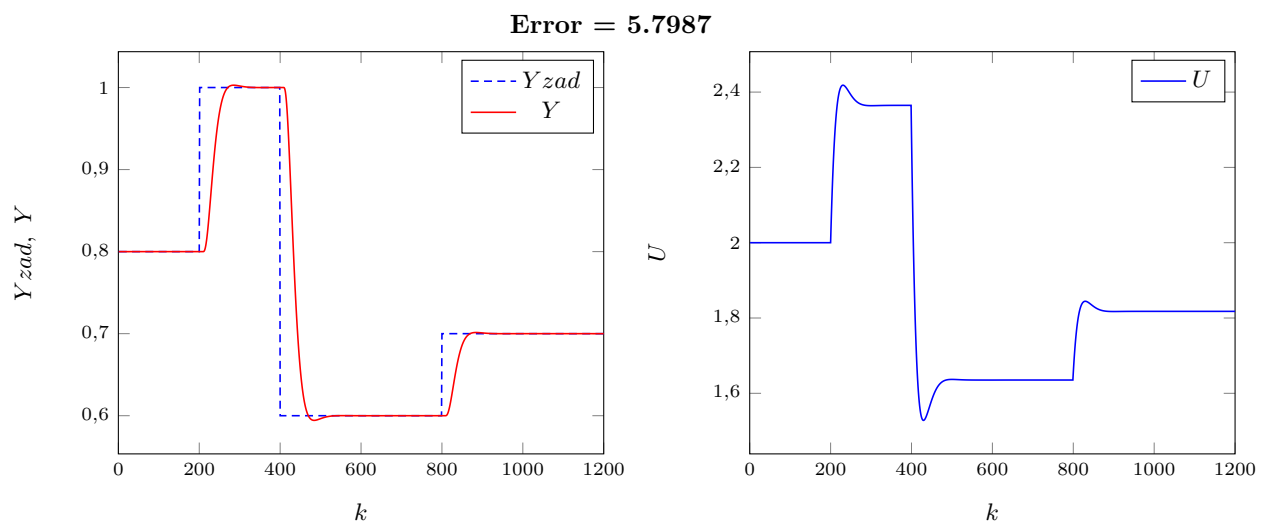
Rys. 5.34. Przebieg oraz sterowanie dla parametrów  $N = 75$ ,  $Nu = 20$ ,  $\lambda = 0.2$ Rys. 5.35. Przebieg oraz sterowanie dla parametrów  $N = 75$ ,  $Nu = 20$ ,  $\lambda = 1$ Rys. 5.36. Przebieg oraz sterowanie dla parametrów  $N = 75$ ,  $Nu = 20$ ,  $\lambda = 5$

Rys. 5.37. Przebieg oraz sterowanie dla parametrów  $N = 100$ ,  $Nu = 5$ ,  $\lambda = 0.2$ Rys. 5.38. Przebieg oraz sterowanie dla parametrów  $N = 100$ ,  $Nu = 5$ ,  $\lambda = 1$ Rys. 5.39. Przebieg oraz sterowanie dla parametrów  $N = 100$ ,  $Nu = 5$ ,  $\lambda = 5$



Rys. 5.40. Przebieg oraz sterowanie dla parametrów  $N = 100$ ,  $Nu = 10$ ,  $\lambda = 0.2$ Rys. 5.41. Przebieg oraz sterowanie dla parametrów  $N = 100$ ,  $Nu = 10$ ,  $\lambda = 1$ Rys. 5.42. Przebieg oraz sterowanie dla parametrów  $N = 100$ ,  $Nu = 10$ ,  $\lambda = 5$

Rys. 5.43. Przebieg oraz sterowanie dla parametrów  $N = 100$ ,  $Nu = 15$ ,  $\lambda = 0.2$ Rys. 5.44. Przebieg oraz sterowanie dla parametrów  $N = 100$ ,  $Nu = 15$ ,  $\lambda = 1$ Rys. 5.45. Przebieg oraz sterowanie dla parametrów  $N = 100$ ,  $Nu = 15$ ,  $\lambda = 5$

Rys. 5.46. Przebieg oraz sterowanie dla parametrów  $N = 100$ ,  $Nu = 20$ ,  $\lambda = 0.2$ Rys. 5.47. Przebieg oraz sterowanie dla parametrów  $N = 100$ ,  $Nu = 20$ ,  $\lambda = 1$ Rys. 5.48. Przebieg oraz sterowanie dla parametrów  $N = 100$ ,  $Nu = 20$ ,  $\lambda = 5$

### 5.3. Automatyczne strojenie regulatora DMC

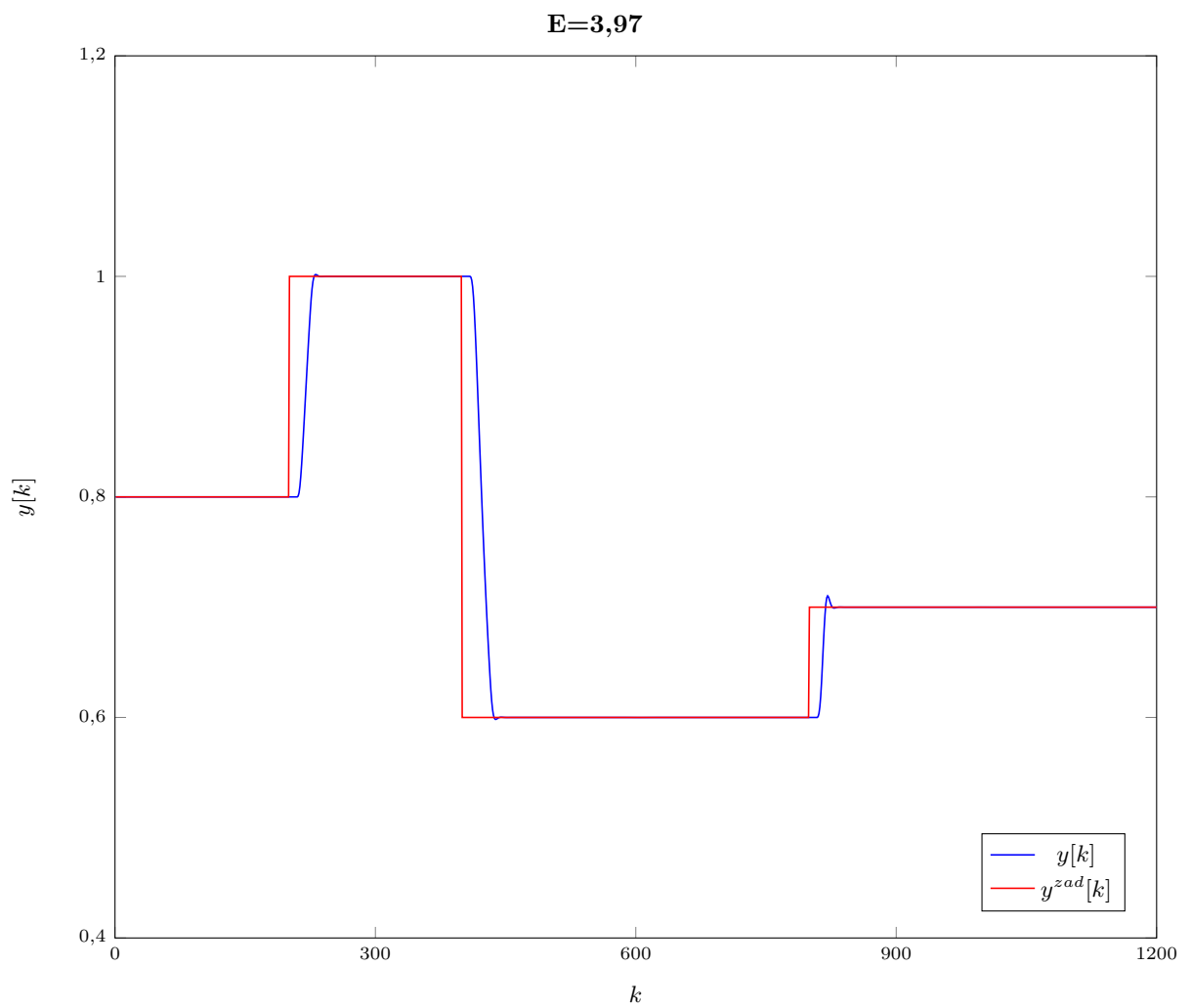
Do automatycznego strojenia regulatora PID użyto już wcześniej używanej funkcji `doDMC.m`, której użyto jako funkcji liczącej wartość wskaźnik jakości (funkcja ją zwraca) w maltabowskim algorytmie `fmincon`, której użyto w skrypcie `DMCOptimizer.m`. Przeprowadzono dwie próby: dla dwóch różnych zestawów parametrów początkowych, co widać w listingu ?? poniżej, pamiętając o ograniczeniach na wielkości parametrów (wielkości horyzontów są zaokrąglane w funkcji `doDMC.m`). Wyniki przedstawiono na rysunkach ?? i ??.

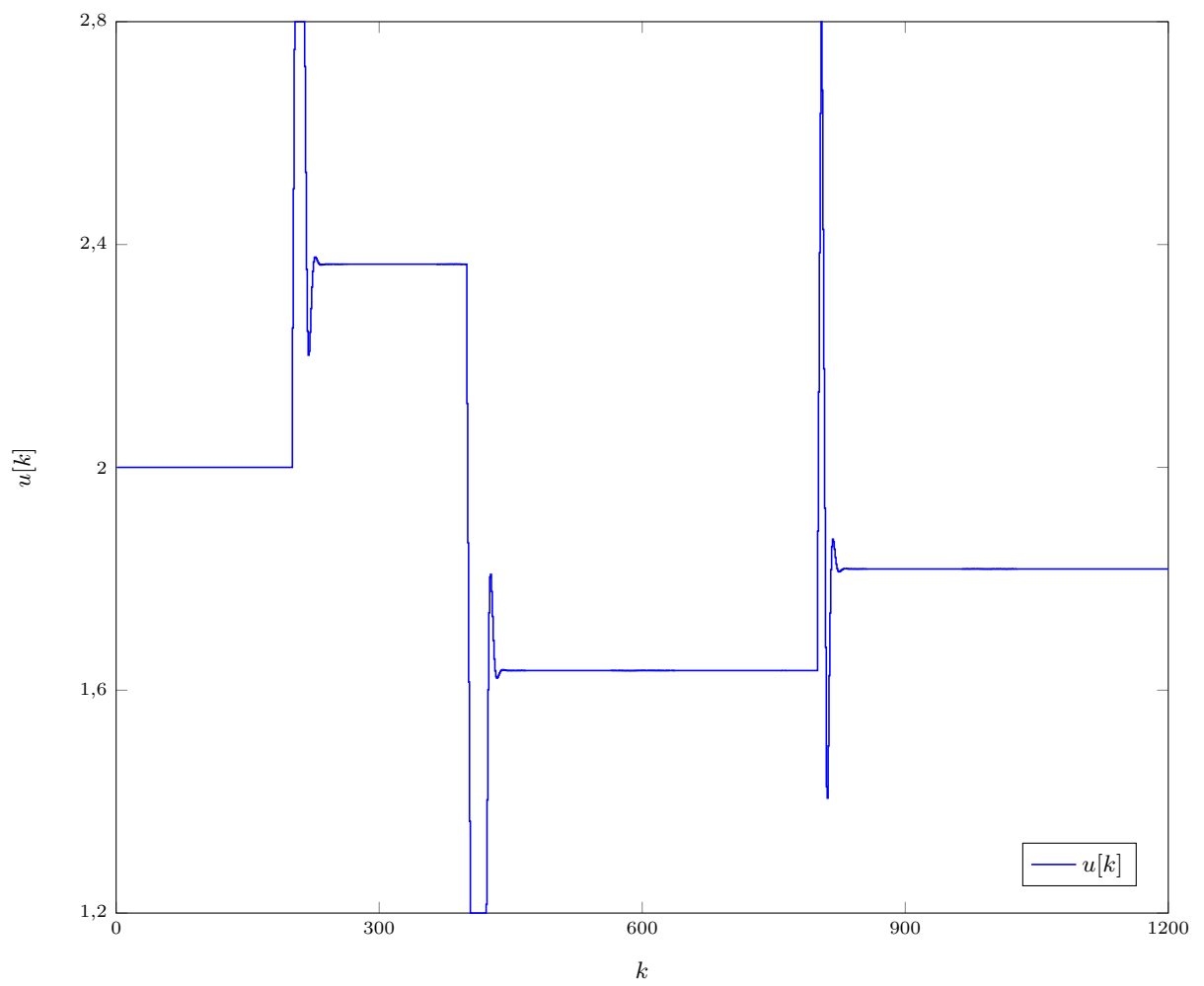
```
x01=[15,15,0.1];
x02=[100,100,100];

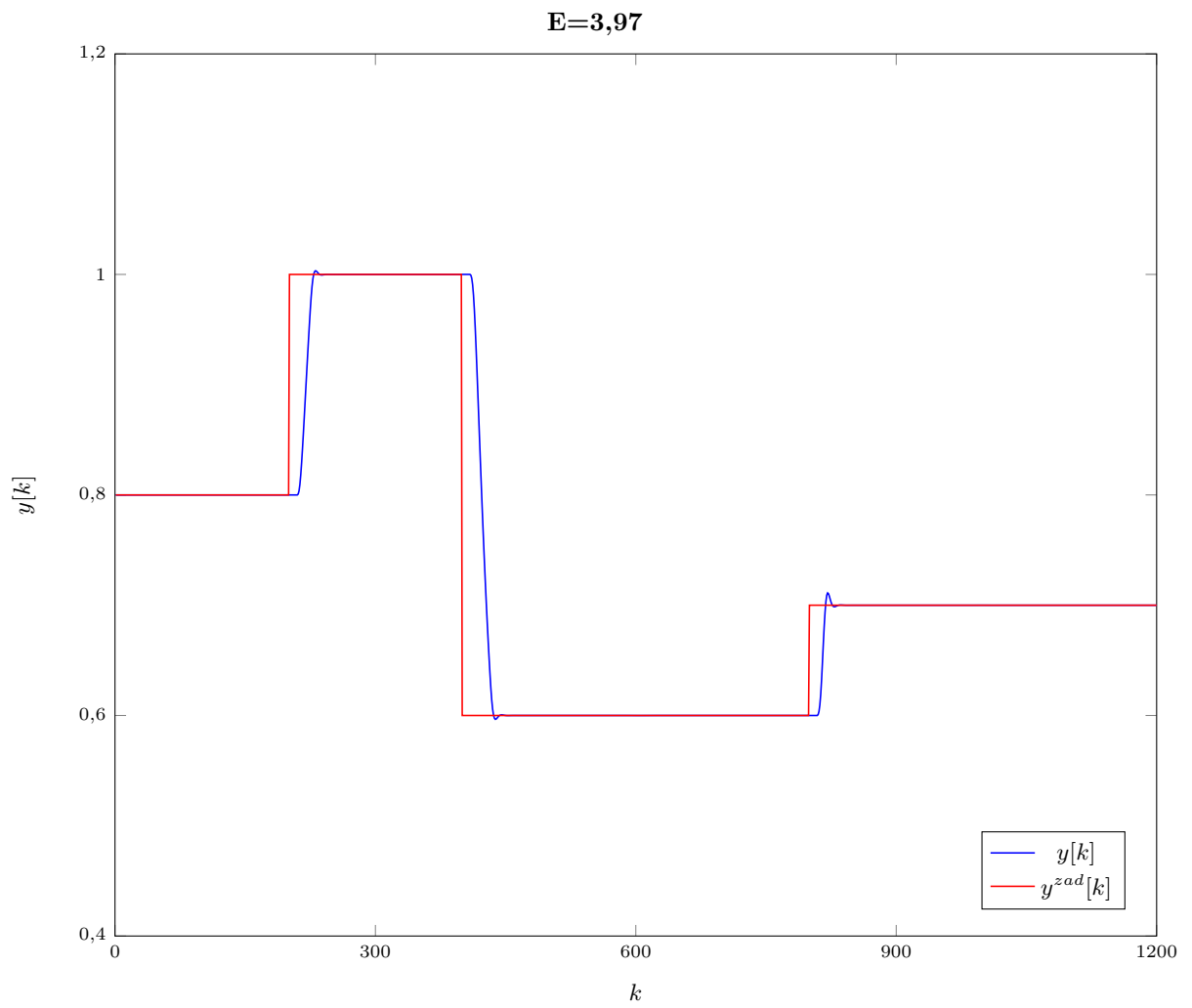
% N>=Nu
A=[-1,1,0];
b=[0];
% N, Nu <= D=120
lb=[1,1,0.01];
ub=[120,120,100];
[paras,error]=fmincon(@doDMC,x01,A,b,[],[],lb,ub);

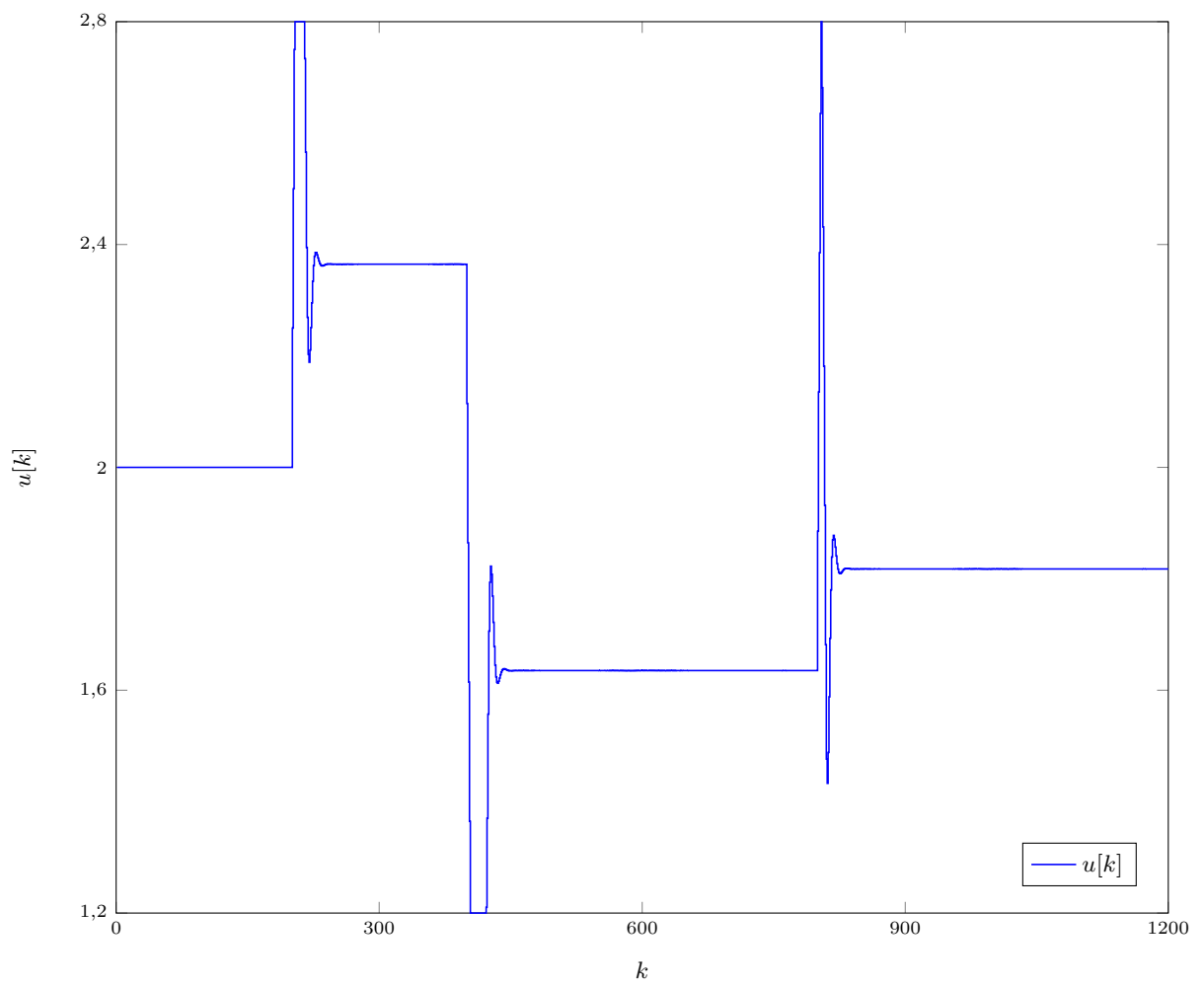
\label{LIST_2}
```

Listing 5.2. Implementacja automatycznego strojenia DMC

Rys. 5.49. Wyjście dla  $N = 25$ ,  $Nu = 5$ ,  $\lambda = 0,043$

Rys. 5.50. Sterowanie dla  $N = 25$ ,  $N_u = 5$ ,  $\lambda = 0,043$

Rys. 5.51. Wyjście dla  $N = 105$ ,  $Nu = 80$ ,  $\lambda = 0,050$

Rys. 5.52. Sterowanie dla  $N = 105$ ,  $N_u = 80$ ,  $\lambda = 0,050$



#### 5.4. Obserwacje i wnioski

Automatycznie dostrojony DMC okazał się lepszy (zgodnie z przewidywaniami), jednak interesujący jest fakt, iż otrzymano dwa tak samo dobre regulatory dla zupełnie różnych horyzontów sterowania i predykcji. Wynika z tego, że tak naprawdę jedyny parametr, która znacząco (oczywiście poza bardzo krótkimi horyzontami predykcji) jest wartością kary za sterowanie  $\lambda$ .

Część II

## Laboratoria