

## STP – Projekt nr 2 – zadanie nr 10

### Punkt 1:

Postać transmitancji:

$$G(s) = \frac{K_o e^{-T_o s}}{(T_1 s + 1)(T_2 s + 1)}$$

Dane:

$$K_o = 7,8; T_o = 5; T_1 = 2,2; T_2 = 4,36$$

Po wstawieniu podanych parametrów transmitancja będzie miała postać:

$$G(s) = \frac{7,8e^{-5s}}{(2,2s + 1)(4,36s + 1)}$$

Transmitancję dyskretną otrzymamy stosując komendy **c2d** oraz **tf** Matlaba w następującym programie (przyjmujemy czas próbkowania  $T_p=0,5s$ ):

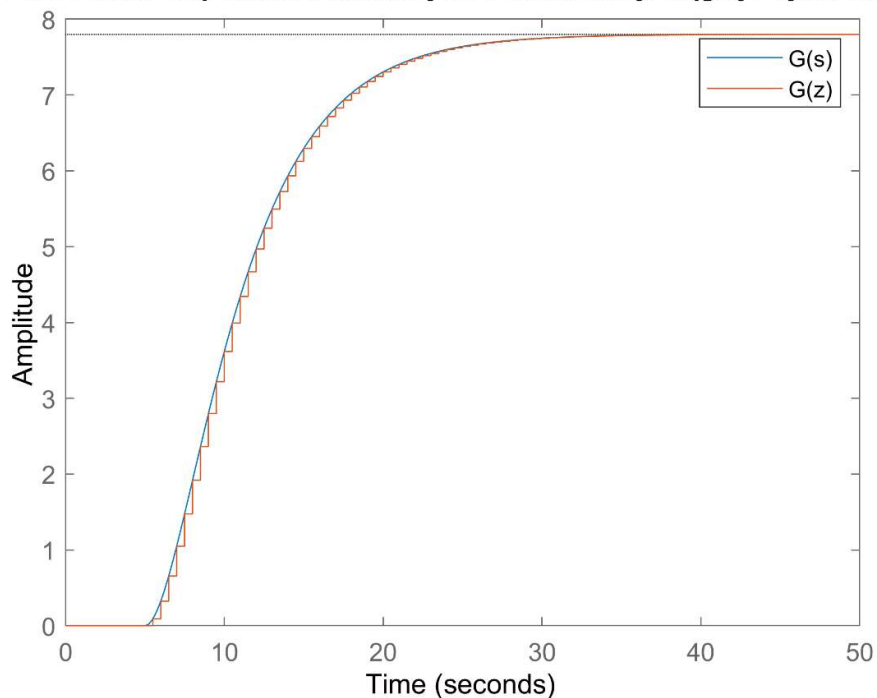
```
Gs = tf(Ko,[T1*T2 T1+T2 1], 'InputDelay',5);  
Gz = c2d(Gs,Tp, 'zoh');
```

Dzięki temu otrzymamy następującą transmitancję dyskretną:

$$G(z) = \frac{0.09079z^{-1} + 0.08101z^{-2}}{1 - 1.688z^{-1} + 0.7104z^{-2}} \cdot z^{-10}$$

**Porównanie odpowiedzi skokowej transmitancji ciągłej i dyskretniej:**

### Porównanie odpowiedzi skokowych transmitancji ciągłej i dyskretniej:



Aby obliczyć wzmocnienie statyczne transmitancji skorzystamy z komendy **dcgain**. Oto uzyskane wyniki:

Dla układu ciągłego:  $K_{stat} = 7,8$ .

Dla układu dyskretnego:  $K_{stat} = 7,8$ .

Jak widzimy powyżej odpowiedzi skokowe oraz wzmocnienia statyczne obu transmitancji są takie same.

### Punkt 2:

Wyznaczenie równania różnicowego na podstawie transmitancji dyskretniej:

$$G(z) = \frac{Y(z)}{U(z)} = \frac{0.09079z^{-1} + 0.08101z^{-2}}{1 - 1.688z^{-1} + 0.7104z^{-2}} \cdot z^{-10}$$

$$0.09079 \cdot U(z)z^{-1} + 0.08101 \cdot U(z)z^{-12} = Y(z) - 1,688 \cdot Y(z)z^{-1} + 0,7104 \cdot Y(z)z^{-2}$$

$$0.09079u(k-11) + 0.08101u(k-12) = y(k) - 1,688y(k-1) + 0,7104y(k-2)$$

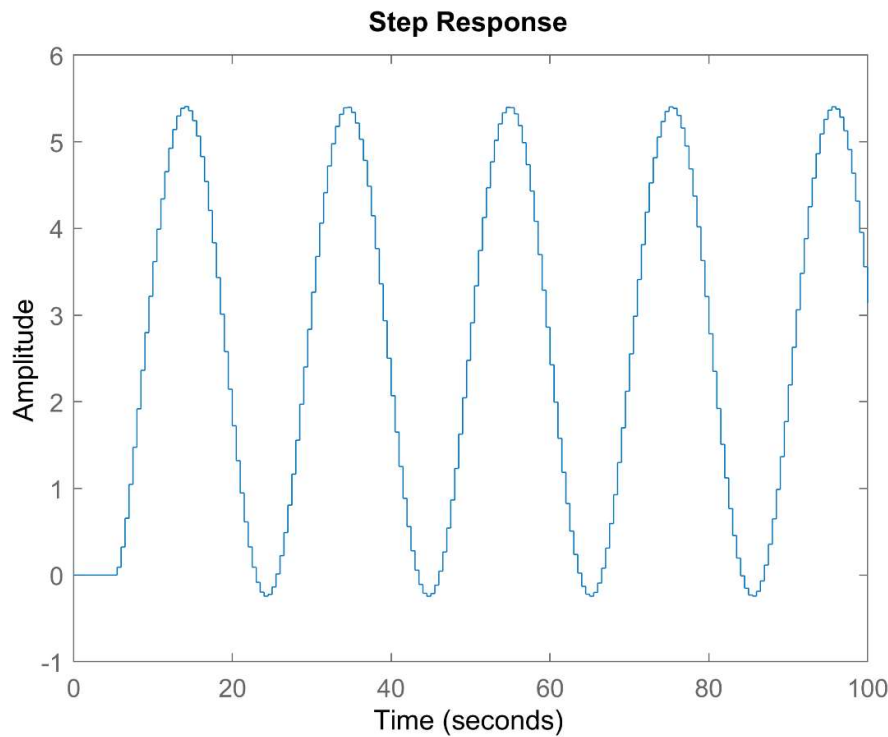
$$y(k) = 1,688y(k-1) - 0,7104y(k-2) + 0,09079u(k-11) + 0,08101u(k-12)$$

### Punkt 3:

Aby wprowadzić układ na granicę stabilności należy przed obiektem dać wzmocnienie oraz dodać do całego systemu sprzężenie zwrotne. Następnie ustalamy odpowiednio wartość dołączonego wzmocnienia i sprawdzamy czy przy odpowiedzi skokowej układu z dołączonym

wzmocnieniem znajduje się na granicy stabilności. Gdy doprowadzimy układ na granicę stabilności należy odczytać w wykresu okres oscylacji. Wartość dołączonego wzmocnienia, które doprowadziło układ do oscylacji to wzmocnienie krytyczne. Mając oba te parametry możemy parametry dyskretnego regulatora PID.

W celu doprowadzenia układu do granicy stabilności wykorzystamy komendę **feedback** dostępną w Matlabie:  $H = \text{feedback}(Gz, 0.2594)$ , dającą następujący rezultat:



W naszym przypadku wzmocnienie krytyczne wyniosło  $K_k=0,2594$  a okres oscylacji  $T_k=21,5$ .

Mając te dane wyznaczmy parametry  $K_r$ ,  $T_i$  oraz  $T_d$ :

$$K_r = 0,6K_k = 0,6 \cdot 0,2594 = 0,15564$$

$$T_i = 0,5T_k = 0,5 \cdot 21,5 = 10,75$$

$$T_d = 0,12T_k = 0,12 \cdot 21,5 = 2,58$$

Następnie wyznaczmy parametry  $r_2$ ,  $r_1$ ,  $r_0$ :

$$r_2 = \frac{KT_d}{T} = \frac{0,15564 \cdot 2,58}{0,5} = 0,8031$$

$$r_1 = K \left( \frac{T}{2T_i} - 2 \frac{T_d}{T} - 1 \right) = 0,15564 \left( \frac{0,5}{2 \cdot 10,75} - 2 \frac{2,58}{0,5} - 1 \right) = -1,7582$$

$$r_0 = K \left( \frac{T}{2T_i} + \frac{T_d}{T} + 1 \right) = 0,15564 \left( \frac{0,5}{2 \cdot 10,75} + \frac{2,58}{0,5} + 1 \right) = 0,9624$$

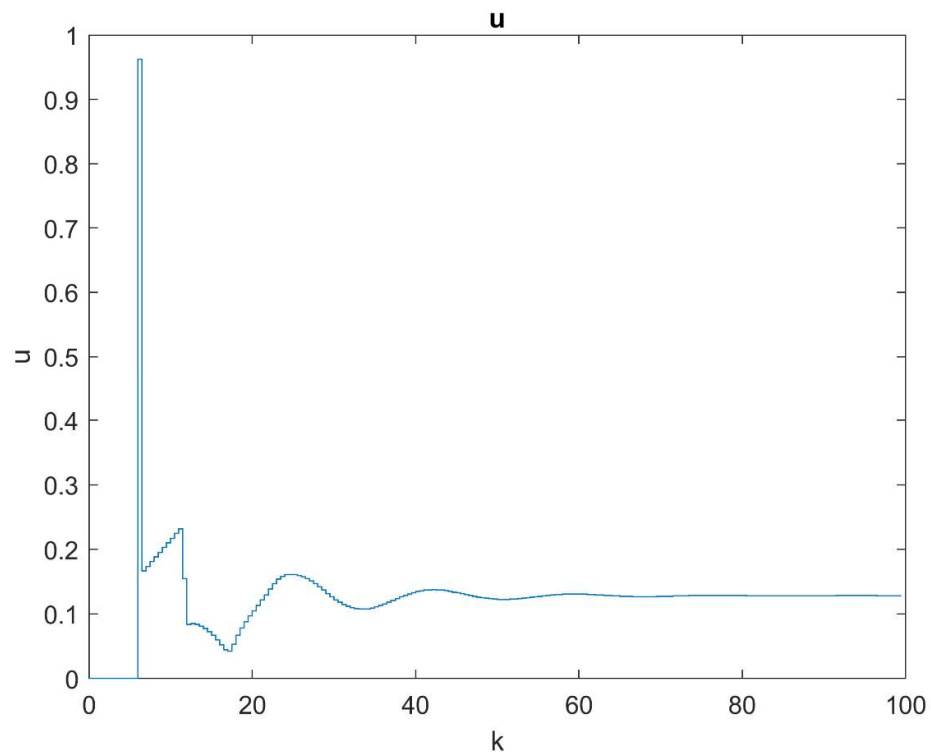
## Punkt 4:

Program do symulacji cyfrowego algorytmu PID:

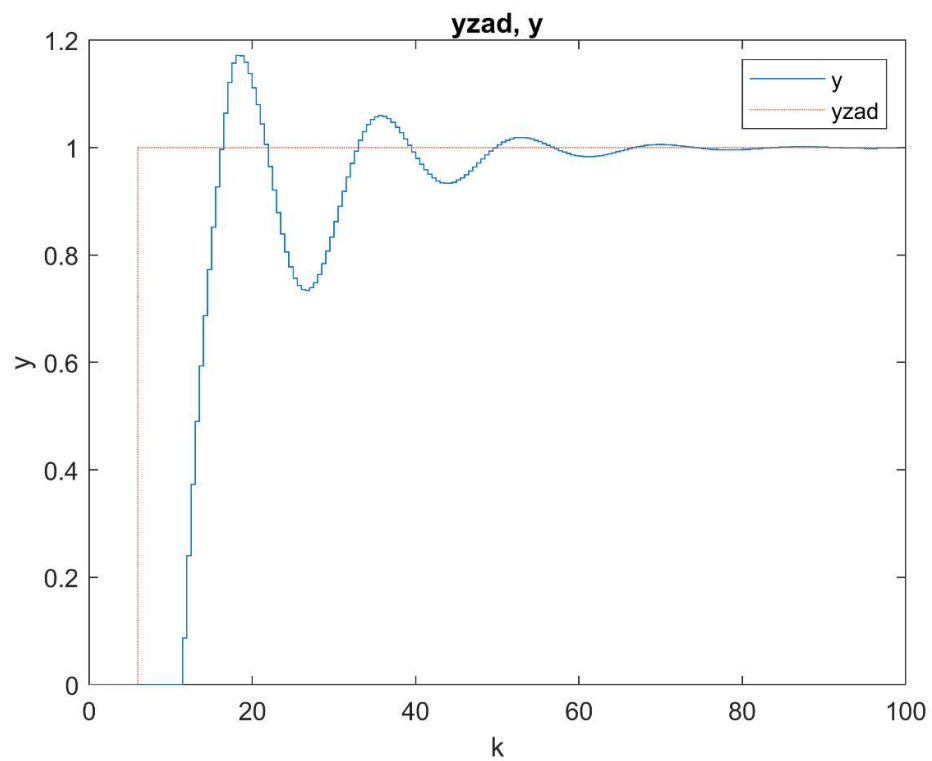
```
kk=200;
u(1:12)=0; y(1:12)=0;
yzad(1:12)=0; yzad(13:kk)=1;
e(1:12)=0;

for k=13:kk
    y(k)=-GzDenominator(2)*y(k-1)-GzDenominator(3)*y(k-2)+GzNumerator(2)*u(k-11)+GzNumerator(3)*u(k-12);
    e(k)=yzad(k)-y(k);
    u(k)=r2*e(k-2)+r1*e(k-1)+r0*e(k)+u(k-1);
end
figure; stairs(u);
title('u'); xlabel('k'); ylabel('u');
print('stp_2_4wykres1.png','-dpng','r400');
figure; stairs(y); hold on; stairs(yzad,':');
title('yzad, y'); xlabel('k'); ylabel('y');
print('stp_2_4wykres2.png','-dpng','r400');
```

Sterowanie  $u$  dla PID:



Wyjście  $y$  oraz  $y_{zad}$  dla PID:



Program do symulacji algorytmu DMC (do odpowiedzi skokowej wykorzystano model z punktu drugiego):

```

D=100;
N=100;
Nu=100;

s=[];
y(1:12)=0;
u(1:12)=0;
u(13:500)=1;

for k=13:500
    y(k)=-GzDenominator(2)*y(k-1)-GzDenominator(3)*y(k-2)+GzNumerator(2)*u(k-11)+GzNumerator(3)*u(k-12);
    s(k)=y(k);
end
s(1:13)=[];

Mp=zeros(N,D-1);
for i=1:D-1
    for j = 1:N
        Mp(j,i)=(s(j+i)-s(i));
    end
end

M=zeros(N,Nu);
for i=1:Nu
    M(i:N,i)=s(1:(N-i+1))';
end

YY=eye(N);
lambda=1;
A=eye(Nu).*lambda;
K=(M'*YY*M+A)^-1*M'*YY;

yzad=[];
ydmc(1:D)=0;
yzad(1:12)=0;
yzad(13:D)=1;
dUp(1:D-1,1)=0;
udmc(1:D)=0;
for k=13:D
    ydmc(k)=-GzDenominator(2)*ydmc(k-1)-GzDenominator(3)*ydmc(k-2)+GzNumerator(2)*udmc(k-11)+GzNumerator(3)*udmc(k-12);
    dU=K*(ones(N,1).*yzad(k)-ones(N,1).*ydmc(k)-Mp*dUp);
    udmc(k)=dU(1)+udmc(k-1);
    dUp(2:D-1)=dUp(1:D-2);
    dUp(1)=dU(1);
end

figure; stairs(udmc);
title('u'); xlabel('k');
print('stp_2_4wykres3.png','-dpng','-r400');
figure; stairs(ydmc); hold on; stairs(yzad,':');
legend('y','yzad')
title('yzad, y'); xlabel('k');
print('stp_2_4wykres4.png','-dpng','-r400');

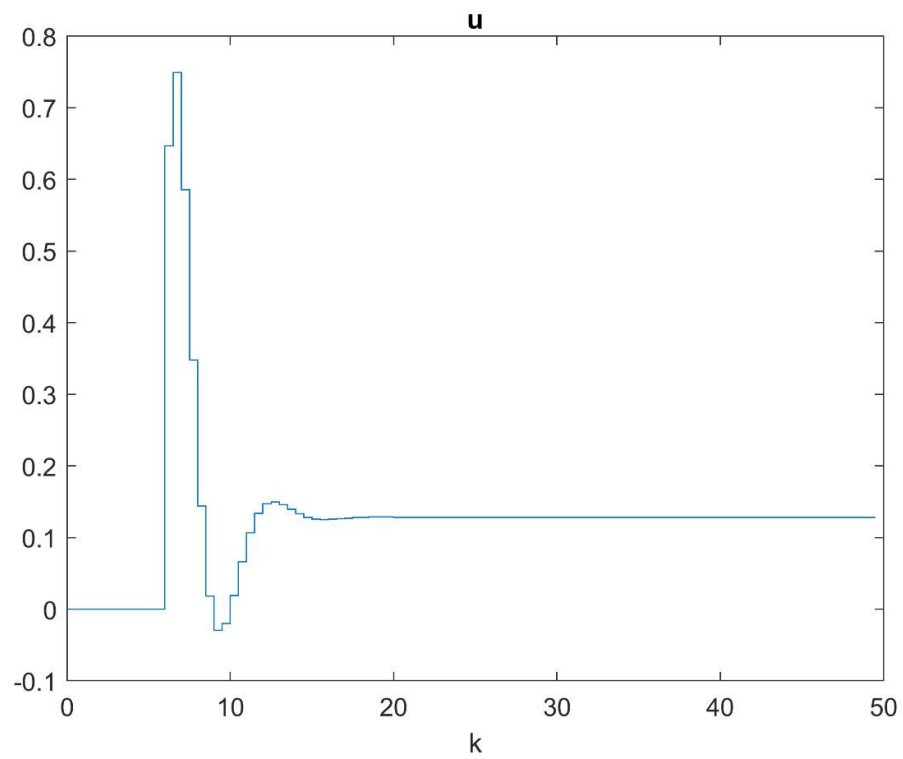
```

$D = 100$

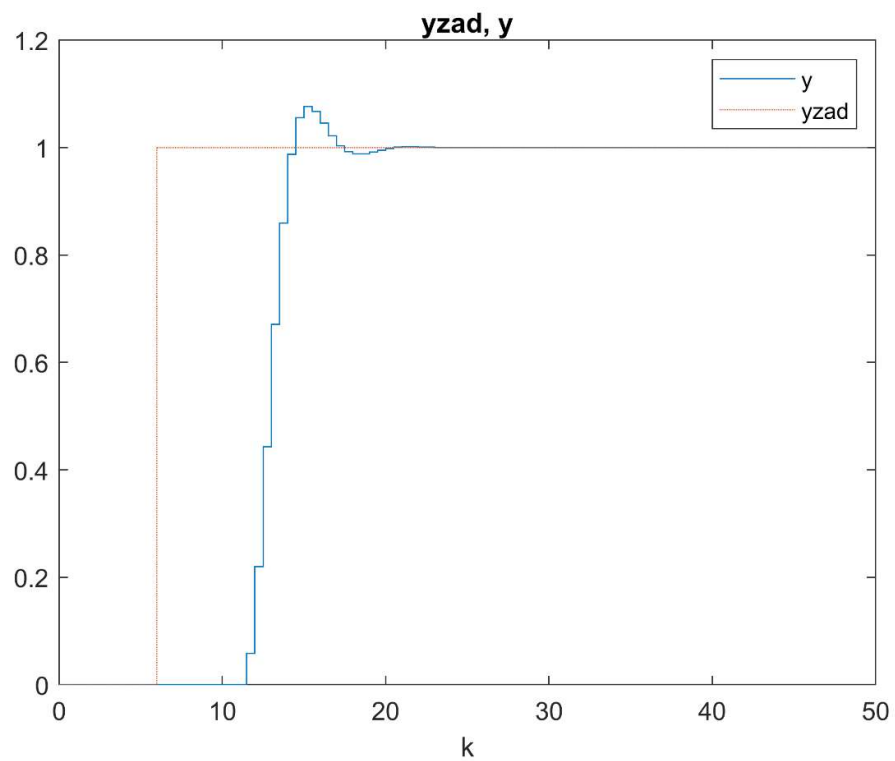
$N = 100$

$N_u = 100$

Sterowanie  $u$  dla DMC:



Wyjście  $y$  oraz  $y_{zad}$  dla DMC:



## Punkt 5:

### Podpunkt a):

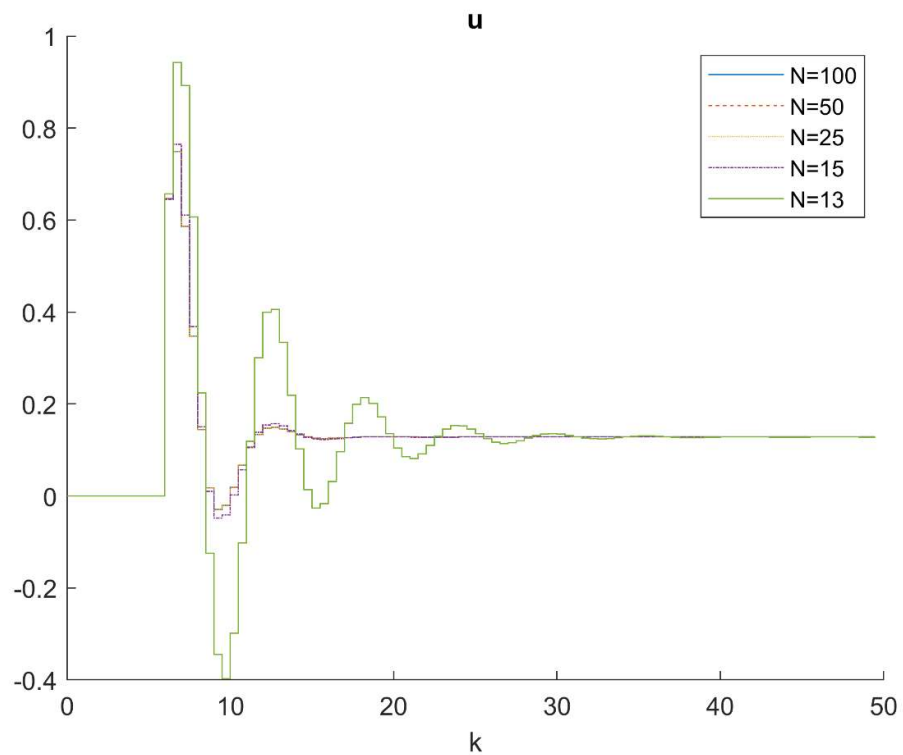
$D = 100$

$N = 100$

$N_u = 100$

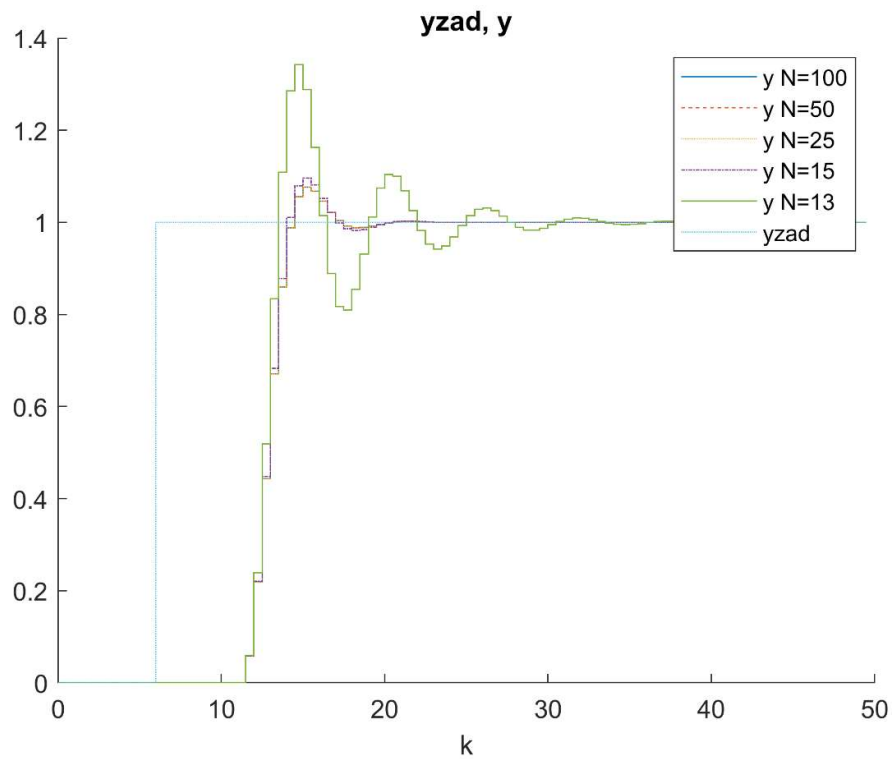
### Podpunkt b):

Sterowanie  $u$  dla DMC dla różnych horyzontów predykcji:



Wyjście  $y$  oraz  $y_{zad}$  dla DMC dla różnych horyzontów predykcji:

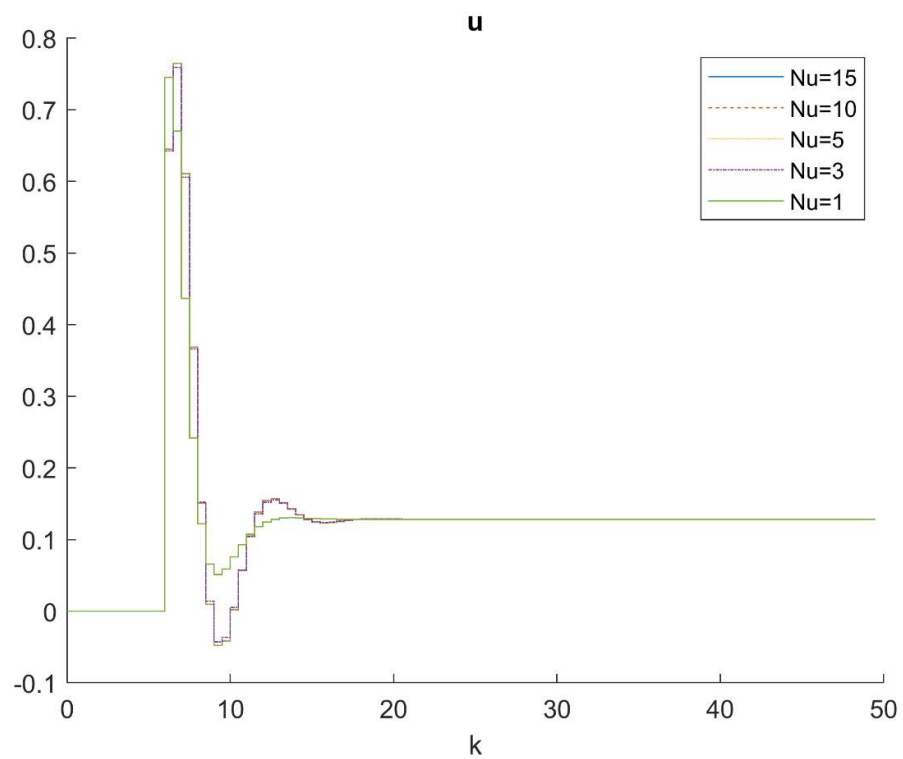




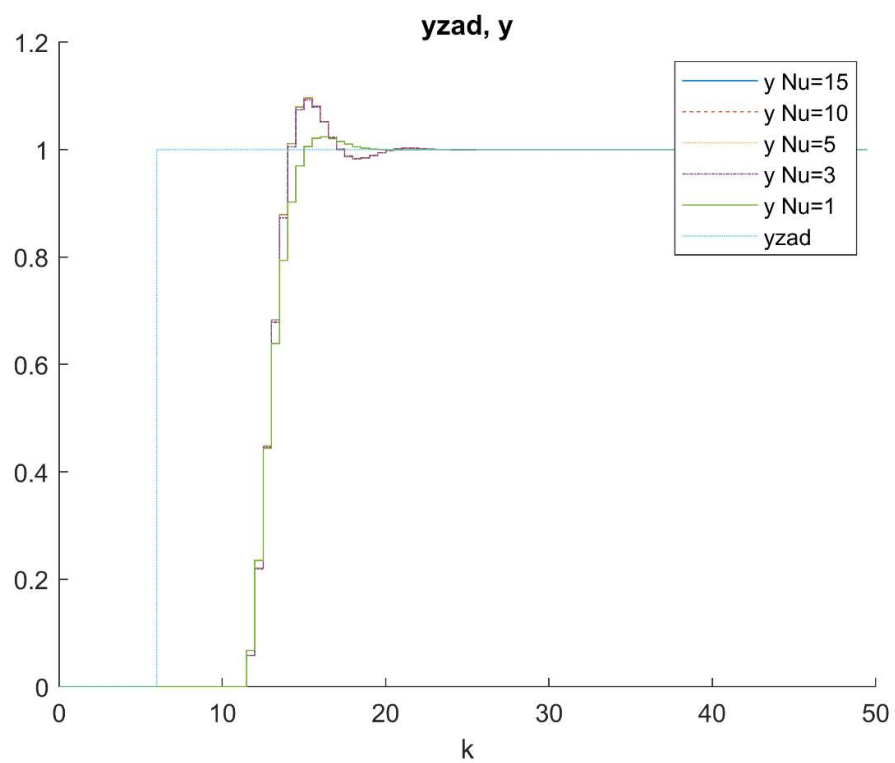
Najlepszym kompromisem pomiędzy nakładem obliczeń a jakością regulacji zapewnia horyzont predykcji  $N=15$  pomimo nieznacznego zwiększenia przeregulowania i czasu regulacji.

**Podpunkt c):**

Sterowanie  $u$  dla DMC dla horyzontu predykcji  $N=15$  i różnych horyzontów sterowania:



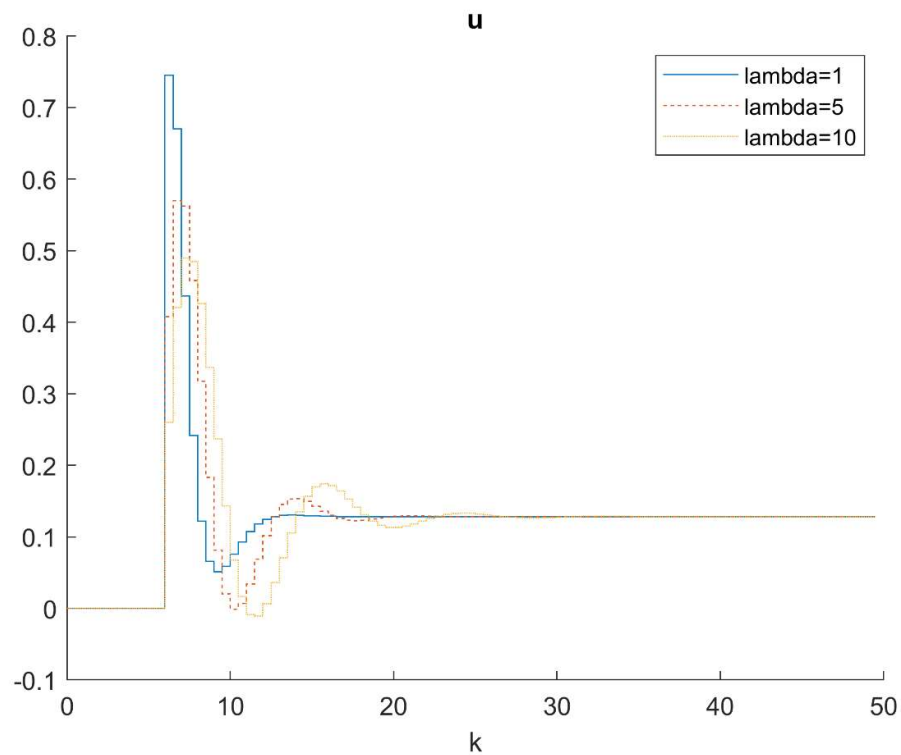
Wyjście  $y$  oraz  $y_{zad}$  dla DMC dla horyzontu predykcji  $N=15$  i różnych horyzontów sterowania:



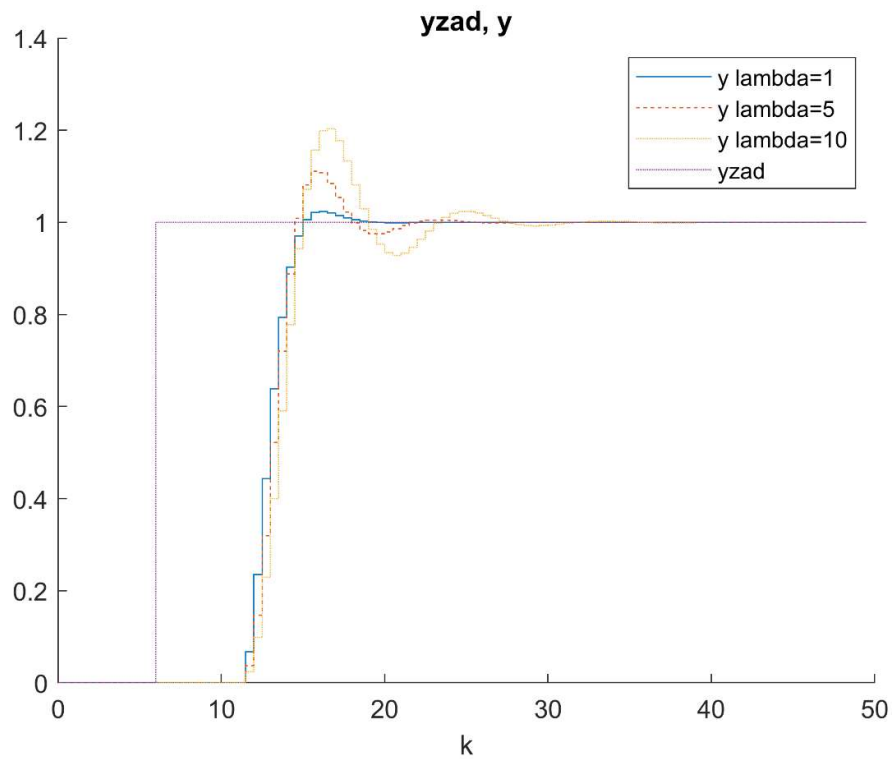
Najkrótszy horyzont sterowania ( $N_u = 1$ ) daje najszybszy czas regulacji oraz najmniejsze przeregulowanie.

**Podpunkt d):**

Sterowanie  $u$  dla DMC dla horyzontu predykcji  $N=15$ , horyzontu sterowania  $N_u=1$  oraz różnych parametrów  $\lambda$ :



Wyjście  $y$  oraz  $y_{zad}$  dla DMC dla horyzontu predykcji  $N=15$ , horyzontu sterowania  $N_u=1$  oraz różnych parametrów  $\lambda$ :



Jako kompromis między szybkością regulacji a kształtem sygnału sterującego wybieramy parametr  $\lambda=5$ .

Ostatecznie, najlepsze parametry dla regulatora DMC to:

$D = 100$ ,

$N = 15$ ,

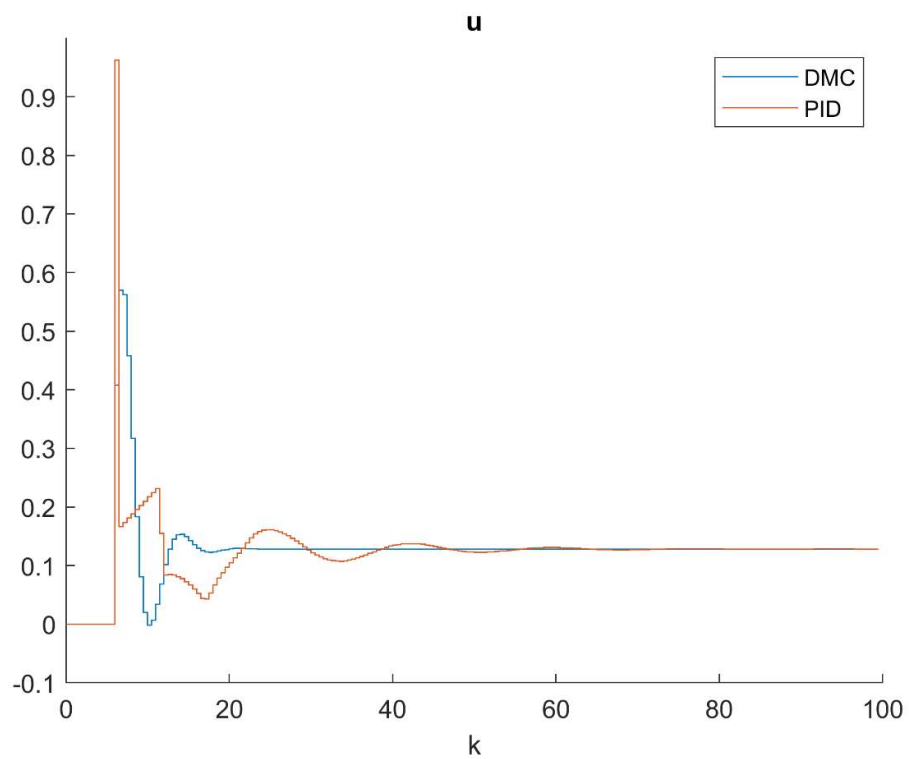
$N_u=1$ ,

$\lambda = 5$ ;

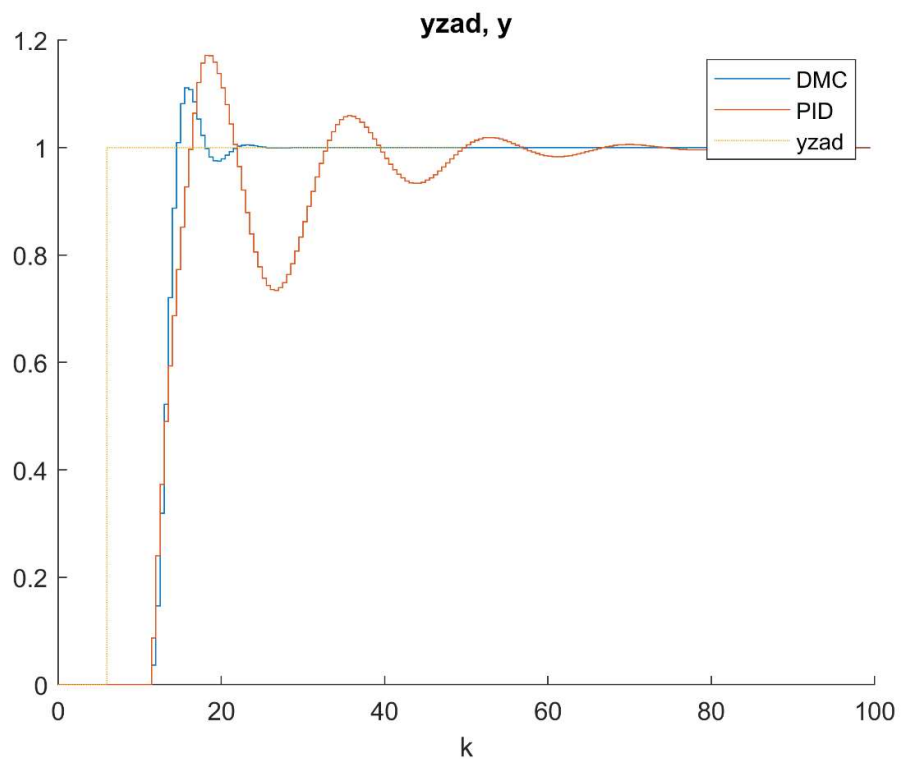
### Punkt 6:

Porównanie jakości regulacji między regulatorem DMC a regulatorem PID:

Sterowanie  $u$ :



Wyjście  $y$  na tle wartości zadanej:



Jak widzimy regulator DMC daje znacznie szybszy czas regulacji oraz mniejsze przeregulowanie.

Krzywą stabilności wyznaczono wstawiając do transmitancji obiektu odpowiednie opóźnienie oraz zmieniając wzmacnienie tak, aby doprowadzić układ do granicy stabilności.

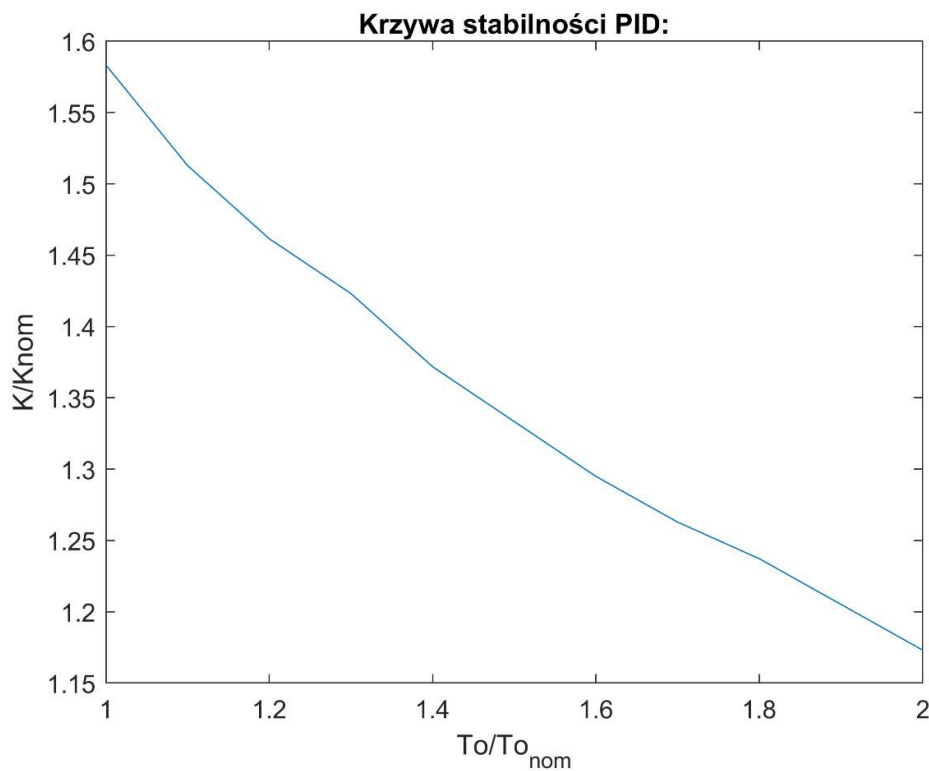
**PID:**

Wektor wzmacnień krytycznych:

$$K = \begin{bmatrix} 12,35 \\ 11,8 \\ 11,4 \\ 11,1 \\ 10,7 \\ 10,4 \\ 10,1 \\ 9,85 \\ 9,65 \\ 9,4 \\ 9,15 \end{bmatrix}$$

$$\frac{K}{K_{nom}} = \frac{K}{7.8}$$

Krzywa stabilności dla regulatora PID:



Obszar pod krzywą oznacza, że algorytm jest stabilny. Obszar nad krzywą oznacza, że algorytm jest niestabilny.

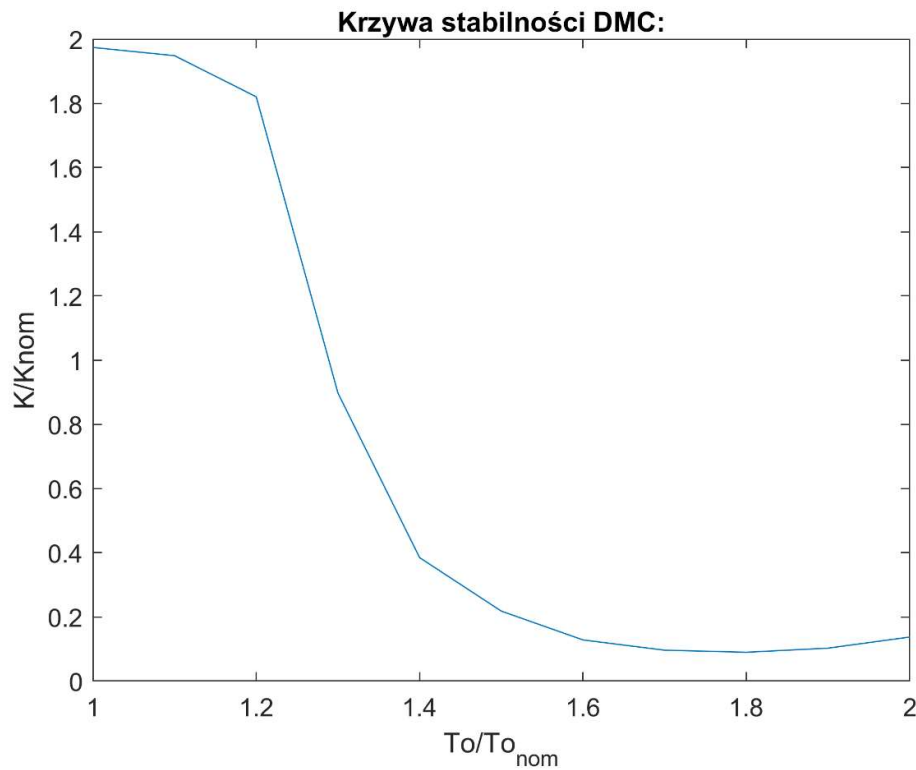
**DMC:**

Wektor wzmocnień krytycznych:

$$K = \begin{bmatrix} 15,4 \\ 15,2 \\ 14,2 \\ 7 \\ 3 \\ 1,7 \\ 1 \\ 0,75 \\ 0,7 \\ 0,8 \\ 1,07 \end{bmatrix}$$

$$\frac{K}{K_{nom}} = \frac{K}{7,8}$$

Krzywa stabilności dla regulatora DMC:



Obszar pod krzywą oznacza, że algorytm jest stabilny. Obszar nad krzywą oznacza, że algorytm jest niestabilny.

Jak widzimy dla algorytmu PID krzywa przypomina linię prostą. Oznacza to, że stosunki  $\frac{K}{K_{nom}}$  oraz  $\frac{T_o}{T_{o,nom}}$  są do siebie odwrotnie proporcjonalne. Dla algorytmu DMC krzywa przypomina kształtem parabolę. Można wysnuć wniosek, że dla nieznacznie większych opóźnień od

nominalnych algorytm DMC pozwala na zastosowanie większego wzmocnienia, jednak dla większych opóźnień przy większych wzmocnieniach lepszym wyborem będzie regulator PID.