Wydział Elektroniki i Technik Informacyjnych
Politechnika Warszawska

Projektowanie układów sterowania (projekt grupowy)

Sprawozdanie z projektu i ćwiczenia laboratoryjnego nr 3, zadanie nr 15

Michał Pióro, Radosław Ślepowroński, Jan Szymczak

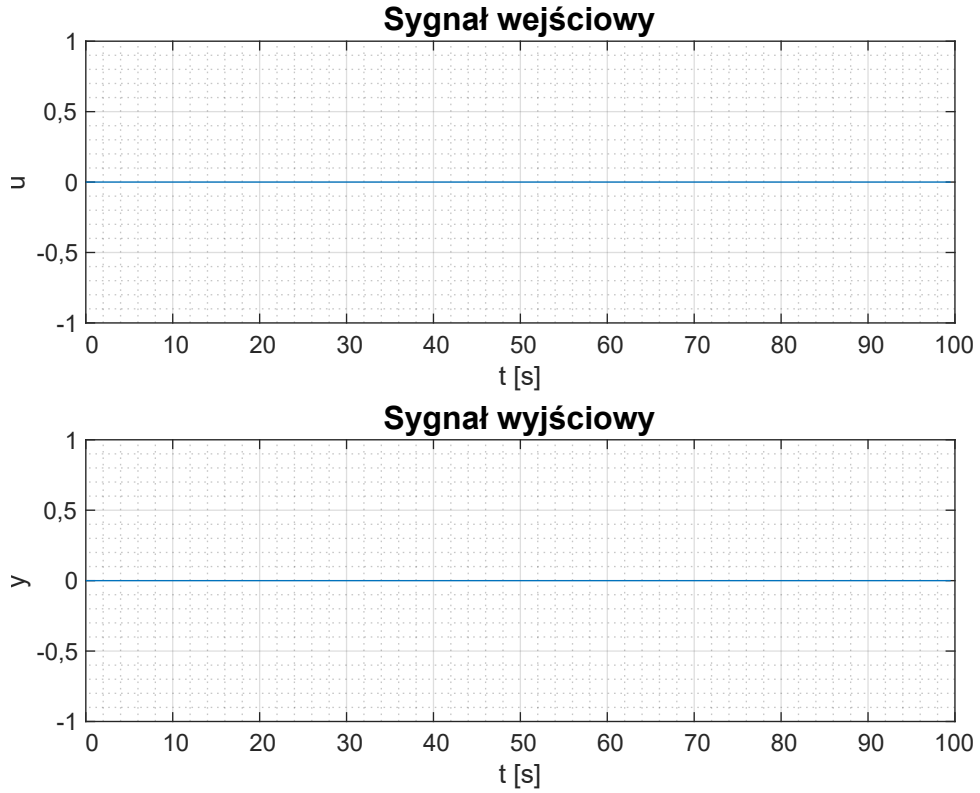
Warszawa, 2024

Spis treści

1. Projekt

1.1. Poprawność wartości U_{pp}, Y_{pp}

Aby zbadać poprawność wartości w punkcie pracy, przeprowadzono symulację, w której na wejście obiektu podawane są stałe wartości $u = 0$. Tak prezentują się wyniki symulacji:

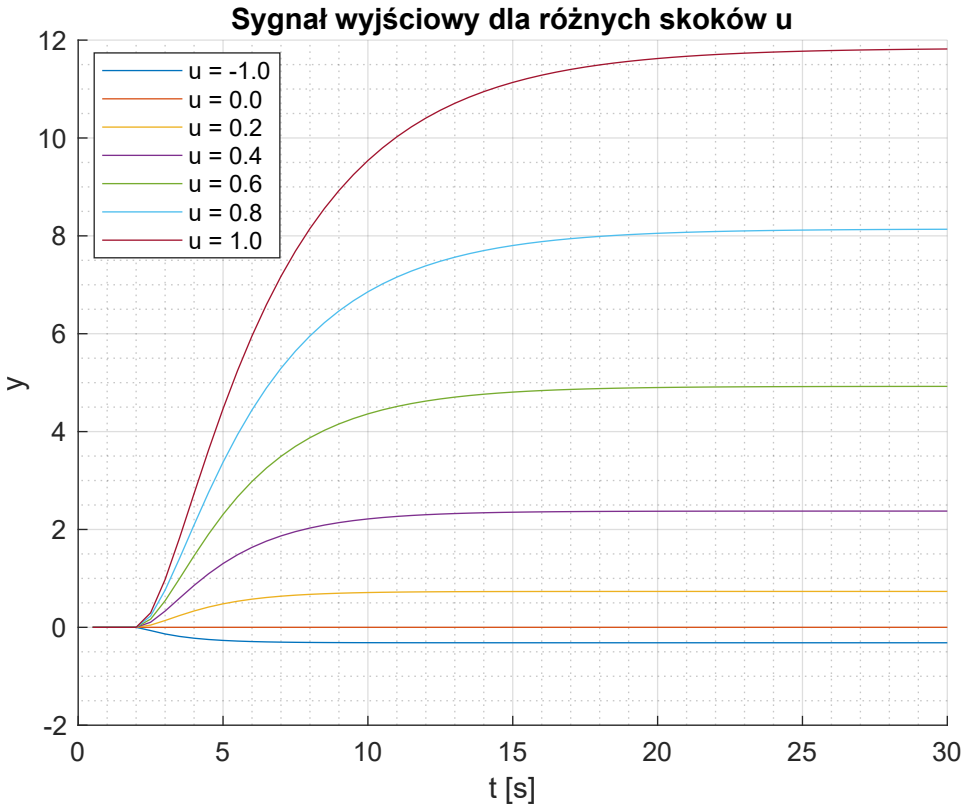


Rys. 1.1. Sprawdzenie poprawności punktu pracy

Jak widać, obiekt jest stabilny i podane wartości sygnałów faktycznie stanowią punkt pracy tego obiektu, ponieważ nie nastąpiły żadne zmiany sygnałów w prezentowanej symulacji.

1.2. Badanie liniowości obiektu

1.2.1. Odpowiedzi skokowe

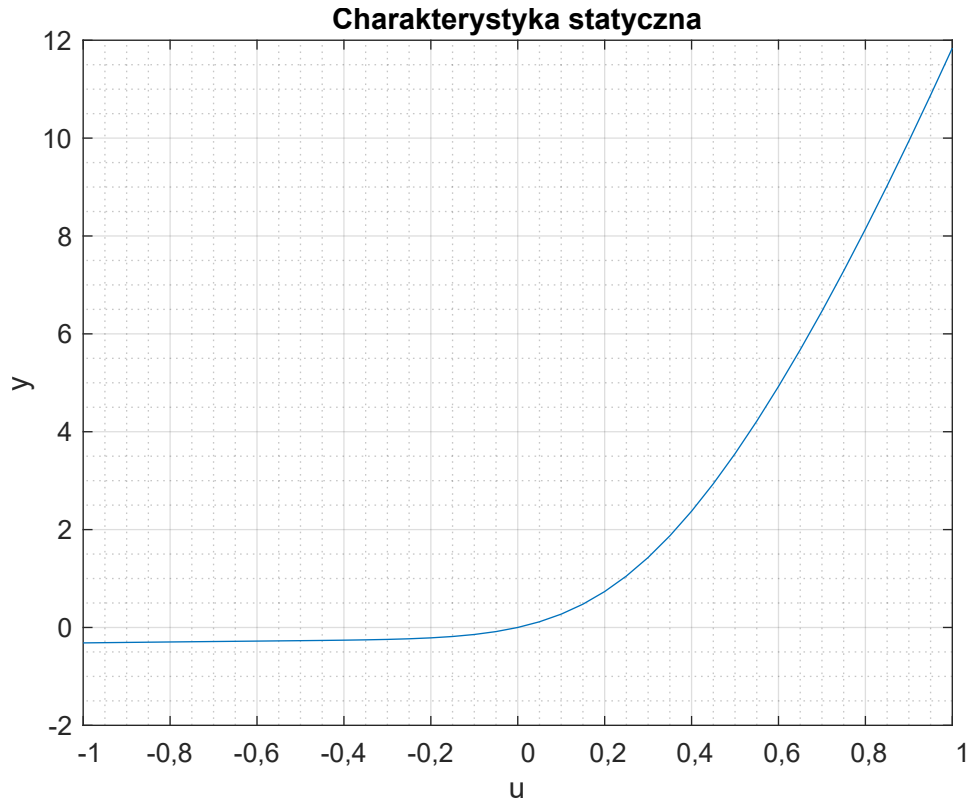


Rys. 1.2. Przykładowe odpowiedzi skokowe procesu

Obiekt charakteryzuje się nieliniową dynamiką, ponieważ przykładowo dwukrotny wzrost wielkości sterującej nie oznacza dwukrotnego wzrostu wielkości wyjściowej - dla $u = 0{,}2$ $y = 0{,}73$, a dla $u = 0{,}4$ $y = 2{,}38$. Do podobnego wniosku można dojść porównując skoki $u = -1$ i $u = 1$. Warto również dodać, że podawanie sterowania z zakresu $\langle -1; -0.2 \rangle$ daje praktycznie takie same wartości na wyjściu, co widać najbardziej w charakterystyce statycznej i dlatego z tego zakresu ukazano tylko przebieg dla jednego sterowania $u = -1$, aby wykres pozostał czytelny.

1.2.2. Charakterystyka statyczna

Charakterystyka statyczna została wyznaczona w sposób eksperymentalny - dla każdego badanego skoku u zostawiano symulację aż do czasu ustabilizowania, zapisywano stabilną wartość wyjściową i zmieniano wartość u.



Rys. 1.3. Charakterystyka statyczna y(u)

Obiekt charakteryzuje się nieliniową statyką. Obiekt jest najbardziej nieliniowy w zakresie u od 0 do 0,4. W związku z tym spodziewane są utrudnienia z regulowaniem w tym zakresie sygnału sterującego.

1.3. Regulatory konwencjonalne

1.3.1. Konwencjonalny regulator PID

Funkcja do przeprowadzenia symulacji na podstawie trajektorii `y_zad` i parametrów PID:

```
function [y, u] = p3_funkcja_pid(kk, yzad, r2, r1, r0)
% wartości ograniczeń
umin = -1; umax=1;

% inicjalizacja
u(1:kk)=0; y(1:kk)=0; e(1:kk)=0;

% warunki początkowe
u(1:6)=0; y(1:6)=0;

% główna pętla symulacyjna
for k=7:kk
    % symulacja obiektu
    y(k)=symulacja_obiektu15y_p3(u(k-5), u(k-6), y(k-1), y(k-2));

    % uchyb regulacji
    e(k)=yzad(k)-y(k);
```

```
    % sygnał sterujący regulatora PID
    u(k)=r2*e(k-2)+r1*e(k-1)+r0*e(k)+u(k-1);

    % ograniczenia wartości sygnału sterującego
    if u(k) < umin
        u(k) = umin;
    elseif u(k) > umax
        u(k) = umax;
    end
end
end
```

Funkcja użyta do wyliczania parametrów dyskretnego regulatora PID:

```
function [r2, r1, r0] = pid_offline(K, Ti, Td, Tp)
r2 = K*Td/Tp;
r1 = K*(Tp/(2*Ti) - (2*Td)/Tp - 1);
r0 = K*(1 + Tp/(2*Ti) + Td/Tp);
end
```

Przykładowe wywołanie programu:

```
K = 0.1;
Ti = 6;
Td = 2;

[r2, r1, r0] = pid_offline(K,Ti,Td,0.5);
[y, u] = p3_funkcja_pid(kk, yzad, r2, r1, r0);
```

Ponieważ obiekt jest nieliniowy ręczne strojenie regulatora PID byłoby czasochłonne i niekoniecznie zadowalające na całym zakresie sygnału sterującego, dlatego dobór nastaw PIDa dokonano przy użyciu metod optymalizacji, a konkretnie algorytmu genetycznego. Funkcją celu było zminimalizowanie błędu kwadratowego na całej trajektorii zadanej. Zostały dobrane tylko jedne nastawy, dlatego zostaną one pokazane w rozdziale 1.3.3, gdzie konwencjonalny regulator PID zostanie porównany z konwencjonalnym DMC.

1.3.2. Konwencjonalny regulator DMC

Funkcja do przeprowadzenia symulacji na podstawie trajektorii `y_zad` i parametrów DMC:

```
function [y, u] = p3_funkcja_dmc(kk, yzad, N, Nu, D, lambda)

% Wartości ograniczeń
umin = -1; umax=1;

% Odpowiedź skokowa zdyskretyzowanego systemu
ys = p3_odpowiedz_skokowa(D, 0, 0.1);

[ke, ku] = DMC_offline(ys, N, Nu, lambda, D);

% Inicjalizacja
u(1:kk)=0; y(1:kk)=0; e(1:kk)=0;
```

```
% Warunki początkowe
u(1:6)=0; y(1:6)=0;
delta_u_p(1:D-1)=0; % Przeszłe przyrosty u

% Główna pętla symulacyjna
for k=7:kk
    % Symulacja obiektu
    y(k)=symulacja_obiektu15y_p3(u(k-5), u(k-6), y(k-1), y(k-2));

    % Uchyb regulacji
    e(k)=yzad(k)-y(k);

    % Obliczenie przyrostu sygnału sterującego DMC
    delta_u = ke * e(k) - ku * delta_u_p';

    % Ograniczenia wartości sygnału sterującego
    if u(k-1)+delta_u < umin
        delta_u = umin-u(k-1);
    elseif u(k-1)+delta_u > umax
        delta_u = umax-u(k-1);
    end

    % Aktualizacja sygnału sterującego
    u(k)=u(k-1)+delta_u;

    % Aktualizacja przeszłych przyrostów sterowania
    for n=D-1:-1:2
        delta_u_p(n) = delta_u_p(n-1);
    end
    delta_u_p(1) = delta_u;
end
end
```

Obliczenia offline do regulatora DMC:

```
function [ke,ku]=DMC_offline(ys,N,Nu,lambda,D)
M = zeros(N,Nu);
for col=1:Nu
    M(col:N,col) = ys(1:N-col+1);
end

Mp = zeros(N,D-1);
for j = 1:D-1
    for i = 1:N
        c = min([i+j,D]);
        Mp(i,j) = ys(c) - ys(j);
    end
end

K = (M.'*M + lambda*eye(Nu,Nu))\M.';
K1 = K(1,:);
ke = sum(K1);
```

```
ku = K1*Mp;
end
```

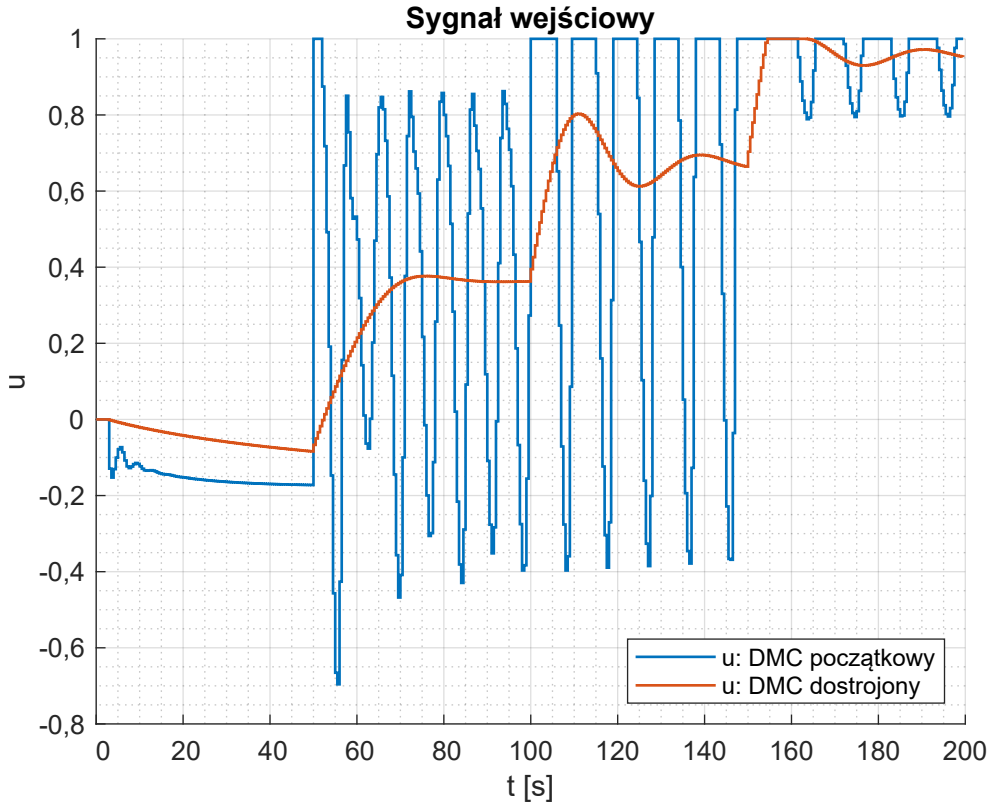
W przypadku regulatora DMC nieliniowość obiektu (jej konsekwencje) jest zdecydowanie bardziej zauważalna (jakość regulacji jest najzwyczajniej słaba), szczególnie z powodu, że odpowiedź skokowa do regulatora brana jest z okolic punktu pracy, który znajduje się w najbardziej nieliniowym zakresie. Parametry początkowego regulatora DMC mają wartości $D = N = N_u = 50$ i $\lambda = 1$. Jak widać na rysunkach 1.4 i 1.5 obiekt przy takich parametrach regulatora wpada w mocne oscylacje. Mimo, że nie są one rosnące, są one bardzo nie pożądane choćby ze względu na swoją amplitudę, która w najgorszych momentach wynosi 3 jednostki. Wybrana trajektoria wartości zadanej to kolejno:

```
    yzad(1:100)=-0.2;yzad(101:200)=2;
    yzad(201:300)=6;yzad(301:400)=11;
```

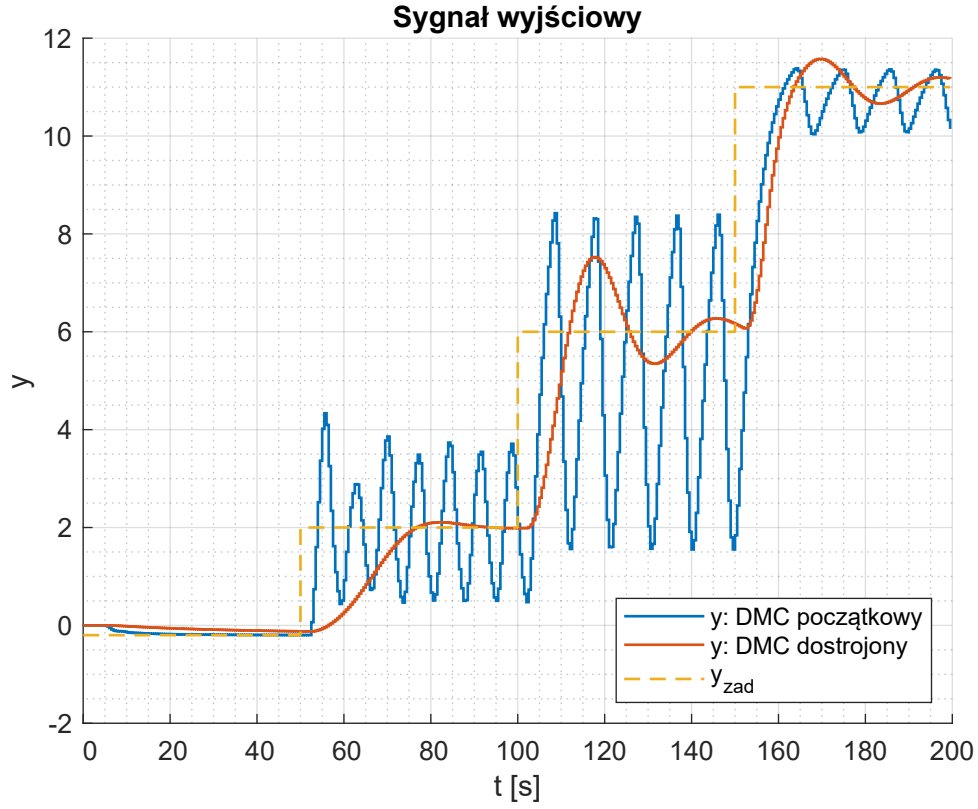
Aby bez zmiany odpowiedzi skokowej zaradzić szybkozmiennym oscylacjom, postanowiono zwiększyć współczynnik kary λ. Zadowalające efekty dawały dopiero wartości w okolicach kilku tysięcy, a ostateczną wartość λ, którą wyznaczono eksperymentalnie na podstawie błędu kwadratowego wynosi $\lambda = 10000$, co jest sporą wartością, jednak przynoszącą efekty. Tabela 1.2 przedstawiająca wskaźnik jakości regulacji pokazuje, że błąd kwadratowy dla nowego strojenia spadł o połowę. Choć działanie regulatora jest dużo wolniejsze, ma on szansę na zbiegnięcie do wartości zadanej. Pozostałe parametry N i N_u postanowiono pozostawić bez zmian, ponieważ ich zmieszanie tylko pogarszało regulację. Działanie regulatora z nową lambdą również widać na rysunkach 1.4 i 1.5.

Tab. 1.1. Parametry konwencjonalnych regulatorów DMC

Parametr	Początkowe	Dostrojone
D	50	50
N	50	50
N_u	50	50
λ	1	10 000



Rys. 1.4. Działanie regulatorów DMC z różnym parametrem λ - 1 oraz 10000



Rys. 1.5. Działanie regulatorów DMC z różnym parametrem λ - 1 oraz 10000

Tab. 1.2. Wskaźnik jakości konwencjonalnych regulatorów DMC

Regulator	Błąd kwadratowy
Początkowy	1345,4
Dostrojony	687,0

1.3.3. Porównanie regulatorów konwencjonalnych

Tak jak zostało to opisane w sekcji 1.3.1, regulator PID został nastrojony metodą optymalizacji, a dobór parametrów został dobrany w rozdziale 1.3.2. Parametry obu zostały pokazane w tabeli 1.3. Błędy każdego z regulatorów pokazano natomiast w tabeli 1.4. W tym przypadku PID radzi sobie znacznie lepiej z regulacją, szybciej narasta po zmianie wartości zadanej, a oscylacje szybciej się stabilizują. Gorsza jakość działania DMC wynika głównie z modelu, który słabo nadaje się do szerokiego zakresu pracy w nieliniowym obiekcie.

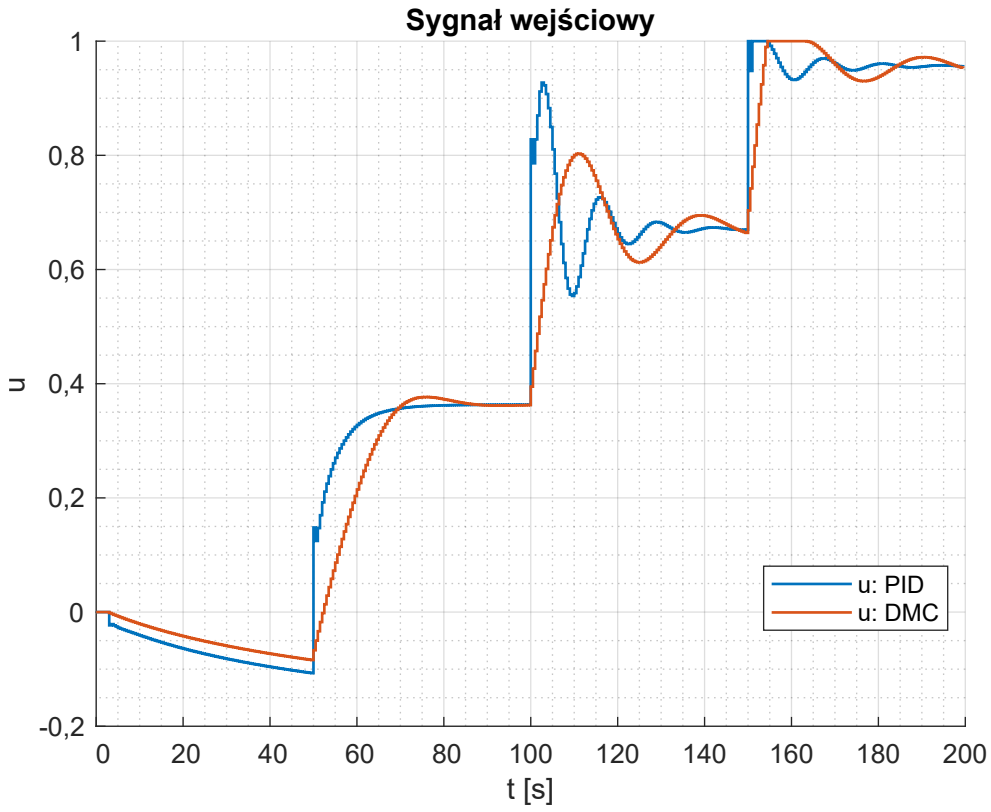
Tab. 1.3. Parametry regulatorów PID i DMC

Parametr	PID
K	0,090
T_i	4,298
T_d	0,116

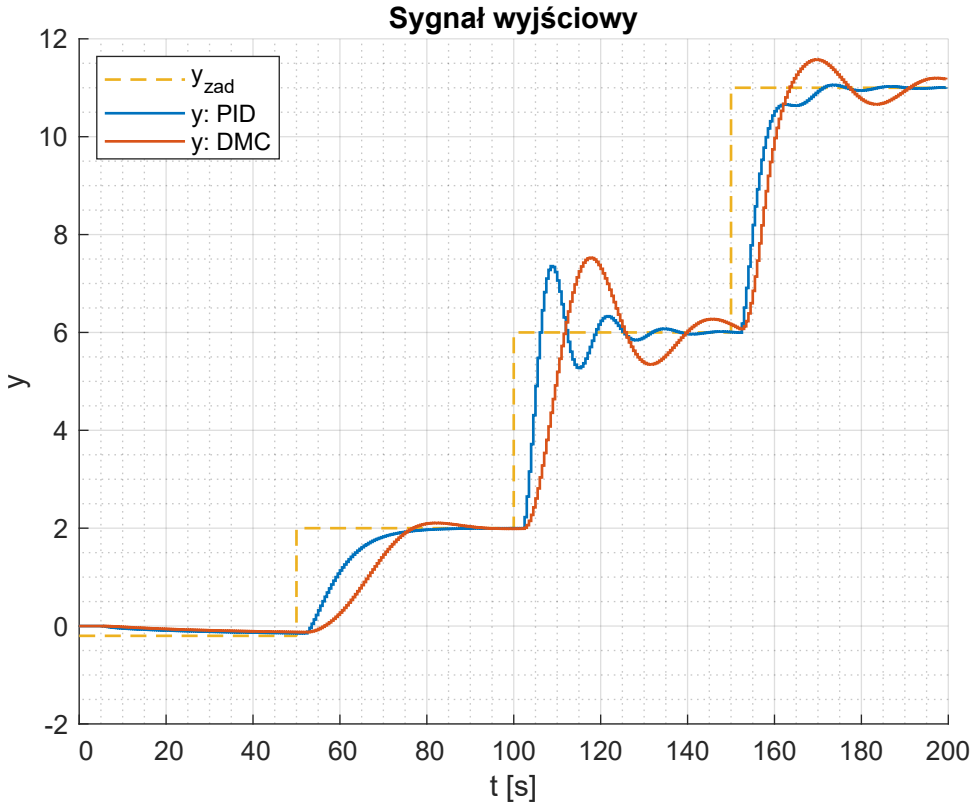
Parametr	DMC
D	50
N	50
N_u	50
λ	10 000

Tab. 1.4. Wskaźnik jakości konwencjonalnych regulatorów PID i DMC

Regulator	Błąd kwadratowy
PID	442,75
DMC	687,00



Rys. 1.6. Porównanie konwencjonalnych regulatorów PID i DMC



Rys. 1.7. Porównanie konwencjonalnych regulatorów PID i DMC

1.4. Rozmyty regulator PID

Zmieniony został sposób przekazywania parametrów dyskretnego PIDa, aby łatwiej zarządzać wszystkimi parametrami:

```
function r = p5_pid_offline_rozmyty(K, Ti, Td, Tp)
r2 = K*Td/Tp;
r1 = K*(Tp/(2*Ti) - (2*Td)/Tp - 1);
r0 = K*(1 + Tp/(2*Ti) + Td/Tp);
r = {r2, r1, r0};
end
```

Funkcja do symulacji rozmytego PIDa:

```
function [y, u] = p5_funkcja_pid_rozmyty(kk, yzad, ...
    parametry_pid, centra_rozmycia, z_switch)
% wartości ograniczeń
umin = -1; umax=1;

% inicjalizacja
u(1:kk)=0; y(1:kk)=0; e(1:kk)=0;

% warunki początkowe
u(1:6)=0; y(1:6)=0;

% główna pętla symulacyjna
for k=7:kk
```

```
    % symulacja obiektu
    y(k)=symulacja_obiektu15y_p3(u(k-5), u(k-6), y(k-1), y(k-2));

    % uchyb regulacji
    e(k)=yzad(k)-y(k);

    % obliczenie przynależności na podstawie aktualnego sygnału y
    mi = fun_przyn_trap(y(k), centra_rozmycia, z_switch);

    % sygnały sterujące każdego z lokalnych regulatorów
    ur = zeros(1, length(centra_rozmycia));
    for i=1:length(parametry_pid)
        r2 = parametry_pid{i}{1};
        r1 = parametry_pid{i}{2};
        r0 = parametry_pid{i}{3};
        ur(i) = r2*e(k-2)+r1*e(k-1)+r0*e(k)+u(k-1);
    end

    % sygnał sterujący regulatora PID - defuzyfikacja
    u(k) = mi * ur';

    % ograniczenia wartości sygnału sterującego
    if u(k) < umin
        u(k) = umin;
    elseif u(k) > umax
        u(k) = umax;
    end
end
end
```

Przykład uruchomienia symulacji rozmytego PIDa:

```
ilosc_modeli = 3;

% parametry lokalnych regulatorów
if ilosc_modeli == 1
    centra_rozmycia = 1.00; % *
    z_switch = 1.00; % *
    % * dowolne, ponieważ w tym przypadku nie ma rozmycia

    K_values = 0.09001;
    Ti_values = 4.29786;
    Td_values = 0.11634;

elseif ilosc_modeli == 2
    centra_rozmycia = [0.00 4.00];
    z_switch = 1.00;
    K_values = [0.19060 0.06232];
    Ti_values = [4.02327 4.31750];
    Td_values = [0.55439 0.05790];

elseif ilosc_modeli == 3
```

```
    centra_rozmycia = [0, 3, 6];
    z_switch = 1;
    K_values = [0.17832 0.10239 0.05427];
    Ti_values = [2.69015 7.84737 3.52440];
    Td_values = [0.94919 0.26134 0.07093];

elseif ilosc_modeli == 4
    centra_rozmycia = [-0.30  1.00  4.00  8.00];
    z_switch = 1.00;
    K_values = [0.24600 0.08191 0.06264 0.06181];
    Ti_values = [6.20168 4.33436 6.84850 3.57889];
    Td_values = [0.56929 0.48366 0.05358 0.05684];

elseif ilosc_modeli == 5
    centra_rozmycia = [0.00  2.00  4.00  8.00 10.00];
    z_switch = 1.00;
    K_values = [0.17329 0.12520 0.04616 0.11658 0.10576];
    Ti_values = [2.53968 5.64913 7.06806 1.26287 4.94235];
    Td_values = [0.99052 0.35427 0.01298 0.09492 0.99590];

end

% Tworzenie struktury parametrów
parametry_pid = cell(1, ilosc_modeli);
for i = 1:ilosc_modeli
    parametry_pid{i} = p5_pid_offline_rozmyty(K_values(i), ...
        Ti_values(i), Td_values(i), Tp);
end

[y, u] = p5_funkcja_pid_rozmyty(kk, yzad, parametry_pid, ...
    centra_rozmycia, z_switch);
```

1.5. Rozmyty regulator DMC

Funkcja do pozyskiwania lokalnych odpowiedzi skokowych:

```
function [y,u] = p3_odpowiedz_skokowa(kk, Ypp, dU)

Td = 4; kk = kk + Td + 3;

% lokalne Upp odpowiednie do Ypp
Upp = znajdz_u_do_odp_DMC(Ypp);

u(1:Td+2)=Upp; u(Td+3:kk)=Upp+dU; y(1:kk)=Ypp;

for k=3+Td:kk
    y(k) = symulacja_obiektu15y_p3(u(k-5),u(k-6),y(k-1),y(k-2));
end

%normalizacja odpowiedzi skokowej
u=(u-Upp)/dU;
```

```
y=(y-Ypp)/dU;
u(1:Td+3)=[];y(1:Td+3)=[];
end
```

Aby odpowiedzi skokowe były użyteczne, proces musi znajdować się w stanie ustalonym przed rozpoczęciem ich pozyskiwania. W tym celu stworzono funkcję, która na podstawie podanego y, charakterystyki statycznej oraz jej interpolacji, wylicza odpowiadającą wartość u. Dzięki temu najwcześniejsze próbki odpowiedzi skokowej są dobrze wyznaczone, co jest kluczowe, ponieważ to właśnie one mają największe znaczenie podczas działania regulatora DMC.

```
function u = znajdz_u_do_odp_DMC(y_szuk)
kk=500;

us=-1:0.05:1;
ys=zeros(length(us),1);

% charakterystyka statyczna
for i=1:length(us)
    u(1:kk)=us(i); y(1:6)=0;
    for k=7:kk
        y(k)=symulacja_obiektu15y_p3(u(k-5),u(k-6),y(k-1),y(k-2));
    end
    ys(i) = y(kk);
end

% Sprawdzenie zakresu
if y_szuk < min(ys) || y_szuk > max(ys)
    error('y_zad znajduje się poza zakresem charakterystyki');
end

% Interpolacja numeryczna
u = interp1(ys, us, y_szuk, 'linear');
end
```

Wyznaczenie parametrów każdego z lokalnych regulatorów i zapisanie w odpowiedniej strukturze do łatwiejszego wykorzystania:

```
function [ke_r, ku_r] = DMC_rozmyty_offline(modele, ...
    N,Nu,lambdy,D)
liczba_modeli = length(modele);
ke_r = [];
ku_r = [];

for i = 1:liczba_modeli
    % Model odpowiedzi skokowej
    ys = modele{i};
    % DMC obliczenia offline
    [ke_r1, ku_r1] = DMC_offline(ys,N,Nu,lambdy(i),D);
    ke_r = [ke_r, ke_r1];
    ku_r = [ku_r; ku_r1];
end
end
```

Przykład uruchomienia symulacji z regulatorem DMC dla przykładowych parametrów:

```
ilosc_modeli = 2;

% parametry lokalnych regulatorów
if ilosc_modeli == 1
    N = 50;
    Nu = 50;
    D = 50;
    lambdy = 1;
    centra_rozmycia = [0]; % *
    z_switch = 1; % *
    % * dowolne, ponieważ w tym przypadku nie ma rozmycia

elseif ilosc_modeli == 2
    N = 50;
    Nu = 50;
    D = 50;
    lambdy = [1,1];
    centra_rozmycia = [0,1];
    z_switch = 1;

elseif ilosc_modeli == 3
    N = 50;
    Nu = 50;
    D = 50;
    lambdy = [1,1,1];
    centra_rozmycia = [0,1,8];
    z_switch = 1;

elseif ilosc_modeli == 4
    N = 50;
    Nu = 50;
    D = 50;
    lambdy = [1,1,1,1];
    centra_rozmycia = [0,2,4,8];
    z_switch = 2;

elseif ilosc_modeli == 5
    N = 50;
    Nu = 50;
    D = 50;
    lambdy = [1,1,1,1,1];
    centra_rozmycia = [0,1,2,4,6];
    z_switch = 1;

end

% Tworzenie modeli w pętli
modele = cell(1, ilosc_modeli);
for i = 1:ilosc_modeli
    modele{i} = p3_odpowiedz_skokowa(D,centra_rozmycia(i),0.1);
```

```
end

% wyznaczenie parametrów każdego z lokalnych regulatorów
[ke_r, ku_r] = DMC_rozmyty_offline(modele, N,Nu,lambdy,D);

[y, u] = p5_funkcja_dmc_rozmyty(kk, yzad, D, centra_rozmycia, ...
    z_switch, ke_r, ku_r);
```

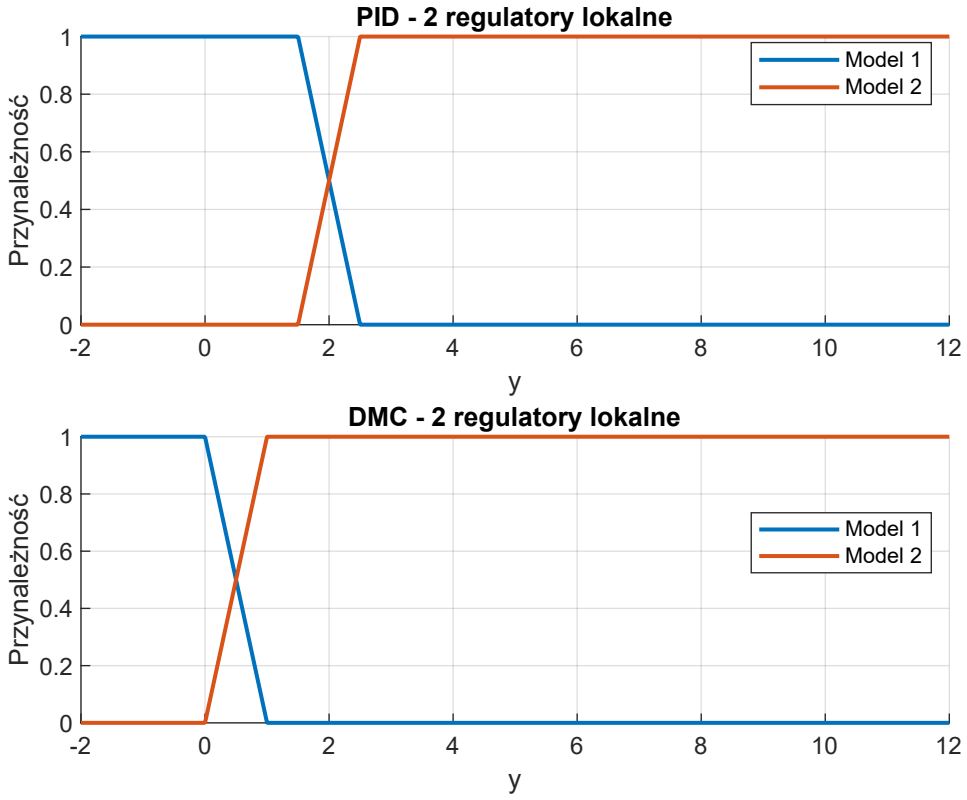
1.6. Funkcja przynależności

W przypadku funkcji przynależności zdecydowano się na zastosowanie funkcji trapezoidalnej, analogicznie jak w przypadku funkcji przynależności opracowanej dla obiektu laboratoryjnego. Wybór ten wynika z tych samych przesłanek, które szczegółowo opisano (chronologicznie wcześniej) w rozdziale 2.4.1. Aby uniknąć powtarzania treści, pozostawiono jedynie odniesienie do tego fragmentu, gdzie wyjaśniono przyjęte założenia.

1.7. Wyniki symulacji regulatorów rozmytych

1.7.1. Dwa regulatory lokalne

Wykresy wykorzystanych funkcji przynależności dla regulatorów:

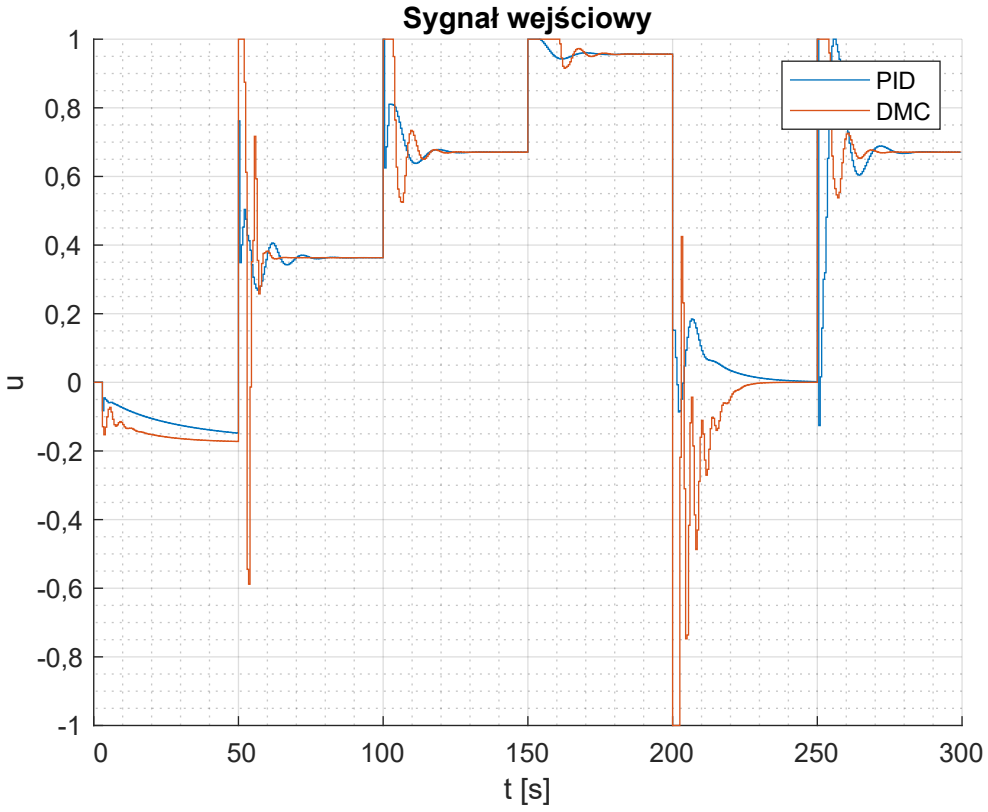


Rys. 1.8. Funkcje przynależności dla regulatorów PID i DMC

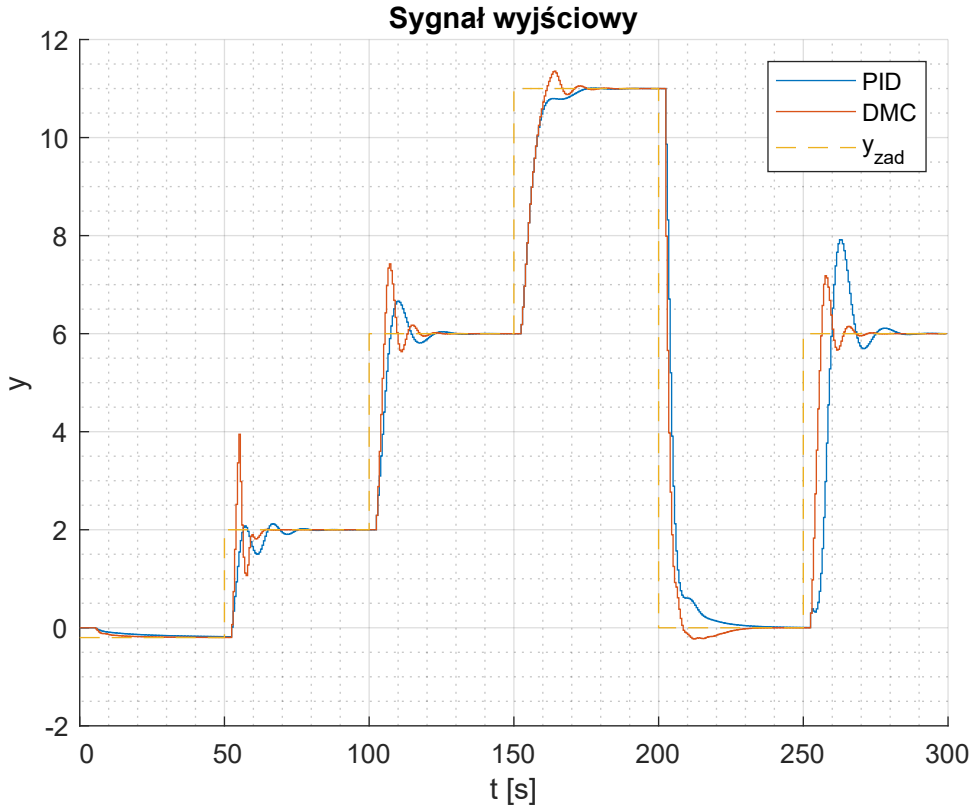
Wybrana trajektoria wartości zadanej używana w przyszłych symulacjach obu regulatorów przedstawia się następująco:

```
yzad(1:100)=-0.2;yzad(101:200)=2;yzad(201:300)=6;
yzad(301:400)=11;yzad(401:500)=0;  yzad(501:600)=6;
```

Uzyskane wyniki będą oceniane najpierw jakościowo na podstawie uzyskanych wykresów, kolejno dla każdej liczby regulatorów lokalnych. Na samym końcu nastąpi porównanie ilościowe regulatorów na podstawie tabeli zawierającej błędy kwadratowe. Nastawy regulatora PID zostały dobrane z pomocą optymalizacji z użyciem algorytmu genetycznego, podobnie jak w przypadku regulatora konwencjonalnego. W przypadku regulatora DMC zastosowano podejście, że dobrą i pewną regulację zapewni założenie, w którym parametry $D = N = N_u$. Analizując odpowiedzi skokowe, przyjęto $D = 50$. Teoretycznie, zmniejszenie parametrów N lub N_u nie powinno przynosić lepszej regulacji, dlatego nie dobieraliśmy tych parametrów. Parametr λ zgodnie z poleceniem wynosi 1 dla każdego regulatora lokalnego.



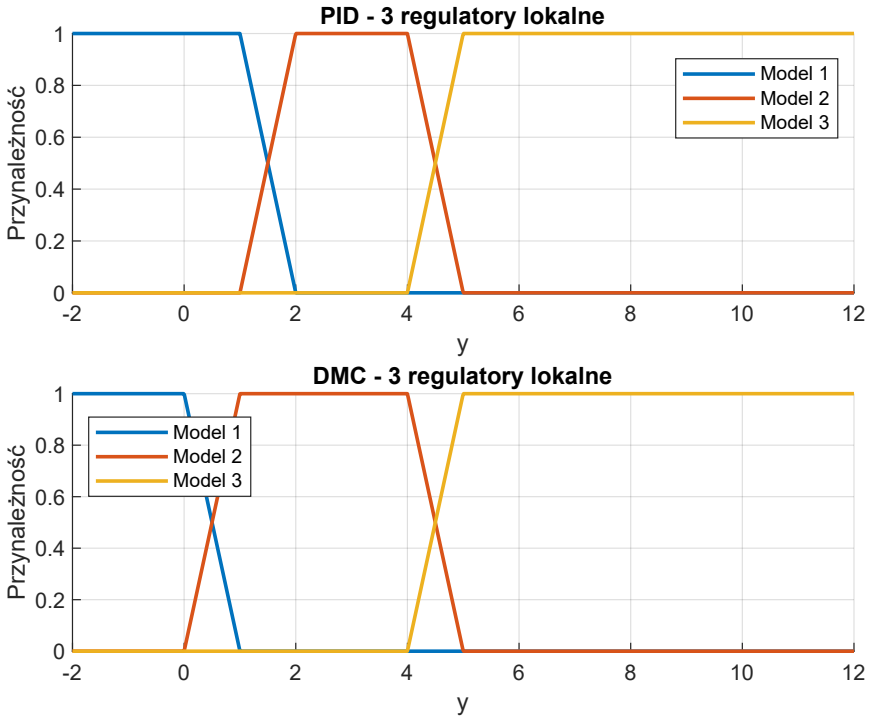
Rys. 1.9. Sygnały wejściowe regulatorów rozmytych, 2 regulatory lokalne



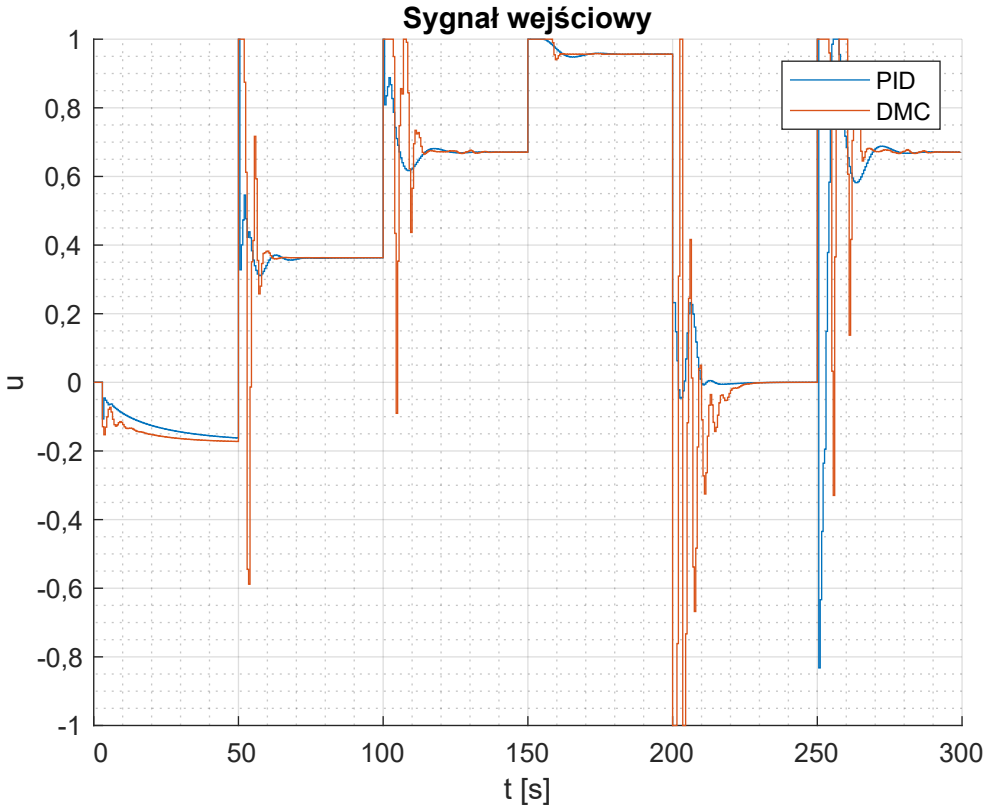
Rys. 1.10. Sygnały wyjściowe regulatorów rozmytych, 2 regulatory lokalne

Dla pierwszych dwóch skoków, zdecydowanie mniejsze przeregulowanie i oscylacje daje regulator PID, jednakże czas ustalenia jest mniejszy dla regulatora DMC. Z kolei w przypadku dwóch następnych skoków regulatory dają bardzo podobne wyniki, z tą różnicą, że DMC w dalszym ciągu generuje małe przeregulowanie, a PID natomiast niedoregulowanie. Czasy ustalenia tym razem są niemal identyczne. Dla ostatniego skoku wartości zadanej regulator DMC zapewnił lepszą regulację - w kontekście każdego wymienionego kryterium: mniejszego przeregulowania, mniejszych oscylacji oraz krótszego czasu ustalenia.
W przypadku sygnału sterującego, wyniki, które daje regulator DMC są zdecydowanie gorsze. Sygnał sterujący przy skokach wartości zadanej wpada w bardzo duże oscylacje i gwałtownie się zmienia. Nie jest to z pewnością coś pożądanego. Aby temu zaradzić należało by najprawdopodobniej zwiększyć parametr λ, jednak w tym podpunkcie założone zostało, że ma wynosić 1. Wpływ tego parametru oraz tezę czy zwiększenie jego wartości poprawi wyniki zostaną zbadane w następnym podpunkcie.

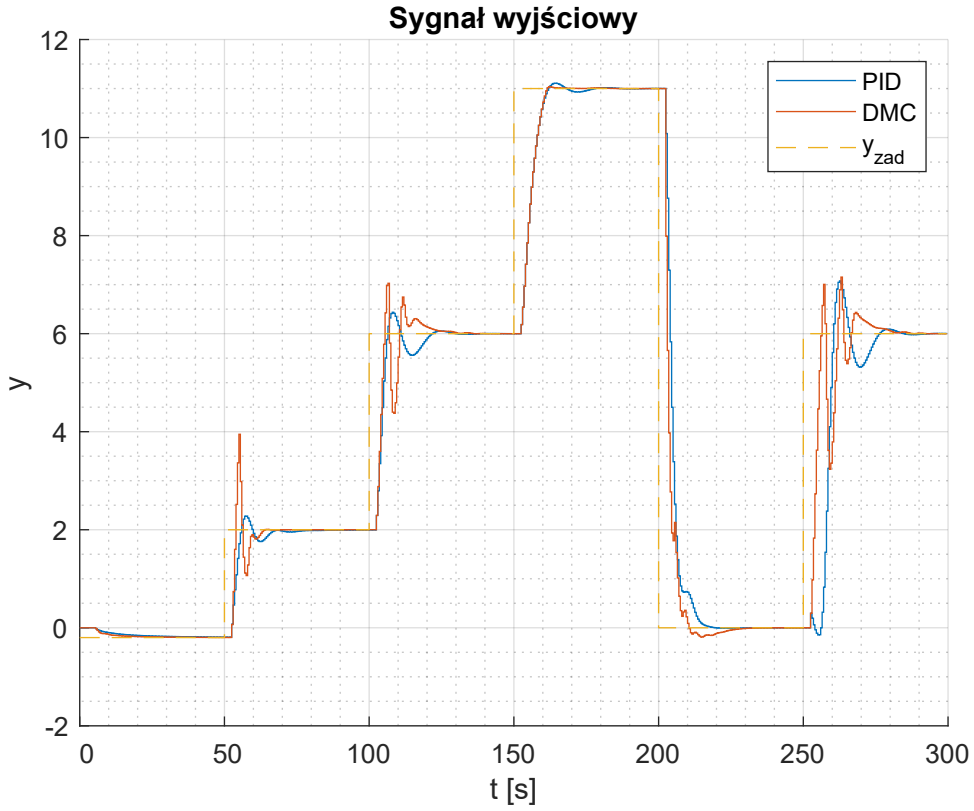
1.7.2. Trzy regulatory lokalne



Rys. 1.11. Funkcje przynależności dla regulatorów PID i DMC



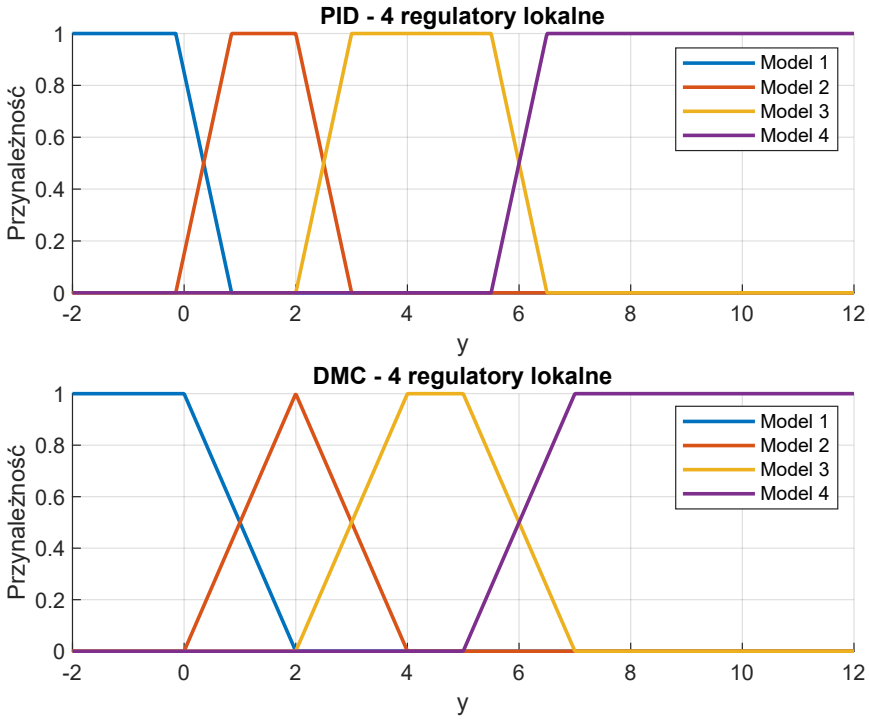
Rys. 1.12. Sygnały wejściowe regulatorów rozmytych, 3 regulatory lokalne



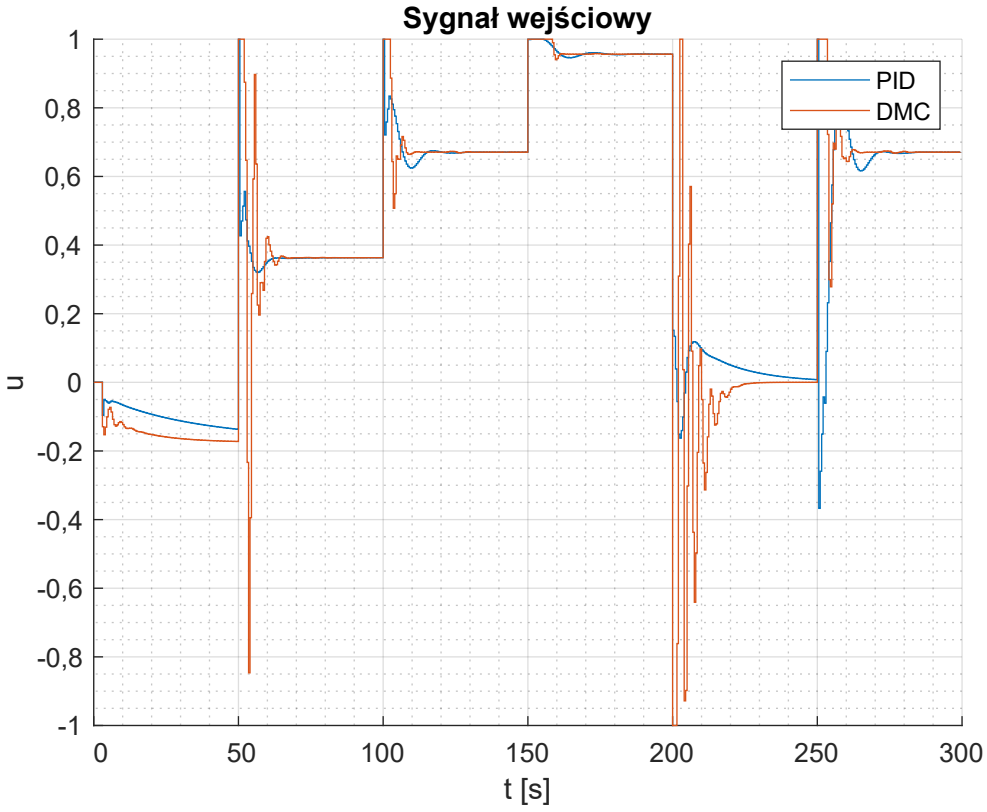
Rys. 1.13. Sygnały wyjściowe regulatorów rozmytych, 3 regulatory lokalne

Prawdę mówiąc, wyniki są nieznacznie lepsze, najbardziej widoczne jest to przy trzecim skoku wartości zadanej, w tym przypadku oba regulatory (DMC lekko lepiej) radzą sobie dobrze z osiągnięciem pożądanej wartości bez większego przeregulowania i oscylacji. DMC w dalszym ciągu generuje większe przeregulowanie i oscylacje, ale ponownie - spowodowane jest to gwałtownymi zmianami sygnału sterującego.

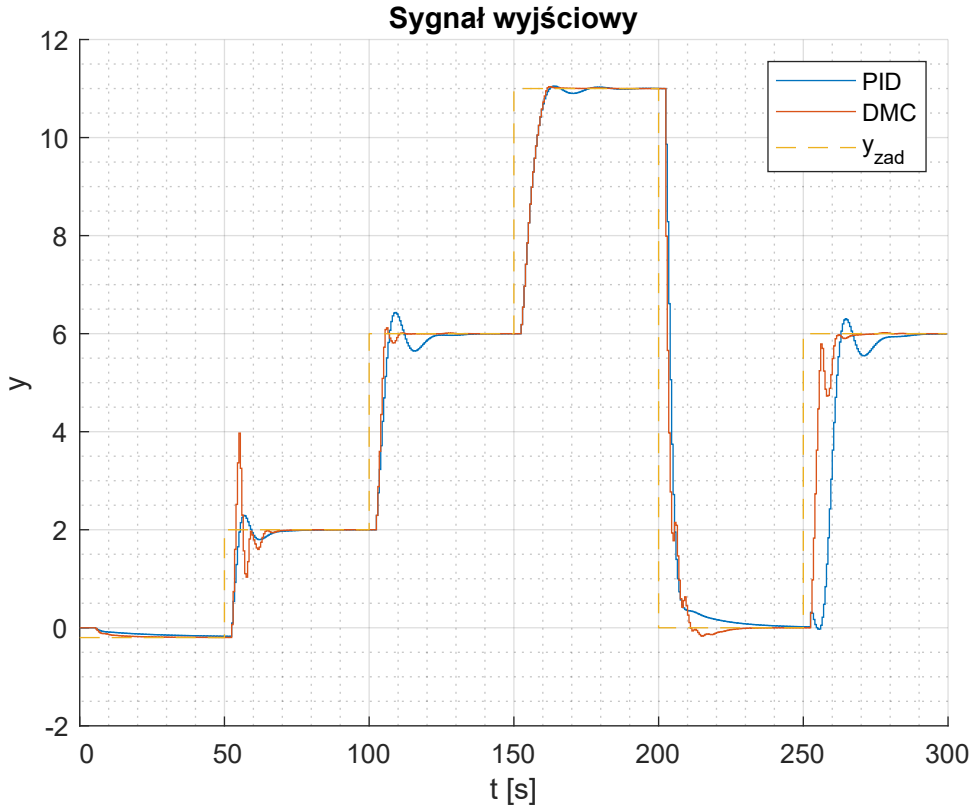
1.7.3. Cztery regulatory lokalne



Rys. 1.14. Funkcje przynależności dla regulatorów PID i DMC



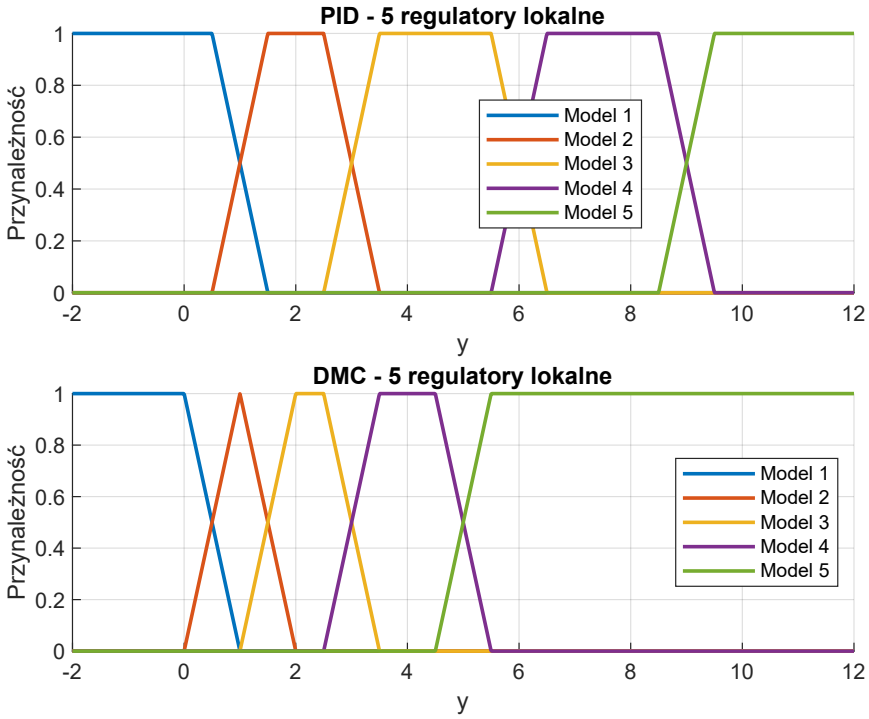
Rys. 1.15. Sygnały wejściowe regulatorów rozmytych, 4 regulatory lokalne



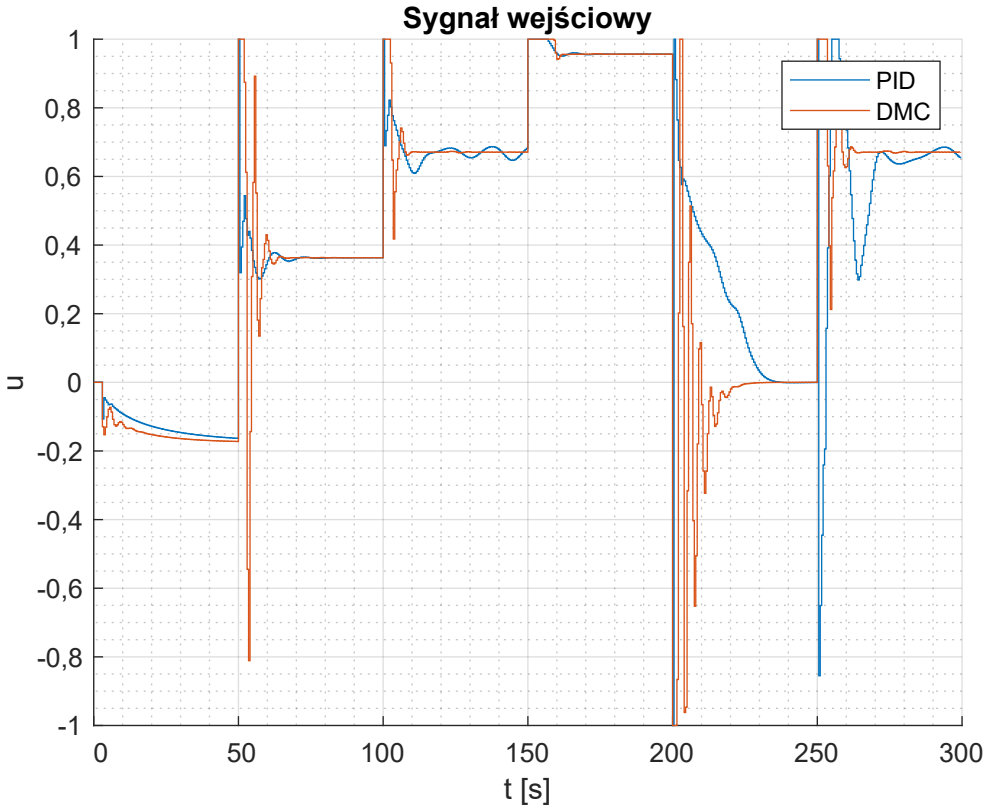
Rys. 1.16. Sygnały wyjściowe regulatorów rozmytych, 4 regulatory lokalne

Cztery regulatory lokalne przyniosły już zauważalną poprawę regulacji pod względem jakości. Regulator PID generuje nieznaczne przeregulowania i małe oscylacje, ale szybko gasnące. Widać, że DMC słabo sobie radzi przy pierwszym skoku wartości zadanej, jest to obszar, w którym obiekt jest najbardziej nieliniowy, stąd tak duże przeregulowanie. Mimo to, DMC reguluje szybciej, a przy następnych skokach też zdecydowanie dokładniej, praktycznie nie generując przeregulowania.

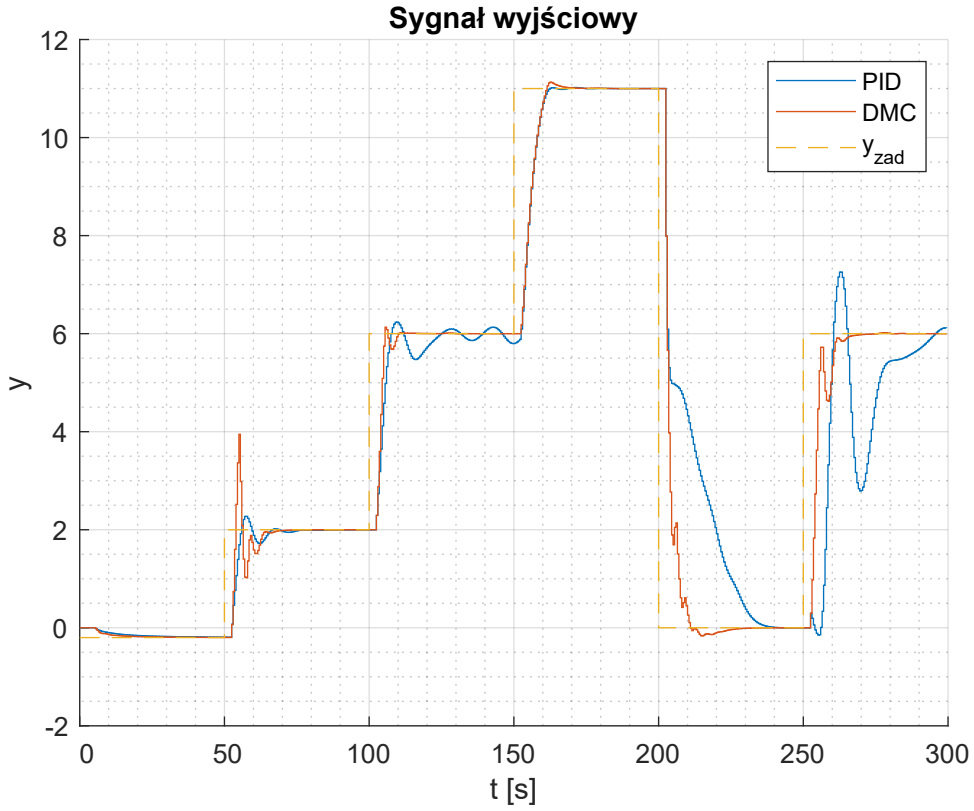
1.7.4. Pięć regulatorów lokalnych



Rys. 1.17. Funkcje przynależności dla regulatorów PID i DMC



Rys. 1.18. Sygnały wejściowe regulatorów rozmytych, 5 regulatory lokalne



Rys. 1.19. Sygnały wyjściowe regulatorów rozmytych, 5 regulatory lokalne

Wyniki dla DMC są niemalże identyczne, co dla 4 regulatorów lokalnych. Z kolei w przypadku regulatora PID widać znaczne pogorszenie się regulacji. Co ciekawe, przetestowano dobór różnych centr rozmycia i przedstawione powyżej wyniki są najlepsze, jakie udało się uzyskać. Dla 5 regulatorów lokalnych, regulator PID zaczyna łatwo wpadać w oscylacje, wielokrotnie zmiany centr rozmycia przynosiły sytuację, w której nawet nie można było mówić już o regulacji - sygnał wyjściowy zdawał się być losowy. Można dojść do wniosku, że w przypadku regulatora PID większa liczba regulatorów lokalnych niekoniecznie skutkuje zawsze lepszą jakością regulacji - istnieje pewna granica, po przekroczeniu której jakość pogarsza się diametralnie.
W przypadku spojrzenia na wyniki regulacji pod kątem ilościowym uzyskane błędy kwadratowe umieszczone zostały w tabeli poniżej:

Regulatory lokalne	**PID**	**DMC**
2	1785	1408
3	1891	1451
4	1885	1414
5	2359	1415

Tab. 1.5. Błąd kwadratowy regulatorów PID i DMC w zależności od ilości regulatorów lokalnych

Wyniki mogą być trochę zaskakujące. W przypadku regulatora PID najlepszą regulacją pod względem ilościowym okazał się regulator z 2 regulatorami lokalnymi. Mimo, że jakościowo, najlepsze okazały się regulatory z 4 regulatorami lokalnymi, dla obu algorytmów: PID i DMC, to pod względem ilościowym sytuacja jest odmienna. W przypadku regulatora DMC błąd tak naprawdę był porównywalny dla każdego przypadku, ale ponownie - najniższy był dla pierwszego regulatora z 2 regulatorami lokalnymi. Należy jednak pamiętać, że opieranie się jedynie na minimalizacji uchybu, nie jest najlepszym podejściem w przypadku strojenia regulatorów. Mogą się zdarzyć chociażby sytuacje, że wejście obiektu w oscylacje przyniesie najlepsze wyniki pod

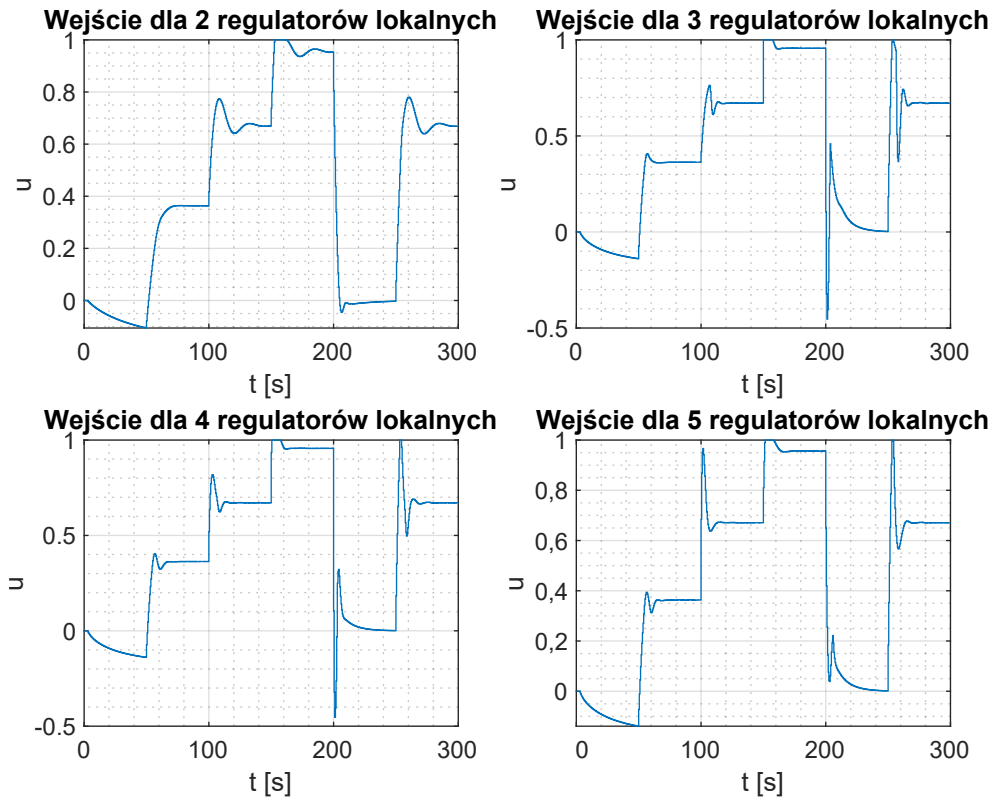
tym względem. Widać to także na opisywanym przykładzie - regulatory z 4 punktami rozmycia generowały najmniejsze przeregulowanie oraz najmniejsze oscylacje - jednak pod względem ilościowym nie wypadały najlepiej. Odpowiednie podejście należy więc dostosowywać indywidualnie do spotkanego problemu oraz do przyjętych przez nas kryteriów.

1.8. Dobór parametru λ

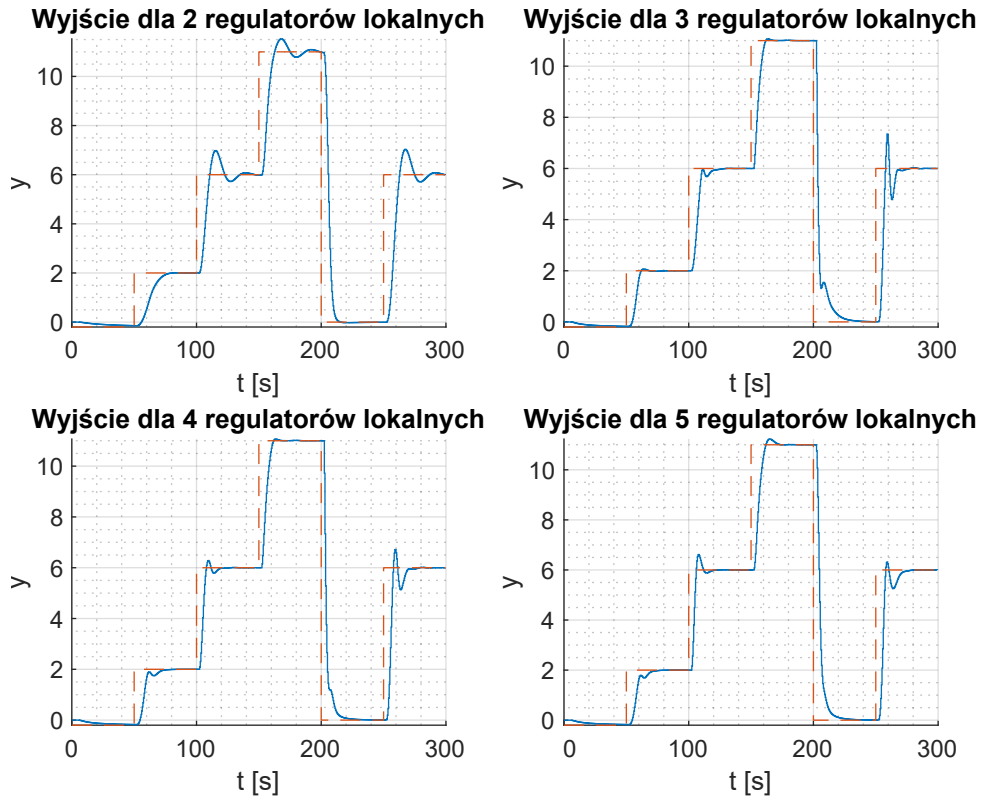
W poniższych symulacjach stosowane były te same funkcje przynależności, które w poprzednim podpunkcie. Podobnie z trajektorią wartości zadanej.
Testowano eksperymentalnie różne wartości parametru λ. Założono, że trzeba zwiększyć jego wartość, aby sygnał sterujący nie zmieniał się tak gwałtownie. To również powinno zmniejszyć przeregulowanie i oscylacje sygnału wyjściowego, jednak może zwiększyć czas ustalenia. Trudno eksperymentalnie dobrać dokładne wartości tego parametru, zwłaszcza przy większej liczbie regulatorów lokalnych. Kierowano się prostą zasadą - jeśli występuje przeregulowanie to należy zwiększyć parametr lambda regulatora lokalnego, który ma największą przynależność w tym zakresie wartości wyjściowych.
Parametry lambda kolejnych regulatorów rozmytych dobrano następująco:

— Dwa regulatory lokalne: $[5000, 5000]$
— Trzy regulatory lokalne: $[1000, 500, 100]$
— Cztery regulatory lokalne: $[1000, 500, 1000, 100]$
— Pięć regulatorów lokalnych: $[1000, 500, 100, 500, 1000]$

Otrzymane wyniki:



Rys. 1.20. Sygnały wejściowe regulatorów rozmytych DMC



Rys. 1.21. Sygnały wyjściowe regulatorów rozmytych DMC

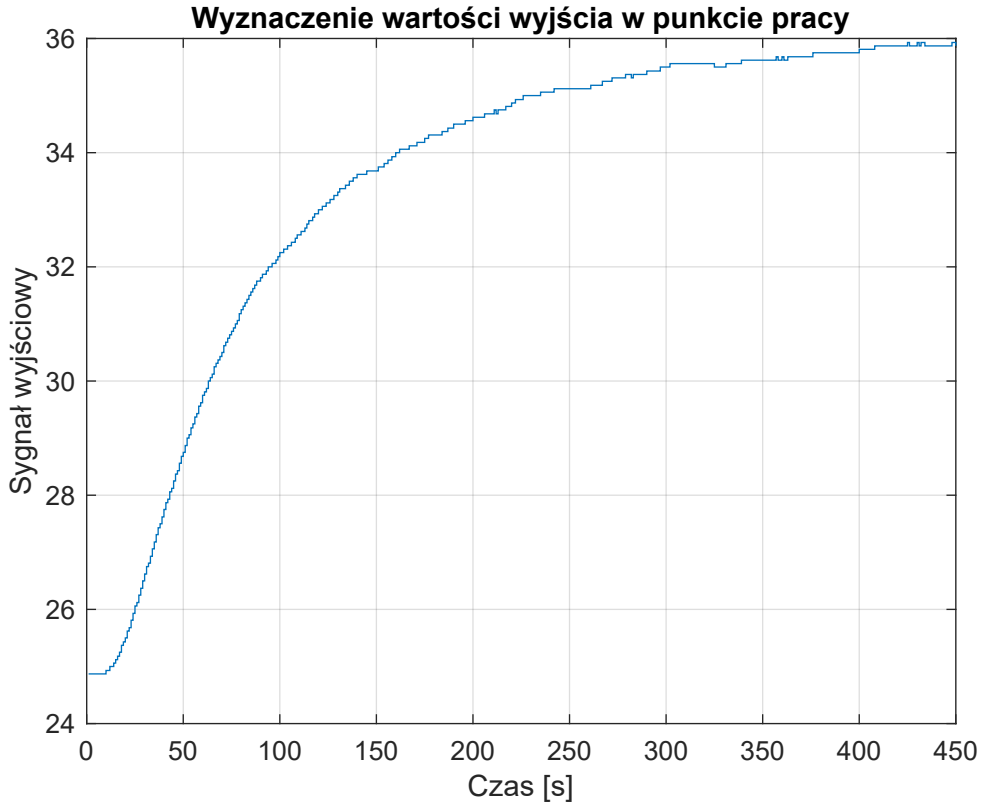
Założona teza sprawdziła się - udało się zredukować gwałtowne oscylacje o dużej amplitudzie sygnału sterującego. Przekłada się to na dokładniejszą regulację i zdecydowanie mniejsze przeregulowanie - zwłaszcza przy pierwszym skoku, z którym regulator DMC radził sobie najgorzej. Przekłada się to jednak na dłuższy czas ustalenia - co z kolei powoduje, że ostatecznie błędy kwadratowe zwiększyły się i, idąc kolejno ze wzrastającą liczbą regulatorów lokalnych, wynoszą odpowiednio: $2330, 1693, 1632, 1863$. Mimo to, zwiększenie znacznie parametru λ przyniosło świetne skutki - regulacja, lekko wolniejsza, jest dokładniejsza. Ponadto sygnał sterujący nie oscyluje gwałtownie, a to jest najważniejszym aspektem tej poprawy - w rzeczywistych obiektach, takie przebiegi, jakie widzimy chociażby na rys. 1.9 w przypadku regulatora DMC są niebezpieczne dla obiektu.

2. Laboratorium

2.1. Sterowanie i komunikacja ze stanowiskiem grzewczo-chłodzącym przy użyciu programu MATLAB oraz określenie punktu pracy

Dla zespołu o numerze 15 moc grzałki w punkcie pracy ma wartość 25.

W celu określenia pomiaru temperatury w punkcie pracy ustawiono wartości sygnałów G_1 na 25 oraz W_1 na 50 i odczekano do ustabilizowania się pomiaru.



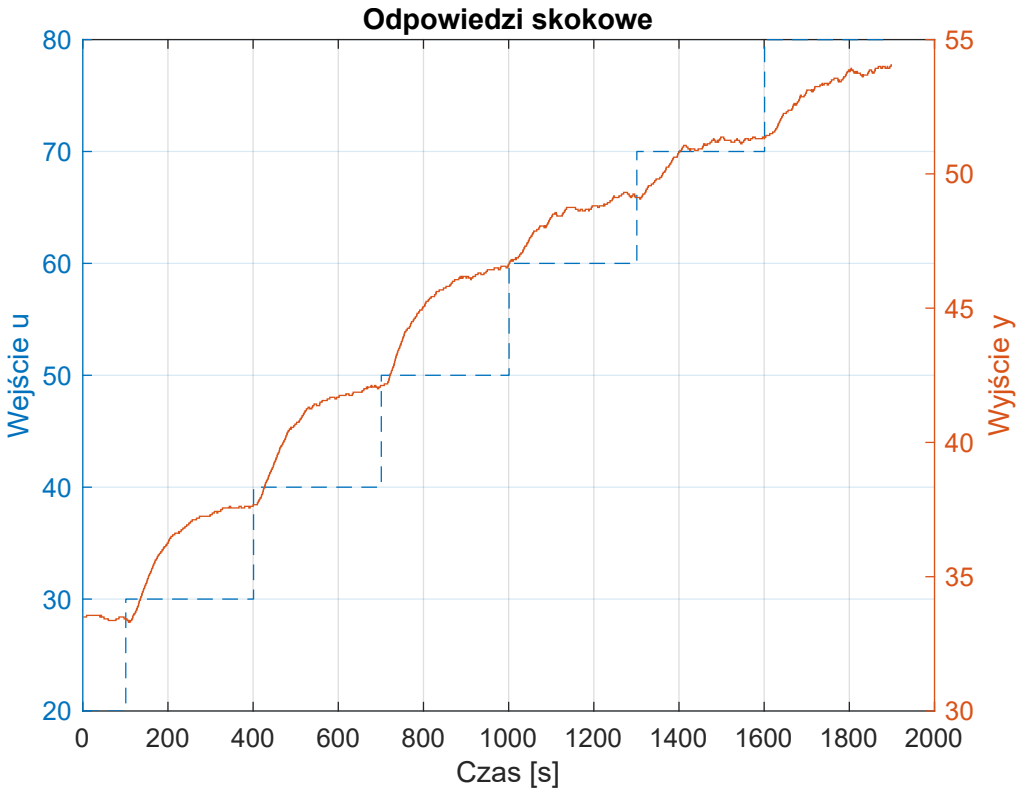
Rys. 2.1. Wykres pomiaru temperatury w punkcie pracy $G_1 = 25$

Temperatura T_{pp} po kilku minutach ustabilizowała się w okolicach wartości 36 °C. W kolejnych częściach laboratorium właśnie ta wartość będzie traktowana jako warunek początkowy, w szczególności przy wyliczaniu trajektorii wartości zadanych.z

2.2. Badanie liniowości obiektu

2.2.1. Odpowiedzi skokowe

W celu zbadania liniowości obiektu przeprowadzono dla szeregu wartości sterowania od 20 do 80 co 10, aż do ustabilizowania się wyjścia y obiektu i zostało przedstawione to na wykresie 2.2. Ta symulacja pozwoliła na późniejsze wyznaczenie charakterystyki statystycznej obiektu przedstawionej na wykresie 2.3.



Rys. 2.2. Wykres odpowiedzi skokowych dla kolejnych wartości u zwiększanych o 10

2.2.2. Charakterystyka statyczna

Rys. 2.3. Charakterystyka statyczna obiektu

Charakterystyka statyczna została wyznaczona poprzez zapisanie ostatniej wartości wyjścia osiągniętej przez obiekt dla danego sterowania. Założono, że 300s jest wystarczające na ustabilizowanie się wartości wyjścia. Jak można zauważyć na wykresie 2.3 badany obiekt nie jest w tym przypadku liniowy co uniemożliwia wyznaczenie jego wzmocnienia, ponieważ zależy ono od wartości wyjścia obiektu. Co warte zauważenia ta charakterystyka statyczna przypomina połączone ze sobą 2 charakterystyki liniowe.

2.3. Regulatory konwencjonalne

2.3.1. Konwencjonalny regulator PID

Do przeprowadzenie eksperymentów został wykorzystany regulator PID z pierwszych zajęć, a jego implementację w MATLAB przedstawiono poniżej. Również parametry zostały wykorzystane te same co na pierwszych zajęciach i przedstawiono je w tabeli 2.1.

Tab. 2.1. Parametry konwencjonalnego regulatora PID

Parametr	Wartość
K	11,0
T_i	80,0
T_D	0,3

```
clear;
addpath ('D:\ SerialCommunication ') ; % add a path
addpath('Funkcje');
initSerialControl COM6 % initialise com port

wykresy_online_konfiguracja;

kk = 3000;

Tpp = 36;

yzad(1:300) = Tpp; yzad(301:600) = Tpp+5;
yzad(601:900) = Tpp+15; yzad(901:kk) = Tpp;

umin = 0; umax=100;

% nastawy PID są takie jak najlepsze z lab 1
K = 11;
Ti = 80;
Td = 0.3;
[r2, r1, r0] = pid_offline(K,Ti,Td,1);

% inicjlizacja
u(1:kk)=0; y(1:kk)=0; e(1:kk)=0;

% warunki początkowe
u(1:2)=25; y(1:2)=Tpp;


% główna pętla symulacyjna
```

```
for k=3:kk
    % symulacja obiektu
    y(k) = readMeasurements(1);

    % uchyb regulacji
    e(k)=yzad(k)-y(k);

    % sygnał sterujący regulatora PID
    u(k)=r2*e(k-2)+r1*e(k-1)+r0*e(k)+u(k-1);

    if u(k) < umin
        u(k) = umin;
    elseif u(k) > umax
        u(k) = umax;
    end

    sendControls (1, 50) ; % W1
    sendNonlinearControls(u(k)); % G1

    % aktualizacja wykresu
    wykresy_online_aktualizacja;

    waitForNewIteration () ; % wait for new iteration
end
```

2.3.2. Konwencjonalny regulator DMC

Do przeprowadzenie eksperymentów został wykorzystany regulator DMC z pierwszych zajęć, a jego implementację w MATLAB przedstawiono poniżej. Wykorzystano wersję oszczędną regulatora DMC. Również parametry zostały wykorzystane te same co na pierwszych zajęciach i przedstawiono je w tabeli 2.2.

Tab. 2.2. Parametry konwencjonalnego regulatora DMC

Parametr	Wartość
N	200
N_u	5
D	700
λ	0,2

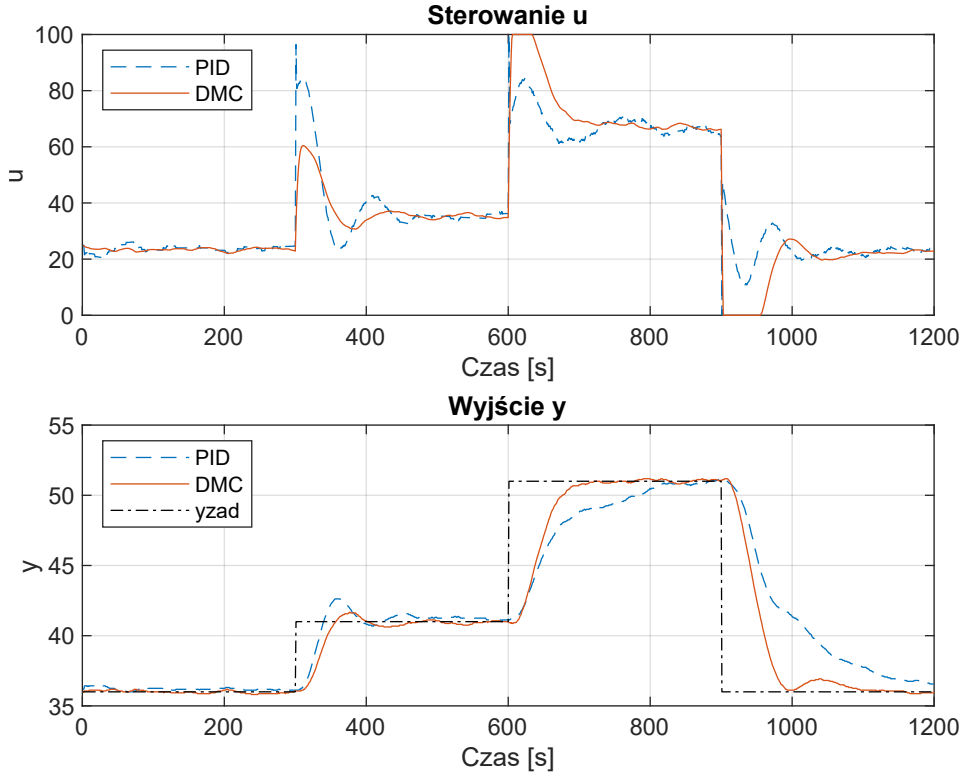
```
clear;
addpath ('D:\ SerialCommunication ') ; % add a path
addpath ('Funkcje')
initSerialControl COM6 % initialise com port

wykresy_online_konfiguracja;

kk=3000; % koniec symulacji
umin = 0; umax=100;

Tpp = 36;
```

```
yzad(1:300) = Tpp; yzad(301:600) = Tpp+5;
yzad(601:900) = Tpp+15; yzad(901:kk) = Tpp;

% nastawy DMC są takie jak najlepsze z lab 1
N = 200;
Nu = 5;
D = 500;
lambda = 0.2;

% Odpowiedź skokowa zdyskretyzowanego systemu
ys = dmc_odp_jedn(750);

[ke,ku] = DMC_offline(ys,N,Nu,lambda,D);

% Inicjalizacja
u(1:kk)=0; y(1:kk)=0; e(1:kk)=0;
delta_u_p(1:D-1)=0; % Przeszłe przyrosty u

% Warunki początkowe
u(1)=25; y(1)=35.93;

% Główna pętla symulacyjna
for k=2:kk
    % Symulacja obiektu
    y(k) = readMeasurements (1);

    % Uchyb regulacji
    e(k)=yzad(k)-y(k);

    % Obliczenie przyrostu sygnału sterującego DMC
    delta_u = ke * e(k) - ku * delta_u_p';

    % Ograniczenia
    if u(k-1)+delta_u < umin
        delta_u = umin-u(k-1);
    elseif u(k-1)+delta_u > umax
        delta_u = umax-u(k-1);
    end

    % Aktualizacja sygnału sterującego
    u(k)=u(k-1)+delta_u;

    % Aktualizacja przeszłych przyrostów sterowania
    for n=D-1:-1:2
        delta_u_p(n) = delta_u_p(n-1);
    end
    delta_u_p(1) = delta_u;

    sendControls (1, 50) ; % W1
    sendNonlinearControls(u(k)); % G1
```

```
    wykresy_online_aktualizacja;

    waitForNewIteration () ;
end
```

2.3.3. Porównanie regulatorów konwencjonalnych

W celu porównania działania regulatorów konwencjonalnych w przypadku obiektu nieliniowego wykonano symulację dla toru wartości zadanej wyjścia obiektu y_{zad}: T_{pp}, $T_{pp}+5$, $T_{pp}+15$, T_{pp}. Na ustabilizowanie się wyjścia obiektu dla każdego skoku y_{zad} przeznaczono po $300s$ co wystarczyło aby, regulator ustabilizował wyjście obiektu na wartości zadanej. Niestety po za jednym wyjątkiem dla regulatora PID przy spadku wartości zadanej o $15°C$, jednakże z powodu ograniczonej ilości czasu zdecydowano, że taki pomiar wystarczył do wystarczającego porównania regulatorów.



Rys. 2.4. Porównanie konwencjonalnych regulatorów DMC oraz PID dla zadanej trajektorii y_{zad}

Na podstawie wykresu 2.4 można jednoznacznie stwierdzić, że regulator DMC lepiej poradził sobie ze zmianą obiektu na nieliniowy mimo korzystania z odpowiedzi skokowej dla poprzedniego liniowego obiektu. Regulator DMC w każdym przypadku osiąga mniejsze przeregulowanie, jeśli do niego dochodzi oraz czas ustalenia wartości wyjścia jest znacznie krótszy co najlepiej widać dla skoków przy których wyjście obiektu przechodzi pomiędzy dwoma częściami charakterystyki, które są osobno w przybliżeniu liniowe. Aby jeszcze lepiej pokazać, że regulator DMC lepiej poradził sobie z tym problemem regulacji obliczono błąd kwadratowy dla trajektorii wartości zadanej y_{zad} pomijając 100 pierwszych próbek, ponieważ zmieniający się punkt pracy obiektu między eksperymentami wynikający między innymi ze zmiany temperatury otoczenia zakłócał otrzymywane wyniki. To podejście zastosowano również podczas liczenia błędów dla wszystkich regulatorów rozmytych. Wskaźnik jakości przedstawiono w tabeli 2.3. Z wartości błędu również

wynika, że regulatora DMC działa lepiej, ponieważ błąd jest ponad 5 krotnie mniejszy niż dla regulatora PID.

Tab. 2.3. Wskaźnik jakości regulatorów konwencjonalnych

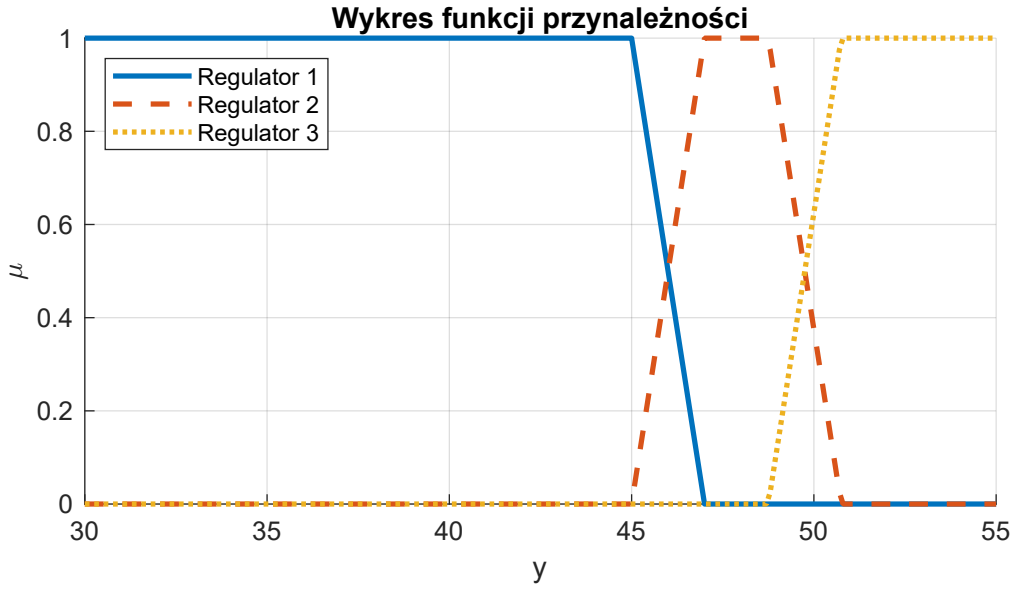
Regulator	Błąd kwadratowy
PID	692,13
DMC	124,96

2.4. Rozmyty regulator PID

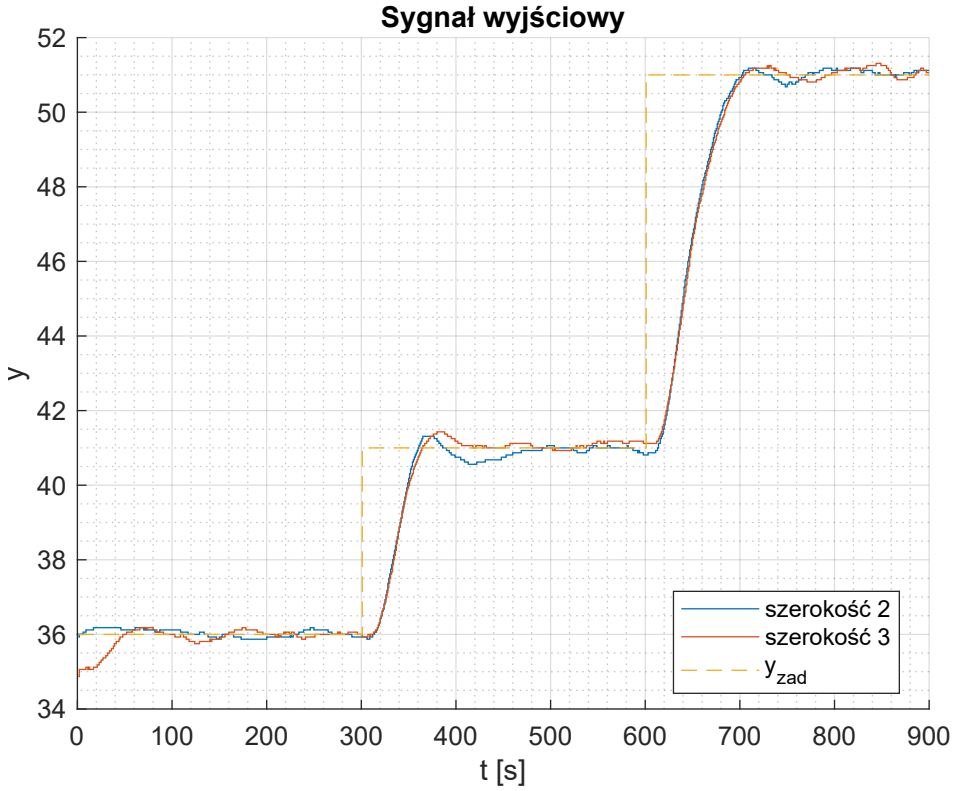
2.4.1. Dobór funkcji przynależności

Na podstawie analizy charakterystyki oraz otrzymanego polecenia zadania zdecydowano się na 3 lokalne regulatory PID, a do rozmywania, czyli obliczania współczynników przynależności wyliczonego sygnału sterującego przez konkretny regulator lokalny zdecydowano się użyć funkcji trapezowej, której implementację przedstawiono poniżej. Wynikło to z tego, że charakterystyka statyczna obiektu nie jest nadmiernie skomplikowana, bo składa się w przybliżeniu z 2 prostych. Rozmywanie przeprowadzono względem wartości wyjścia obiektu, ponieważ to od niej niego zależy nieliniowość w charakterystyce obiektu. Przedziały wartości wyjścia dla których dany regulator lokalny jest jedynym uwzględnianym przyjęto w taki sposób, że dla, patrząc rosnąco względem wartości wyjścia na charakterystyce statycznej, pierwszej prostej do punktu załamania przyjęto pierwszy regulator, a dla drugiej 2 regulatory, ponieważ pomimo uznania tej charakterystyki w przybliżeniu za liniową dane były mniej dokładne, w związku z czym 1 z tych regulatorów umieszczono w przedziale, gdy wartość wyjścia była w okolicach punktu załamania, a kolejny umieszczono bliżej końca zakresu badanych wartości wyjścia obiektu. Dodatkowo, aby zapewnić możliwie dużą płynność w zakresie, gdy następuję przełączenie regulatorów zdecydowano się na przedział o szerokości 3, gdzie dochodziło do stopniowej zamiany między regulatorami lokalnymi. Przetestowano również przedział o szerokości 2 jednakże dla niepełnej trajektorii zmian wartości zadanej dla regulatora DMC, co przedstawiono na wykresie 2.6 gdzie zauważono, że oczekiwana płynność nie została uzyskana dla przedziału o szerokości 2.

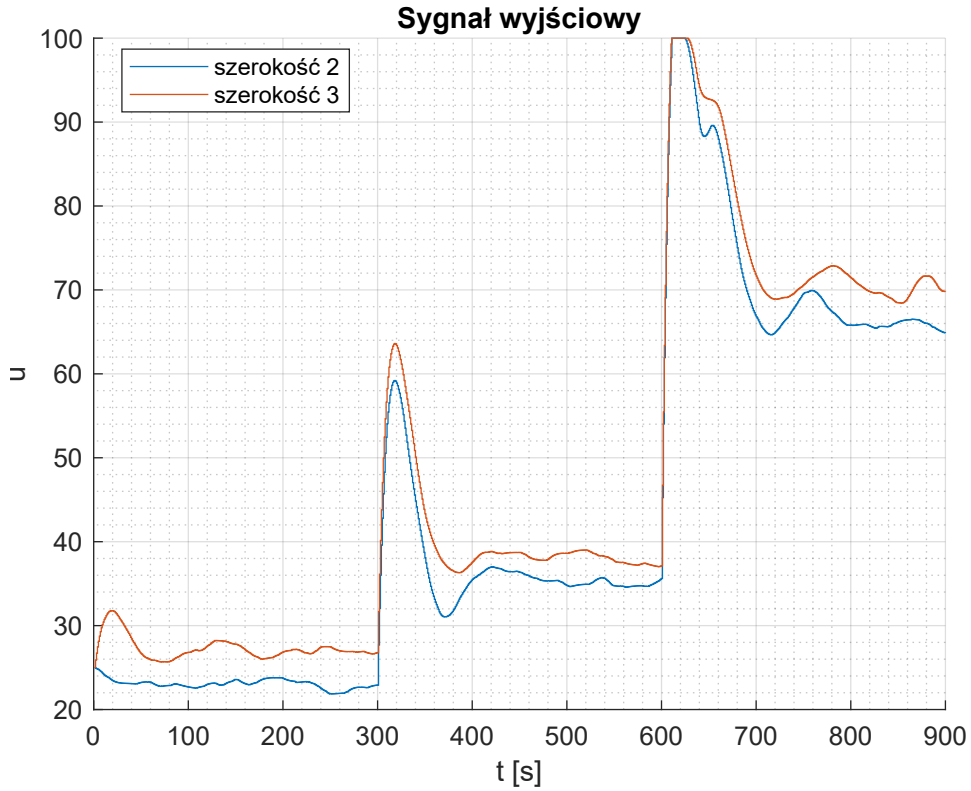
W ten sposób otrzymano funkcje przynależności przedstawione na wykresie 2.5. Te same funkcje zostały wykorzystane również w regulatorze DMC, ich implementację przedstawiono poniżej. Ostatecznie otrzymywane stopnie przynależności są znormalizowane.



Rys. 2.5. Wykres funkcji przynależności regulatorów lokalnych PID oraz DMC



Rys. 2.6. Porównanie działania regulatora PID o tych samych nastawach z inną szerokością przejściową funkcji przynależności



Rys. 2.7. Porównanie działania regulatora PID o tych samych nastawach z inną szerokością przejściową funkcji przynależności

```
function y = fun_trapezowa(x, a, b, c, d)
% FUN_TRAPEZOWA Oblicza wartość funkcji przynależności o kształcie
    % trapezowym lub prostokątnym prawostronnie domkniętym
    %
    % Argumenty:
    %   x - wartość, dla której obliczana jest przynależność.
    %   a - początek trapezu (lewy kraniec przedziału),
    %       gdzie funkcja przynależności wynosi 0.
    %   b - początek płaskiej części trapezu,
    %       gdzie funkcja przynależności zaczyna wynosić 1.
    %   c - koniec płaskiej części trapezu,
    %       gdzie funkcja przynależności zaczyna spadać z 1.
    %   d - koniec trapezu (prawy kraniec przedziału),
    %       gdzie funkcja przynależności wraca do 0.
    %
    % Zwraca:
    %   y - stopień przynależności wartości x
    %       do trapezoidalnego zbioru rozmytego.

    if x <= a || x > d
        y = 0;  % Poza zakresem trapezu
    elseif x >= b && x <= c
        y = 1;  % W płaskiej części trapezu
    elseif x > a && x < b
        y = (x - a) / (b - a);  % Wzrost funkcji
```

```
    else  % x > c && x < d
        y = (d - x) / (d - c);  % Spadek funkcji
    end
end
```

```
function [mi] = fun_przyn_trap(z, z_vals, z_switch)
% FUN_PRZYN_TRAP Oblicza wartości funkcji przynależności trapezowej
% dla danego parametru `z`, wykorzystując zadane punkty centralne `z_vals`.
% Funkcja modeluje stopień przynależności do zbiorów rozmytych w oparciu
% o równomiernie rozmieszczone punkty centralne (modele lokalne).
%
% Argumenty wejściowe:
%   z        - Wartość wejściowa (np. sterowanie lub wyjście),
%              dla której obliczana jest przynależność
%              do każdego zbioru rozmytego.
%   z_vals   - Wektor wartości centralnych
%              dla każdego zbioru rozmytego
%              (środków funkcji trapezowych).
%   z_switch - Długość strefy przejściowej
%              (część nachylona trapezu),
%              kontrolująca szerokość funkcji przynależności.
%
% Zwraca:
%   mi - Wektor o długości równej liczbie modeli lokalnych,
%        zawierający stopnie przynależności
%        wartości `z` do każdego modelu lokalneg

    % Liczba Modeli
    numModels=length(z_vals);

    % Ustawienie wartości 'z' poza zakresem na wartość graniczną
    if z < min(z_vals)
        mi = [1 zeros(1, numModels-1)];
    elseif z > max(z_vals)
        mi = [zeros(1, numModels-1) 1];
    else
        % Inicjalizacja wektora mi
        mi = zeros(1, numModels);
        for i = 1:numModels
            % Ustawienie punktów 'a', 'b', 'c', 'd' dla funkcji trapezowej
            if i == 1
                % Lewy kraniec
                a = z_vals(1)-1;
                b = z_vals(1);
                c = z_vals(i)+(z_vals(i+1)-z_vals(i))/2-z_switch/2;
                d = z_vals(i)+(z_vals(i+1)-z_vals(i))/2+z_switch/2;
            elseif i == numModels
                % Prawy kraniec
                a = z_vals(i)-(z_vals(i)-z_vals(i-1))/2-z_switch/2;
                b = z_vals(i)-(z_vals(i)-z_vals(i-1))/2+z_switch/2;
                c = z_vals(i);
```

```
                d = z_vals(i)+1;
            else
                % Środkowe modele
                a = z_vals(i)-(z_vals(i)-z_vals(i-1))/2-z_switch/2;
                b = z_vals(i)-(z_vals(i)-z_vals(i-1))/2+z_switch/2;
                c = z_vals(i)+(z_vals(i+1)-z_vals(i))/2-z_switch/2;
                d = z_vals(i)+(z_vals(i+1)-z_vals(i))/2+z_switch/2;
            end

            % Wyznaczenie przynależności
            mi(i) = fun_trapezowa(z, a, b, c, d);
        end
    end
end
```

2.4.2. Implementacja rozmytego regulatora PID

Implementacja regulatora rozmytego PID rożni się tylko częściowo od konwencjonalnego, ponieważ wymaga ona uwzględnienia faktu obecności kilku regulatorów lokalnych zamiast jednego. W związku z tym w części offline czyli przed rozpoczęciem eksperymentu wyliczano parametry r_2, r_1 i r_0 dla każdego z 3 regulatorów lokalnych na podstawie odrębnych parametrów dla każdego z nich. Z kolej w części online dla każdej chwili k wyliczano stopnie przynależności regulatorów lokalnych dla aktualnej wartości wyjścia obiektu y, a następnie obliczano wartości sterowania dla każdego lokalnego regulatora PID i wymnażano je przez jego stopień przynależności, po czym tak przeliczone wartości sterowania sumowano co dawało ostateczną wartość sterowania przekazywaną do obiektu. Implementacja rozmytego regulatora PID znajduje się poniżej.

```
clear;
addpath ('D:\ SerialCommunication ') ; % add a path
addpath('Funkcje');
addpath('funkcje_przynaleznosci')

% initSerialControl COM4 % initialise com port
wykresy_online_konfiguracja;

kk = 3000;

Tpp = 40;

yzad(1:300) = Tpp; yzad(301:600) = Tpp+5;
yzad(601:900) = Tpp+15; yzad(901:kk) = Tpp;

umin = 0; umax=100;

% nastawy PID są takie jak najlepsze z lab 1
K = [11, 2, 3];
Ti = [80, 2, 3];
Td = [0.3, 2, 3];
r0(3,1)=0;
r1(3,1)=0;
r2(3,1)=0;
for i=1:3
```

```
    [r2(i), r1(i), r0(i)] = pid_offline(K(i),Ti(i),Td(i),1);
end

% punkty rozmycia
y_rozm=[44, 48, 51.5];

% inicjlizacja
u(1:kk)=0; y(1:kk)=0; e(1:kk)=0;

% warunki początkowe
u(1:2)=25; y(1:2)=32.68;


% główna pętla symulacyjna
for k=3:kk
    % symulacja obiektu
    y(k) = readMeasurements(1);

    % uchyb regulacji
    e(k)=yzad(k)-y(k);

    % obliczenie przynależności każdej reguły
    mi=fun_przyn_trap(y(k), y_rozm, 2);

    % sygnał sterujący regulatora PID
    u(k)=mi*(r2*e(k-2)+r1*e(k-1)+r0*e(k)+u(k-1));

    if u(k) < umin
        u(k) = umin;
    elseif u(k) > umax
        u(k) = umax;
    end

    sendControls (1, 50) ; % W1
    sendNonlinearControls(int64(u(k))); % G1

    % aktualizacja wykresu
    wykresy_online_aktualizacja;

    waitForNewIteration () ; % wait for new iteration
end
```

2.4.3. Strojenie rozmytego regulatora PID

Strojenie poszczególnych regulatorów lokalnych PID rozpoczęto od przeanalizowania działania konwencjonalnego regulatora PID, ponieważ jego działanie nie różni się od działania rozmytego regulatora PID o tych samych parametrach 2.4. Zauważono, że dla pierwszego regulatora lokalnego warto spróbować zmniejszyć przeregulowanie w związku z czym postanowiono zmniejszyć wzmocnienie K. W przypadku 2 i 3 regulatora lokalnego, które znajdują się w drugiej w przybliżeniu liniowej części charakterystyki statycznej, postanowiono zastosować takie same nastawy w każdym z rozpatrywanych przypadków. Zaczęto od nieznacznego zwiększenia wzmoc-

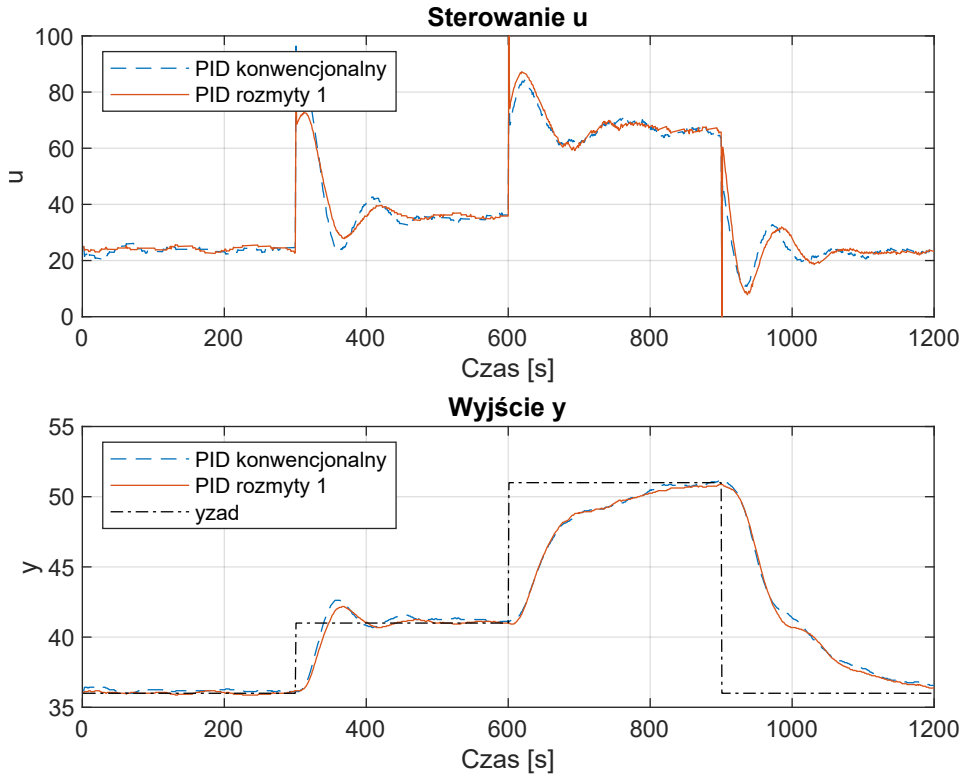
nienia tak aby przyspieszyć działanie regulatora. Ostateczne parametry przedstawiono w tabeli 2.5, a porównanie działania przedstawiono na wykresie 2.8

Tab. 2.4. Parametry rozmytego regulatora PID 0

Parametr	PID lokalny 1	PID lokalny 2	PID lokalny 3
K	11,0	11,0	11,0
T_i	80,0	80,0	80,0
T_D	0,3	0,3	0,3

Tab. 2.5. Parametry rozmytego regulatora PID 1

Parametr	PID lokalny 1	PID lokalny 2	PID lokalny 3
K	9,0	14,0	14,0
T_i	80,0	80,0	80,0
T_D	0,3	0,3	0,3

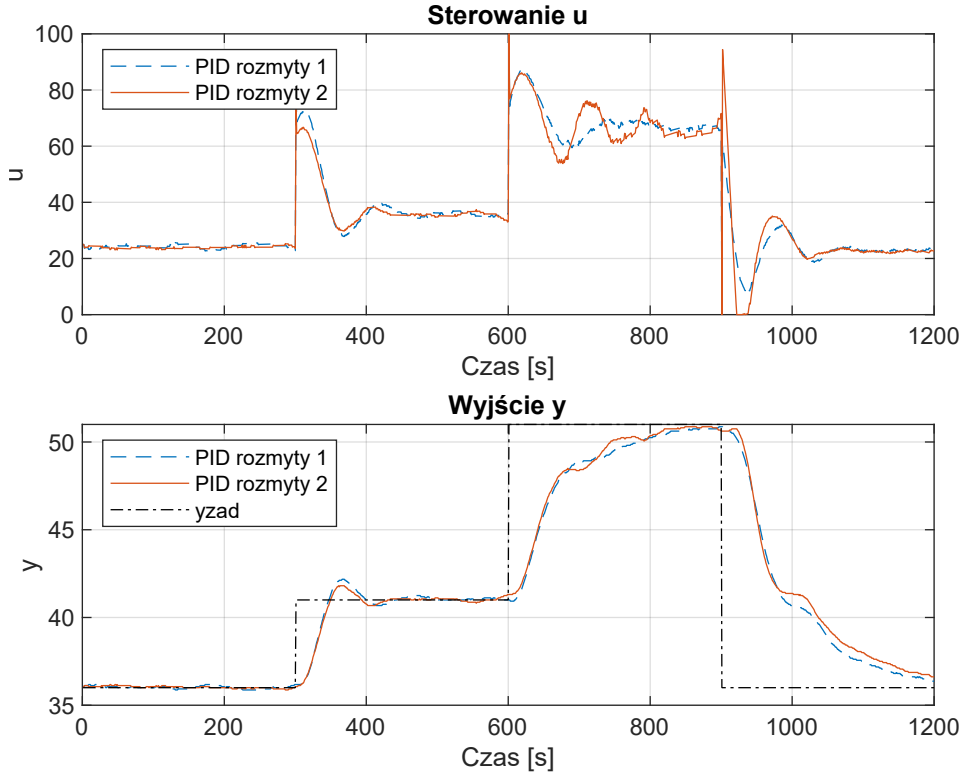


Rys. 2.8. Porównanie rozmytego regulatora PID 0 z rozmytym regulatorem PID 1

Na podstawie powyższych wyników 2.8, zaobserwowano poprawę jakości regulacji, jednakże otrzymany wynik pozostawał niewystarczająco zadowalający w związku z czym powtórzono procedurę zmiany parametrów. Dla regulatora 1 ponownie zmniejszono wzmocnienie, a w tym przypadku także zmniejszono wpływ całkowania, aby zmniejszyć oscylacje, za to w przypadku regulatorów lokalny 2 i 3 bardziej znacząco zwiększono wzmocnienie, oraz zwiększono wpływ całkowania. Otrzymane parametry zostały przedstawione w tabeli 2.6, a działanie regulatora na wykresie 2.9.

Tab. 2.6. Parametry rozmytego regulatora PID 2

Parametr	PID lokalny 1	PID lokalny 2	PID lokalny 3
K	8,0	22,0	22,0
T_i	90,0	70,0	70,0
T_D	0,3	0,3	0,3

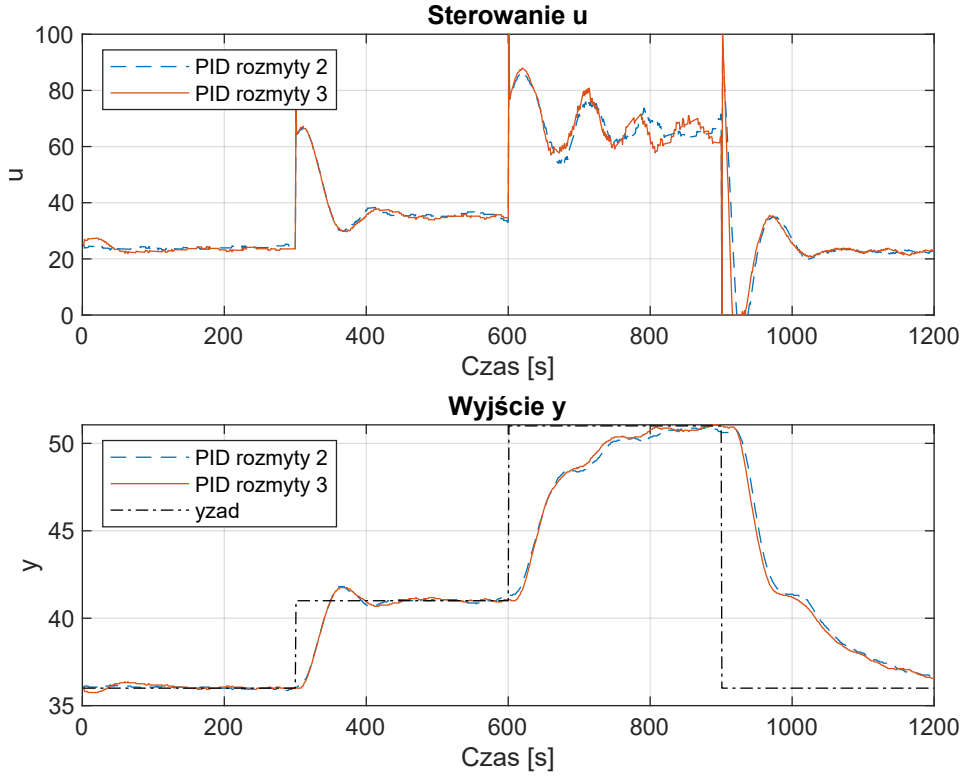


Rys. 2.9. Porównanie rozmytego regulatora PID 1 z rozmytym regulatorem PID 2

Analizując otrzymane wyniki na wykresie 2.9, zauważono poprawę regulacji w górę dla lokalnego regulatora 1, jednakże w warunkach pracy na laboratorium został przeoczony spadek jakości regulacji w przypadku skoków w dół, przez co uznano, że regulator pierwszy pozwala na zadowalającą jakość regulacji i pozostawiono jego parametry z tabeli 2.6 jako ostateczne, a skupiono się jedynie na zmianie parametrów regulatorów 2 i 3 lokalnego, zauważając, że poprawiła się jakość regulacji w obszarze ich działania ponownie zwiększono człon proporcjonalny oraz zwiększono wpływ członu całkującego i dobrane parametry dla ostatniego testu regulatora PID zapisano w tabeli 2.7, działanie tego regulatora przedstawiono na wykresie 2.10.

Tab. 2.7. Parametry rozmytego regulatora PID 3

Parametr	PID lokalny 1	PID lokalny 2	PID lokalny 3
K	8,0	30,0	30,0
T_i	90,0	60,0	60,0
T_D	0,3	0,3	0,3



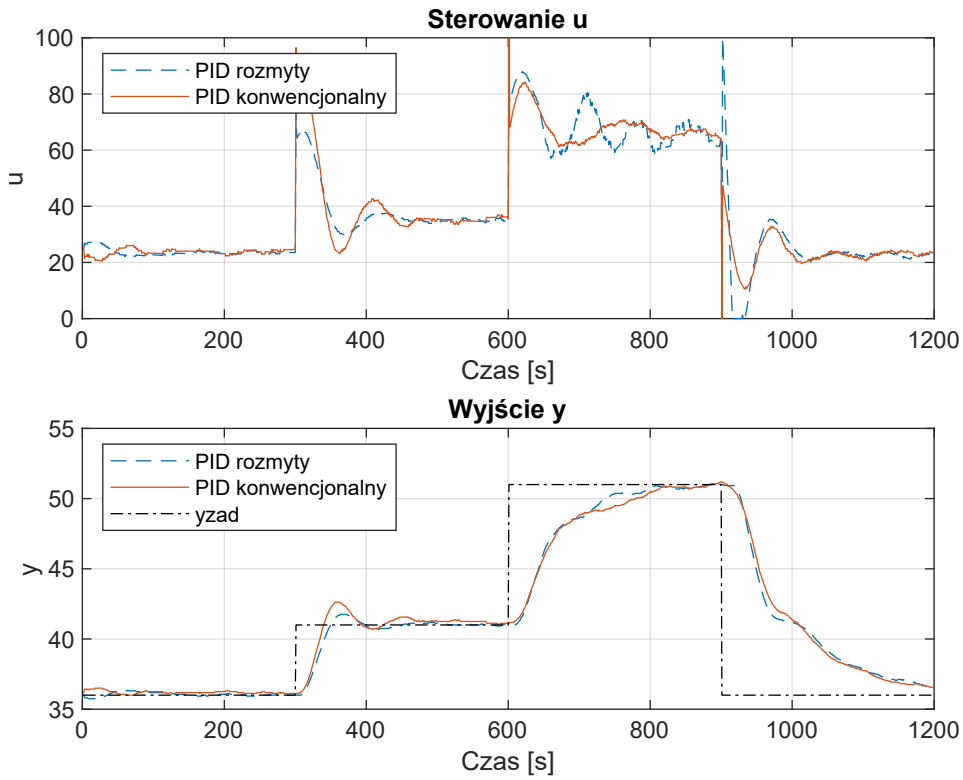
Rys. 2.10. Porównanie rozmytego regulatora PID 2 z rozmytym regulatorem PID 3

Patrząc na otrzymane wyniki na wykresie 2.10 zdecydowano, że lepiej poradził sobie 3 regulator rozmyty PID, więc uznano go za najlepszy i przyjęto do dalszej analizy porównawczej z pozostałymi regulatorami. Aby upewnić się, że podjęta została dobra decyzja policzono błąd kwadratowy jak w przypadku regulatorów konwencjonalnych czyli od chwili $t = 101$ do $t = 1200$ i błędy te dla każdego z 4 analizowanych regulatorów rozmytych PID przedstawiono w tabeli 2.8 i dopiero wtym momencie dostrzeżonon błąd popełniony przy strojeniu pierwszego regulatora lokalnego, na którego poprawę nie wystarczyło czasu. Zauważono go ponieważ znacznie mniejszy błąd otrzymano dla regulatora, który wydawało się wcześniej, że jest słabszy niż oczekiwano. Prawdopodobnie połączenie parametrów regulatora 1 lokalnego z tabeli 2.5 oraz parametrów regulatorów lokalnych 2 i 3 z tabeli 2.7 dałoby jeszcze lepszy wynik i to takie rozwiązanie powinno być dalej porównywane jednakże przez brak czasu nie udało się otrzymać wyników eksperymentu dla takiego regulatora w związku zz czym do porównania przyjęto regulator rozmyty 3 PID jednocześnie pamiętając o popełnionym błędzie.

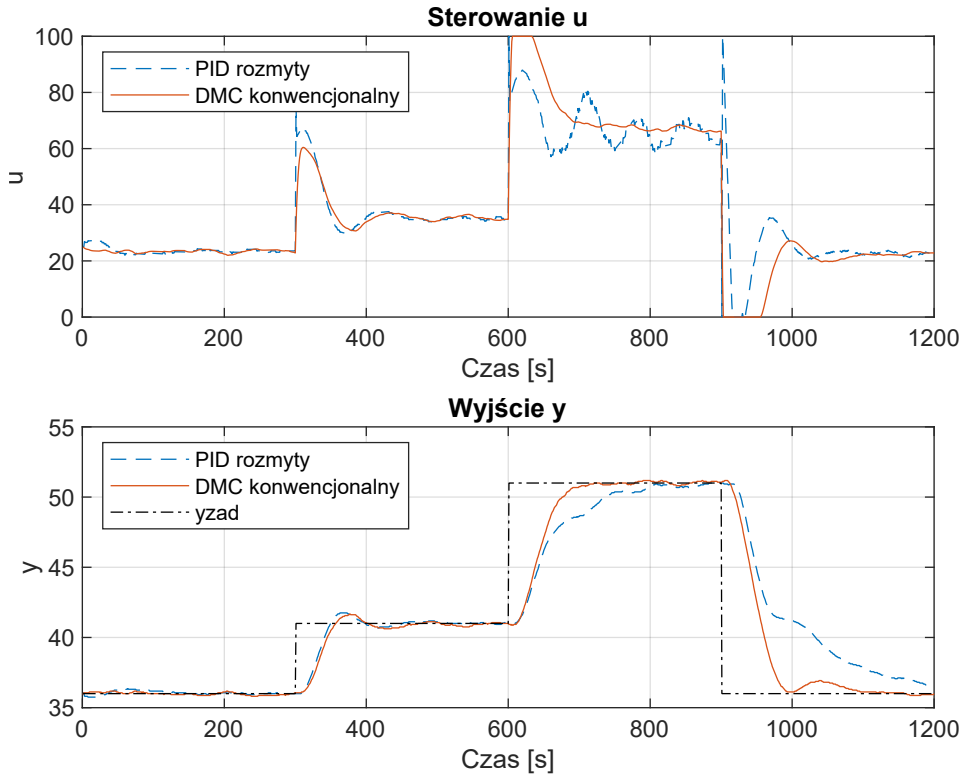
Tab. 2.8. Wskaźnik jakości regulatorów rozmytych PID

Regulator	Błąd kwadratowy
PID 0	692,13
PID 1	515,63
PID 2	639,16
PID 3	601,26

2.4.4. Porównanie rozmytego regulatora PID z regulatorami konwencjonalnymi



Rys. 2.11. Porównanie rozmytego regulatora PID z konwencjonalnym regulatorem PID



Rys. 2.12. Porównanie rozmytego regulatora PID z konwencjonalnym regulatorem DMC

Obserwując wyniki porównania przedstawione na wykresie 2.11 można stwierdzić, że lepiej sprawdzającym się do obiektów nieliniowych regulatorem PID jest rozmyty, a nie konwencjonalny, ponieważ pozwala on na dostosowanie parametrów dla konkretnego obszaru wartości wyjścia i/lub wejścia obiektu i daje znacznie lepsze rezultaty w regulacji, które pokazuje również błąd dla zadanej trajektorii przedstawiony w tabeli 2.9. Za to już w porównaniu z regulatorem konwencjonalnym DMC regulator rozmyty PID poradził sobie znacznie gorzej co wynika z błędu w przytoczonej wyże tableii oraz z wykresu 2.12, gdzie widać, że zajmuje mu znacznie dłużej ustaelenie wartości na wyjściu, oscylacje są znacznie większe w przypadku pierwszego skoku jak i przeregulowanie, jednakże najgorzej wypada wspomniany czas ustalenia wartości, ponieważ zależnie od skoku wraz z każdym kolejnym wygląda to coraz gorzej gdyż dla skoku do $yzad = T_{pp} + 15$ ledwo się wyrobił, a dla $yzad = T_{pp}$ kompletnie sobie nie poradził, gdzie DMC dało radę bez większych znaczących problemów. Z tych rozważań wychodzi wniosek, że gdy nie ma możliwości pobrania odpowiedzi skokowej obiektu to lepiej zastosować regulator PID rozmyty aniżeli konwencjonalny.

Tab. 2.9. Wskaźnik jakości regulatorów konwencjonalnych oraz rozmytego regulatora PID

Regulator	Błąd kwadratowy
PID konwencjonalny	692,13
DMC konwencjonalny	124,96
PID rozmyty	601,26

2.5. Rozmyty regulator DMC

2.5.1. Implementacja rozmytego regulatora DMC

Rozmyty regulator DMC korzysta z funkcji przynależności takich samych jak rozmyty regulator PID, które zostały przedstawione na wykresie 2.5. W części offline regulatora DMC zaimplementowano wyznaczanie 3 lokalnych odpowiedzi jednostkowych zamiast jednej, gdyż każdy lokalny regulator DMC musi posiadać swoja. Do tego parametry k_e oraz k_u zostały wyznaczone dla każdego regulatora DMC na podstawie jego własnych parametrów oraz odpowiedzi skokowej otrzymanej dla niego. W części online tak jak w przypadku rozmytego regulatora PID w każdej chwili k wyznaczano stopnie przynależności dla regulatorów lokalnych. Następnie dla każdego regulatora osobno wyznacza była zmiana wartości sterowania Δu oraz przemnażana przez stopień przynależności, aby ostatecznie tak otrzymane wartości zsumować otrzymując docelową zmianę sterowania Δu. Warto w tym przypadku wspomnieć, że każdy z regulatorów lokalnych korzysta z tego samego uchybu wartości zadanej oraz wektora przeszłych sterowań, ponieważ odnoszą się one do obiektu, a nie do konkretnego regulatora lokalnego.

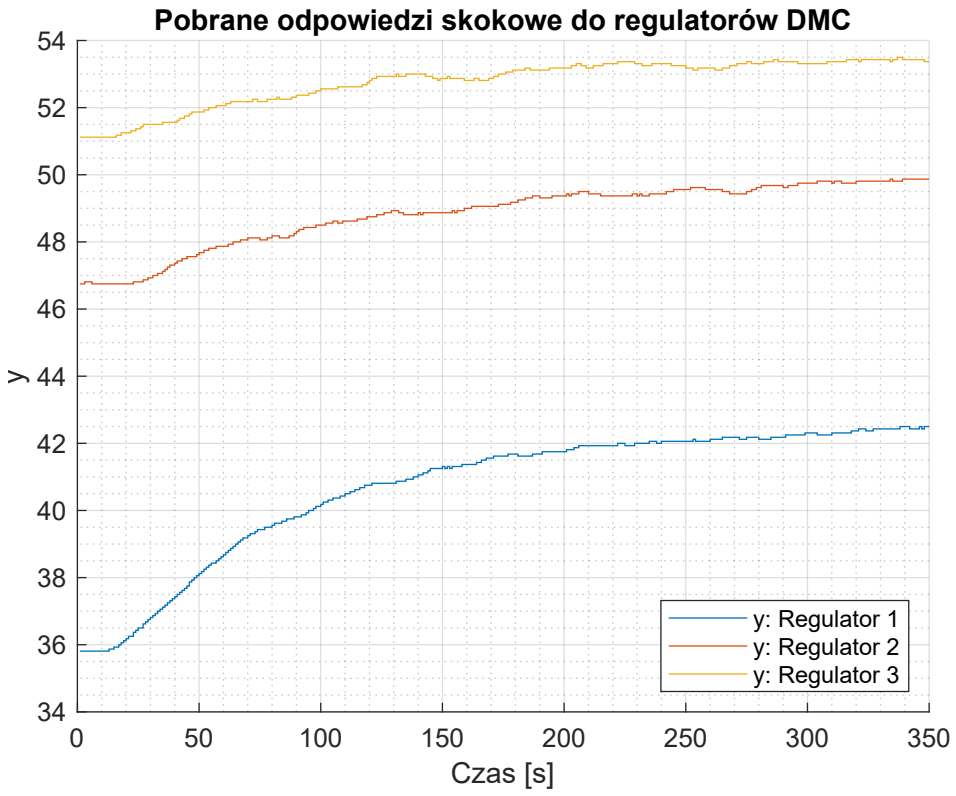
```
kk=3000; % koniec symulacji
umin = 0; umax=100;

Tpp = 36;

yzad(1:300) = Tpp; yzad(301:600) = Tpp+5;
yzad(601:900) = Tpp+15; yzad(901:kk) = Tpp;

D = 500;
N = D;
Nu = D;
lambda = [0.6,0.4,0.4];

% Odpowiedzi skokowe z modeli
```

```
K = [0.44229, 0.32123, 0.22986];
Td = [7, 7, 7];
T1 = [10.2021, 6.8717, 15.3719];
T2 = [76.437, 110.917, 77.9063];
for i=1:3
    ys(:,i) = model(T1(i),T2(i),K(i),Td(i),D);
end

% wyznaczenie ke i ku
ke(3,1)=0;
ku(3,D-1)=0;
for i=1:3
    [ke(i),ku(i,:)] = DMC_offline(ys(:,i),N,Nu,lambda(i),D);
end

% punkty rozmycia
y_rozm=[44, 48, 51.5];

% Inicjalizacja
u(1:kk)=0; y(1:kk)=0; e(1:kk)=0;
delta_u_p(1:D-1)=0; % Przeszłe przyrosty u

% Warunki początkowe
u(1)=25; y(1)=35.18;

% Główna pętla symulacyjna
for k=2:kk
    % Symulacja obiektu
    y(k) = readMeasurements(1);

    % Uchyb regulacji
    e(k)=yzad(k)-y(k);

    mi=fun_przyn_trap(y(k), y_rozm, 3);
    disp(mi);

    % Obliczenie przyrostu sygnału sterującego DMC
    delta_u = mi*(ke * e(k) - ku * delta_u_p');

    % Ograniczenia
    if u(k-1)+delta_u < umin
        delta_u = umin-u(k-1);
    elseif u(k-1)+delta_u > umax
        delta_u = umax-u(k-1);
    end

    % Aktualizacja sygnału sterującego
    u(k)=u(k-1)+delta_u;

    % Aktualizacja przeszłych przyrostów sterowania
    for n=D-1:-1:2
```

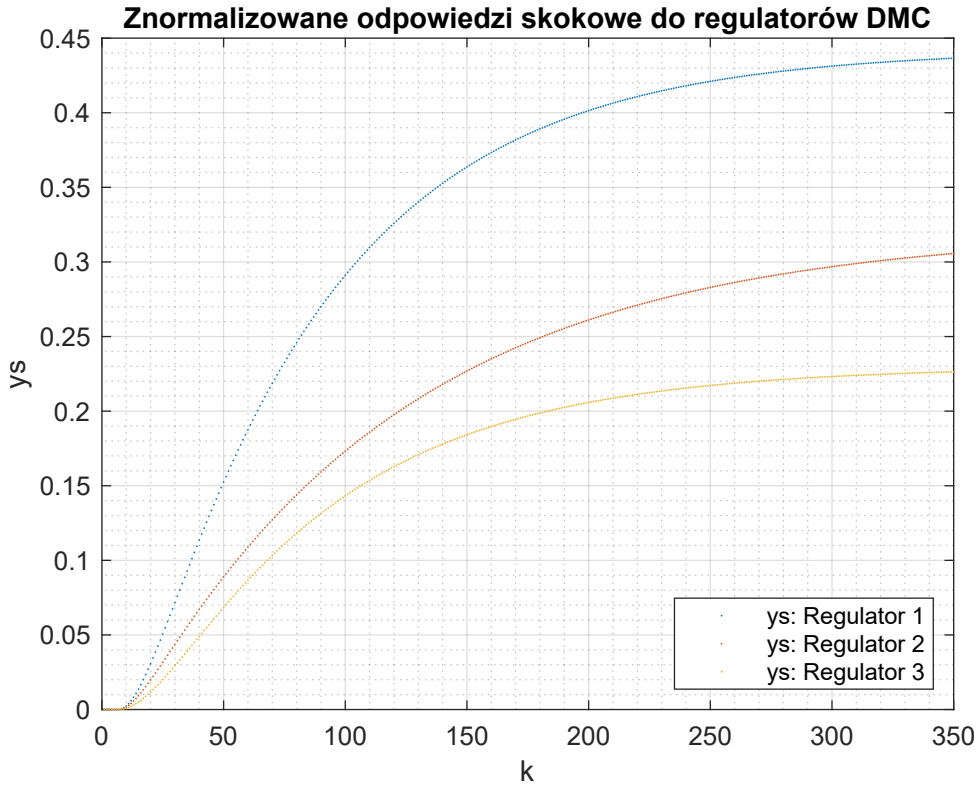
```
        delta_u_p(n) = delta_u_p(n-1);
    end
    delta_u_p(1) = delta_u;

    sendControls (1, 50) ; % W1
    sendNonlinearControls(u(k)); % G1

    wykresy_online_aktualizacja;

    waitForNewIteration () ;
end
```

2.5.2. Znormalizowane odpowiedzi skokowe do rozmytego regulatora DMC

W celu zaprojektowania regulatora DMC rozpoczęto od pobrania 3 odpowiedzi skokowych, po jednej dla każdego lokalnego regulatora DMC. Dla pierwszego pobrano jako skok z $u = U_{pp} = 25$ do $u = 40$, dla drugiego z $u = 50$ do $u = 60$, a dla trzeciego z $u = 65$ do $u = 75$. Wartości te wybrano w taki sposób, żeby jak najszerzej pokrywały zakres wartości wyjścia, gdzie lokalny regulator jest uwzględniany, jednocześnie zostawiając pole do zmian funkcji przynależności. Tak pobrane odpowiedzi zostały przedstawione na wykresie 2.13. Następnie odpowiedzi zostały aproksymowane jak na poprzednich laboratoriach i na podstawie otrzymanych modeli wyznaczano konkretne wartości y dla znormalizowanych odpowiedzi skokowych dla każdego regulatora, których to wartości przedstawiono na wykresie 2.14.



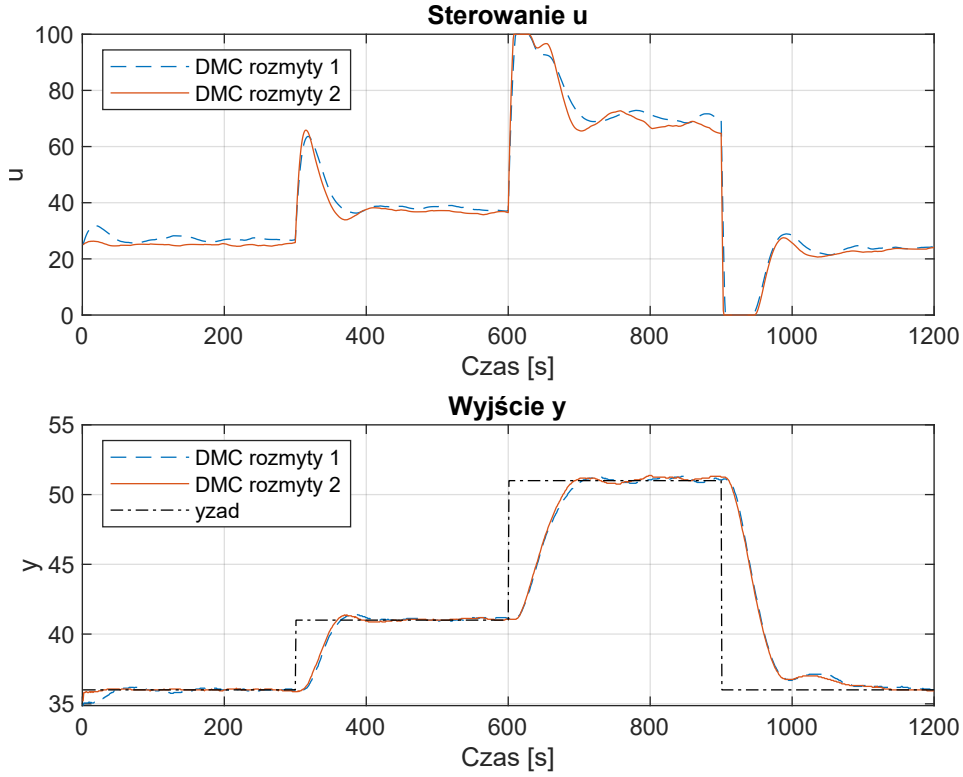
Rys. 2.13. Odpowiedzi skokowe dla lokalnych regulatorów DMC pobrane z obiektu



Rys. 2.14. Znormalizowane odpowiedzi skokowe dla lokalnych regulatorów DMC

2.5.3. Dobór parametru λ

Eksperyment mające na celu dobór parametru λ rozmytego regulatora DMC dobierano przeprowadzając symulacje dla uprzednio wspomnianej trajektorii wartości zadanej wyjścia y_{zad} czyli T_{pp}, $T_{pp}+5$, $T_{pp}+15$, T_{pp} oraz czas na ustalenie wartości sterowania wynosił $300s$. Wszystkie horyzonty ustawiono na tą samą wartość, co horyzont dynamiki, więc $D = N = N_u$. Jako pierwszą przeprowadzono symulację dla parametrów $\lambda = 1$. Na podstawie otrzymanych dla tego eksperymentu wyników otrzymanych na wykresie 2.15 zauważono, że przeregulowanie jest minimalne w przypadku regulatora 1 lokalnego, a w przypadku regulatorów 2 i 3 jest niezauważalne w związku z czym podjęto próbę przyspieszenia działania regulatora poprzez docelowo dwukrotne zmniejszenie parametru λ, ale aby uwzględnić powyższe obserwacje ostatecznie bardziej zmniejszono ten parametr dla regulatorów 2 i 3 do $\lambda = 0{,}4$, a dla regulatora trochę mniej do $\lambda = 0{,}6$. Otrzymany wynik działania dla tego regulatora przedstawiono na tym samym wykresie 2.15. Jak można zauważyć na tych przebiegach otrzymany regulator charakteryzował się minimalnie szybszym działaniem, co jest pożądane, jednakże jego zastosowanie powodowało minimalnie większe oscylacje zwłaszcza w przedziale wartości wyjścia działania regulatorów 2 i 3, które mimo wszystko były zbliżone do wartości błędu pomiaru wartości wyjścia obiektu. Dodatkowo aby lepiej porównać regulatory rozmyte DMC policzono wcześniej wspomniany błąd kwadratowy dla przyjętej trajektorii wartości zadanej opisanej w powyższych rozdziałach. Wartości tych błędów przedstawiono w tabeli 2.10, wskazują one na nieznacznie lepsze działanie regulatora 1 dla parametrów $\lambda = 1$, jednakże uznano, że analiza jakościowa wskazuje, na na tyle znacznie lepsze działanie regulatora dl zmienionych parametrów λ, że wybrano go do dalszego porównania.



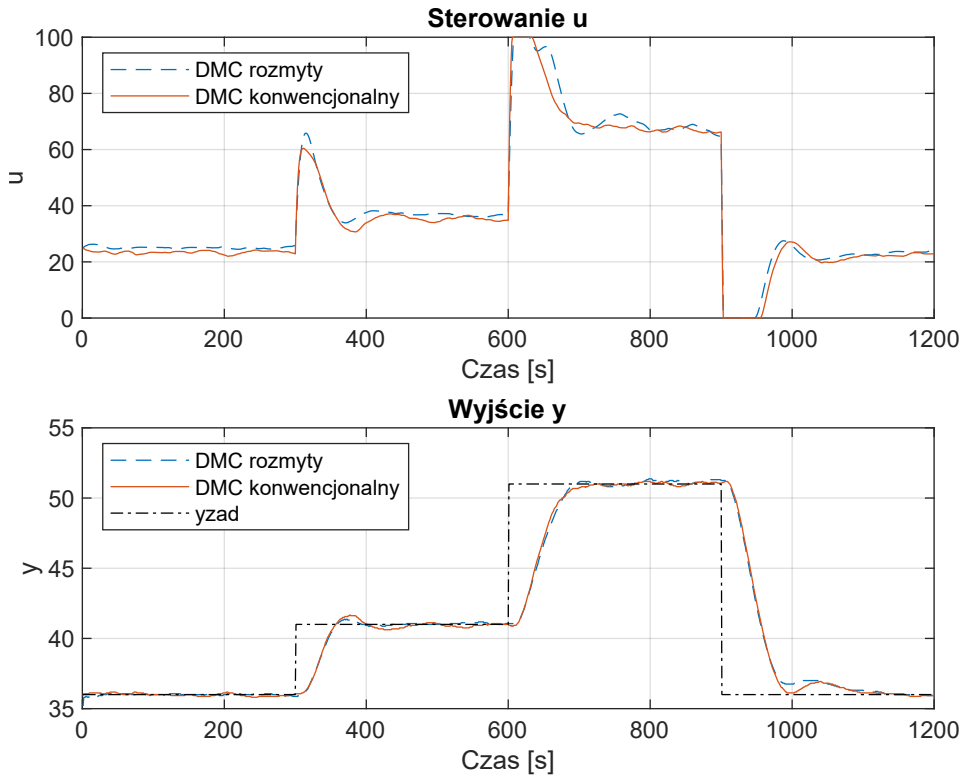
Rys. 2.15. Wykres sterowania oraz wyjścia obiektu dla rozmytego regulatora DMC

Tab. 2.10. Wskaźnik jakości regulatorów rozmytych DMC podczas doboru parametru λ

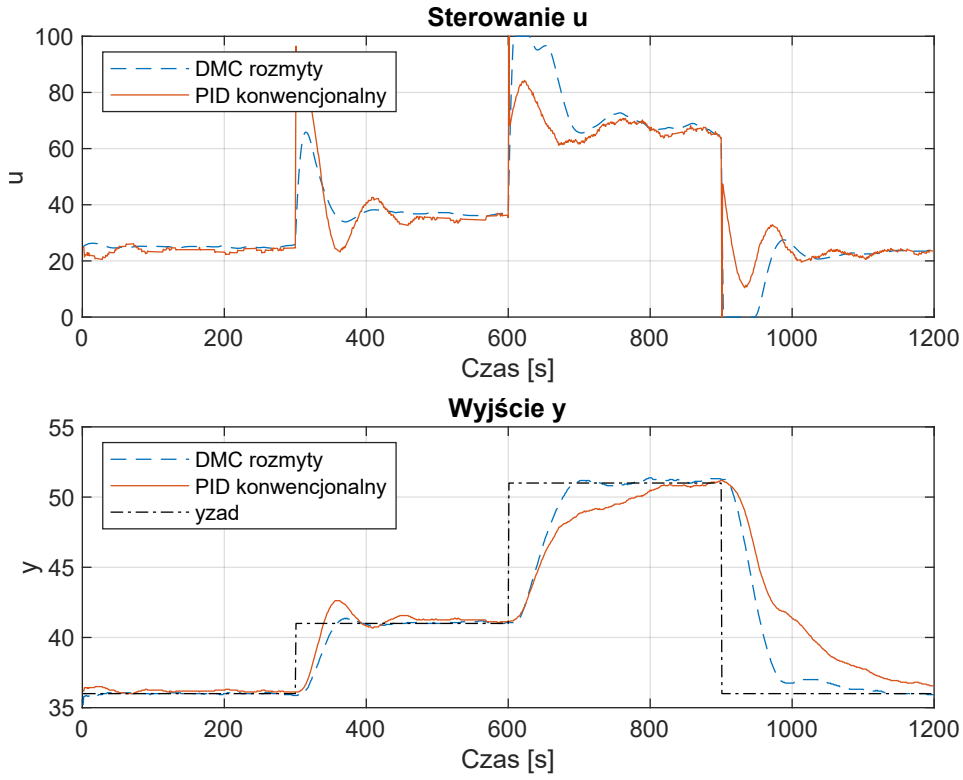
Regulator	Błąd kwadratowy
Rozmyty DMC 1	174,72
Rozmyty DMC 2	178,69

2.6. Porównanie rozmytego regulatora DMC z rozmytym PID i regulatorami konwencjonalnymi

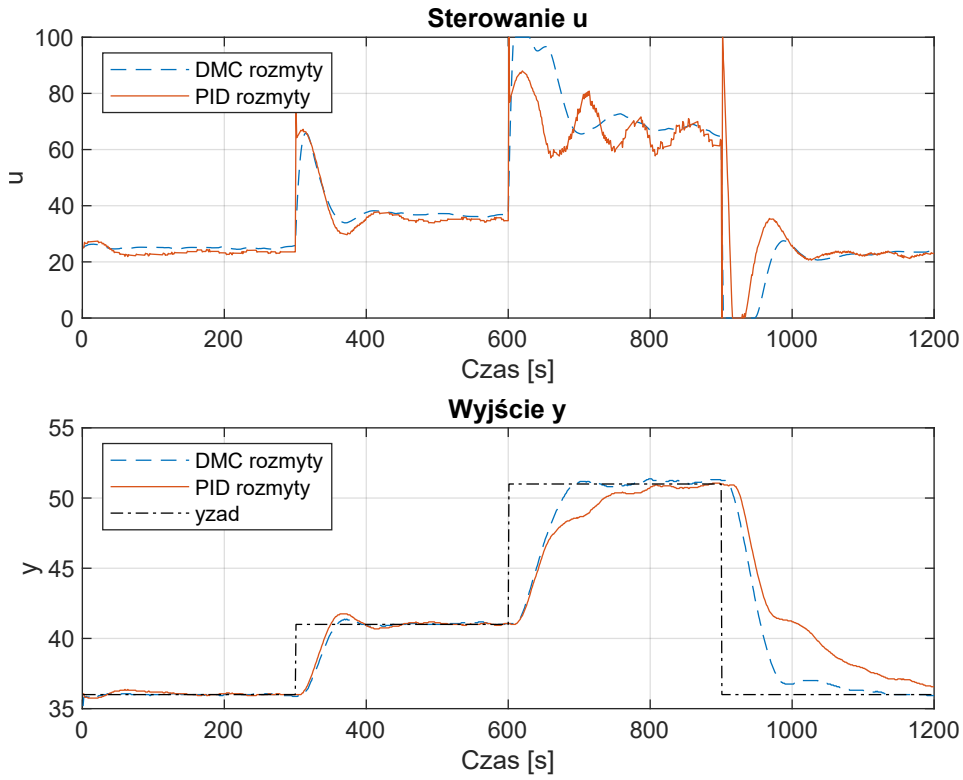
W celu porównania rozmytego regulatora DMC z przedstawionymi powyżej regulatorami, przedstawiono jego przebiegi wartości sterowania oraz wyjścia obiektu na tle wartości pozyskanych z pozostałych regulatorów oraz umieszczono policzone dla badanej trajektorii błędy kwadratowe dla wszystkich regulatorów i przedstawiono w tabeli 2.11.



Rys. 2.16. Porównanie rozmytego regulatora DMC z konwencjonalnym regulatorem DMC



Rys. 2.17. Porównanie rozmytego regulatora DMC z konwencjonalnym regulatorem PID



Rys. 2.18. Porównanie rozmytego regulatora DMC z rozmytym regulatorem PID

Tab. 2.11. Wskaźnik jakości regulatorów konwencjonalnych i rozmytych

Regulator	Błąd kwadratowy
Konwencjonalny PID	692,13
Konwencjonalny DMC	124,96
Rozmyty PID	601,26
Rozmyty DMC	178,69

Na podstawie powyższych wyników częściowo jak się spodziewano lepsze rezultaty otrzymano dla regulatorów DMC niż regulatorów PID, których działanie zależy jedynie od dostrojenia regulatora, a nie jak w przypadku regulatorów DMC częściowo od tego a częściowo od jakości pobranych odpowiedzi skokowych. W badanym przypadku dla regulatorów DMC na podstawie wykresu 2.16 uznano, że lepiej z regulacją radzi sobie wersja rozmyta DMC ponieważ osiąga ona znacznie mniejsze prze regulowania gdy do nich dochodzi, do tego ma mniejsze oscylacje oraz czas ustalenia wartości na wyjściu jest lepszy. Co ciekawe błąd obliczony i przedstawiony w tabeli 2.11 wskazuje na co innego, jednakże wynika to prawdopodobnie z tego, że regulator rozmyty w przypadku spadku wartości zadanej nie odbija się od niej jak konwencjonalny tylko schodzi schodkami w związku z czym mim podobnego jak nie lepszego czasu ustalenia się wartości wyjścia jego błąd jest większy co w tym wypadku w pewien sposób zakłóca wiarygodność wskaźnika jakości. W związku z tym wniosek z całych laboratoriów nasuwa się taki, że w przypadku procesów nieliniowych warto jest stosować regulatory rozmyte, gdy zasoby pozwalają na nastrojenie większej ilości regulatorów lokalnych i/lub pobranie większej ilości odpowiedzi skokowych.