

Wydział Elektroniki i Technik Informacyjnych
Politechnika Warszawska

Projektowanie układów sterowania
(projekt grupowy)

Sprawozdanie z ćwiczenia laboratoryjnego nr 4

Mateusz Koroś, Ksawery Pasikowski, Mateusz Morusiewicz

Warszawa, 2017

Spis treści

1. Zad. 1	2
2. Zad. 2	3
3. Zad.3	5
4. Zad. 4	7
4.1. PID	7
4.2. DMC	8
5. Zad. 5	12
6. Zad. 6	14
7. Zad. 7	16
8. Zad. 8	17

1. Zad. 1

Możliwość sterowania i pomiaru w komunikacji ze stanowiskiem została sprawdzona poprzez funkcję `readMeasurements`, `sendControls` oraz `sendNonLinearControls`. Sygnały sterujące, które były obsługiwane to moc na grzałce $G1$ (wysłana za pomocą funkcji `sendNonLinearControls`) oraz moc wiatraka $W1$ (wysłana za pomocą `sendControls`), natomiast mierzona była temperatura $T1$ w otoczeniu grzałki $G1$. W punkcie pracy, tzn. dla $(W1, G1) = (50, 29)$ pomiar temperatury $T1$ wyniósł 34°C .

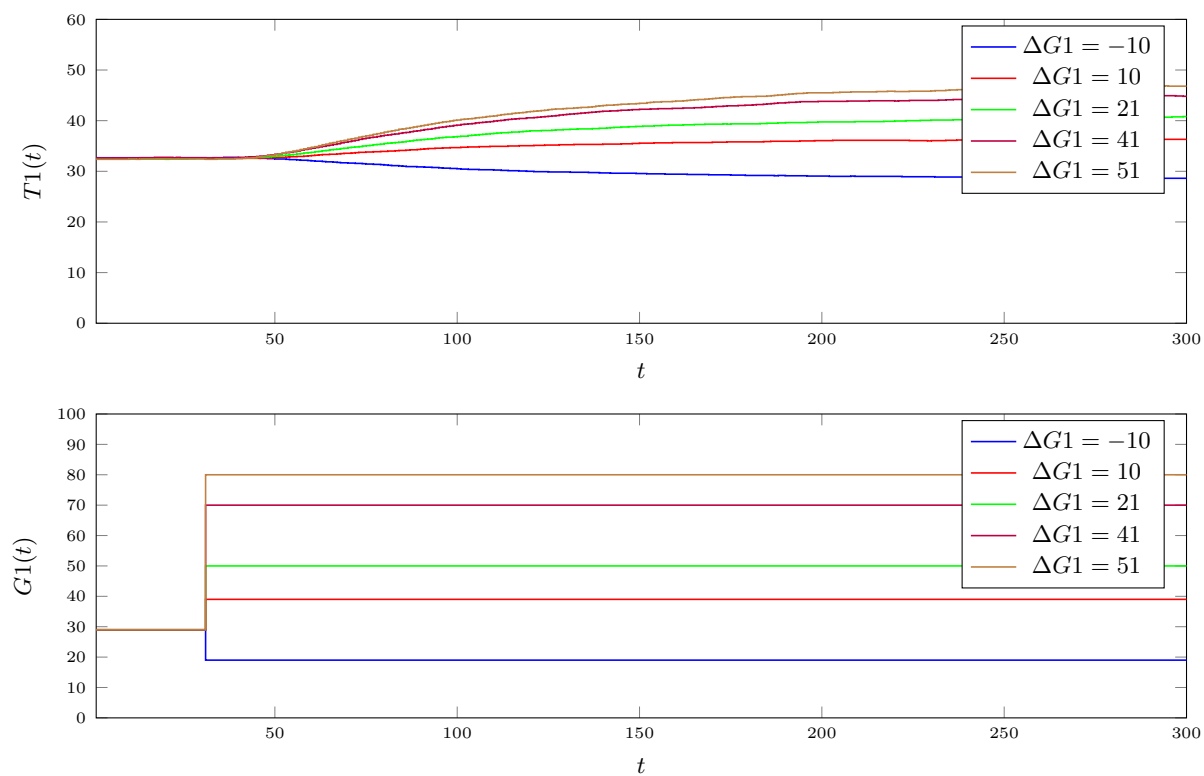
2. Zad. 2

Skoki sterowania wynosiły odpowiednio:

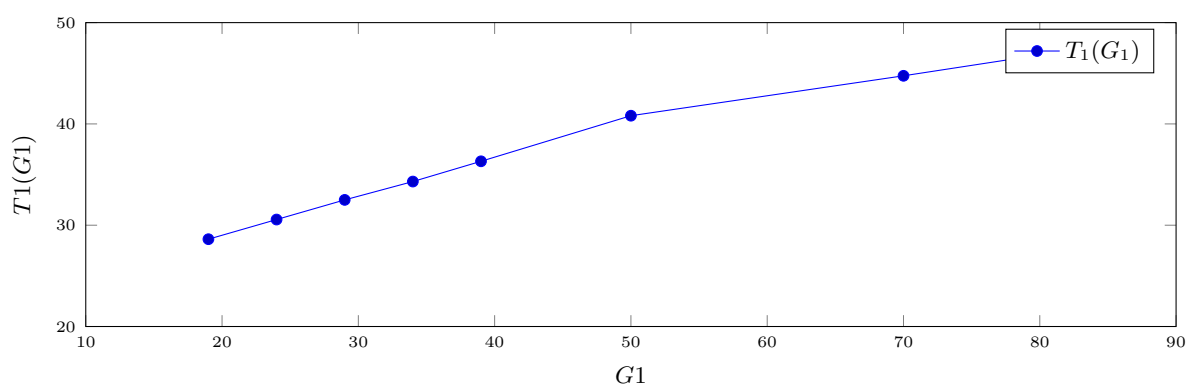
- $\Delta u = 10$
- $\Delta u = -10$
- $\Delta u = 21$
- $\Delta u = 41$
- $\Delta u = 51$

Odpowiedzi skokowe widać na wykresie 2.1

Właściwości statycznych obiektu nie można określić jako liniowych, gdyż zmiana sterowania nie powoduje liniowej zmiany sygnału wyjściowego. Nie da się więc obliczyć wzmocnienia statycznego obiektu. Charakterystyka jest jednak (w przybliżeniu) lokalnie liniowa, co widać na wykresie 2.2



Rys. 2.1. Odpowiedzi skokowe



Rys. 2.2. Charakterystyka statyczna obiektu

3. Zad.3

Najlepiej nadającą się odpowiedzią skokową była ta, o wartości $\Delta u = 21$. Została ona przekształcona do tej, używanej w algorytmie DMC w następujący sposób:

```
Ysk = (s - 32) / 21;
```

Następnie została wykonana aproksymacja odpowiedzi skokowej, do której został użyty człon inercyjny drugiego rzędu z opóźnieniem, opisany transmitancją:

$$G(z) = \frac{b_1 z^{-1} + b_2 z^{-2}}{1 + a_1 z^{-1} + a_2 z^{-2}} z^{-T_d}$$

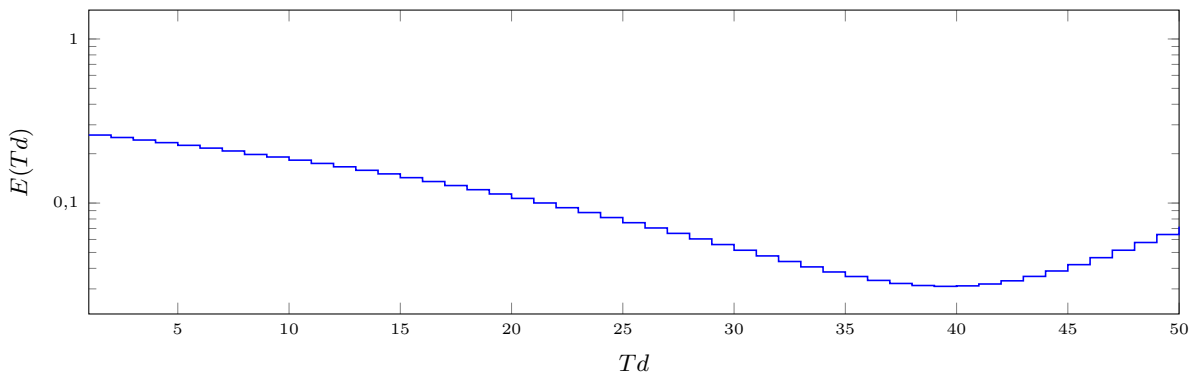
Gdzie:

$$\begin{aligned} a_1 &= -\alpha_1 - \alpha_2 \\ a_2 &= \alpha_1 \alpha_2 \\ \alpha_1 &= e^{-\frac{1}{T_1}} \\ \alpha_2 &= e^{-\frac{1}{T_2}} \\ b_1 &= \frac{K}{T_1 - T_2} [T_1(1 - \alpha_1) - T_2(1 - \alpha_2)] \\ b_2 &= \frac{K}{T_1 - T_2} [\alpha_1 T_2(1 - \alpha_2) - \alpha_2 T_1(1 - \alpha_1)] \end{aligned} \quad (3.1)$$

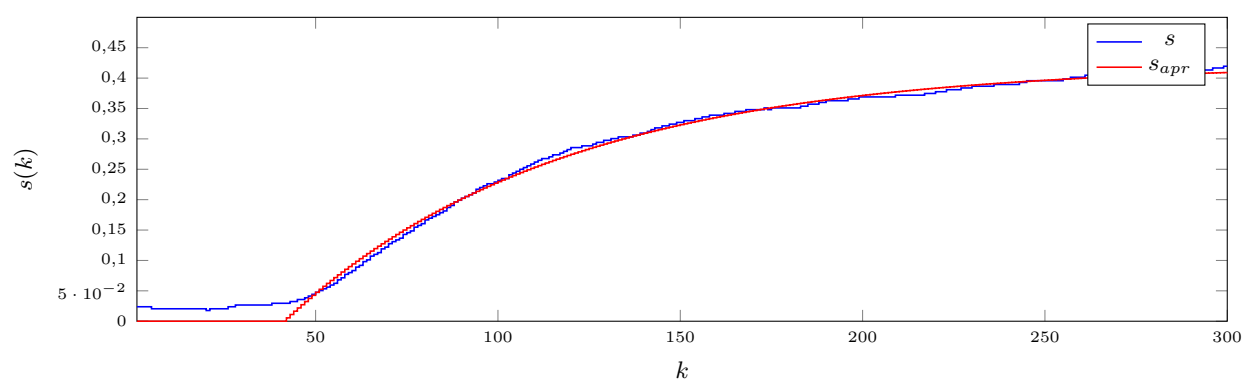
Po przekształceniu powyższego równania otrzymujemy równanie różnicowe postaci:

$$y(k) = b_1 u(k - T_D - 1) + b_2 u(k - T_d - 2) - a_1 y(k - 1) - a_2 y(k - 2)$$

W celu doboru parametrów modelu została użyta funkcja wewnętrzna środowiska MATLAB: `fmincon()`. Parametry modelu zostały dobrane w taki sposób, aby błąd średniokwadratowy między odpowiedzią aproksymowaną, a tą rzeczywistą był jak najmniejszy. Z wykresu 3.1 widać, że w najlepszym przypadku wynosił on 0,0311. Oryginalną i aproksymowaną odpowiedź skokową widać na rysunku 3.2.



Rys. 3.1. Błąd aproksymacji odpowiedzi skokowej



Rys. 3.2. Oryginalna i aproksymowana odpowiedź skokowa

4. Zad. 4

4.1. PID

```
addpath('F:\SerialCommunication');
initSerialControl COM14

Upp = 29;
Ypp = 32.5;
Kk = 800;
U_min = 0;
U_max = 100;

% nastawy regulatora PID
% Kp = 3 ;
% Ti = 10 ;
% Td = 3.2 ;
Kp = 6;%5.94 ;
Ti = 65;%5.64 ;
Td = 1.25;%3.16 ;
Tp = 1;

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

% warunki poczatkowe
u(1:31) = Upp ;
U(1:31) = Upp ;
y(1:31) = Ypp ;
y2(1:31) = Ypp ;
e(1:31) = 0 ;
delta_u = 0;
index = 1;
yzads = [60, 30];
yzad = yzads(index);    %skok wartosci zadanej
yzad2 = yzad - Ypp;
yzadVec(1:800) = yzad;
sendControls (1,W1);

figure;
% glowna petla symulacji
for k = 32 : 800
if mod(k,400) == 0
index = index + 1;
```



```

if index > length(yzads)
index = length(yzads);
end
yzad = yzads(index);
yzad2 = yzad - Ypp;
end
yzadVec(k) = yzad;

y(k) = readMeasurements(1);

y2(k) = y(k) - Ypp;
e(k) = yzad2 - y2(k) ;

u(k) = r2 * e(k-2) + r1 * e(k-1) + r0 * e(k) + u(k-1) ;

delta_u = u(k) - u(k-1);

%if delta_u > dU_max
%    delta_u = dU_max;
%elseif delta_u < -dU_max
%    delta_u = -dU_max;
%end

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

if u(k) > U_max - Upp
u(k) = U_max - Upp;
elseif u(k) < U_min - Upp
u(k) = U_min - Upp;
end

U(k) = u(k) + Upp;
sendNonlinearControls(U(k));

stairs(y);
pause(0.01);

waitForNewIteration();
end

E = (yzadVec - y)*(yzadVec - y)';

```

4.2. DMC

```

function [ y, u, E ] = DMC( D_, N_, Nu_, lambda_, Kk_)
Upp = 29;
Ypp = 34;

Kk = Kk_;

```

```

D = D_;
N = N_;
Nu = Nu_;
lambda = lambda_;

U_max = 100;
U_min = 0;

yzad = 37;    %skok wartosci zadanej
yzadVec(1:30) = Ypp;
yzadVec(31:Kk) = yzad;

s = load('Sapr_zad3');
s = s.Sapr;
%s(length(s) : 400) = s(length(s));

%sygnal sterujacy
u = Upp + zeros(1,Kk);

%wyjscie ukladu
y = zeros(1,Kk) + Ypp ;

du = (zeros(1,Kk))' ;

M = zeros(N, Nu) ;
for i = 1:N
for j = 1:Nu
if (i-j+1 > 0)
M(i,j) = s(i-j+1) ;
else
M(i,j) = 0 ;
end
end
end

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

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

%liczenie ke
ke = 0;
for i = 1:N

```

```
ke = ke + K(1, i);
end

kju = K(1,:)*Mp;

addpath ('F:\SerialCommunication'); % add a path
initSerialControl COM14 % initialise com port
sendControls ([1,5],[50,Upp]);

figure;

for k = 1:30
y(k) = readMeasurements(1);

stairs(y);
pause(0.01);

waitForNewIteration();
end

for k = 31:Kk

y(k) = readMeasurements(1);

sum = 0; %suma potrzebna do obliczenia składowej swobodnej
for j = 1:D-1
if(k-j > 0)
sum = sum + kju(j)*du(k-j);
%w innym przypadku du = 0 wiec sum sie nie zmienia
end
end

du(k) = ke * (yzadVec(k)-y(k)) - sum;

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

if u(k) > U_max - Upp
u(k) = U_max - Upp;
elseif u(k) < U_min - Upp
u(k) = U_min - Upp;
end

sendNonlinearControls(u(k)+Upp);

stairs(y);
pause(0.01);

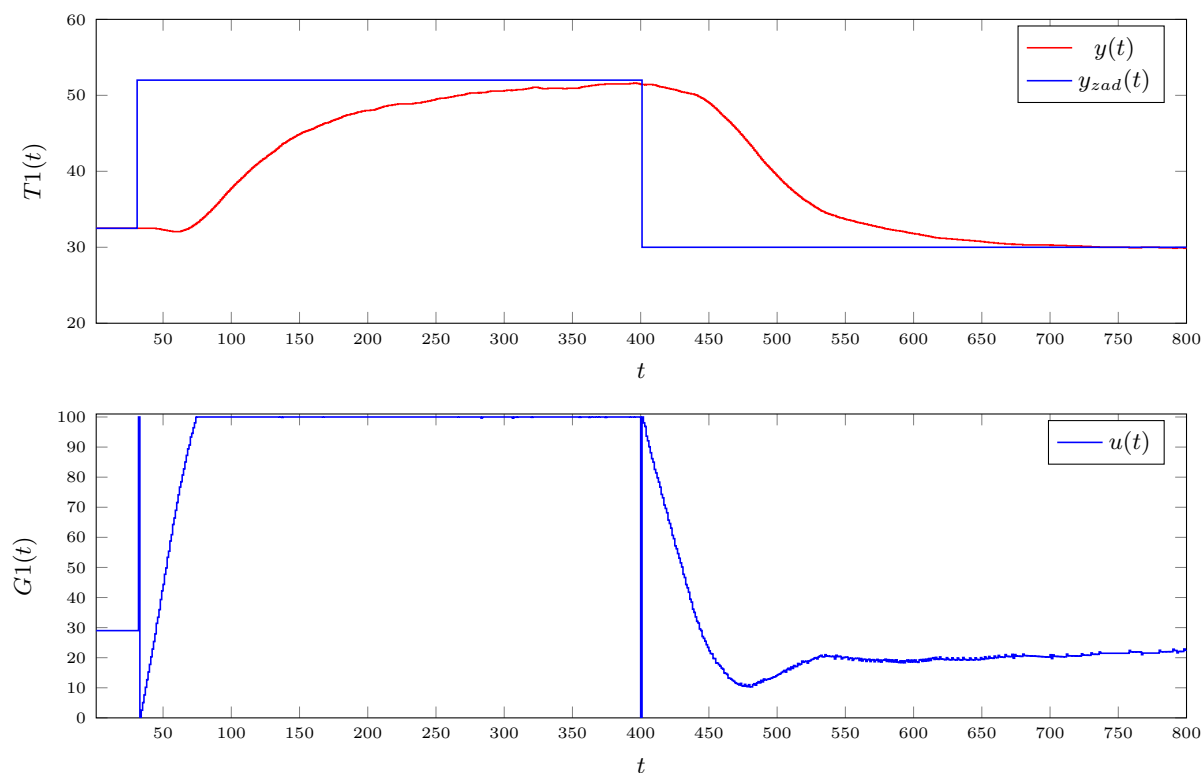
waitForNewIteration();

end
```

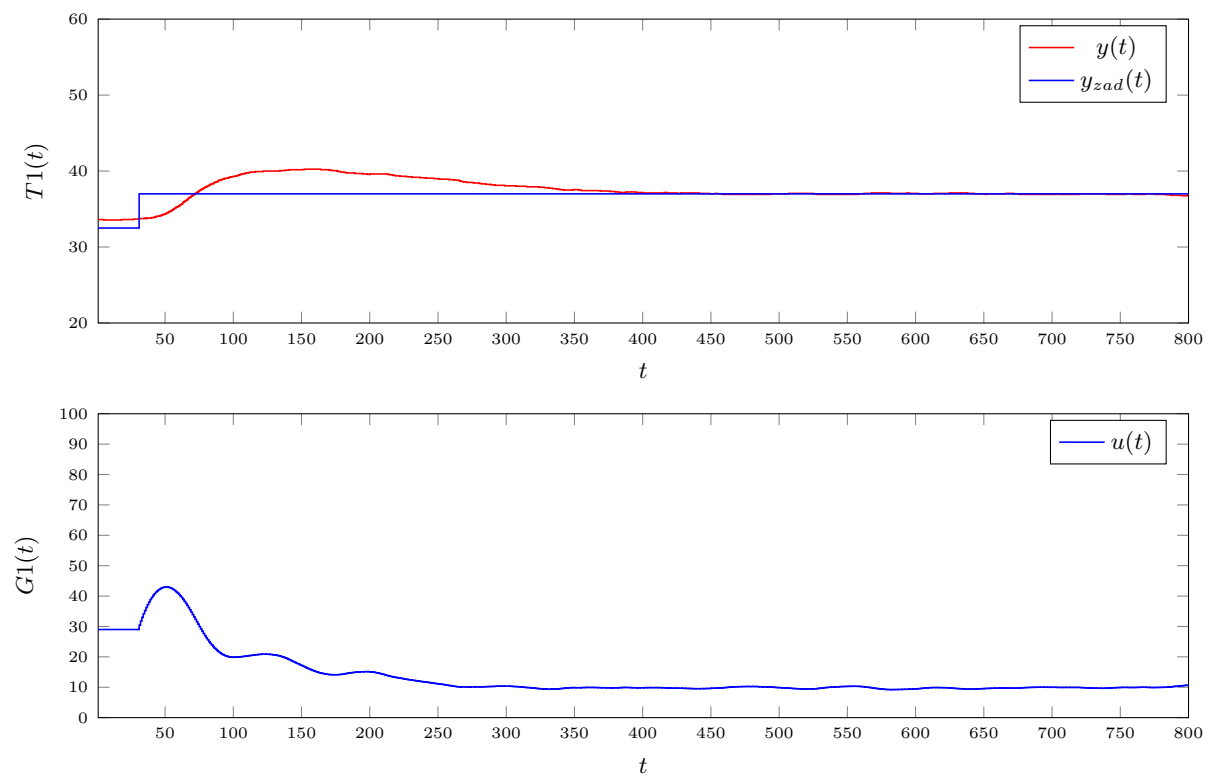
```
% wskaźnik jakości regulacji  
E = (yzadVec - y)*(yzadVec - y)';
```

5. Zad. 5

Regulator PID został przetestowany poprzez zmianę wartości zadanej z punktu pracy do 50°C a następnie do 30°C . Parametry regulatora wynosiły $K = 6$, $T_i = 65$, $T_d = 1,25$ a wyniki eksperymentu z nim widać na rysunku 5.1. Regulator DMC o parametrze $\lambda = 5$ został przetestowany dla skoku wartości zadanej do 37°C , co widać na rysunku 5.2. Oba regulatory co prawda osiągnęły wartości zadane, nie było również przeregulowania, jednak nieliniowy charakter obiektu utrudnił im zadanie. Rozwiązaniem tego problemu może być zastąpienie regulatorów tradycyjnych rozmytymi.



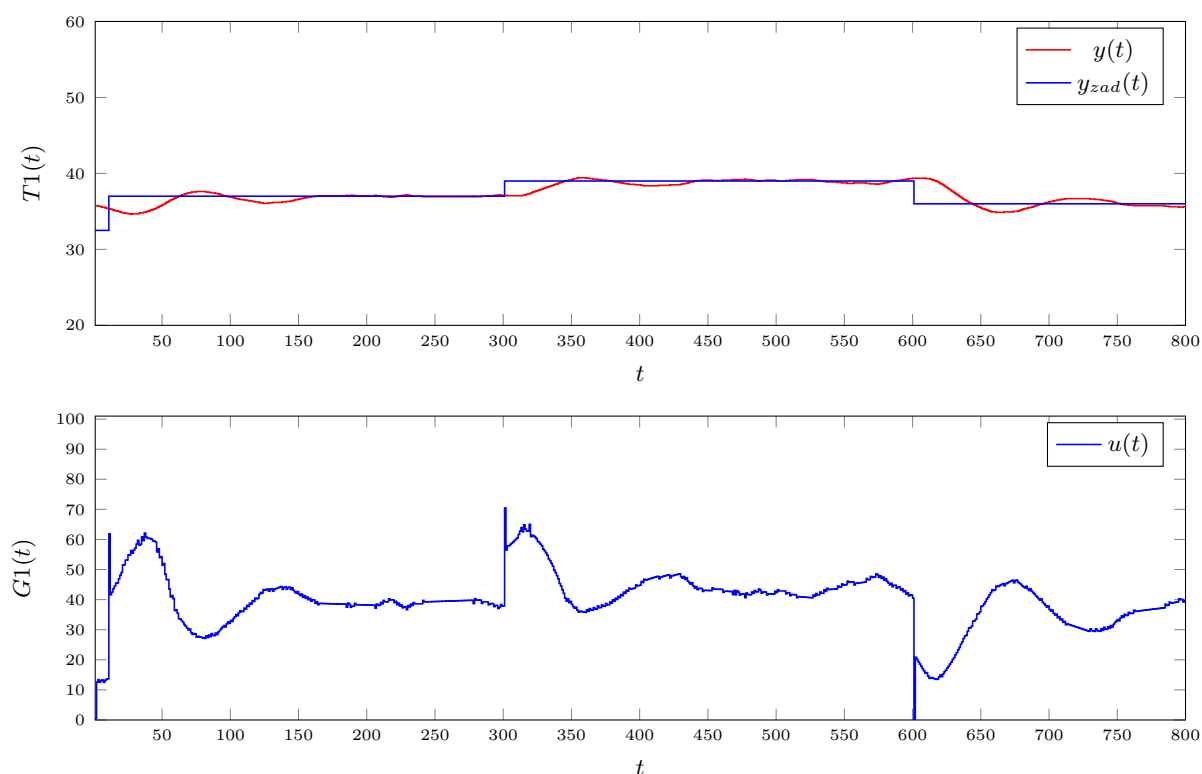
Rys. 5.1. Tradycyjny regulator PID



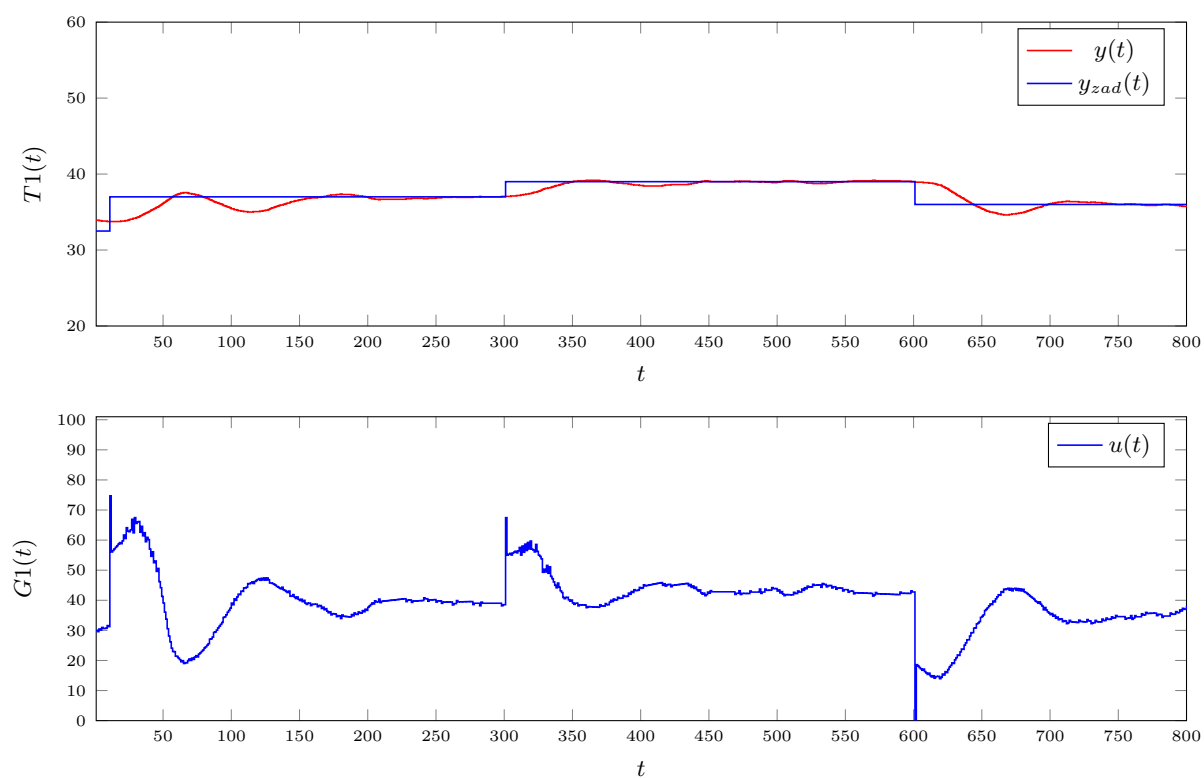
Rys. 5.2. Tradycyjny regulator DMC

6. Zad. 6

Rozmyty regulator PID został przetestowany dla większej ilości zmian wartości zadanej niż zwykły (szybsza regulacja pozwoliła na więcej skoków), składał się on z dwóch regulatorów, gdzie pierwszy miał parametry: $K = 8$, $T_i = 50$, $T_d = 0,8$, natomiast drugi: $K = 25$, $T_i = 40$, $T_d = 1,1$. Rozpoczął on w punkcie pracy a wartości zadane wynosiły $37^\circ C$, $39^\circ C$ i $36^\circ C$. Wyniki eksperymentu widać na wykresie 6.1. Przetestowany został też regulator o parametrach: $K_1 = 9$, $T_{i1} = 40$, $T_{d1} = 0,8$, $K_2 = 21$, $T_{i2} = 36$, $T_{d2} = 1$. Jak widać na wykresie 6.2 nie poradził on sobie lepiej niż poprzedni, gdyż trochę wolniej dochodził do wartości zadanej i miał nieznacznie większe przeregulowanie, stąd pierwszy regulator można uznać za lepszy. Oba regulatory rozmyte szybciej jednak osiągały wartość zadaną niż regulator tradycyjny, co świadczy o tym, że wprowadzenie regulatora rozmytego zamiast zwykłego regulatora PID jest dobrym pomysłem w przypadku sterowania obiektami nieliniowymi.



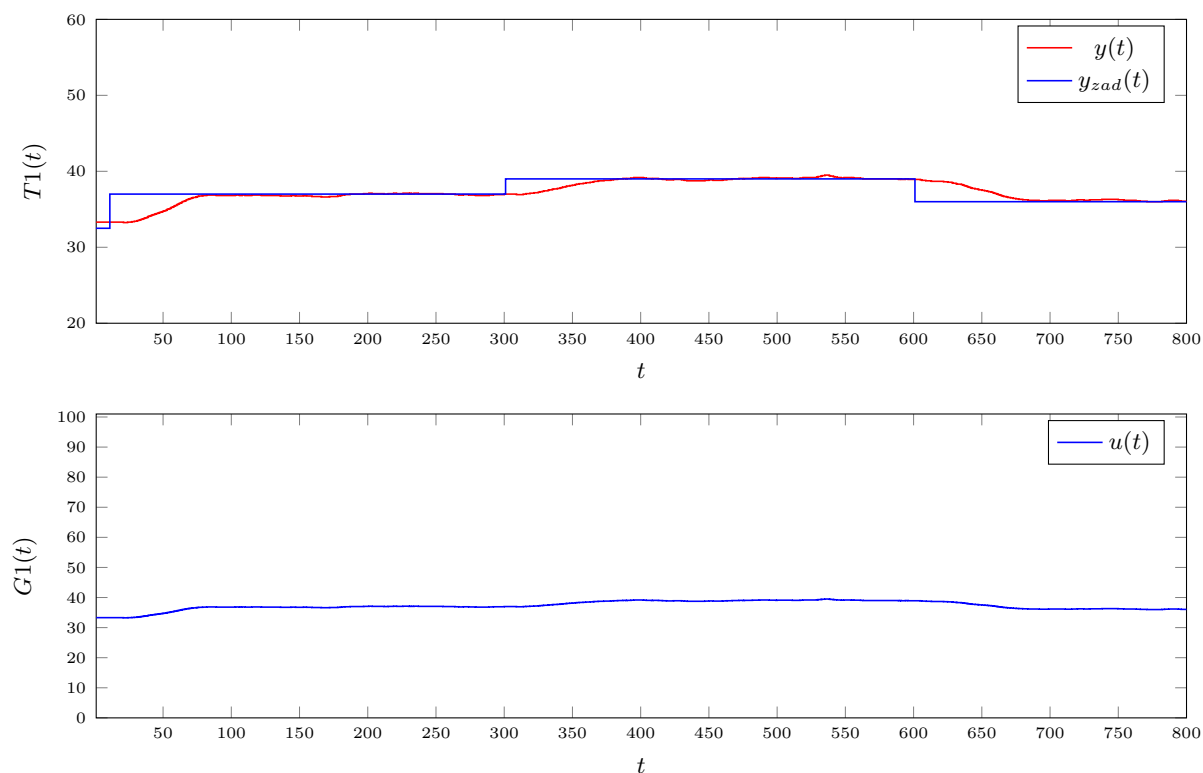
Rys. 6.1. Rozmyty regulator PID o parametrach: $K_1 = 8$, $T_{i1} = 50$, $T_{d1} = 0,8$, $K_2 = 25$, $T_{i2} = 40$, $T_{d2} = 1,1$



Rys. 6.2. Rozmyty regulator PID o parametrach: $K_1 = 9$, $T_{i1} = 40$, $T_{d1} = 0,8$, $K_2 = 21$, $T_{i2} = 36$, $T_{d2} = 1$

7. Zad. 7

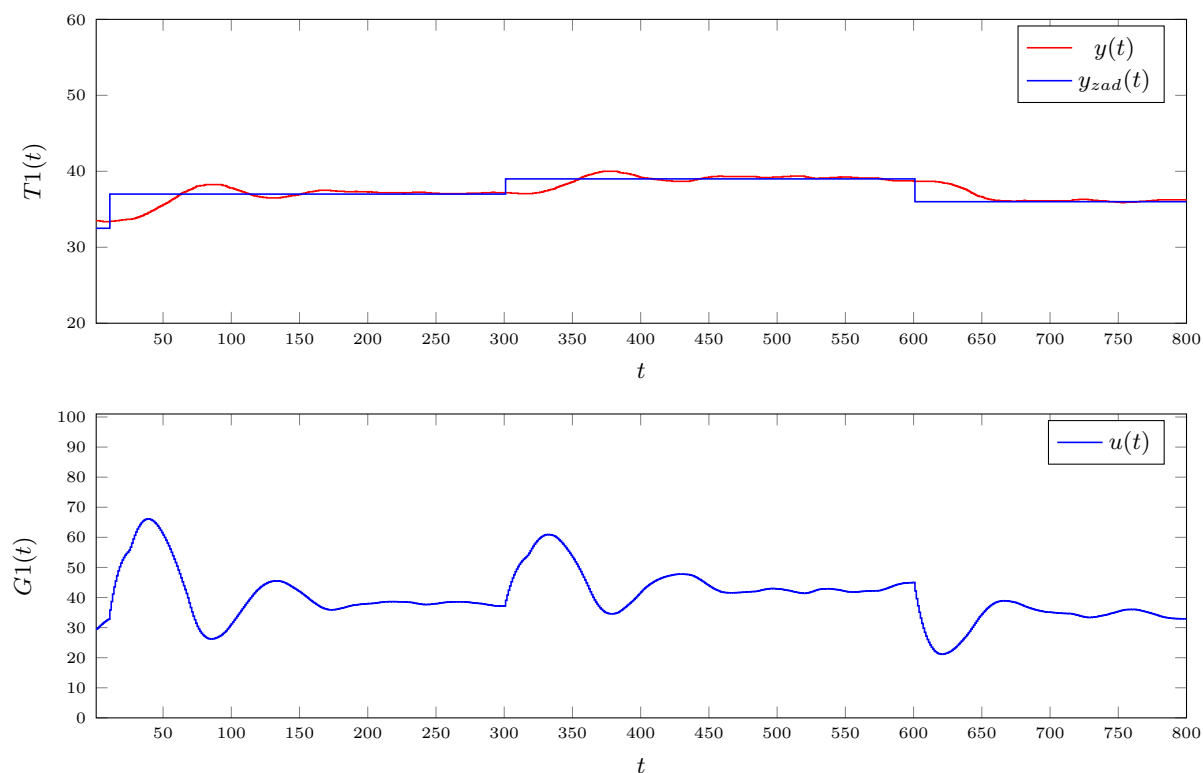
Rozmyty regulator DMC również składał się z dwóch regulatorów. Parametr λ pierwszego wynosił 5 a drugiego 10. Został on poddany takim samym wymuszeniom jak rozmyty regulator PID, tj. wartości zadane wynosiły $37^{\circ}C$, $39^{\circ}C$ i $36^{\circ}C$. Wyniki eksperymentu widać na wykresie 7.1. Jakość regulacji tego regulatora jest najlepsza jak dotąd i niezaprzeczalnie pokazuje w tym przypadku wyższość regulatora rozmytego nad tradycyjnym. Wartość zadana osiągnana jest szybko przy prawie zerowym przeregulowaniu i stabilnym sterowaniu.



Rys. 7.1. Rozmyty regulator DMC o $\lambda_1 = 5, \lambda_2 = 10$

8. Zad. 8

Sprawdzony został również rozmyty regulator DMC o parametrach $\lambda_1 = 1$, $\lambda_2 = 2$, którego działanie widać na wykresie 8.1. Poradził on sobie gorzej niż ten o $\lambda_1 = 5$ i $\lambda_2 = 10$. Wartość zadana osiągnana była dużo później oraz pojawiło się przeregulowanie. Porównanie obu rozmytych regulatorów DMC widać na wykresie 8.2. Mimo nie najlepszej jakości regulacji wciąż był lepszy niż regulator DMC w wersji tradycyjnej. Wynika z tego, że warto użyć regulatora rozmytego zamiast zwykłego DMC, gdy sterowany obiekt ma charakterystykę nieliniową.



Rys. 8.1. Rozmyty regulator DMC o $\lambda_1 = 1$, $\lambda_2 = 2$

