Sprawozdanie STP
Projekt nr.2
Zadanie 9

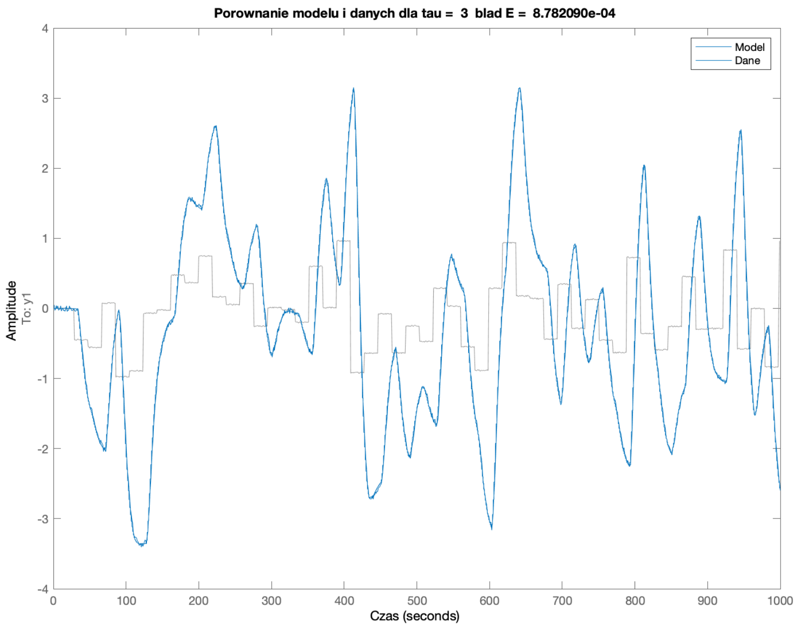
Kajetan Kaczmarek

18 stycznia 2019

1. Wyznaczamy modele w postaci :

$$y(k) = b_\tau u(k-\tau) + b_{\tau+1} u(k-\tau-1) - a_1 y(k-1) - a_2 y(k-2)$$

Gdzie τ to opóźnienie systemu. Wybrałem model dla $\tau = 3$, wykres wyjścia modelu w porównaniu do danych z pliku :



Rysunek 1: Model dla $\tau = 3$

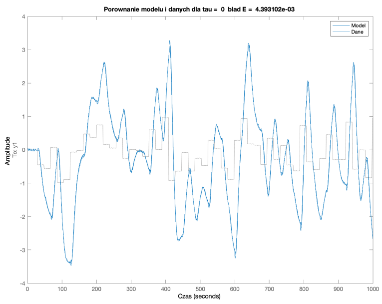
Obliczona transmitancja to

$$G(z) = -\frac{0.0535\,z^{-4} + 0.05131^{-5}}{1 - 1.664\,z^{-1} + 0.692\,z^{-2}}$$

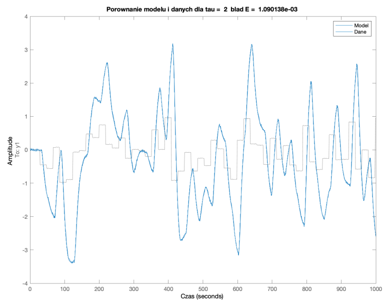
Błędy dla kolejnych wartości opóźnienia :

τ	Błąd	τ	Błąd
0	0.00439310239906314	1	0.00183921094764360
2	0.00109013781103131	3	0.000878208985142877
4	0.00127255618226417	5	0.00265973087538625
6	0.00466320637124297	7	0.00657471083346129
8	0.0144459616721526	9	0.0291933730835489
10	0.0505076805408859	11	0.0815671234071185
12	0.115793906377645	13	0.141062965735366
14	0.192723451767430	15	0.224393357249063
16	0.279145538665909	17	0.327832479282838
18	0.266753410572330	19	0.526295935361537
20	6.68422436659490	X	X

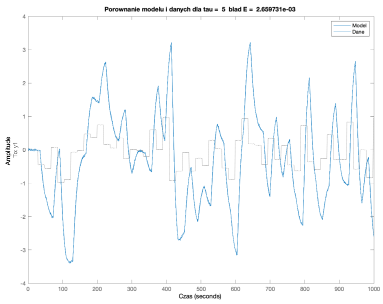
Jak widać błąd jest najmniejszy dla opóźnienia równego 3. Ponieżej podaje przykładowe inne modele dla porównania.Można zauważyć że dla opóźnienia $\tau = 3$ odwzorowanie jest najwierniejsze.



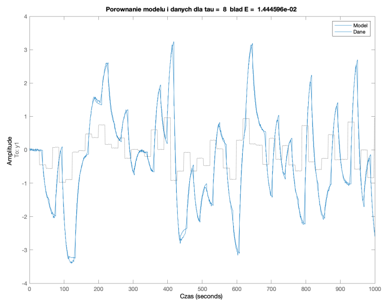
(a) Model dla $\tau = 0$



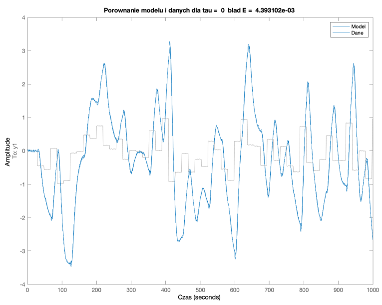
(b) Model dla $\tau = 2$



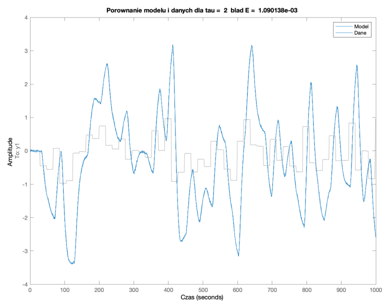
(c) Model dla $\tau = 5$



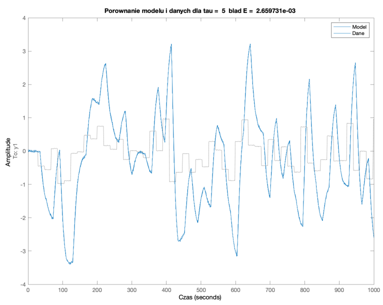
(d) Model dla $\tau = 8$



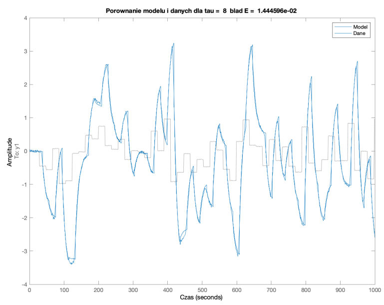
(a) Model dla $\tau = 10$



(b) Model dla $\tau = 12$



(c) Model dla $\tau = 15$



(d) Model dla $\tau = 20$

- Trasmitancja dla $\tau = 0$

$$G(z) = -\frac{-0.1237\, z^{-1} + 0.202^{-2}}{1 - 1.732\, z^{-1} + 0.753\, z^{-2}}$$

- Trasmitancja dla $\tau = 2$

$$G(z) = -\frac{-0.03262\, z^{-3} + 0.132^{-4}}{1 - 1.678\, z^{-1} + 0.7041\, z^{-2}}$$

- Trasmitancja dla $\tau = 5$

$$G(z) = -\frac{0.2905\, z^{-6} - 0.2136^{-7}}{1 - 1.725\, z^{-1} + 0.7455\, z^{-2}}$$

- Trasmitancja dla $\tau = 8$

$$G(z) = -\frac{1.03\, z^{-9} - 0.5651^{-10}}{1 - 1.0.7708\, z^{-1} - 0.1014\, z^{-2}}$$

- Trasmitancja dla $\tau = 10$

$$G(z) = -\frac{1.463\, z^{-11} - 1.075^{-12}}{1 - 0.9145\, z^{-1} + 0.026\, z^{-2}}$$

- Trasmitancja dla $\tau = 12$

$$G(z) = -\frac{1.741\, z^{-13} - 1.463^{-14}}{1 - 1.048\, z^{-1} + 0.1325\, z^{-2}}$$

- Trasmitancja dla $\tau = 15$

$$G(z) = -\frac{2.085\, z^{-16} - 1.941^{-17}}{1 - 1.078\, z^{-1} + 0.1273\, z^{-2}}$$

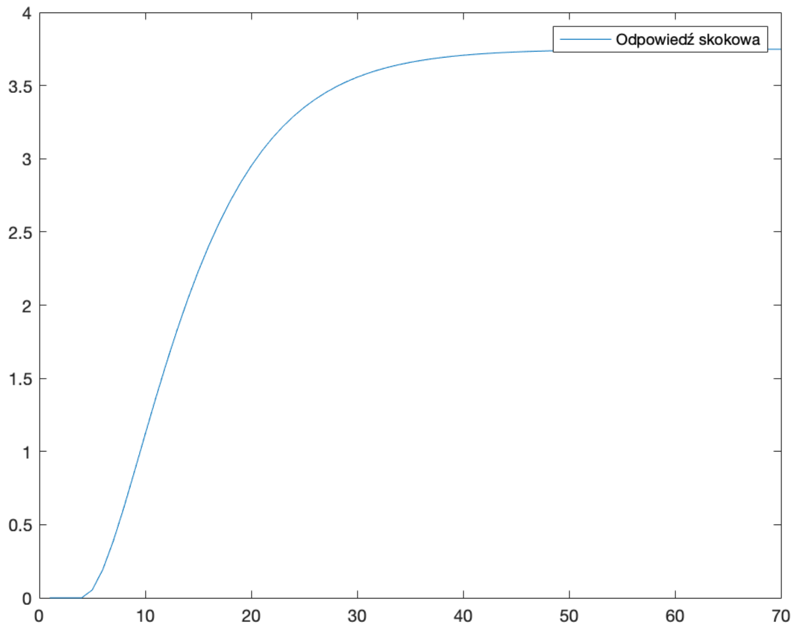
- Trasmitancja dla $\tau = 20$

$$G(z) = -\frac{2.306\, z^{-21} - 2.116^{-22}}{1 - 0.8561\, z^{-1} - 0.0563\, z^{-2}}$$

Kod użyty do policzenia punktu 1go:

```
%%%%% %%%%% %%%%% %%%%% %%%%% %%%%% %%%%% %%%%% %%%%% %%%%% %%%%% %%%%%
% Projekt nr. 2 STP − Kajetan Kaczmarek
% Punkt 1 − wyznaczenie modelu oraz symulacja
%%%%% %%%%% %%%%% %%%%% %%%%% %%%%% %%%%% %%%%% %%%%% %%%%% %%%%% %%%%%
function [systems] = P1()
%%%%% %%%%% %%%%% %%%%% %%%%% %%%%% %%%%% %%%%% %%%%% %%%%% %%%%% %%%%%
% Inicjalizacja − okres próbkowania i maksymalne testowane opoznienie
Ts = 1; maxTau = 20;
%%%%% %%%%% %%%%% %%%%% %%%%% %%%%% %%%%% %%%%% %%%%% %%%%% %%%%% %%%%%
% Pobranie danych z pliku
[u,y]=getDataFromFile("dane9.txt");
% Wyliczenie modeli wej¶ciowych w oparciu o dane
modelData = iddata(y,u,Ts);
systems = idtf(zeros(1,1,maxTau));
% Wyznaczenie modeli o opoznieniu do maxTau
for j = 0:maxTau
    tau = j;
    delay = tau + 1;
    % Podajemy licznik i mianownik tak aby otrzymaæ model z dwoma
    % skladnikami w mianowniku i liczniku
    N = [zeros(delay ,1)' NaN NaN];
    D = [ 1 NaN NaN];

    model = idtf(N,D,Ts);
    % Oznaczamy warto¶ci licznika do dwóch ostatnich jako niezmienialne
    % zera na potrzeby tfest
    for i = 1:delay
        model.Structure.num.Free(i) = false;
    end
    % Wyliczamy po kolei nasze systemy
    systems(:,:,j+1) = tfest(modelData,model);
end
% Wyznaczenie bledu dla kazdego modelu i narysowanie wykresow
Errors = zeros(1,maxTau+1);
for j = 0:maxTau
   [~,Errors(j+1)]=P1_Draw(systems(:,:,j+1),u,y,j);
end
```

2. Wzmocnienie statyczne : K = 3,7432
   Odpowiedź skokowa :



Rysunek 4: Odpowiedź skokowa dla $\tau = 3$

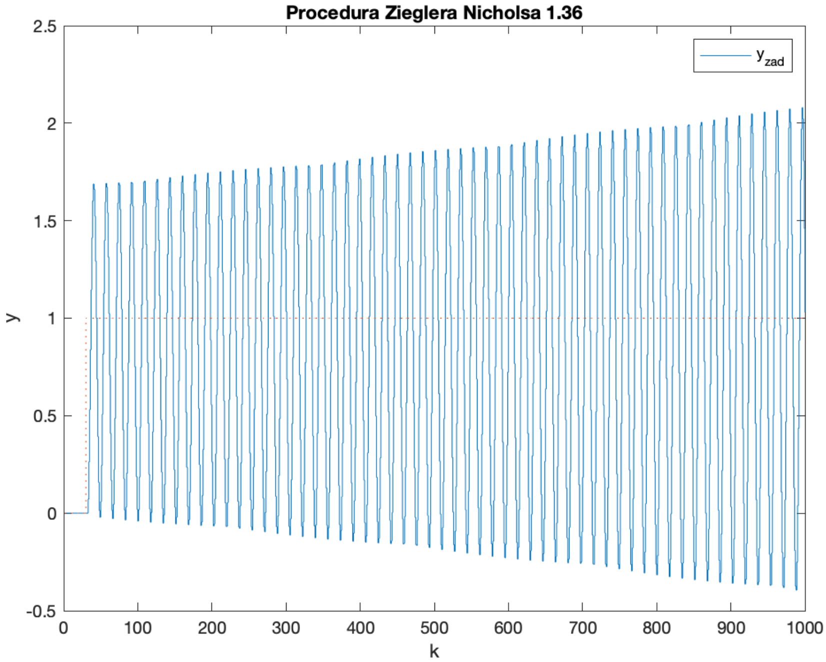
Oraz w formie tabeli :

I	1	2	3	4	5
S	0	0	0	0	0.05350337749
I	6	7	8	9	10
S	0.1938495617	0.3903675229	0.620261978	0.8668251224	1.118028917
I	11	12	13	14	15
S	1.365420087	1.60325549	1.827828011	2.036943139	2.22951446
I	16	17	18	19	20
S	2.405252755	2.564428553	2.707692175	2.835938574	2.950206968
I	21	22	23	24	25
S	3.051607351	3.141267655	3.220296689	3.289759025	3.350658852
I	26	27	28	29	30
S	3.403930501	3.450433829	3.49095312	3.526198425	3.556808576
I	31	32	33	34	35
S	3.583355254	3.606347689	3.626237655	3.643424532	3.658260277
I	36	37	38	39	40
S	3.671054189	3.682077398	3.69156704	3.699730096	3.706746891
I	41	42	43	44	45
S	3.712774264	3.717948414	3.722387449	3.726193656	3.729455513
I	46	47	48	49	50
S	3.73224947	3.734641511	3.736688539	3.738439577	3.739936825
I	51	52	53	54	55
S	3.741216583	3.742310053	3.743244033	3.744041531	3.744722282
I	56	57	58	59	60
S	3.745303209	3.745798813	3.746221515	3.746581948	3.746889213
I	61	62	63	64	65
S	3.747151093	3.747374245	3.747564356	3.747726286	3.747864188
I	66	67	68	69	70
S	3.747981606	3.748081566	3.748166649	3.748239059	3.748300673

Kod użyty do policzenia punktu 2go:

```
%%%%% %%%%% %%%%% %%%%% %%%%% %%%%% %%%%% %%%%% %%%%% %%%%% %%%%% %%%%%
% Projekt nr. 2 STP - Kajetan Kaczmarek
% Punkt 2 - wyznaczenie odpowiedzi skokowej
% Model z tau = 3 daje najmniejszy sredni blad
%%%%% %%%%% %%%%% %%%%% %%%%% %%%%% %%%%% %%%%% %%%%% %%%%% %%%%% %%%%%
function [] = P2()
systems = P1();
sys = systems(:,:,4);
%Odpowiedz skokowa
s = step(sys)
%Wzmocnienie statyczne
K = dcgain(sys);
% K = 3,7432
plot(s)
```

3. Algorytm PID - $K_{kr} = 1.36, T_{kr} = 20$
   Oscylacje krytyczne metodą Zieglera-Nicholsa:

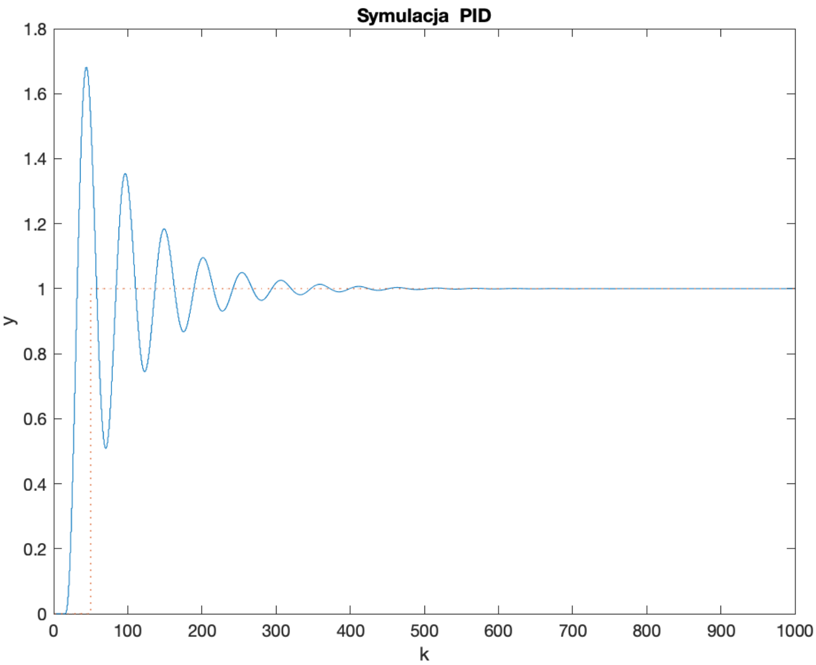


Rysunek 5: Metoda Zieglera-Nicholsa

Wyznaczenie wzmocnienia krytycznego metodą Zieglera - Nicholsa :

```
%%%%% %%%%% %%%%% %%%%% %%%%% %%%%% %%%%% %%%%% %%%%% %%%%% %%%%% %%%%%
% Projekt nr. 2 STP - Kajetan Kaczmarek
% Punkt 3 - dobor parametrow metoda Zieglera-Nicholsona
% Program w oparciu o algorytm przedstawiony na stronie 90-91 skryptu
% do przedmiotu
%%%%% %%%%% %%%%% %%%%% %%%%% %%%%% %%%%% %%%%% %%%%% %%%%% %%%%% %%%%%

%inicjalizacja
a1=-1.664; a0=0.692; b1=0.0535; b0=0.05131;
K_r_Base =0.05;
kk=1000;

for i = 27:0.1:28
    y = zeros(kk,1);
    %warunki poczatkowe
    u(1:4)=0;
    yzad(1:29)=0; yzad(30:kk)=1;
    e(1:4)=0;
    K_r=K_r_Base*i;
    for k=5:kk %glowna petla symulacyjna , k_min = 6 dla zachowania
        ciaglosci
        %symulacja obiektu
         y(k)=b1*u(k-3)+b0*u(k-4)-a1*y(k-1)-a0*y(k-2);
        %uchyb regulacji
         e(k)=yzad(k)-y(k);
        %sygnal sterujacy regulatora PID
         u(k)=K_r*e(k);
    end

    % drukuj i zapisz wyniki symulacji
    P3_Draw(K_r, y, yzad);
end
```

Wykres dla algorytmu PID :



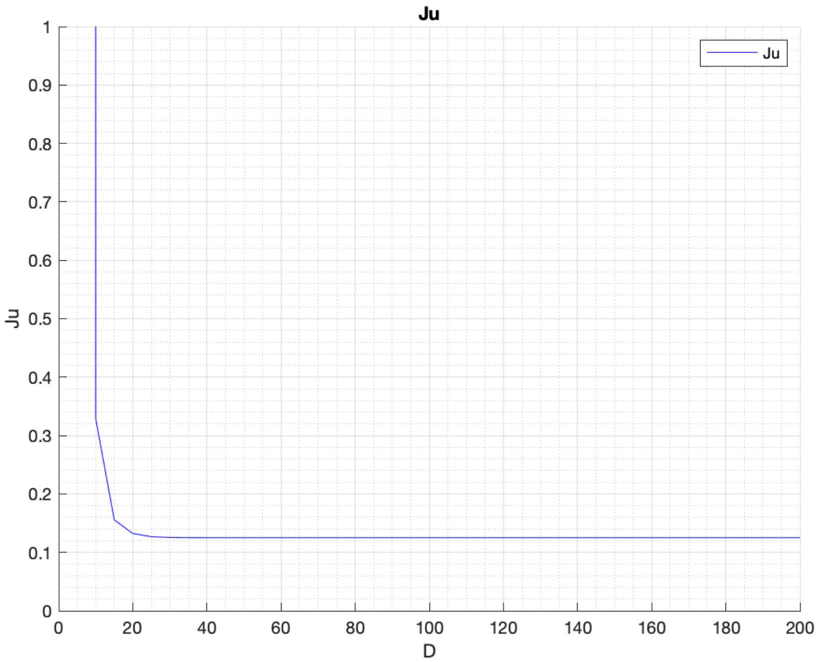
Rysunek 6: Algorytm PID

Symulacja algorytmu PID :

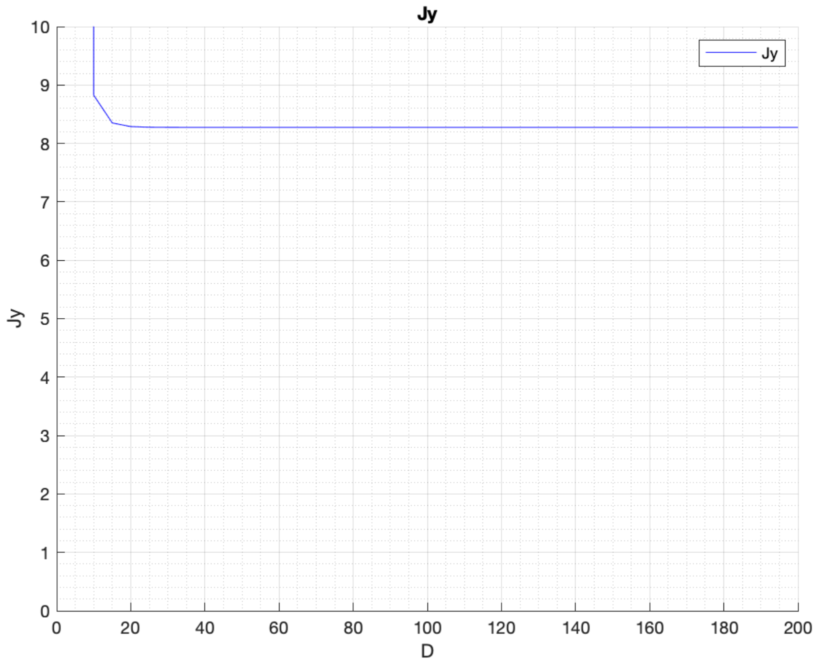
```
%%%%% %%%%% %%%%% %%%%% %%%%% %%%%% %%%%% %%%%% %%%%% %%%%% %%%%% %%%%%
% Projekt nr. 2 STP - Kajetan Kaczmarek
% Punkt 4 -symulacja regulatora PID
%%%%% %%%%% %%%%% %%%%% %%%%% %%%%% %%%%% %%%%% %%%%% %%%%% %%%%% %%%%%

%inicjalizacja
kk = 1000; % czas symylacji
y_zad = zeros(1,kk); % wartosc zadana w funkcji czasu
y_zad(13:kk) = 1; % skok wartosci zadanej do 1 w k = 13
y=zeros(1,kk);
u=zeros(1,kk);
e=zeros(1,kk);

%Parametry naszego modelu
a1=-1.664; a0=0.692; b1=0.0535; b0=0.05131; T= 0.5;
%Procedura ZN
K_kr = 1.36; T_kr =20 ;
%Wyliczenie parametrów PID
K = 0.6*K_kr; Ti = 0.5*T_kr; Td = 0.12*T_kr;
%The controller parameters are proportional gain K, integral time Ti, and derivative time Td
% Wartosci PID dyskretnego policzone z ww.
r1 = K*(1 + T/(2*Ti) + Td/T );
r2 = K*(T/(2*Ti) - 2*Td/T - 1);
r3 = K*Td/T;

%warunki poczatkowe
u(1:4)=0; y(1:4)=0;
yzad(1:49)=0; yzad(50:kk)=1;
e(1:4)=0;

for k=6:kk
    y(k)=b1*u(k-3)+b0*u(k-4)-a1*y(k-1)-a0*y(k-2);
```

```
    e(k) = y_zad(k) - y(k);
    u(k) = u(k-1) + r1*e(k) + r2*e(k) +r3*e(k);
end
f = figure();
stairs(y); hold on; stairs(yzad,':');
title('Symulacja PID' ); xlabel('k'); ylabel('y');
title = sprintf(strrep(['ModelsP3/PID/P4-2_r1_' num2str(r1) 'r2_' num2str(
    r2) 'r3_' num2str(r3)],'.',''));
title = strrep(title,'.','_');
print(f, title, '-dpng');
```

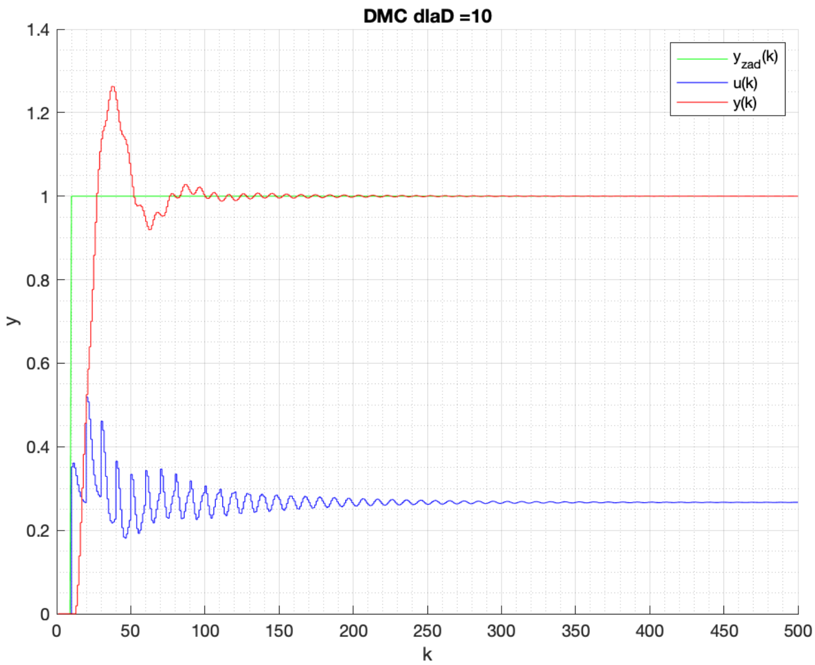
4. Wyznaczanie parametrów DMC
   Wyznaczone parametry : D = 50, N = 15, Nu = 2, $\lambda = 3$



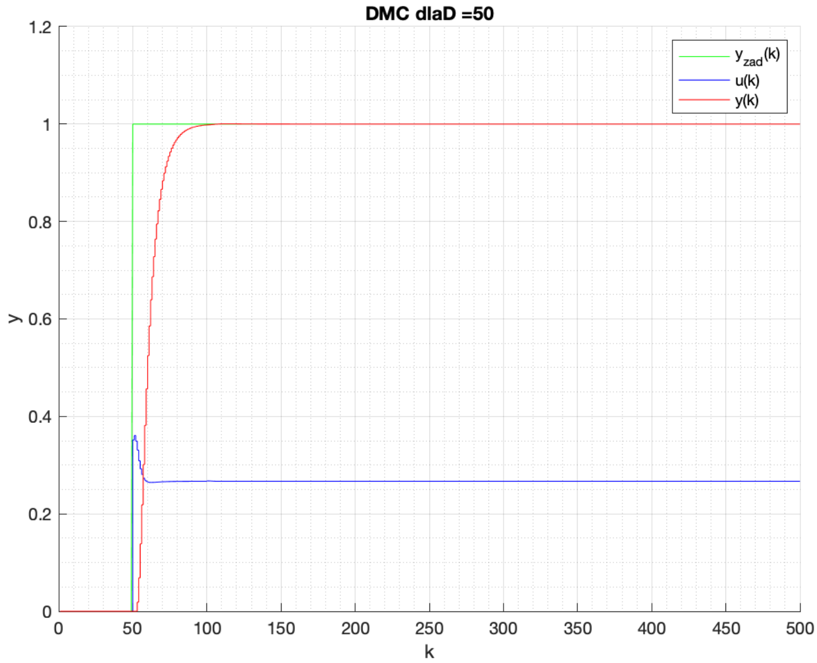
Rysunek 7: Wyznaczone parametry Ju dla różnych wartości D



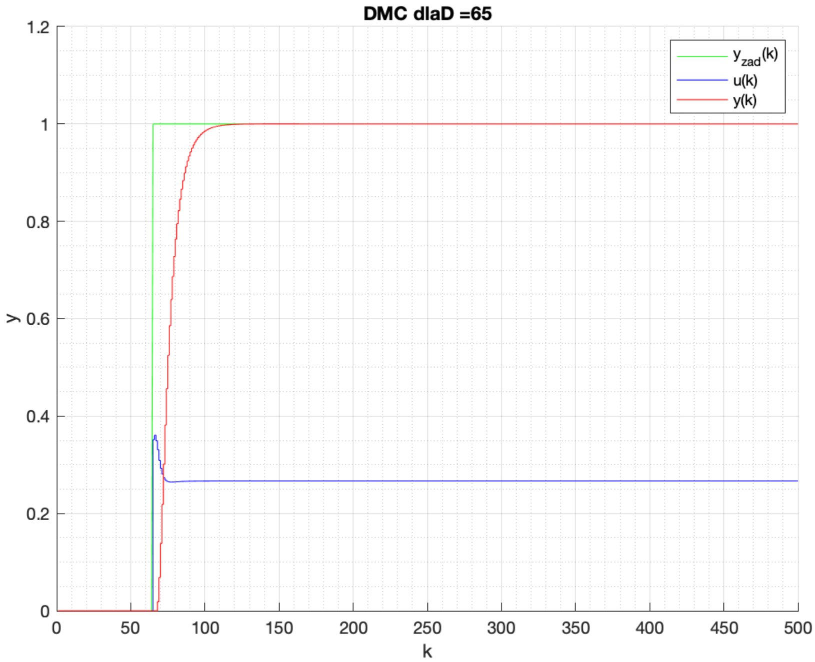
Rysunek 8: Wyznaczone parametry Jy dla różnych wartości D



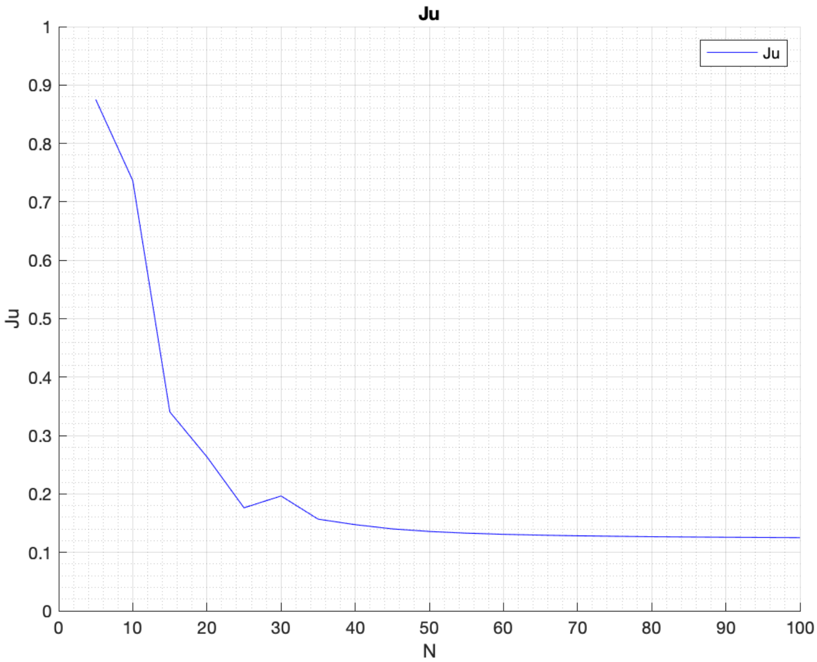
Rysunek 9: Różne wartości D - przebiegi DMC



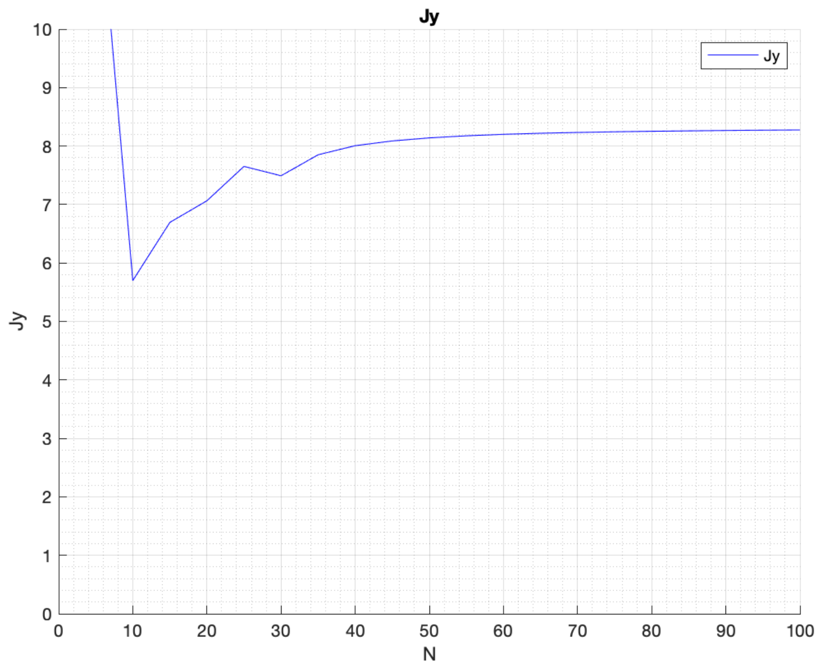
Rysunek 10: Różne wartości D - przebiegi DMC



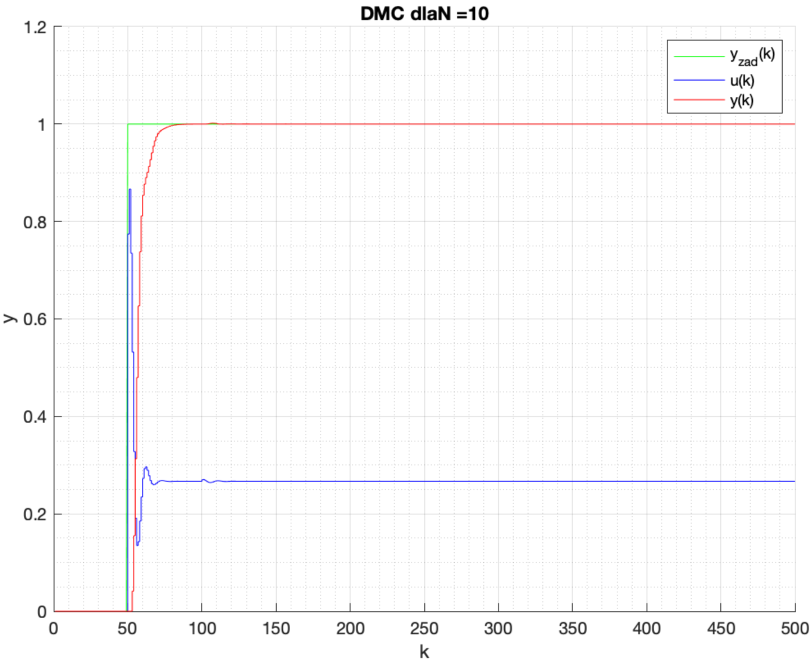
Rysunek 11: Różne wartości D - przebiegi DMC



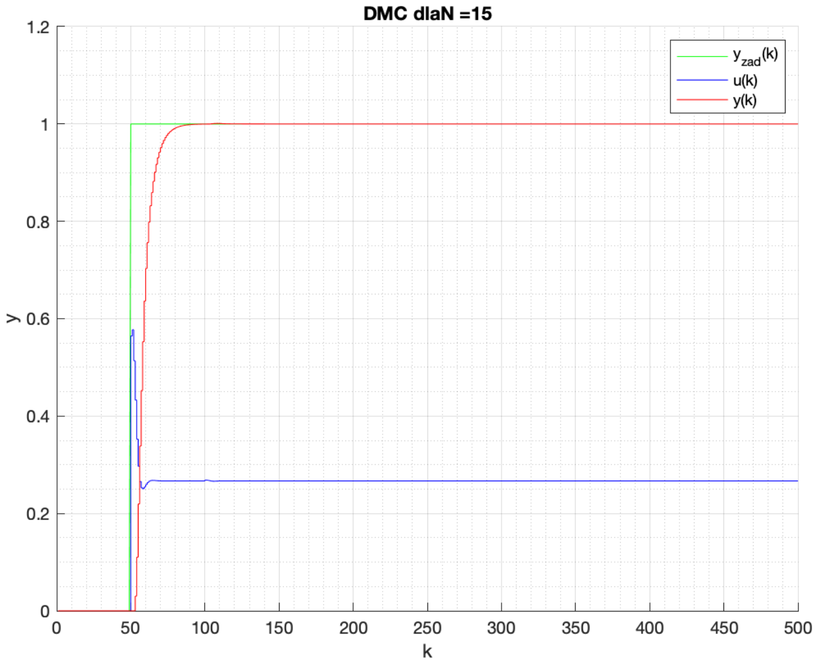
Rysunek 12: Wyznaczone parametry Ju dla różnych wartości N



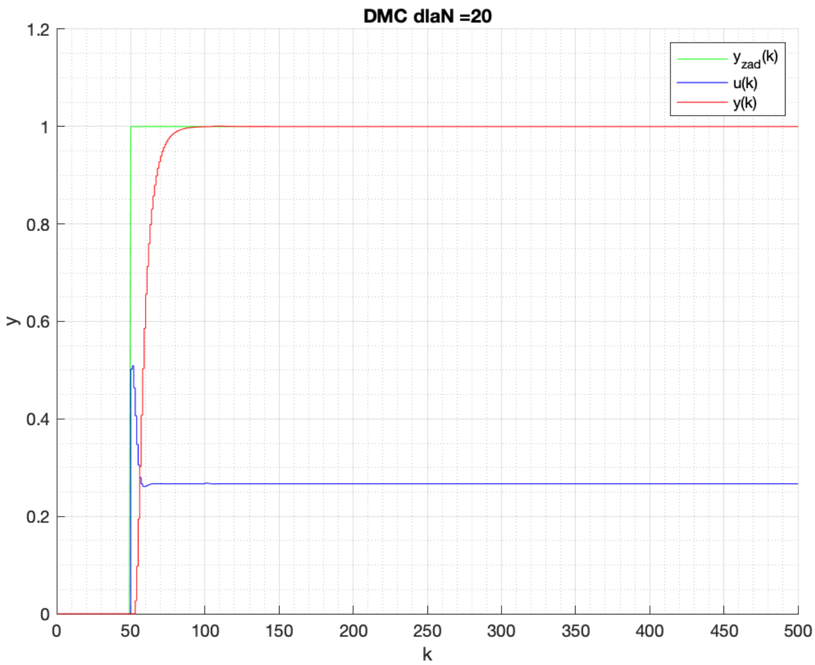
Rysunek 13: Wyznaczone parametry Jy dla różnych wartości N



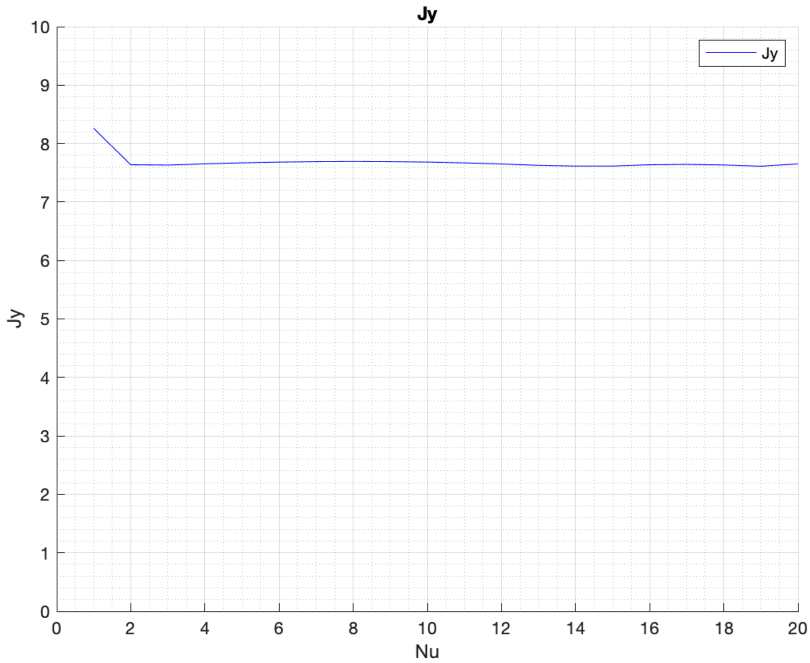
Rysunek 14: Różne wartości N - przebiegi DMC



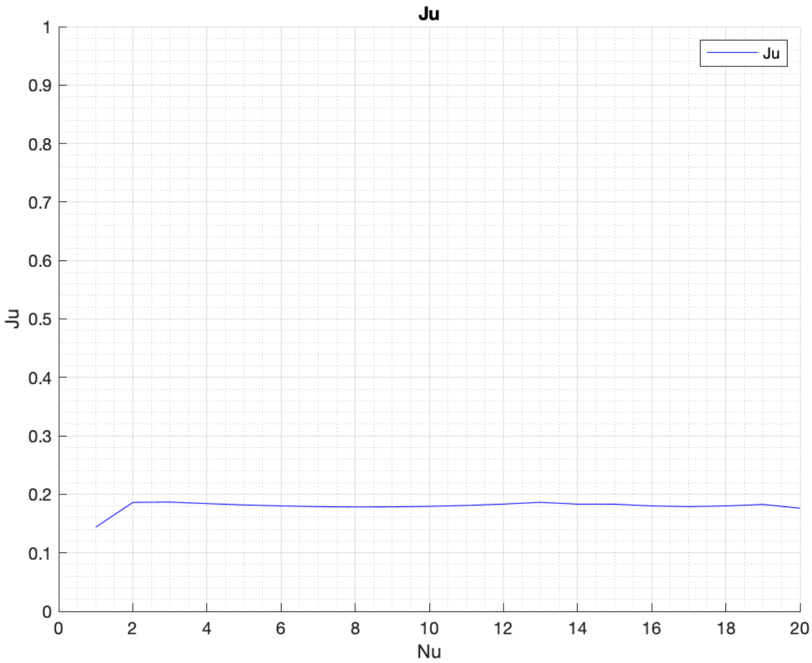
Rysunek 15: Różne wartości N - przebiegi DMC



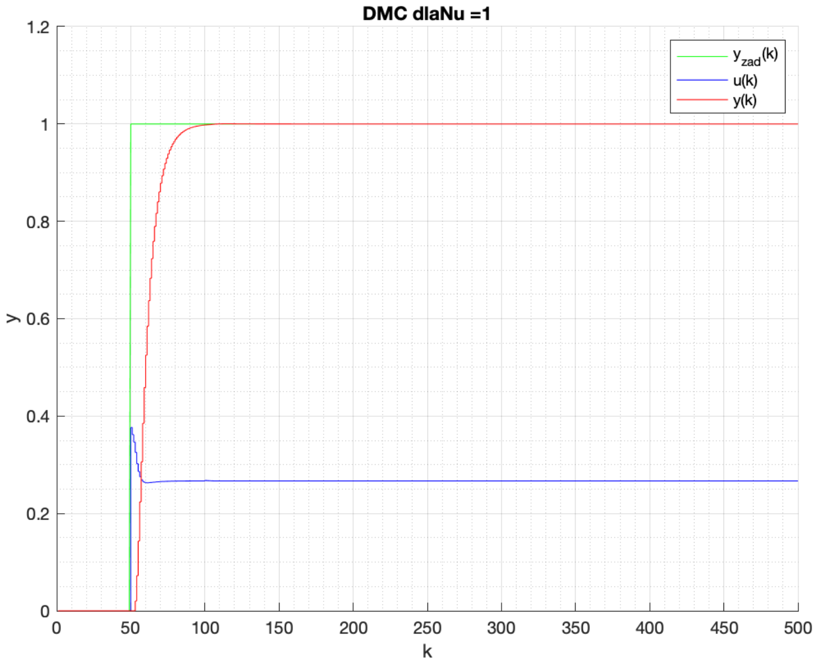
Rysunek 16: Różne wartości N - przebiegi DMC



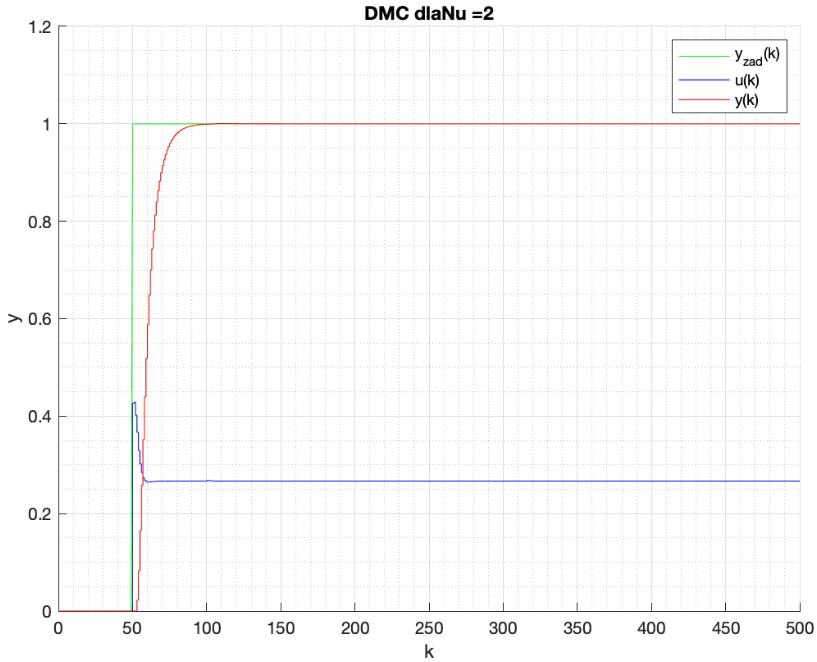
Rysunek 17: Wyznaczone parametry Jy dla różnych wartości Nu



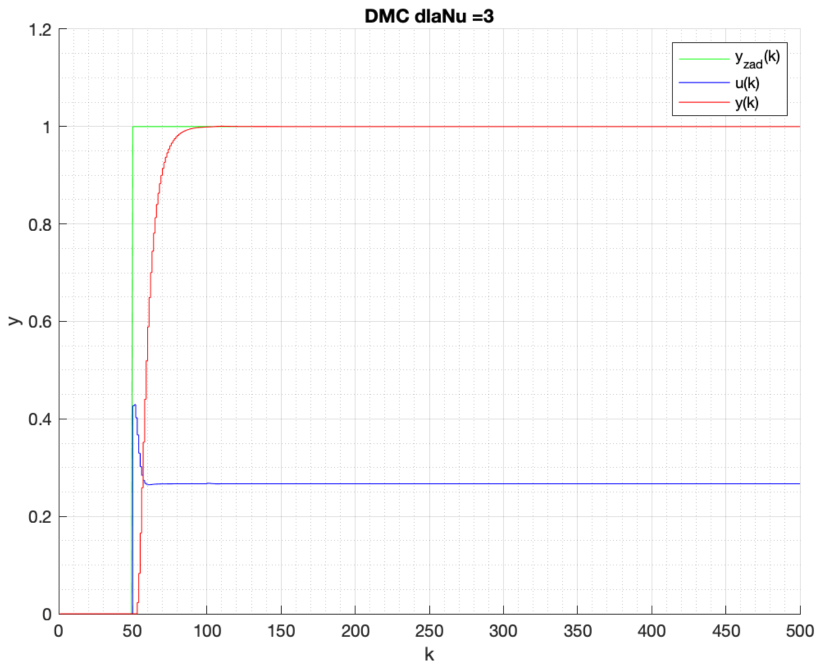
Rysunek 18: Wyznaczone parametry Ju dla różnych wartości Nu



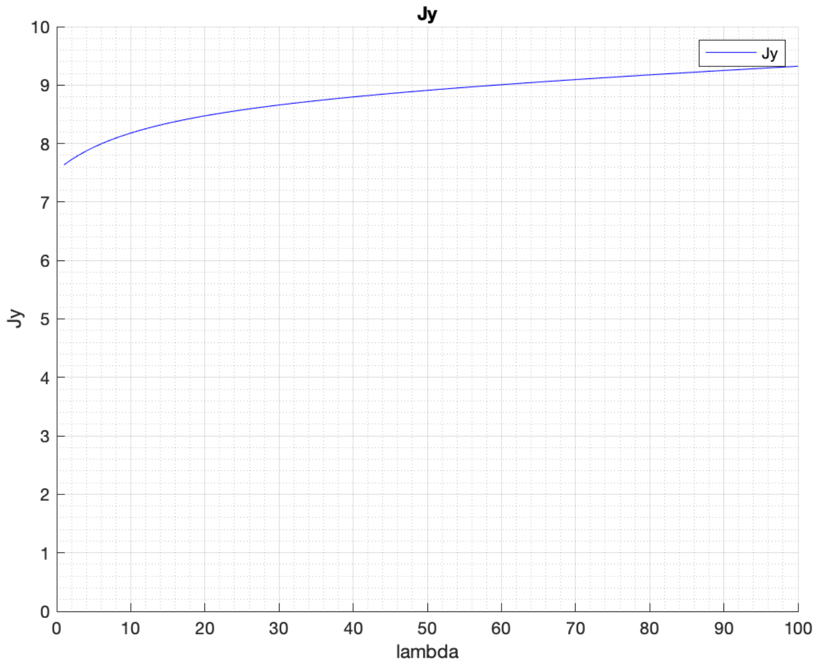
Rysunek 19: Różne wartości Nu - przebiegi DMC



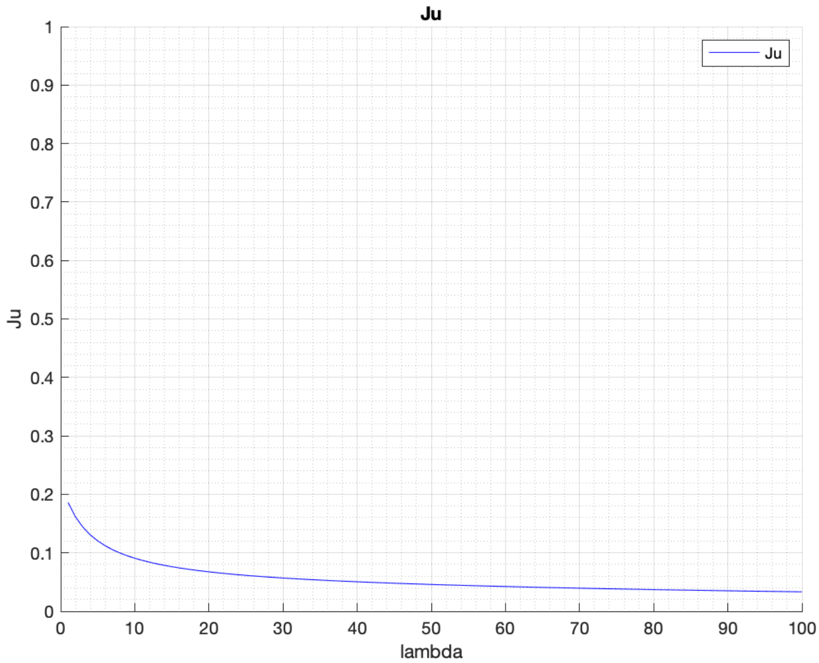
Rysunek 20: Różne wartości Nu - przebiegi DMC



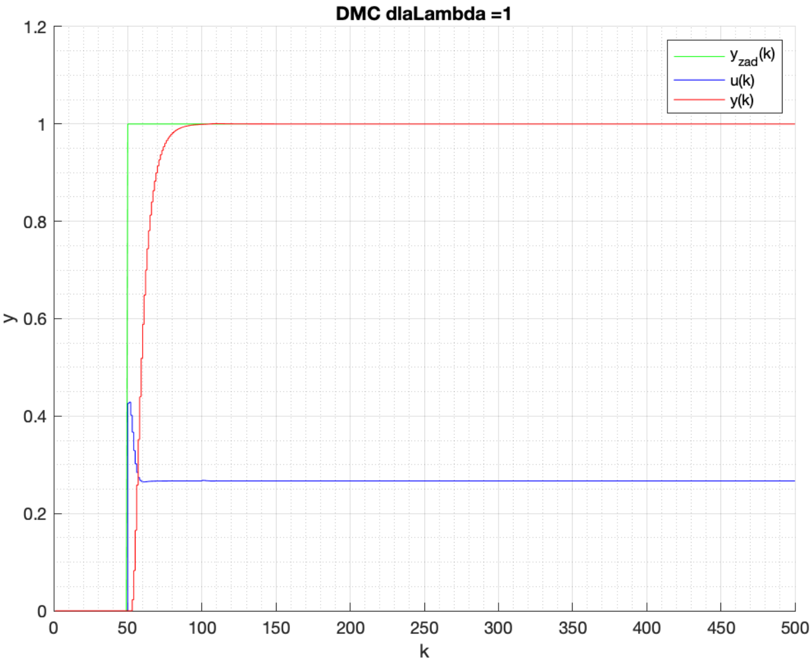
Rysunek 21: Różne wartości Nu - przebiegi DMC



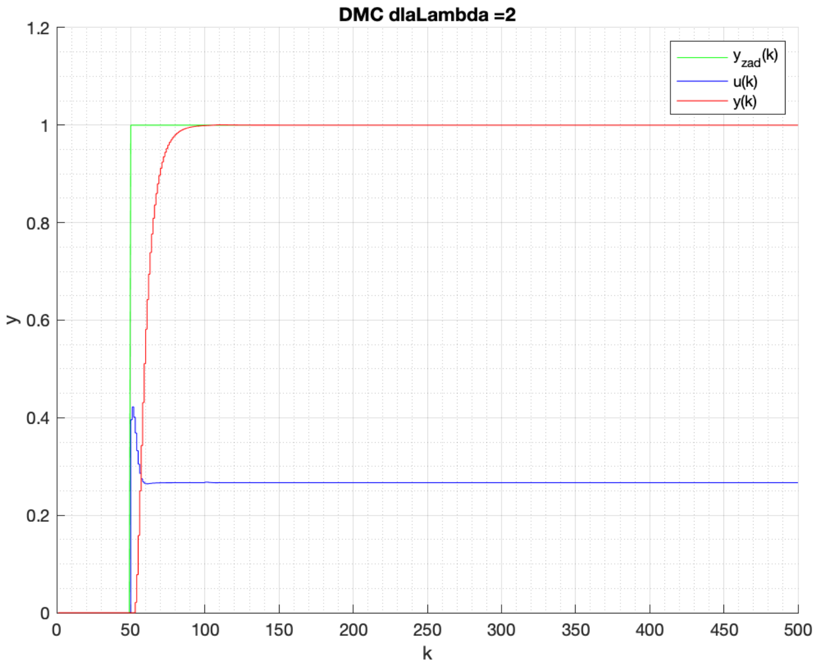
Rysunek 22: Wyznaczone parametry Jy dla różnych wartościλ



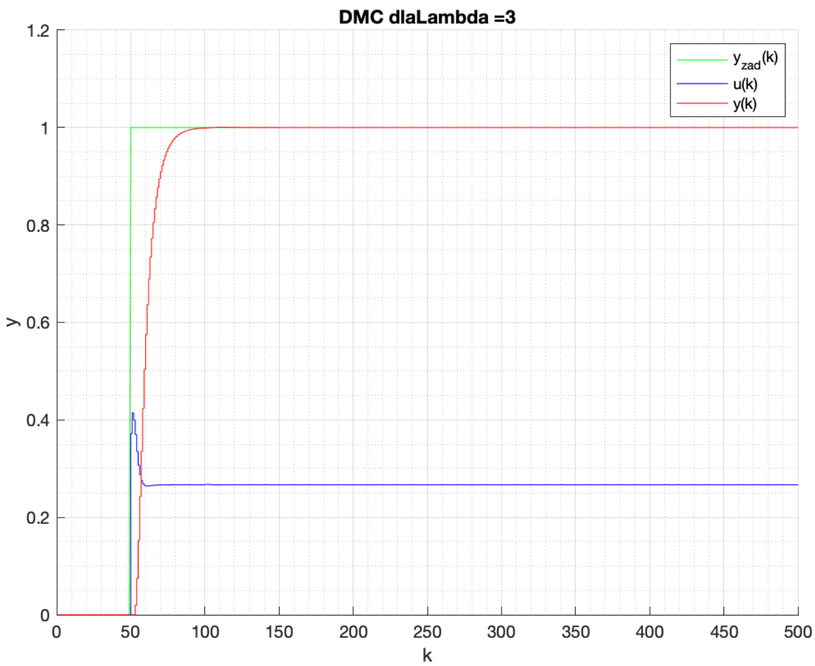
Rysunek 23: Wyznaczone parametry Ju dla różnych wartościλ



Rysunek 24: Różne wartości lambda - przebiegi DMC



Rysunek 25: Różne wartości lambda - przebiegi DMC



Rysunek 26: Różne wartości lambda - przebiegi DMC

Dla każdego parametru starałem się wybrać odpowiednią wartość.Dla D zdecydowałem się na wartość D = 50 jako że wtedy wszystkie błędy ostatecznie się wygładzają i dla kolejnych wartości symulacji wyniki są odpowiednie.W związku z błędem w obliczeniach wykresy J były źle obliczane i nie mogłem się nimi posiłkować, teraz dysponując tymi danymi rozważyłbym wybranie D = 40 czy nawet 35, jako że drobne zakłócenia nadal występujące mają ostatecznie dość mały wpływ na jakość sterowania. Stosując podobną logikę wybrałem wartości N = 15 jako że oferuje dobry balans pomiędzy parametrami Ju i Jy, dla Nu okazało się że już dla wartości 2 kolejna wersja modelu nie przynosi znaczącej poprawy, natomiast dla lambdy zdecydowałem że $\lambda = 3$ daje odpowiednio dokładne wyniki.

Kod Do wyznaczenia p. 4:

```
%%%%% %%%%% %%%%% %%%%% %%%%% %%%%% %%%%% %%%%% %%%%% %%%%% %%%%% %%%%%
% Projekt nr. 2 STP - Kajetan Kaczmarek
% Punkt 4 symulacja regulatora DMC,  wybrane par. : D - D = 50
%%%%% %%%%% %%%%% %%%%% %%%%% %%%%% %%%%% %%%%% %%%%% %%%%% %%%%% %%%%%
clear;
% Inicjalizacja
% Model
systems = P1();
sys = systems(:,:,4);

value = 'D';
lambda = 1;
Dmax = 200;
Nu = 20;
N=100;
kk = 500;
step = 5;
Ju = zeros(Dmax/step,2);
Jy = zeros(Dmax/step,2);
i=1;
for D = Dmax:-step:5
    [y,y_zad,u,Jy(i,1),Ju(i,1)] = DMCnoLimit(sys ,N,Nu,D,lambda,kk);
    Jy(i,2) = D;Ju(i,2) = D;
    DMC_Draw(kk, y,y_zad, u,D,value,'P4');
    i=i+1;
end
DrawJ(Jy,'D','Jy',10);
DrawJ(Ju,'D','Ju',1);
```

```
%%%%% %%%%% %%%%% %%%%% %%%%% %%%%% %%%%% %%%%% %%%%% %%%%% %%%%% %%%%%
% Projekt nr. 2 STP - Kajetan Kaczmarek
% Punkt 4 symulacja regulatora DMC, wybrane par. : D - D = 50, N = 15,
% Nu = 2, lambda = 3
%%%%% %%%%% %%%%% %%%%% %%%%% %%%%% %%%%% %%%%% %%%%% %%%%% %%%%% %%%%%
clear;
% Inicjalizacja
% Model
systems = P1();
sys = systems(:,:,4);
a=sys.Denominator;b=sys.Numerator;T= sys.Ts;
value = 'Lambda';
lambdaMax = 100;
D = 50;
Nu = 2;
N = 25;
kk = 500;
Ju = zeros(lambdaMax,2);
Jy = zeros(lambdaMax,2);
i=1;
for lambda = lambdaMax:-1:1
    [y,y_zad,u,Jy(i,1),Ju(i,1)] = DMCnoLimit(sys ,N,Nu,D,lambda,kk);
    Jy(i,2) = lambda;Ju(i,2) = lambda;

    DMC_Draw(kk, y,y_zad, u,lambda,value,'P4');
    i=i+1;
```

```
end
DrawJ(Jy,'lambda','Jy',10);
DrawJ(Ju,'lambda','Ju',1);
```

```
%%%%% %%%%% %%%%% %%%%% %%%%% %%%%% %%%%% %%%%% %%%%% %%%%% %%%%% %%%%%
% Projekt nr. 2 STP - Kajetan Kaczmarek
% Punkt 4 symulacja regulatora DMC,  wybrane par. : D - D = 50, N = 15
%%%%% %%%%% %%%%% %%%%% %%%%% %%%%% %%%%% %%%%% %%%%% %%%%% %%%%% %%%%%
clear;
% Inicjalizacja
% Model
systems = P1();
sys = systems(:,:,4);

value = 'N';
lambda = 1;
D = 50;
Nu = 20;
Nmax=100;
kk = 500;
step = 5;
Ju = zeros(Nmax/step,2);
Jy = zeros(Nmax/step,2);
i=1;
for N = Nmax:-step:5
    [y,y_zad,u,Jy(i,1),Ju(i,1)] = DMCnoLimit(sys ,N,Nu,D,lambda,kk);
    Jy(i,2) = N;Ju(i,2) = N;
    DMC_Draw(kk, y,y_zad, u,N,value,'P4');
    i=i+1;
end
DrawJ(Jy,'N','Jy',10);
DrawJ(Ju,'N','Ju',1);
```

```
%%%%% %%%%% %%%%% %%%%% %%%%% %%%%% %%%%% %%%%% %%%%% %%%%% %%%%% %%%%%
% Projekt nr. 2 STP - Kajetan Kaczmarek
% Punkt 4 symulacja regulatora DMC, wybrane par. : D - D = 50, N = 15,
% Nu = 2
%%%%% %%%%% %%%%% %%%%% %%%%% %%%%% %%%%% %%%%% %%%%% %%%%% %%%%% %%%%%
clear;
% Model
systems = P1();
sys = systems(:,:,4);
% Wartosc oceniana
value = 'Nu';

lambda = 1;
D = 50;
Numax = 20;
N=25;
kk = 500;
Ju = zeros(Numax,2);
Jy = zeros(Numax,2);
i=1;
for Nu = Numax:-1:1
    [y,y_zad,u,Jy(i,1),Ju(i,1)] = DMCnoLimit(sys ,N,Nu,D,lambda,kk);
    Jy(i,2) = Nu;Ju(i,2) = Nu;
    DMC_Draw(kk, y,y_zad, u,Nu,value,'P4');
    i=i+1;
end
DrawJ(Jy,'Nu','Jy',10);
DrawJ(Ju,'Nu','Ju',1);
```

Ogólny algorytm DMC bez ograniczeń :

```
function [y,y_zad,u,Jy,Ju] = DMCnoLimit(sys,N,Nu,D,lambda,kk)
% Model
a=sys.Denominator;b=sys.Numerator;T= sys.Ts;
skok = step(sys,1:T:N+Nu+D);


% Wyzerowanie do kolejnych obliczen
y_zad = zeros(1,kk);
y_zad(D:kk) = 1;
y=zeros(1,kk);
u=zeros(1,kk);
du = zeros(1,kk);

% Obliczanie macierzy predykcji
M_P = zeros(N,D-1);

```

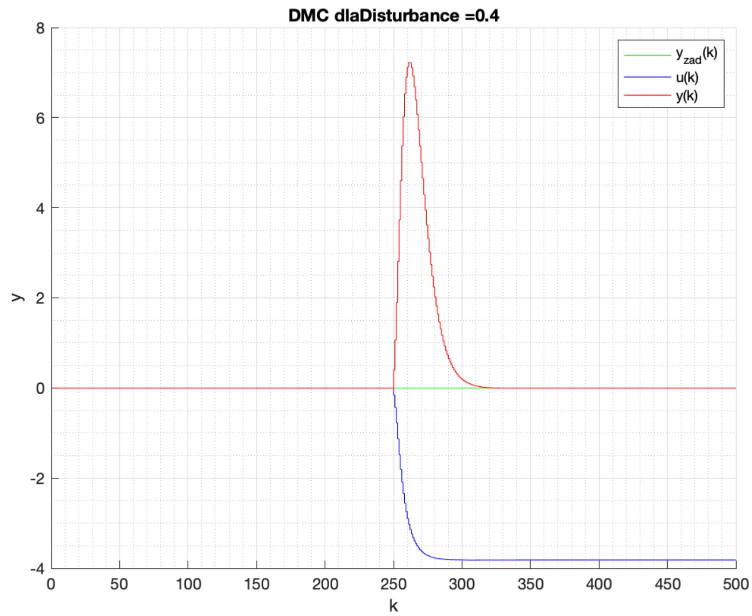
```
for j=1:(D-1)
    for i=1:N
        M_P(i,j) = skok(i+j) - skok(j);
    end
end

% Wyznaczenie macierzy dynamicznej
M_D = zeros(N, Nu);
for i=1:Nu
    for j=1:N
        if j >= i
            M_D(j+i-1,i) = skok(j);
        else
            M_D(j,i) = 0;
        end
    end
end



% Wyznaczenie macierzy K

K = (M_D'*M_D + lambda*eye(Nu))^-1 * M_D';

K1 = K(1,1:N);
ke = sum(K1);

for k=D:kk
    y(k)=b(2)*u(k-3)+b(3)*u(k-4)-a(2)*y(k-1)-a(3)*y(k-2);
    swob = 0;
    for j=1:(D-1)
        ku = K1*M_P(:,j);
        swob = swob + ku*du(k-j);
    end
    du(k) = ke*(y_zad(k) - y(k)) - swob;
    u(k) = u(k-1) + du(k);
end
[Jy,Ju] = getJ(y,y_zad,u,kk);
```

5. DMC z zakłóceniem

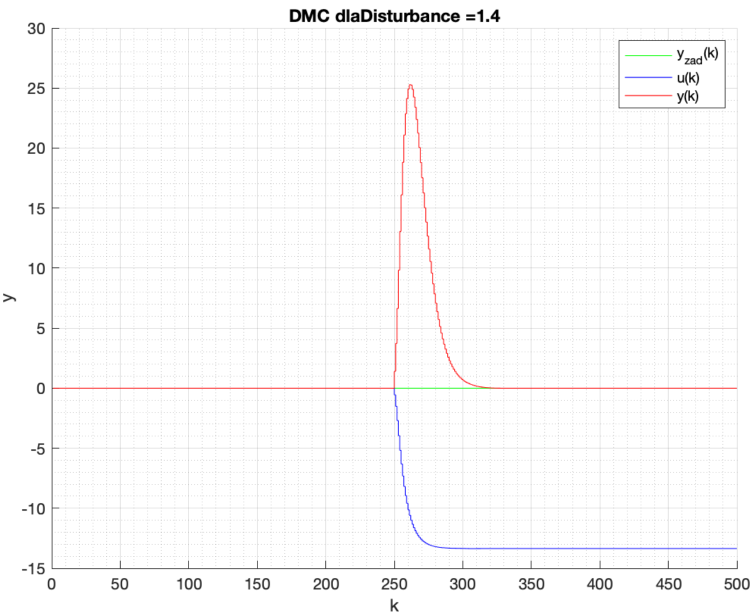
W wersji DMC z wprowadzonym zakłóceniem badałem zakres skoku zakłóceń od 0 do 5, i wyniki były bardzo powtarzalne.Kształt sygnału był taki sam dla wszystkich wartości, a amplituda sygnału wyjściowego wydaje się być proporcjonalna do skali zakłócenia

```
%%%%% %%%%% %%%%% %%%%% %%%%% %%%%% %%%%% %%%%% %%%%% %%%%% %%%%% %%%%%
% Projekt nr. 2 STP - Kajetan Kaczmarek
% Punkt 5 symulacja regulatora DMC z zakloceniem, wybrane par. : D = 50, N = 15,
% Nu = 2, lambda = 3
%%%%% %%%%% %%%%% %%%%% %%%%% %%%%% %%%%% %%%%% %%%%% %%%%% %%%%% %%%%% % Model
systems = P1();
sys = systems(:,:,4);
a=sys.Denominator;b=sys.Numerator;T= sys.Ts;
value = 'Disturbance';
lambda = 2;
D = 50;
Nu = 2;
N = 25;
a=sys.Denominator;b=sys.Numerator;T= sys.Ts;
skok = step(sys,1:T:N+Nu+D);

for disturbance_amp = 0.1:0.1:5
    % Wyzerowanie do kolejnych obliczen
    y_zad = zeros(1,kk);
    disturbance = zeros(1,kk);
    disturbance(kk/2:end) = disturbance_amp;
    y=zeros(1,kk);
    u=zeros(1,kk);
    du = zeros(1,kk);

    % Obliczanie macierzy predykcji
    M_P = zeros(N,D-1);

    for j=1:(D-1)
        for i=1:N
            M_P(i,j) = skok(i+j) - skok(j);
        end
    end

    % Wyznaczenie macierzy dynamicznej
    M_D = zeros(N, Nu);
    for i=1:Nu
        for j=1:N
            if j >= i
                M_D(j+i-1,i) = skok(j);
            else
                M_D(j,i) = 0;
            end
        end
    end



    % Wyznaczenie macierzy K

    K = (M_D'*M_D + lambda*eye(Nu))^-1 * M_D';

    K1 = K(1,1:N);
    ke = sum(K1);

    for k=D:kk
        y(k)=b(2)*u(k-3)+b(3)*u(k-4)-a(2)*y(k-1)-a(3)*y(k-2) + disturbance(k);
        swob = 0;
        for j=1:(D-1)
            ku = K1*M_P(:,j);
            swob = swob + ku*du(k-j);
        end
        du(k) = ke*(y_zad(k) - y(k)) - swob;
        u(k) = u(k-1) + du(k);
    end
    DMC_Draw(kk, y,y_zad, u,disturbance_amp,value,"P4");
end
```

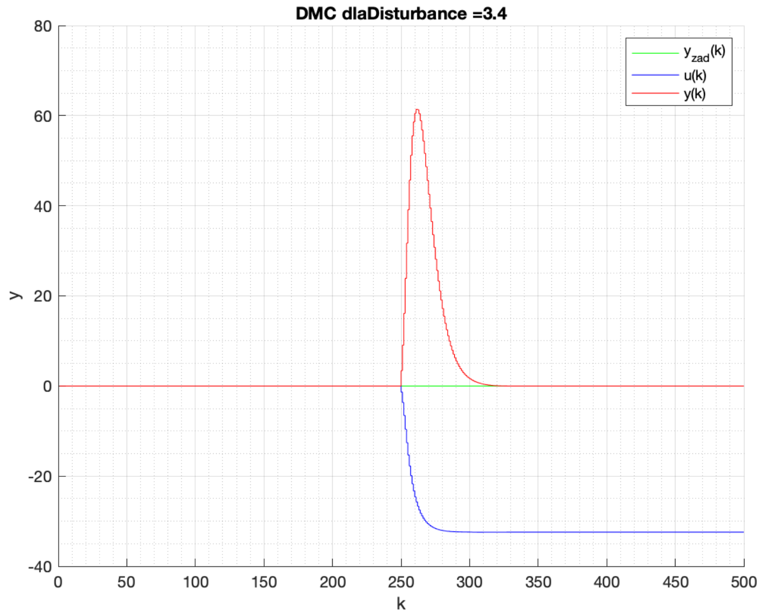
Przykładowe wykresy :

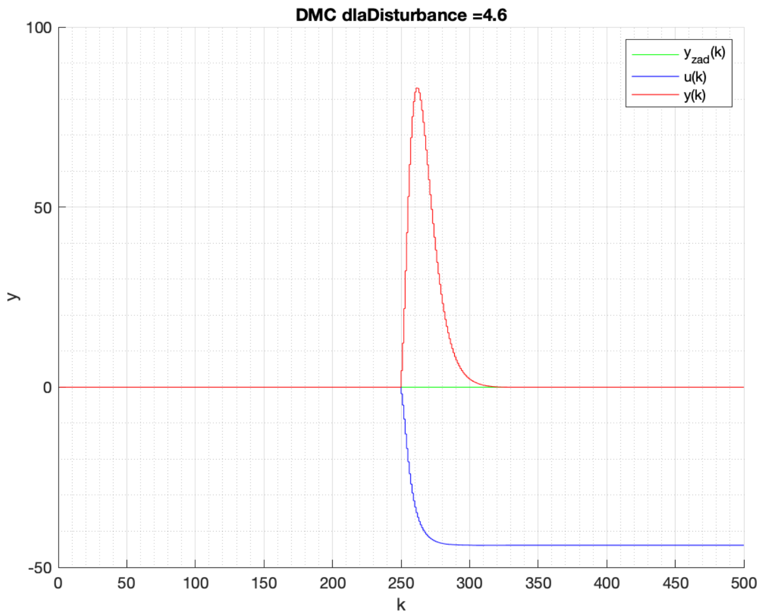


DMC dlaDisturbance =0.4
$y_{zad}(k)$
u(k)
y(k)
y
k



DMC dlaDisturbance =1.4
$y_{zad}(k)$
u(k)
y(k)
y
k


DMC dlaDisturbance =3.4
y_zad(k)
u(k)
y(k)
y
k
80
60
40
20
0
-20
-40
0
50
100
150
200
250
300
350
400
450
500


DMC dlaDisturbance =4.6
y_zad(k)
u(k)
y(k)
y
k
100
50
0
-50
0
50
100
150
200
250
300
350
400
450
500

6. Ograniczenia w symulacji DMC

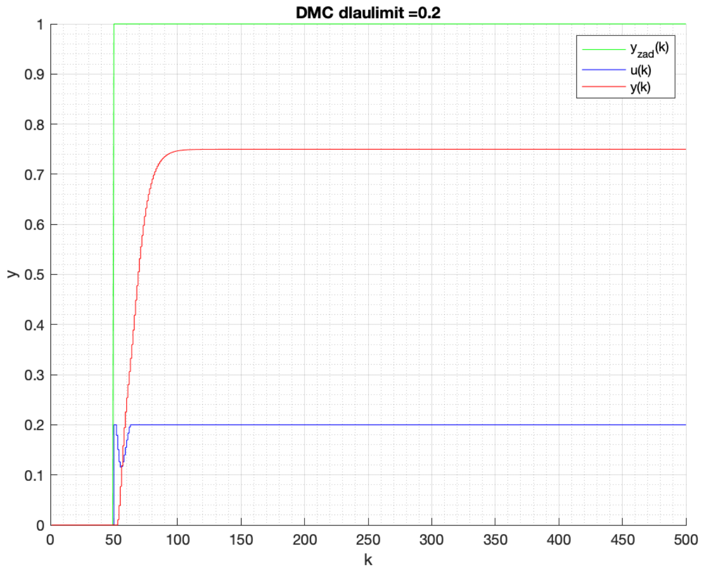
- Ograniczenie wartości U

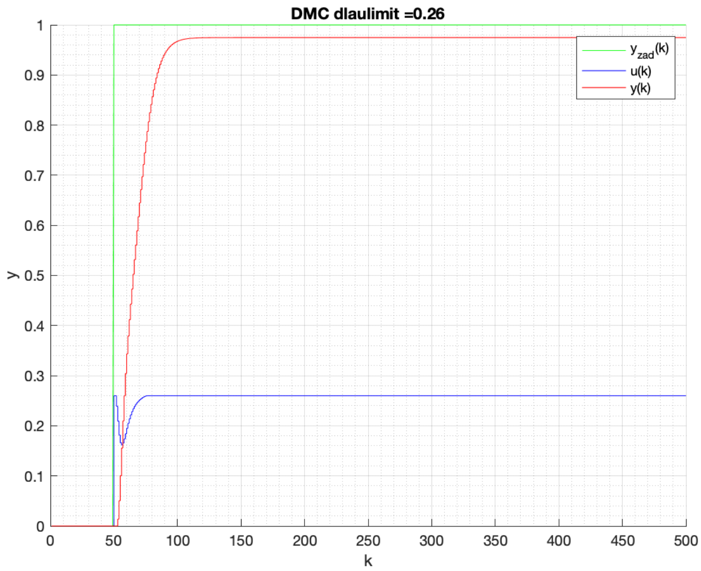
  Dla ograniczeń sygnału sterującego wybrałem wartość optymalną umax = 0.27.Dalsze zwiększanie tej wartości nie ma dużego wpływu na regulacje, za to zmniejszanie szybko prowadzi do uniemożliwienia sensownej regulacji. Kod :

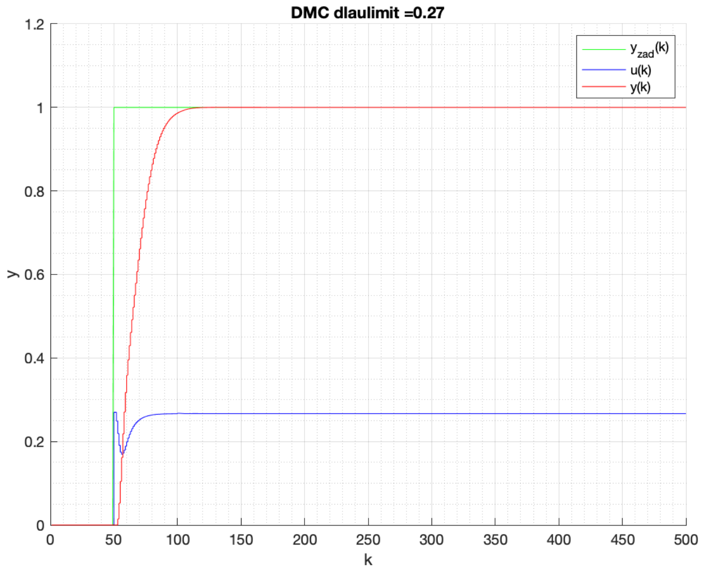
```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Projekt nr. 2 STP - Kajetan Kaczmarek
% Punkt 6 symulacja regulatora DMC, wybrane par. : D - D = 50, N = 15,
% Nu = 2
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Model
systems = P1();
sys = systems(:,:,4);
a=sys.Denominator;b=sys.Numerator;T= sys.Ts;
kk=500;
lambda = 2;
D = 50;
Nu = 2;
N = 25;
skok = step(sys,1:T:N+Nu+D);
dumax=1;
umax=0.3;
for ulimit = 0.2:0.01:umax
    % Wyzerowanie do kolejnych obliczen
    y_zad = zeros(1,kk);
    y_zad(D:kk) = 1;
    y=zeros(1,kk);
    u=zeros(1,kk);
    du = zeros(1,kk);

    % Obliczanie macierzy predykcji
    M_P = zeros(N,D-1);

    for j=1:(D-1)
        for i=1:N
            M_P(i,j) = skok(i+j) - skok(j);
        end
    end

    % Wyznaczenie macierzy dynamicznej
    M_D = zeros(N, Nu);
    for i=1:Nu
        for j=1:N
            if j >= i
                M_D(j+i-1,i) = skok(j);
            else
                M_D(j,i) = 0;
            end
        end
    end



    % Wyznaczenie macierzy K

    K = (M_D'*M_D + lambda*eye(Nu))^-1 * M_D';

    K1 = K(1,1:N);
    ke = sum(K1);

    for k=D:kk
        y(k)=b(2)*u(k-3)+b(3)*u(k-4)-a(2)*y(k-1)-a(3)*y(k-2);
        swob = 0;
        for j=1:(D-1)
            ku = K1*M_P(:,j);
            swob = swob + ku*du(k-j);
        end
        du(k) = ke*(y_zad(k) - y(k)) - swob;
        u(k) = u(k-1) + du(k);
        if(u(k) > ulimit)
            u(k) = ulimit;
        end
        if(u(k) < -ulimit)
```

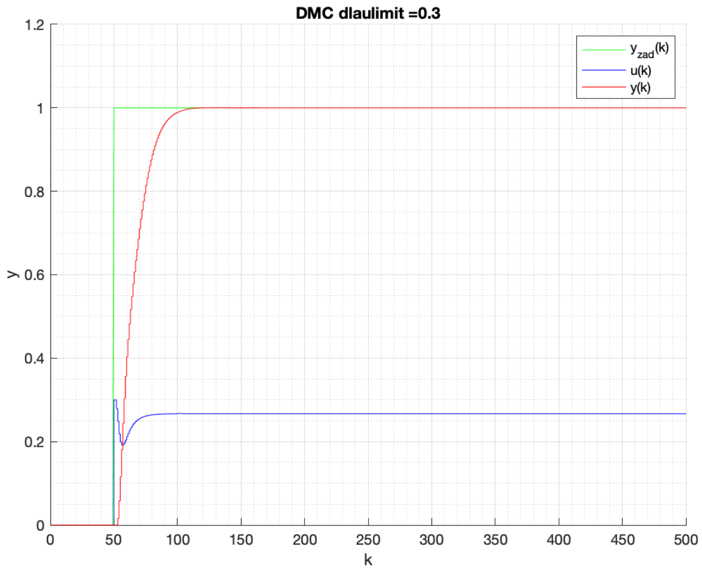
```
                u(k) = -ulimit;
            end
        end
        DMC_Draw(kk, y,y_zad, u,ulimit,"ulimit","P6");
    end
```

Wykresy przykładowe:







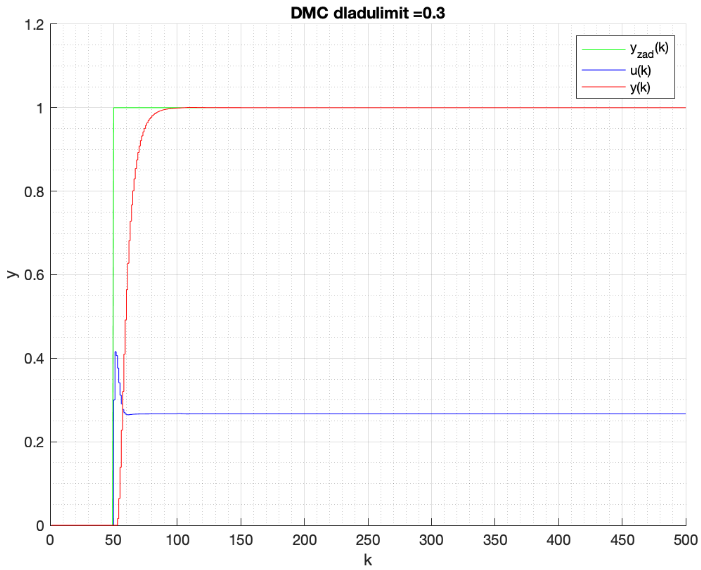


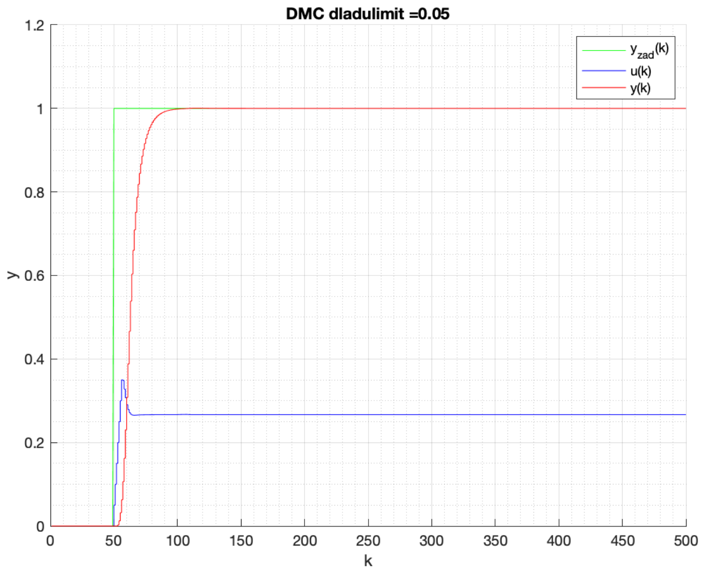
- Ograniczenie wartości dU
  Dla ograniczeń przyrostów wymierny wpływ zaczyna się wygładzać dla współczynnika około 0.05, a jako odpowiednie ograniczenie wybrałem 0.09 Kod :

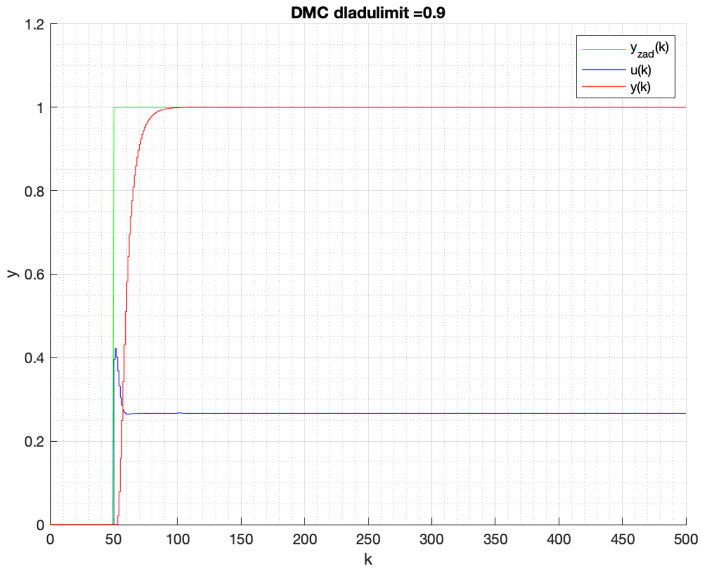
```
%%%%% %%%%% %%%%% %%%%% %%%%% %%%%% %%%%% %%%%% %%%%% %%%%% %%%%%
    %%%%%
% Projekt nr. 2 STP - Kajetan Kaczmarek
% Punkt 6 symulacja regulatora DMC, wybrane par. : D - D = 50, N = 15,
% Nu = 2
%%%%% %%%%% %%%%% %%%%% %%%%% %%%%% %%%%% %%%%% %%%%% %%%%% %%%%%
    %%%%%
% Model
```

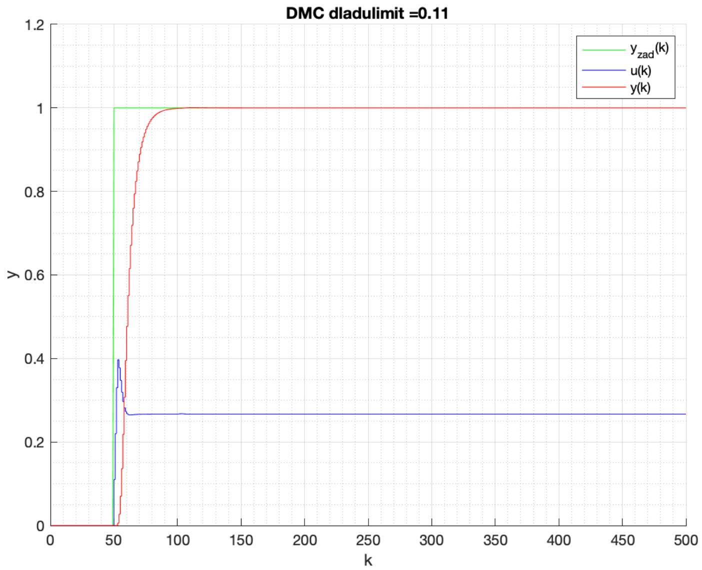
```
systems = P1();
sys = systems(:,:,4);
a=sys.Denominator;b=sys.Numerator;T= sys.Ts;
kk=500;
lambda = 2;
D = 50;
Nu = 2;
N = 25;
skok = step(sys,1:T:N+Nu+D);
dumax=1;
umax=1;
for ulimit = 0:0.01:dumax
    % Wyzerowanie do kolejnych obliczen
    y_zad = zeros(1,kk);
    y_zad(D:kk) = 1;
    y=zeros(1,kk);
    u=zeros(1,kk);
    du = zeros(1,kk);

    % Obliczanie macierzy predykcji
    M_P = zeros(N,D-1);

    for j=1:(D-1)
        for i=1:N
            M_P(i,j) = skok(i+j) - skok(j);
        end
    end

    % Wyznaczenie macierzy dynamicznej
    M_D = zeros(N, Nu);
    for i=1:Nu
        for j=1:N
            if j >= i
                M_D(j+i-1,i) = skok(j);
            else
                M_D(j,i) = 0;
            end
        end
    end



    % Wyznaczenie macierzy K

    K = (M_D'*M_D + lambda*eye(Nu))^-1 * M_D';

    K1 = K(1,1:N);
    ke = sum(K1);

    for k=D:kk
        y(k)=b(2)*u(k-3)+b(3)*u(k-4)-a(2)*y(k-1)-a(3)*y(k-2);
        swob = 0;
        for j=1:(D-1)
            ku = K1*M_P(:,j);
            swob = swob + ku*du(k-j);
        end
        du(k) = ke*(y_zad(k) - y(k)) - swob;
        if(du(k) > ulimit)
            du(k) = ulimit;
        end
        if(du(k) < -ulimit)
            du(k) = -ulimit;
        end
        u(k) = u(k-1) + du(k);
    end
    DMC_Draw(kk, y,y_zad, u,ulimit,"dulimit","P6");
end
```

Wykresy przykładowe:



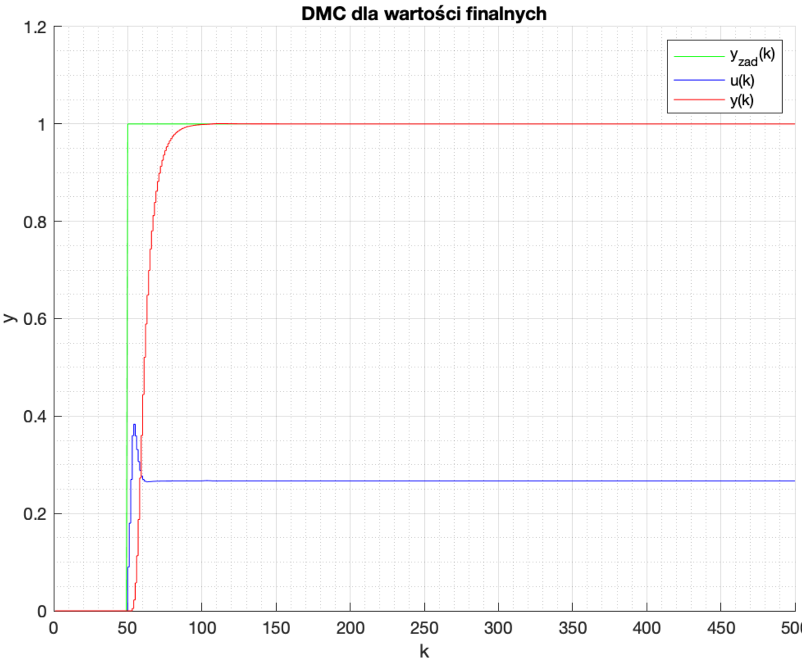






- Ograniczenie wartości U oraz dU

  Wybrane wartosci : dulimit = 0.27 , dlimit = 0.09
  Jak widać na wykresie sterowanie jest skuteczne dla wybranych wartości ograniczeń Wykres:



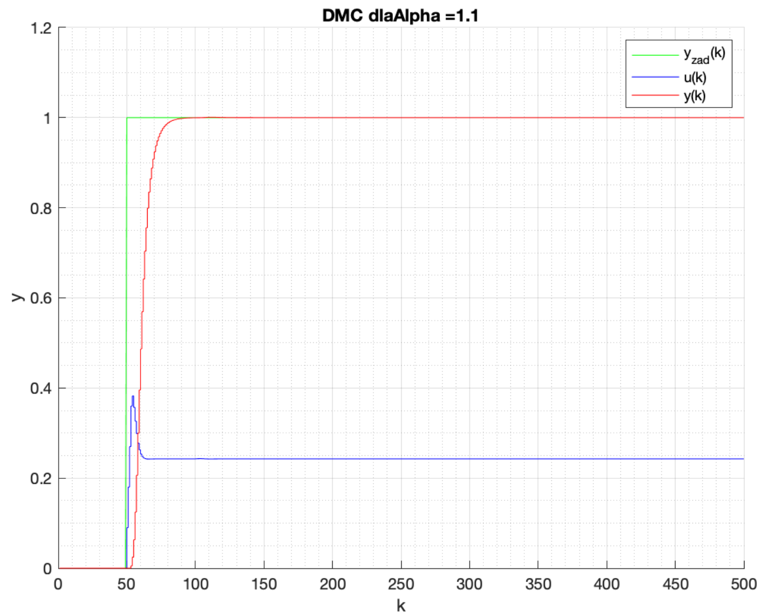
Rysunek 27: Sterowanie dla dulimit = 0,27 , dlimit = 0,09

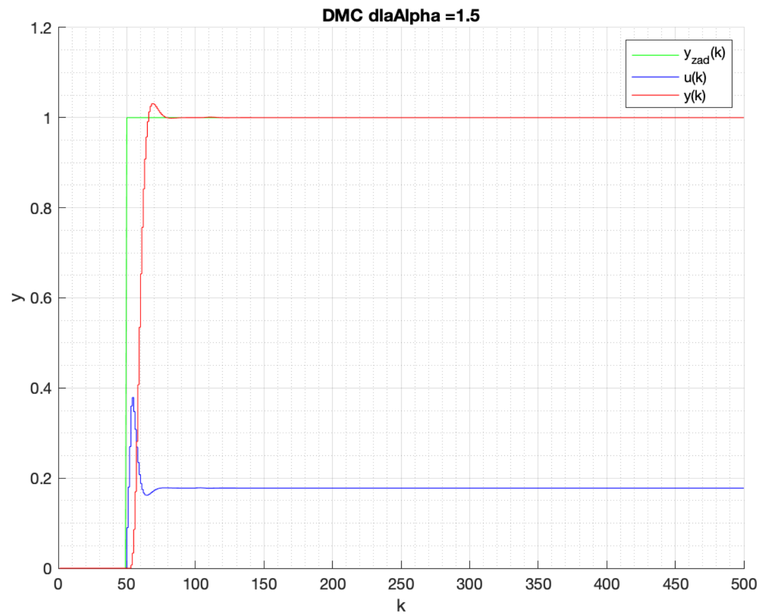
Kod :

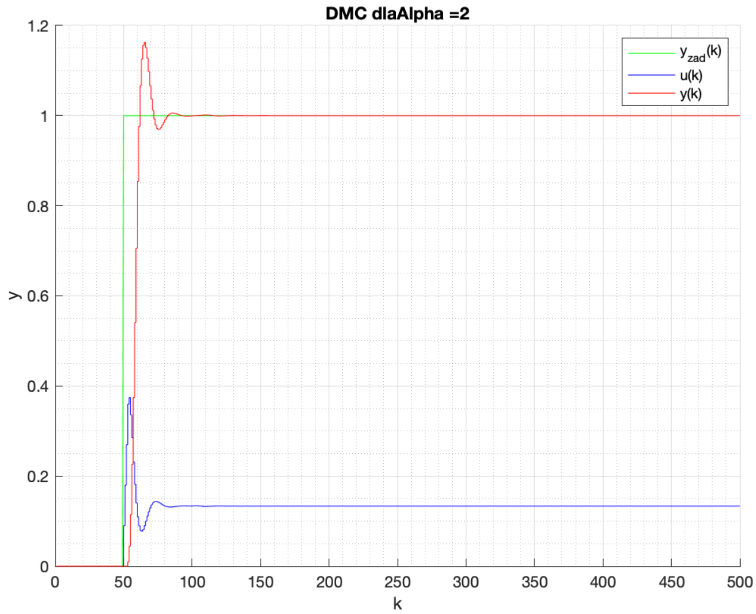
```
%%%%% %%%%% %%%%% %%%%% %%%%% %%%%% %%%%% %%%%% %%%%% %%%%% %%%%%
    %%%%%
% Projekt nr. 2 STP - Kajetan Kaczmarek
% Punkt 6 symulacja regulatora DMC, wybrane par. : D - D = 50, N = 15,
% Nu = 2
% Wybrane wartosci : dulimit = 0.27 , dlimit = 0.09
%%%%% %%%%% %%%%% %%%%% %%%%% %%%%% %%%%% %%%%% %%%%% %%%%% %%%%%
    %%%%%
%%%    Inicjalizacja
%%%%% %%%%% %%%%% %%%%% %%%%% %%%%% %%%%% %%%%% %%%%% %%%%% %%%%%
    %%%%%
% Model
systems = P1();sys = systems(:,:,4);
% Parametry
a=sys.Denominator;b=sys.Numerator;T= sys.Ts;
kk=500;lambda = 2; D = 50;Nu = 2;N = 25;
% Odpowiedz skokowwa systemu
skok = step(sys,1:T:N+Nu+D);
% Ograniczenia dla symulacji
dumax=0.09;umax=0.27;
% Macierze wynikowe
y_zad = zeros(1,kk);y_zad(D:kk) = 1;
y=zeros(1,kk);
u=zeros(1,kk);
du = zeros(1,kk);

%%%%% %%%%% %%%%% %%%%% %%%%% %%%%% %%%%% %%%%% %%%%% %%%%% %%%%%
    %%%%%
%%%    Obliczenia
%%%%% %%%%% %%%%% %%%%% %%%%% %%%%% %%%%% %%%%% %%%%% %%%%% %%%%%
    %%%%%
% Obliczanie macierzy predykcji
M_P = zeros(N,D-1);
for j=1:(D-1)
    for i=1:N
        M_P(i,j) = skok(i+j) - skok(j);
```

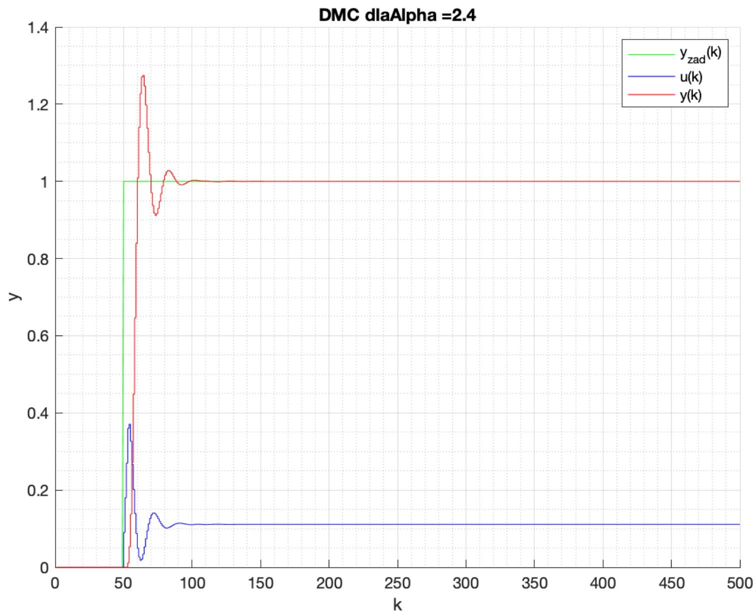
```
    end
end

% Wyznaczenie macierzy dynamicznej
M_D = zeros(N, Nu);
for i=1:Nu
    for j=1:N
        if j >= i
            M_D(j+i-1,i) = skok(j);
        else
            M_D(j,i) = 0;
        end
    end
end

% Wyznaczenie macierzy K
K = (M_D'*M_D + lambda*eye(Nu))^-1 * M_D';
K1 = K(1,1:N); ke = sum(K1);
for k=D:kk
    y(k)=b(2)*u(k-3)+b(3)*u(k-4)-a(2)*y(k-1)-a(3)*y(k-2);
    swob = 0;
    for j=1:(D-1)
        ku = K1*M_P(:,j);
        swob = swob + ku*du(k-j);
    end
    du(k) = ke*(y_zad(k) - y(k)) - swob;
    % Ograniczenia
    if(u(k) > umax)
        u(k) = umax;
    end
    if(u(k) < -umax)
        u(k) = -umax;
    end

    if(du(k) > dumax)
        du(k) = dumax;
    end
    if(du(k) < -dumax)
        du(k) = -dumax;
    end

    u(k) = u(k-1) + du(k);
end
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%    Drukuje wykres dla wybrancyh wartosci
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

f = figure('visible','off');
t = linspace(1,kk,kk);
hold on
plot(t, y_zad, 'g'); stairs(t, u, 'b'); stairs(t, y, 'r');
grid on
grid minor
legend('y_{zad}(k)', 'u(k)', 'y(k)');
xlabel('k');ylabel('y');
title('DMC dla warto¶ci finalnych');
hold off
print(f, sprintf(strcat('P6_3_DMC_Koncowe_.png')),'-dpng');
```

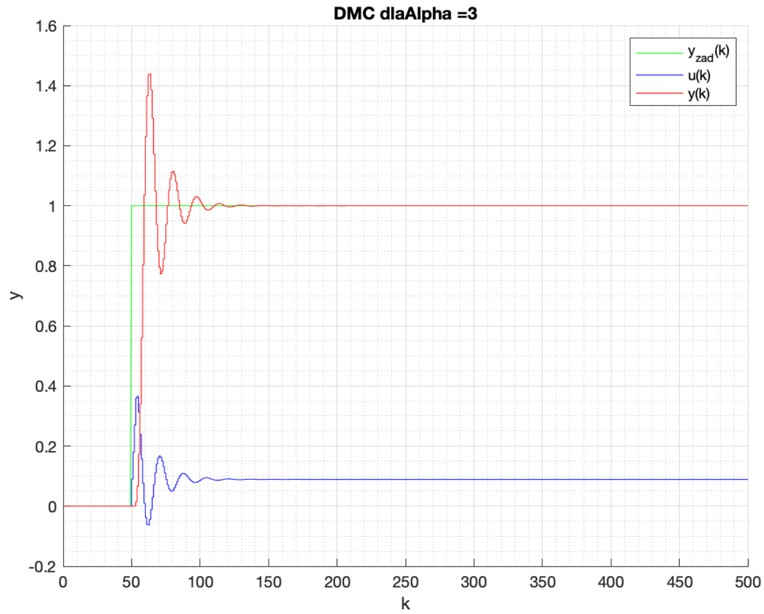
7. Zadanie dodatkowe
   Jak widać na wykresach dla wartości α na początku system DMC dobrze radzi sobie z kompensowaniem błędów. Dla $\alpha = 1,5$ zaczynamy obserwować przeregulowanie które szybko staje się widoczne.Dla wartości krytycznej $\alpha = 4.4$ system przestaje być zbieżny oraz spełniać swoją rolę

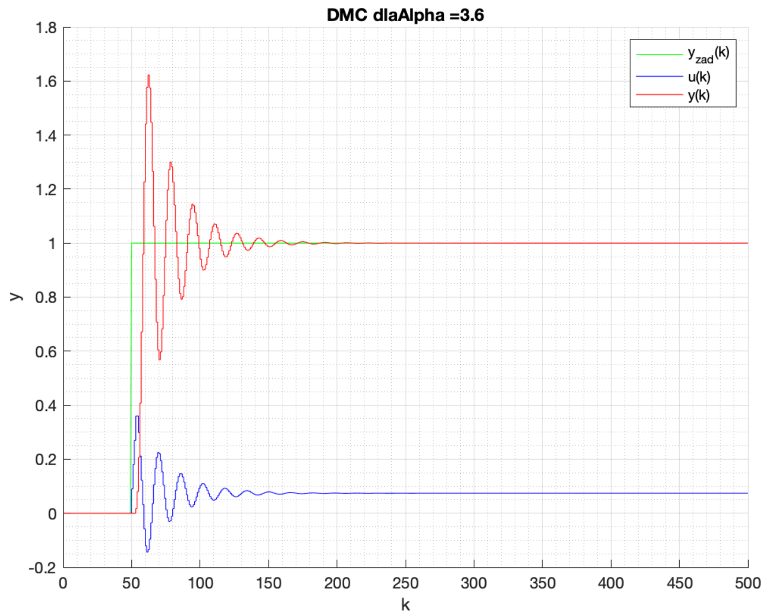

DMC dlaAlpha =1.1
y
k
$y_{zad}(k)$
u(k)
y(k)

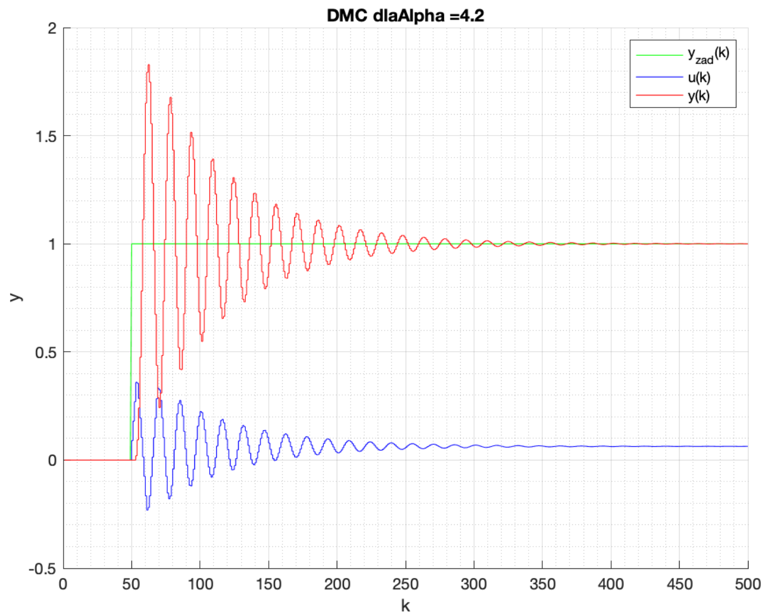

DMC dlaAlpha =1.5
y
k
$y_{zad}(k)$
u(k)
y(k)

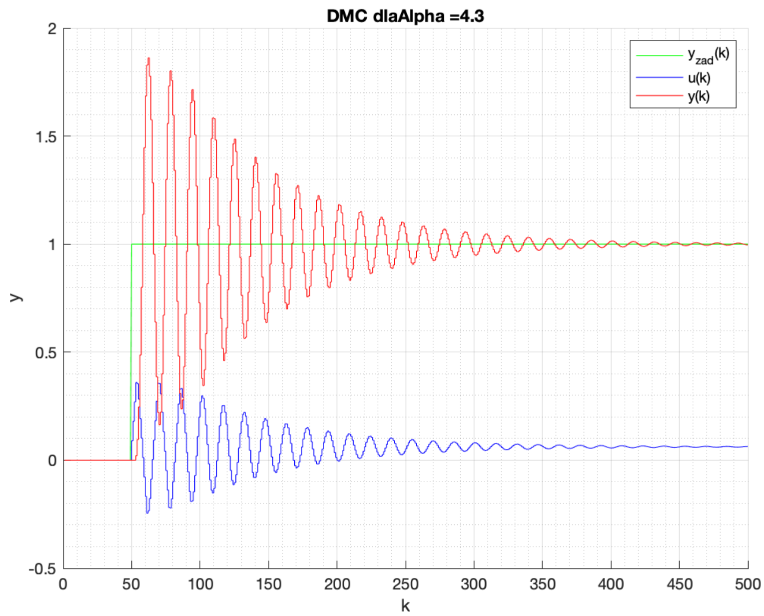

DMC dlaAlpha =2
$y_{zad}(k)$
u(k)
y(k)
y
k

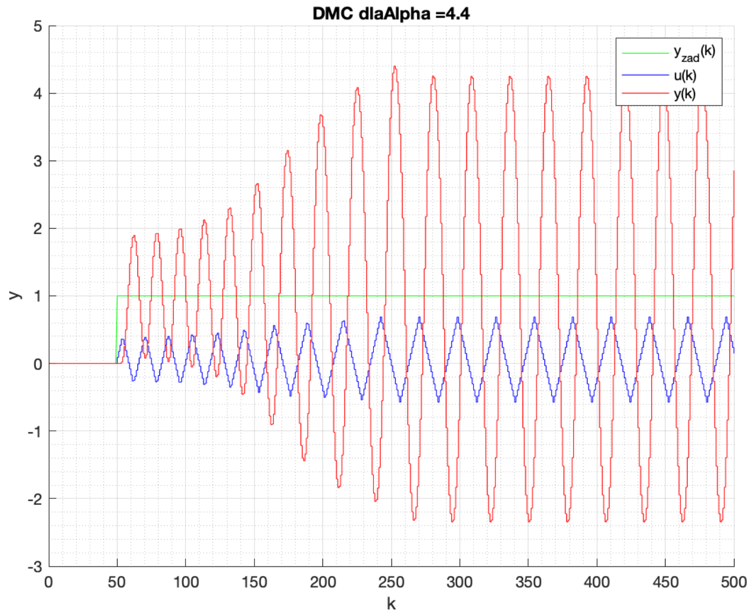

DMC dlaAlpha =2.4
$y_{zad}(k)$
u(k)
y(k)
y
k

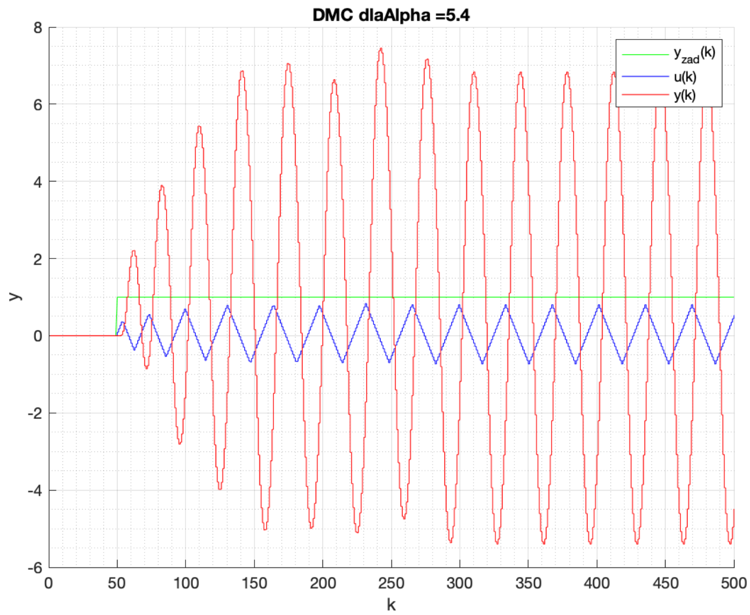

DMC dlaAlpha =3
y
k
$y_{zad}(k)$
u(k)
y(k)

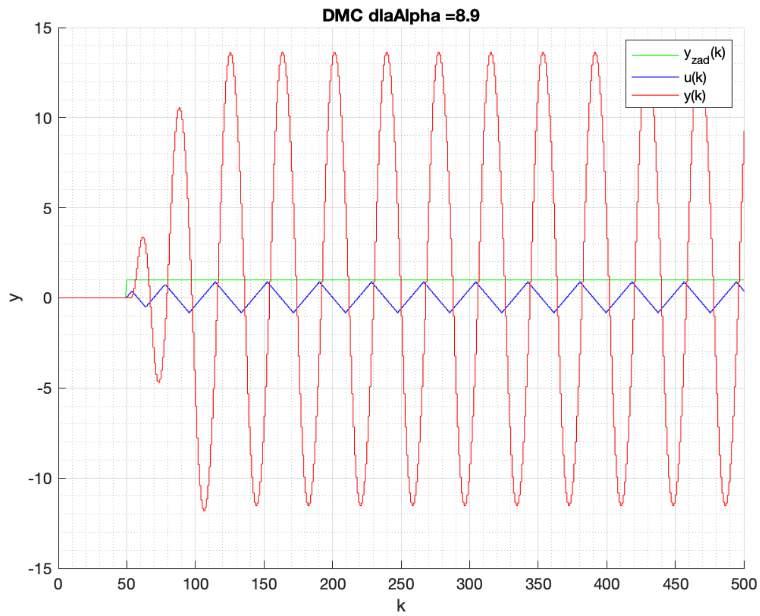

DMC dlaAlpha =3.6
y
k
$y_{zad}(k)$
u(k)
y(k)

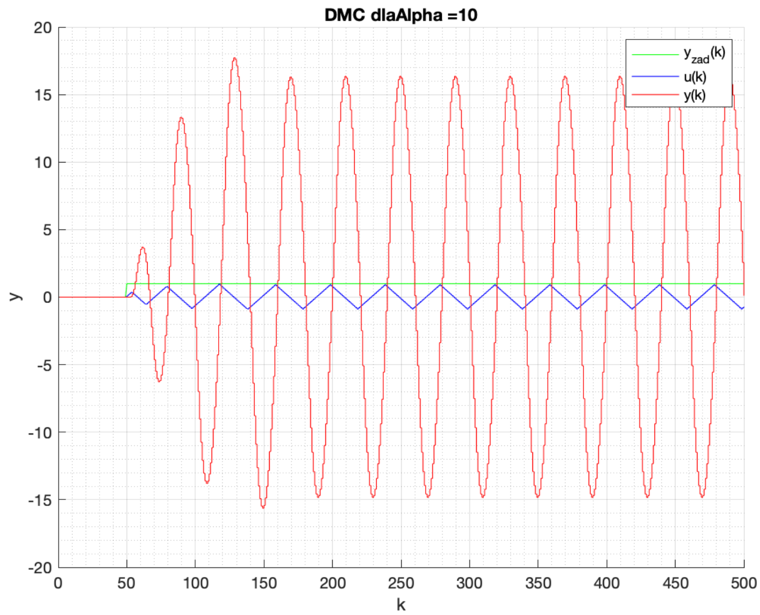

DMC dlaAlpha =4.2
y
k
$y_{zad}(k)$
u(k)
y(k)


DMC dlaAlpha =4.3
y
k
$y_{zad}(k)$
u(k)
y(k)

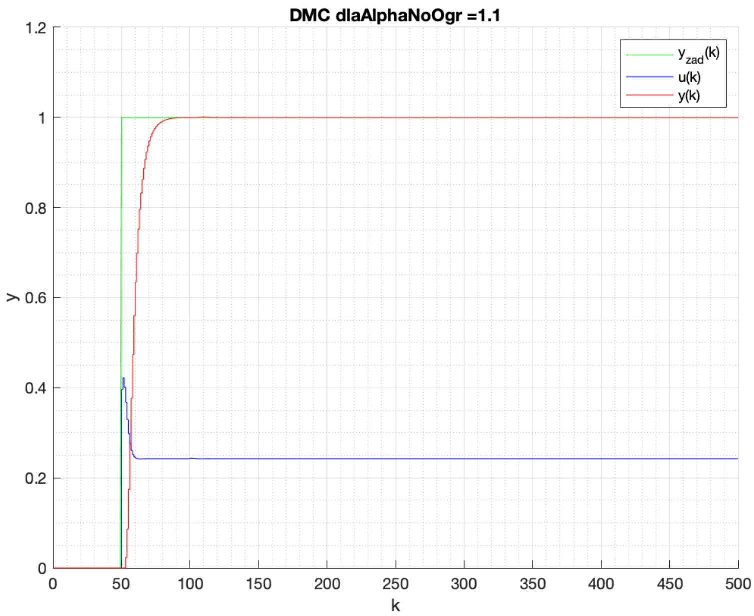

DMC dlaAlpha =4.4
y
k
y_zad(k)
u(k)
y(k)

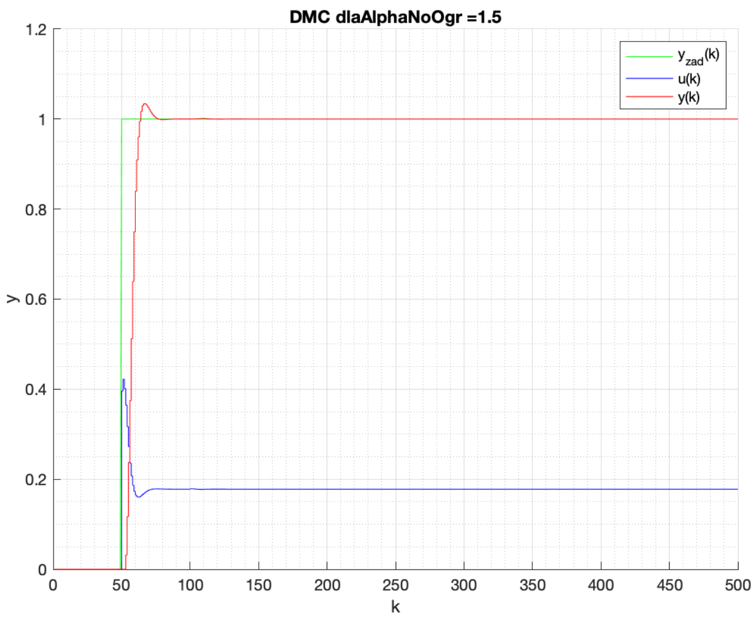

DMC dlaAlpha =5.4
y
k
y_zad(k)
u(k)
y(k)

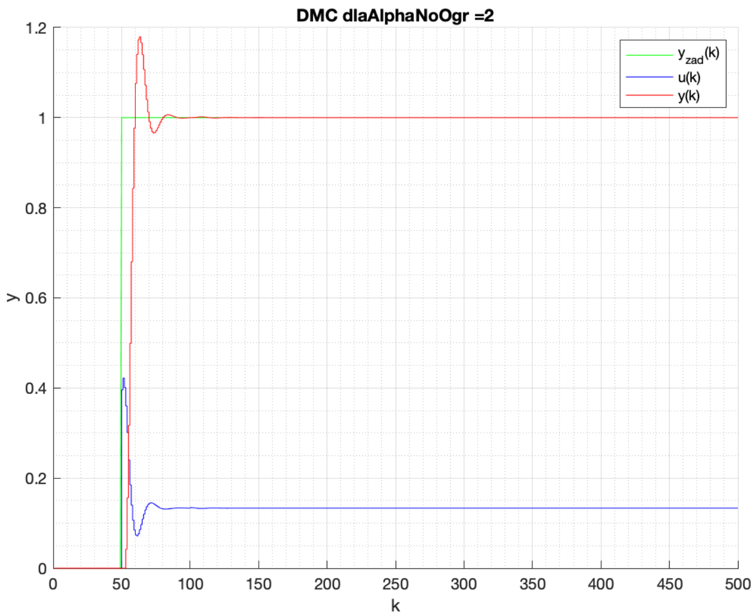

DMC dlaAlpha =8.9
y
k
$y_{zad}(k)$
u(k)
y(k)

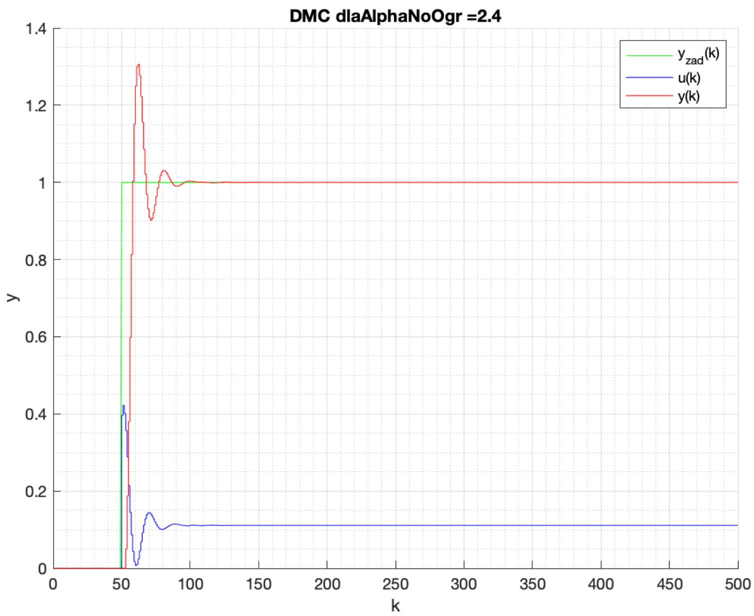

DMC dlaAlpha =10
y
k
$y_{zad}(k)$
u(k)
y(k)

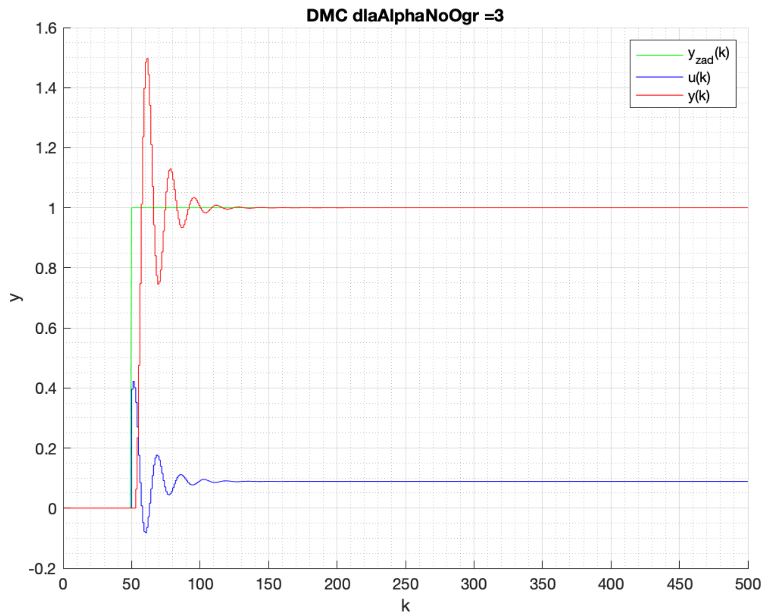
Przeprowadziłem także ten sam eksperyment ale dla DMC bez ograniczeń. System był odrobinę bardziej odporny na zmienność układu ale w ostateczności poddał się po wartości $\alpha = 4.8$

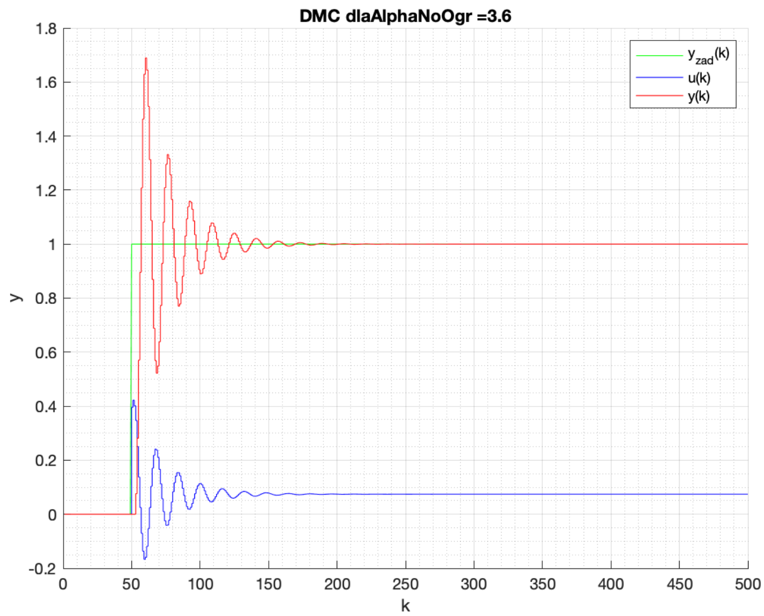


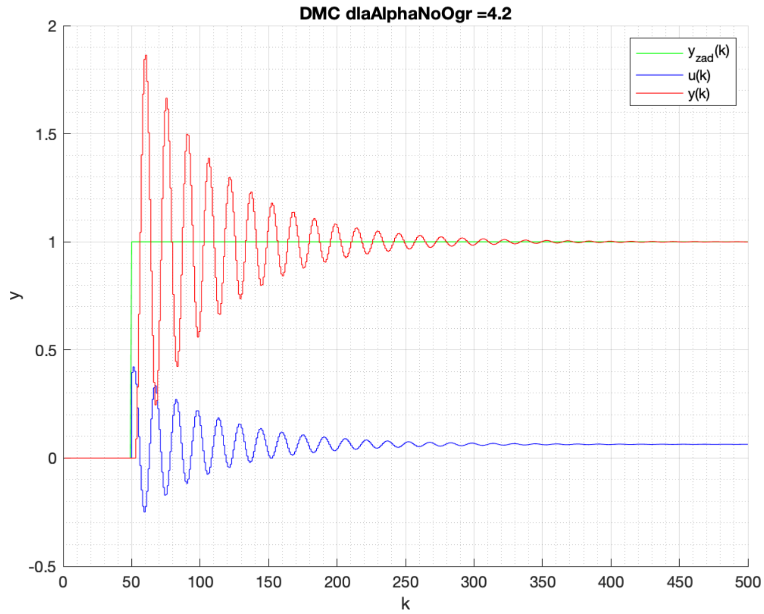


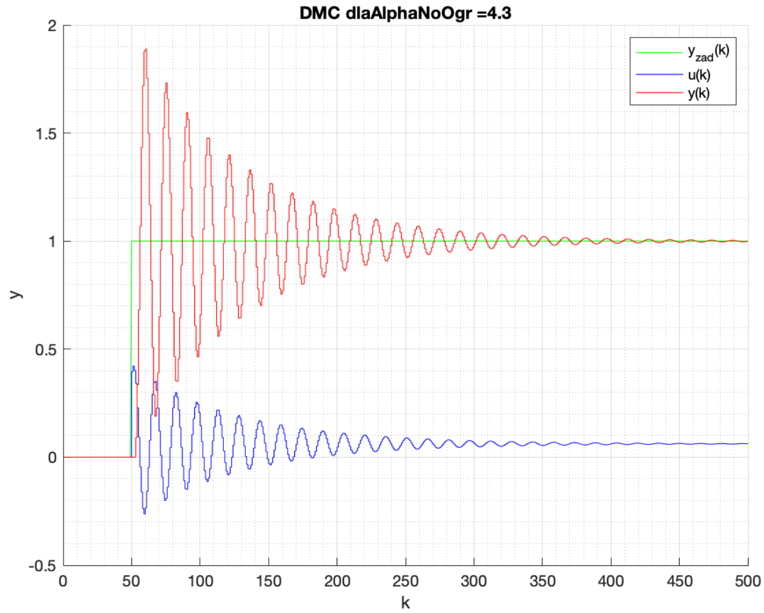

DMC dlaAlphaNoOgr =2
y
k
y_zad(k)
u(k)
y(k)

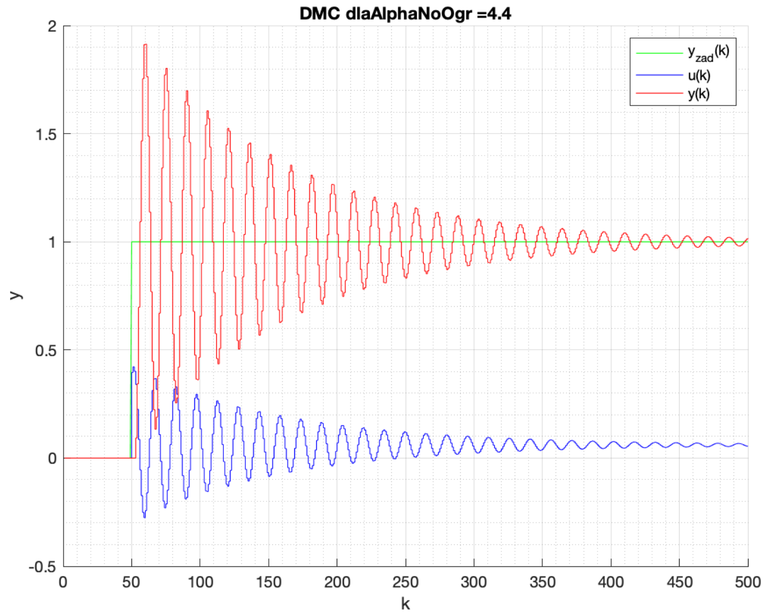

DMC dlaAlphaNoOgr =2.4
y
k
y_zad(k)
u(k)
y(k)

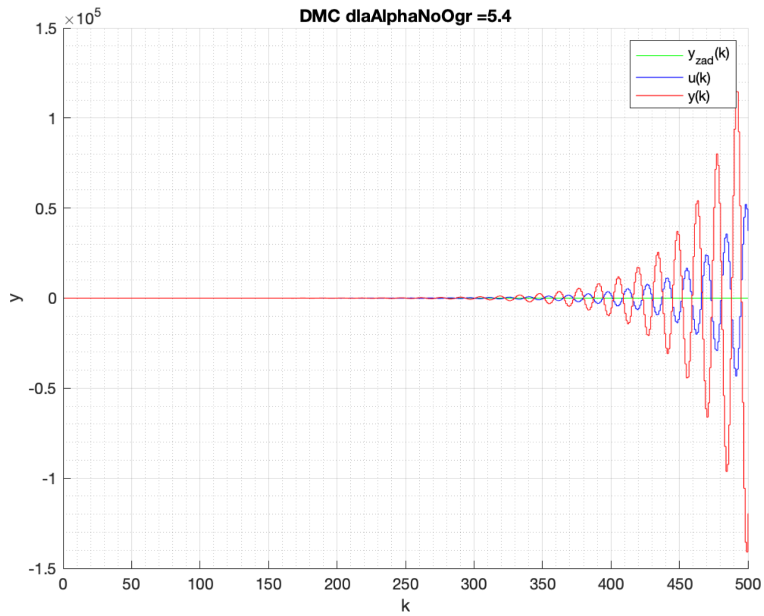

DMC dlaAlphaNoOgr =3
$y_{zad}(k)$
u(k)
y(k)
y
k

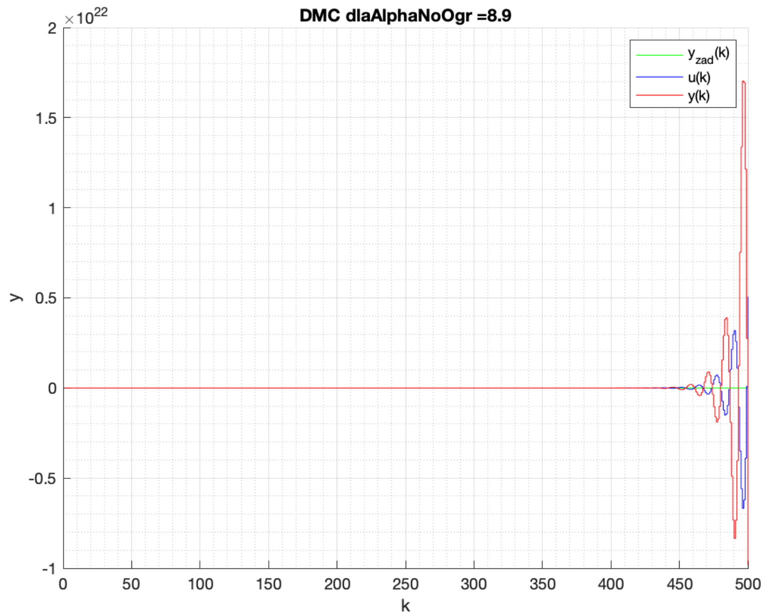

DMC dlaAlphaNoOgr =3.6
$y_{zad}(k)$
u(k)
y(k)
y
k

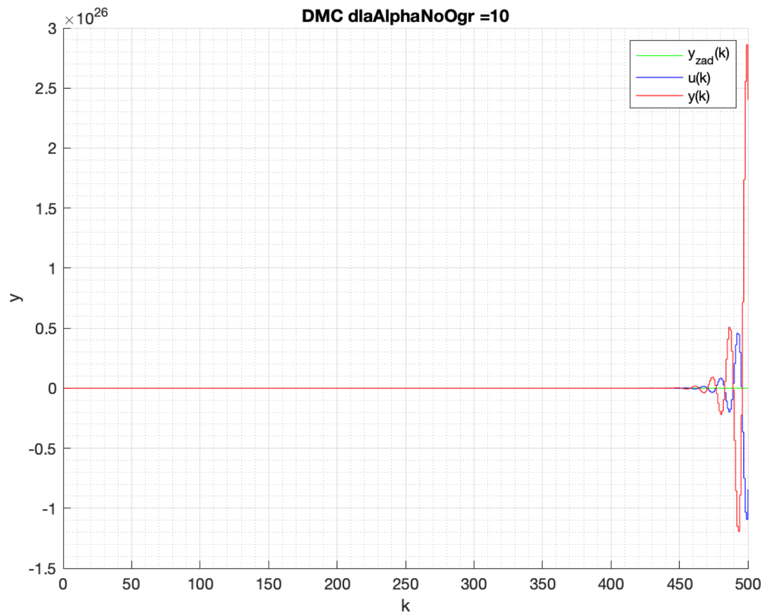

DMC dlaAlphaNoOgr =4.2
$y_{zad}(k)$
u(k)
y(k)
y
k


DMC dlaAlphaNoOgr =4.3
$y_{zad}(k)$
u(k)
y(k)
y
k


DMC dlaAlphaNoOgr =4.4
$y_{zad}(k)$
u(k)
y(k)
y
k
2
1.5
1
0.5
0
-0.5
0
50
100
150
200
250
300
350
400
450
500


DMC dlaAlphaNoOgr =5.4
$\times 10^5$
$y_{zad}(k)$
u(k)
y(k)
y
k
1.5
1
0.5
0
-0.5
-1
-1.5
0
50
100
150
200
250
300
350
400
450
500


DMC dlaAlphaNoOgr =8.9
×10^22
$y_{zad}(k)$
u(k)
y(k)
y
k
2
1.5
1
0.5
0
-0.5
-1
0
50
100
150
200
250
300
350
400
450
500


DMC dlaAlphaNoOgr =10
×10^26
$y_{zad}(k)$
u(k)
y(k)
y
k
3
2.5
2
1.5
1
0.5
0
-0.5
-1
-1.5
0
50
100
150
200
250
300
350
400
450
500