Wydział Elektroniki i Technik Informacyjnych
Politechnika Warszawska

Projektowanie układów sterowania (projekt grupowy)

Sprawozdanie z projektu i ćwiczenia laboratoryjnego nr 1, zadanie nr 6

Michał Szpunar, Kacper Michalski, Mateusz Palkowski

Warszawa, 2020

Spis treści

1. Wstęp

1.1. Cel projektu

Celem projektu było zbadanie właściwości danego obiektu oraz próba regulacji z wykorzystaniem dyskretnych algorytmów PID oraz DMC w wersji analitycznej. Częścią zadania było również uwzględnienie ograniczeń sterowania narzuconych w treści projektu.
Symulacja obiektu odbywała się za pośrednictwem funkcji `symulacja_obiektu6Y`. Ustalono punkt pracy dla wartości $U_{\mathrm{PP}} = 1{,}1, Y_{\mathrm{PP}} = 2{,}5$, natomiast ograniczenia wartości sygnału sterującego miały wartości $U^{\mathrm{min}} = 0{,}6, U^{\mathrm{max}} = 1{,}6$.

1.2. Opis algorytmów

1.2.1. PID

W zadaniu projektowym wykorzystany został regulator PID. Algorytm ten, na podstawie obliczonej wartości uchybu oraz dobranych nastaw, wyznacza wartość sterowania dla chwili k. Elementami struktury algorytmu są następujące stałe:

— K - stała proporcjonalna
— T_i - stała całkowania
— T_d - stała różniczkowania
— T - czas próbkowania

Dobranie nastaw algorytmu oznacza znalezienie możliwie optymalnych nastaw zapewniających najlepszą jakość regulacji.

Po wyznaczeniu parametrów, należy obliczyć wpółczynniki prawa regulacji używając następujących wzorów:

$$r_2 = \frac{KTd}{T} \tag{1.1}$$

$$r_1 = K(\frac{T}{2T_i} - \frac{2T_d}{T} - 1) \tag{1.2}$$

$$r_0 = K(\frac{T}{2T_i} + \frac{T_d}{T} + 1) \tag{1.3}$$

Prawo regulacji rgulatora opisane jest równaniem:

$$u(k) = r_2 e(k-2) + r_1 e(k-2) + r_0 e(k) + u(k-1) \tag{1.4}$$

1.2.2. DMC

Regulator DMC jest algorytmem predykcyjnym wyznaczjącym trajektorię sygnału wyjściowego oraz przyszłe przyrosty sterowań. DMC potrzebuje wcześniejszej informacji o obiekcie w postaci odpowiedzi skokowej. Parametrami algorytmu są:

— D - horyzont dynamiki
— N - horyzont predykcji
— N_u - horyzont sterownia

— λ - kara za zmianę sterownia

Strojenie algorytmu polega na odpowiednim dobraniu parametrów tak, by zapewnić możliwie najlepszą jakość regulacji.

Aby otrzymać prawo regulacji, należy wyznaczyć szereg współczynników:
Macierz dynamiczną oraz macierz K:

$$M = \begin{bmatrix} s_1 & 0 & \cdots & 0 \\ s_2 & s_1 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ s_N & s_{N-1} & \cdots & s_{N-N_u+1} \end{bmatrix} \tag{1.5}$$

$$K = (M^T \Psi M + \Lambda)^{-1} M^T \Psi \tag{1.6}$$

Macierz M^P oraz wektor zmian sterowania ΔU^P:

$$M^P = \begin{bmatrix} s_2 - s_1 & s_3 - s_2 & \cdots & s_D - s_{D-1} \\ s_3 - s_1 & s_4 - s_2 & \cdots & s_{D+1} - s_{D-1} \\ \vdots & \vdots & \ddots & \vdots \\ s_{N+1} - s_1 & s_{N+2} - s_2 & \cdots & s_{N+D-1} - s_{D-1} \end{bmatrix} \tag{1.7}$$

$$\Delta U^P(k) = \begin{bmatrix} \Delta u(k-1) \\ \Delta u(k-2) \\ \vdots \\ \Delta u(k-(D-1)) \end{bmatrix} \tag{1.8}$$

Na podstawie powyższych macierzy oraz wektorów, można obliczyć parametry reguatora:

$$k_e = \sum_{i=1}^{N} K_{1,i} \tag{1.9}$$

$$k_u = \overline{K}_1 M^P \tag{1.10}$$

a następnie wyznaczyć sterowanie z następującego prawa regulacji:

$$e(k) = y_{zad}(k) - y(k) \tag{1.11}$$

$$u(k|k) = u(k-1) + k_e e(k) - k_u \Delta U^P(k) \tag{1.12}$$

Ograniczenie wartości sygnału sterującego przez wartości maksymalną i minimalną wykonane jest w następujący sposób:

1. jeżeli $u(k|k) < u_{min}$ wtedy $u(k|k) = u_{min}$
2. jeżeli jeżeli $u(k|k) > u_{max}$ wtedy $u(k|k) = u_{max}$ max
3. $u(k) = u(k|k)$

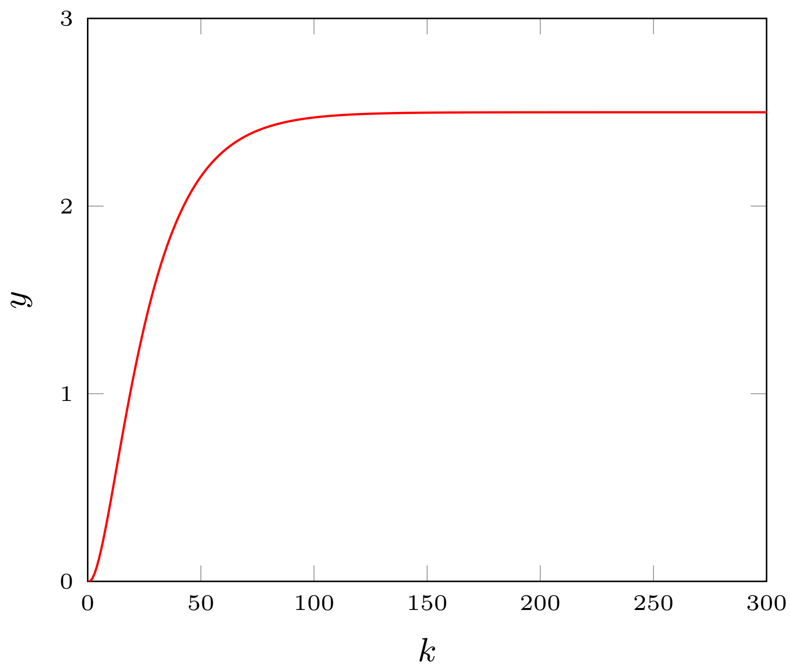
2. Sprawdzenie poprawności wartości początkowych U_{PP} oraz Y_{PP}

Aby sprawdzić poprawność założonego punktu pracy, przyjęto, że sterowanie obietku na czas symulacji będzie stałe, równe U_{PP}. Po czasie ustalenia się odpowiedzi obiektu, zbadano wartość wyjścia oraz porównano ją z Y_{PP}.

Poniżej zamieszczono kod programu użytego do rozwiązania zadania:

```
%wyznaczanie y_pp dla u_pp = 1.1
u_pp = 1.1;

t_sim = 300;
y = [0;0];

for k=2:t_sim
    y_temp = symulacja_obiektu6Y(u_pp,u_pp,y(k),y(k-1));
    y = [y;y_temp];
end

y_pp = y(end);
```

Po ustalonym czasie symulacji `t_sim = 300` zbadano przebieg sygnału wyjściowego.



Rys. 2.1. Odpowiedź układu dla ustalonego $u = U_{\mathrm{PP}}$

Wartość, na której ustalił się sygnał wyjściowy to 2,5, co potwierdza poprawność założonego punktu pracy.

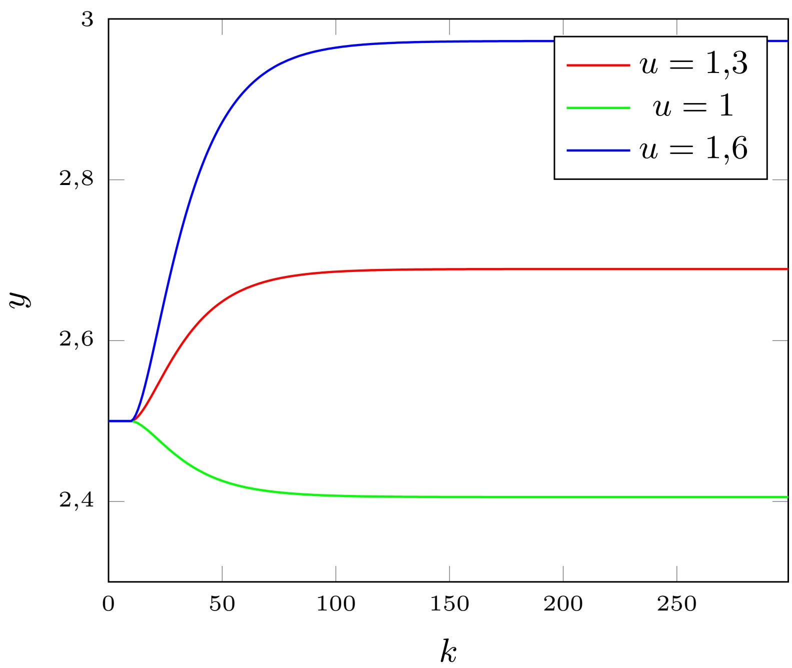
3. Badanie obiektu

3.1. Symulacja różnych odpowiedzi skokowych obiektu

Poniżej zamieszczono kod programu przeprowadzającego symulację obiektu dla ustalonegych u z przedziału $\langle U^{\min}, U^{\max}\rangle$ uzyskując różne odpowiedzi skokowe.

```
t_sim2 = 300;

%konstruowanie sygnalow sterujacych
step_tim = 0;

u_base = ones(1,step_tim)*u_pp;
u_step_temp = ones(1,t_sim2 - step_tim)*1.3;
u_step2_temp = ones(1,t_sim2 - step_tim);
u_step3_temp = ones(1,t_sim2 - step_tim)*1.6;

u_step = [u_base, u_step_temp];
u_step2 = [u_base, u_step2_temp];
u_step3 = [u_base, u_step3_temp];

y = ones(t_sim2, 1)*y_pp;
y2 = ones(t_sim2, 1)*y_pp;
y3 = ones(t_sim2, 1)*y_pp;

for k = 3:t_sim2
    if k-11 <= 0
        y(k) = symulacja_obiektu6Y(u_pp,u_pp,y(k-1),y(k-2));
        y2(k) = symulacja_obiektu6Y(u_pp,u_pp,y2(k-1),y2(k-2));
        y3(k) = symulacja_obiektu6Y(u_pp,u_pp,y3(k-1),y3(k-2));
    else
        y(k) = symulacja_obiektu6Y(u_step(k-10),u_step(k-11),y(k-1),y(k-2));
        y2(k) = symulacja_obiektu6Y(u_step2(k-10),u_step2(k-11),y2(k-1),y2(k-2));
        y3(k) = symulacja_obiektu6Y(u_step3(k-10),u_step3(k-11),y3(k-1),y3(k-2));
    end
end
```

W wyniku przeprowadzenia symulacji, otrzymano następujące przebiegi:



Rys. 3.1. Odpowiedzi układu dla ustalonegych u z przedziału $\langle U^{\min}, U^{\max}\rangle$

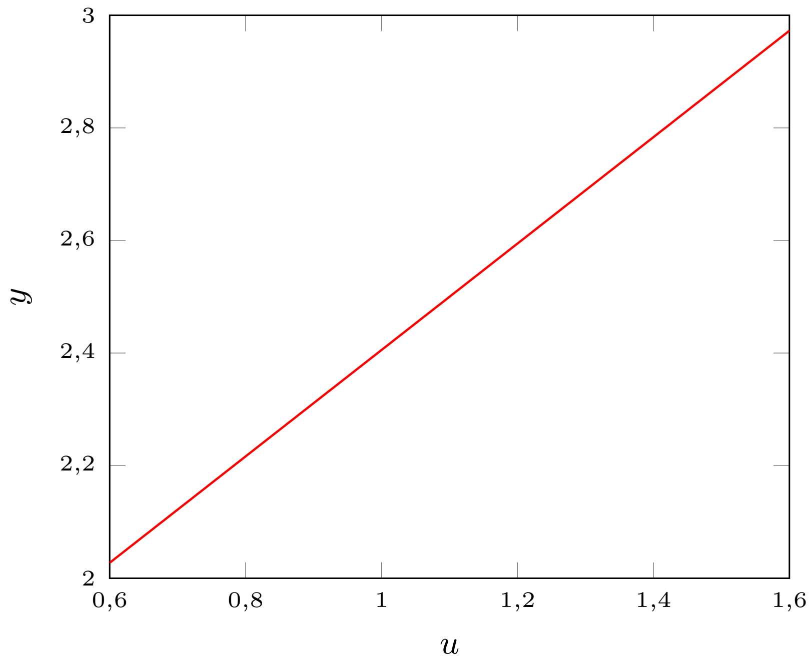
Na pierwszy rzut oka, obiekt wydaje się mieć liniową strukturę. Aby jednak to potwierdzić, należy wyznaczyć charakterystykę statyczną.

3.2. Charakterystyka statyczna

Aby wyznaczyć charakterystykę statyczną, stworzono symulację dla równiomiernie rozłożonych wartości sterowania z zakresu $\langle U^{\min}, U^{\max} \rangle$. Dla każdej z wartości sterowania wykonywano skok, a następnie czekano, aż wartość sygnału wyjściowego się ustabilizuje. Kod symulacji zamieszczony został poniżej:

```
t_sim = 400;
y_wyj = [];

for u = 0.6:0.01:1.6
    y = [0;0];
    for k=2:t_sim
        y_temp = symulacja_obiektu6Y(u,u,y(k),y(k-1));
        y = [y;y_temp];
    end
    y_wyj = [y_wyj; y(end)];
end
```

Poniżej zaprezentowany został wynik symulacji: Wyraźnie widać, że charakterystyka statyczna obiektu jest liniowa. Można zatem obliczyć wzmocnienie statyczne obiektu mając na uwadzę zależność:



Rys. 3.2. Przybliżona charakterystyka statyczna układu

$$K_{\mathrm{stat}} = \frac{Y^{\max} - Y^{\min}}{U^{\max} - U^{\min}} \tag{3.1}$$

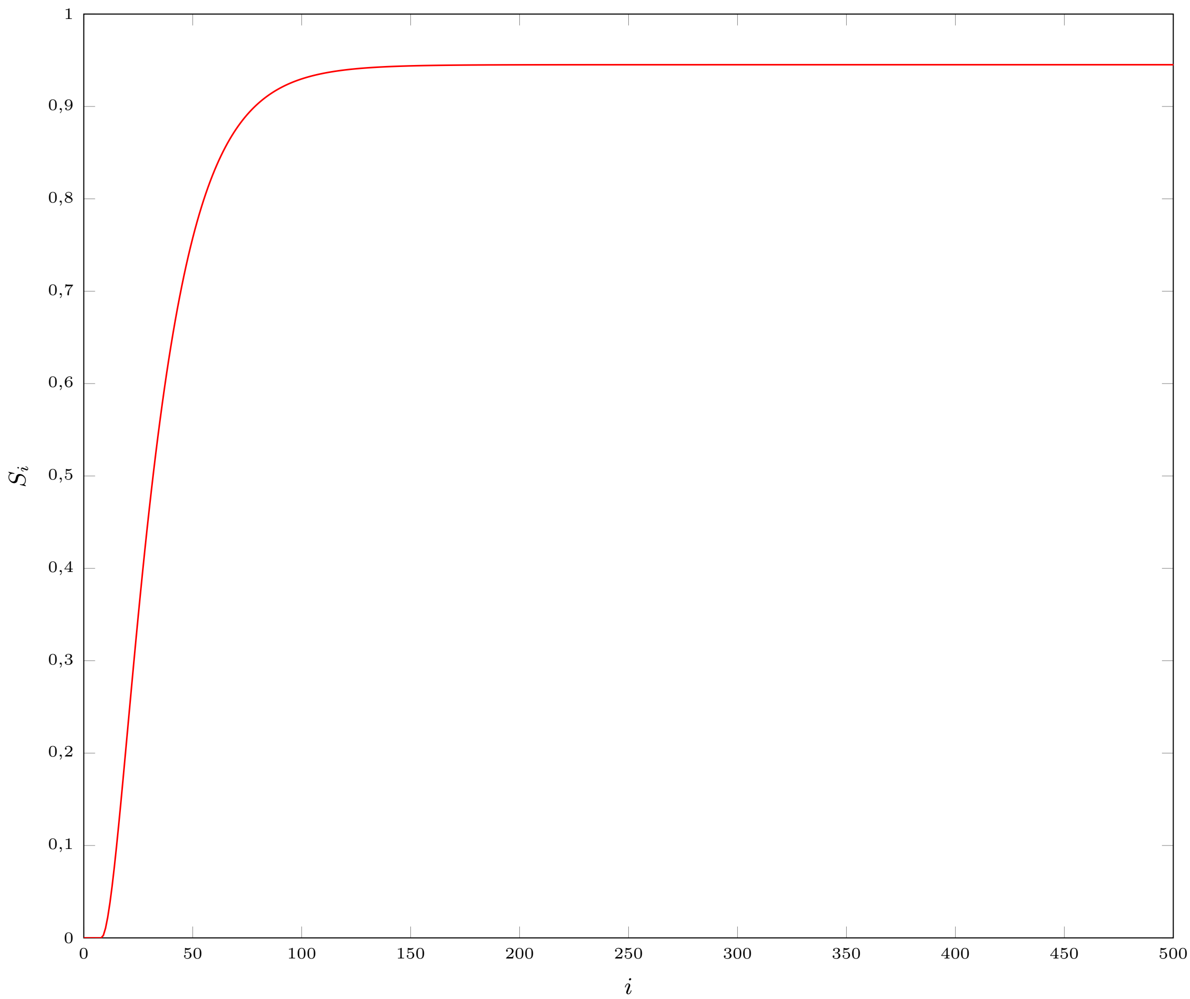
Na podstawie powyższych rozważań obliczono $K_{\mathrm{stat}} = 0{,}936$

4. Odpowiedź skokowa

Odpowiedź skokowa to odpowiedź układu na wymuszenie w postaci skoku jednostkowego w chwili $k = 0$. Z powodu ograniczeń sterowania obiektu, skok jednostkowy jest niemożliwy, dlatego należy przeskalować odpowiedź skokową w następujący sposób:

$$\triangle S_i = \frac{S_i^0 - Y_{pp}}{\triangle U}, \text{ dla } i = 1, \ldots, N \tag{4.1}$$

gdzie N to wybrana długość odpowiedzi skokowej. Odpowiedź skokowa prezentowana na wykresie 4.1 została otrzymana z przesklowania odpowiedzi obiektu po skoku z $u_{pp} = 1{,}1$ do $u = 1{,}4$.



Rys. 4.1. Odpowiedź skokowa

5. Implementacja PID

Poniższy listing zawiera implementacje dyskretnego regulatora PID w w języku `Matlab`. W liniach 18, 19 i 20 obliczane są parametry r_0, r_1 i r_2, które następnie są wykorzystywane razem z uchybami z chwil k, $k-1$ i $k-2$ do obliczenia przyrostu sterowania $\triangle U$ w lini 47. W związku z istnieniem ograniczeń w postaci

$$-\triangle U^{\max} \leqslant \triangle U(k) \leqslant \triangle U^{\max} \tag{5.1}$$

gdzie $\triangle U^{max} = 0{,}1$ oraz

$$0{,}6 \leqslant U(k) \leqslant 1{,}6 \tag{5.2}$$

należy przycinać sygnał wyjściowy regulatora. Dzieje się to w liniach 50-60. Aby ograniczyć liczbę instrukcji `if` najpierw ograniczany jest przyrost sygnału sterującego $\triangle U$, a potem wartość sygnału sterującego U.

```
1 % Ograniczenia
2 du_max = 0.1;
3 u_min = 0.6;
4 u_max = 1.6;
5 % Punkt pracy
6 u_pp = 1.1;
7 y_pp = 2.5;
8 % Wskaznik jakosci
9 piderr = 0;
10 %Okres regulacji
11 T=1;
12 % Nastawy PID
13 params = [4.4817,   14.4945,    6.8265];
14 K = params(1);
15 Ti = params(2);
16 Td = params(3);
17
18 r0 = K * (1 + (T/(2*Ti)) + (Td/T));
19 r1 = K * ((T/(2*Ti)) - (2*Td/T) - 1);
20 r2 = K*Td/T;
21
22 % Czas symulacji
23 t_sim = 800;
24 % Zadana trajektoria
25 y_zad = ones(t_sim, 1) * 2.7;
26 y_zad(100:250) = 2.9;
27 y_zad(250:400) = 2.7;
28 y_zad(400:600) = 2.4;
29 % Inicjalizacja wektorow y, u, e
30 y = ones(t_sim, 1) * y_pp;
31 u = ones(t_sim, 1) * u_pp;
32 e = zeros(t_sim, 1);
```

```
% Petla, w ktorej odbywa sie symulacja
for k = 3:t_sim
    % Symulacja obiektu
    if k-11 <= 0
        y(k) = symulacja_obiektu6Y(u_pp,u_pp,y(k-1),y(k-2));
    else
        y(k) = symulacja_obiektu6Y(u(k-10),u(k-11),y(k-1),y(k-2));
    end
    % Obliczenie uchybu
    e(k) = y_zad(k) - y(k);
    % Obliczenie wskaznika jakosci
    piderr = piderr + e(k)^2;
    % Obliczenie przyrostu sterowania
    du = r2*e(k-2) + r1*e(k-1) + r0*e(k);

    % Nalozenie ograniczen
    if du>du_max
        du = du_max;
    elseif du<-du_max
        du = -du_max;
    end
    uk = u(k-1) + du;
    if uk>u_max
        uk = u_max;
    elseif uk<u_min
        uk = u_min;
    end
    u(k) = uk;
end
```

6. Implementacja DMC

Zaimplementowany regualtor to DMC w wersji analitycznej „oszczędnej", czyli w każdej chwili liczona jest tylko obecny przyrost sterowania, a nie cały wektor przewidywanych przyrostów. Poniższy fragment kodu zawiera inicjalizacje potrzebnych parametrów D, N, N_u i λ oraz oblicza macierze, które są wyznaczane „offline", czyli $\boldsymbol{M}$, $\boldsymbol{M}_\mathrm{p}$ i $\boldsymbol{K}$. Na ich podstawie obliczane są k_e i $\boldsymbol{k}_\mathrm{u}$. DMC również obowiązują ogarniczenia, są zaimplementowane dokładnie tak samo jak w przypadku PID.

```
% Ograniczenia
du_max = 0.1;
u_min = 0.6;
u_max = 1.6;
% Punkt pracy
u_pp = 1.1;
y_pp = 2.5;
% Nastawy
D = 200;
N = 32;
Nu = 3;
lambda = 1;
% Macierz M
M = zeros(N,Nu);
for i = 1:size(M,1)
    for j = 1:size(M,2)
        if i>=j
            M(i,j) = s(i-j+1);
        end
    end
end
% Macierz Mp
Mp = zeros(N,D-1);
for i = 1:size(Mp,1)
    for j = 1:size(Mp,2)
        if i+j<D
            Mp(i,j) = s(i+j) - s(j);
        else
            Mp(i,j) = s(D) - s(j);
        end
    end
end
% Macierz K
K = ((M'*M + lambda*eye(Nu))^-1)*M';
ke = sum(K(1,:));
ku = zeros(D-1,1);
for i = 1:D-1
```

```
    ku(i) = K(1,:) * Mp(:,i);
end
```

Poniższy fragment zawiera główną pętle symulacji. W lini 20 obliczany jest przyrost sterowania, na który zostają nałożone ograniczenia. Zmienna `current_sum` obliczana w lini 16 i używana do obliczenia przyrostu sterowania to składnik sumy składającej się na przyrost sterowania

$$\boldsymbol{k}_{\mathrm{u}}\triangle\boldsymbol{U}^{\mathrm{P}}(k) \tag{6.1}$$

```
1  % Petla, w ktorej odbywa sie symulacja
2  for k = 3:t_sim4
3      % Symulacja obiektu
4      if k-11 <= 0
5          y(k) = symulacja_obiektu6Y(u_pp,u_pp,y(k-1),y(k-2));
6      else
7          y(k) = symulacja_obiektu6Y(u(k-10),u(k-11),y(k-1),y(k-2));
8      end
9      % Obliczenie uchybu
10     e(k) = y_zad(k) - y(k);
11     dmcerr = dmcerr + e(k)^2;
12     % Obliczenie sumy
13     current_sum = 0;
14     for i = 1:D-1
15         if k-i > 1
16             current_sum = current_sum + ku(i) * du(k-i);
17         end
18     end
19     % Obliczenie przyrostu sterowania
20     duk = ke*e(k) - current_sum;
21 
22     % Nalozenie ograniczen
23     if duk>du_max
24         duk = du_max;
25     elseif duk<-du_max
26         duk = -du_max;
27     end
28     du(k) = duk;
29     uk = u(k-1) + duk;
30     if uk>u_max
31         uk = u_max;
32     elseif uk<u_min
33         uk = u_min;
34     end
35     u(k) = uk;
36 end
```

Działanie obu regulatorów zostanie zaprezentowane w kolejnych punktach.

7. zad5

Zadanie 5 polegało na dobraniu nastaw regulatora PID i DMC metodą eksperymentalną w oparciu o przebiegi i wskaźnik jakości regulacji E:

$$E = \sum_{k=1}^{k_{\mathrm{konc}}} (y^{\mathrm{zad}}(k) - y(k))^2 \tag{7.1}$$

dla zaproponowanej przez nas trajektorii zmian sygnału zadanego y^{zad} (Rys 6.1.).

Ponieważ wskaźnik E zliczany był już w poprzednim zadaniu, kod Matlabowy będzie tu taki sam jak w zadaniu 4.

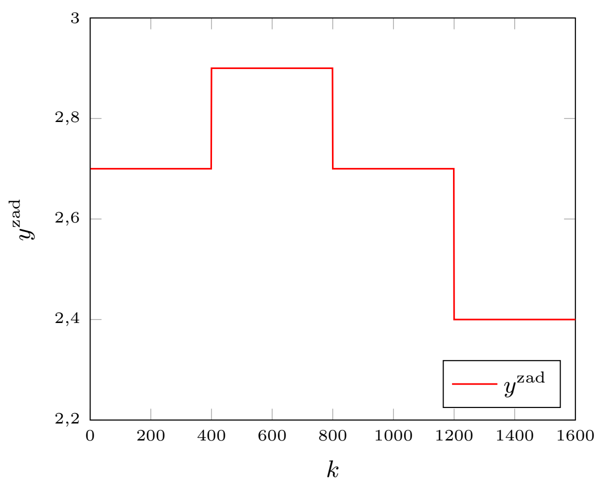
7.1. Eksperymentalne dobranie parametrów PID

Do eksperymentalnego dobrania parametrów PID użyto nastepującej metody eksperymentalnej:

1. Zwiekszanie wpływu członu P przy wyłączonych członach I i D, aż dla pojedyńczego skoku wartości zadanej, wartość regulowana będzie wykonywała stałe, niegasnące oscylacje wokół wartości zadanej.
2. Dla wartości równej połowie uzyskanej w kroku pierwszym i wyłączonym członie D, dobranie wartości T_{i} dającej zadowalające wyniki.
3. Dla wartości K i T_{i} dobranych w kroku drugim, dobranie wartości T_{d} dającej zadowalające wyniki.

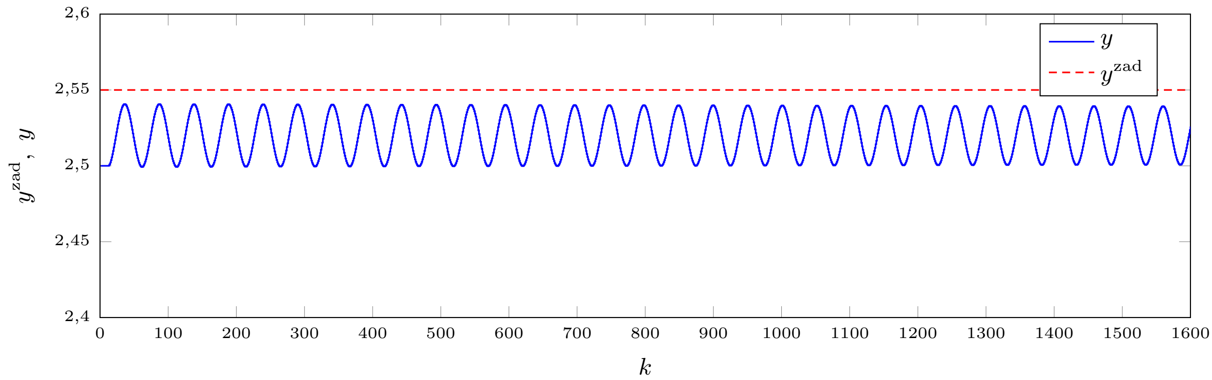
Stałe oscylacje obiektu osiągnięto dla $K_{\mathrm{kryt}} = 3{,}95$ (Rys 6.2.).

Warto zaznaczyć, że z powodu ograniczeń, każda wartość oscylacji powodująca ich rośnięcie doprowadzi w końcu do stałych oscylacji - należy więc być ostrożnym z dobieraniem parametru K, żeby parametr u nie został obcięty przez minimalizację, gdyż mogłoby to fałszywie sugerować stałość oscylacji.

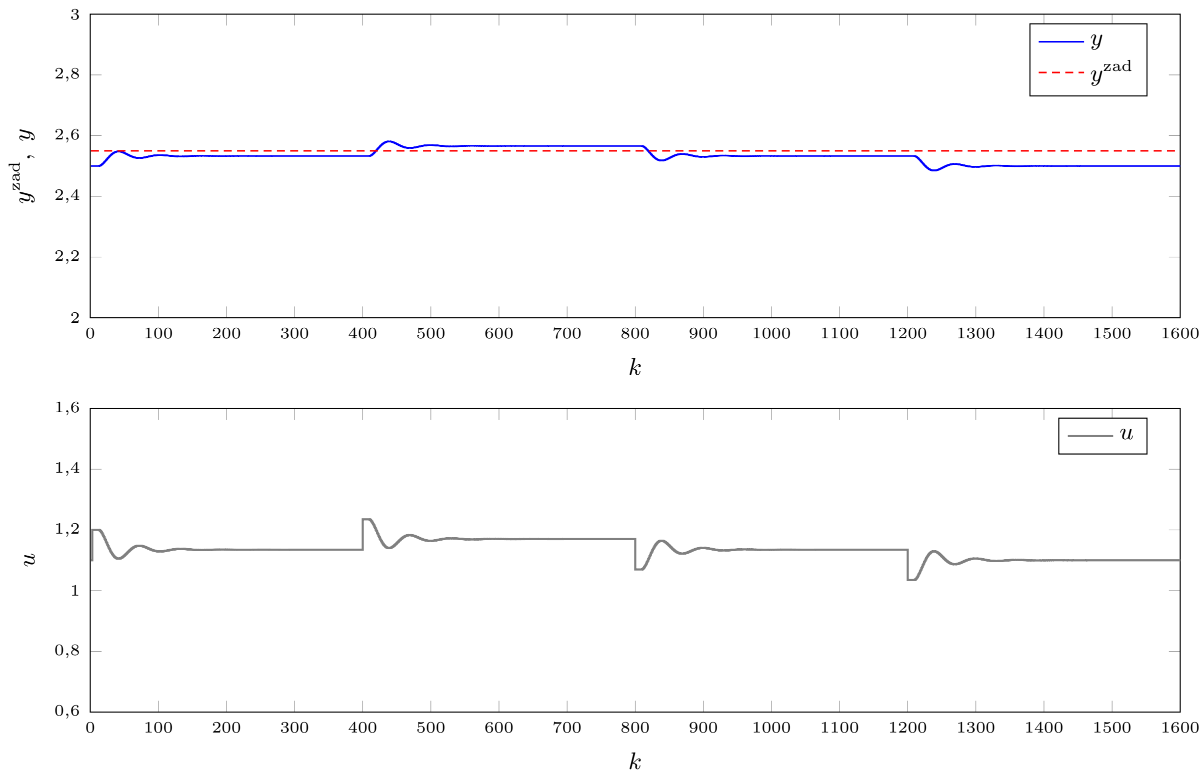


Rys. 7.1. Trajektoria zmian sygnału y^{zad} na której przeprowadzimy symulacje

Dla $K = \frac{1}{2}K_{\mathrm{kryt}}$ wyznaczonego z oscylacji otrzymujemy przebieg z Rys 6.3.
Uchyb ustalony jest zdecydowanie z wysoki, żeby uznać regulację za zadowalającą. Dobierze-



Rys. 7.2. Stałe (w pewnym przybliżeniu) oscylacje wartości wyjściowej y

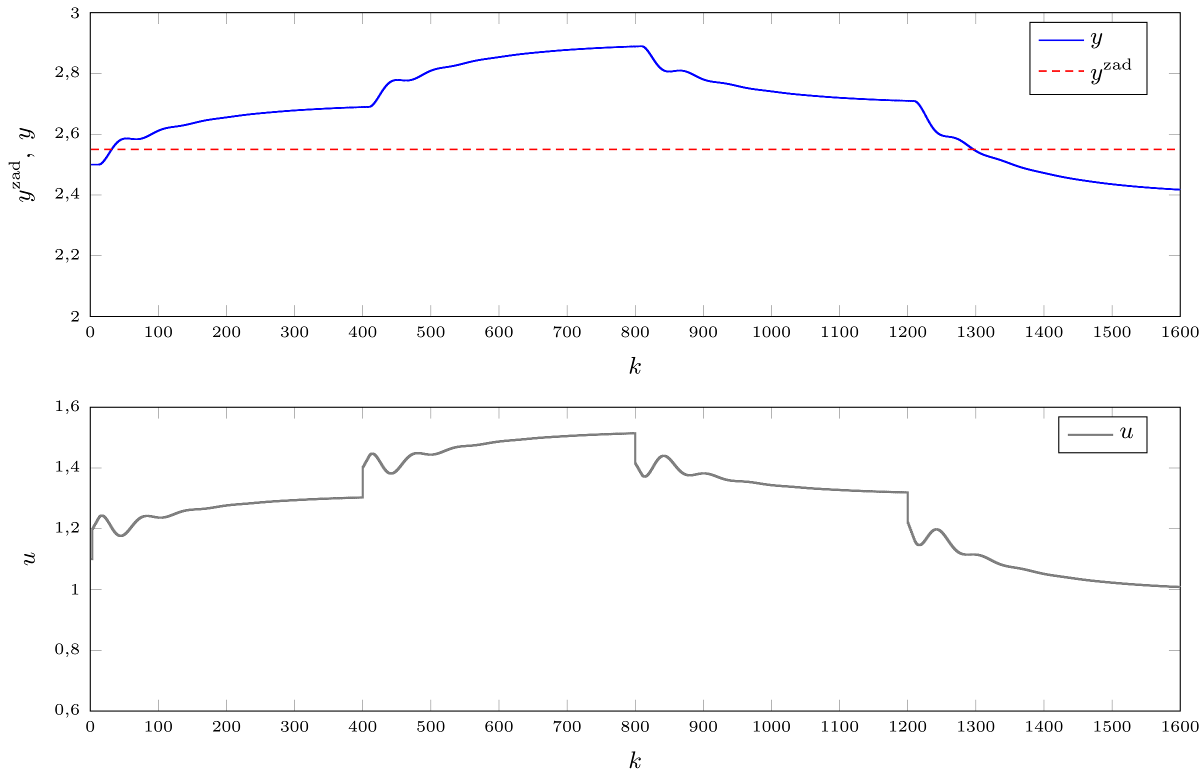


Rys. 7.3. Regulator PID dla $K = 1{,}975$, $T_{\mathrm{i}} = inf$, $T_{\mathrm{d}} = 0$

my więc teraz parametr T_i włączając tym samym człon całkujący. Powinno pomóc to zredukować uchyb ustalony. Na rysunkach 6.4 - 6.12 przedstawiono regulację dla różnych, eksperymentalnie dobieranych wartości parametru T_i.

Widzimiy, że najmniejszy błąd E wystąpił dla $T_{/mathrmi} = 24$. Warto też zauważyć, że dla mniejszych wartości T_i zaczęły występować wolno gasnące oscylacje wartości wyjściowej. Jak widać jednak na Rys 6.9, oscylacje te są na tyle małe i gasną wystarczająco szybko, że można je tolerować. Następnym krokiem jest dobranie parametru T_d. Dobierany był dla parametru $K = 1{,}975$ i parametru $T_\mathrm{i} = 24$.

Jak widać z tabelki dodanie członu różniczkującego zwiększyło wartość błędu E. Na rysunkach widać jednak, że wystepujące wcześniej oscylacje znacznie zmniejszyły się dzięki działaniu członu, co jest pożądane, zwłaszcza dla obiektów rzeczywistych. Można dobrać różne wartości T_d w zależności od pożądanego tłumienia oscylacji, jako kompromis między tłumieniem a wskaźnikiem jakości E wybrane zostało $T_\mathrm{d} = 3$ (Rys 6.17). Ostateczne parametry PID otrzymane w wyniku

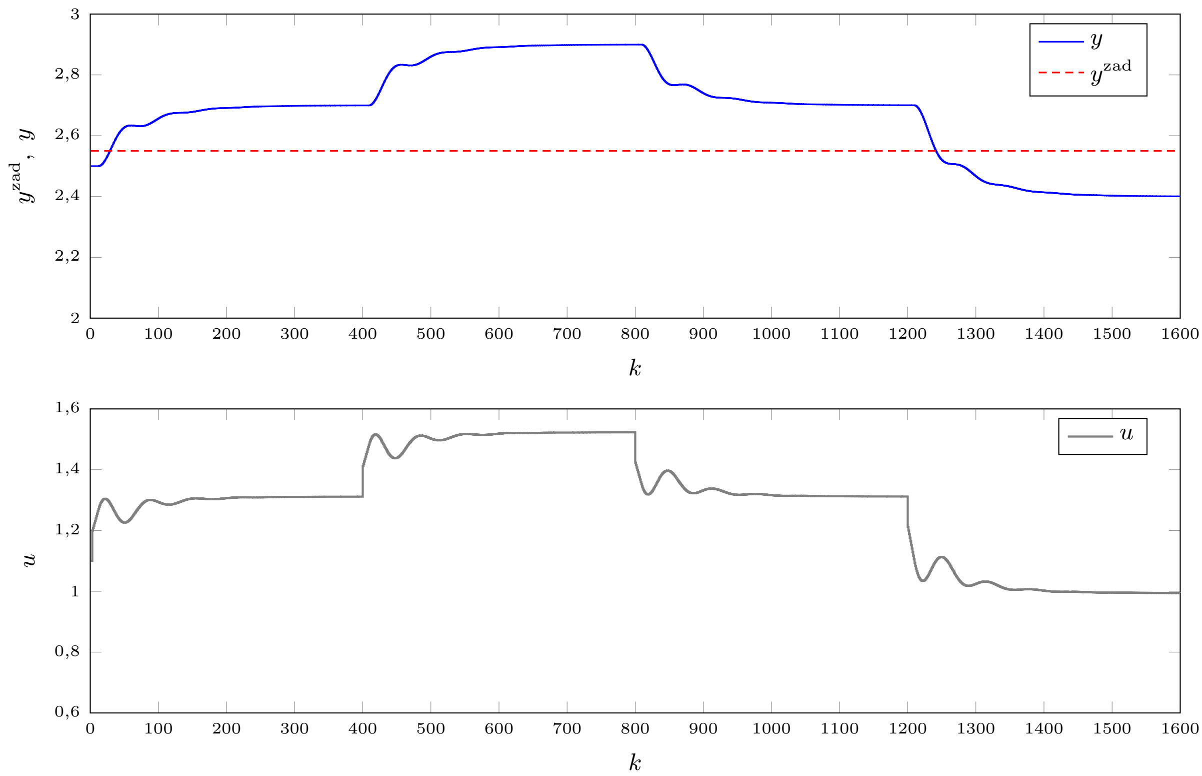


Rys. 7.4. Regulator PID dla $K = 1{,}975$, $T_\mathrm{i} = 100$, $T_\mathrm{d} = 0$

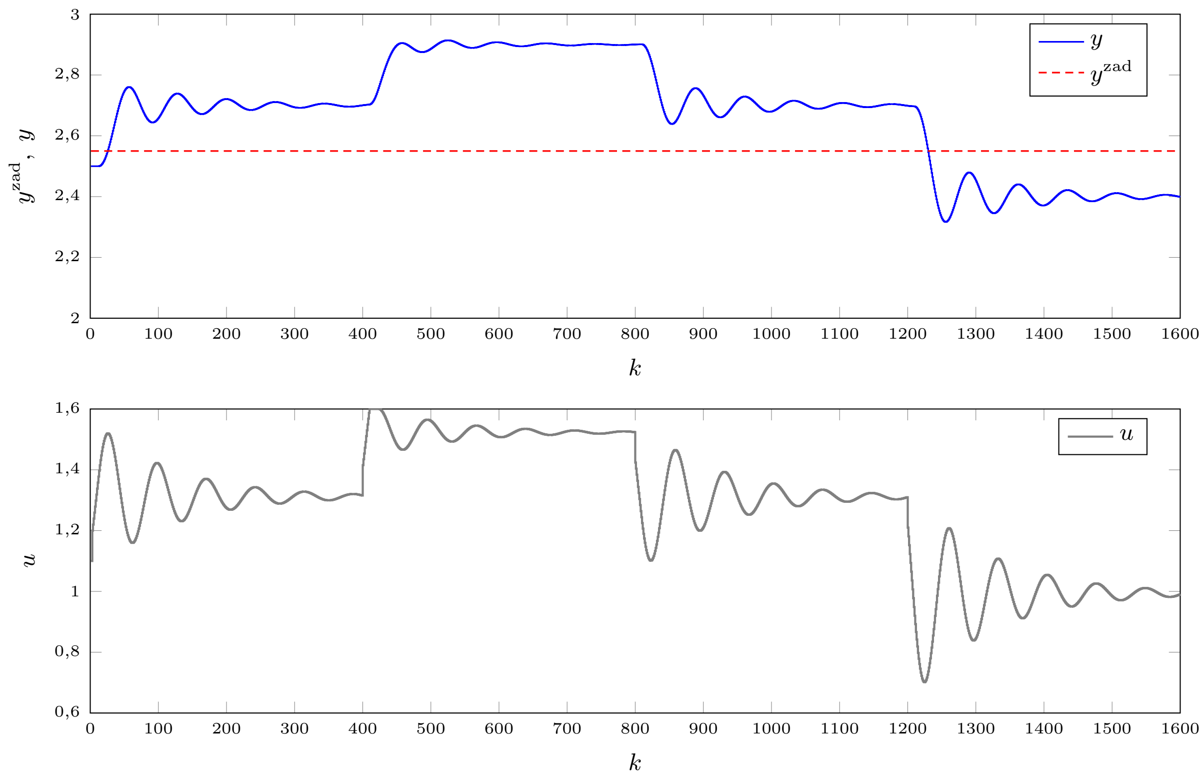
Tab. 7.1. Porównanie wielkości błędu E dla różnych wartości parametru T_i i dla parametru $K = 1{,}975$

T_i	E
inf	71,3457
100	13,9366
50	7,8122
30	5,8606
27	5,6896
25	5,6210
24	5,6081
22	5,6506
20	5,8522
10	18,3320

eksperymentów to: $K = 1{,}975$, $T_\mathrm{i} = 24$, $T_\mathrm{d} = 3$. Oczywiście nie są to pewnie idealne parametry regulatora.



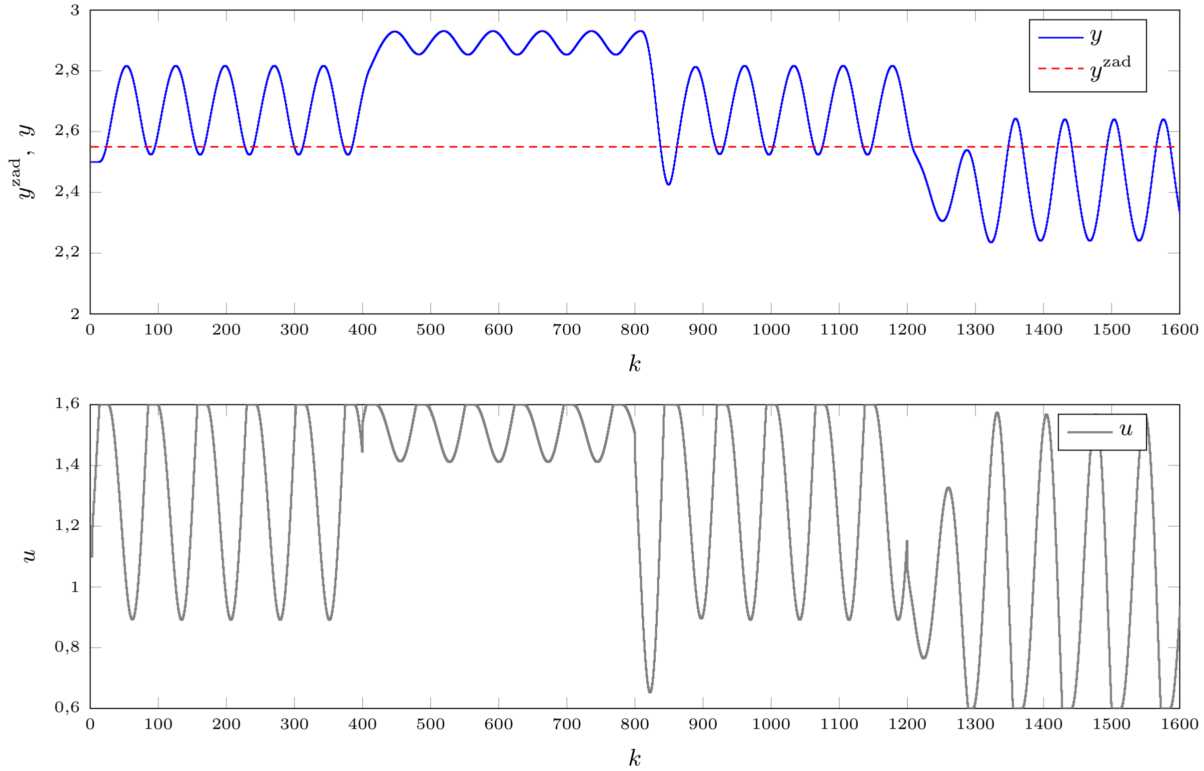
Rys. 7.5. Regulator PID dla $K = 1{,}975$, $T_\mathrm{i} = 50$, $T_\mathrm{d} = 0$



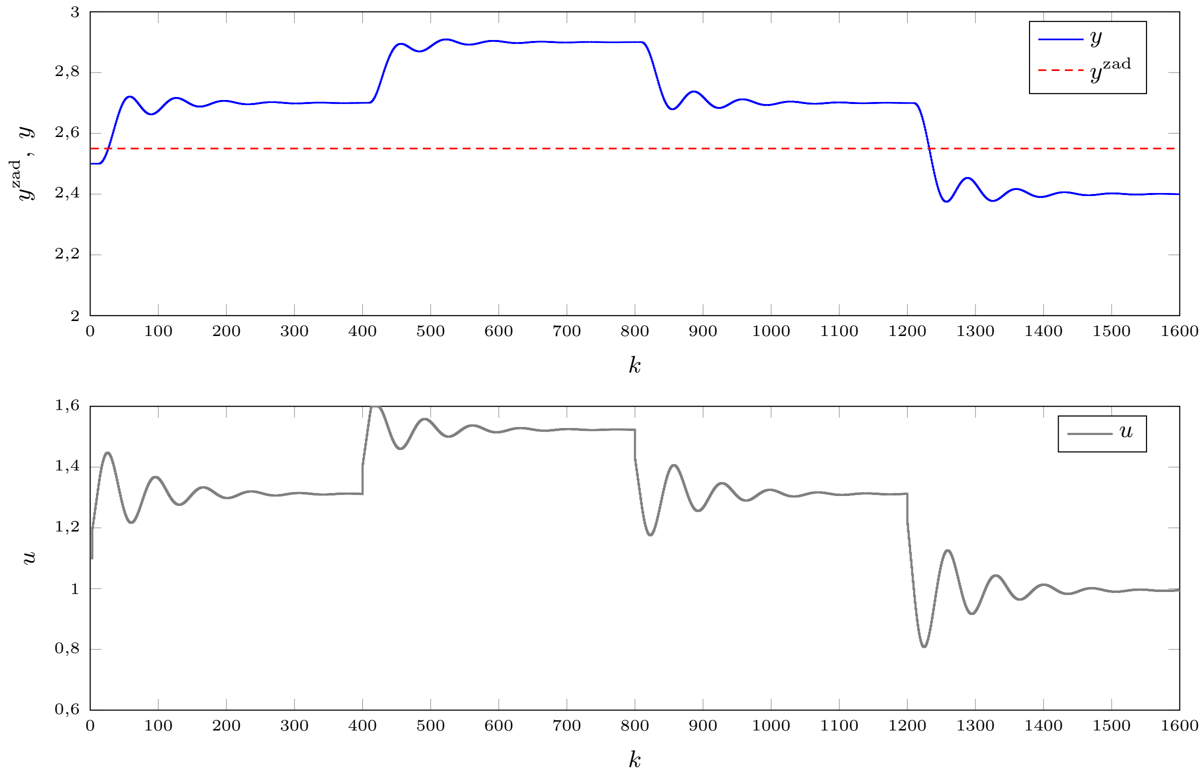
Rys. 7.6. Regulator PID dla $K = 1{,}975$, $T_\mathrm{i} = 20$, $T_\mathrm{d} = 0$

7.2. Eksperymentalne dobranie parametrów DMC

Do eksperymentalnego dobrania parametrów DMC użyto nastepującej metody eksperymentalnej:

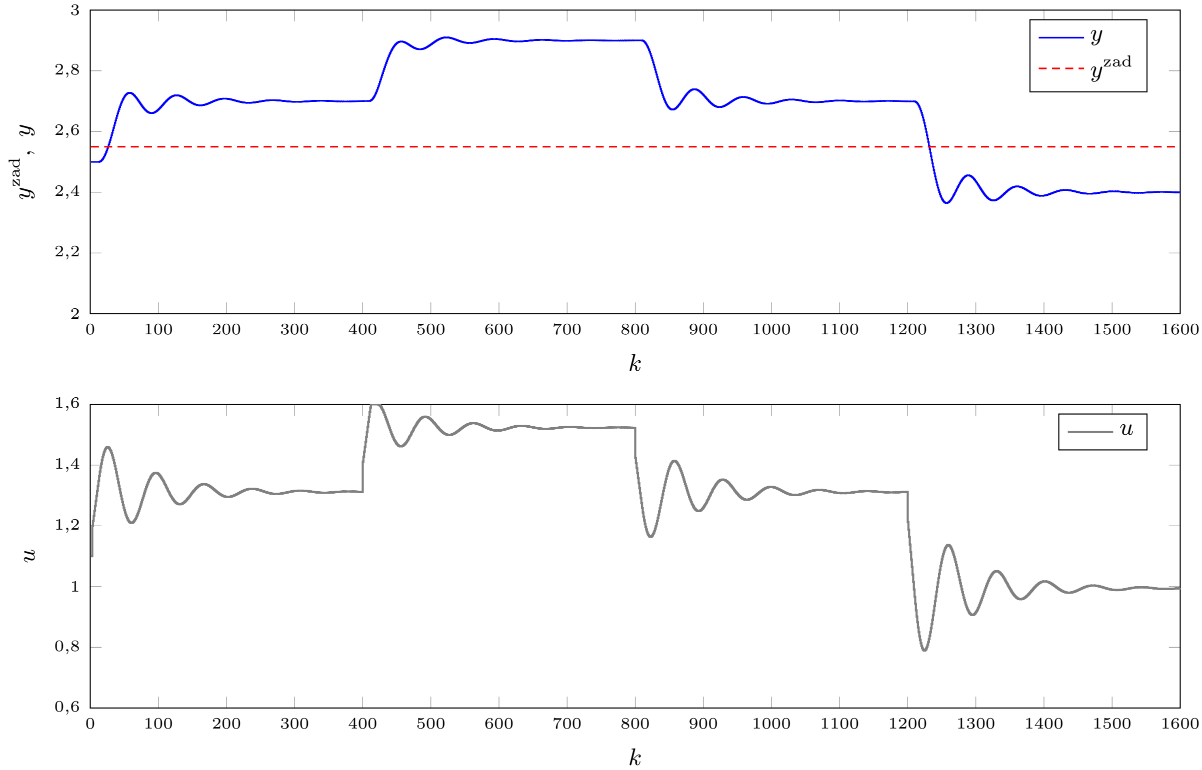


Rys. 7.7. Regulator PID dla $K = 1{,}975$, $T_\mathrm{i} = 10$, $T_\mathrm{d} = 0$

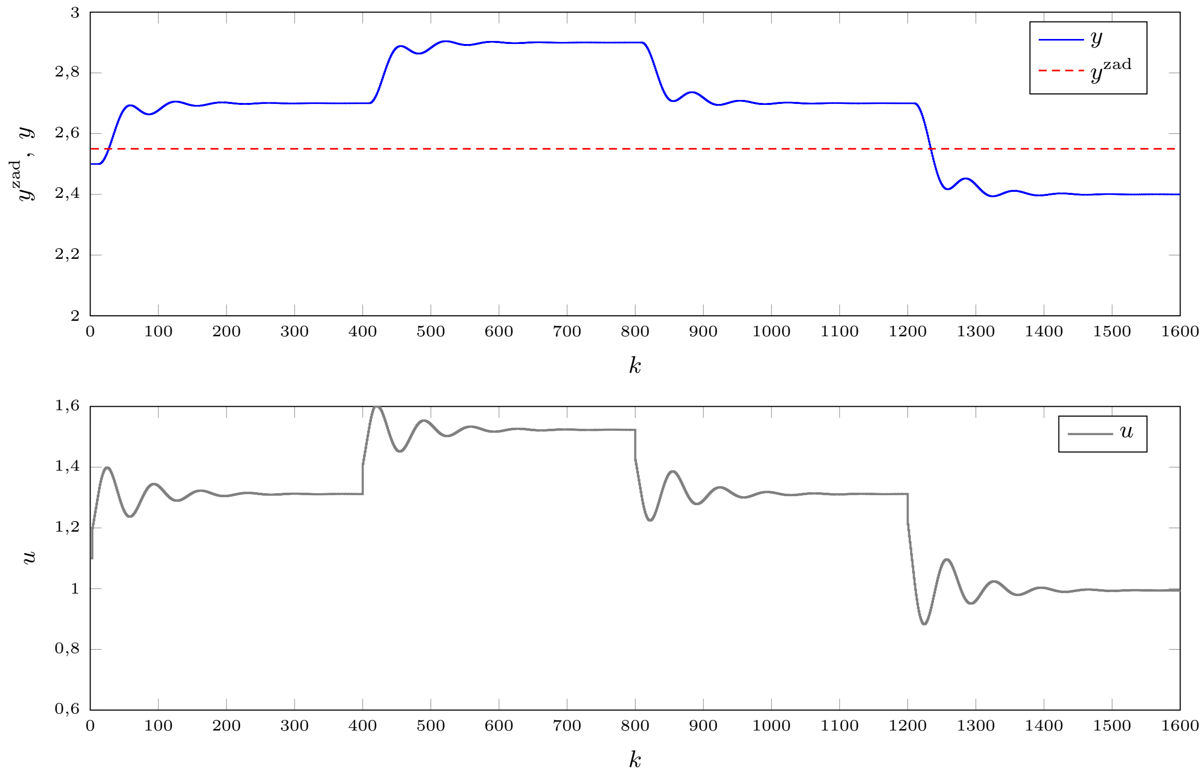


Rys. 7.8. Regulator PID dla $K = 1{,}975$, $T_\mathrm{i} = 25$, $T_\mathrm{d} = 0$

1. Przyjęcie maksymalnego możliwego parametry D, równego długości odpowiedzi skokowej aż do jej ustabilizowania, w naszym przypadku $D = 200$.

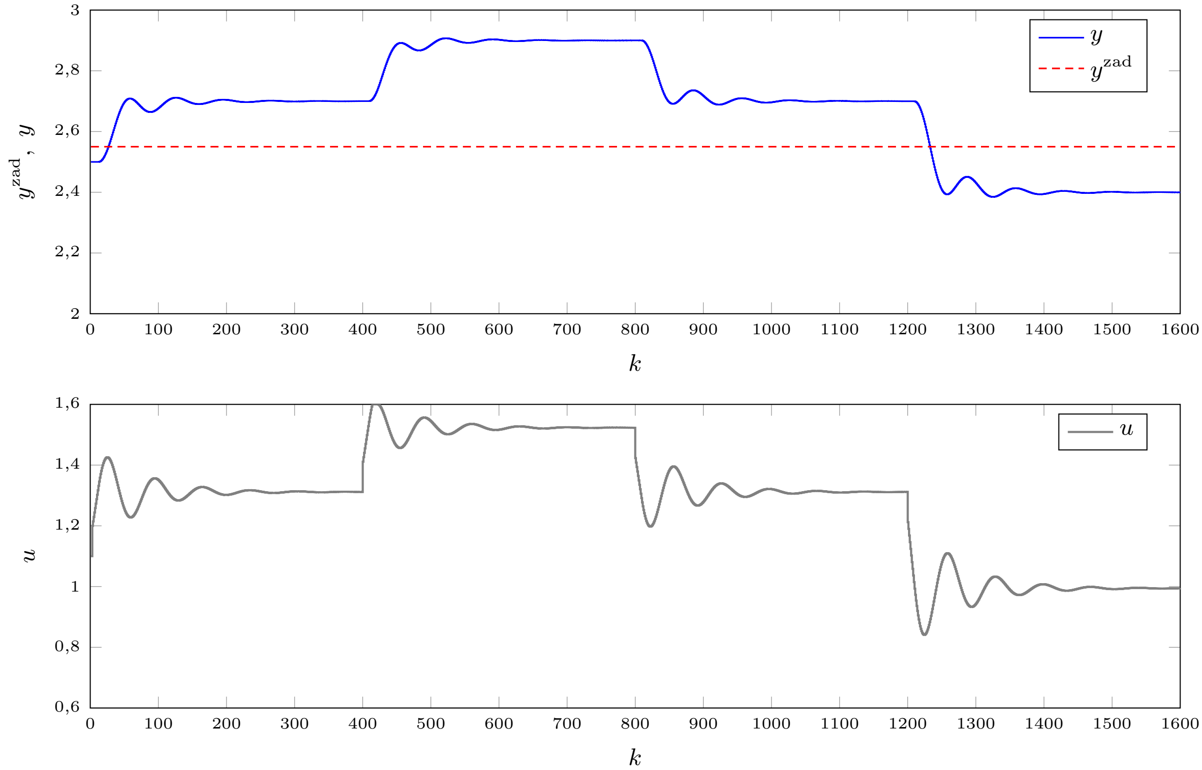


Rys. 7.9. Regulator PID dla $K = 1{,}975$, $T_\mathrm{i} = 24$, $T_\mathrm{d} = 0$

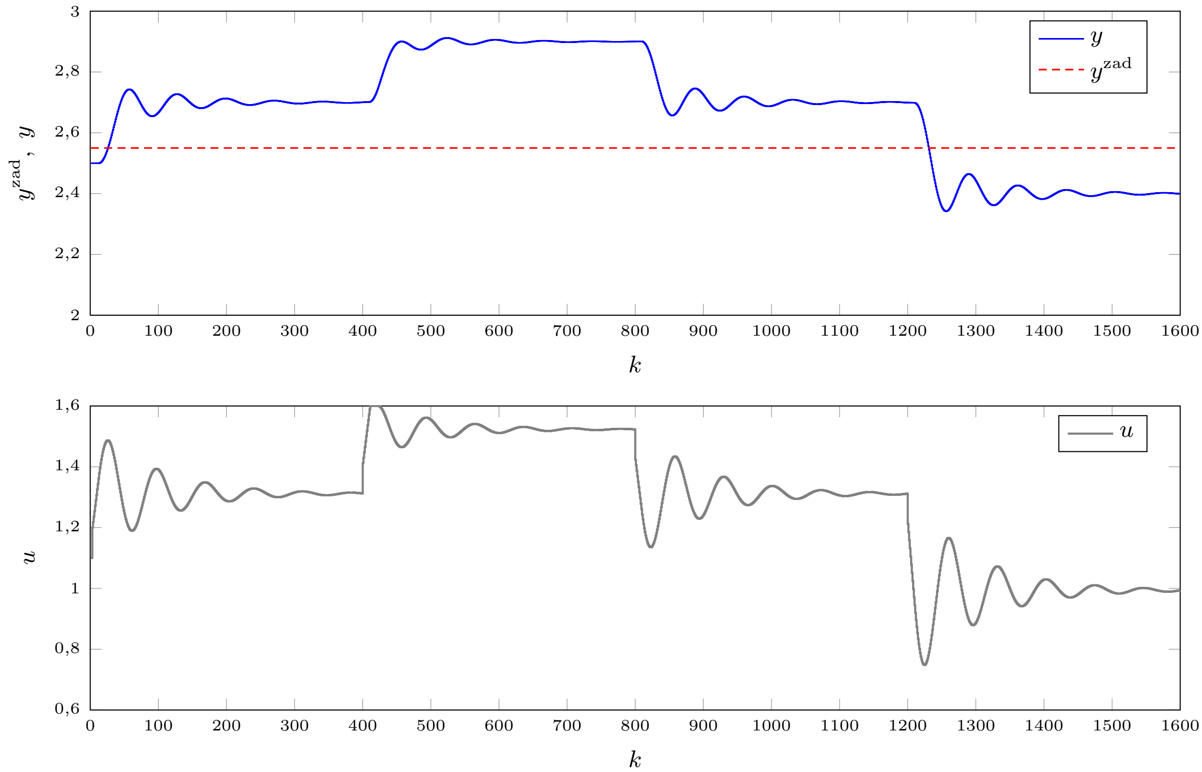


Rys. 7.10. Regulator PID dla $K = 1{,}975$, $T_\mathrm{i} = 30$, $T_\mathrm{d} = 0$

2. Dla maksymalnie dużego parametru D stopniowe zmniejszanie parametrów N i N_u (N musi byc cały czas równe N_u, aż osiągniemy zadowalające wyniki.

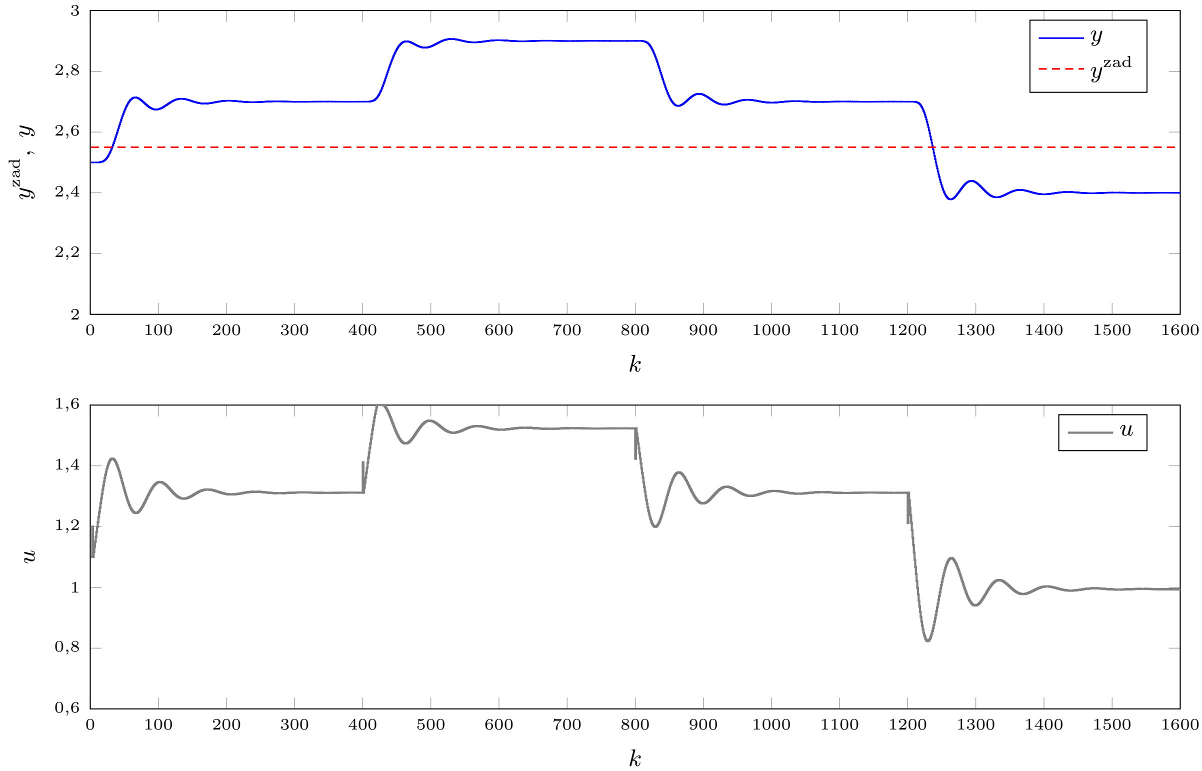


Rys. 7.11. Regulator PID dla $K = 1{,}975$, $T_i = 27$, $T_d = 0$

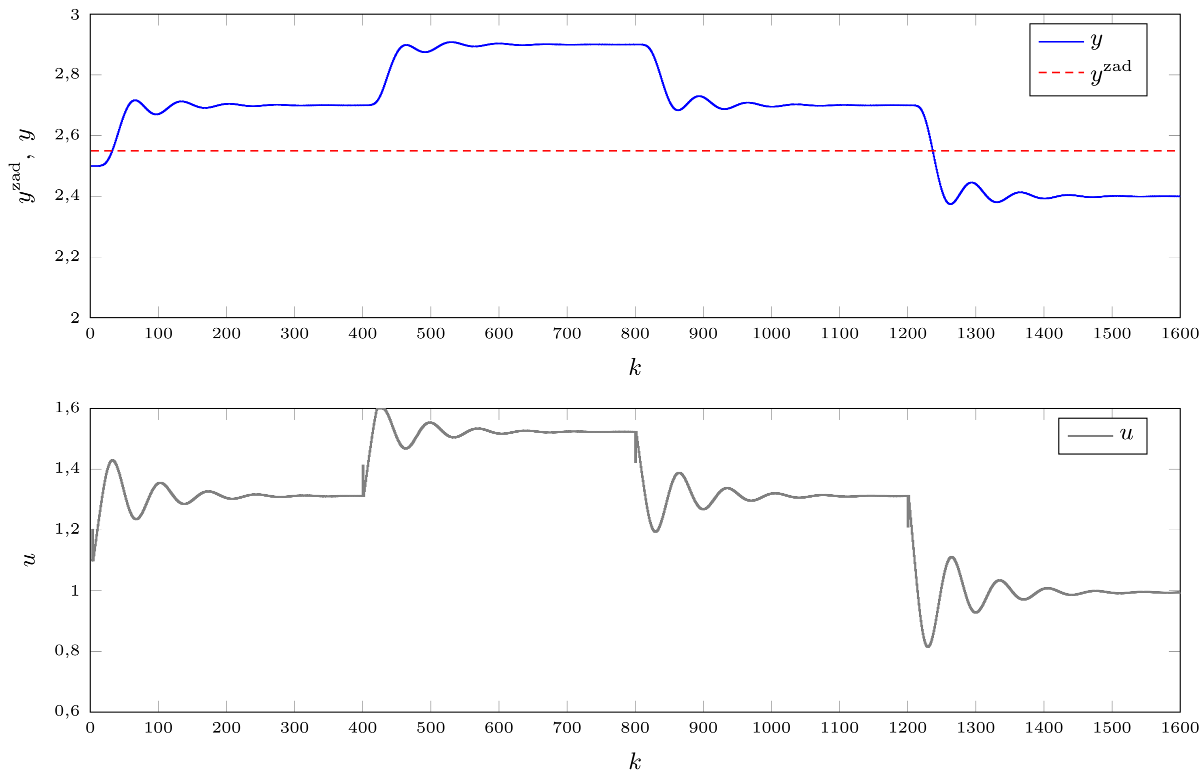


Rys. 7.12. Regulator PID dla $K = 1{,}975$, $T_i = 22$, $T_d = 0$

3. Dla tak dobranych parametrów D i N dobranie parametru N_{u} dającego zadowalające wyniki.

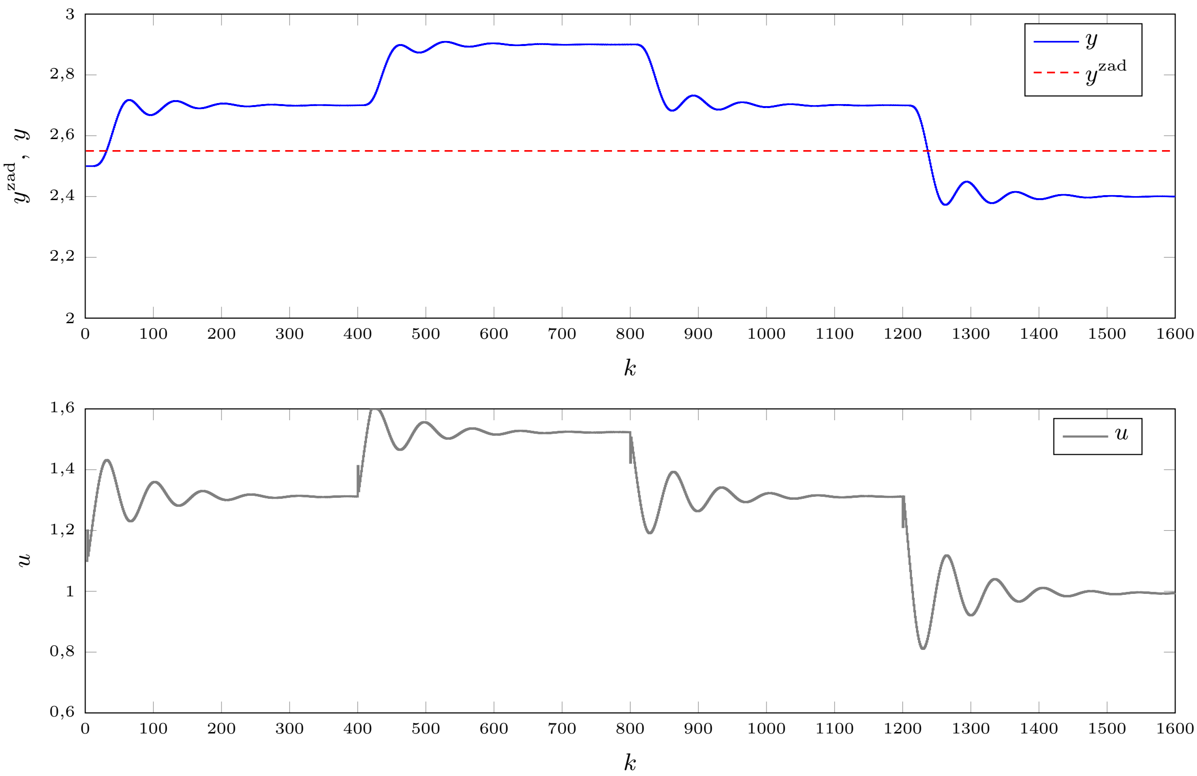


Rys. 7.13. Regulator PID dla $K = 1{,}975$, $T_{\mathrm{i}} = 24$, $T_{\mathrm{d}} = 1$

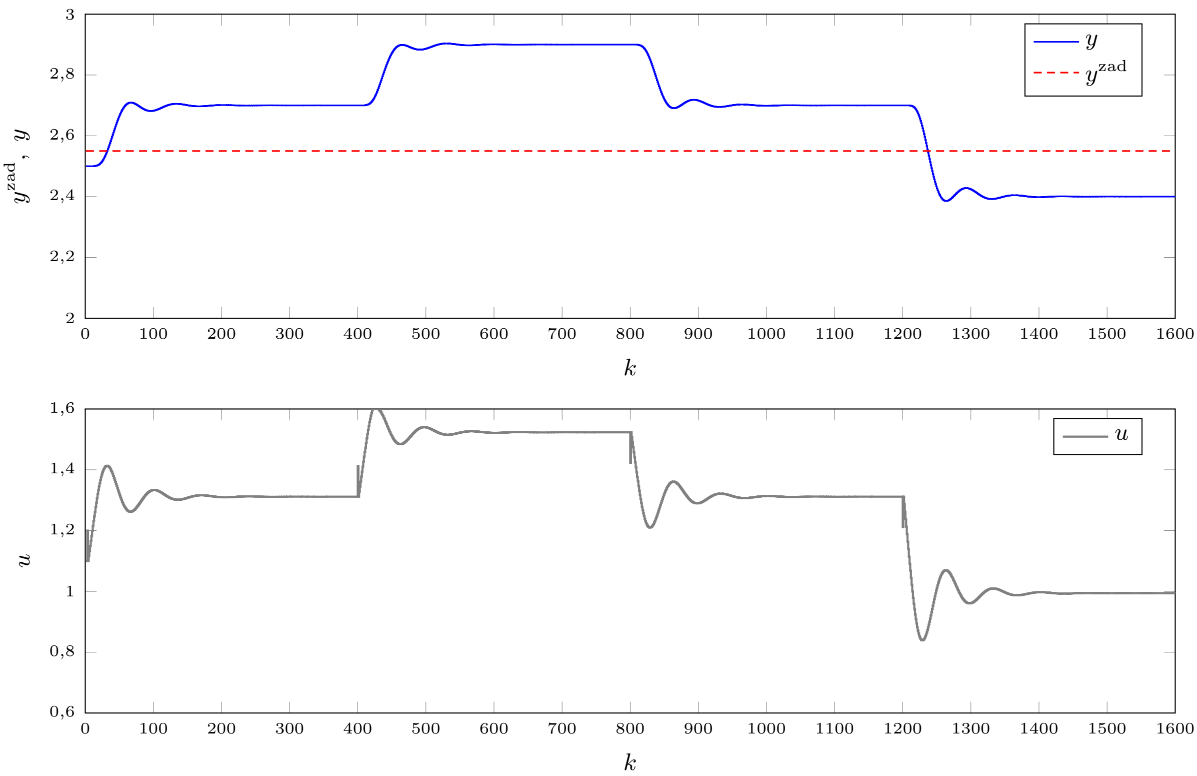


Rys. 7.14. Regulator PID dla $K = 1{,}975$, $T_{\mathrm{i}} = 24$, $T_{\mathrm{d}} = 0{,}5$

Wyjściowy regulator DMC ($D = N = N_\mathrm{u} = 200$) przeedstawiony jest na Rys 6.13. Po błędzie i ryunku widać, że już i taki regulator DMC sprawia się lepiej niż regulator PID. Ponieważ dla

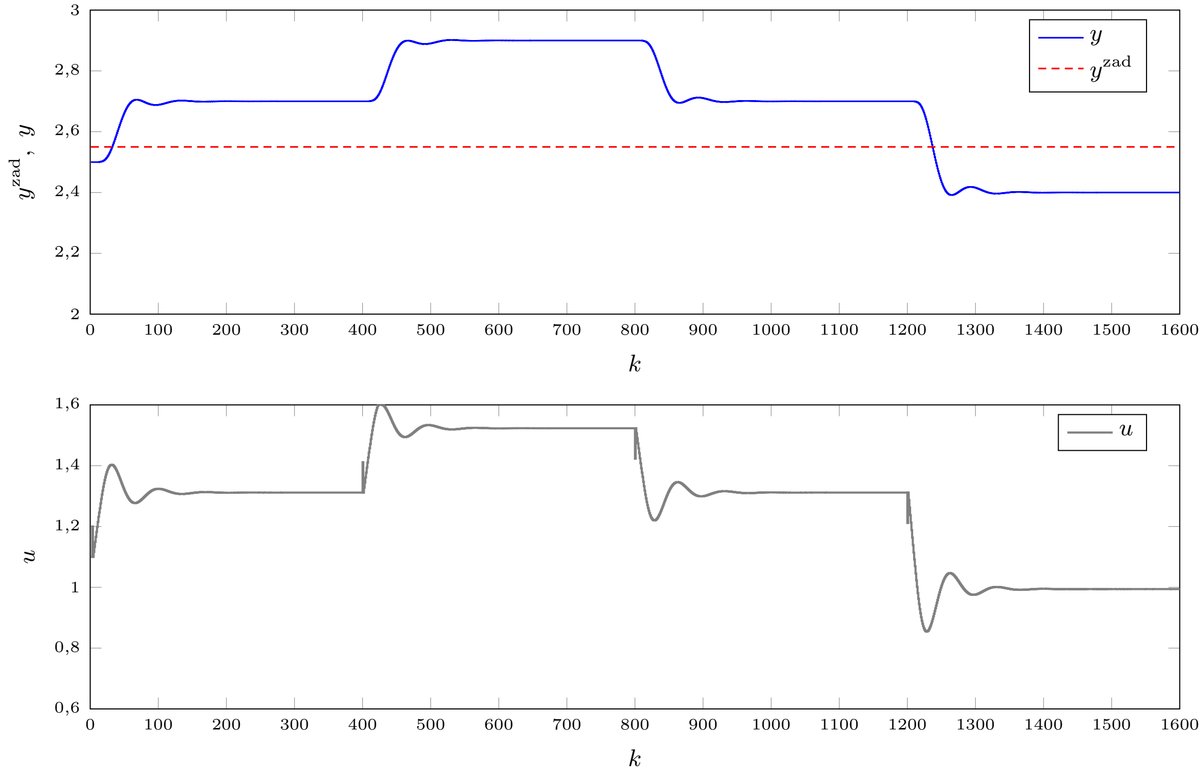


Rys. 7.15. Regulator PID dla $K = 1{,}975$, $T_\mathrm{i} = 24$, $T_\mathrm{d} = 0{,}25$

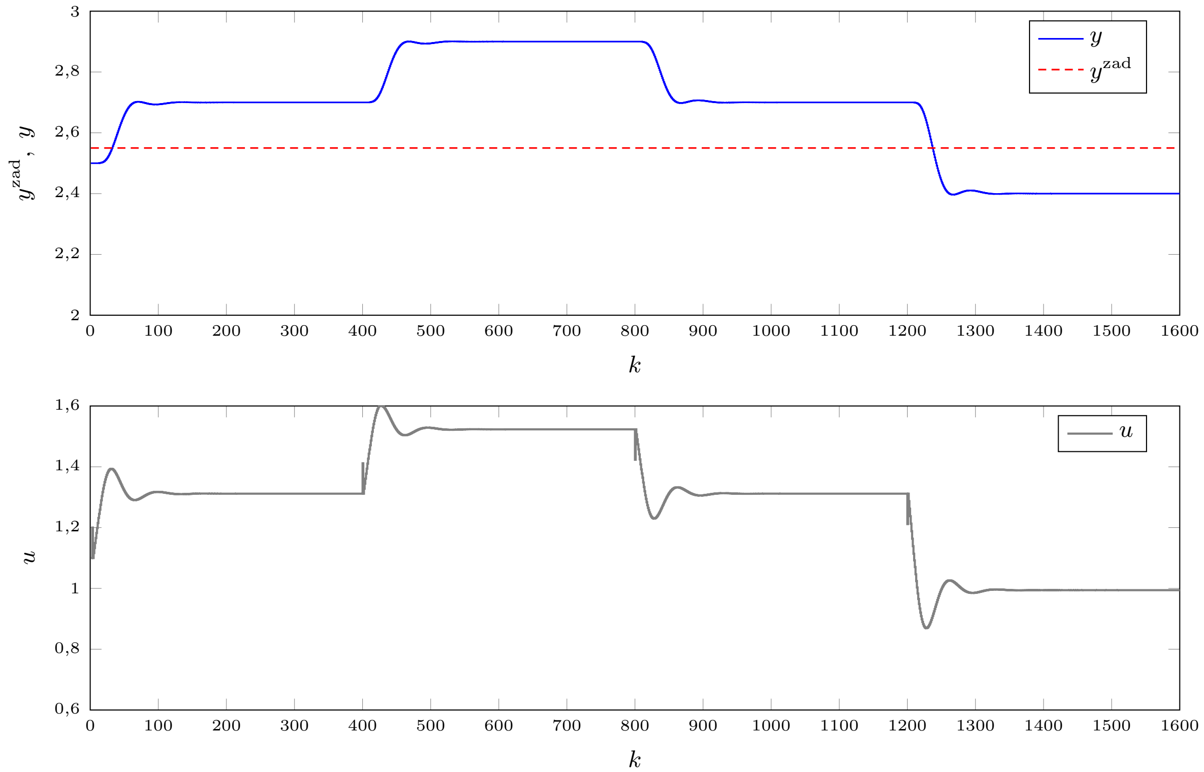


Rys. 7.16. Regulator PID dla $K = 1{,}975$, $T_\mathrm{i} = 24$, $T_\mathrm{d} = 2$

wartości $N \geqslant 50$ regulator zachowywał się w przybliżeniu identycznie nie zamieszczone zostały wykresy dla nich.



Rys. 7.17. Regulator PID dla $K = 1{,}975$, $T_{\mathrm{i}} = 24$, $T_{\mathrm{d}} = 3$

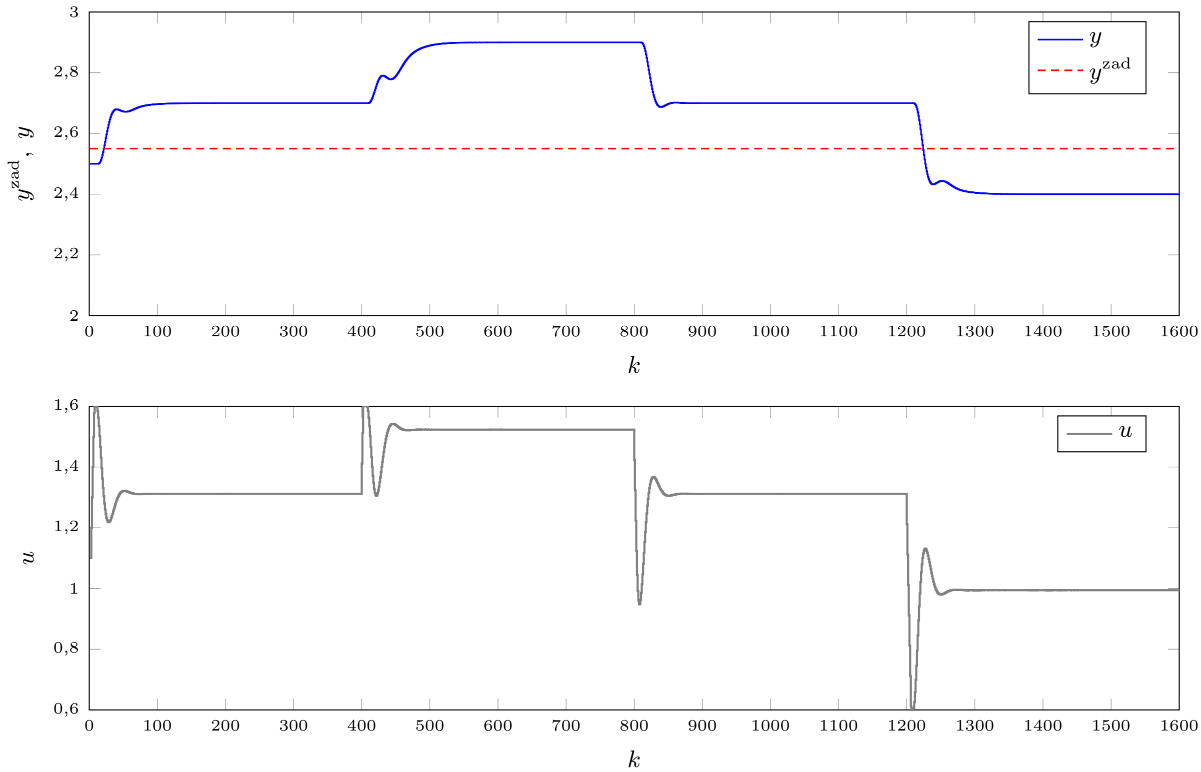


Rys. 7.18. Regulator PID dla $K = 1{,}975$, $T_{\mathrm{i}} = 24$, $T_{\mathrm{d}} = 4$

Większość przebiegów zapewnia zadowalający kształt funkcji wyjściowej, dlatego wybierając najlepszą wartość parametru N kierować będziemy się wartością wskaźnika jakości E, co sprawia, że najlepszą regulację otrzymujemy dla $N = N_\mathrm{u} = 32$. Wartość parametru N_u dobierana więc

Tab. 7.2. Porównanie wielkości błędu E dla różnych wartości parametru T_d i dla parametru $K = 1.975$ oraz parametru $T_\mathrm{i} = 24$

T_d	E
0	5,6081
0,5	6,6090
0,25	6,4963
1	6,5853
2	6,5632
3	6,5644
4	6,5808

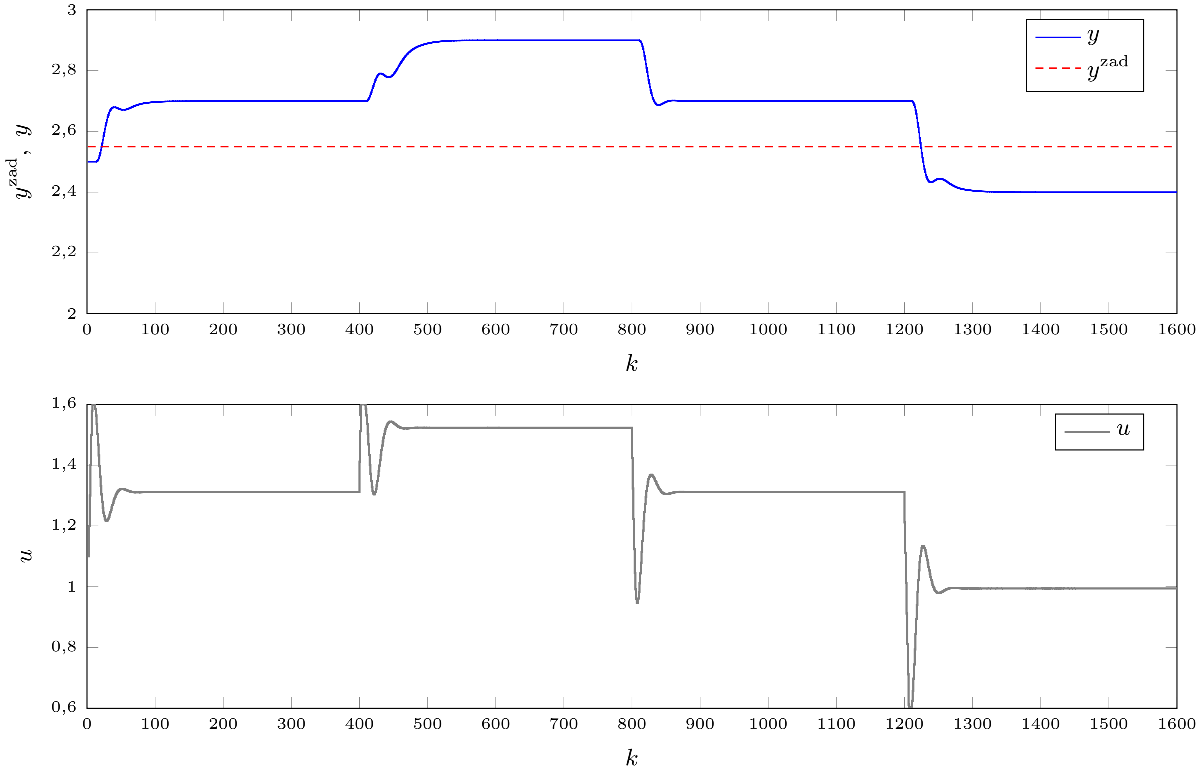


Rys. 7.19. Regulator DMC dla $D = 200$, $N = 200$, $N_\mathrm{u} = 200$

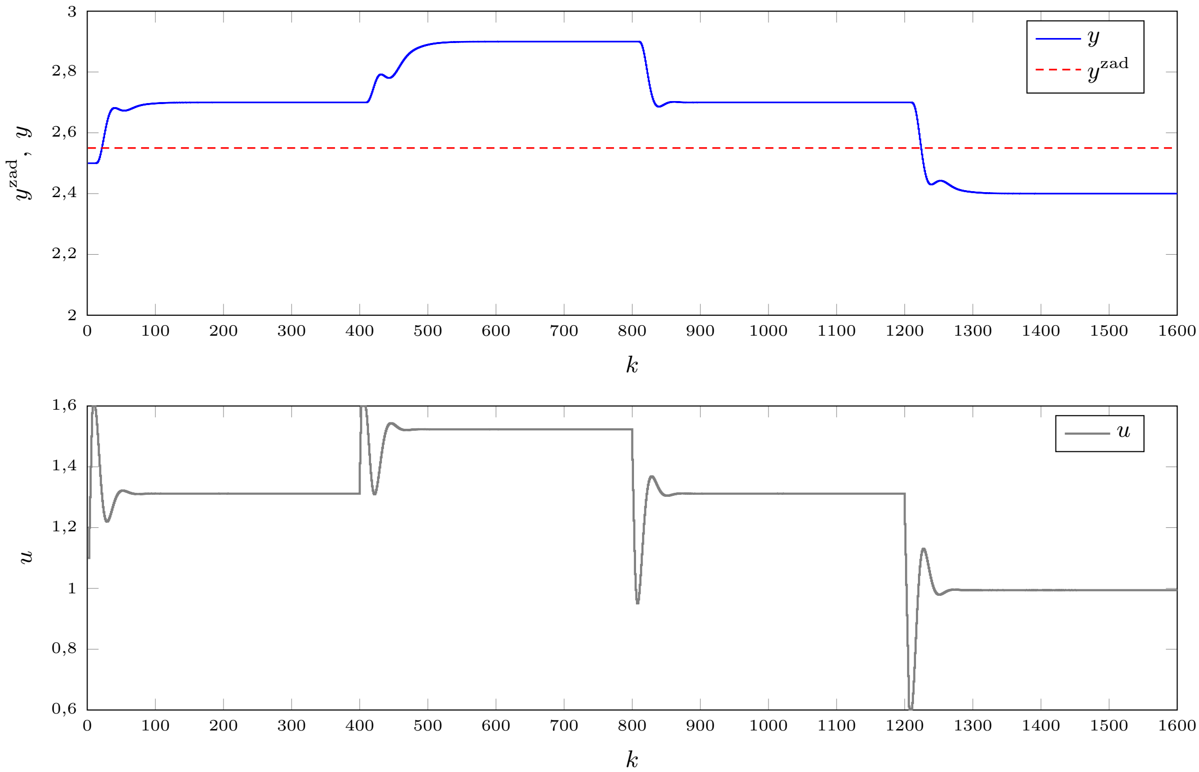
Tab. 7.3. Porównanie wielkości błędu E dla różnych wartości parametrów N i N_u i dla parametru $D = 200$

N i N_u	E
200	4,8557
150	4,8557
100	4,8557
50	4,8556
40	4,8571
35	4,8403
30	4,8356
32	4,8301
27	4,8876
25	4,9796
20	5,3776

będzie dla $D = 200$ i $N = 32$. Ponieważ dla wartości $N_\mathrm{u} \geqslant 18$ regulator zachowywał się w przybliżeniu identycznie nie zamieszczone zostały wykresy dla nich.

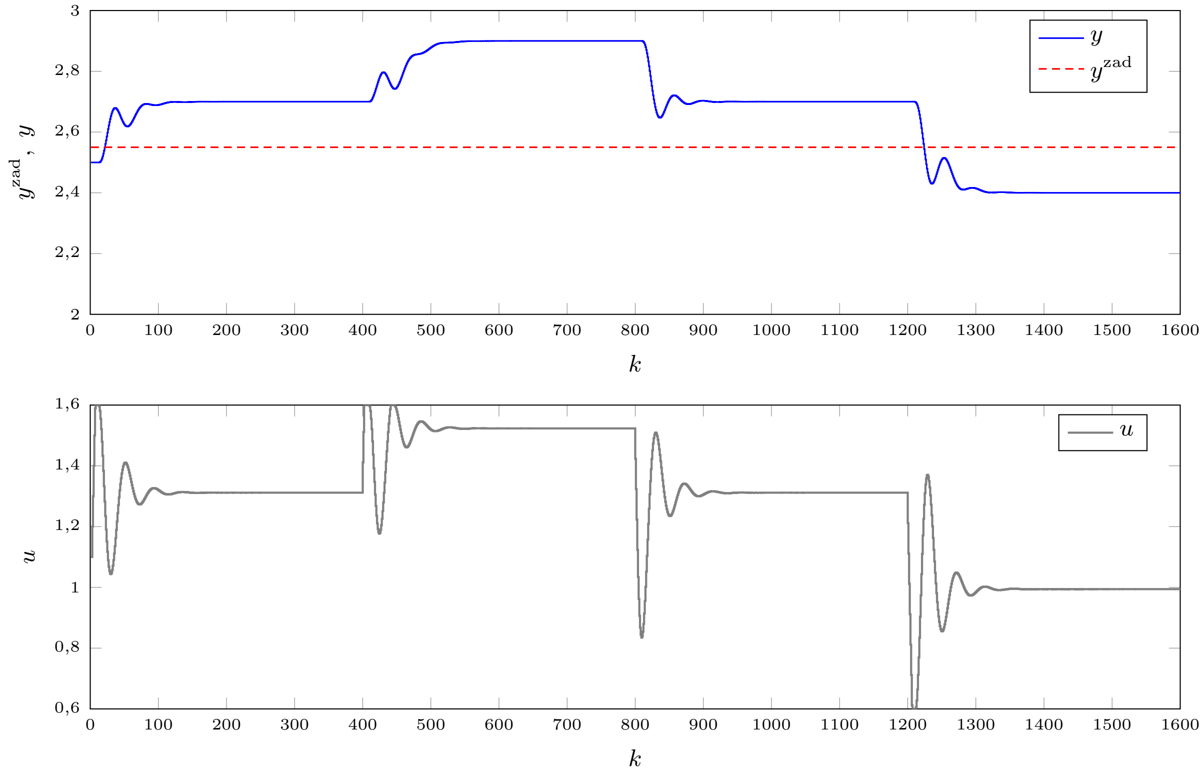


Rys. 7.20. Regulator DMC dla $D = 200$, $N = 40$, $N_\mathrm{u} = 40$

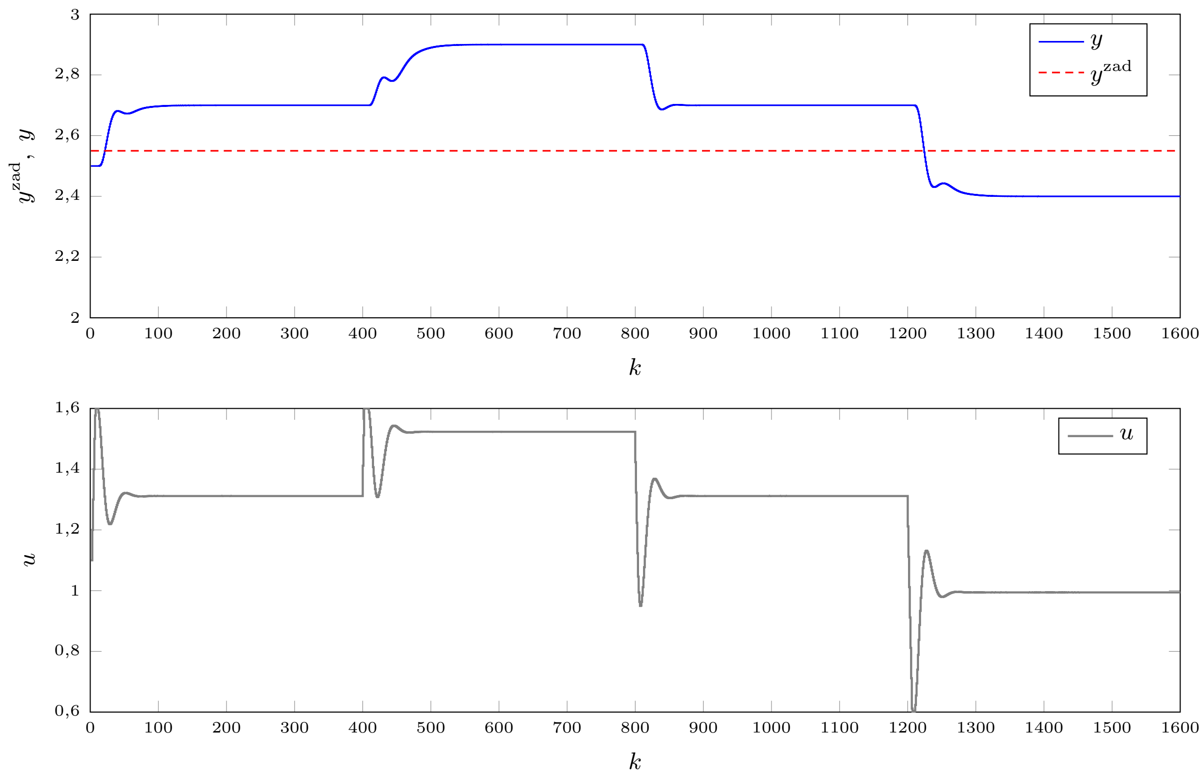


Rys. 7.21. Regulator DMC dla $D = 200$, $N = 30$, $N_\mathrm{u} = 30$

Dla różnych wartości parametru N_u przebiegi są w większości podobne, jedynie wartość wskaź-

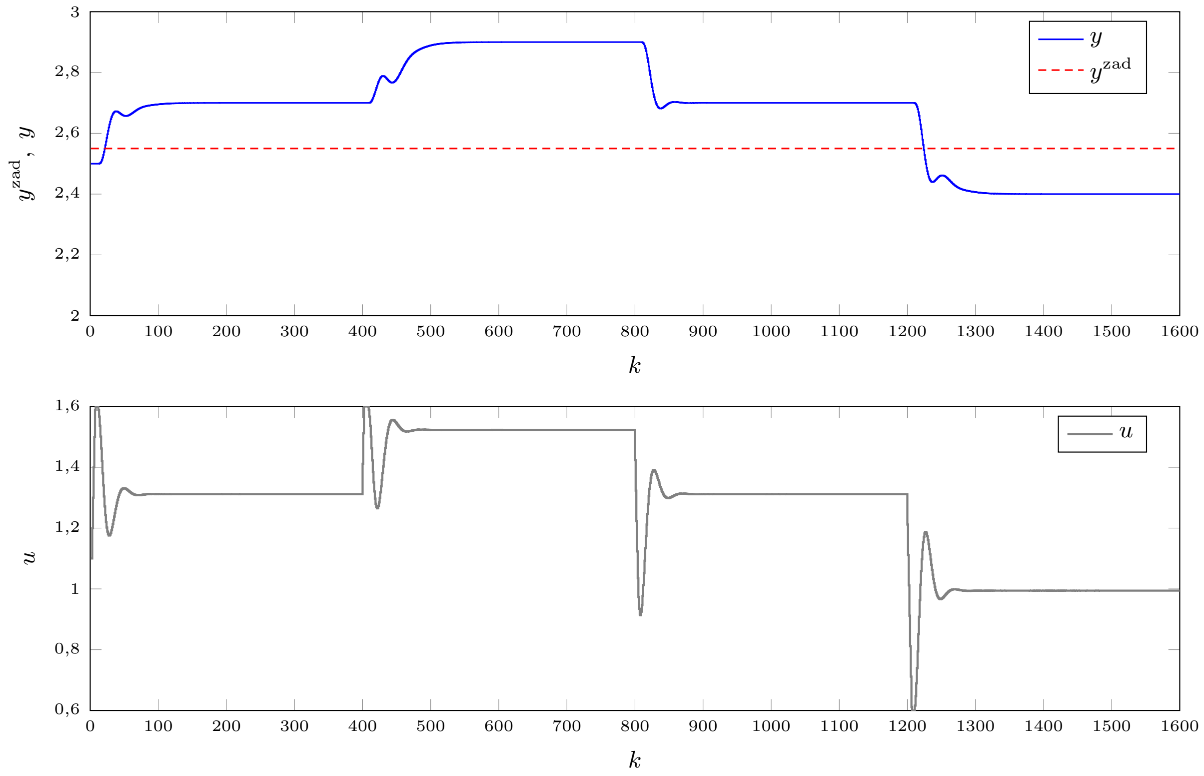


Rys. 7.22. Regulator DMC dla $D = 200$, $N = 20$, $N_\mathrm{u} = 20$

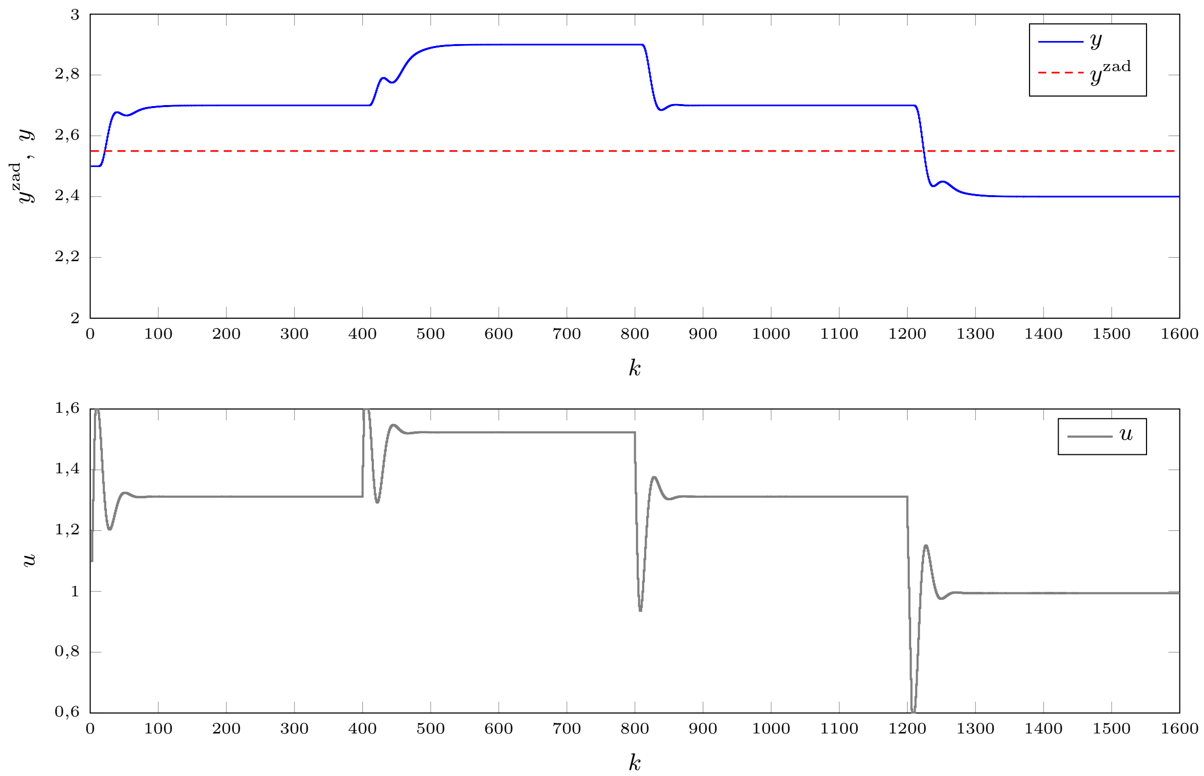


Rys. 7.23. Regulator DMC dla $D = 200$, $N = 35$, $N_\mathrm{u} = 35$

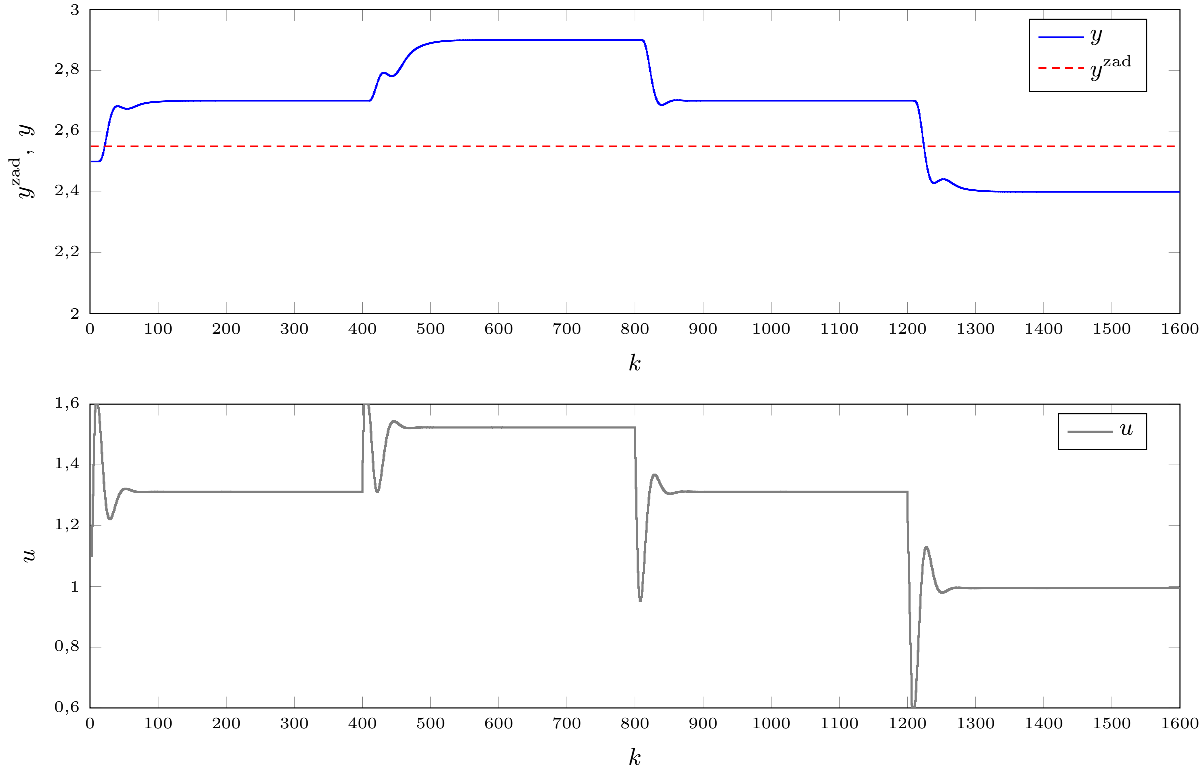
nika jakości E różni się nieznacznie między nimi. Z tego powodu, wybrany zostanie regulator o parametrach $D = 200$, $N = 32$ i $N_\mathrm{u} = 4$ (Rys 6.31).



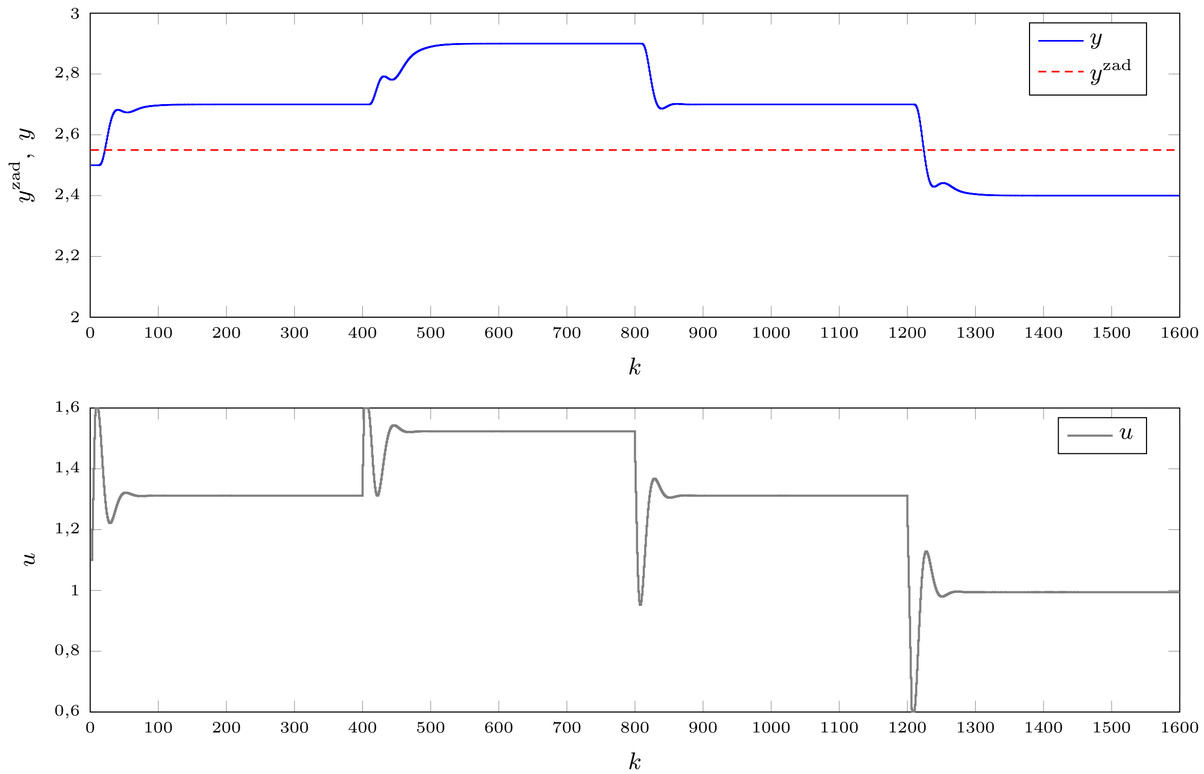
Rys. 7.24. Regulator DMC dla $D = 200$, $N = 25$, $N_\mathrm{u} = 25$



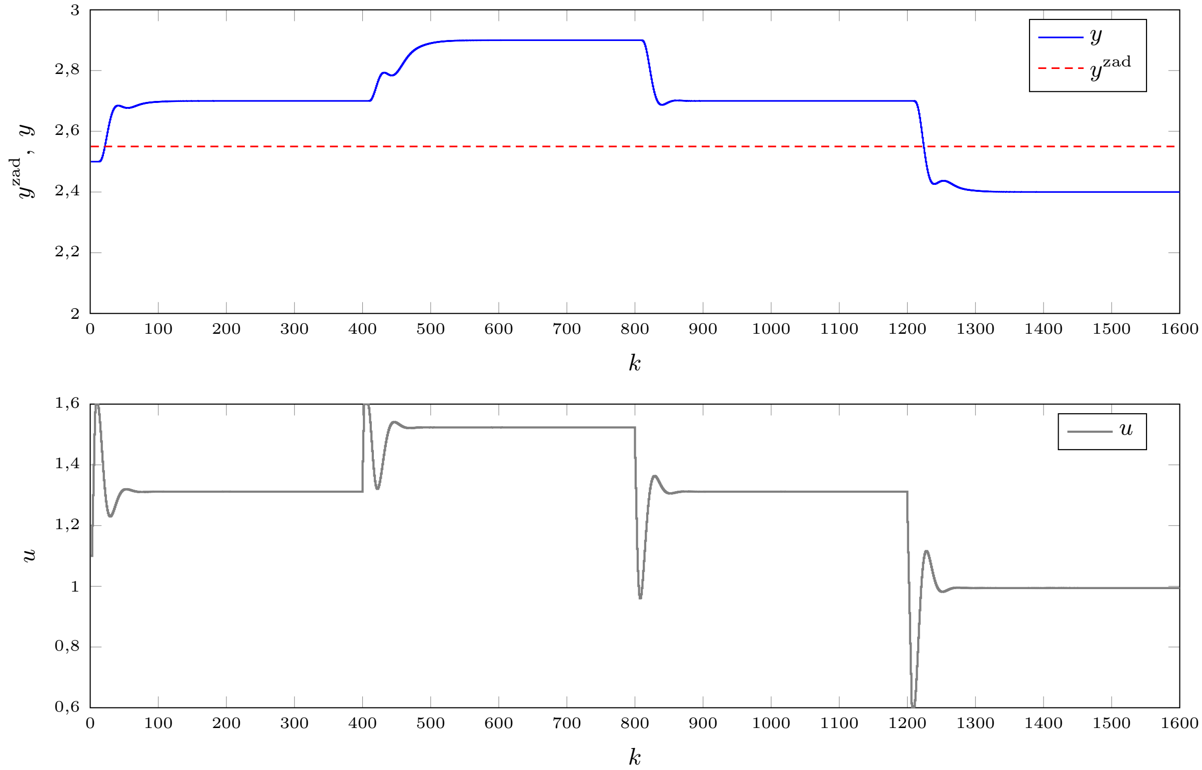
Rys. 7.25. Regulator DMC dla $D = 200$, $N = 27$, $N_\mathrm{u} = 27$



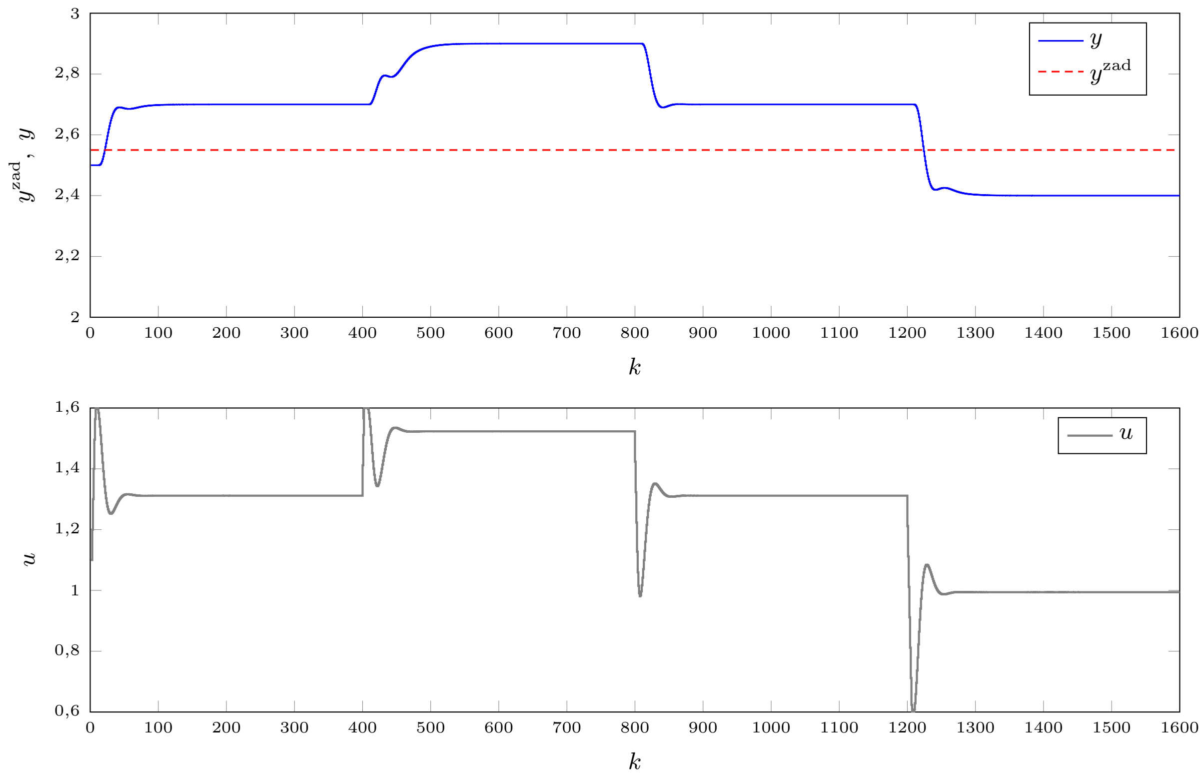
Rys. 7.26. Regulator DMC dla $D = 200$, $N = 32$, $N_u = 32$



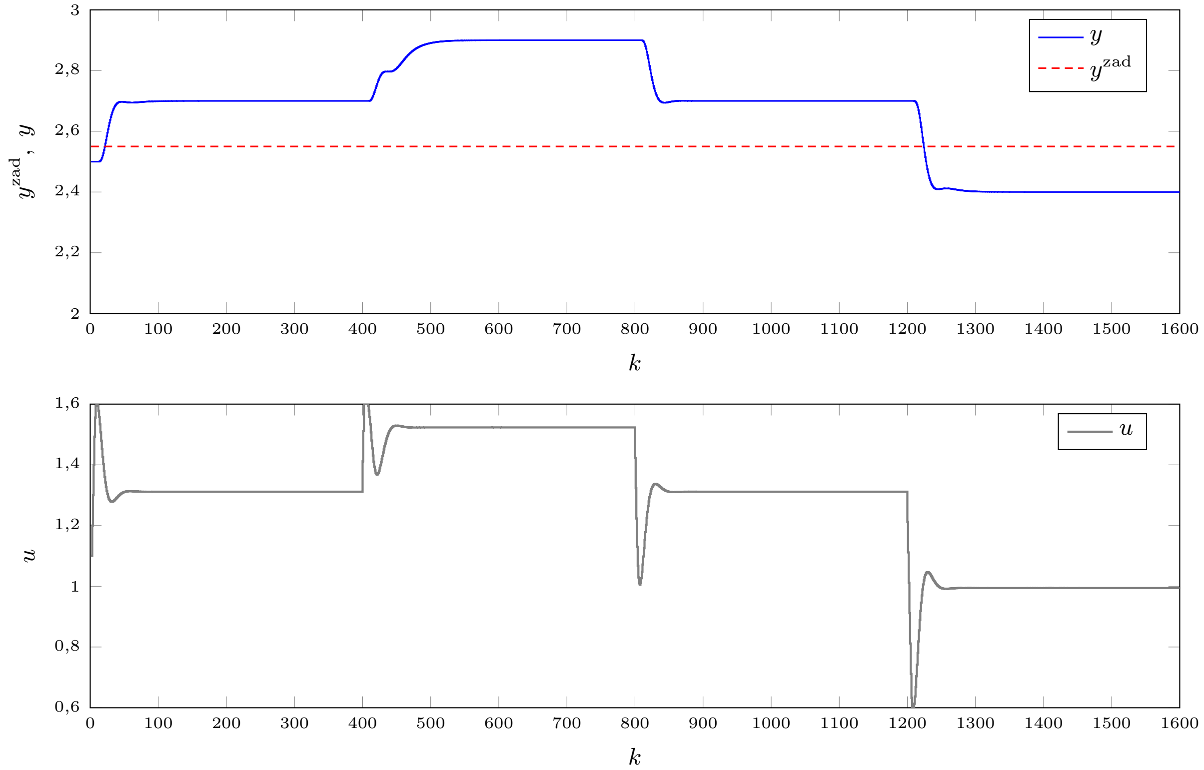
Rys. 7.27. Regulator DMC dla $D = 200$, $N = 32$, $N_u = 15$



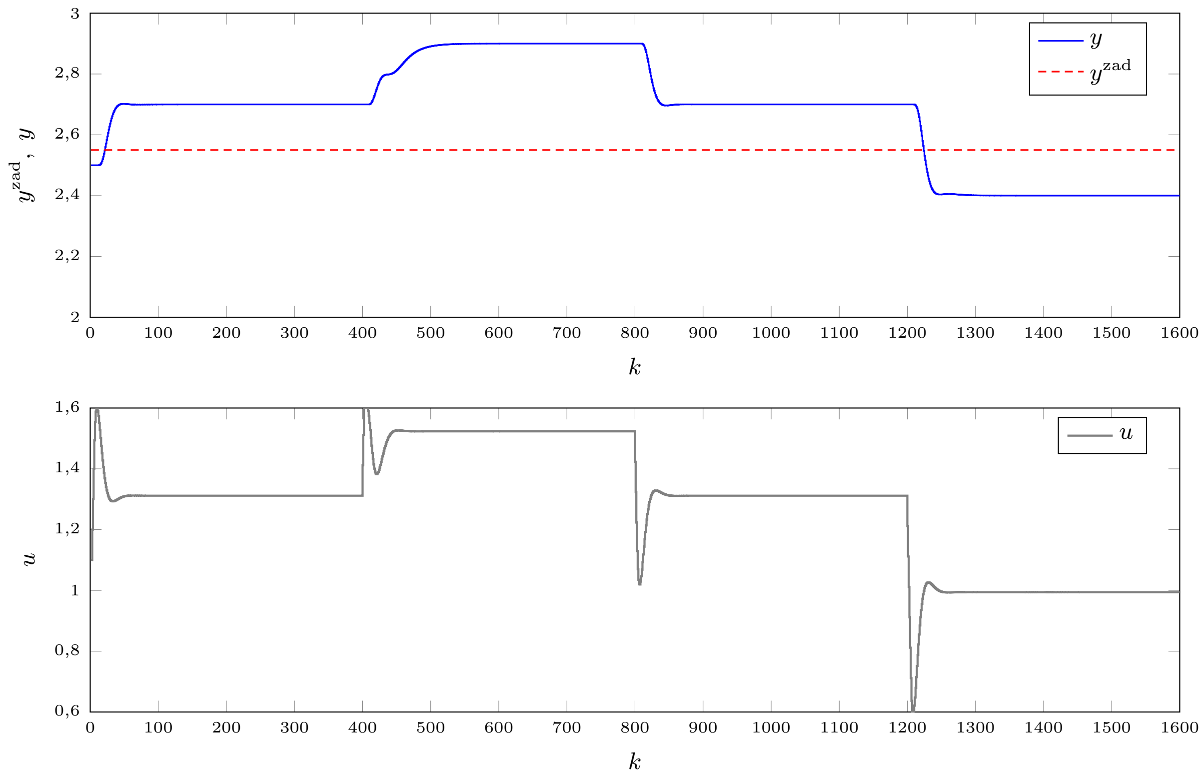
Rys. 7.28. Regulator DMC dla $D = 200$, $N = 32$, $N_{\mathrm{u}} = 10$



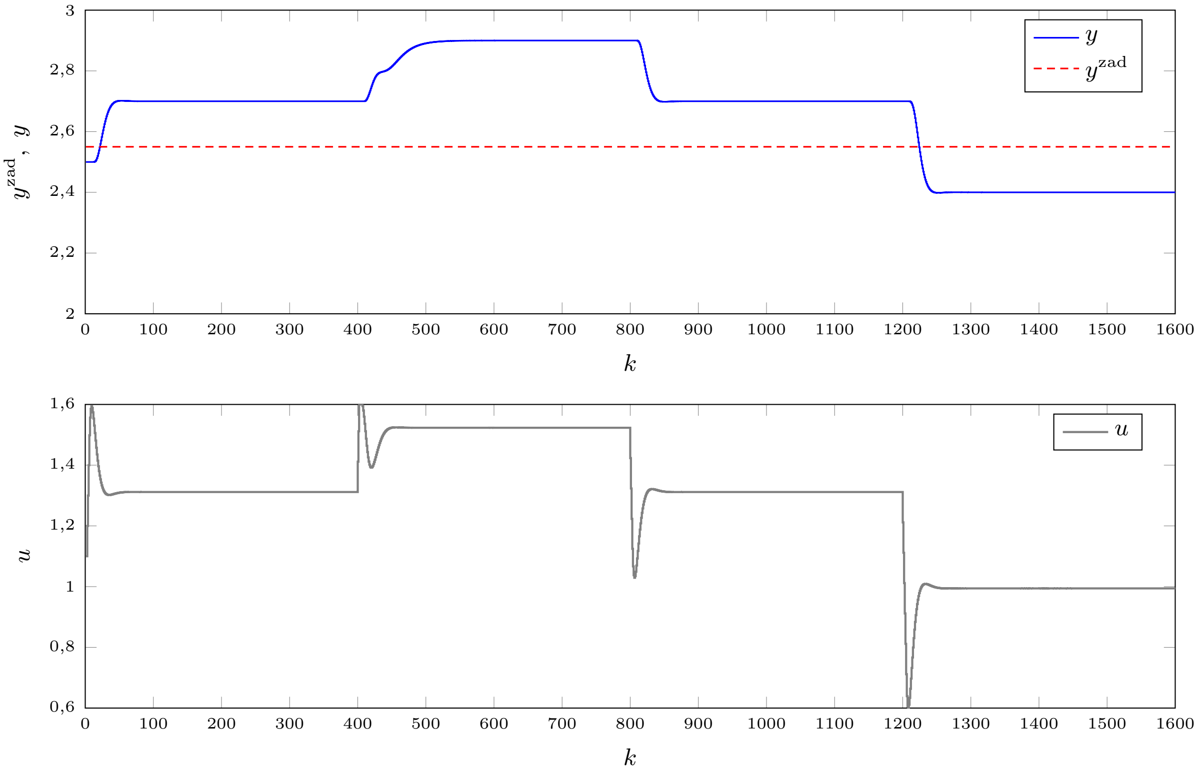
Rys. 7.29. Regulator DMC dla $D = 200$, $N = 32$, $N_{\mathrm{u}} = 7$



Rys. 7.30. Regulator DMC dla $D = 200$, $N = 32$, $N_\mathrm{u} = 5$



Rys. 7.31. Regulator DMC dla $D = 200$, $N = 32$, $N_\mathrm{u} = 4$



Rys. 7.32. Regulator DMC dla $D = 200$, $N = 32$, $N_{\mathrm{u}} = 3$

Tab. 7.4. Porównanie wielkości błędu E dla różnych wartości parametru N_{u} i dla parametrów $D = 200$ i $N = 32$

N_{u}	E
32	4,8301
25	4,8301
20	4,8301
18	4,8301
15	4,8288
10	4,8027
7	4,7471
5	4,7036
4	4,6905
3	4,7039

8. zad6

8.1. Optymalizacja regulatora PID

W celu optymalizacji użyta zostanie funkcja *fmincon* z Matlaba. Znajduje ona lokalne minimum funkcji na podstawie jej gradientu. Aby użyc jej do optymalizacji regulatora PID zdefiniowano funkcję ewaluującą jakość regulacji, która zwraca wskaźnik jakości E, a jako argument bierze wektor nastaw K, T_i i T_d regulatora. Funkcja *fmincon* będzie starała się ten wskaźnik zminimalizować, zaczynając od podanego przez nas wektora nastaw. Funkcja ewaluująca zwraca wskaźnik jakości regulacji E. Wywołanie funkcji *fmincon* wyglądać może na przykład tak:

```
%wywolanie funkcji ewaluujacej regulator PID
x01 = [1.975, 24, 3];   %punkt poczatkowy optymalizacji
fmincon(@pid_eval, x01)   %wywolanie funkcji
```

Po znalezieniu przez funkcję *fmincon* minimum lokalnego, możemy wprowadzić te parametry do skryptu z poprzedniego zadania, i sprawwdzić czy rzeczywiście działają lepiej. Na rysunku 7.1 przedstawiono odnalezione przez funkcję *fmincon* rozwiazanie. Widać jednak, że otrzymany regulator pracuje w stanie ciągłych oscylacji, i mimo że są one bardzo małe (na tyle małe, że nie sprawiają dużej różnicy dla wskaźnika jakości E), są one niepożądane na obiektach realnych. Regulator ten średnio więc się nadaje do zastosowania na obiekcie realnym. Sposobem na poprawę, mogłoby być wprowadzenie jakiegoś dodatkowego współczynnika kary za oscylacje wokół wartości zadanej, bądź duże wydłużenie czasu w której wartość zadana jest stała, żeby spróbować skłonić *fmincon* do faworyzowania nieoscylujących regulatorów. Rys 7.2 przedstawia regulator otrzymany w wyniku optymalizacji funkcją *fmincon* dla ośmiokrotnie wydłużonej symulacji. Zgodnie z oczekiwaniami, otrzymany regulator PID ma dużo mniejsze tendencje do oscylowania. Dla bardziej wydłużonej symulacji, powinno móc się uzyskać regulator dużo mniej podatny na oscylacje. Przez to jednak, wskaźnik jakości jest nieco gorszy.

8.2. Optymalizacja regulatora DMC

W przypadku regulatora DMC niemożliwa jest optymalizacja funkcją *fmincon*, ponieważ parametry regulatora DMC są zawsze dodatnimi liczbami całkowitymi. Z powodu użycia *floor* w funkcji ewaluującej regulator DMC (analogiczna do funkcji ewaluującej regulator PID), funkcja *fmincon* od razu zakończy optymalizację, twierdząc, że znalazła lokalne minimum. Jeżeli ma być odnaleziony najlepszy regulator DMC, należy użyć innej metody optymalizacji. Wybrana została funkcja *ga* („Genetic Algorithm"), która jest częścią *Global Optimization Toolbox* Matlaba. Wywołanie funkcji *ga* może wyglądać tak:

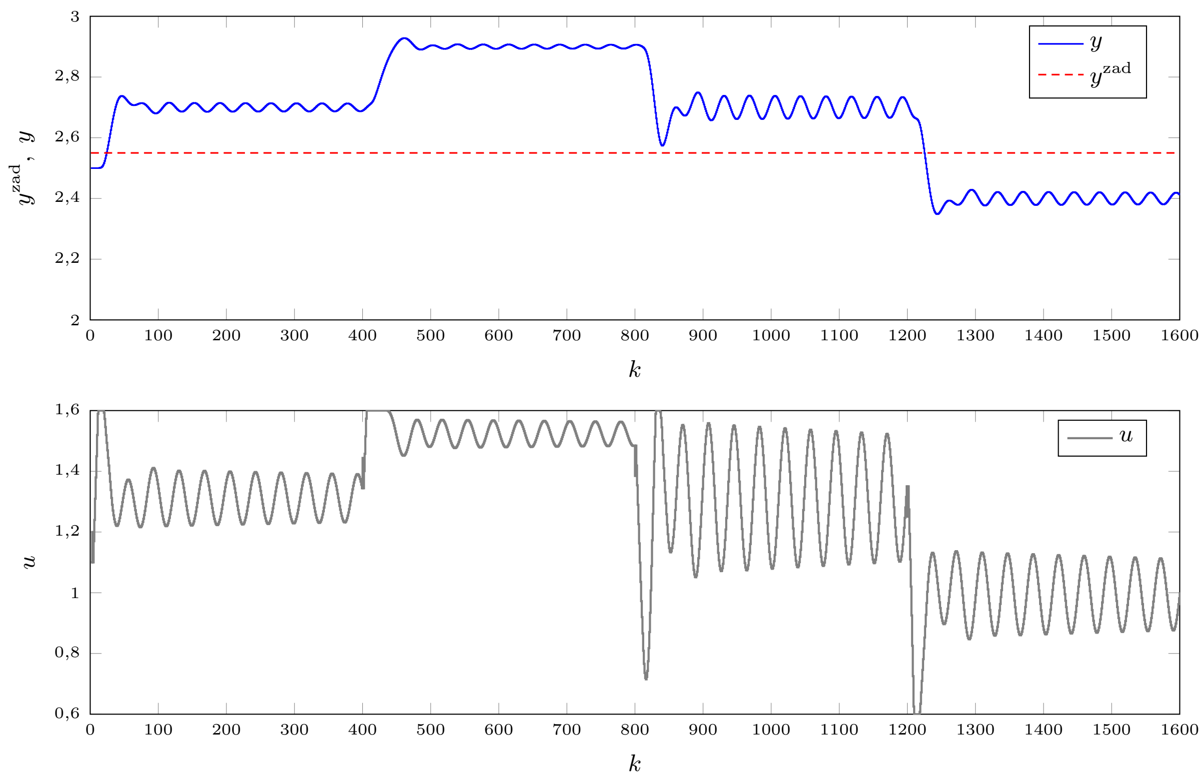
[b]

Tab. 8.1. Porównanie wielkości błędu E dla różnych wartości parametru T_d i dla parametru $K = 1.975$ oraz parametru $T_\mathrm{i} = 24$

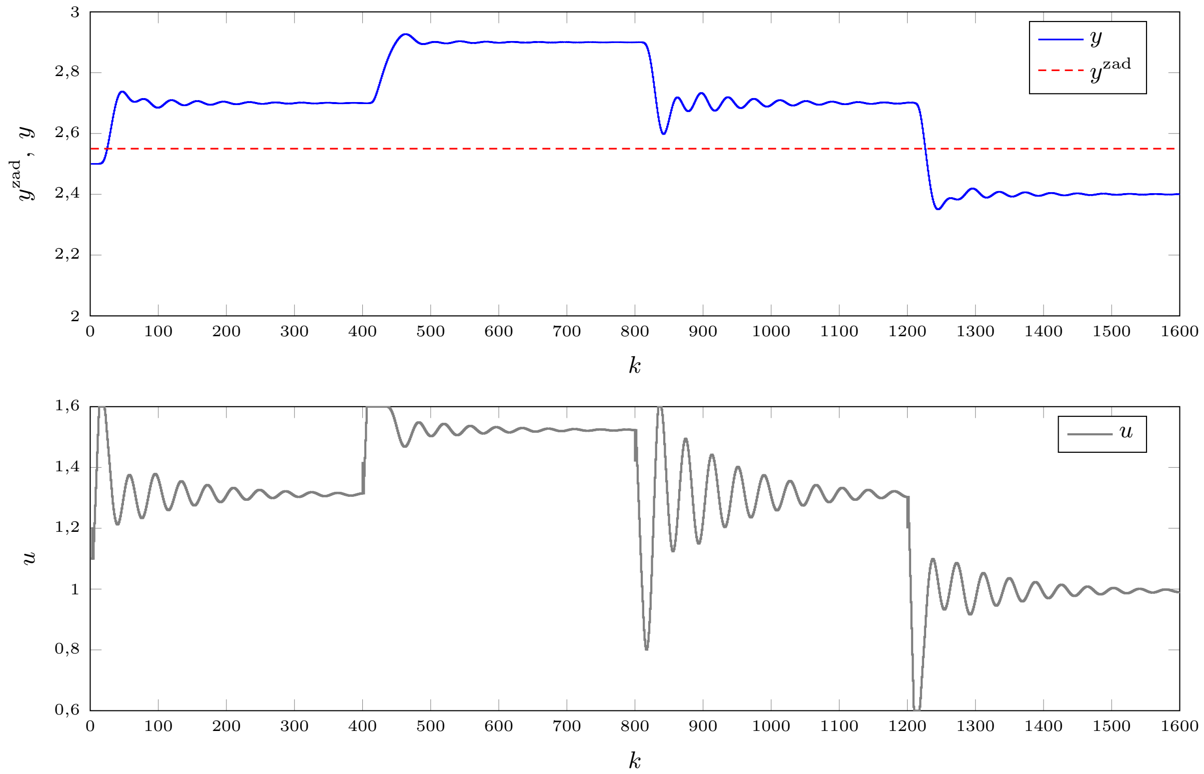
K	T_i	T_d	E
4,3405	12,4148	8,097	5,0221
3,9628	13,6246	7,8695	5,1382

```
%wywolanie funkcji ewaluujacej regulator PID
x01 = [1.975, 24, 3];    %punkt poczatkowy optymalizacji
fmincon(@pid_eval, x01)    %wywolanie funkcji
%wywolanie funkcji ewaluujacej regulator DMC
A = [-1, 1];
b = 0;            %ograniczenie liniowe -N+Nu<=0, czyli Nu<=N
Aeq = [];
beq = [];
lb = [1, 1];
ub = [200, 200];%ograniczenia wartosci N i Nu 1<=N<=200 i 1<=Nu<=200
nonlcon = [];
IntCon = [1, 2];     %pierwszy i drugi argument musza byc calkowite
ga(@dmc_eval,2,A,b,Aeq,beq,lb,ub,nonlcon,IntCon)
```

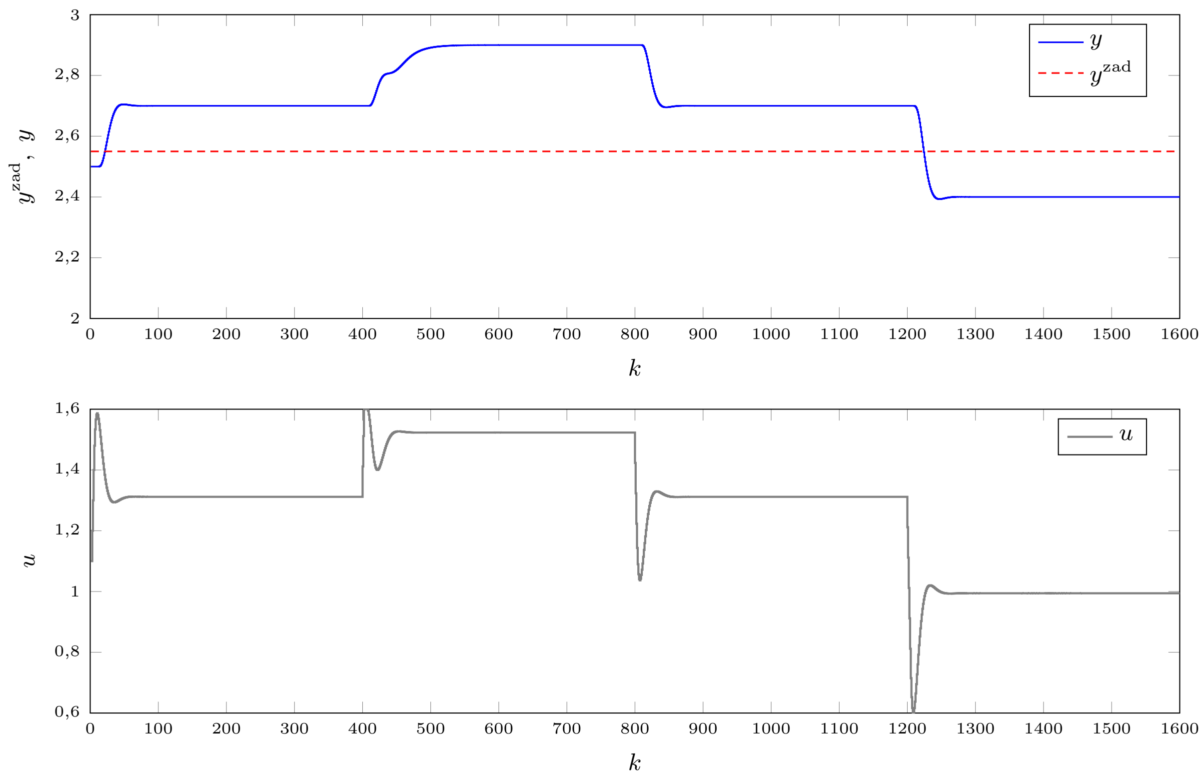
Tym spsobem otrzymaliśmy optymalne parametry regulatora DMC dla wybranej przez nas trajektorii wartości zadanej: $D = 200$, $N = 44$, $N_\mathrm{u} = 5$. Regulator przedstawiony jest na Rys 7.3, a jego wskaźnik jakości $E = 4{,}6546$.



Rys. 8.1. Regulator PID $K = 4{,}34$, $T_\mathrm{i} = 12{,}41$, $T_\mathrm{d} = 8{,}1$ otrzymany z punktu początkowego optymalizacji $K = 1{,}975$, $T_\mathrm{i} = 24$, $T_\mathrm{d} = 3$



Rys. 8.2. Regulator PID $K = 3{,}963$, $T_\mathrm{i} = 13.62$, $T_\mathrm{d} = 7.87$ otrzymany z punktu początkowego optymalizacji $K = 1{,}975$, $T_\mathrm{i} = 24$, $T_\mathrm{d} = 3$



Rys. 8.3. Regulator DMC dla $D = 200$, $N = 44$, $N_\mathrm{u} = 5$