

Wydział Elektroniki i Technik Informacyjnych  
Politechnika Warszawska

Projektowanie układów sterowania  
(projekt grupowy)

Sprawozdanie z projektu i ćwiczenia laboratoryjnych  
1, zadanie nr 1

Kacper Marchlewicz, Michał Kwarciański, Adam Wróblewski  
grupa nr 7

Warszawa, 2022

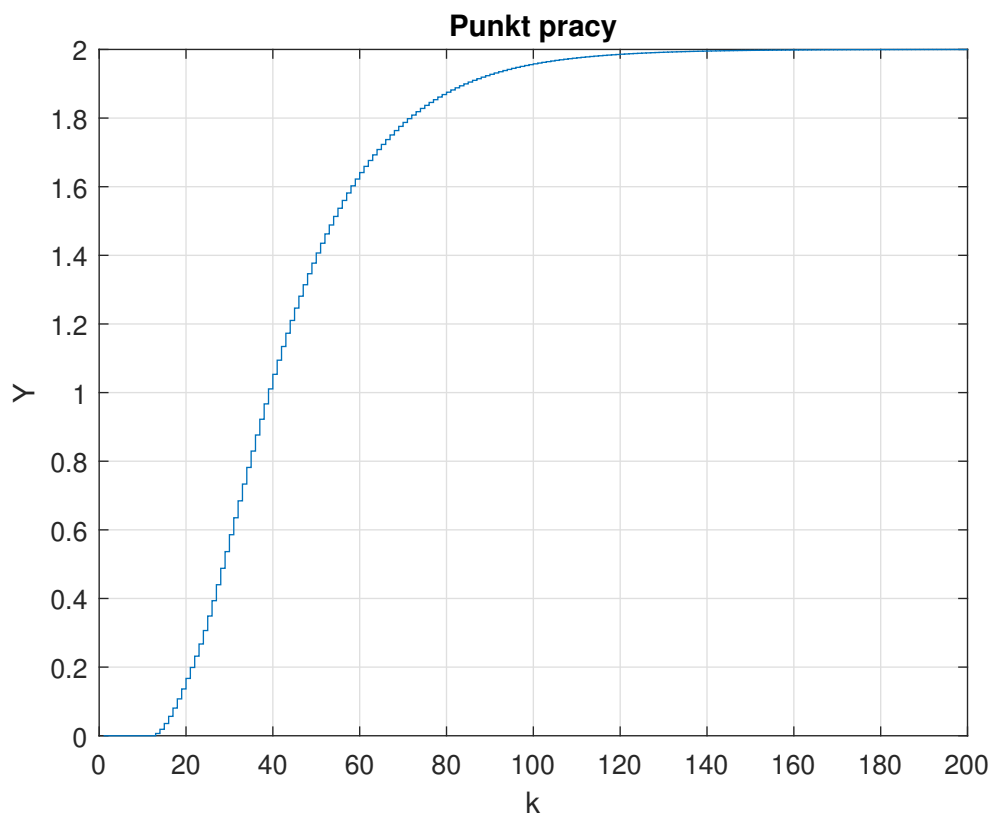
# Spis treści

<b>1. Część projektowa</b>	2
1.1. Sprawdzenie poprawności wartości $U_{pp}$ i $Y_{pp}$	2
1.2. Symulacyjne wyznaczenie odpowiedzi skokowych procesu	2
1.3. Przekształcenie odpowiedzi skokowej dla algorytmu DMC	4
1.4. Symulacja cyfrowego algorytmu PID oraz algorytmu DMC	4
1.5. Dobranie nastaw regulatora PID i parametrów algorytmu DMC metodą eksperymentalną	6
1.6. Dobranie nastaw regulatora PID i algorytmów DMC w wyniku optymalizacji wskaźnika jakości	9
<b>2. Część laboratoryjna</b>	11
2.1. Sprawdzenie możliwości sterowania i pomiaru w komunikacji ze stanowiskiem	11
2.2. Wyznaczenie odpowiedzi skokowych	11
2.3. Otrzymanie odpowiedzi skokowej do algorytmu DMC	13
2.4. Cyfrowy algorytm PID i DMC	14
2.5. Dobranie nastaw regulatora PID i algorytmu DMC metodą eksperymentalną	14

# 1. Część projektowa

## 1.1. Sprawdzenie poprawności wartości $U_{pp}$ i $Y_{pp}$

W celu sprawdzenia poprawności punktu pracy, sygnał sterujący ustawiamy na wartość 0.8 i obserwujemy wyjście obiektu.



Rys. 1.1. Sprawdzenie punktu pracy

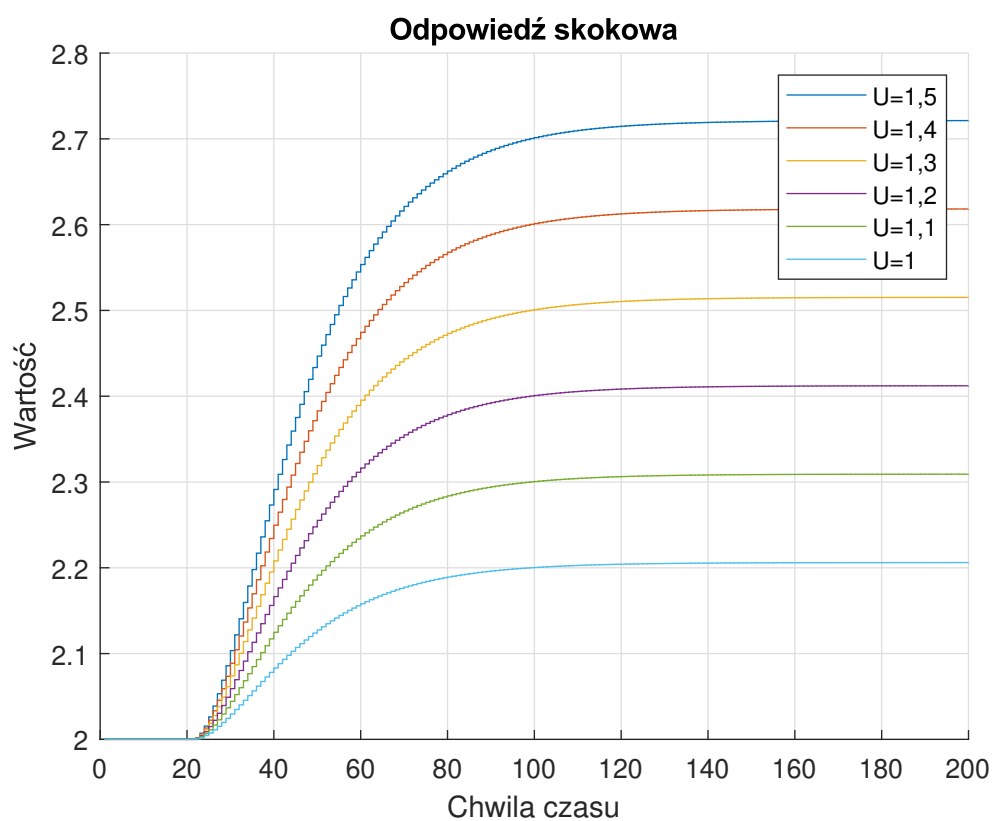
Wyjście symulowanego obiektu ustaliło się na wartości  $Y_{pp}=2$  co potwierdza poprawność punktu pracy. Rysunek 1.1

## 1.2. Symulacyjne wyznaczenie odpowiedzi skokowych procesu

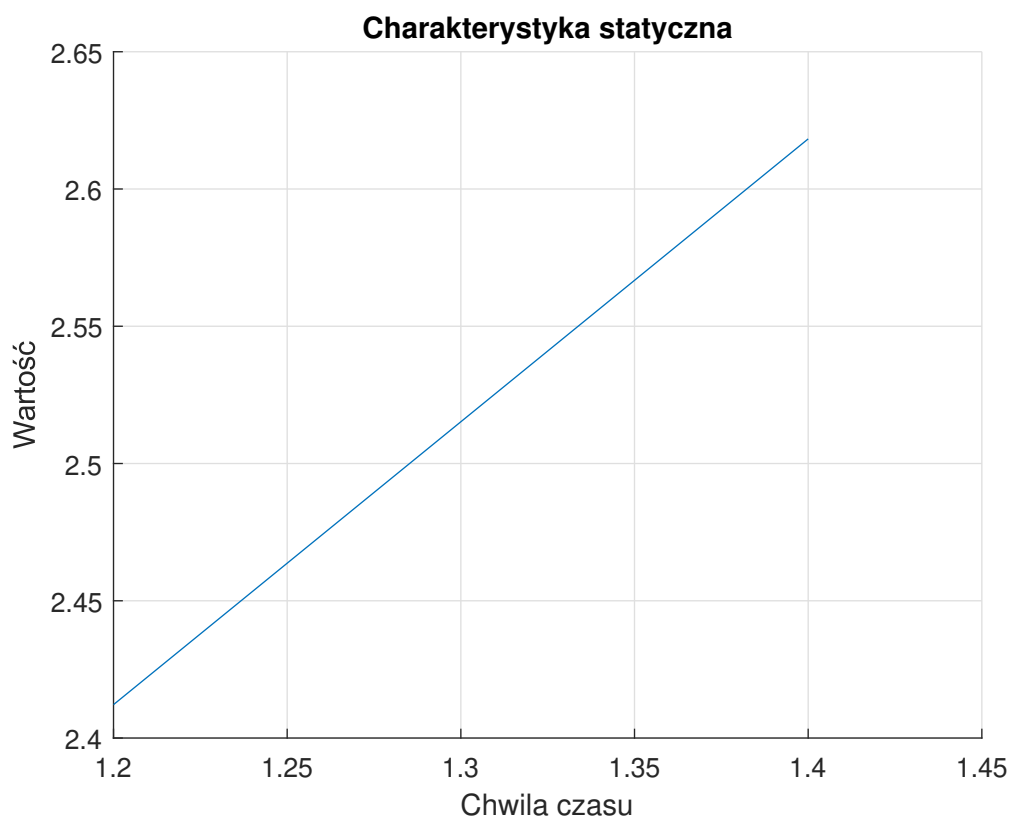
Wykonaliśmy szereg pomiarów odpowiedzi skokowych dla zmian sygnału sterującego w chwili  $k=12$  i wartości sterowania  $U = 1,1; 1,2; 1,3; 1,4; 1,5$ .

Następnie symulując odpowiedź układu dla powyższych wartości sygnału sterującego otrzymaliśmy następującą charakterystykę statyczną.

Na podstawie otrzymanego wykresu (rysunek 1.3) można stwierdzić, że właściwości procesu są w liniowe. Wzmocnienie statyczne  $K_{stat}$  wynosi 1,0304.



Rys. 1.2. Odpowiedzi skokowe



Rys. 1.3. Charakterystyka statyczna

### 1.3. Przekształcenie odpowiedzi skokowej dla algorytmu DMC.

Algorytm DMC używa jednostkowej odpowiedzi skokowej - czyli takiej w której skok sterowania następuje od wartości 0 do 1. W celu przekształcenia odpowiedzi skokowej do postaci wykorzystywanej w algorytmie DMC Stosujemy wzór:

$$s_i = \frac{y_i - Y_{pp}}{\delta u} \quad (1.1)$$

gdzie:

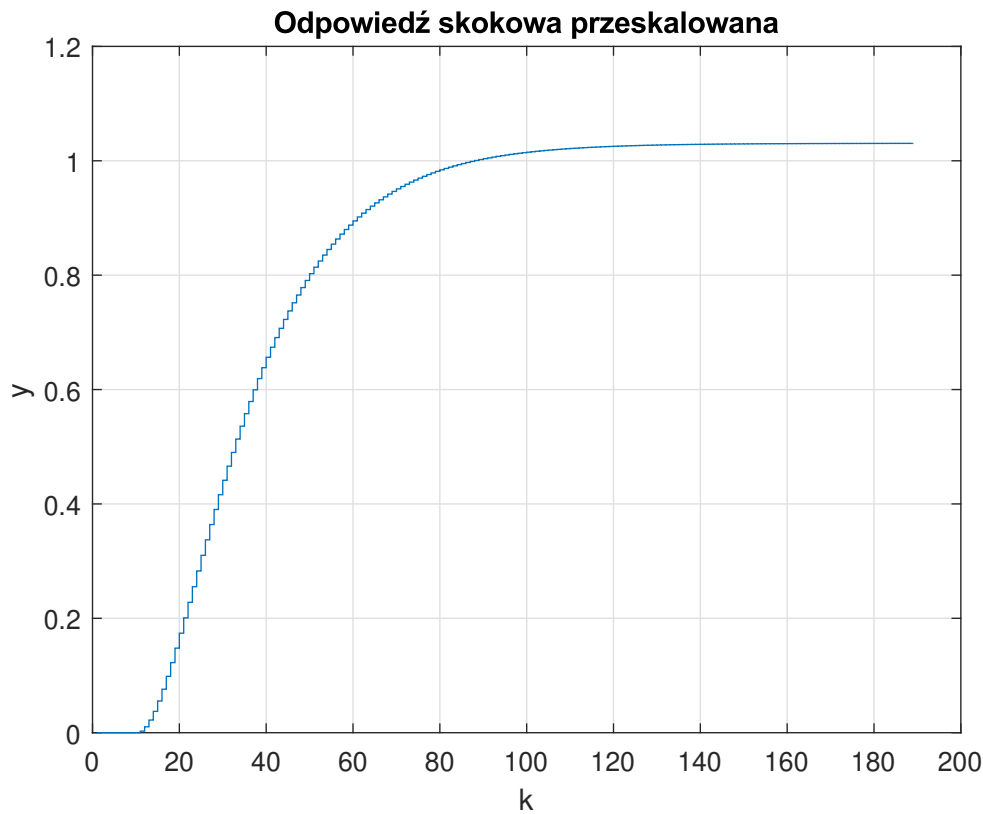
$s_i$  - kolejne wartości przeskalowanej odpowiedzi skokowej.

$y_i$  - wartości wyjścia nieprzeskalowanej odpowiedzi w kolejnych chwilach.

$Y_{pp}$  - wartość  $y$  w chwili  $k = 0$ .

$\delta u$  - wartość skoku sterowania.

Odpowiedź skokowa, którą można wykorzystać w algorytmie DMC przedstawiona jest na rysunku 1.4



Rys. 1.4. Odpowiedź skokowa przeskalowana

### 1.4. Symulacja cyfrowego algorytmu PID oraz algorytmu DMC

Napisaaliśmy program do symulacji cyfrowego algorytmu PID oraz algorytmu DMC z uwzględnieniem ograniczeń wartości sygnału sterującego oraz ograniczeń szybkości jego zmian:

$$0,1 \leq U(k) \leq 1,5 \quad (1.2)$$

$$-0,2 \leq \Delta U(k) \leq 0,2 \quad (1.3)$$

Prawo regulacji cyfrowego regulatora PID:

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

$$r_0 = K \left( 1 + \frac{T}{2T_i} + \frac{T_d}{T} \right) \quad (1.5)$$

$$r_1 = K \left( \frac{T}{2T_i} - \frac{2T_d}{T} - 1 \right) \quad (1.6)$$

$$r_2 = K \frac{T_d}{T} \quad (1.7)$$

Wartości  $K$  - wzmacnienie,  $T_d$  - stała różniczkowania,  $T_i$  - stała całkowania dobieramy tak samo jak w ciągłym regulatorze PID, natomiast  $T$  - okres próbkowania zależy od konkretnego zadania, w naszym przypadku wynosi 0,5. W każdej chwili dyskretnej wyznaczane jest wyjście obiektu, oraz uchyb i na tej podstawie zgodnie ze wzorem 1.4 wyliczane jest sterowanie

Fragment głównej pętli algorytmu PID zaimplementowanej w Matlabie:

```
for k=12:Ts
    %symulacja obiektu
    Y(k)=symulacja_obiektu7Y_p1(U(k-10),U(k-11),Y(k-1),Y(k-2));

    %uchyb regulacji
    e(k)=yzad(k)-Y(k);

    %sygnał sterujący regulatora PID
    U(k)=r2*e(k-2)+r1*e(k-1)+r0*e(k)+U(k-1);
```

Dla PID przed główną pętlą dodaliśmy obliczenia parametrów regulatora, wartości zadane, które mają się zmieniać po odpowiednio dużej ilości chwil  $k$ . W głównej pętli po otrzymaniu najnowszego wyjścia  $y$  procesu liczymy w następującej kolejności: uchyb, sterowanie, a także uwzględniamy ograniczenia sterowania.

Algorytm DMC jest bardziej skomplikowanym algorytmem w porównaniu do dyskretnego PID, jego celem jest minimalizacja wskaźnika jakości  $J$ .

$$J(k) = \sum_{p=1}^N (y^{zad}(k+p|k) - \bar{y}(k+p|k))^2 + \lambda \sum_{p=0}^{N_k=1} (\Delta U(k+p))^2 \quad (1.8)$$

Poniżej przedstawiamy fragment głównej pętli algorytmu DMC:

```
for k=start:Ts
    %symulacja obiektu
    y(k)=symulacja_obiektu7Y_p1(u(k-10),u(k-11),y(k-1),y(k-2));

    %Obliczenie DU_p
    for d=1:(D-1)
        DU_p(d) = u(k-d) - u(k-d-1);
    end

    %Pomiar wyjścia
    Y = ones(N, 1) * y(k);
```

```
%Obliczenie Y_0
yo = M_p * DU_p + Y;

Y_zad = ones(N, 1) * yzad(k);

%Obliczenie sterowania
DU = K * (Y_zad - yo);
u(k) = u(k-1) + DU(1);
```

Obliczamy macierz  $K$  (licząc macierz  $M$ ), macierz  $M_p$  i inicjalizujemy macierz  $DU_p$ . W pętli głównej po pomiarze aktualnego wyjścia obliczamy macierz  $DU_p$ , uzupełniamy macierz  $Y$  (pomiar wyjścia),  $Y_o$ ,  $Y_{zad}$  (wartości zadane). W kolejnych liniijkach obliczamy sterowanie, uwzględniamy ograniczenia i tak jak dla PIDa publikujemy obliczoną wartość sterowania

### 1.5. Dobranie nastaw regulatora PID i parametrów algorytmu DMC metodą eksperymentalną

Nastawy regulatora PID wyznaczyliśmy eksperymentalnie. Najpierw znaleźliśmy nastawy które zapewniały w miarę dobrą regulację a następnie zmienialiśmy kolejne wartości parametrów regulatora PID, aż do uzyskania satysfakcjonującej jakości regulacji.

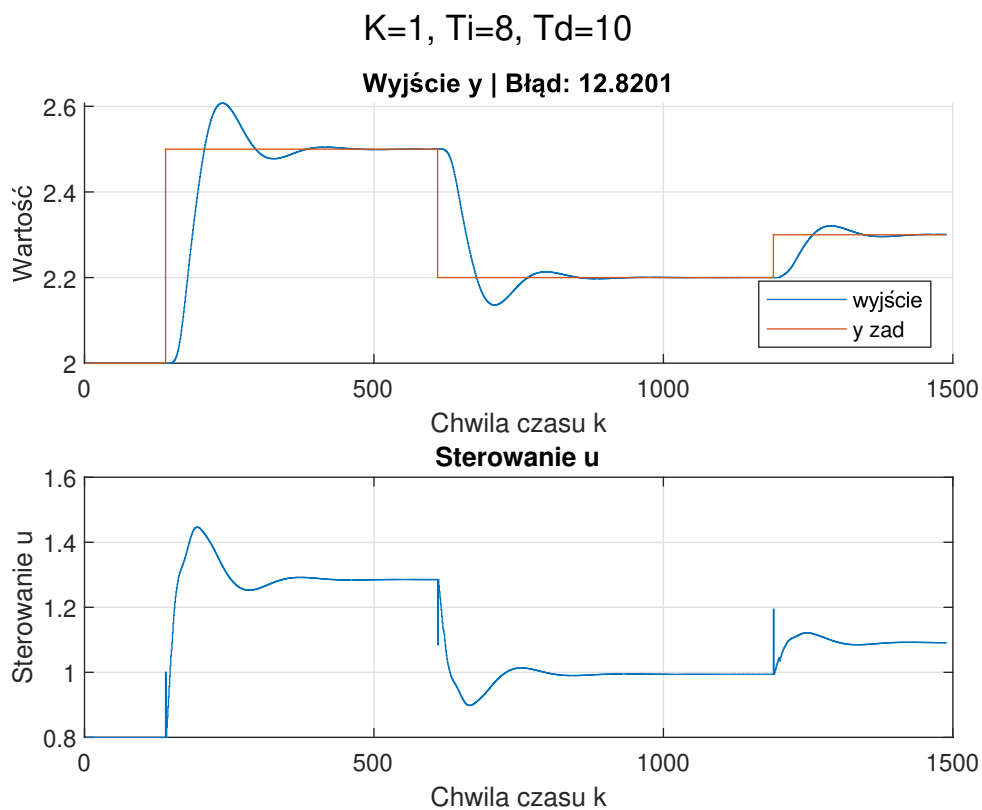
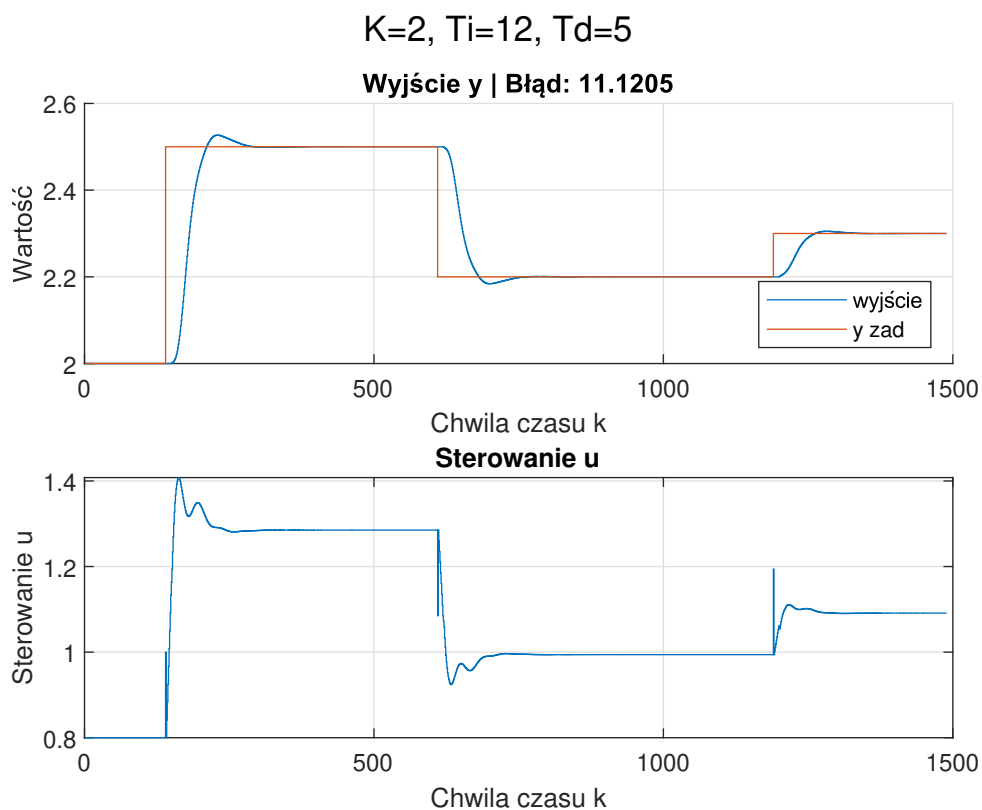
Dobór nastaw regulatora PID:

- Nastawy z Rys 1.5 powodują duże przeregulowanie i oscylacje na wyjściu procesu, potrzebny jest długi czas na osiągnięcie wartości zadanej, ponadto wyliczony błąd jest duży.
- Nastawy z Rys 1.6 są lepsze od poprzednich, wyjście szybciej osiąga wartość zadaną, przeregulowanie i wyliczony błąd są mniejsze. Sygnał sterujący zmienia się bardziej gwałtownie.
- Nastawy z Rys 1.7 są najlepszymi nastawami, jakie udało nam się osiągnąć strojąc regulator ręcznie. Wyjście bardzo szybko osiąga wartość zadaną, obserwujemy praktycznie brak oscylacji, a wyliczony błąd jest mały.

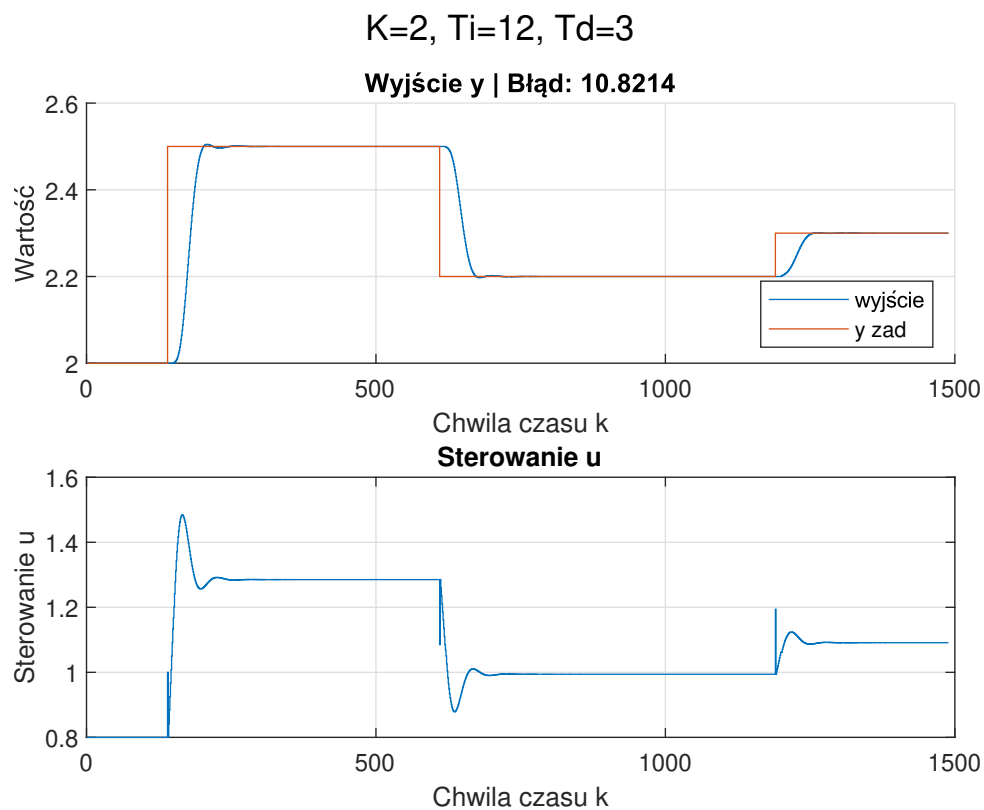
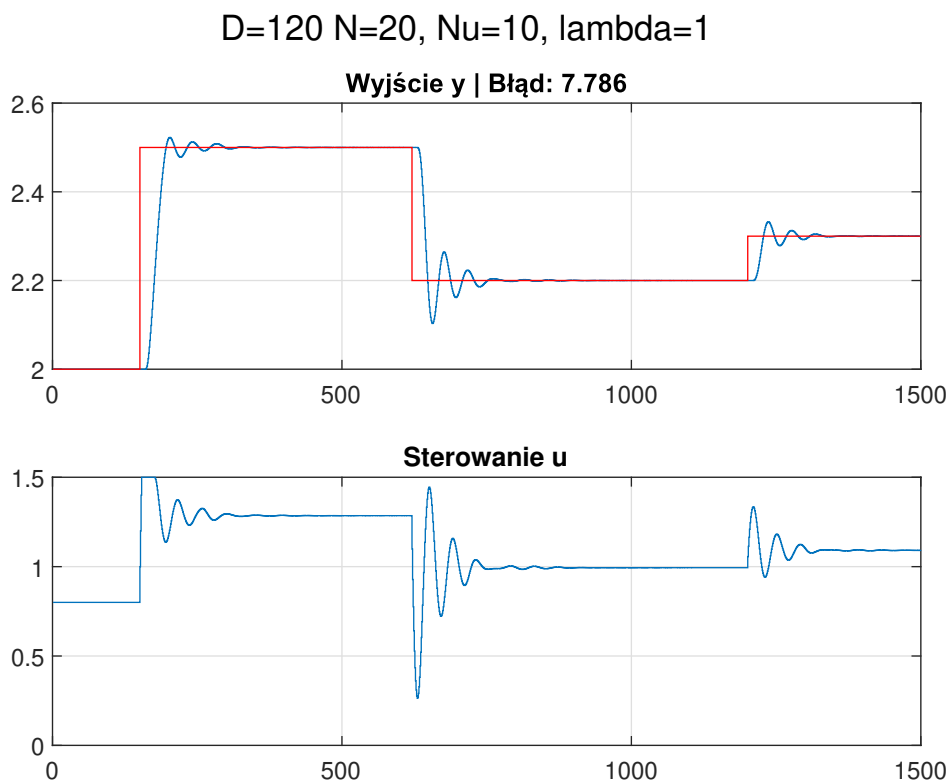
Nastawy regulatora DMC dobieraliśmy eksperymentalnie zmieniając wartości parametrów horyzontu predykcji  $N$ , horyzontu sterowania  $N_u$  oraz parametru  $\lambda$ . Jedynym stałym parametrem jest horyzont dynamiki  $D$  który wyznaczyliśmy na podstawie odpowiedzi skokowej i jest równy  $D = 120$ .

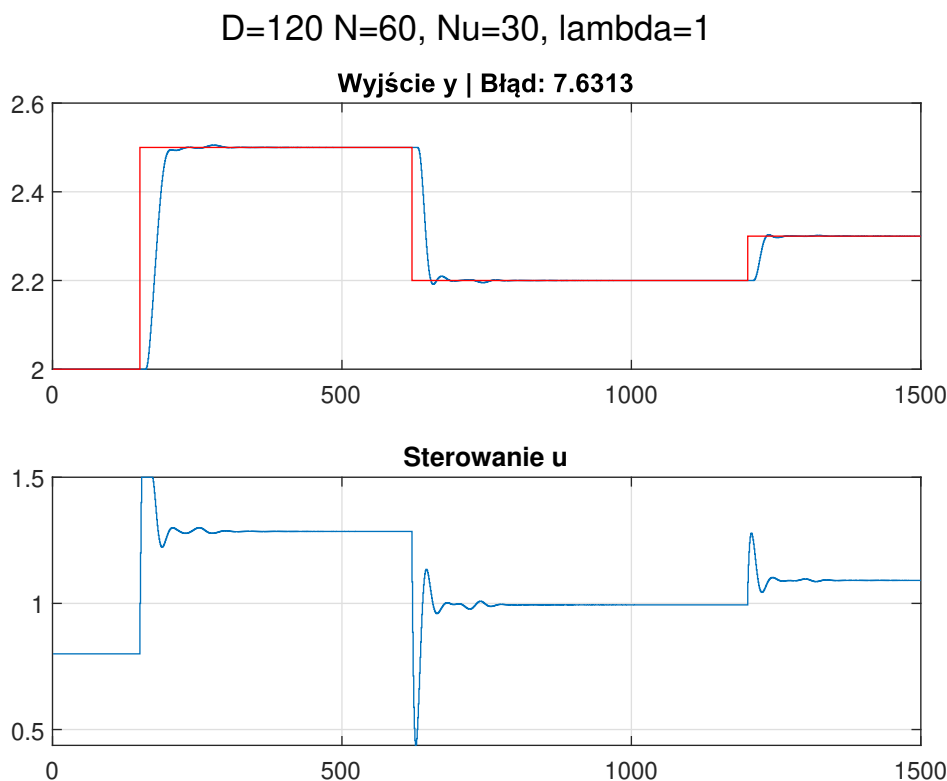
Dobór nastaw regulatora DMC:

- Nastawy z Rys 1.8 powodują niewielkie przeregulowanie, lecz dość długo trwające oscylacje na wyjściu procesu. Wyliczony błąd jest dużo mniejszy w porównaniu z regulatorem PID.
- Nastawy z Rys 1.9 są lepsze od poprzednich, obserwujemy bardzo niewielkie oscylacje na wyjściu procesu, przeregulowanie w zasadzie nie występuje, wyliczony błąd nieznacznie się zmniejszył. Sygnał sterujący zmienia się bardziej gwałtownie.

Rys. 1.5. Wyjście i sterowanie regulatora PID przy nastawach  $K = 1, T_i = 8, T_d = 10$ Rys. 1.6. Wyjście i sterowanie regulatora PID przy nastawach  $K = 2, T_i = 12, T_d = 5$



Rys. 1.7. Wyjście i sterowanie regulatora PID przy nastawach  $K = 2, T_i = 12, T_d = 3$ Rys. 1.8. Wyjście i sterowanie regulatora DMC przy nastawach  $D = 120, N = 20, N_u = 10, \lambda = 1$



Rys. 1.9. Wyjście i sterowanie regulatora DMC przy nastawach  $D = 120$ ,  $N = 60$ ,  $N_u = 30$   $\lambda = 1$

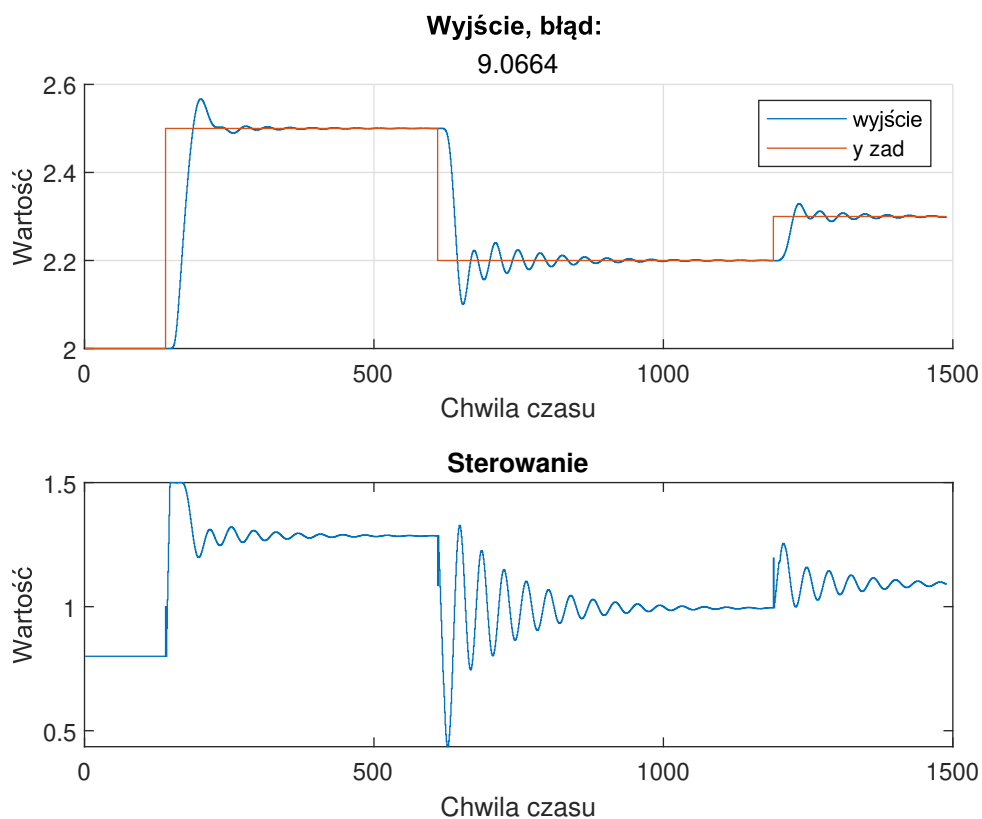
### 1.6. Dobranie nastaw regulatora PID i algorytmów DMC w wyniku optymalizacji wskaźnika jakości

Optymalizację wskaźnika jakości dokonaliśmy dla regulatora za pomocą funkcji *fmincon*. Ustawiliśmy punkt początkowy:  $K=5$ ,  $T_i=15$ ,  $T_d=5$ . Funkcja zwróciła nam następujące wartości parametrów:  $K=3,5858$ ,  $T_i=7,8832$ ,  $T_d=4,1766$ . Działanie regulatora PID z tymi nastawami przedstawia rysunek 1.10.

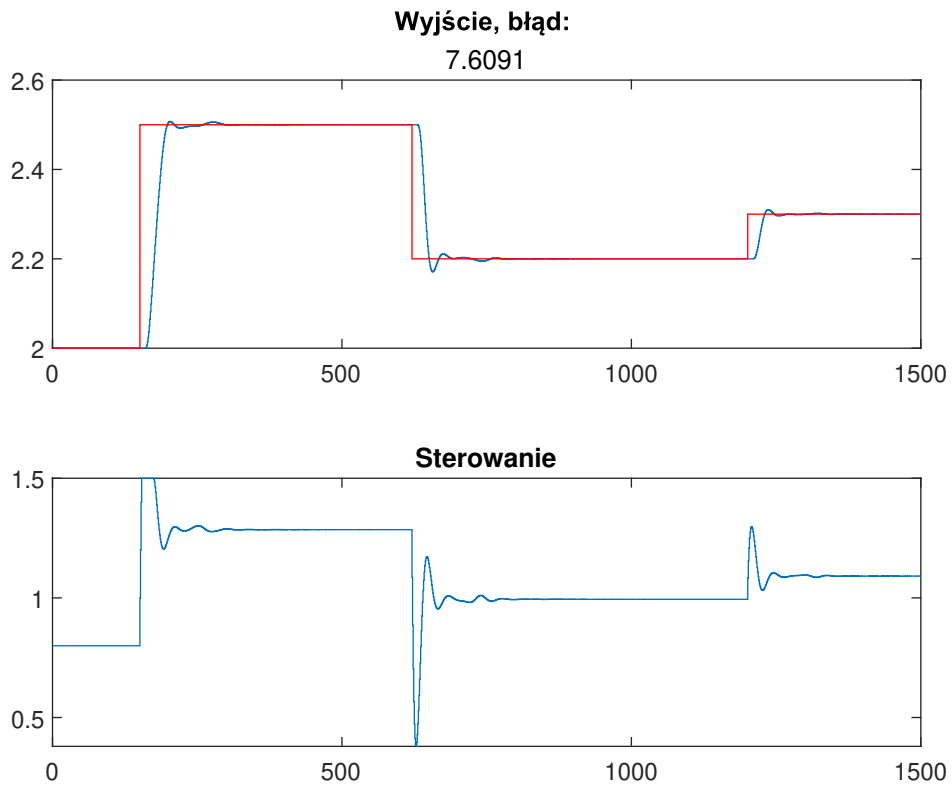
Wskaźnik jakości regulacji dla parametrów regulatora PID, otrzymanych wskutek optymalizacji wyniósł  $E=9,0664$ . Jest on mniejszy od metody eksperymentalnej. Na przebiegu występują niepożądane oscylacje, co sugeruje nieodpowiednie parametry, pomimo małego błędu. Ręczne nastrojenie regulatora dało lepsze rezultaty.

Optymalizację wskaźnika jakości dla regulatora DMC wykonaliśmy przy użyciu algorytmu genetycznego *ga*. Dla optymalizacji *ga* przyjęliśmy ograniczenia dolne:  $N_{min}=1$ ,  $N_{umin}=1$ ,  $T_{dmin}=0,1$ . I następujące ograniczenia górne:  $N_{max}=200$ ,  $N_{umax}=1$ ,  $T_{dmax}=0,1$ . Funkcja zwróciła nam następujące wartości parametrów:  $N=138$ ,  $N_u=35$ ,  $\lambda=0,8$ . Działanie regulatora z tymi parametrami jest przedstawione na rysunku 1.11

Zminimalizowany błąd wyniósł  $E=7,6091$ . Jest on mniejszy od metody eksperymentalnej. Przebieg wygląda akceptowalnie, co potwierdza poprawność doboru parametrów.



Rys. 1.10. Przebieg wyjścia i sterowania regulatora PID przy optymalizacji wskaźnika

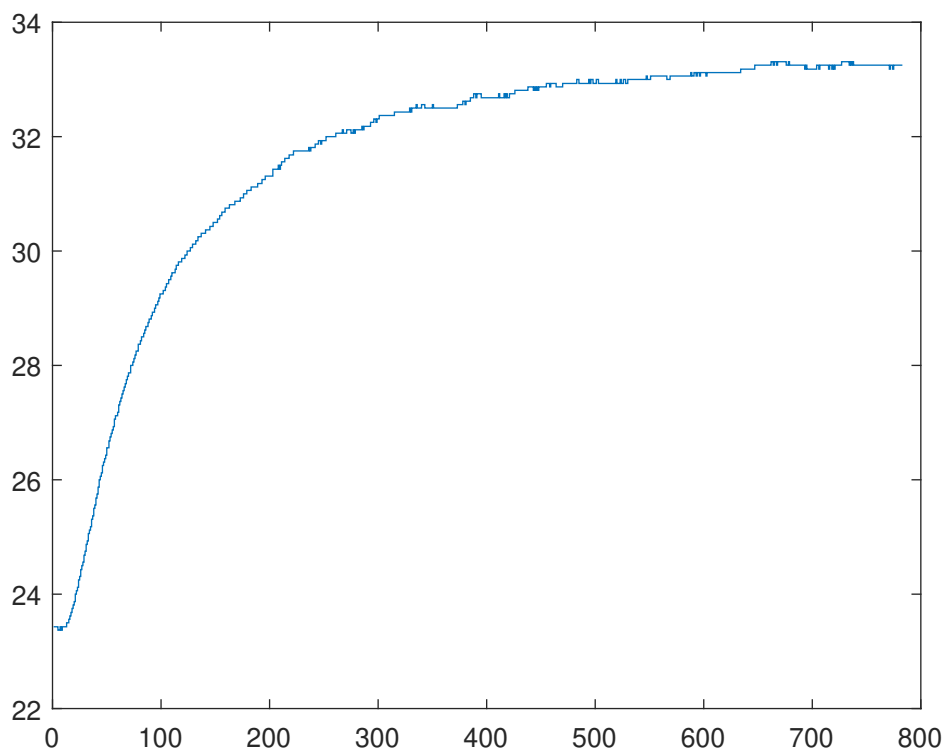


Rys. 1.11. Przebieg wyjścia i sterowania regulatora DMC przy optymalizacji wskaźnika

## 2. Część laboratoryjna

### 2.1. Sprawdzenie możliwości sterowania i pomiaru w komunikacji ze stanowiskiem

Uruchomiliśmy skrypt *MinimalWorkingExample* z ustawionymi parametrami  $W1 = 50$  i  $G1 = 27$ . Zatrzymaliśmy powyższy skrypt w momencie, gdy zauważyliśmy, że temperatura jest już stabilna (jak na warunki laboratoryjne tzn.: przechodzące osoby i tymczasowy przeciąg). Otrzymane dane umieściliśmy na wykresie: 2.1.

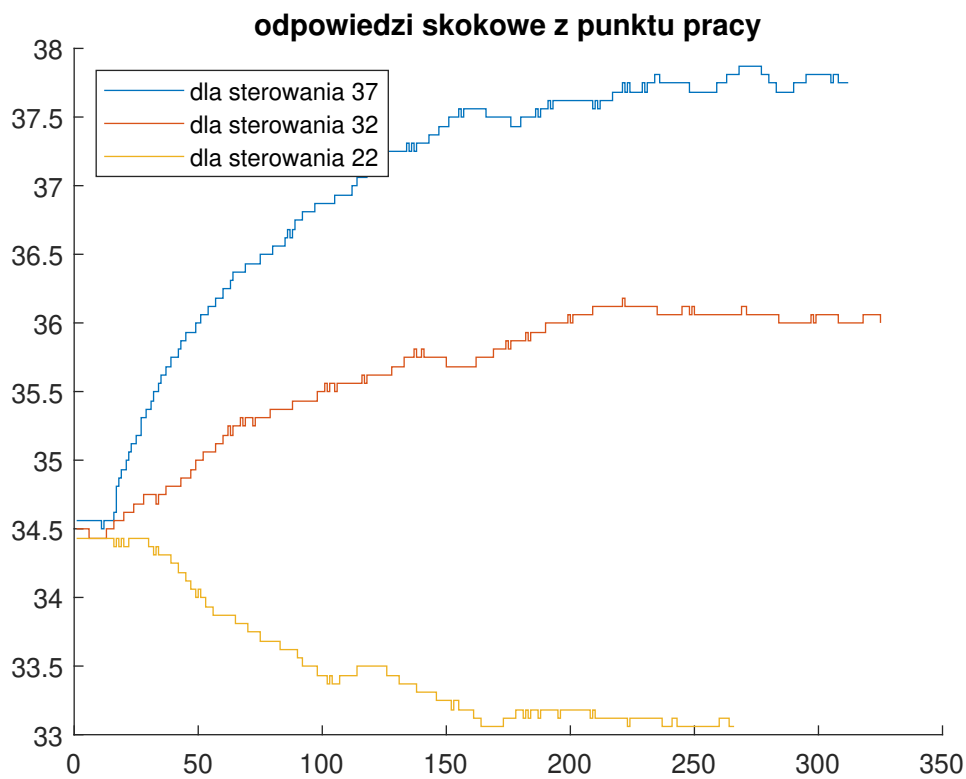


Rys. 2.1. Pierwsze uruchomienie stanowiska - otrzymany punkt pracy

Z wykresu 2.1 można odczytać, że punkt pracy wynosi około  $33^{\circ}\text{C}$ . W późniejszych zadaniach z racji zauważalnego wzrostu temperatury w sali laboratoryjnej punkt pracy wzrósł o około  $0,6^{\circ}\text{C}$ .

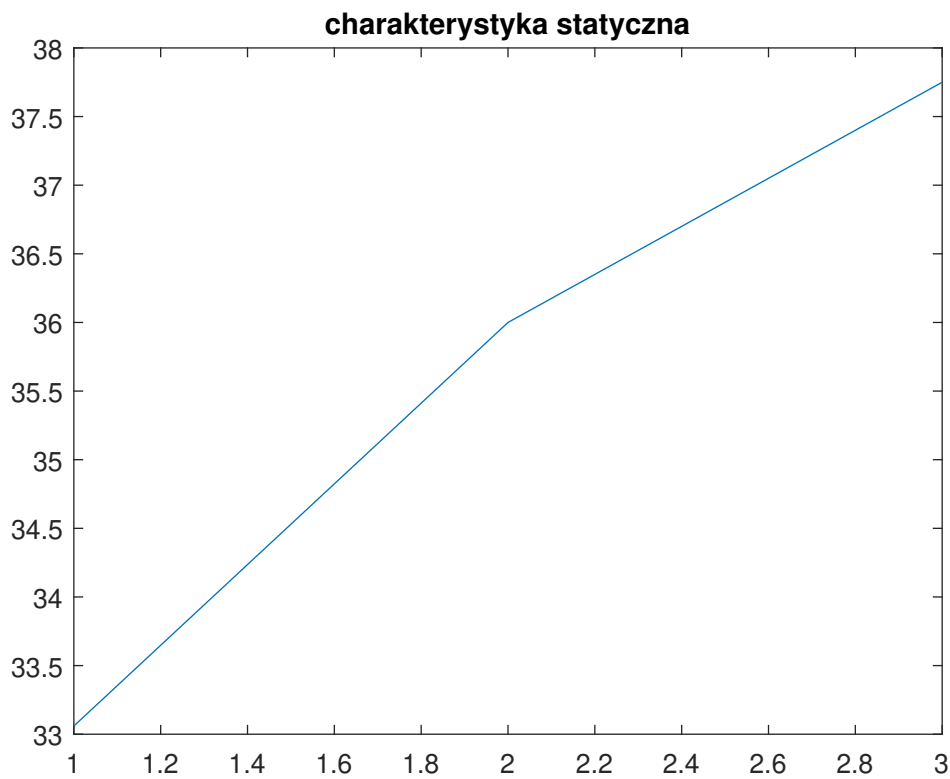
### 2.2. Wyznaczenie odpowiedzi skokowych

Znając punkt pracy wyznaczyliśmy 3 odpowiedzi skokowe (skoki sterowania do 22, 32 i 37). Dla powyższych skoków otrzymaliśmy odpowiedzi skokowe: 2.2.



Rys. 2.2. Odpowiedzi skokowe (skoki do 22, 32 i 37)

Zauważalny jest powolny wzrost punktu pracy. Otrzymaną charakterystykę statyczną 2.2

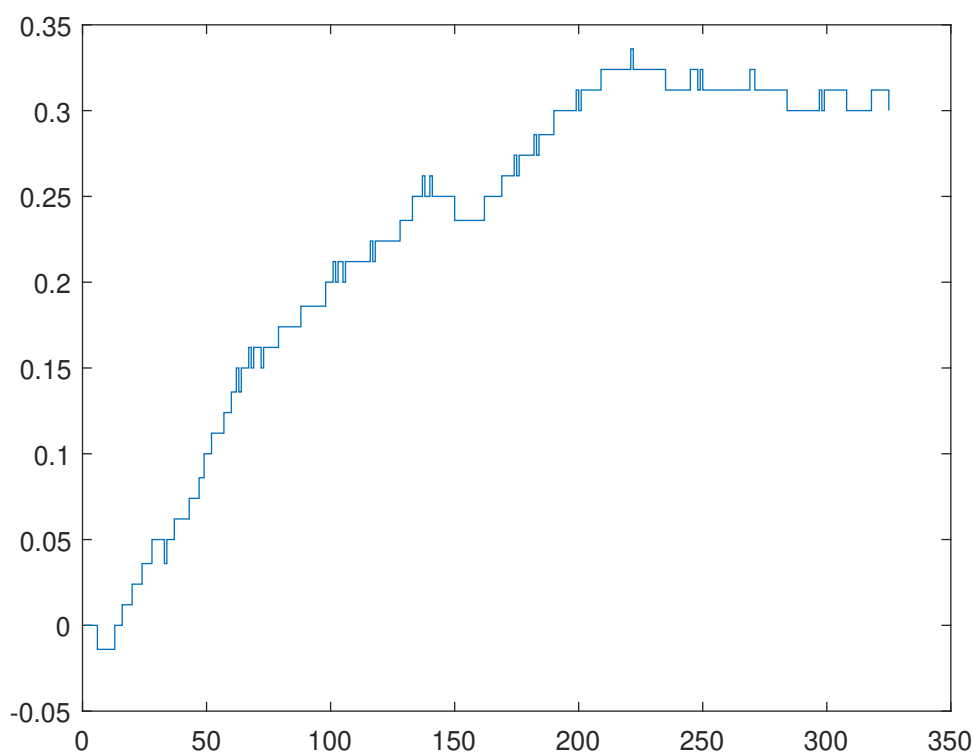


Rys. 2.3. Charakterystyka statyczna wyznaczona na podstawie 3 odpowiedzi skokowych

można uznać za liniową. Jej niedokładność wynika z niedużej ilości skoków użytych do otrzymania jej i wzrostu punktu pracy.

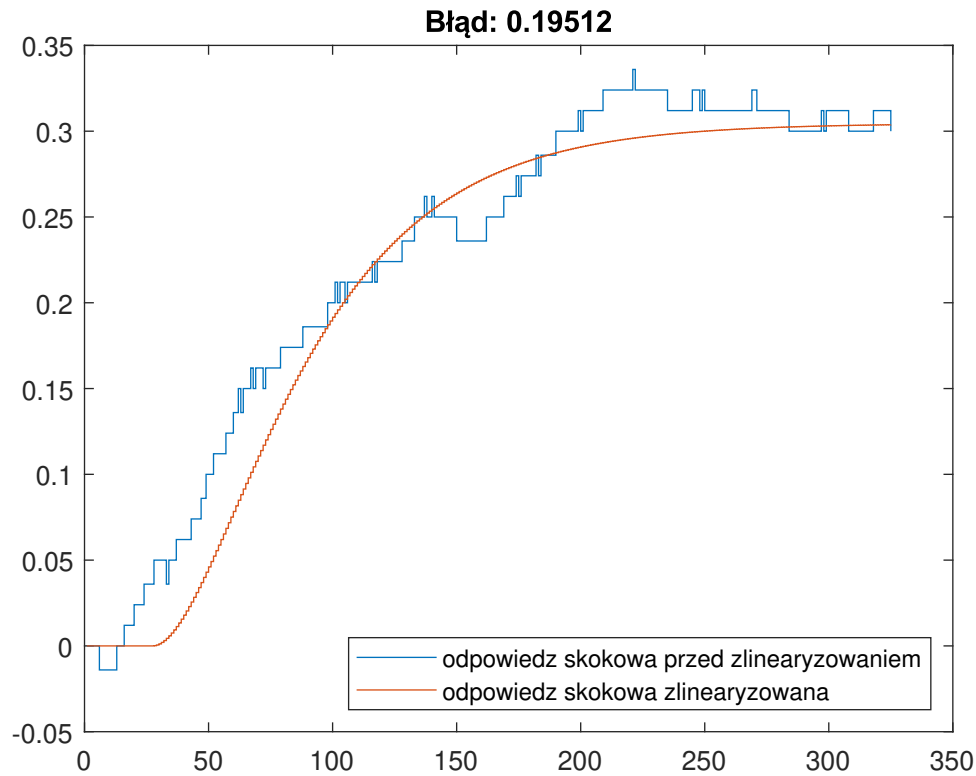
### 2.3. Otrzymanie odpowiedzi skokowej do algorytmu DMC

Do otrzymania odpowiedzi skokowej nadającej się do użycia w regulatorze DMC wybraliśmy odpowiedź uzyskaną ze skoku sterowania z punktu pracy (27) do wartości 32. Następnie korzystając z wzoru 1.1 otrzymujemy odpowiedź skokową przedstawioną na rysunku: 2.4.



Rys. 2.4. Charakterystyka statyczna dla skoku sterowania z 27 do 32

Następnie przekształconą odpowiedź skokową aproksymujemy, używając w tym celu członu inercyjnego drugiego rzędu z opóźnieniem. Otrzymujemy odpowiedź skokową przedstawioną na rysunku 2.5:



Rys. 2.5. Charakterystyka statyczna po linearyzacji

W celu znalezienia najlepszych parametrów, użyliśmy funkcji *fmincon*. Otrzymane parametry:  $T_1 = 44, 57$ ,  $T_2 = 24, 67$ ,  $K = 0, 3$ ,  $T_d = 15$ .

Otrzymana odpowiedź jest dobrym przybliżeniem. Widać to w po bardzo dobrze znalezionym wzmacnieniu które pokrywa się z częścią przed aproksymacją.

## 2.4. Cyfrowy algorytm PID i DMC

W celu napisania programów do regulacji PID oraz DMC, odpowiednio zmodyfikowaliśmy plik *MinimalWorkingExample*.

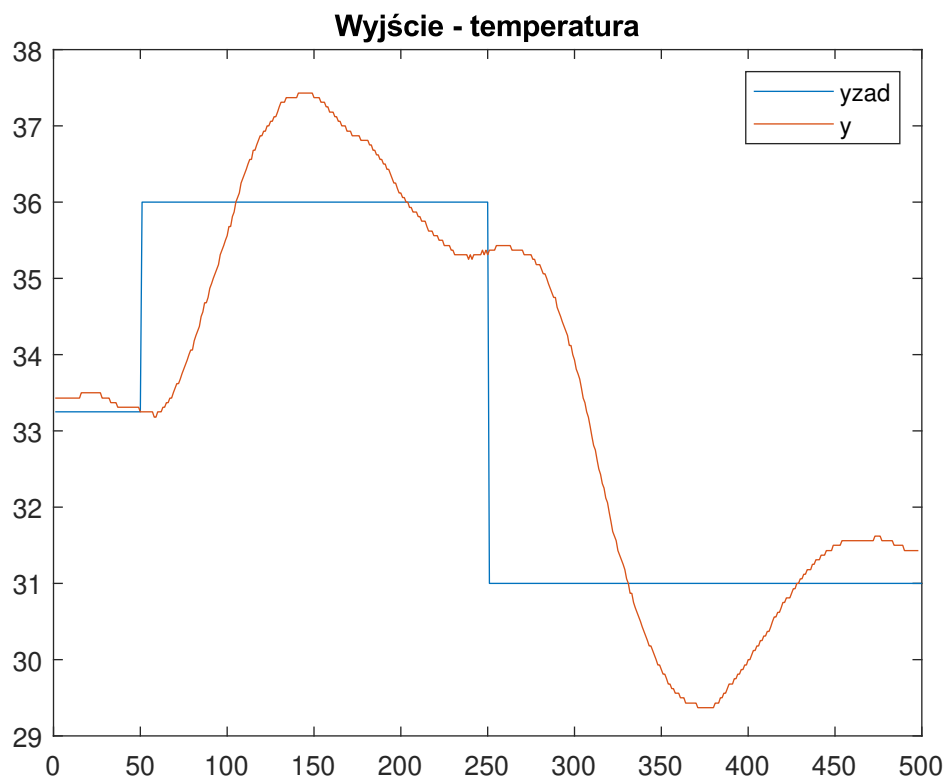
Dla PID dodaliśmy przed główną pętlą obliczenia parametrów regulatora, wartości zadane, które mają się zmieniać po odpowiednio dużej ilości chwil  $k$  (okres wynosi około sekundę, gdyż jest to jedna iteracja programu, a program w każdej iteracji czeka sekundę na nowe dane). W głównej pętli po otrzymaniu najnowszych pomiarów temperatury liczymy w następującej kolejności: uchyb, sterowanie, uwzględniamy ograniczenia obiektu ( $0 < G1 < 100$ ). Na końcu wyliczone sterowanie ustawiamy w odpowiednim miejscu w funkcji *sendControls*, która wysyła je do grzałki.

Dla DMC pobieramy model (obliczony w zadaniu 3), ustalamy parametry regulatora, wartości zadane. Obliczamy macierz  $K$  (licząc macierz  $M$ ), macierz  $M_p$  i inicjalizujemy macierz  $DU_p$ . W pętli głównej po pomiarze aktualnej temperatury obliczamy macierz  $DU_p$ , uzupełniamy macierze  $Y$  (pomiar wyjścia),  $Y_o$ ,  $Y_{zad}$  (wartości zadane). W kolejnych liniach obliczamy sterowanie, uwzględniamy ograniczenia i tak jak dla PIDa publikujemy obliczoną wartość sterowania.

## 2.5. Dobranie nastaw regulatora PID i algorytmu DMC metodą eksperymentalną

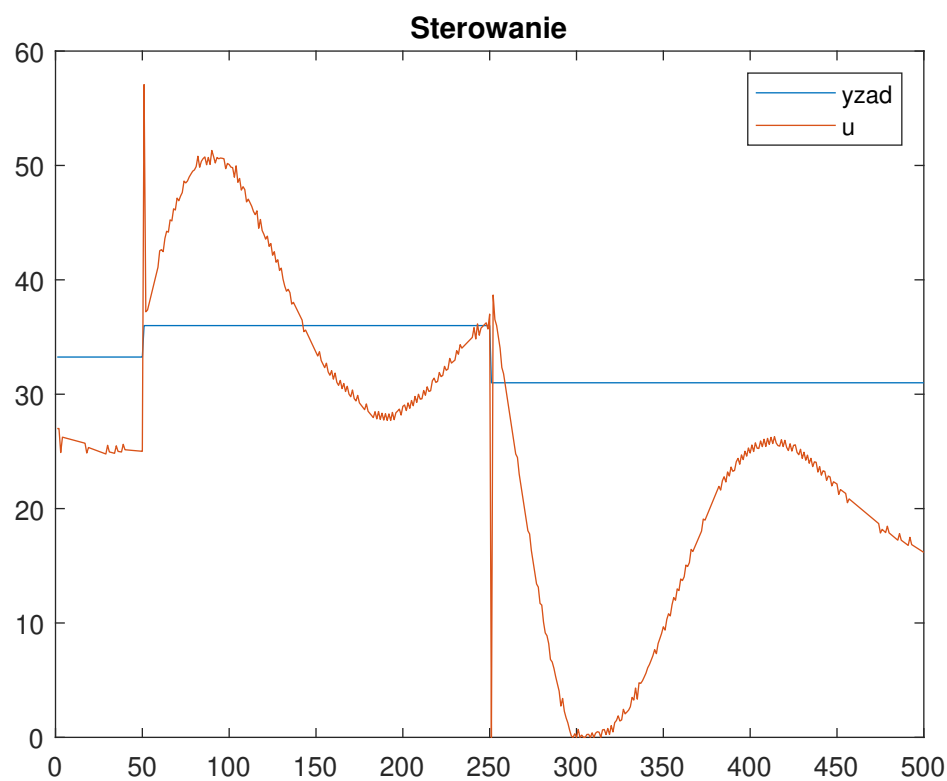
Rezultaty dla PID:

1. Pierwsza próba strojenia:
  - Parametry regulatora ciągłego:  $K = 3,85$ ,  $T_d = 1$ ,  $T_i = 17$ .
  - Parametry regulatora dyskretnego:  $r_0 = 11,66$ ,  $r_1 = -19,14$ ,  $r_2 = 7,7$ .
  - Otrzymany łączny błąd: 603,8. Średni błąd na pojedynczą chwilę: 1,207
  - Wyjście jest przedstawione na rysunku: 2.6, a sterowanie na rysunku 2.7
2. Druga próba strojenia:
  - Parametry regulatora ciągłego:  $K = 5$ ,  $T_d = 1$ ,  $T_i = 20$ .
  - parametry dyskretnego:  $r_0 = 10,125$ ,  $r_1 = -14,875$ ,  $r_2 = 5$
  - Otrzymany łączny błąd: 506,3. Średni błąd na pojedynczą chwilę: 0,675
  - Wyjście jest przedstawione na rysunku: 2.8, a sterowanie na rysunku 2.9

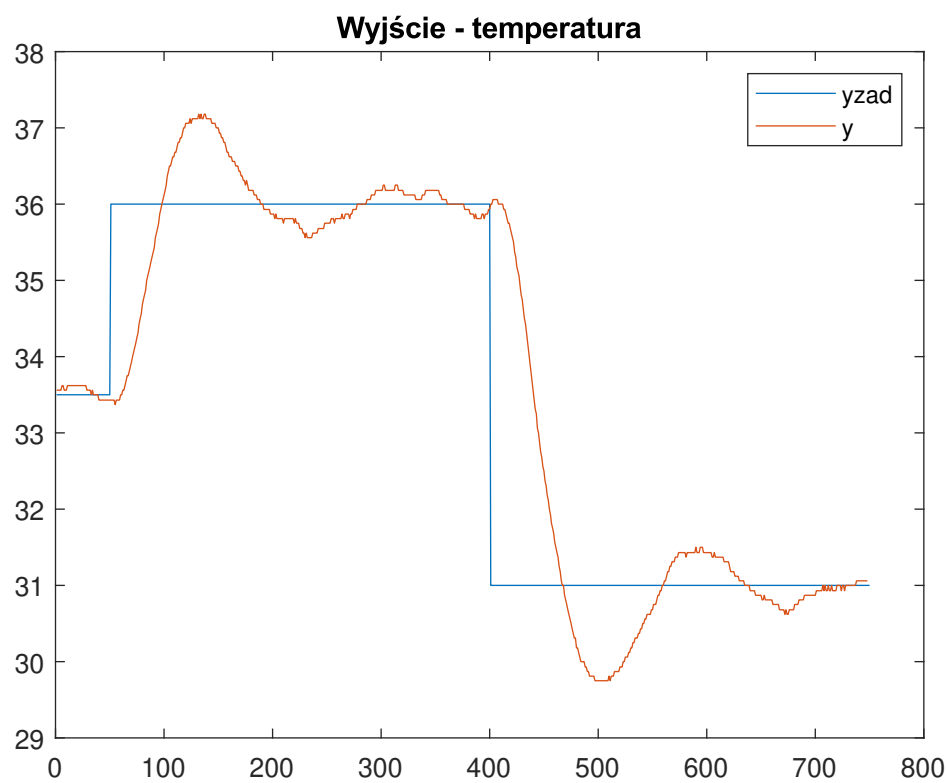


Rys. 2.6. Wyjście testowanego regulatora PID (1)

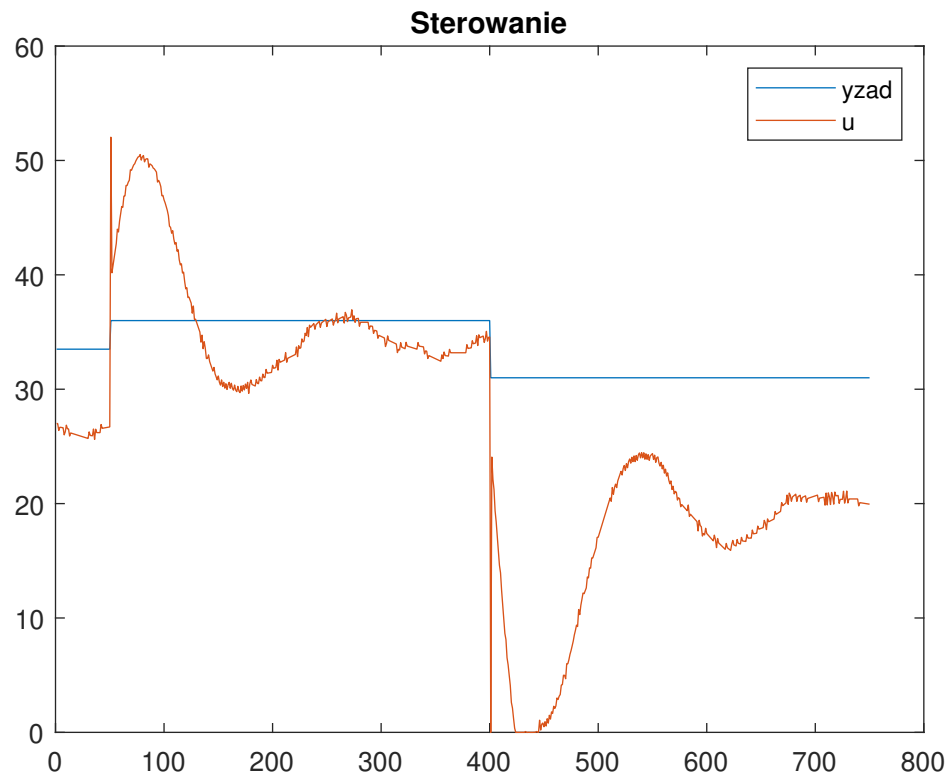




Rys. 2.7. Sterowanie testowanego regulatora PID (1)



Rys. 2.8. Wyjście testowanego regulatora PID (2)



Rys. 2.9. Sterowanie testowanego regulatora PID (2)

Rezultaty dla DMC:

1. Pierwsza próba strojenia:

- Parametry regulatora:  $D = 250$ ,  $N = 260$ ,  $N_u = 260$ ,  $\lambda = 1$ .
- Otrzymany łączny błąd: 516,97. Średni błąd na pojedynczą chwilę: 0,608
- Wyjście jest przedstawione na rysunku: 2.10, a sterowanie na rysunku 2.11

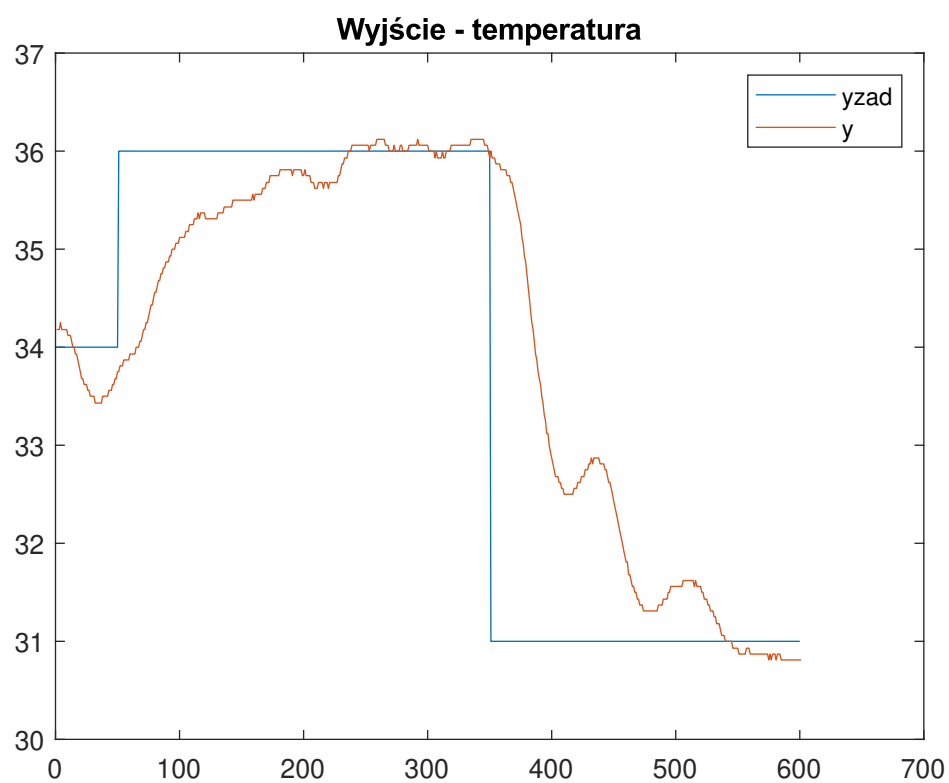
2. Druga próba strojenia:

Zauważyliśmy błąd w implementacji i naprawiliśmy początkowy spadek sterowania do 0.

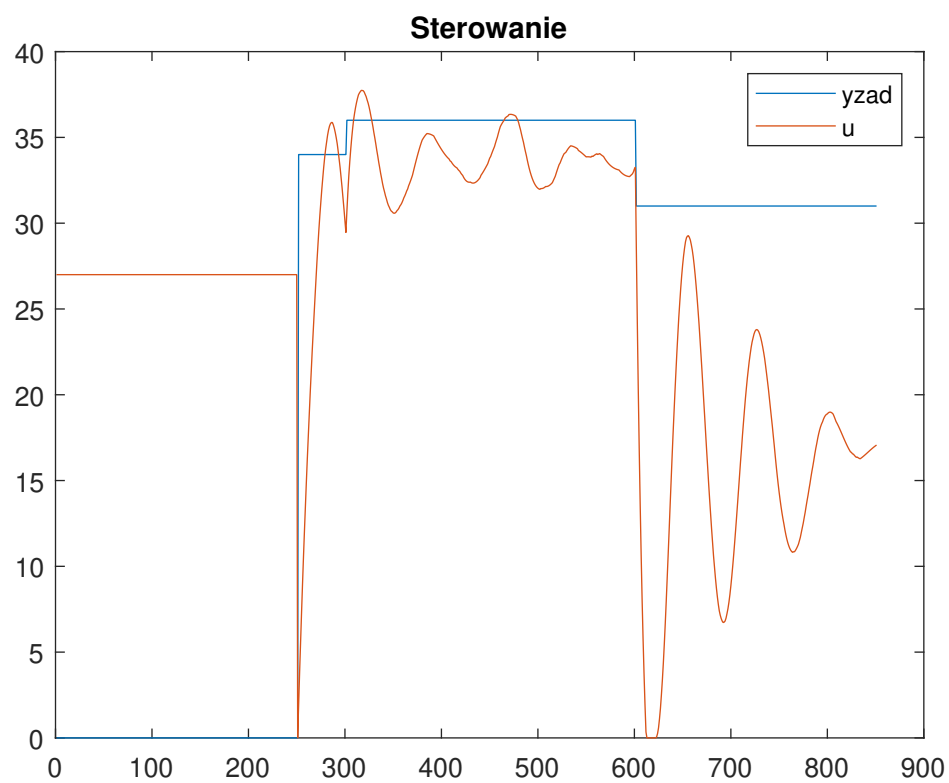
- Parametry regulatora:  $D = 250$ ,  $N = 260$ ,  $N_u = 260$ ,  $\lambda = 1$ .
- Otrzymany łączny błąd: 487,12. Średni błąd na pojedynczą chwilę: 0,57.
- Wyjście jest przedstawione na rysunku: 2.12, a sterowanie na rysunku 2.13

3. Trzecia próba strojenia:

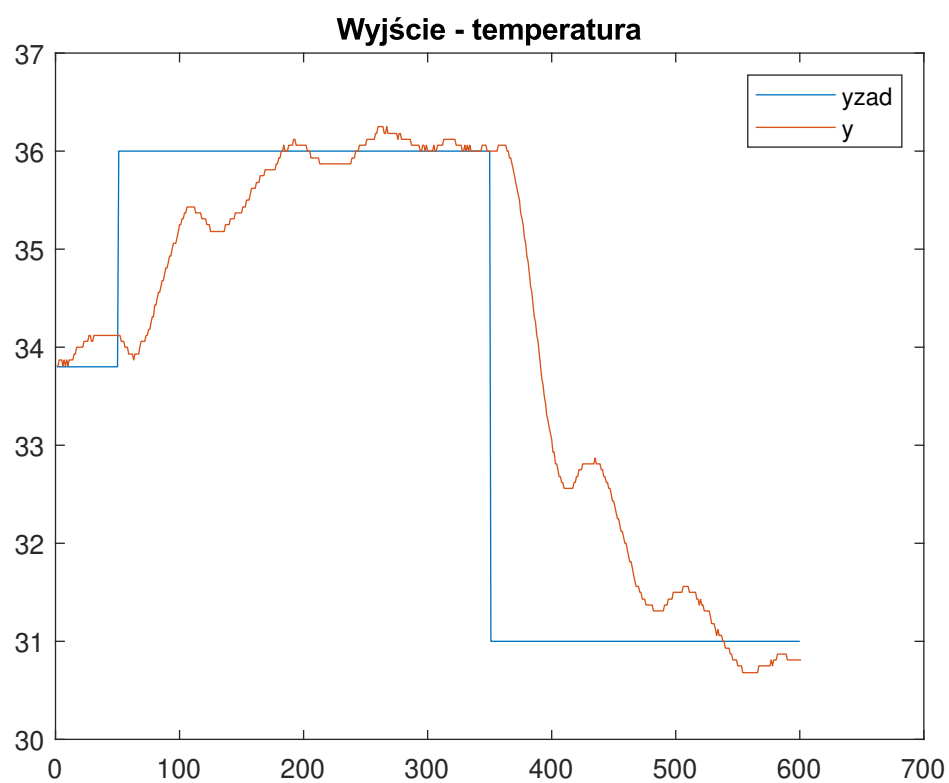
- Parametry regulatora:  $D = 250$ ,  $N = 260$ ,  $N_u = 260$ ,  $\lambda = 1$ .
- Otrzymany łączny błąd: 299,48. Średni błąd na pojedynczą chwilę: 0,352
- Wyjście jest przedstawione na rysunku: 2.14, a sterowanie na rysunku 2.15



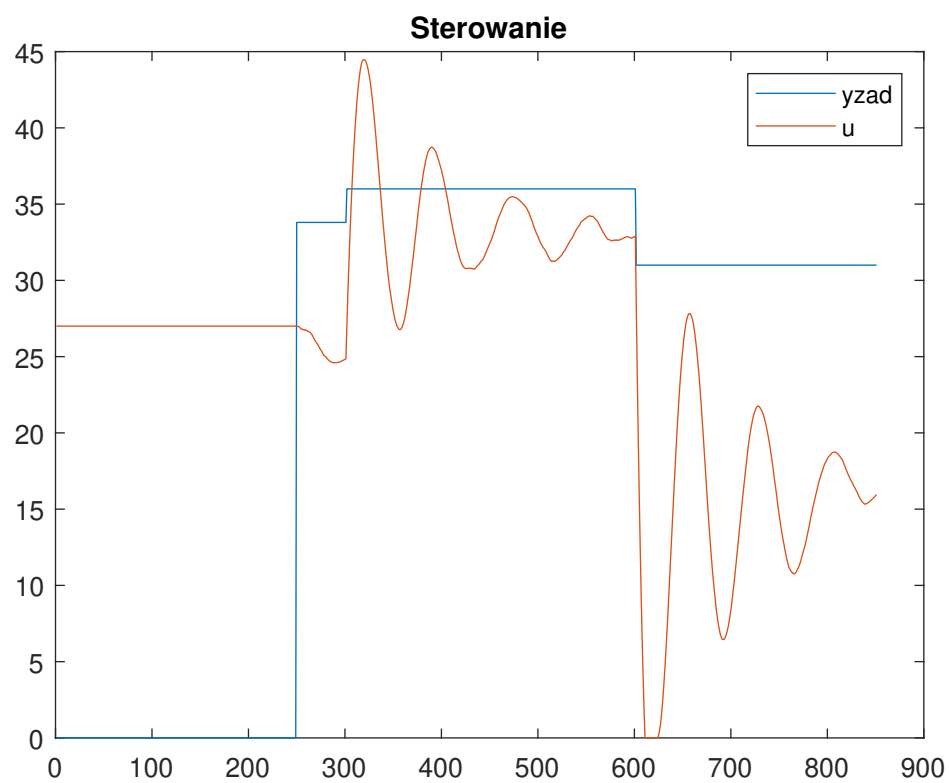
Rys. 2.10. Wyjście testowanego regulatora DMC (3)



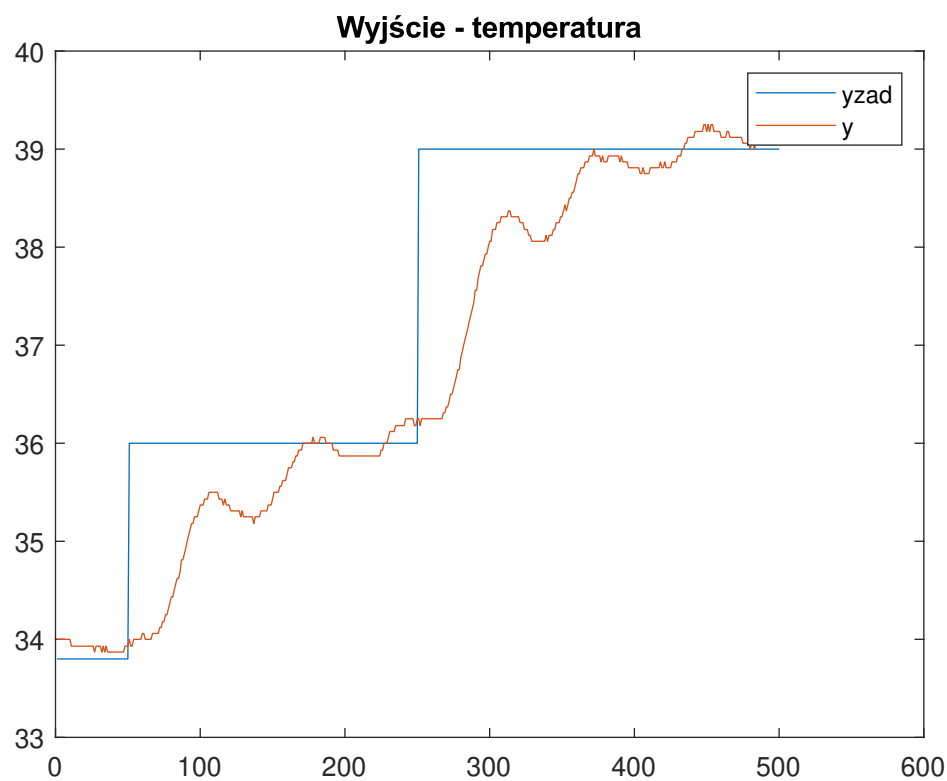
Rys. 2.11. Sterowanie testowanego regulatora DMC (3)



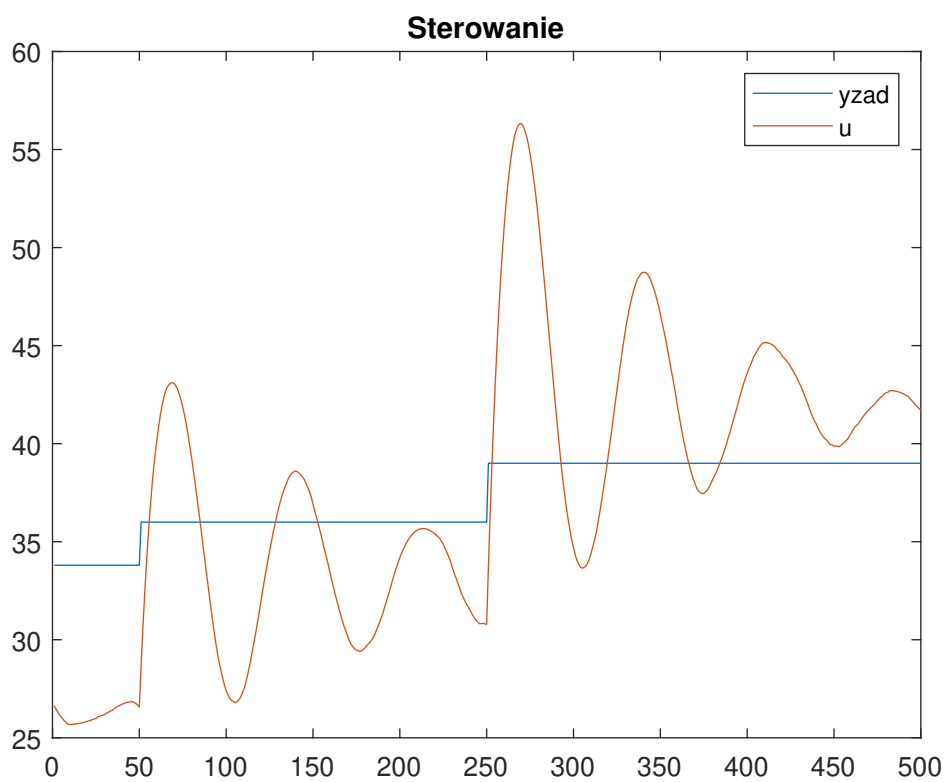
Rys. 2.12. Wyjście testowanego regulatora DMC (4)



Rys. 2.13. Sterowanie testowanego regulatora DMC (4)



Rys. 2.14. Wyjście testowanego regulatora DMC (5)



Rys. 2.15. Sterowanie testowanego regulatora DMC (5)

Przy wszystkich próbach z regulatorem DMC można zauważyć między innymi oscylacje sygnału sterującego. Wynika to prawdopodobnie niewystarczające dostrójenie regulatora.

Zauważyliśmy również, że używany przez DMC model odpowiedzi skokowej był wykonany dla małego skoku sterowania co skutkowało niewielkim wzrostem temperatury (o około  $1.5^{\circ}\text{C}$ ), również to mogło mieć wpływ na ostateczne wyniki. Żeby regulator działał lepiej należało by poświęcić więcej czasu na dostrojenie algorytmu i poprawienie odpowiedzi skokowej. W wyniku eksperymentów, zauważyć można, że nie najlepiej dostrojony regulator DMC osiągnął mniejszy błąd niż regulator PID.