Wydział Elektroniki i Technik Informacyjnych
Politechnika Warszawska

STP

Sprawozdanie z projektu nr 2, zadanie 16

Paweł Grzankowski

Warszawa, 2024

Spis treści

1. Wstęp

Projekt dotyczy implementacji regulatorów PID, DMC oraz GPC do regulacji obiektu opisanego transmitancją:

$$G(s) = \frac{K_0 e^{-T_0 s}}{(T_1 s + 1)(T_2 s + 1)} \tag{1.1}$$

gdzie $K_0 = 4$, $T_0 = 5$, $T_1 = 1,69$, $T_2 = 5,36$.

Paweł Grzankowski, Warszawa, maj 2024

2. Transmitancja dyskretna

2.1. Wyznaczenie transmitancji dyskretnej

Na podstawie transmitancji ciągłej obiektu, opisanej we wstępie, i wykorzystując program MATLAB wyznaczono transmitancję dyskretną:

$$G(z) = z^{-10} \frac{0.04857z + 0.04267}{z^2 - 1.655z + 0.6776} \tag{2.1}$$

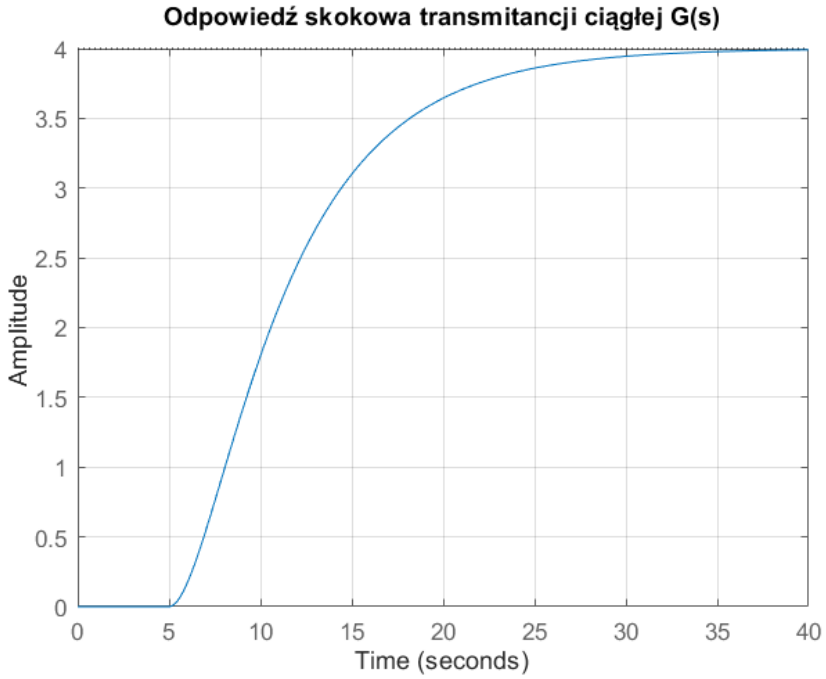
Poniżej przedstawiono kod MATLAB, który posłużył do wyznaczenia transmitancji dyskretnej:

```
% Define the continuous transfer function
K_o = 4;
T_o = 5;
T_1 = 1.69;
T_2 = 5.36;
Tp = 0.5; % Sampling period

% Define the continuous transfer function G(s)
G_s = tf(K_o, [T_1*T_2, T_1 + T_2, 1], 'InputDelay', T_o);

% Discretize using zero-order hold (ZOH)
G_z = c2d(G_s, Tp, 'zoh');
```

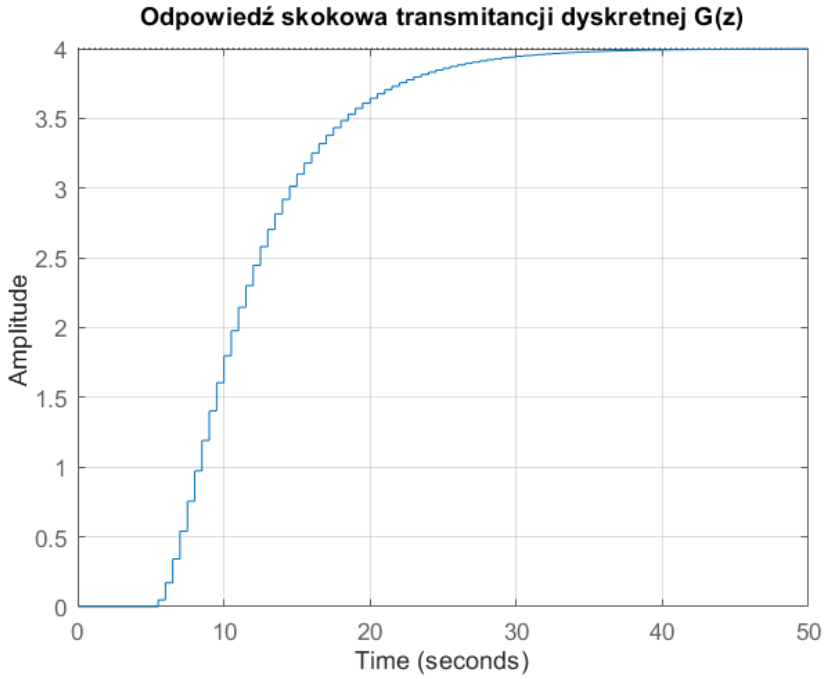
Parametr 'zoh' w funkcji `c2d` oznacza, że dyskretyzacja została przeprowadzona metodą zero-order hold (ekstrapolator zerowego rzędu).

2.2. Porównanie odpowiedzi skokowych

Poniżej przedstawione są odpowiedzi skokowe obiektu dla transmitancji ciągłej i dyskretnej.

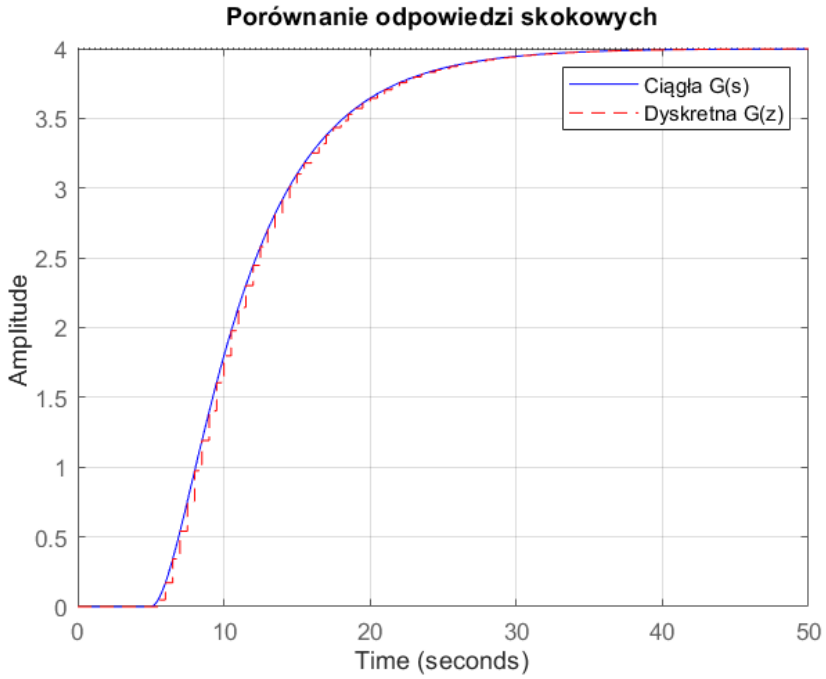


Rys. 2.1. Odpowiedź skokowa obiektu dla transmitancji ciągłej



Rys. 2.2. Odpowiedź skokowa obiektu dla transmitancji dyskretnej

Jak widać na powyższych wykresach, odpowiedzi skokowe obiektu dla transmitancji ciągłej i dyskretnej są bardzo zbliżone. Oznacza to, że dyskretyzacja została przeprowadzona poprawnie. Poniżej widać ich porównanie na jednym wykresie dla lepszego zobrazowania podobieństwa.



Rys. 2.3. Porównanie odpowiedzi skokowych obiektu dla transmitancji ciągłej i dyskretnej

2.3. Wyznaczenie równań różnicowych

Na podstawie transmitancji dyskretnej wyznaczono równania różnicowe opisujące obiekt, które ma następującą postać:

$$y(k) = \sum_{i=1}^{n} b_i y(k-i) + \sum_{i=1}^{m} c_i u(k-i) \tag{2.2}$$

Implementacja w MATLABie wygląda następująco:

```
% Extract coefficients
[c, b] = tfdata(G_z, 'v');

y_fun = @(k, y, u) (-b(2)*y(k-1) - b(3)*y(k-2) + c(2)*u(k-11)
+ c(3)*u(k-12));
```

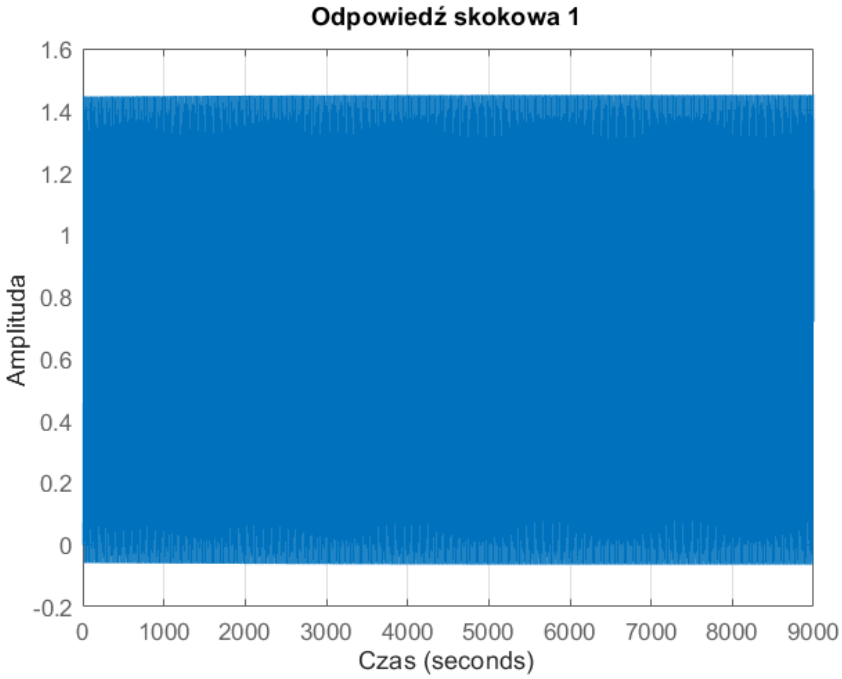
Powyższy kod pozyskuje współczynniki transmitancji dyskretnej i na ich podstawie tworzy równanie różnicowe `y_fun`.

3. Regulator PID

3.1. Metoda Zieglera-Nicholsa

Korzystając z metody Zieglera-Nicholsa wyznaczone zostały parametry regulatora PID. Pierwszym krokiem było znalezienie wzmocnienia krytycznego aby nastąpiły stałe oscylacje. Znalezione zostało eksperymentalnie i wyniosło $K_k = 0.56509$. Wykres oscylacji przedstawiony jest na rysunku 3.1, a kod do jego wyznaczenia jest przedstawiony poniżej.

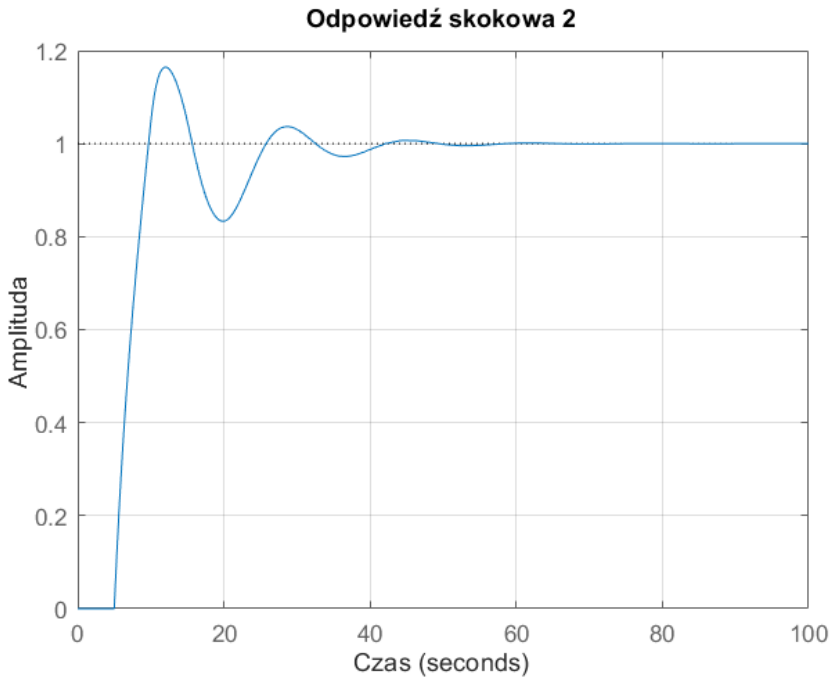
```
K_k = 0.56509; % ultimate gain
T_k = 20; % ultimate period
P = pidstd(K_k, inf, 0);
response = feedback(P*G_s, 1);
```



Rys. 3.1. Oscylacje

Znając wartość wzmocnienia krytycznego można wyznaczyć okres oscylacji, który wyniósł $T_k = 20$ oraz K_r, T_i oraz T_d. Korzystając z tych wartości uzyskano odpowiedź skokową przedstawioną na rysunku 3.2. Kod do jej wyznaczenia przedstawiony jest poniżej.

```
K_p = 0.6 * K_k;
T_i = 0.5 * T_k;
T_d = 0.12 * T_k;
P = pidstd(K_p, T_i, T_d);
response = feedback(P*G_s, 1);
```



Rys. 3.2. Odpowiedź skokowa

Dzięki wyznaczonym wcześniej parametrów można było określić wartości r_0, r_1 oraz r_2, dyskretnego regulatora PID. Wartości parametrów przedstawione są w tabeli 3.1. Kod do ich wyznaczenia przedstawiony jest poniżej.

```
Tp = 0.5;

r_0 = K_p * (1 + Tp/(2*T_i) + T_d/Tp);
r_1 = K_p * (-1 + Tp/(2*T_i) - 2*T_d/Tp);
r_2 = K_p * (T_d/Tp);
```

r_0	1.975
r_1	-3.5855
r_2	1.6275

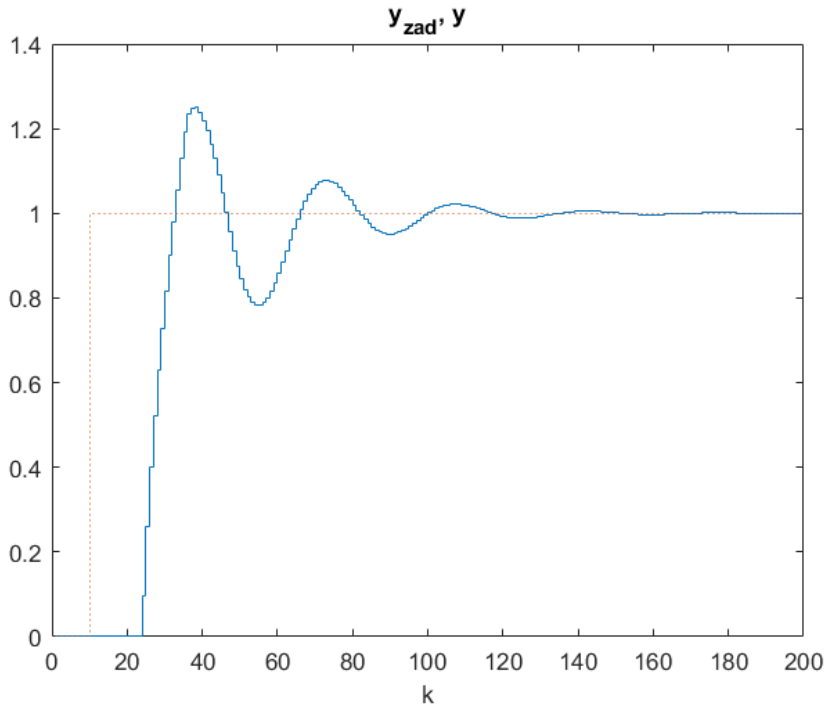
Tab. 3.1. Parametry regulatora PID

3.2. Implementacja regulatora PID

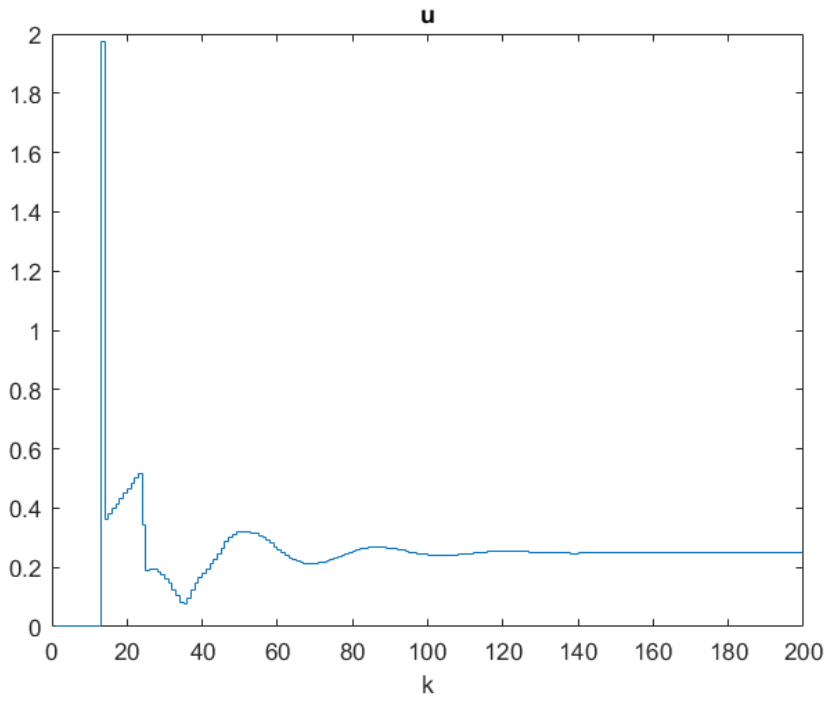
Wyznaczone wcześniej wartości r_0, r_1 oraz r_2 zostały użyte do zaprojektowania regulatora PID w programie MATLAB. Kod przedstawiony jest poniżej.

```
kk=200;

y_zad(1:9)=0; y_zad(10:kk)=1;
y(1:13)=0;
u(1:13)=0;
e(1:13)=0;

for k=13:kk
    y(k) = y_fun(k, y, u);
    e(k) = y_zad(k)-y(k);
    u(k) = r_2*e(k-2)+r_1*e(k-1)+r_0*e(k)+u(k-1);
end
```

Uzyskujemy w ten sposób odpowiedź skokową regulatora oraz wykres sygnału sterującego. Odpowiedzi te przedstawione są na rysunkach 3.3 oraz 3.4.



Rys. 3.3. Odpowiedź skokowa regulatora PID

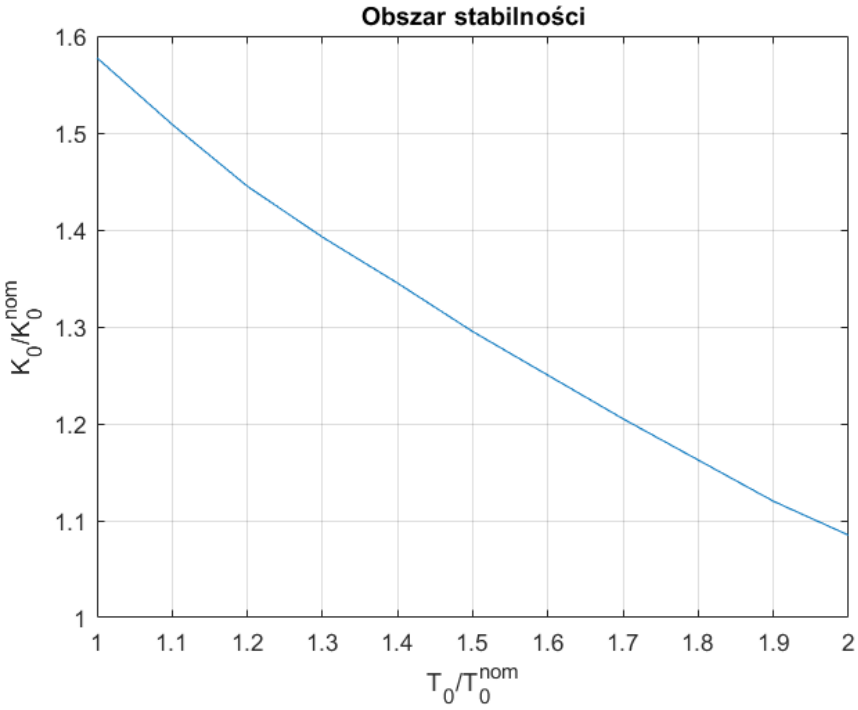


Rys. 3.4. Sygnał sterujący regulatora PID

3.3. Obszar stabilności regulatora PID

Wyznaczenie obszaru stabilności sprowadza się do przeprowadzenia eksperymentów z różnymi wartościami parametru T_0, w celu znalezienia takiej wartości K, która powoduje stabilne

oscylacje. Czynność ta jest powtarzana do momentu znalezienia wszystkich punktów w zadanym przedziale. Wyniki eksperymentów przedstawione są na rysunku 3.5.



Rys. 3.5. Obszar stabilności regulatora PID

Dokładne wartości punktów stabilności przedstawione są w tabeli 5.1.

T_0/T_0^{nom}	K/K_{nom}
1.0	1.5775
1.1	1.5088
1.2	1.4450
1.3	1.3925
1.4	1.3450
1.5	1.2950
1.6	1.2500
1.7	1.2050
1.8	1.1625
1.9	1.1200
2.0	1.0850

Tab. 3.2. Obszar stabilności regulatora PID

4. Regulator DMC

4.1. Implementacja regulator DMC

Na podstawie odpowiedzi skokowej wyznaczonej w punkcie drugim zadania, zaimplementowano regulator DMC w programie MATLAB. Najpierw należy zdefiniować parametry początkowe regulatora, wektory reprezentujące wyjście sygnału sterującego oraz wyjścia oraz odpowiedzi skokowe na podstawie modelu z punktu 2 zadania. Poniższy kod przedstawia implementację tej części zadania.

```
kk = 100;

wy_u = zeros(1, kk);
wy_y = zeros(1, kk);
y_zad = 1;

s = step(G_z);
s(1) = [];
length(s)

D=92;
N=20;
Nu=2;

lambda = 5;

y(1:12) = 0;
u(1:12) = 0;

deltaupk = zeros(1, D-1);
```

Następnie należy zdefiniować macierze M i M^P na podstawie odpowiedzi skokowej. Wartości te są wyznaczane na podstawie wzorów:

$$M = \begin{bmatrix} s_1 & 0 & \cdots & 0 \\ s_2 & s_1 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ s_N & s_{N-1} & \cdots & s_{N-Nu+1} \end{bmatrix} \tag{4.1}$$

$$M^P = \begin{bmatrix} s_2 - s_1 & s_3 - s_2 & \cdots & s_D - s_{D-1} \\ s_3 - s_1 & s_4 - s_2 & \cdots & s_{D+1} - s_{D-1} \\ \vdots & \vdots & \ddots & \vdots \\ s_{N+1} - s_1 & s_{N+2} - s_2 & \cdots & s_{N+D-1} - s_{D-1} \end{bmatrix} \tag{4.2}$$

Znając macierz M jesteśmy w stanie obliczyć macierz K za pomocą wzoru:

$$K = (M^T M + \lambda I)^{-1} M^T \tag{4.3}$$

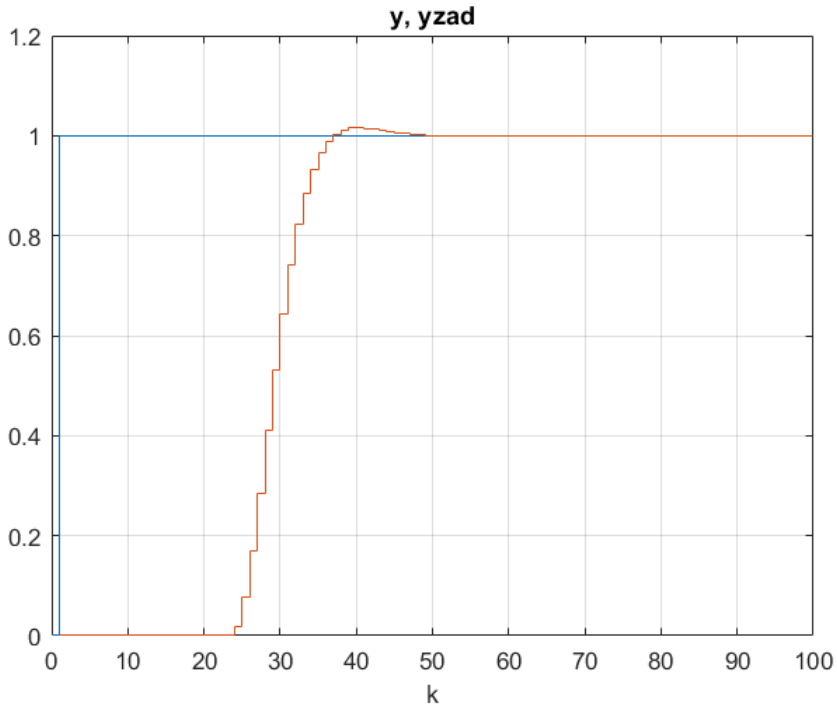
Poniższy kod przedstawia implementację powyższych równań w programie MATLAB.

```
M = zeros(N, Nu);

for i=1:N
    for j=1:Nu
        if i>=j
            M(i, j) = s(i-j+1);
        end
    end
end

MP = zeros(N, D-1);

for i=1:N
    for j=1:D-1
        if i+j<=D
            MP(i, j) = s(i+j)-s(j);
        else
            MP(i, j) = s(D)-s(j);
        end
    end
end

I = eye(Nu);
K = (M' * M + lambda * I)\M';
Ku = K(1, :) * MP;
Ke = sum(K(1, :));
```

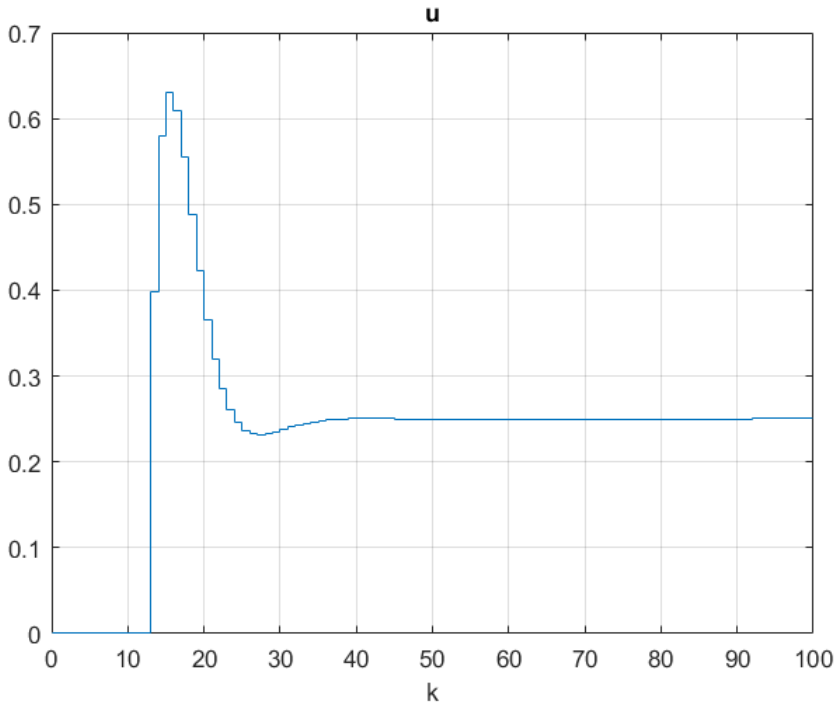
Następnie należy zaimplementować pętlę regulacji. W każdej iteracji obliczane są kolejne wartości sygnału sterującego oraz wyjścia obiektu. Poniższy kod przedstawia implementację regulatora DMC w programie MATLAB.

```
for k=13:kk
    y(k) = y_fun(k, y, u);

    ek = y_zad - y(k);

    deltauk = Ke * ek - Ku * deltaupk';

    for n=D-1:-1:2
        deltaupk(n) = deltaupk(n-1);
    end

    deltaupk(1) = deltauk;

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

    wy_u(k) = u(k);
    wy_y(k) = y(k);
end
```

Korzystając z opisanego powyżej algorytmu regulatora DMC uzyskano odpowiedź skokową obiektu (rys. 4.1) oraz wykres sygnału sterującego (rys. 4.2).



Rys. 4.1. Odpowiedź skokowa obiektu z regulatorem DMC



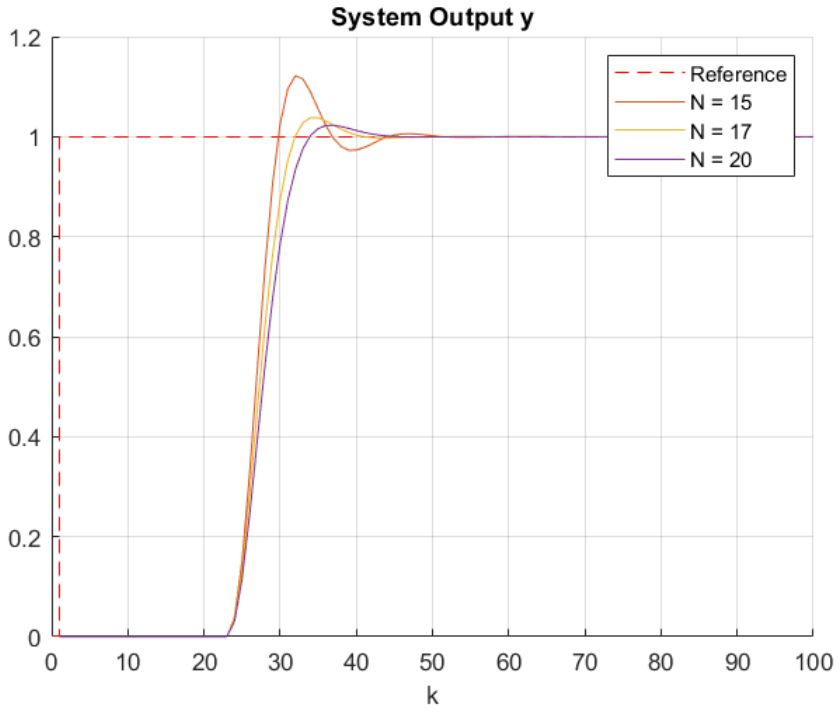
Rys. 4.2. Sygnał sterujący obiektu z regulatorem DMC

4.2. Optymalizacja parametrów regulatora DMC

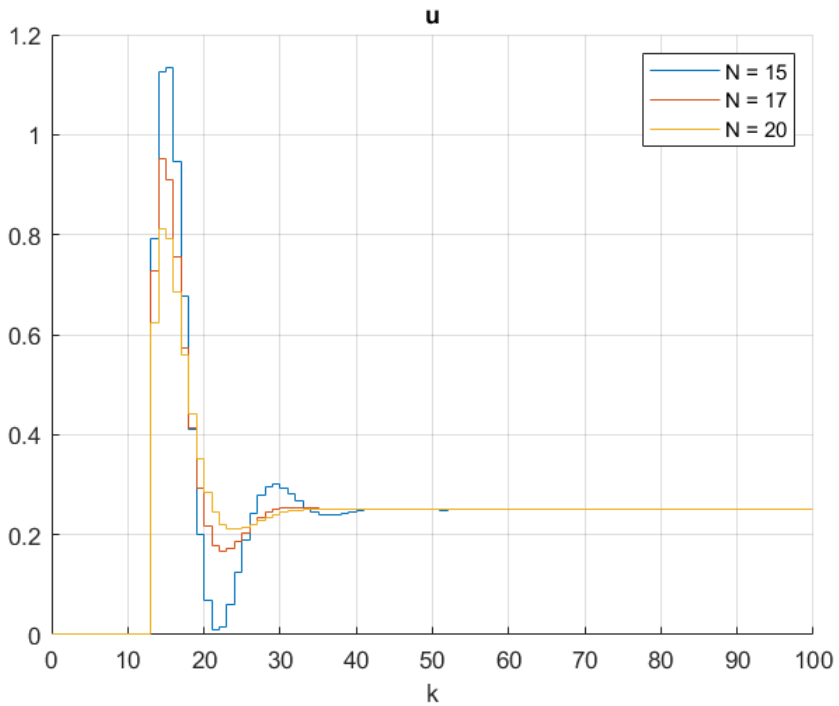
W celu optymalizacji parametrów regulatora DMC dobrano horyzont dynamiki $D = 92$ równy długości odpowiedzi skokowej (wektor s). Następnie przetestowano różne wartości pa-

rametrów N, Nu oraz λ. Poniżej przedstawiono wykresy odpowiedzi skokowych i sygnałów sterujących dla różnych wartości parametrów regulatora DMC.

Horyzont predykcji



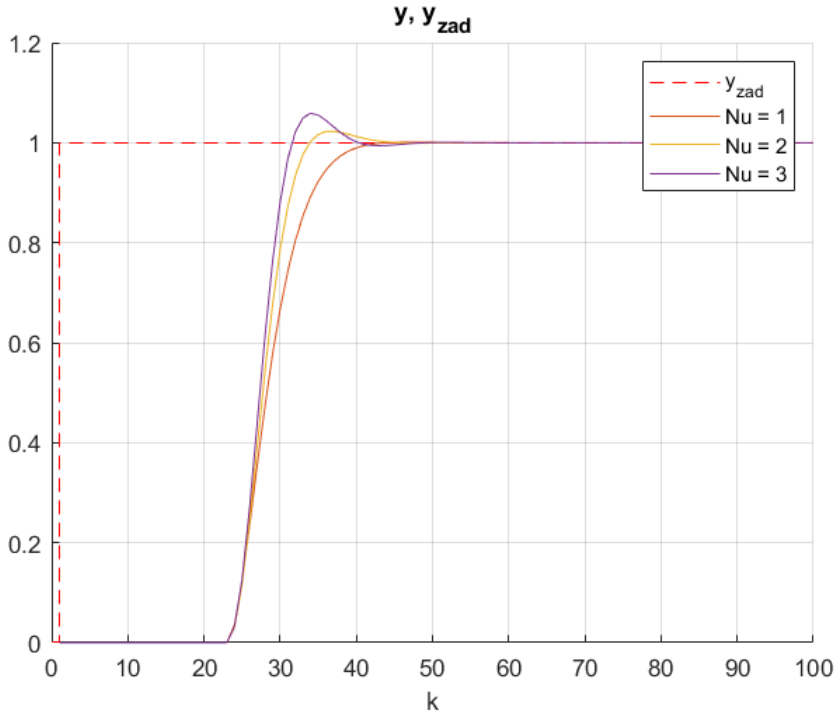
Rys. 4.3. Odpowiedź skokowa obiektu dla różnych wartości parametru N



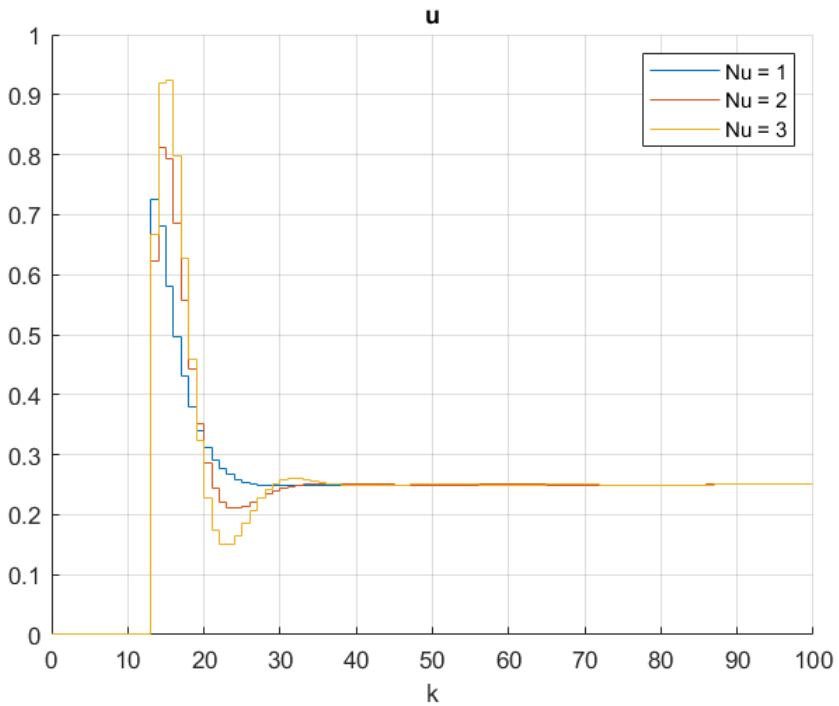
Rys. 4.4. Sygnał sterujący obiektu dla różnych wartości parametru N

Dobranie wartości $N < 15$ prowadziło do oscylacji oraz dłuższego czasu reakcji regulatora. Wartość $N = 20$ pozwala na szybką regulację oraz brak oscylacji, dlatego została przyjęta jako optymalna.

Horyzont sterowania



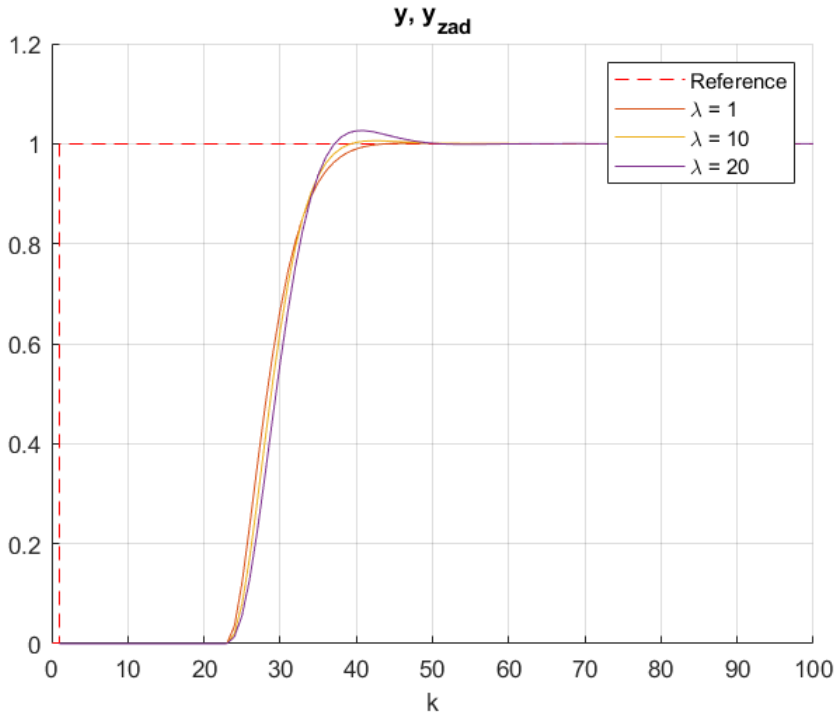
Rys. 4.5. Odpowiedź skokowa obiektu dla różnych wartości parametru Nu



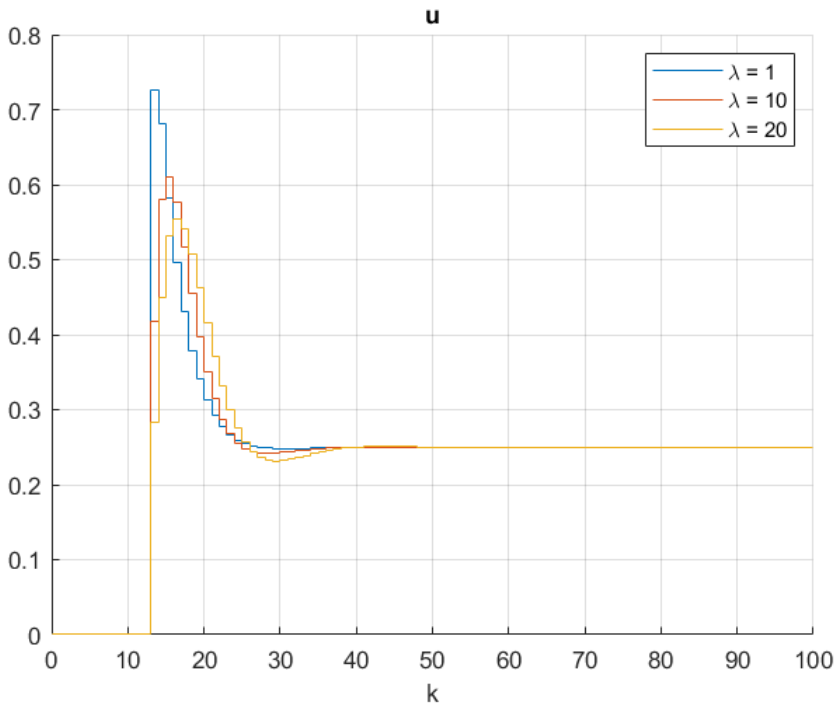
Rys. 4.6. Sygnał sterujący obiektu dla różnych wartości parametru Nu

Odwrotnie do paramteru N, wartość $Nu = 1$ pozwala na szybką regulację, z kolei wartości $Nu > 2$ powodują oscylacje. Na tej podstawie zdecydowano się zastosować wartość $Nu = 1$.

Parametr λ



Rys. 4.7. Odpowiedź skokowa obiektu dla różnych wartości parametru λ



Rys. 4.8. Sygnał sterujący obiektu dla różnych wartości parametru λ

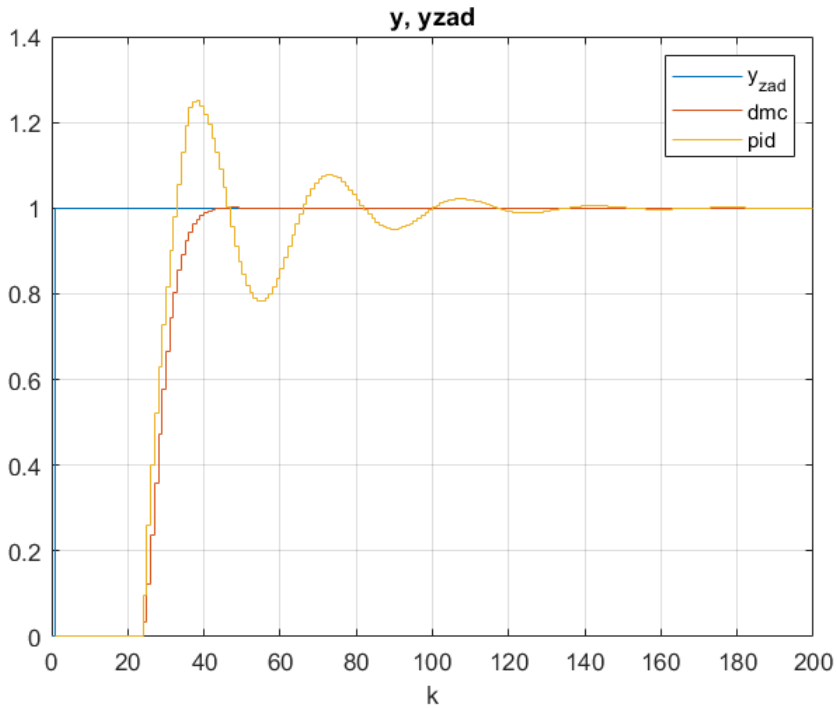
Na podstawie powyższych eksperymentów wybrany został parametr $\lambda = 1$ ze względu na szybkość regulacji oraz brak oscylacji.

Komentarz

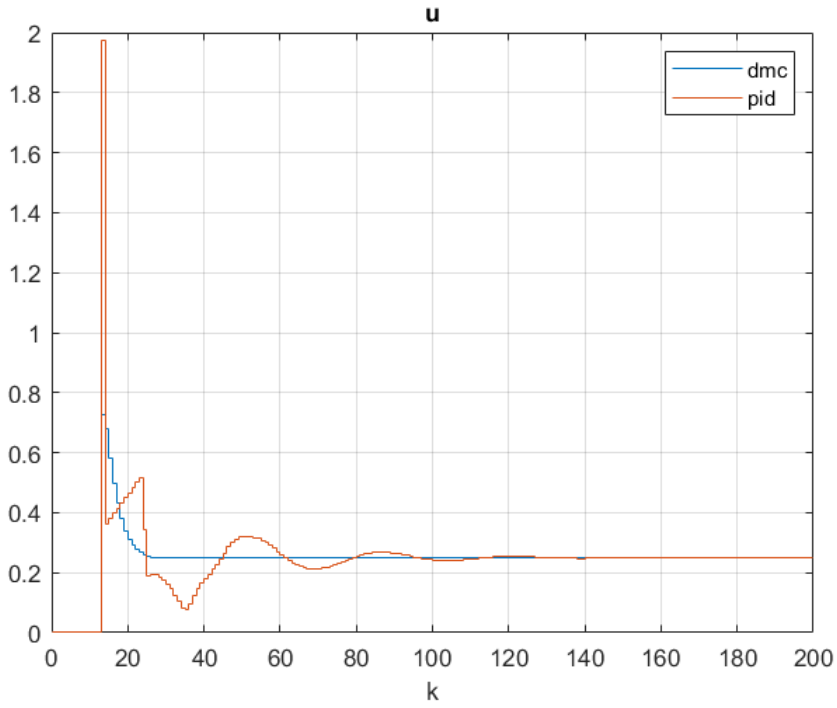
Przy dobieraniu dopowiednich parametrów ważne jest zachowanie balansu pomiędzy szybkością, a stabilnością regulacji. Zwiększając parametr λ zwęksamy stabilność regulacji, jednak tracimy na szybkości, dlatego należy dobrać parametry tak aby regulator spełniał wybrane wymogi.

4.3. Porównanie regulatorów PID i DMC

Po dostrojeniu parametrów regulatora DMC, porównano sygnały sterujące i odpowiedzi skokowe obiektu z regulatorem PID oraz DMC. Poniżej przedstawiono wykresy odpowiedzi skokowych dla obu regulatorów.



Rys. 4.9. Odpowiedź skokowa obiektu z regulatorem PID i DMC



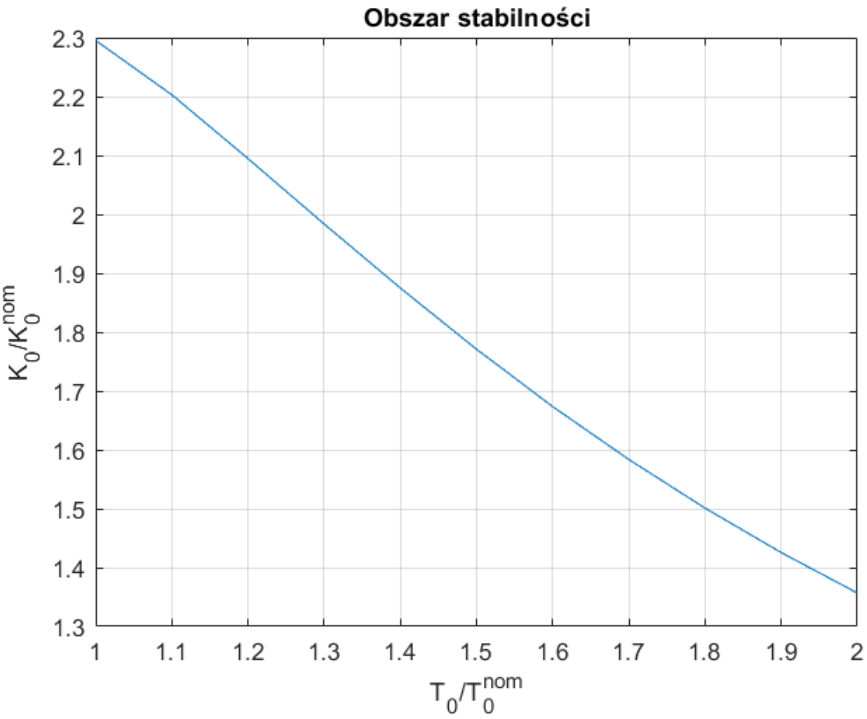
Rys. 4.10. Sygnał sterujący obiektu z regulatorem PID i DMC

Komentarz

Na podstawie odpowiedzi skokowych obiektu z regulatorem PID oraz DMC od razu widać, że regulator DMC działa znacznie lepiej niż regulator PID. Regulator DMC jest bardziej agresywny, co pozwala na szybsze osiągnięcie wartości zadanej. Regulator PID jest bardziej stabilny, ale osiąga wartość zadana znacznie wolniej posiadając zarówno znaczne oscylacje.

4.4. Obszary stabilności regulatora DMC

Obszar stabilności jest wyznaczany analogicznie jak w przypadku regulatora PID. Przedstawiony jest na poniższym wykresie.



Rys. 4.11. Obszar stabilności regulatora DMC

Poniżej przedstawiono tabelę z dokładnymi wartościami punktów na powyższym wykresie.

T_0/T_0^{nom}	K/K_{nom}
1.0	2.2953
1.1	2.2030
1.2	2.0946
1.3	1.9834
1.4	1.8748
1.5	1.7708
1.6	1.6734
1.7	1.5835
1.8	1.5010
1.9	1.4255
2.0	1.3567

Tab. 4.1. Obszar stabilności regulatora PID

5. Regulator GPC

5.1. Implementacja algorytmu GPC

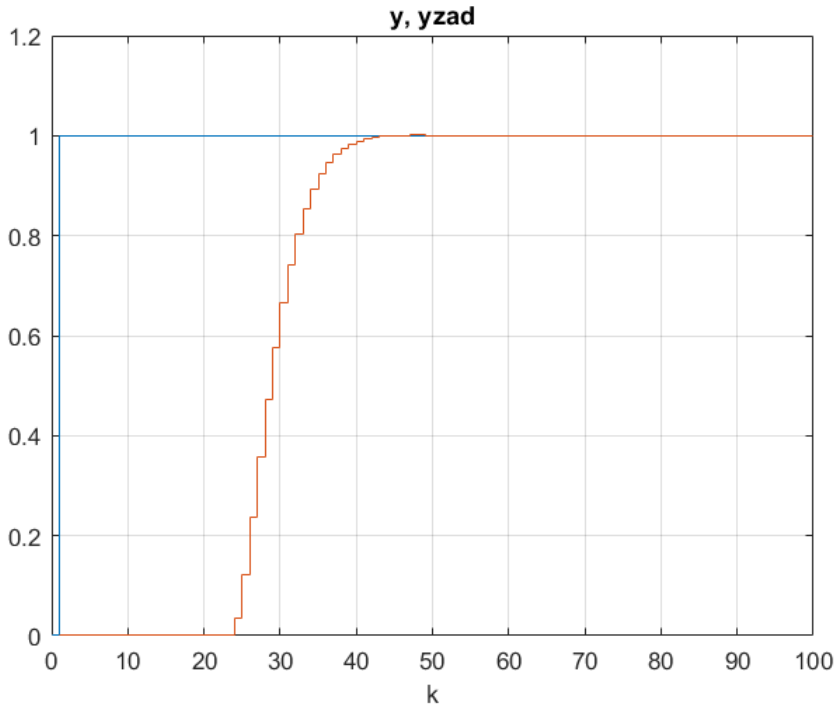
Na podstawie odpowiedzi skokowej wyznaczonej w punkcie drugim zadania, zaimplementowano regulator GPC w programie MATLAB. Jest ona podobna do regulatora DMC, gdzie główna różnica polega na innym wyznaczaniu trajektorii swobodnej.

$$\begin{aligned} y^0(k+p \mid k) = & \sum_{i=1}^{N_{um}(p)} b_i u(k-i+p \mid k) + \sum_{i=N_{um}(p)+1}^{n_B} b_i u(k-i+p) \\ & - \sum_{i=1}^{N_{y0}(p)} a_i y^0(k-i+p \mid k) - \sum_{i=N_{y0}(p)+1}^{n_A} a_i y(k-i+p) + d(k) \end{aligned} \tag{5.1}$$

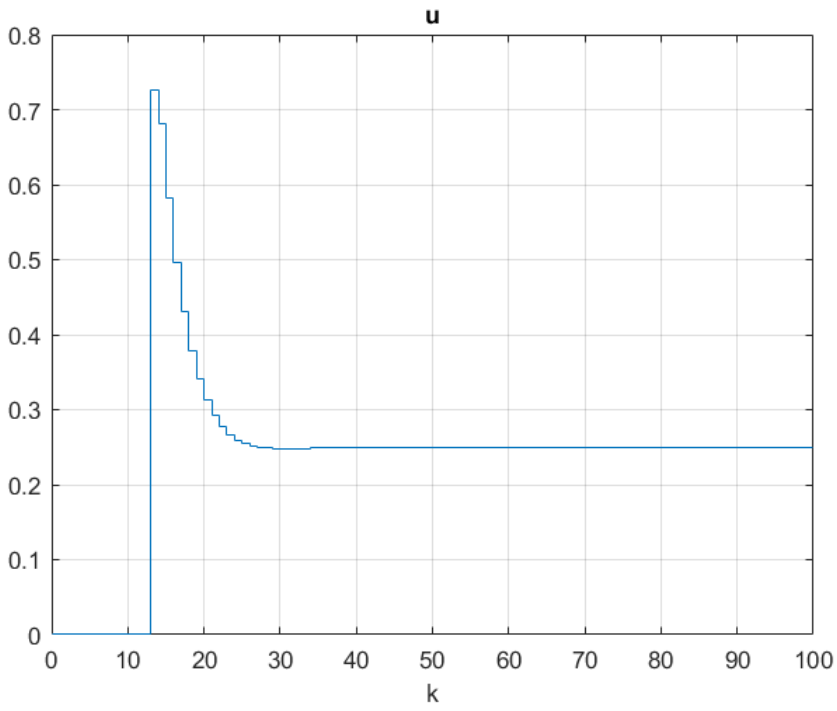
Kod implementacji głównej pętli algorytmu GPC w programie MATLAB przedstawiono poniżej.

```
for k = 13:kk
    y(k) = y_fun(k, y, u) + disturbance(k);

    d(k) = y(k) - y_fun(k, y, u);

    y_0 = zeros(1, N);

    y_0(1) = -b(2)*y(k) - b(3)*y(k-1) + c(2)*u(k-10)
    + c(3)*u(k-11) + d(k);
    y_0(2) = -b(2)*y_0(1) - b(3)*y(k) + c(2)*u(k-9)
    + c(3)*u(k-10) + d(k);
    for p=3:N
        if p <= 10
            y_0(p) = -b(2)*y_0(p-1) - b(3)*y_0(p-2) + c(2)*u(k-11+p)
            + c(3)*u(k-12+p) + d(k);
        else
            y_0(p) = -b(2)*y_0(p-1) - b(3)*y_0(p-2) + c(2)*u(k-1)
            + c(3)*u(k-1) + d(k);
        end
    end

    deltauk = K(1, :)*(yzad*ones(1, N) - y_0)';

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

    wyu(k) = u(k);
    wyy(k) = y(k);
end
```

Implementując parametry wykorzystane w algorytmie DMC, do regulatora GPC, otrzymano następujące wyniki symulacji. Na poniższych rysunkach przedstawiono odowiedź skokową oraz sygnał sterujący tego regulatora.



Rys. 5.1. Odpowiedź skokowa regulatora GPC

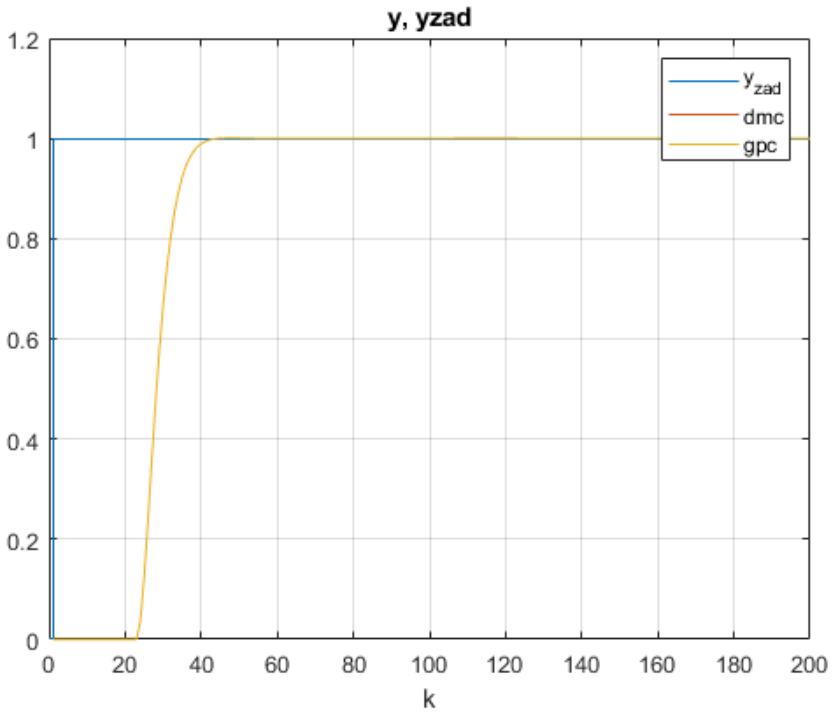


Rys. 5.2. Sygnał sterujący regulatora GPC

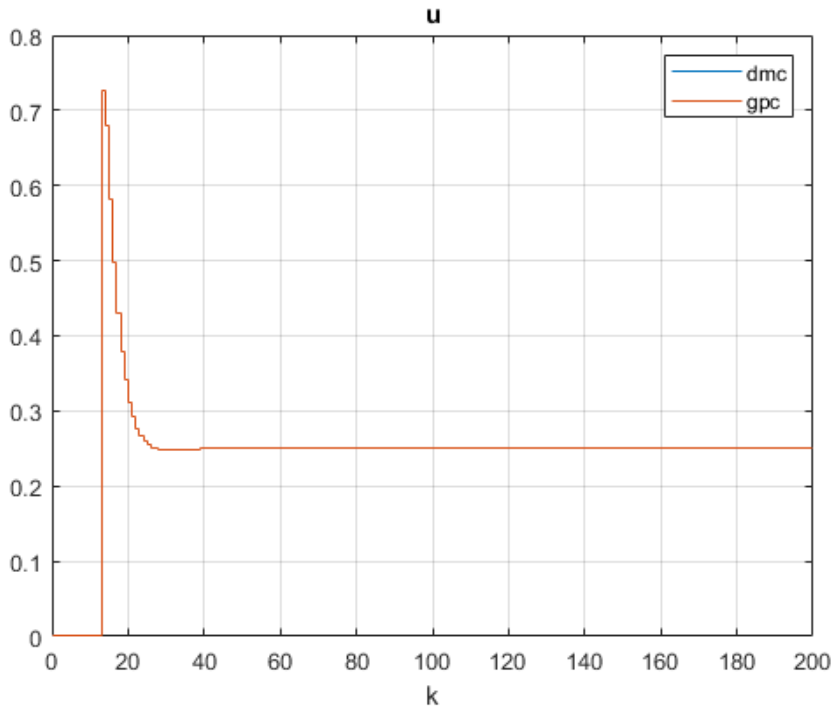
5.2. Porównanie regulatorów DMC i GPC

Mając dostrojone oba regulatory, można porównać ich działanie. Zostaną porównane na dwa sposoby: przy skokowej zmianie wartości zadanej orazy przy skokowej zmianie niemierzalnego zakłócenia.

5.2.1. Zmiana wartości zadanej

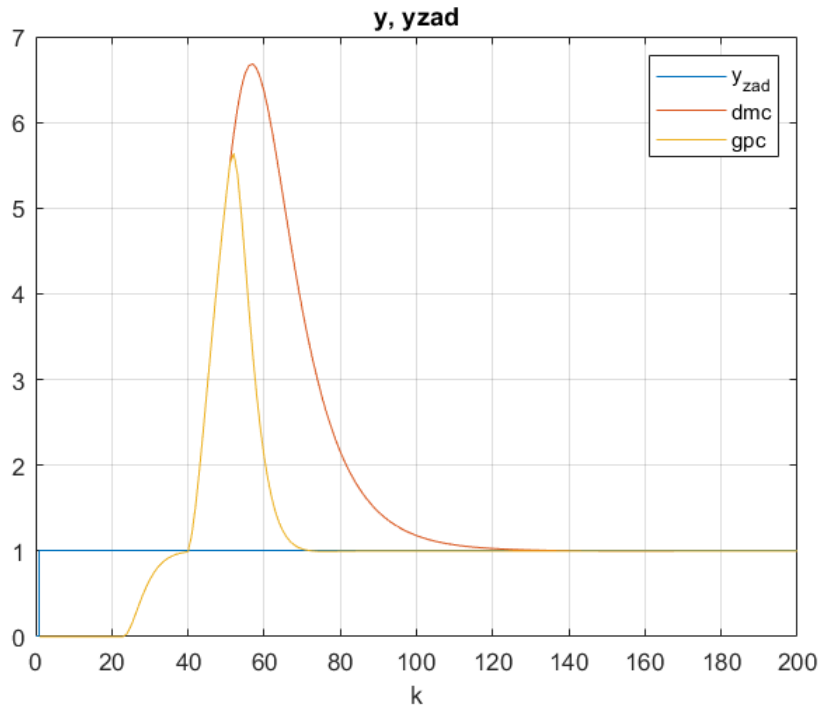


Rys. 5.3. Porównanie odpowiedzi skokowej regulatorów DMC i GPC

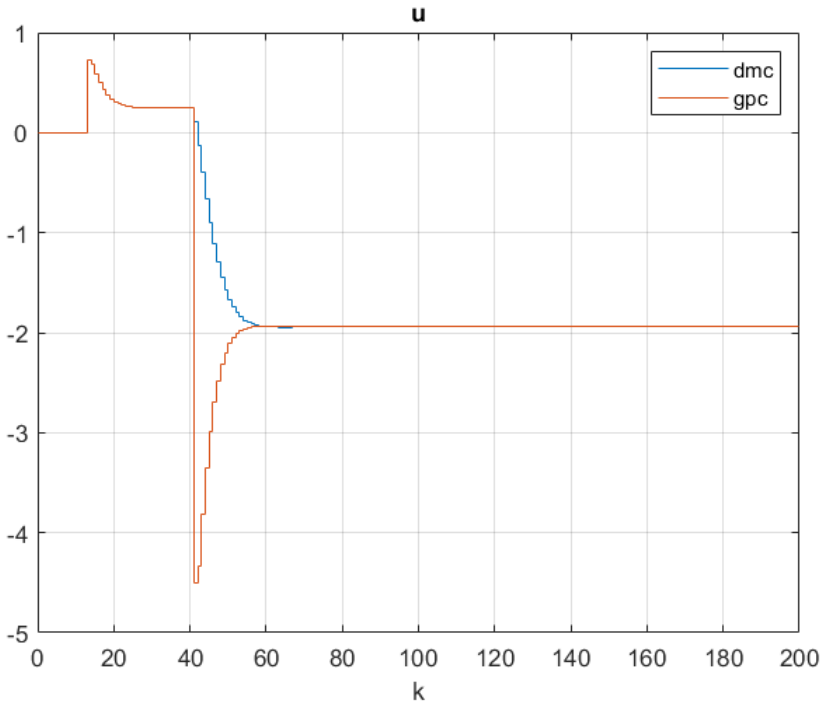


Rys. 5.4. Porównanie sygnału sterującego regulatorów DMC i GPC

5.2.2. Zmiana zakłócenia



Rys. 5.5. Porównanie odpowiedzi skokowej regulatorów DMC i GPC



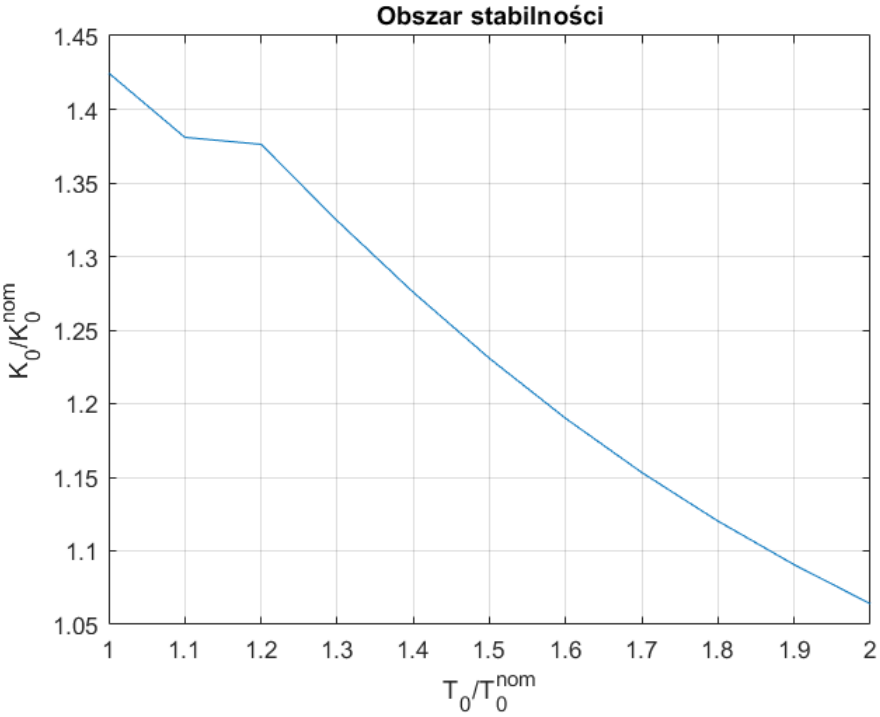
Rys. 5.6. Porównanie sygnału sterującego regulatorów DMC i GPC

Komentarz

Na podstawie odpowiedzi skokowych obiektu z regulatorami DMC oraz GPC można zauważyć, że regulator GPC działa bardzo podobnie do regulatora DMC przy braku zakłóceń. Jednak w sytuacji ich wystąpienia regulator GPC znacznie szybciej reaguje na zmiany, osiągając wartość zadaną w krótszym czasie, jednak przy gwałtowniejszej zmianie sygnału sterującego. Regulator DMC jest bardziej stabilny, ale osiąga wartość zadana nawet dwukrotnie wolniej.

5.3. Obszar stabilności regulatora GPC

Obszar stabilności przedstawia poniższy wykres.



Rys. 5.7. Obszar stabilności regulatora GPC

Dokładne wyniki przedstawione są w poniższej tabeli.

T_0/T_0^{nom}	K/K_{nom}
1.0	1.4246
1.1	1.3809
1.2	1.3762
1.3	1.3242
1.4	1.2756
1.5	1.2307
1.6	1.1900
1.7	1.1532
1.8	1.1202
1.9	1.0906
2.0	1.0640

Tab. 5.1. Obszar stabilności regulatora PID

Komentarz

Na wykresie obszaru stabilności widać wyraźnie niespodziewany skok dla wartości $T_0/T_0^{nom} < 1.2$. Może być to spowodowane wcześniej dobranymi parametrami regulatora, choć są to tylko przypuszczenia. Niemniej jednak widać podobieństwo obszarów stabliności regulatorów DMC i GPC, co może wynikać z podobnego sposobu działania obu regulatorów.