

Wydział Elektroniki i Technik Informacyjnych
Politechnika Warszawska

Projektowanie układów sterowania
(projekt grupowy)

Sprawozdanie z projektu i ćwiczenia laboratoryjnego
nr 2, zadanie nr 12

Paulina Dąbrowska, Miłosz Kowalewski,
Adam Rybojad, Mikołaj Wewiór

Warszawa, 2023

Spis treści

1. CZĘŚĆ PROJEKTOWA	2
1.1. Poprawność punktu pracy	2
1.2. Odpowiedzi skokowe	3
1.2.1. Wejście - wyjście	3
1.2.2. Zakłócenie - wyjście	4
1.2.3. Wzmocnienie statyczne obu torów procesu	5
1.2.4. Charakterystyka statyczna procesu $y(u, z)$	5
1.3. Odpowiedzi skokowe algorytmu DMC	6
1.4. Symulacja regulacji DMC	7
1.4.1. Implementacja	7
1.4.2. Dobór parametrów i ocena jakości	10
1.5. Regulacja DMC z zakłóceniem	12
1.6. Regulacja przy sinusoidalnym zakłóceniu	13
1.7. Regulacja przy szumie pomiarowym	14
2. CZĘŚĆ LABORATORYJNA	15
2.1. Punkt pracy stanowiska	15
2.2. Odpowiedzi skokowe zakłócenie-wyjście	15
2.3. Aproksymacja odpowiedzi skokowej	16
2.4. Regulacja DMC	19
2.4.1. Dobór parametrów	20
2.4.2. Podsumowanie	24

1. CZĘŚĆ PROJEKTOWA

1.1. Poprawność punktu pracy

Sprawdzenie poprawności punktów pracy polegało na obserwacji wyjścia symulowanego obiektu projektowego *12y_p2* przy niezmiennym sterowaniu *U_{pp}* oraz stałymi zakłoceniami. Punkty pracy *U_{pp}* oraz *Y_{pp}* odczytaliśmy po wywołaniu w terminalu Matlab: *symulacja_obiektu12y_p1*. W tym celu skorzystano z funkcji:

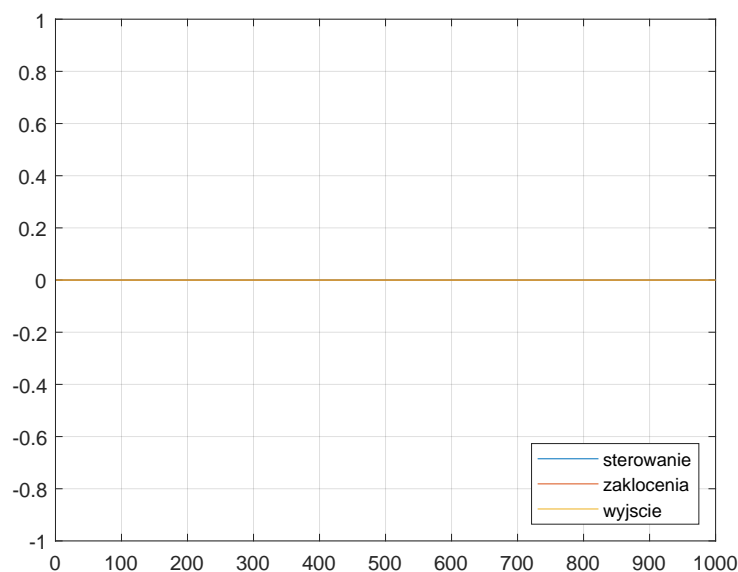
```
% inicjalizacja
% (będzie wykorzystywana w kolejnych zadaniach/listingach)

T = 0.5;      % czas próbkowania
n = 1000;     % czas trwania symulacji
start = 9;    % moment rozpoczęcia pomiarów

Upp = 0;  u(1:n) = Upp;
Ypp = 0;  y(1:start-1) = Ypp;
Zpp = 0;  z(1:n) = Zpp;

% koniec inicjalizacji

for k = start:n
    y(k) = symulacja_obiektu12y_p2(u(k-7), u(k-8), ...
        z(k-4), z(k-5), y(k-1), y(k-2));
end
```



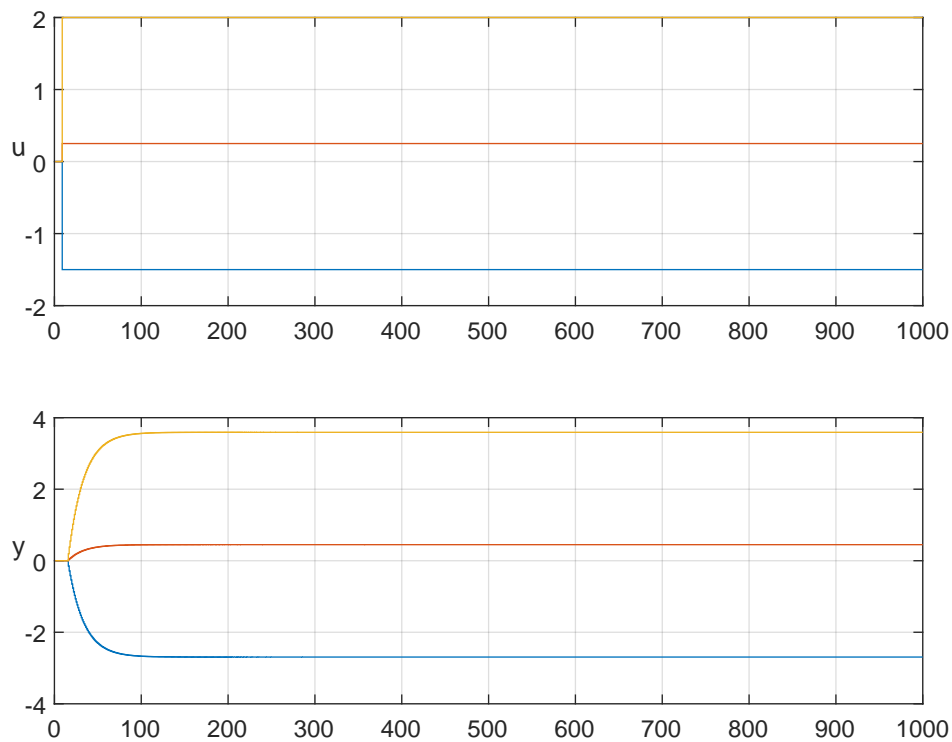
Rys. 1. punkt pracy

Jak widać, przy niezmiennym sterowaniu o wartości 0 i stałych zakłóceniach równych 0, wyjście obiektu nie zmienia się, pozostając równe 0, co potwierdza tym samym, że punkt pracy jest stabilny.

1.2. Odpowiedzi skokowe

1.2.1. Wejście - wyjście

```
Uskok = {-1.5, 0.25, 2};  
  
for iter_u = 1:size(Uskok(:))  
    u(start:n) = Uskok{iter_u};  
  
    for k = start:n  
        y(k) = symulacja_obiektu12y_p2(u(k-7), u(k-8), ...  
            z(k-4), z(k-5), y(k-1), y(k-2));  
    end  
end
```



Rys. 2. odpowiedź skokowa w torze wejście-wyjście

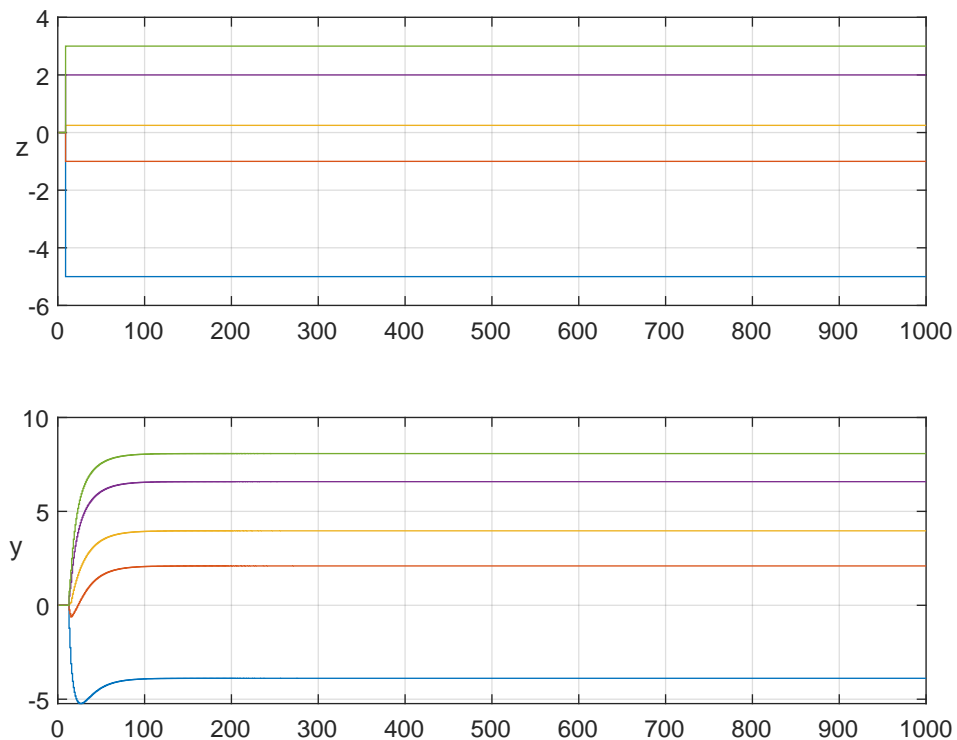
Właściwości dynamiczne obiektu dla toru wejście-wyjście są liniowe - dla danej zmiany sterowania wyjście reaguje proporcjonalnie do jego zmiany

1.2.2. Zakłócenie - wyjście

```
Zskok = {-5, -1, 0.25, 2, 3};

for iter_z = 1:size(Zskok(:))
    z(start:n) = Zskok{iter_z};

    for k = start:n
        y(k) = symulacja_obiektu12y_p2(u(k-7), u(k-8), ...
            z(k-4), z(k-5), y(k-1), y(k-2));
    end
end
```



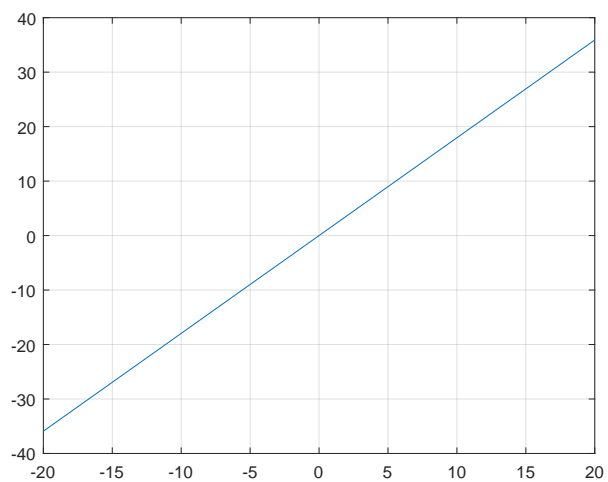
Rys. 3. odpowiedź skokowa w torze zakłócenie-wyjście

Właściwości dynamiczne dla toru zakłócenie-wyjście nie są liniowe. Wynika to stąd, że dla danej zmiany sterowania nie obserwujemy proporcjonalnych zmian na wyjściu. Przykładem może być skok sterowania z 0 na -1 , gdzie na wyjściu pojawia się wartość około 2,1. Nieliniowość jest również uwydatniona przez widoczne nieliniowe efekty (np. nieminimalnofazowość) dla ujemnych skoków sterowania

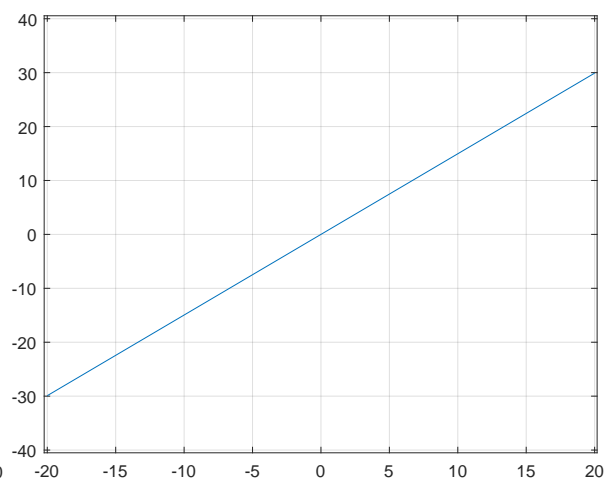
1.2.3. Wzmocnienie statyczne obu torów procesu

Właściwości statyczne dla obu torów sygnałów sterujących są liniowe. Wzmocnienia wyznaczono ze wzoru:

$$K_{stat} = \frac{y^{\infty} - y^0}{\Delta u}$$

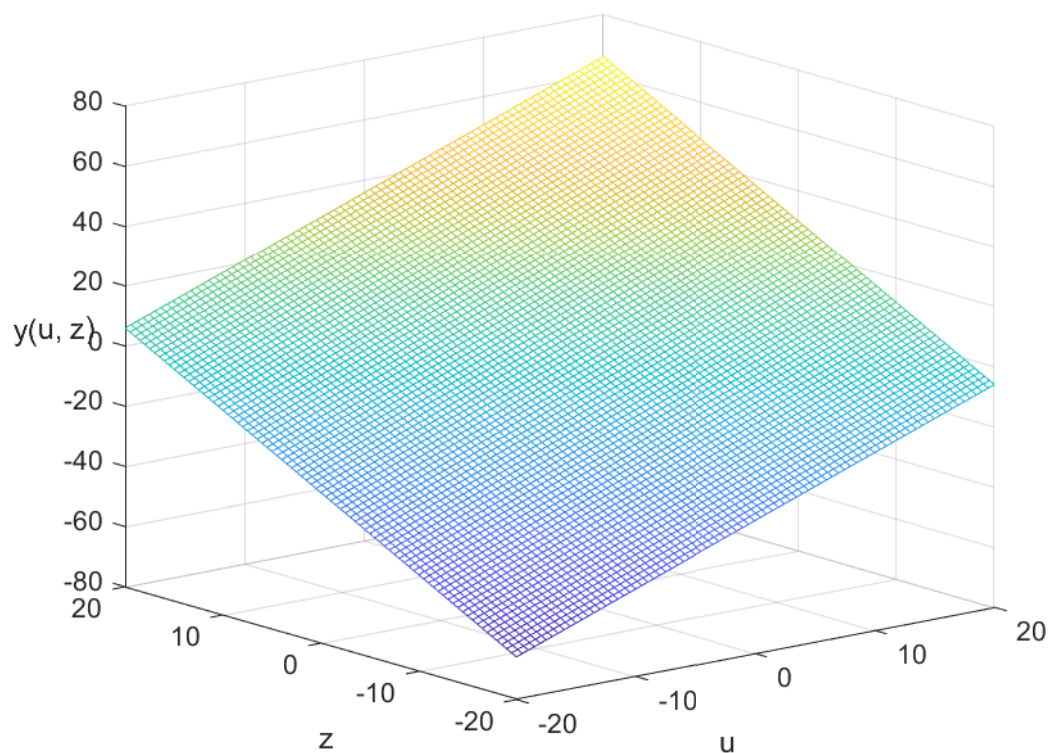


Rys. 4. wzmocnienie toru wejście-wyjście
 $K_{stat} = 1,7955$



Rys. 5. wzmocnienie toru zakłócenie-wyjście
 $K_{stat} = 1,4955$

1.2.4. Charakterystyka statyczna procesu $y(u, z)$



Rys. 6. wykres charakterystyki statycznej procesu

1.3. Odpowiedzi skokowe algorytmu DMC

Wyznaczenie odpowiedzi skokowej dla sterowania:

```
% (inicjalizacja z pierwszego zadania)
u(start:n) = 1;
z(start:n) = 0;

for k = start:n
    y(k) = symulacja_obiektu12y_p2(u(k-7), u(k-8), ...
        z(k-4), z(k-5), y(k-1), y(k-2));
end

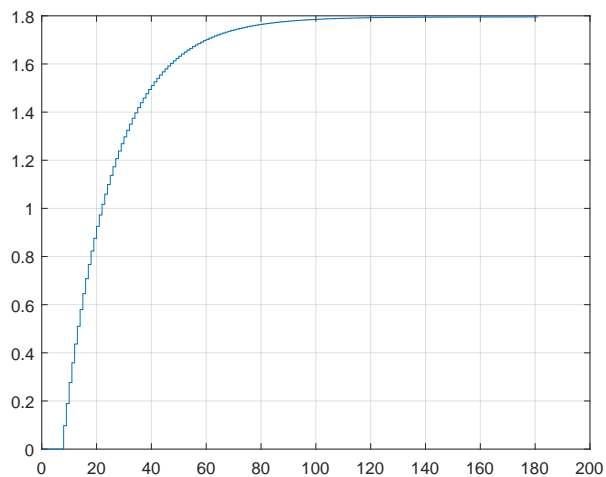
D = 200;
s = y(start:D+start)';
```

Wyznaczenie odpowiedzi skokowej dla zakłóceń:

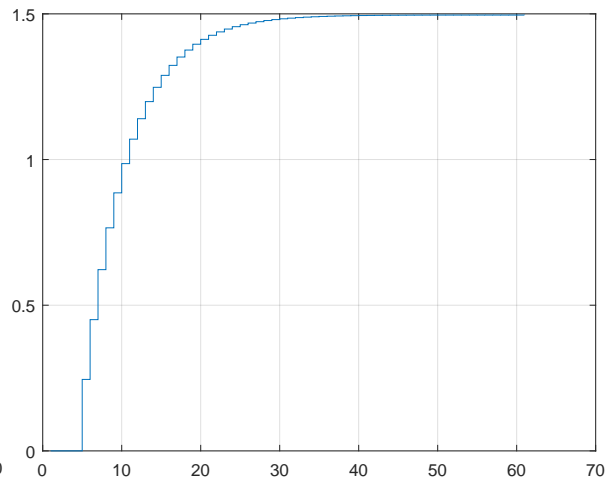
```
% (inicjalizacja z pierwszego zadania)
u(start:n) = 0;
z(start:n) = 1;

for k = start:n
    y(k) = symulacja_obiektu12y_p2(u(k-7), u(k-8), ...
        z(k-4), z(k-5), y(k-1), y(k-2));
end

D = 70;
sz = y(start:D+start)';
```



Rys. 7. odpowiedź skokowa dla wejścia sterowanego



Rys. 8. odpowiedź skokowa dla zakłóceń

1.4. Symulacja regulacji DMC

1.4.1. Implementacja

```
% (opis funkcji 'dmc_offline' oraz 'regulacja_dmc'
% znajduje się w kolejnych listingach)

ust = 150;

u(1:n) = 0;
y(1:n) = 0;
z(1:n) = 0;
yzad(start:n) = 1;
% parametry - wartości będą opisane w następnej sekcji
D = d;
Dz = dz;
N = n;
Nu = nu;
lambda = 1;

[Mp, Mzp, K] = dmc_offline(s, sz, D, Dz, N, Nu, lambda);
k = start;

while k < n
    k = k + 1;
    y(k) = symulacja_obiektu12y_p2(u(k-7), u(k-8), ...
        z(k-4), z(k-5), y(k-1), y(k-2));

    if pomiar_zaklocen == 0
        zpom = zeros(1,n);
    else
        zpom = z;
    end

    du = regulacjaDMC(k, u, y, yzad, z, zpom, D, Dz, N, K, Mp, Mzp);

    u(k) = u(k-1) + du;
end
```



```

function [Mp, Mzp, K] = dmc_offline(s, sz, D, Dz, N, Nu, lambda)

% macierz M
M = zeros(N, Nu);
for c = 1:Nu % kolumny
    for r = 1:N % wiersze

        if (r-c+1) <= 0
            M(r, c) = 0;
        elseif (r-c+1) > D
            M(r, c) = s(D);
        else
            M(r, c) = s(r-c+1);
        end
    end
end

% macierz Mp
Mp = zeros(N, D-1);
for c = 1:D-1 % kolumny
    for r = 1:N % wiersze

        if r+c > D
            scr = s(D);
        else
            scr = s(c+r);
        end

        if c > D
            sc = s(D);
        else
            sc = s(c);
        end

        Mp(r, c) = scr - sc;
    end
end

% Macierz Mzp
Mzp2 = zeros(N, Dz-1);
Mzp_kol1 = ones(N, 1);
for c = 1:Dz-1 % kolumny
    for r = 1:N % wiersze

        if r+c > Dz
            szcr = sz(Dz);
        else
            szcr = sz(c+r);
        end

        if c > Dz

```

```

        szc = sz(Dz);
    else
        szc = sz(c);
    end

    Mzp2(r, c) = szcr - szc;

    if r > Dz
        Mzp_kol1(r) = sz(Dz);
    else
        Mzp_kol1(r) = sz(r);
    end
end
end

Mzp = [Mzp_kol1, Mzp2];

% wektor K
I = eye(Nu);
K = ((M'*M + lambda*I)^(-1))*M';

end

```

```

function du = regulacjaDMC(k, u, ypom, yzad, z, zpom, ...
                        D, Dz, N, K, Mp, Mzp)

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

    ster = 0;
    zakl = 0;

    Ke = 0;
    for i = 1 : N
        Ke = Ke + K(1, i);
    end

    for ju = 1:D-1

        ku = K(1,:) * Mp(:,ju);

        if k-ju < 1
            du_kmj = 0;
        elseif k-ju == 1
            du_kmj = u(k-ju);
        else
            du_kmj = u(k-ju) - u(k-ju-1);
        end

        ster = ster + ku*du_kmj;
    end
end

```

```

for jz = 0:Dz-1

    kz = K(1,:) * Mzp(:, jz+1);

    if k-jz < 1
        dz_kmj = 0;
    elseif k-jz == 1
        dz_kmj = zpom(k-jz);
    else
        dz_kmj = zpom(k-jz) - zpom(k-jz-1);
    end

    zakl = zakl + kz*dz_kmj;
end

du = Ke * e(k) - ster - zakl;
end

```

1.4.2. Dobór parametrów i ocena jakości

Do uzyskania optymalnych nastaw regulatora użyliśmy implementacji algorytmu genetycznego z Matlabowego pakietu Global Optimization Toolbox. Horyzont dynamiki zakłóceń został ustawiony na zero, ponieważ nie uwzględniamy na razie toru zakłóceń.

```

objective = @(x) DMC (x(1), x(2), x(3), x(4));

x_initial = [200,, 200, 80, 10];
lb = [10, 10, 1, 0];
ub = [240, 240, 240, 100];

options = optimoptions('ga', 'MaxGenerations', 100, ...
    'PopulationSize', 50, 'Display','iter');

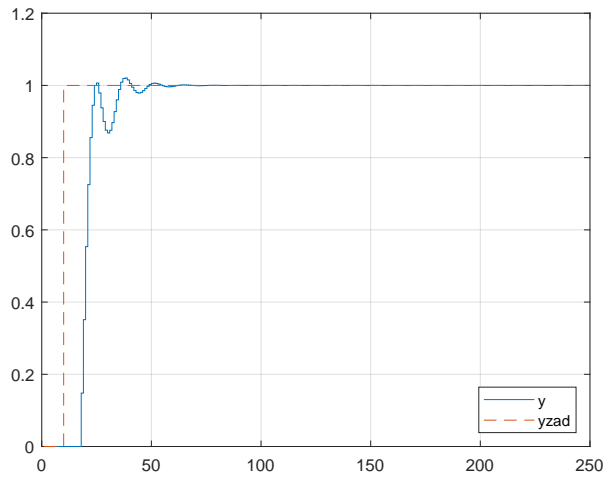
[x_optimal, fval] = ga(objective, 4, [], [], [], [], ...
    lb, ub, 1:4, options);

```

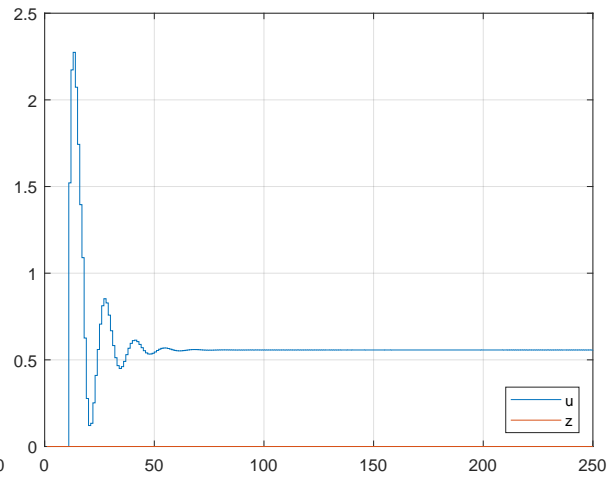
Otrzymane nastawy to:

$$D = 213, Dz = 0, N = 27, Nu = 10, \lambda = 0,24$$

Błąd dla takich parametrów wyniósł 9,5285.



Rys. 9. wyjście obiektu

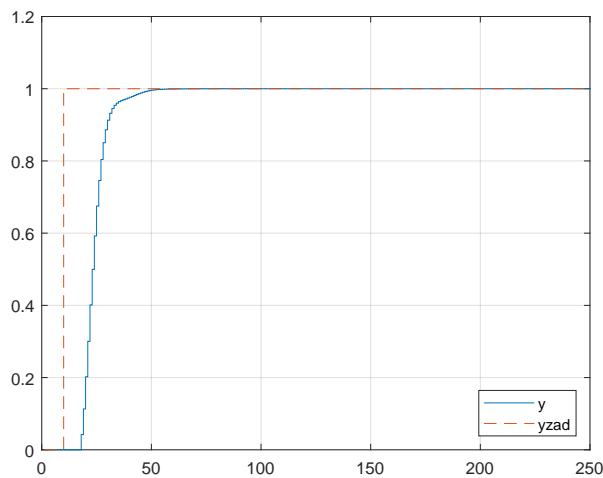


Rys. 10. sterowanie regulatora DMC

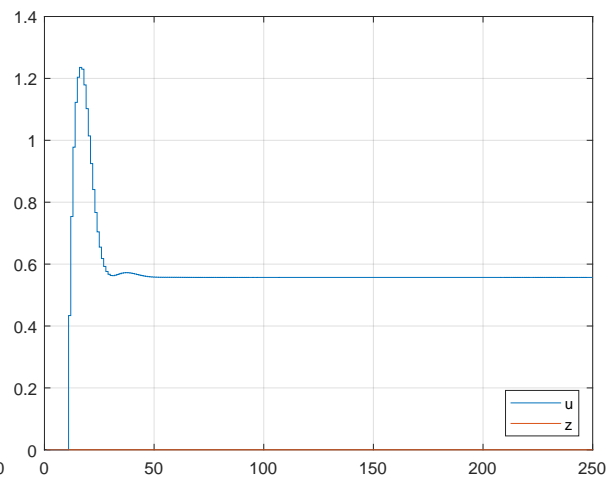
Powyższe przebiegi nie są satysfakcjonujące - szybko zmieniają swoje wartości, oscylują oraz widoczne jest wysokie przesterowanie w przypadku sygnału sterującego. Dlatego potrzebowaliśmy zmienić nastawy. Zaczęliśmy od zwiększenia wartości lambdy. Okazało się to wystarczająco satysfakcjonujące rozwiązanie, więc wartości horyzontów pozostały niezmienione. Pomimo, że błąd wzrósł, to jakościowo przebiegi znacznie się polepszyły.

$$D = 213, Dz = 0, N = 27, Nu = 10, \text{lambda} = 4$$

Błąd dla takich parametrów wyniósł 11,8731.



Rys. 11. wyjście obiektu

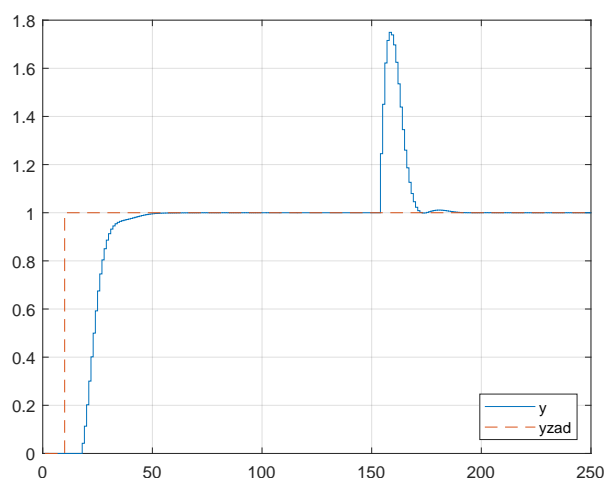


Rys. 12. sterowanie regulatora DMC

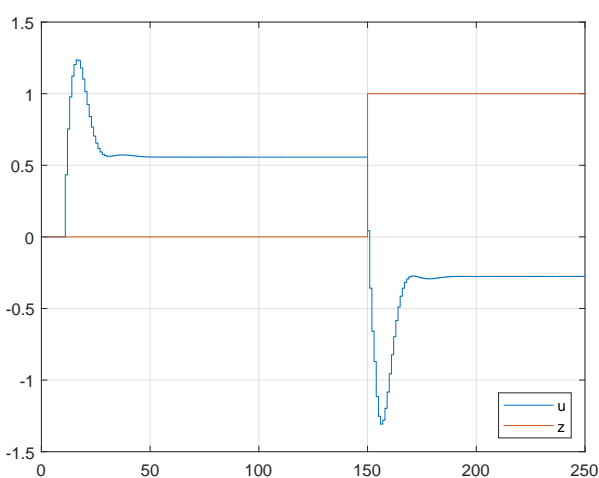
W dalszej części będziemy korzystać z nastaw z większą wartością lambdy.

1.5. Regulacja DMC z zakłóceniem

Dla horyzontu dynamiki zakłóceń przyjęliśmy wartość 65 - jest to moment, gdy odpowiedź skokowa toru zakłóceń jest już stabilna (widać to bardzo dobrze na rys. 8)

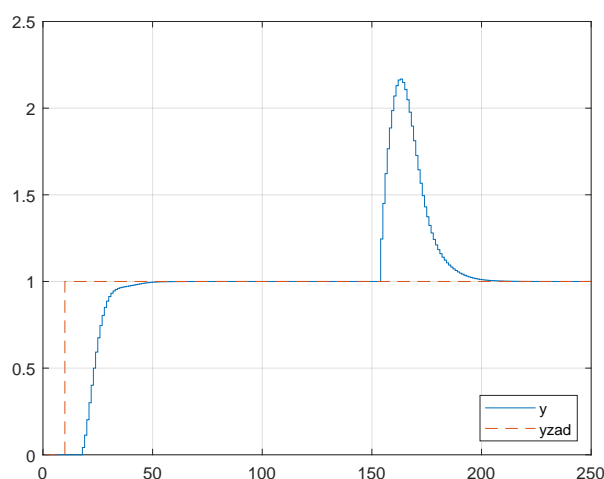


Rys. 13. wyjście obiektu

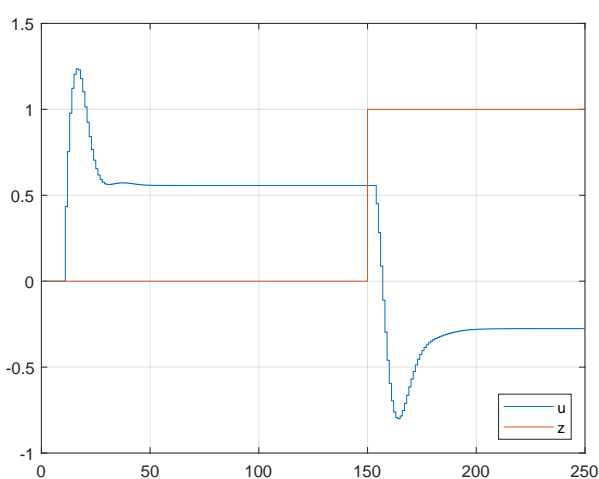


Rys. 14. sterowanie regulatora DMC

Wyniki eksperymentu dla regulatora DMC z pomiarem zakłóceń



Rys. 15. wyjście obiektu



Rys. 16. sterowanie regulatora DMC

Wyniki eksperymentu dla regulatora DMC bez pomiaru zakłóceń.

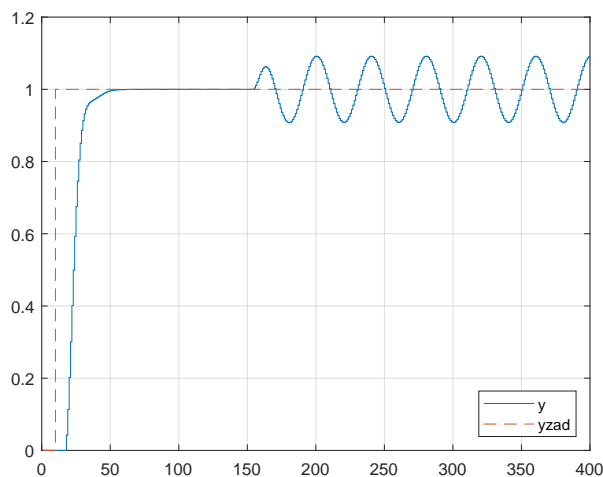
Widoczne jest, że pomiar zakłóceń pozwala osiągnąć ponowne ustabilizowanie się obiektu w krótszym czasie niż brak tego pomiaru. Dodatkowo wpływa to na mniejszy pik wartości wyjścia. Jednak sam pomiar zakłóceń nie jest konieczny, aby obiekt był stabilny po odpowiednio długim czasie.

1.6. Regulacja przy sinusoidalnym zakłóceniu

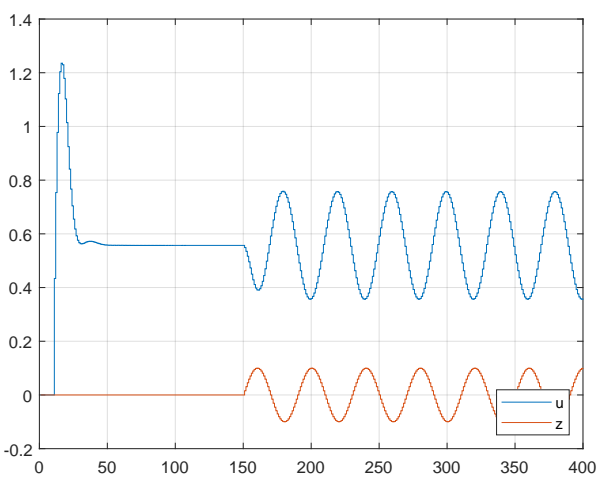
Do obiektu wprowadzono sinusoidalne zakłócenia o amplitudzie A równej 0,1 oraz okresie T wynoszącym 40.

```
A = 0.1; T = 40;  
if k>=ust  
    z(k) = A*sin(2*pi * k / T + pi/2);  
end
```

Uwzględniony pomiar zakłóceń:

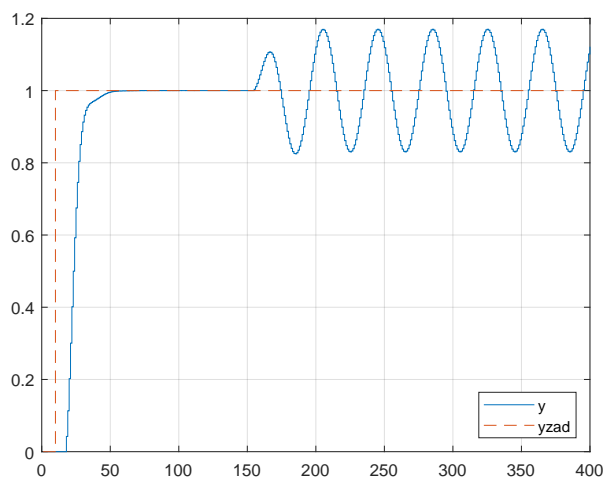


Rys. 17. wyjście obiektu

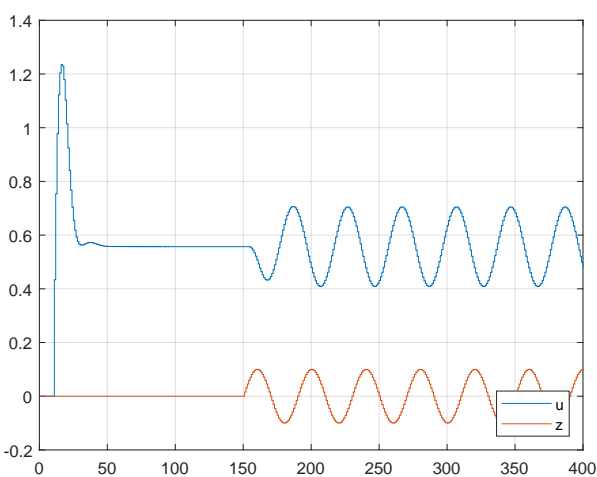


Rys. 18. sterowanie regulatora DMC

Brak pomiaru zakłóceń:



Rys. 19. wyjście obiektu



Rys. 20. sterowanie regulatora DMC

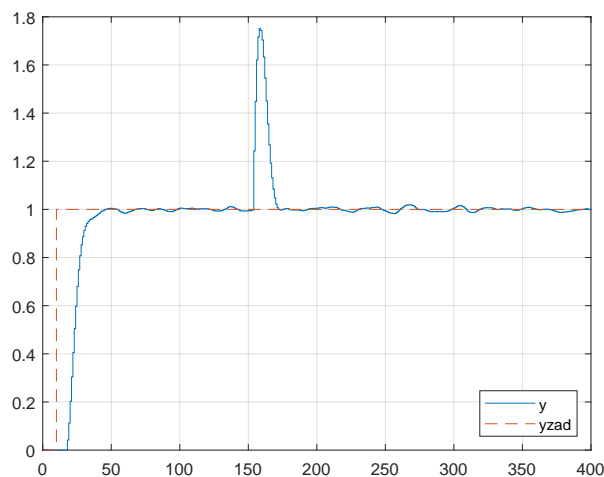
Podobnie jak poprzednio pomiar zakłóceń zmniejsza ich wpływ na sygnał wyjściowy, ale nie niweluje ich zupełnie. Jest to spowodowane tym, że znamy odpowiedź układu na skokową zmianę wejść niesterowanych, a nie znamy odpowiedzi na sinusoidalny przebieg tego sygnału.

1.7. Regulacja przy szumie pomiarowym

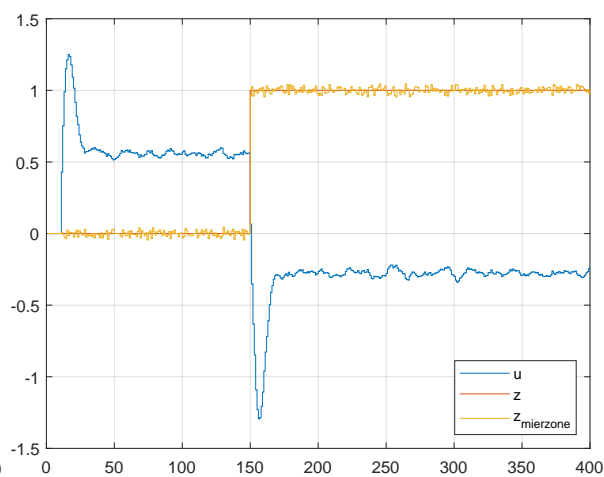
Do generacji szumu użyto bardzo prostą metodę generacji liczb pseudolosowych, przyjmujących wartości od $-0,05$ do $0,05$ skwantowanych co $0,005$.

```
A = 0.005;  
zpom(k) = z(k) + A*(randi(10) - randi(10));
```

Uwzględniony pomiar zakłóceń:

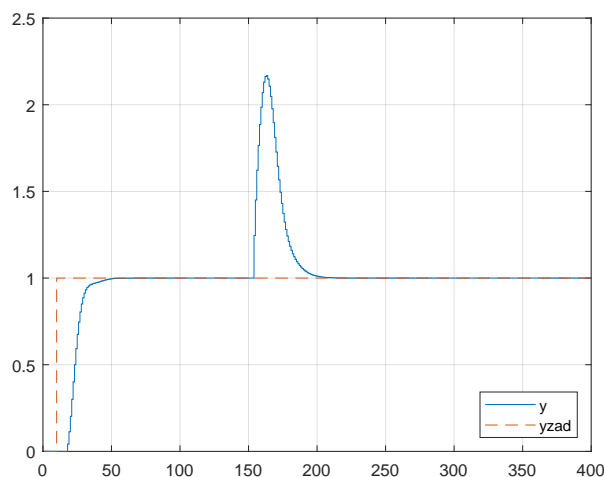


Rys. 21. wyjście obiektu

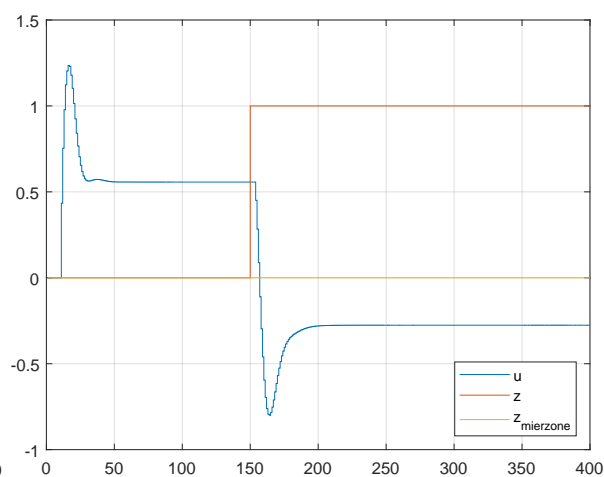


Rys. 22. sterowanie regulatora DMC

Brak pomiaru zakłóceń:



Rys. 23. wyjście obiektu



Rys. 24. sterowanie regulatora DMC

Regulacja z pomiarem zakłóceń podobnie jak we wcześniejszych przypadkach jest szybsza i generuje mniejsze piki na wyjściu. Widzimy jednak, że pomiar obciążony szumem powoduje, że na wyjściu również się on pojawia.

2. CZĘŚĆ LABORATORYJNA

2.1. Punkt pracy stanowiska

Sterowanie stanowiska laboratoryjnego odbywało się za pomocą MATLABA funkcją `obiekt12(u, z)`, która jest edycją otrzymanego przez nas kodu funkcji `MinimalWorkingExample()`.

Sygnal sterujący wentylatorem W1 ustawiano jako argument funkcji `sendControls()`, której pierwszym argumentem była tablica z ID elementów do ustawienia, a drugim tablica z wartościami sygnałów. ID wentylatora W1 to 1. Do sterowania grzałką i zakłóceniami użyto `sendControlsToG1AndDisturbance(u, z)`, gdzie `u` i `z` przekazywane są jako argumenty głównej funkcji `obiekt12()`.

```
function y = obiekt12(u, z)

    %% synchronising with the control process
    % wait for new batch of measurements to be ready

    addpath('D:\SerialCommunication'); % add a path to the functions
    initSerialControl COM5 % initialise com port
    waitForNewIteration();

    sendControls(1, ... send for these elements
                50); % new corresponding control values

    sendControlsToG1AndDisturbance(u, z); % new corresponding control values

    measurement = readMeasurements(1:1);
    y = measurement

end
```

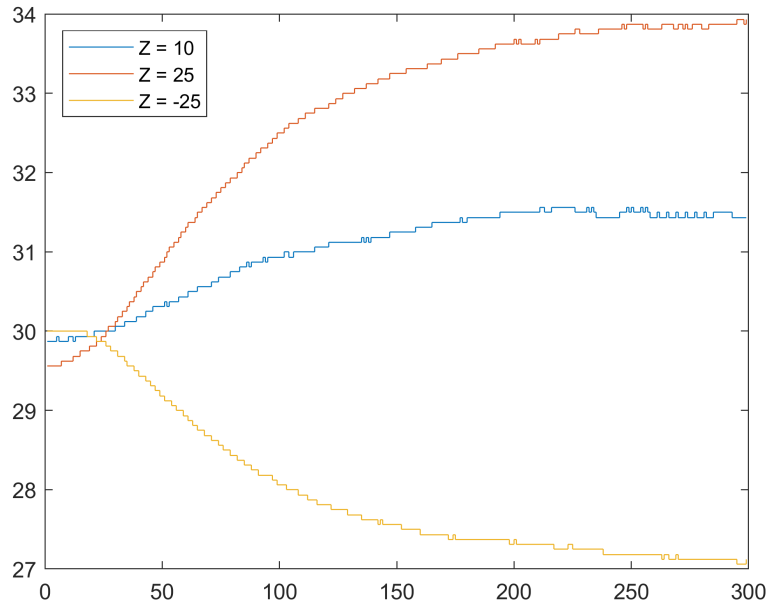
Należało również określić wartość sterowania grzałką oraz zakłóceniami w punktu pracy:

$$G1 = 27, Z = 0$$

Temperatura punktu pracy wyniosła 29,75 °C

2.2. Odpowiedzi skokowe zakłócenie-wyjście

Odpowiedzi skokowe z punktu pracy przeprowadziliśmy dla 3 różnych parametrów zakłócenia.



Rys. 25. Odpowiedzi skokowe dla różnych wartości Z .

Zgodnie z wykresem można określić, że właściwości statyczne obiektu są w przybliżeniu liniowe. Stąd możemy wyznaczyć wzmocnienie statyczne w torze zakłócenie-wyjście. Dla pomiarów rozpoczynających się z punktu pracy: $(u, y) = 27,; 29,75$ (temperatura minimalnie się różniła tj. zmieniał się punkt pracy między kolejnymi pomiarami, lecz temperatura ta była w przedziale od $29.56\text{ }^{\circ}\text{C}$ do $30\text{ }^{\circ}\text{C}$). Z badań otrzymujemy wzmocnienia równe:

$K = 0,167$, dla zmiany parametru zakłócenia $Z = 10$,

$K = 0,1748$, dla zmiany parametru zakłócenia $Z = 25$,

$K = 0,144$, dla zmiany parametru zakłócenia $Z = -20$.

Wartości te są względnie zbliżone, więc można przyjąć, że wzmocnienie określone będzie średnią arytmetyczną tych wartości, stąd: $K \approx 0,162$. Oznacza to, że właściwości statyczne są rzeczywiście w przybliżeniu liniowe.

2.3. Aproksymacja odpowiedzi skokowej

Model zakłóceń oraz model obiektu zostały wyznaczone przy pomocy funkcji *fmincon*. Poniżej przedstawiono implementację funkcji obliczającej optymalne parametry transmitancji modelu i zakłóceń.

```
%% Optymalizacja modelu
objective = @(x) aproksymacja (x(1), x(2), x(3));
x_initial = [90, 100, 0.3];

options = optimset('Display', 'iter');
[x_optimal, fval] = fmincon(objective, x_initial, ...
    [], [], [], [], [], [], options);

%% Funkcja błędu:
function err = aproksymacja(T1, T2, K)
```

```

Td = 17;           % opoznienie
if(T1 ~= T2)      % dla T1 == T2 współczynniki wychodzą NaN
    load("pomiary/lab1_pomiary.mat");
    pomiary = z2_27_35;
    for k = 1:300
        if k > 10
            sp(k-10) = (pomiary(k)-pomiary(10))/8;
        end
    end
end

%      Sz = load("s_z.mat");
%      s_z = Sz.s_z;

% czas trwania procesu
duration = size(sp, 2);    % obiekt
%      duration = size(s_z, 2); % zakłócenia

% liczba eulera
eu = exp(1);

% wyliczenie stałych
alfa1 = eu^(-1/T1);
alfa2 = eu^(-1/T2);
a1 = -alfa1 - alfa2;
a2 = alfa1*alfa2;
b1 = (K/(T1-T2))*(T1*(1-alfa1)-T2*(1-alfa2));
b2 = (K/(T1-T2))*(alfa1*T2*(1-alfa2)-alfa2*T1*(1-alfa1));

% inicjalizacja modelu
ymod(1:duration) = 0;
u(1:19) = 1;
u(20:duration) = 1;
e(1:duration) = 0;

% fmincon czasem wylicza niepoprawne współczynniki
if(~isnan(a1) && ~isnan(a2) && ~isnan(b1) && ~isnan(b2))
    err = 0;
    for k=20:duration
        ymod(k) = b1*u(k-Td-1) + ...
            b2*u(k-Td-2) - ...
            a1*ymod(k-1) - ...
            a2*ymod(k-2);

        e(k) = sp(k) - ymod(k);
%      e(k) = s_z(k) - ymod(k);
        err = err + e(k)^2;
    end
end
end
end
end

```

Po przeprowadzonej optymalizacji, otrzymano optymalne parametry to:

$$T_1 = 10,$$

$$T_2 = 75,$$

$$K = 0,307.$$

Uśredniając opóźnienie ze zmierzonych odpowiedzi skokowych (zarówno przy podnoszeniu jak i obniżaniu temperatury obiektu), otrzymuje się opóźnienie:

$$T_p = 17.$$

A zatem transmitancja modelu będzie wyrażać się wzorem:

$$G(s) = \frac{0,307}{(10s + 1)(75s + 1)} e^{-17s}$$

Natomiast dla zakłóceń:

$$T_1^z = 70,$$

$$T_2^z = 10,$$

$$K^z = 0,15.$$

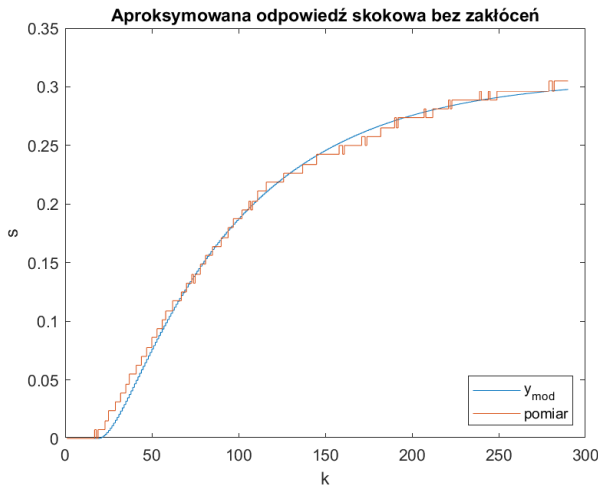
Uśredniając opóźnienie ze zmierzonych odpowiedzi skokowych (zarówno przy podnoszeniu jak i obniżaniu temperatury obiektu), otrzymuje się opóźnienie:

$$T_p = 17.$$

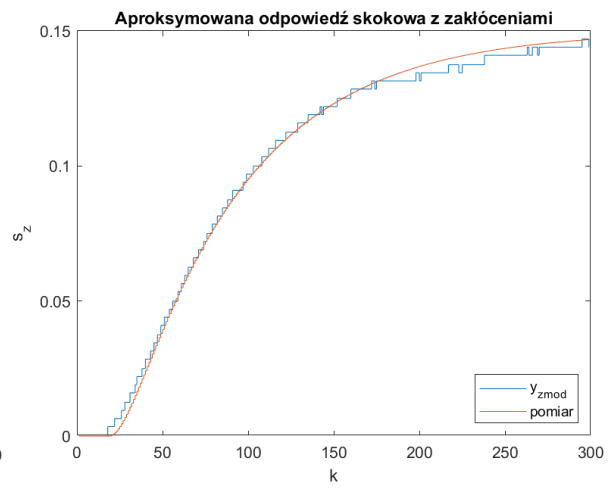
A zatem transmitancja zakłóceń będzie wyrażać się wzorem:

$$Z(s) = \frac{0,15}{(10s + 1)(75s + 1)} e^{-17T_p s}$$

Na rysunkach 26. oraz 27. przedstawiono wizualizację modelowych odpowiedzi skokowych.



Rys. 26.



Rys. 27.

2.4. Regulacja DMC

Na wstępie należy zaznaczyć, że zajęciach laboratoryjnych użyto takiej samej implementacji regulatora DMC, co w projekcie. Mając na uwadze, że symulacje projektowe przebiegły pomyślnie, trudno było przypuszczać, że przeciwnie będzie podczas pracy na stanowisku.

W sprawozdaniu zostaną przedstawione wszystkie pomiary jakie udało się uzyskać oraz podjęta zostanie próba wytłumaczenia, co mogło powodować błędy.

Poniżej przedstawiono zmienioną implementację regulatora DMC pod obiekt grzewczo-chłodzący.

```
% Regulator DMC
[Mp, Mzp, K] = dmc_offline(s, sz, D, Dz, N, Nu, lambda);
k = 100;
while(1) % na rzeczywistym obiekcie
    k = k + 1

    % sprawdzanie warunków dla skoku zakłóceń
    % w chwili ustalenia się obiektu

    if k>D+start+Dz
        z(k:n)=Z2;
        y(k) = obiekt12(u(k-1), Z2);
    elseif k > D+start
        z(k:n) = Z1;
        y(k) = obiekt12(u(k-1), Z1);
    else
        y(k) = obiekt12(u(k-1), 0);
    end

    if(pomiar_zaklocen)
        du = regulacjaDMC(k, u, y, yzad, ...
            z, D, Dz, N, K, Mp, Mzp);
    else
        du = regulacjaDMC(k, u, y, yzad, ...
            zeros(1,n), D, Dz, N, K, Mp, Mzp);
    end

    if du > delta_u_max
        du = delta_u_max;
    elseif du < (-1)*delta_u_max
        du = -delta_u_max;
    end

    u(k) = u(k-1) + du;

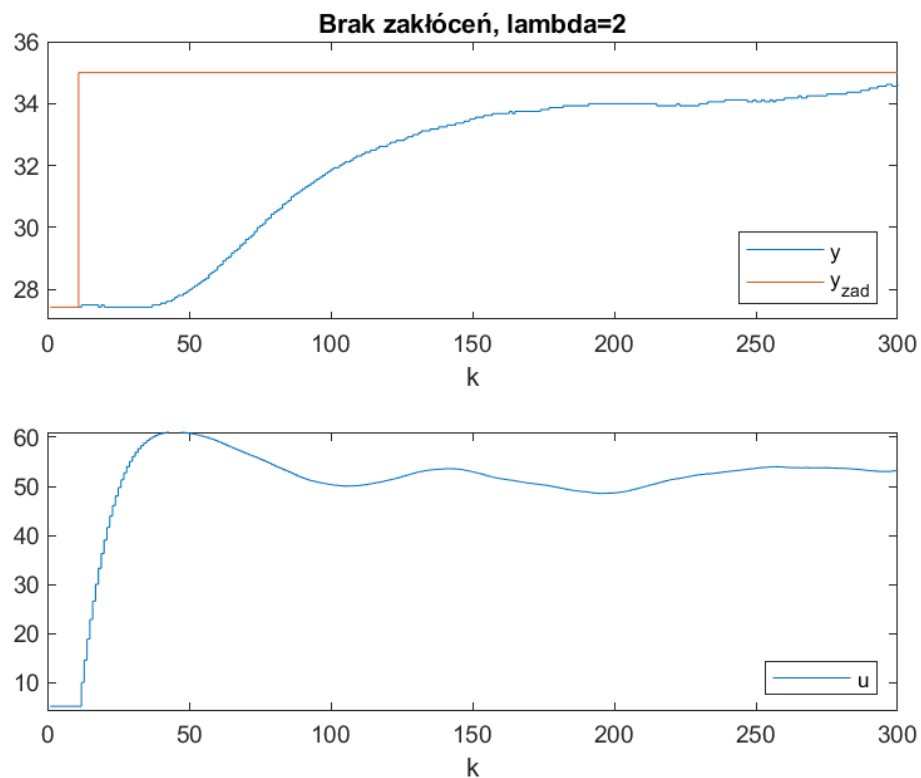
    % uwzględnienie ogarniczeń sterowania
    if(u(k) > 100)
        u(k) = 100;
    elseif(u(k) < 0)
        u(k) = 0;
    end
end
```

2.4.1. Dobór parametrów

Na podstawie rysunków 26. oraz 27. przyjęto, że modele obiektu oraz zakłóceń ustalają się wystarczająco w 290. chwili. W związku z tym horyzont dynamiki, zakłóceń oraz predykcji ustalono na 290. Na początek pomiarów przyjęto parametry:

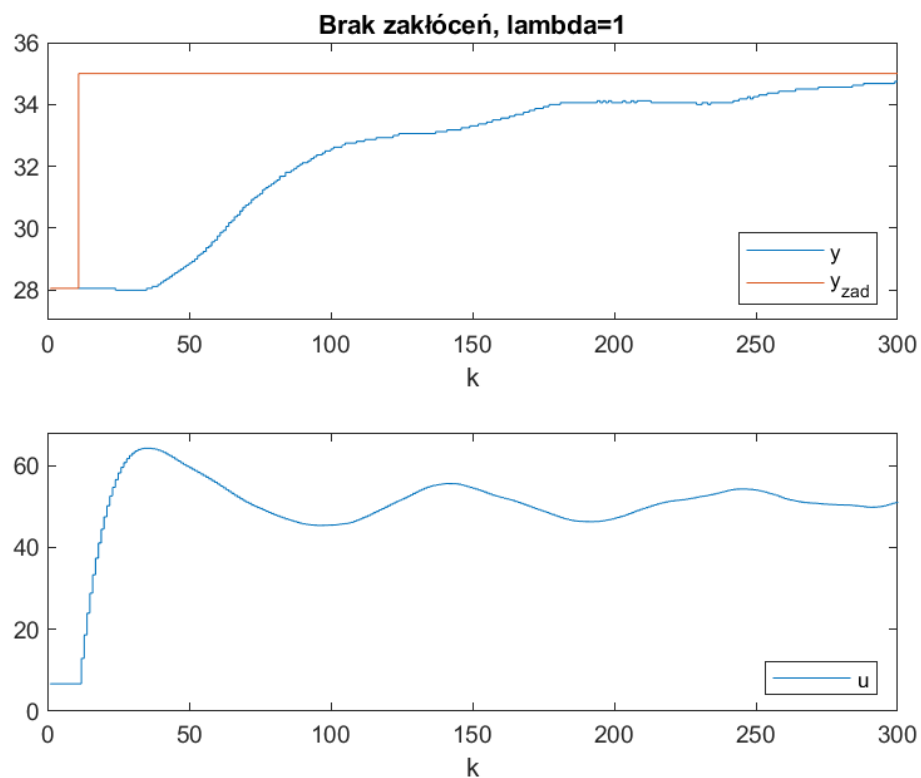
$$D = 290, D^z = 290, N = 290, N_u = 80, \lambda = 2$$

Pierwsze pomiary wykonano przy zerowych zakłóceniach na całym torze sterowania.



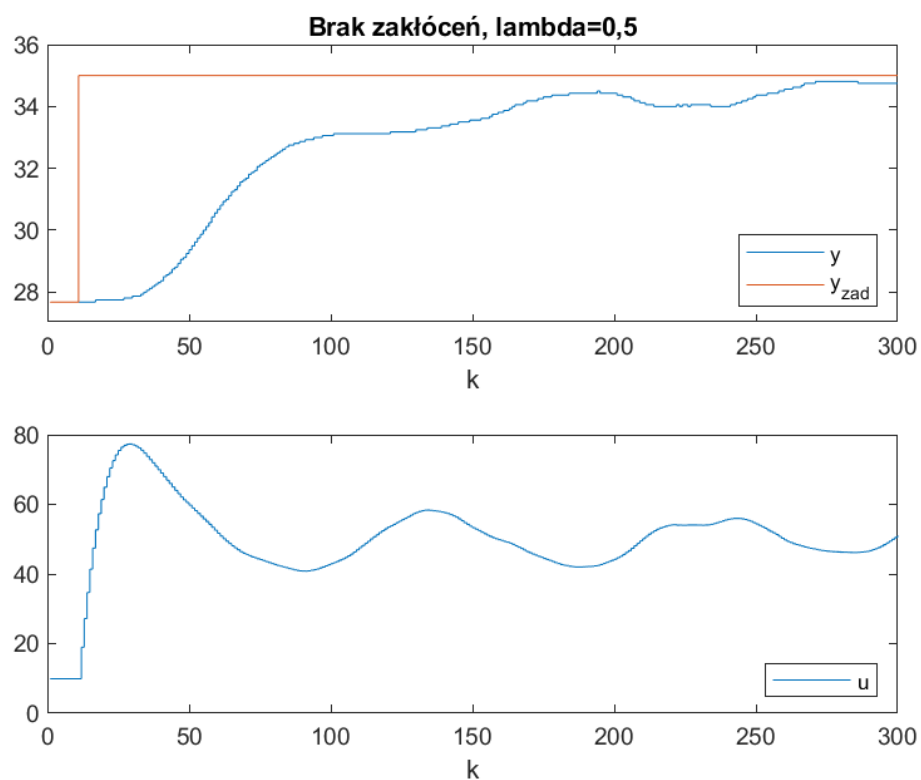
Rys. 28. Wyjście obiektu oraz sterowanie

Odpowiedź obiektu była zbyt wolna, zatem zmniejszono parametr λ dwukrotnie.



Rys. 29. Wyjście obiektu oraz sterowanie

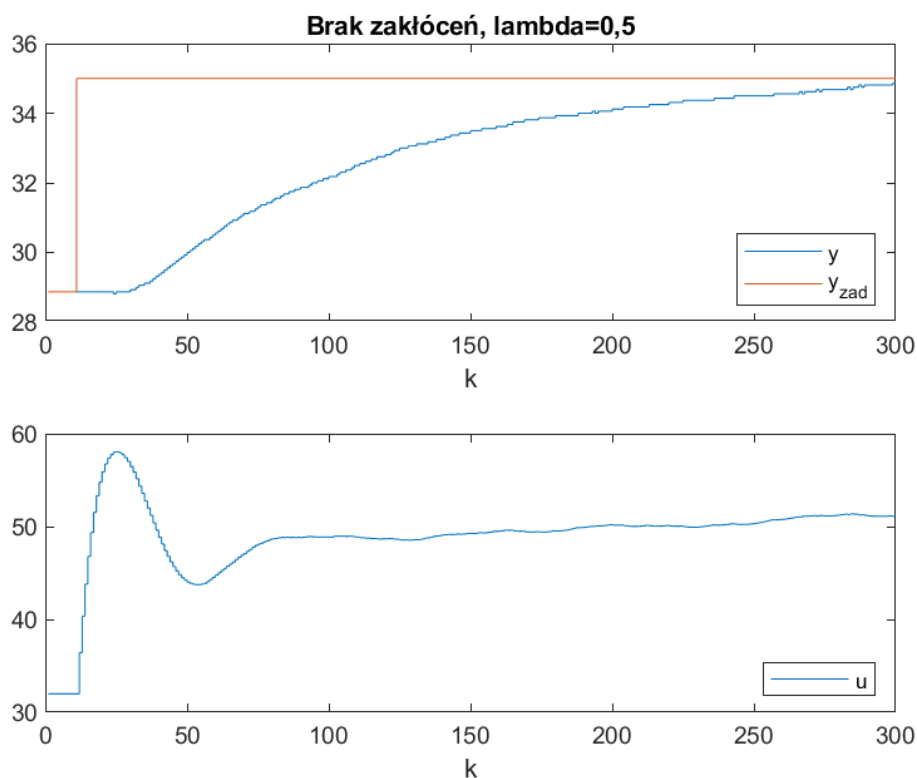
Skok sterowania jest większy, ale odpowiedź obiektu była dalej zbyt wolna, stąd ponownie zmniejszono parametr λ dwukrotnie.



Rys. 30. Wyjście obiektu oraz sterowanie

Zmniejszenie parametru λ poprawiło tempo osiągnięcia wartości zadanej przez obiekt. Co istotne, widać niepokojące oscylacje w sterowaniu oraz w odpowiedzi obiektu. W tym momencie zdano sobie sprawę z tego, że w implementacji regulatora występuje błąd - w złej kolejności wykonywane były polecenia warunkowe i niepoprawnie było odczytywane wyjście oraz zadawane sterowanie. Dodatkowo, algorytm był oparty na zmierzonej odpowiedzi skokowej, a nie aproksymowanej.

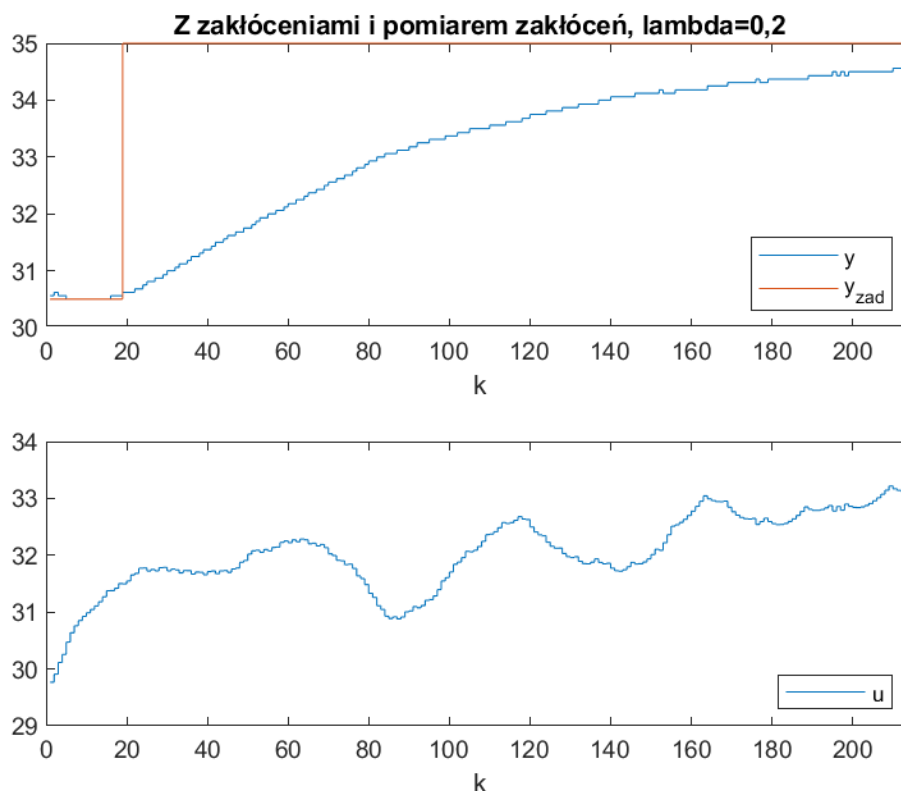
W kodzie naniesiono poprawki i jeszcze raz przeprowadzono pomiar z tymi samymi nastawami.



Rys. 31. Wyjście obiektu oraz sterowanie

Niepokojące oscylacje zostały zlikwidowane, nie mniej jednak prędkość osiągnięcia wartości zadanej przez obiekt jest wciąż zbyt mała, dlatego parametr λ został zmniejszony do 0,2.

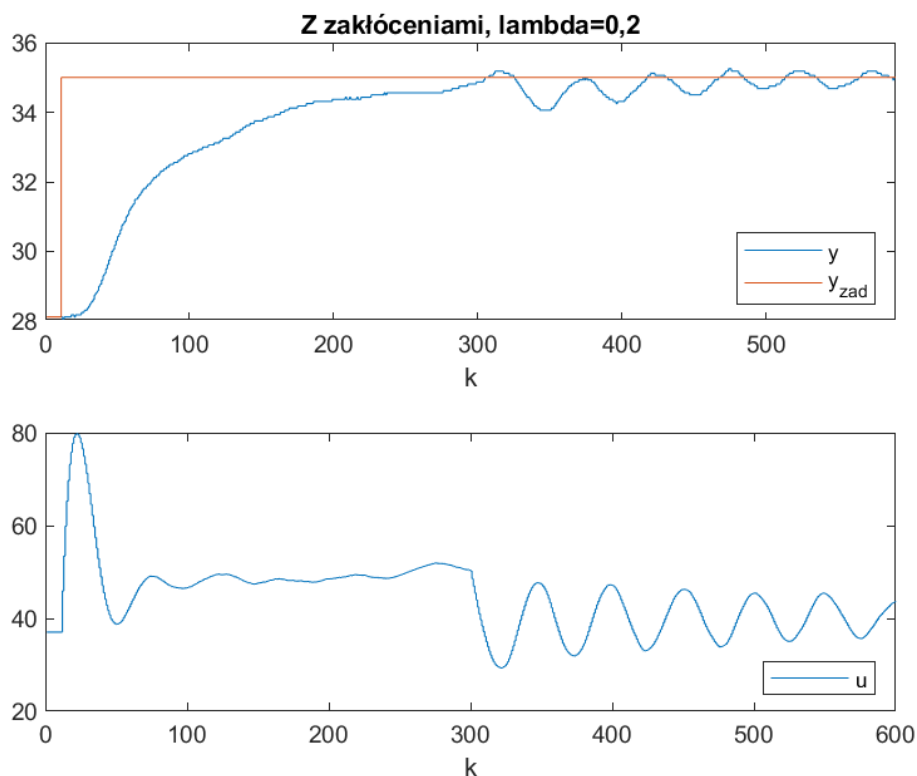
Następnie w pomiarach uwzględniono skok zakłóceń do 30 w chwili ustalenia się wyjścia obiektu na wartości zadanej, a więc w $k = 300$.



Rys. 32. Wyjście obiektu oraz sterowanie

Odczytane z rysunki 32. pomiary były wyjątkowo mało zrozumiałe. Tor sygnału sterującego bardzo wyraźnie oscyluje i wygląda jakby regulował obiektem o zmiennych zakłóceniach. Najbardziej osobliwy jest fakt, że zakłócenia miały pojawić się dopiero w chwili $k = 300$. Pomiary zostały przerwane wcześniej ze względu na dziwne zachowanie sterowania i odpowiedź obiektu nie była została nawet zidentyfikowana po skoku zakłóceń. Z drugiej strony, wyjście obiektu nie oscyluje, za to bardzo powoli się ustala do wartości zadanej.

W tym momencie zauważono kolejny błąd w implementacji kodu związany z odczytywaniem wartości wyjścia obiektu i zadawaniem sterowania. W kodzie naniesiono poprawki i przeprowadzano pomiary bez odczytu zakłóceń (ze względu na małą ilość czasu w laboratorium).



Rys. 33. Wyjście obiektu oraz sterowanie

W przypadku regulacji DMC bez pomiaru zakłóceń, zachowanie obiektu jest bardziej zrozumiałe - obiekt ustala się na wartości zadanej w chwili $k = 300$. W tym momencie na obiekt zaczynają działać zakłócenia, przez co widać, że wartość sterowania wyraźnie spada. Od tego momentu na wyjściu obiektu pojawiają się niegasnące oscylacje, co jest mało pożądanym zachowaniem.

2.4.2. Podsumowanie

Po ostatnim pomiarze widocznym na rysunku 33., termin laboratorium dobiegł końca i nie było więcej czasu na poprawianie kodu i przeprowadzanie pomiarów. Ewidentnie w implementacji algorytmu DMC, z której korzystaliśmy podczas pomiarów występował błąd, który wpłynął znacząco na wyniki pomiarów. Poniżej wypunktowano trudności, z jakimi się napotkano i których nie udało się ich rozwiązać:

- Wyjście obiektu zbyt długo się ustala na wartość zadaną.
- Jakość regulacji z pomiarem zakłóceń jest gorsza, niż przy jego braku (co niezgodne jest z symulacjami projektowymi i założeniami ćwiczenia)*.
- W przypadku regulacji z pomiarem zakłóceń, w torze sterowania występowały oscylacje. Nie było czasu na sprawdzenie działania regulacji po naniesionych poprawkach w kodzie.
- W przypadku regulacji bez pomiaru zakłóceń, w torze sterowania i w torze odpowiedzi obiektu występowały niegasnące oscylacje. Co więcej miały one inny charakter, niż te opisane w poprzednim punkcie - były wyraźnie okresowe.

(*) Pod koniec laboratorium udało się usprawnić implementację algorytmu - ustawiliśmy odpowiednią kolejność warunków, ich kryteria oraz poprawiliśmy macierz M_{zp} . Po tych poprawkach regulator zaczął działać lepiej. Zauważalne wtedy było, że obiekt szybciej się ustalił, jednak nie zdążyliśmy zapisać poprawnie pomiarów.