

Wydział Elektroniki i Technik Informacyjnych
Politechnika Warszawska

Projektowanie układów w sterowania
(projekt grupowy)

Sprawozdanie z projektu i ćwiczenia laboratoryjnego
nr 3, zadanie nr 1

Zespół Z01

Hubert Kozubek, Przemysław Michalczewski

Warszawa, 2021

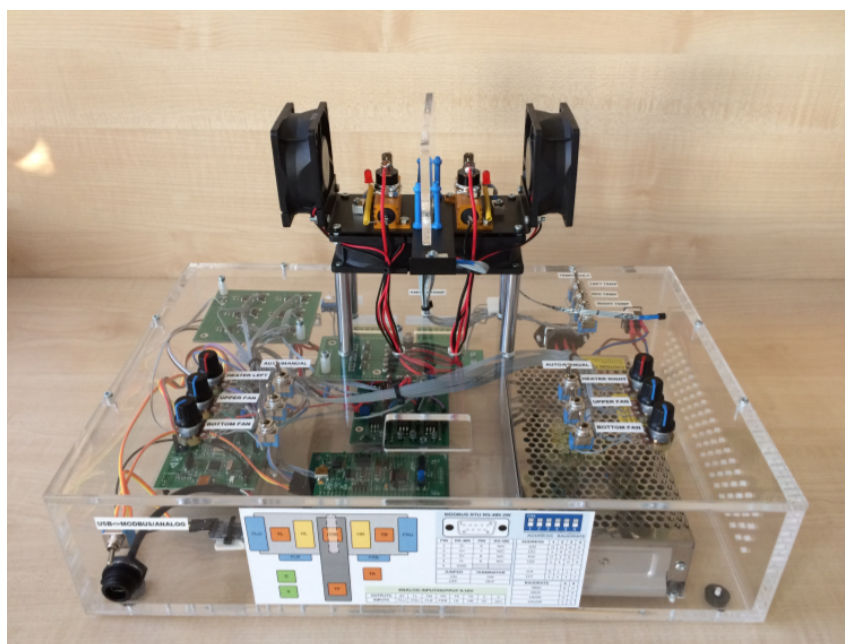
Spis treści

1. Laboratorium	2
1. Cel laboratorium	2
2. Przebieg laboratorium	2
3. Punkt pracy stanowiska	3
4. Wzmocnienie w funkcji sterowania	3
5. Regulatory dla obiektu liniowego	4
6. Rozmyty PID	5
7. Rozmyty DMC	5
8. Dobór parametrów λ	6
2. Projekt	7
1. Informacje wstępne	7
2. Sprawdzenie poprawności punktu pracy	7
3. Odpowiedzi skokowe procesu	7
4. Regulatory cyfrowe	9
5. Regulatory rozmyte	18

1. Laboratorium

1. Cel laboratorium

Celem niniejszego laboratorium była implementacja, weryfikacja poprawności działania i dobór parametrów algorytmów regulacji jednowymiarowego nieliniowego procesu laboratoryjnego dla stanowiska grzejąco-chłodzącego przedstawionego na rys. 1.1.



Rys. 1.1. Stanowisko grzejąco-chłodzące używane w trakcie laboratoriów.

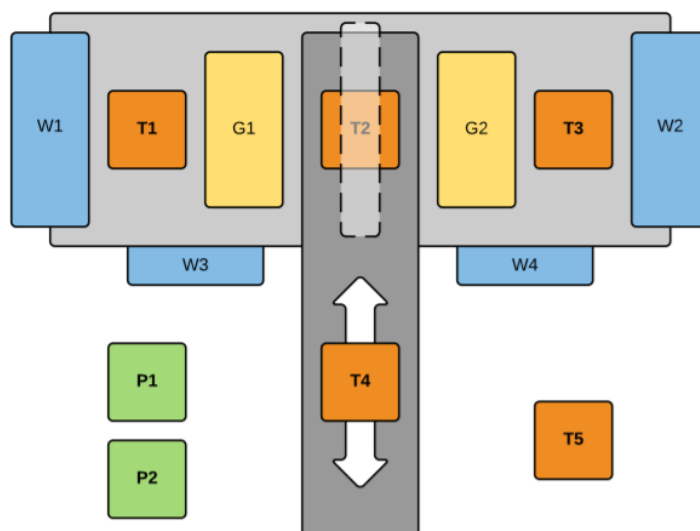
2. Przebieg laboratorium

Rozpoczynając pracę na stanowisku grzejąco-chłodzącym sprawdzono możliwość sterowania i pomiaru w komunikacji za stanowiskiem. W szczególności sygnały sterujące wykorzystywane podczas niniejszego laboratorium W1, G1, Z oraz pomiaru T1 (elementy wykonawcze przedstawiono na rys. 1.2). Przez cały czas trwania laboratorium moc wentylatora W1 była ustawiona na 50%, a wentylator był traktowany jako cecha otoczenia. Dodatkowo sprawiał on, że temperatura grzałki opadała szybciej, co było szczególnie przydatne pomiędzy doświadczeniami.

W ramach laboratorium należało wykonać 6 zadań:

1. Odczytać wartość pomiaru temperatury dla termometru T1 dla mocy 26% grzałki G1 w stanie ustalonym (wyznaczyć punkt pracy).
2. Przeprowadzić eksperyment mający na celu określenie wzmocnienie w funkcji sterowania: dla kolejnych wartości sterowania: 20, 30, ..., 80 pozyskać wartość ustabilizowanego sygnału wyjściowego i na ich podstawie narysować punkty tworzące charakterystykę statyczną
3. Dla trajektorii zmian sygnałów zadanych: T_{pp} , $T_{pp} + 5$, $T_{pp} + 15$, T_{pp} przetestować regulatory z laboratorium 1

4. Zaimplementować rozmyty algorytm PID i dla tej samej trajektorii zmian sygnału wartości zadanej spróbować dobrać parametry lokalnych algorytmów PID w taki sposób, aby osiągnąć lepszą jakość regulacji w porównaniu z regulatorem pojedynczym
5. Zaimplementować rozmyty algorytm DMC w najprostszej wersji analitycznej, o parametrach $N_u = N = D$ i $\lambda = 1$ i dla powyższej trajektorii zmian sygnału wartości zadanej wykonać eksperymenty dla 3 regulatorów lokalnych.
6. Dobrać parametry określające karę za przyrosty sterowania lokalnych algorytmów DMC metodą eksperymentalną.



Rys. 1.2. Schemat stanowiska grzejąco-chłodzącego; zaznaczone elementy wykonawcze: wentylatory W1, W2, W3, W4, grzałki G1, G2, czujniki temperatury T1, T2, T3, T4, T5 (temperatura otoczenia), pomiar prądu P1, pomiar napięcia P2.

3. Punkt pracy stanowiska

W celu wyznaczenia punktu pracy stanowiska dla mocy grzałki $G1=26\%$ zadano tę wartość dla sygnału sterującego grzałką za pomocą polecenia `sendNonlinearControls(u)`. Następnie poczekano, aż temperatura $T1$ ustali się. Wynik eksperymentu przedstawiono na rys. 1.3. Odczytana wartość temperatury dla termometru $T1$ wyniosła $31,06\text{ }^{\circ}\text{C}$.

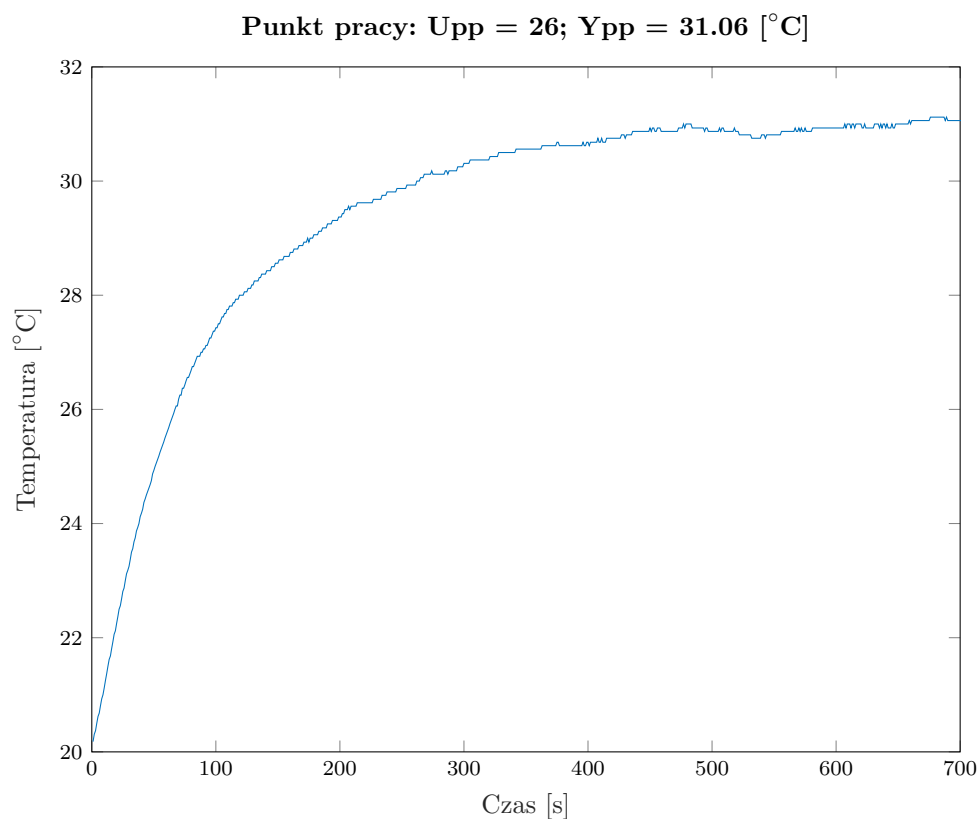
4. Wzmocnienie w funkcji sterowania

W celu wyznaczenia wzmocnienia w funkcji sterowania na stanowisku zadawano kolejne wartości sterowania, a następnie czekali aż wyjście się ustabilizuje. W taki sposób przeprowadzono 6 eksperymentów zadając kolejno wartości sterowania: 20, 30, 40, 50, 60, 70, 80. Wyniki eksperymentów przedstawiono na rys. 1.4

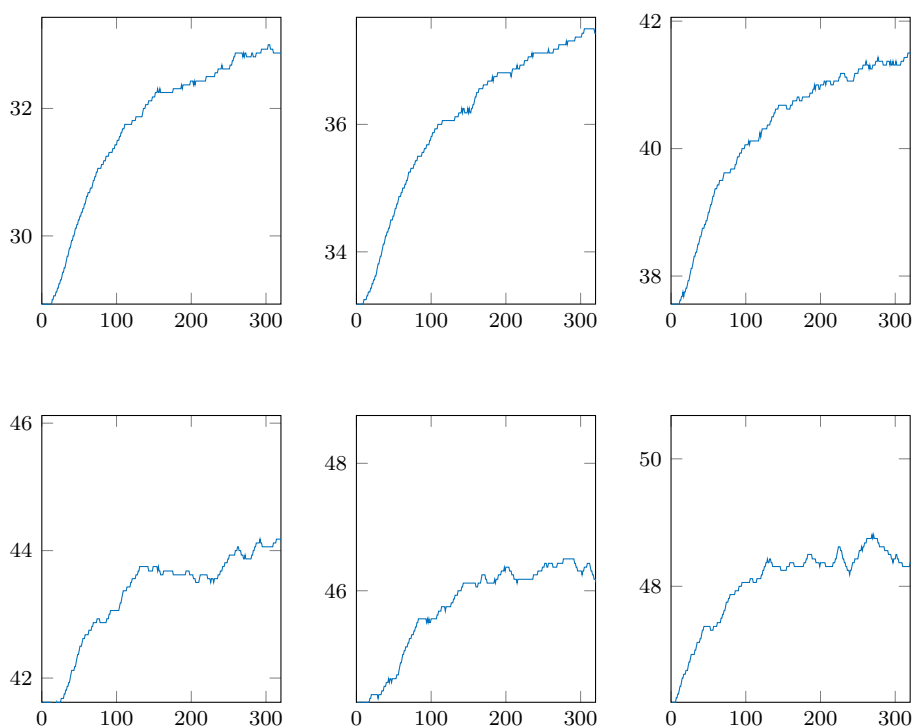
Na podstawie przeprowadzonych eksperymentów sporządzono wykres punktowy pokazujący charakterystykę statyczną procesu. Wykres ten jest przedstawiony na rys. 2.3

TODO: Regresja liniowa + K_{stat}

Jak widać z wykresu charakterystykę można przybliżyć do charakterystyki liniowej. Niemniej jednak widoczna jest nieliniowość, w szczególności porównując wartości dla u z przedziału (20,50) i z przedziału (50,80) Dla przybliżonej liniowo charakterystyki $K_{stat} =$



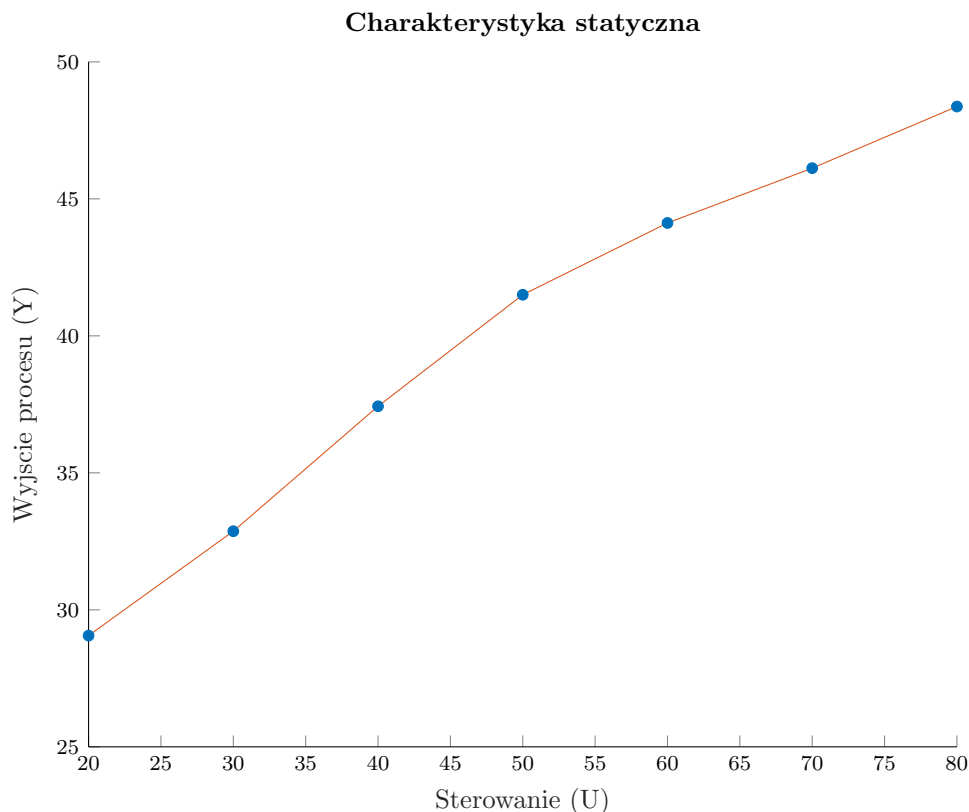
Rys. 1.3. Ustalenie się temperatury dla punktu pracy.



Rys. 1.4. Odpowiedzi stanowiska dla kolejnych wartości sterowania

5. Regulatory dla obiektu liniowego

Wykorzystując regulatory zaimplementowane na laboratorium 1 został przeprowadzony eksperyment, którego celem było sprawdzić jak regulatory przeznaczone do regulacji obiektów li-



Rys. 1.5. Charakterystyka statyczna procesu

niowych poradzą sobie z obiektem nieliniowym. W tym celu jako trajektorię wartości zadanej przyjęto 3 skoki: z 31,06 do 36,06, z 36,06 do 46,06 i z 46,06 do 31,06. Następnie dla danej trajektorii uruchomiono regulator PID, a następnie powtórzono eksperyment dla regulatora DMC. Wyniki eksperymentu przedstawiono odpowiednio na rys. TODO: PID liniowy i rys. TODO: DMC Liniowy.

6. Rozmyty PID

W celu lepszej regulacji obiektu nieliniowego został zaimplementowany regulator rozmyty PID. W tym celu utworzono funkcje dzwonową, która decydowała który regulator i w jakim stopniu ma wpływ na wyjście. Odpalanie się regulatorów PID zależało od aktualnej wartości wyjścia obiektu. Regulator został podzielony na 3 regulatory lokalne, każdy z nich został dostrojony metodą eksperymentalną, gdzie wartości startowe wzięto z regulatora PID z 1 laboratorium. Dzięki charakterystyce statycznej można było np. stwierdzić że regulator dla wyższych wartości wyjścia powinien mieć większy parametr proporcjonalny, niż regulator lokalny dla mniejszych wartości wyjścia, z racji że wzmocnienia statyczne rośnie wolniej dla wyższych wartości wyjścia, niż dla niższych. Działanie regulatora rozmytego przedstawiono na rys. TODO: fuzzy PID.

7. Rozmyty DMC

Również dla regulatora DMC skorzystano z 3 regulatorów lokalnych, także sterowanych wartościami wyjścia obiektu. Dla regulatorów lokalnych wybrano odpowiednio odpowiedź skokową dla sterowań z 20 do 30, z 40 do 50 i z 60 do 70. Dla tak przygotowanych regulatorów wybrano

najprostsze parametry $D = N = N_u = 320$ i $\lambda = 1$. Następnie rozmyty regulator DMC został przetestowany dla tej samej trajektorii co w sekcji 5.

8. Dobór parametrów λ

TODO:

2. Projekt

1. Informacje wstępne

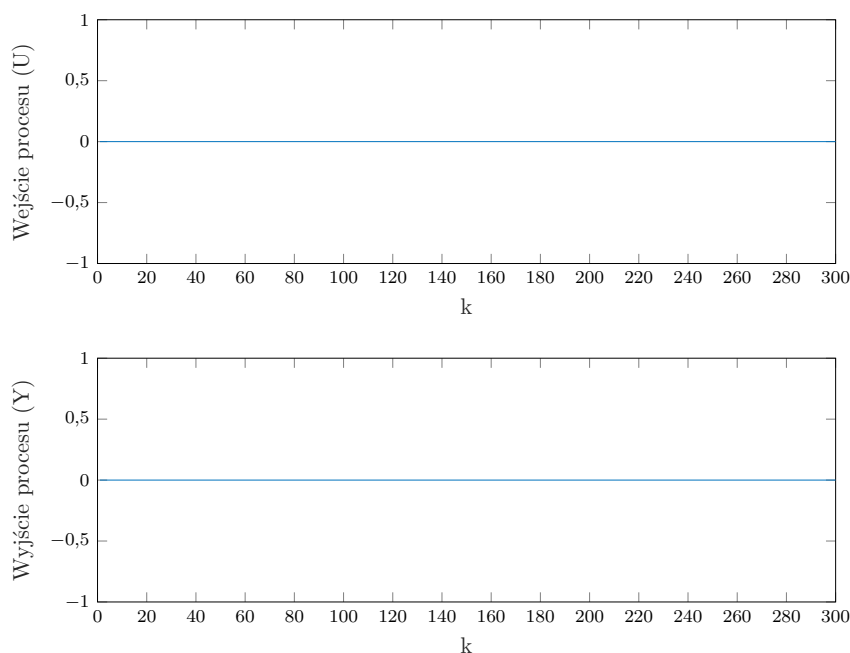
Zadanie projektowe wykorzystywało symulowany obiekt regulacji. Wyjście obiektu można wyznaczyć przy pomocy polecenia:

$$y(k) = \text{symulacja_obiektu_1y_p3}(u(k-5), u(k-6), y(k-1), y(k-2))$$

Wartości sygnałów wejścia, wyjścia i zakłócenia procesu w punkcie pracy wynoszą $u = y = z = 0$; okres próbkowania wynosi 0,5 sekundy.

2. Sprawdzenie poprawności punktu pracy

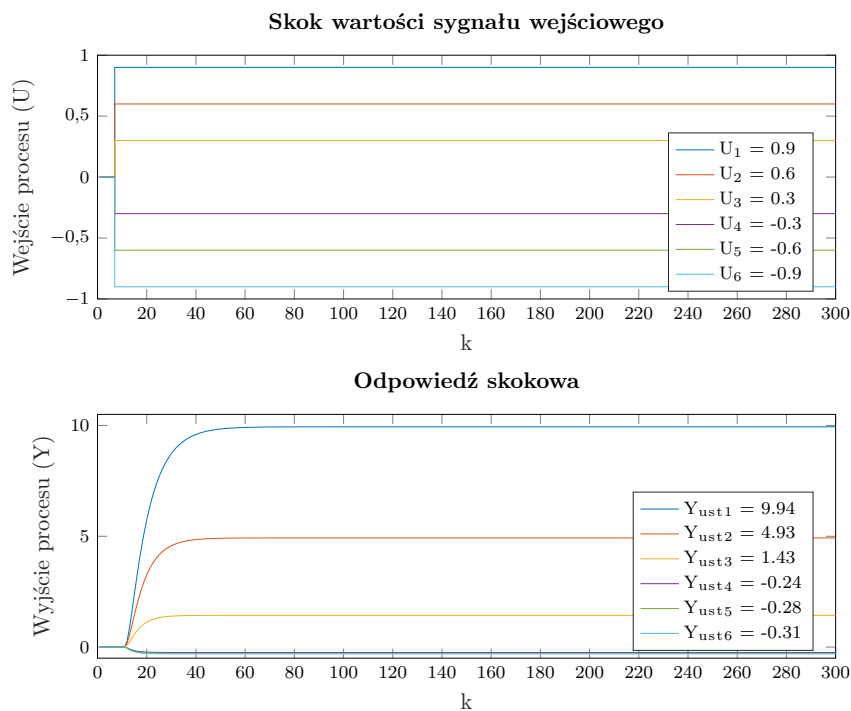
W celu sprawdzenia poprawności punktu pracy została przeprowadzona symulacja, gdzie na wejście podano sterowanie $u = 0$, jako poprzednie wartości wyjścia podano $y = 0$ i sprawdzono wartość wyjścia procesu w następnych chwilach. Wyniki symulacji przedstawiono na rys. 2.1. Wartość sygnału wyjściowego pozostała bez zmian, co potwierdza poprawność podanego punktu pracy.



Rys. 2.1. Sprawdzanie poprawności punktu pracy

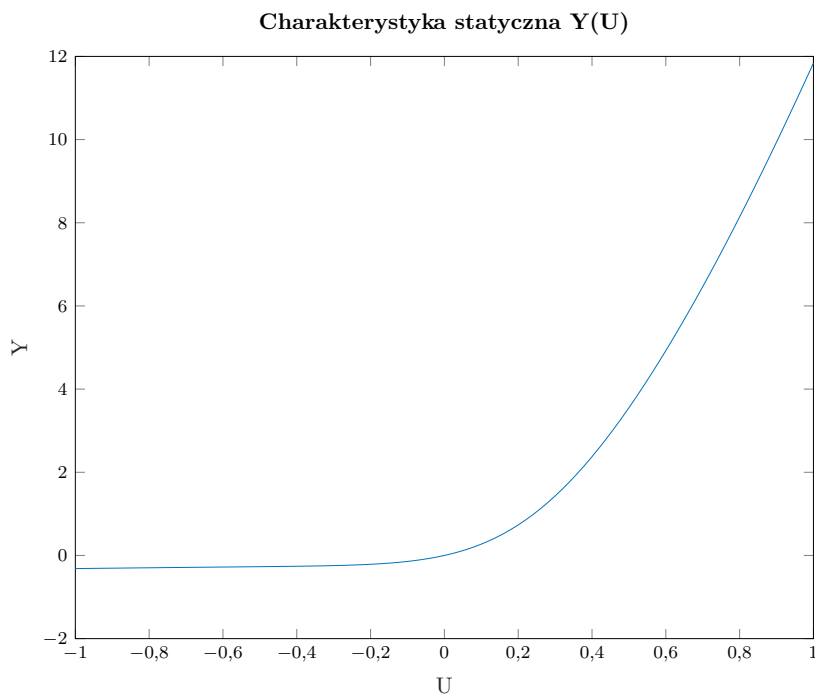
3. Odpowiedzi skokowe procesu

Wyznaczone zostały odpowiedzi skokowe dla kilku skoków wartości sygnału sterującego, gdzie wartość początkowa sterowania wynosiła $u=0$. Wyniki symulacji przedstawiono na rys. 2.2.



Rys. 2.2. Odpowiedzi skokowe procesu

Na podstawie odpowiedzi skokowych wyznaczono charakterystykę statyczną procesu. Charakterystyka jest widoczna na rys. 2.3. Z wykresu możemy łatwo stwierdzić, że charakterystyka statyczna tego procesu nie jest liniowa.



Rys. 2.3. Charakterystyka statyczna procesu

4. Regulatory cyfrowe

W celu regulacji danego procesu zostały zaimplementowany prosty cyfrowy regulator PID oraz analityczny regulator DMC.

Regulator PID:

```
function U = PID(e, K, Ti, Td, Tp, Umin, Umax)
% PID controller
% U = PID(e, K, Ti, Td, Tp, Gmin, Gmax) gives control signal for PID controller.
%
% Arguments:
% e - error value (e = y_zad - y);
% K - gain factor;
% Ti - integration time constant (Ti ~= 0);
% Td - differentiation time constant;
% Tp - sampling period (Tp > 0);
% Umin - lower limit of U;
% Umax - upper limit of U;
%
% Warning!
% Upop is value of last iteration controll signal
% in init it should have value of current work point
%
% See also PID_fuzzy, DMC, DMC_fuzzy.

persistent Upop
persistent e0
persistent e1
persistent e2
persistent r2
persistent r1
persistent r0

if isempty(e0)
    Upop = 0;           % sterowanie w punkcie pracy
    e0=0;
    e1=0;

    r2 = K*Td/Tp;
    r1 = K*(Tp/(2*Ti)-2*Td/Tp - 1);
    r0 = K*(1+Tp/(2*Ti) + Td/Tp);
end

% przesuniecie uchybow
e2 = e1;
e1 = e0;
e0 = e;

U = r2*e2 + r1*e1 + r0*e0 + Upop;

if U > Umax
    U = Umax;
```

```

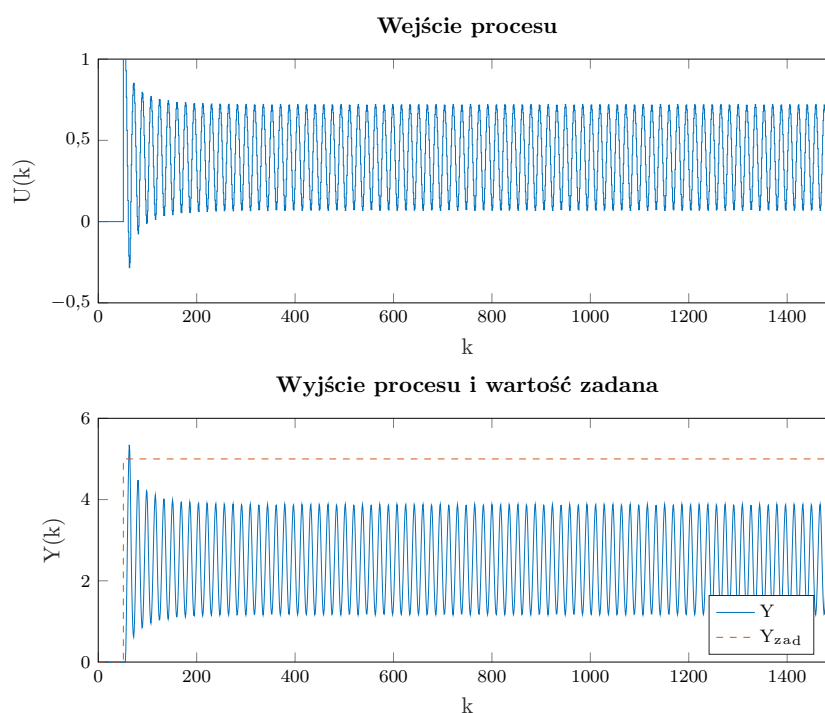
end

if U < Umin
    U = Umin;
end

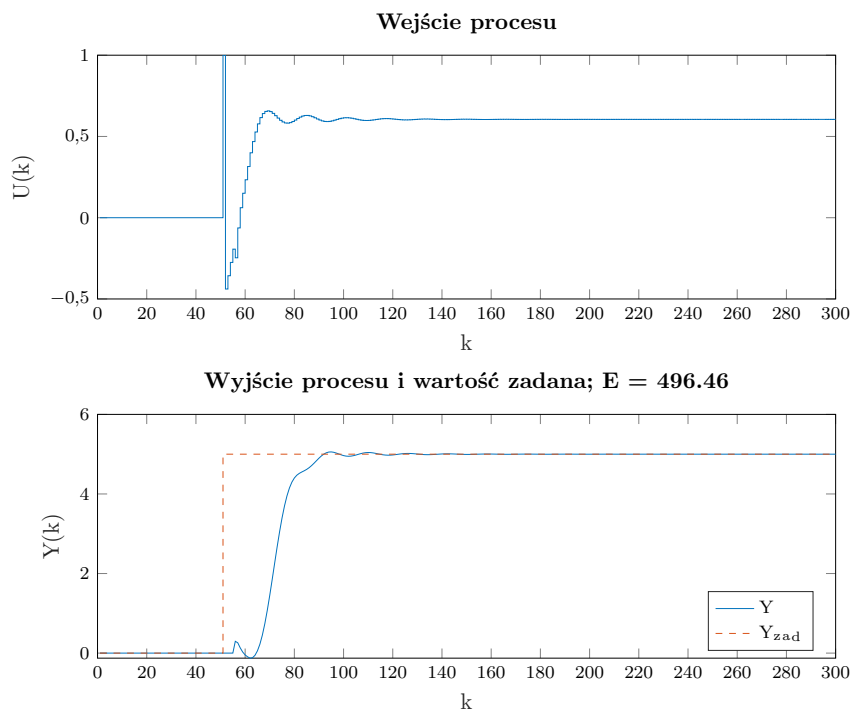
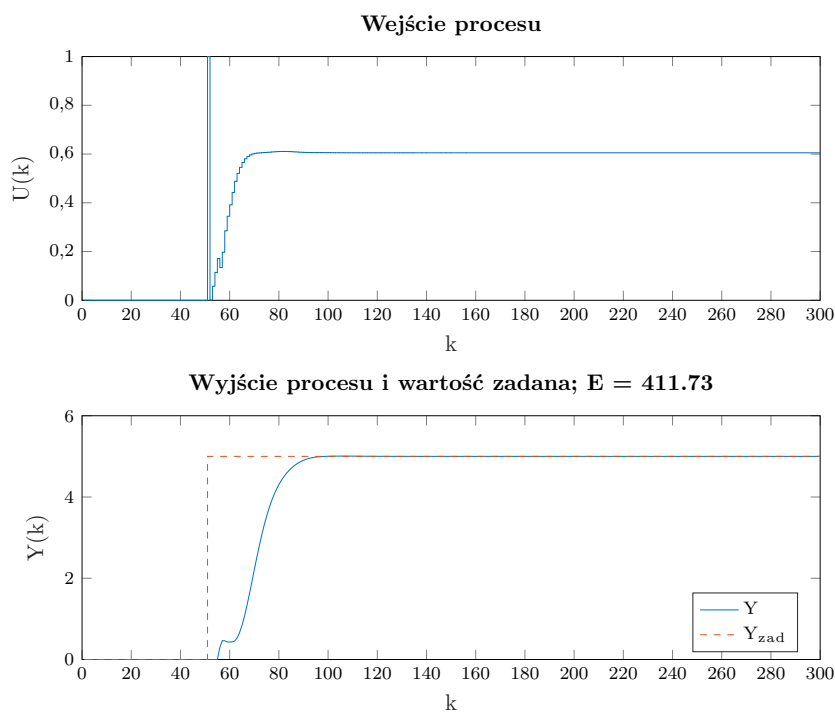
Upop = U;
end

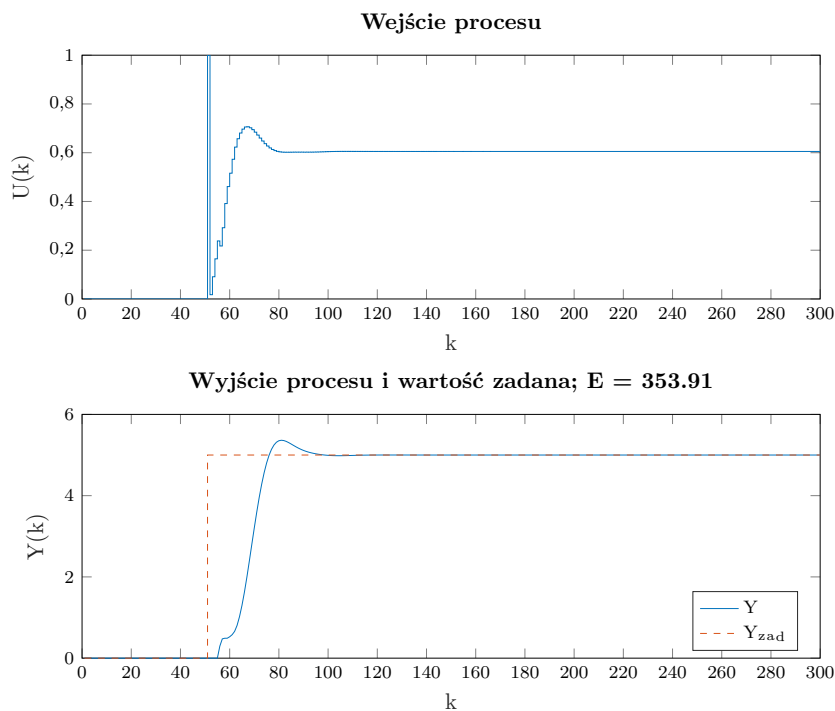
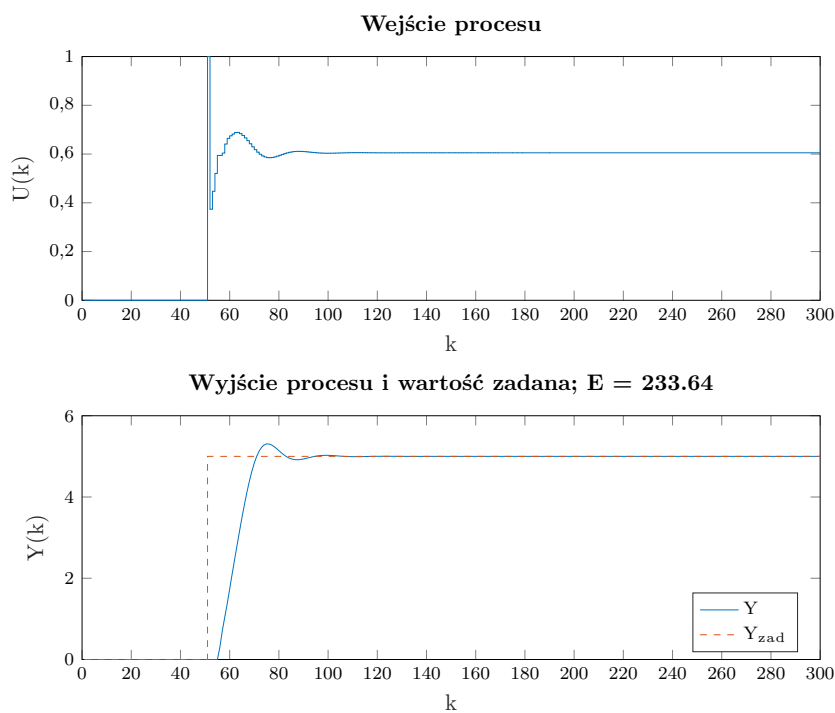
```

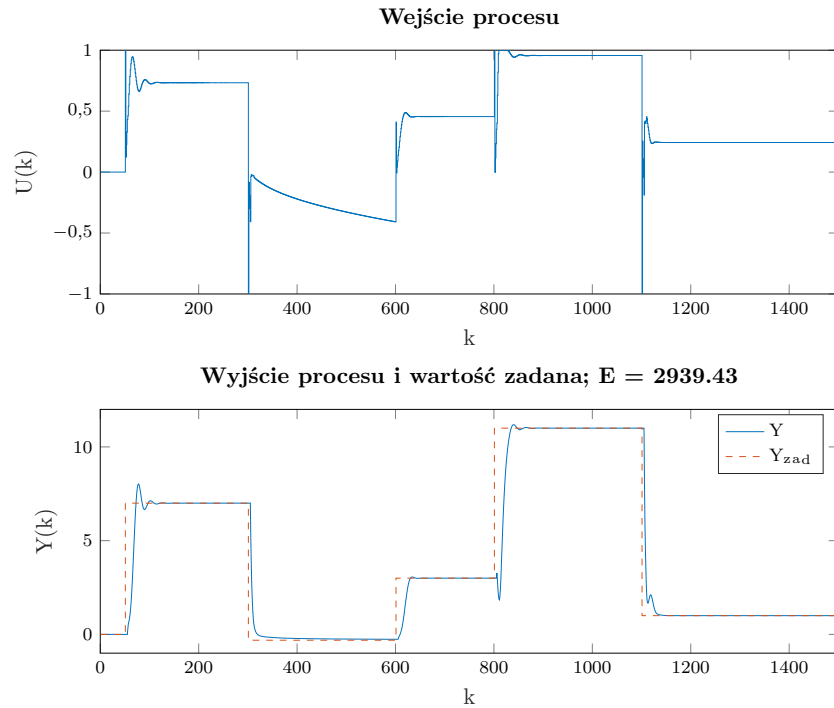
Dla podanego regulatora dobrano nastawy w sposób eksperymentalny. W pierwszej kolejności skorzystano z metody Zieglera-Nicholsa, do wyznaczenia początkowej wartości parametrów. Metoda Zieglera-Nicholsa została przedstawiona na rys. 2.4. Następnie wartości parametrów w niewielkim stopniu modyfikowano i testowano dla tej samej trajektorii wartości zadanej. Eksperymenty wykonywane podczas dobierania parametrów regulatora PID przedstawiono na rys. 2.5, rys. 2.6, rys. 2.7 i rys. 2.8. Przebieg regulatora PID z ostatecznymi nastawami został przedstawiony na rys. 2.9.



Rys. 2.4. Metoda Zieglera-Nicholsa, $K_{kryt} = 0,24$, $T_{kryt} = 8,8$

Rys. 2.5. Regulator PID, $K = 0,144$, $T_i = 4,4$, $T_d = 1,056$ Rys. 2.6. Regulator PID, $K = 0,1$, $T_i = 4,4$, $T_d = 1,056$

Rys. 2.7. Regulator PID, $K = 0,1, T_i = 3,4, T_d = 1,056$ Rys. 2.8. Regulator PID, $K = 0,1, T_i = 3,4, T_d = 0,7$

Rys. 2.9. Trajektoria docelowa, regulator PID, $K = 0,1, T_i = 3,4, T_d = 0,7$

Regulator DMC:

```
function U = DMC(S, yzad, y, D, N, Nu, lambda, Umin, Umax)
% DMC controller
% U = DMC(S, yzad, y, D, N, Nu, lambda, Umin, Umax) gives control signal for D
%
% Arguments:
% S - step answer for DMC controller (matrix)
% yzad - set point value
% y - controlled variable
% D - dynamic horizon
% N - prediction horizon
% Nu - controll horizon
% lambda - penatly factor
% Umin - lower limit of U;
% Umax - upper limit of U;
%
% Warning!
% Upop is value of last iteration controll signal
% in init it should have value of current work point
%
% See also DMC_fuzzy, PID, PID_fuzzy.
persistent init
persistent M
persistent Mp
persistent K
persistent dUP
persistent Upop
```

```

if isempty(init)
    % przedluzenie wektora S
    for i = D+1:D+N
        S(i) = S(D);
    end

    % Inicjalizacja macierzy
    M = zeros(N, Nu);
    for i = 1:Nu
        M(i:N, i)=S(1:N-i+1);
    end

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

    I = eye(Nu);

    K = ((M'*M + lambda*I)^(-1))*M';
    dUP = zeros(D-1,1);
    Upop = 0;
    init = 1;
end

% liczone online
Yzad = yzad*ones(N,1);
Y = y*ones(N,1);

Y0 = Y + Mp*dUP;
dU = K*(Yzad - Y0);
du = dU(1);

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

U = Upop + du;

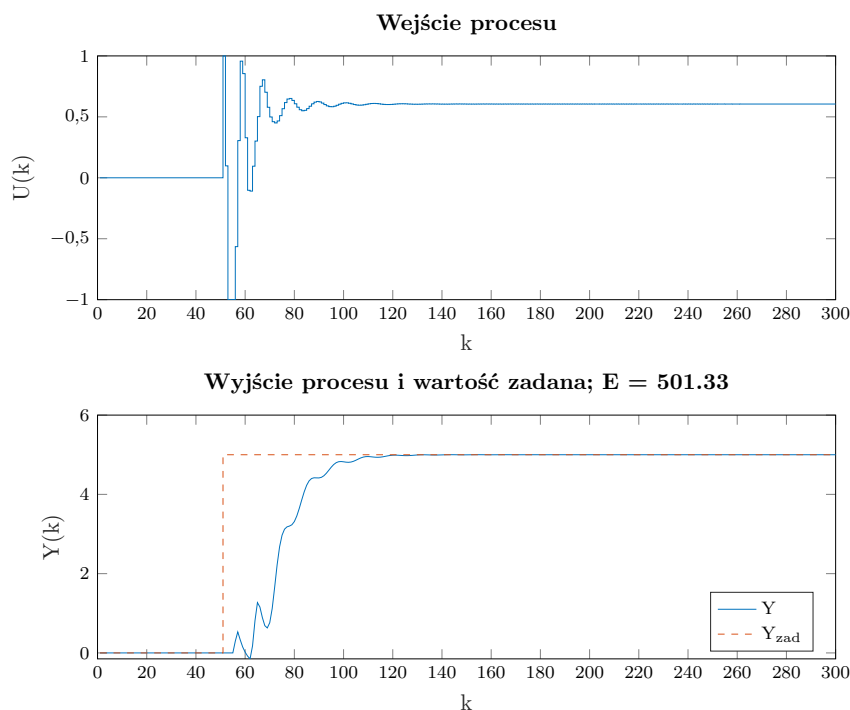
if U > Umax
    U = Umax;
end

if U < Umin
    U = Umin;
end

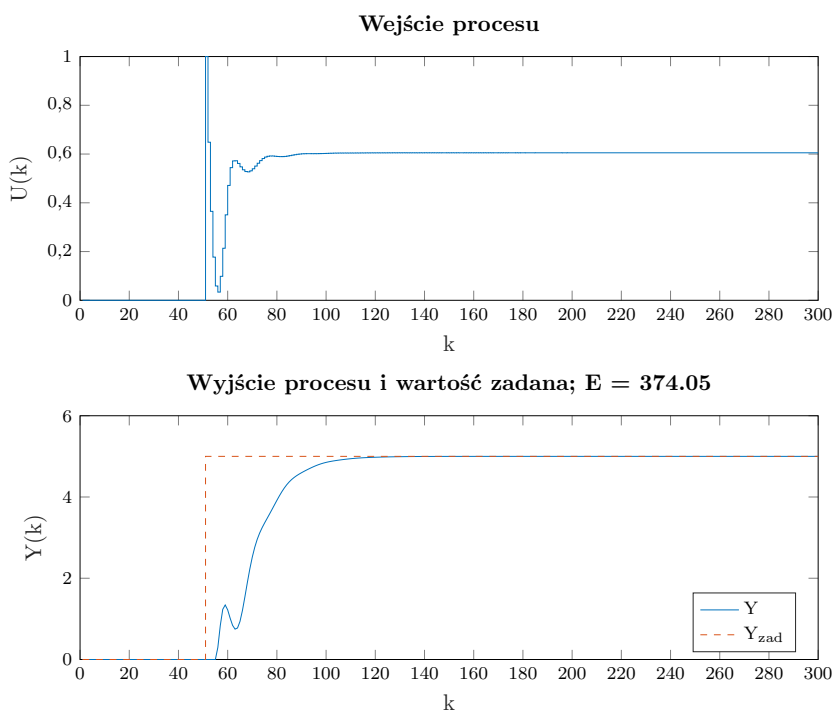
Upop = U;
end

```

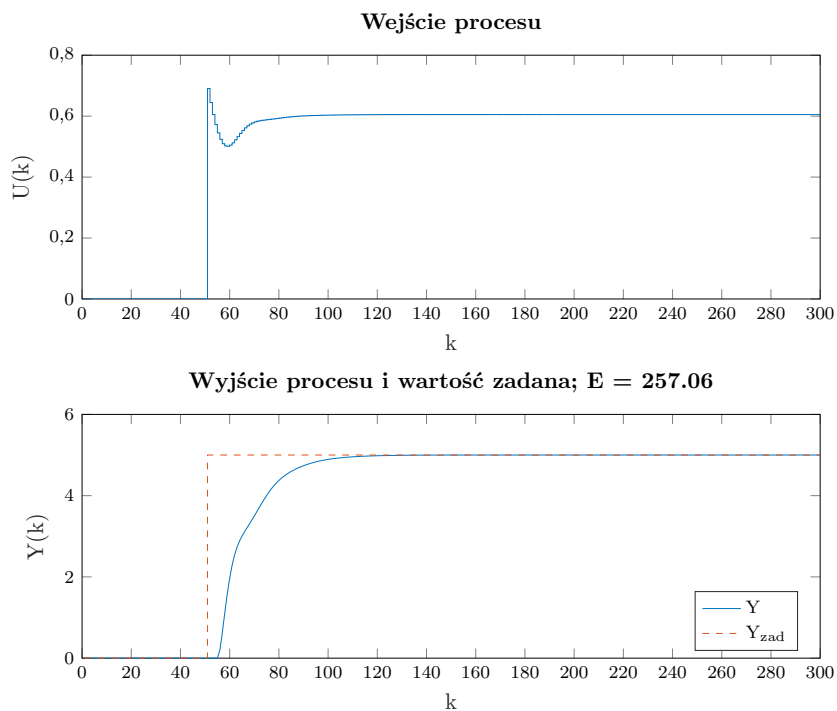
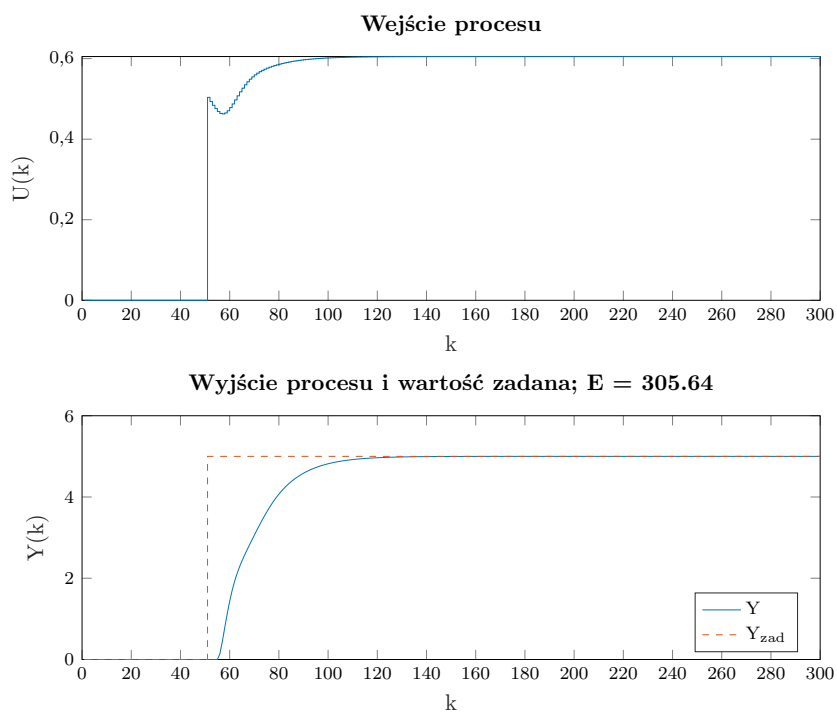
Również dla regulatora DMC wartości parametrów były dobierane w sposób eksperymentalny. Wartości parametrów były modyfikowane i porównywane w celu znalezienia najlepszego nastawu. Eksperymenty wykonywane podczas dobierania parametrów regulatora DMC przedstawiono na rys. 2.10 - rys. 2.15. Ostateczny sposób działania regulatora DMC przedstawiono na rys. ??.

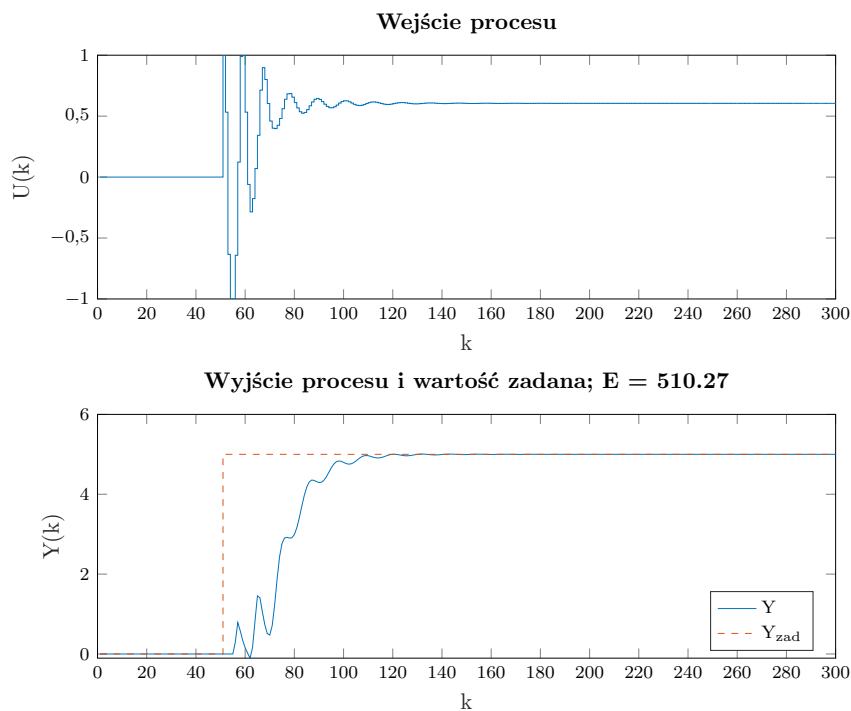
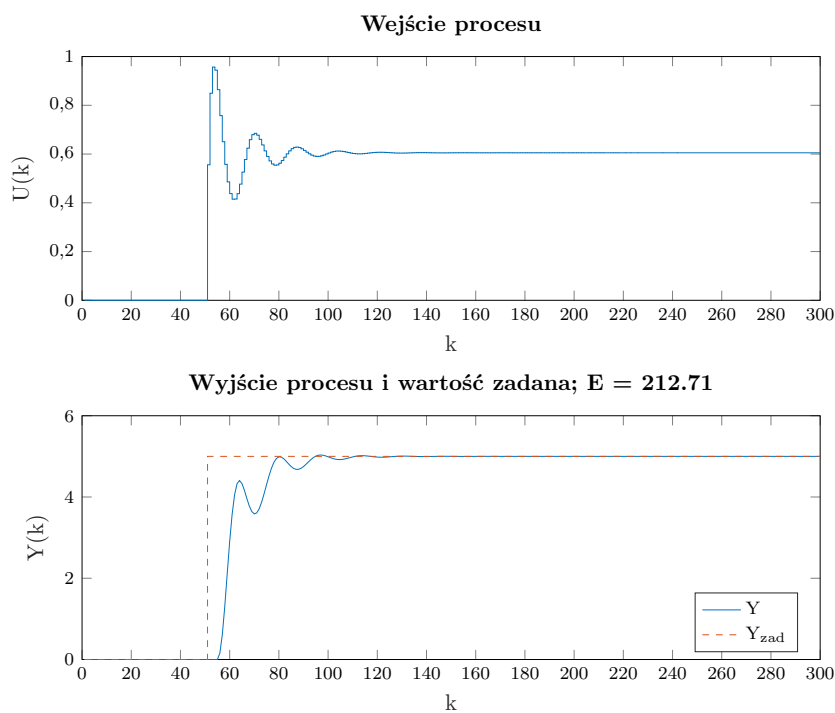


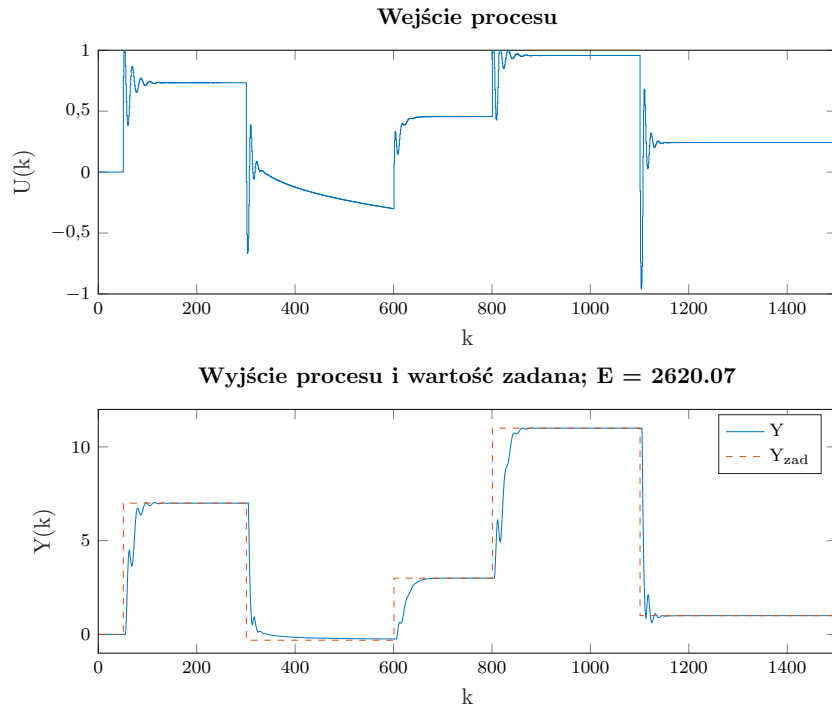
Rys. 2.10. Regulator DMC, $N = 7$, $N_u = 1$, $\lambda = 1$



Rys. 2.11. Regulator DMC, $N = 10$, $N_u = 1$, $\lambda = 1$

Rys. 2.12. Regulator DMC, $N = 20$, $N_u = 1$, $\lambda = 1$ Rys. 2.13. Regulator DMC, $N = 40$, $N_u = 1$, $\lambda = 1$

Rys. 2.14. Regulator DMC, $N = 20$, $N_u = 5$, $\lambda = 1$ Rys. 2.15. Regulator DMC, $N = 20$, $N_u = 20$, $\lambda = 50$

Rys. 2.16. Trajektoria docelowa, regulator DMC, $N = 20$, $N_u = 20$, $\lambda = 50$

5. Regulatory rozmyte

Regulator rozmyty PID:

```
function U = PID_fuzzy(e, num, u, K, Ti, Td, Tp, Umin, Umax)
% PID controller in fuzzy version
% U = PID_fuzzy(e, num, K, Ti, Td, Tp, Umin, Umax) gives control signal
% for fuzzy PID controller. It uses process output (y) as selection
% variable. It uses bell function as membership function (gbellmf).
%
% Arguments:
% e - error value (e = y_zad - y);
% num - number of local linear PID controllers (2, 3, 4,...)
% u - value of process input
% K [num, 1] - gain factor;
% Ti [num, 1] - integration time constant (Ti ~ = 0);
% Td [num, 1] - differentiation time constant;
% Tp - sampling period (Tp > 0);
% Umin - lower limit of U;
% Umax - upper limit of U;
%
% Warning!
% Upop is value of last iteration control signal
% in init it should have value of current work point
%
% See also PID, DMC_fuzzy, DMC, gbellmf.

persistent Upop
persistent e0
```

```

persistent e1
persistent e2
persistent r2
persistent r1
persistent r0
persistent a
persistent b
persistent center

if isempty(e0)
    Upop = 0;           % sterowanie w punkcie pracy
    e0=0;
    e1=0;

    r2 = zeros(num, 1);
    r1 = zeros(num, 1);
    r0 = zeros(num, 1);

    for i = 1:num
        r2(i) = K(i)*Td(i)/Tp;
        r1(i) = K(i)*(Tp/(2*Ti(i))-2*Td(i)/Tp - 1);
        r0(i) = K(i)*(1+Tp/(2*Ti(i)) + Td(i)/Tp);
    end

    % wyznaczenie funkcji przynależności (gbellmf)

    interval = (Umax - Umin)/(num - 1);

    % kształt funkcji dzwonowej
    %dla num =2:
    %     a = 0.7;           % przedział wartości maksymalnej
    %     b = 3;             % kształt zboczy funkcji

    %dla num = 3:
    %     a = 0.5;           % przedział wartości maksymalnej
    %     b = 1.5;
    %

    %dla num = 4:
    %     a = 0.3;           % przedział wartości maksymalnej
    %     b = 1.2;           % kształt zboczy funkcji
    %

    %dla num = 5:
    %     a = 0.1;           % przedział wartości maksymalnej
    %     b = 1.5;           % kształt zboczy funkcji

    center = zeros(num, 1);

    for i = 0:(num-1)
        center(i+1) = Umin + interval*i;
    end

```

```
end

% przesuniecie uchybow
e2 = e1;
e1 = e0;
e0 = e;

% Kolejno dla każdego regulatora lokalnego:
%   wyznaczenie sterowań dla poszczególnych 'num' regulatorów
%   wyznaczenie współczynników przynależności
%   wnioskowanie rozmyte

w = zeros(num, 1);
U = 0;

for i = 1:num
    u_fuzzy = Upop + r2(i)*e2 + r1(i)*e1 + r0(i)*e0;
    w(i) = gbellmf(u, [a b center(i)]);
    U = U + w(i)*u_fuzzy;
end
U = U/sum(w);

% ograniczenia sterowania
if U > Umax
    U = Umax;
end

if U < Umin
    U = Umin;
end

Upop = U;
end
```