

**Wydział Elektroniki i Technik Informacyjnych
Politechnika Warszawska**

**Projektowanie układów sterowania
(projekt grupowy)**

Sprawozdanie z ćwiczenia laboratoryjnego nr 3

Mateusz Koroś, Ksawery Pasikowski, Mateusz Morusiewicz

Warszawa, 2017

Spis treści

1. Zad. 1	2
2. Zad. 2	3
3. Zad. 3	4
4. Zad. 4	7
4.1. PID	7
4.2. DMC	9
5. Zad. 5	12

1. Zad. 1

Możliwość sterowania i pomiaru w komunikacji ze stanowiskiem została sprawdzona poprzez funkcję `readMeasurements` oraz `sendControls`. Sygnały sterujące, które były obsługiwane to moc na grzałkach $G1$ i $G2$ oraz moc wiatraków $W1$ i $W2$, natomiast mierzona była temperatura $T1$ w otoczeniu grzałki $G1$ oraz temperatura $T2$ w punkcie pomiędzy grzałkami $G1$ i $G2$. W punkcie pracy, tzn. dla $(W1, W2, G1, G2) = (50, 50, 29, 34)$ pomiar temperatury $T1$ wyniósł $36,5^{\circ}C$, natomiast pomiar temperatury $T2$ wyniósł $38,5^{\circ}C$.

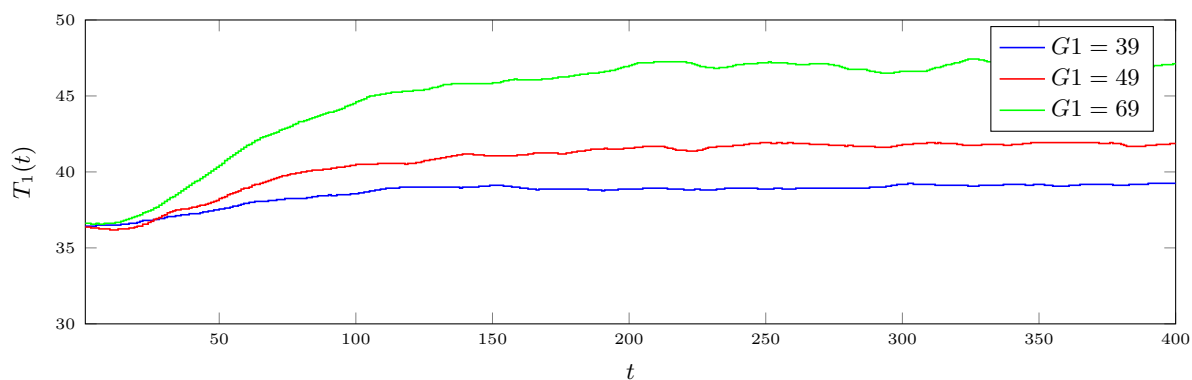
2. Zad. 2

Skoki sterowania wynosiły odpowiednio:

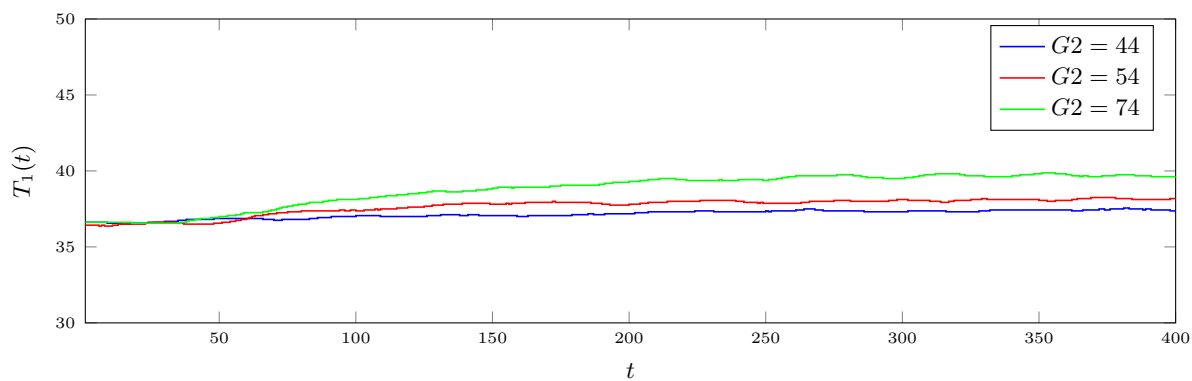
- $\Delta u = 10$
- $\Delta u = 20$
- $\Delta u = 40$

Odpowiedzi skokowe widać na wykresach ??, ??, ?? i ??

Właściwości statyczne obiektu można określić jako liniowe, gdyż zmiana sterowania powoduje liniową zmianę sygnału wyjściowego. Charakterystyki statyczne zostały przedstawione na wykresie 2.2 Wzmocnienie statyczne procesu dla każdego z torów zostało obliczone ze wzoru:



Rys. 2.1. Skrośne odpowiedzi skokowe czujnika T_1 dla różnych skoków sterowania grzałki G_1



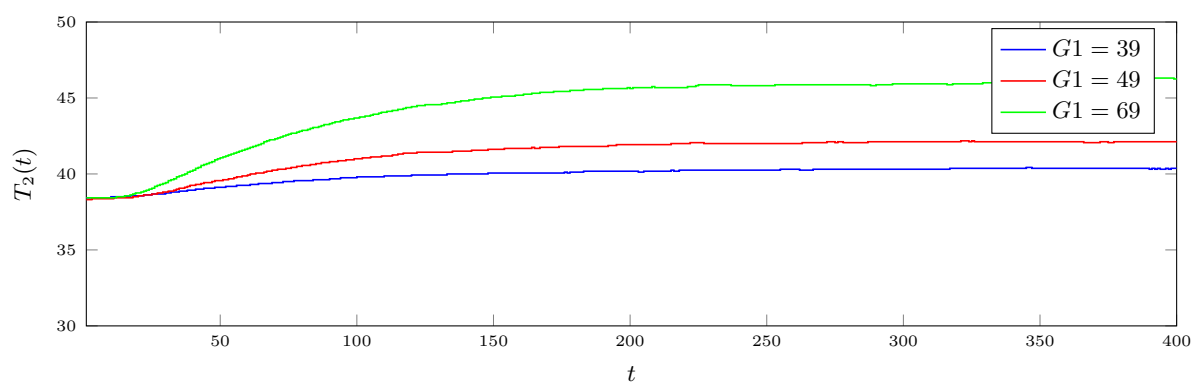
Rys. 2.2. Skrośne odpowiedzi skokowe czujnika T_1 dla różnych skoków sterowania grzałki G_2

$$K_{11} = \frac{\Delta y_1}{\Delta u_1} \approx 0,26$$

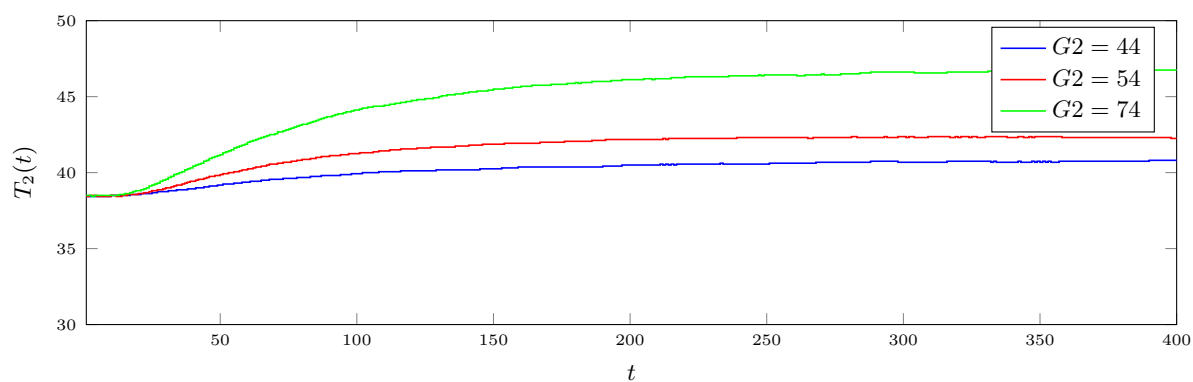
$$K_{12} = \frac{\Delta y_1}{\Delta u_2} \approx 0,08$$

$$K_{21} = \frac{\Delta y_2}{\Delta u_1} \approx 0,20$$

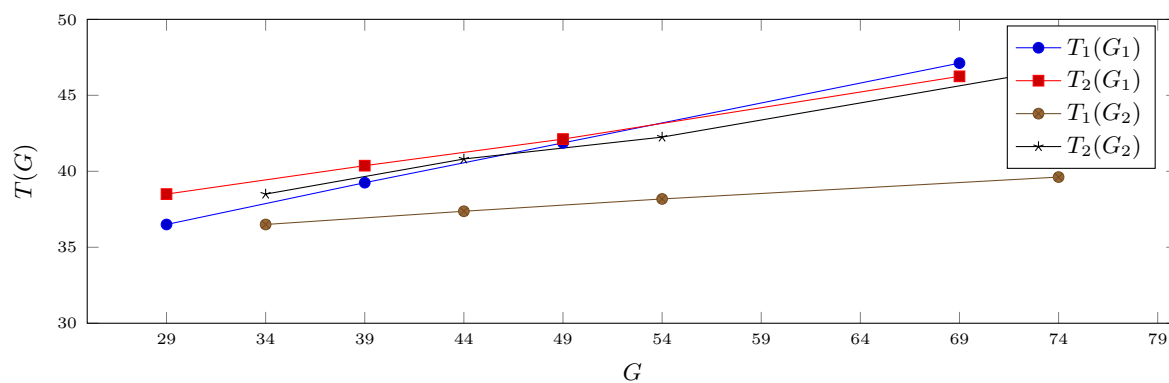
$$K_{22} = \frac{\Delta y_2}{\Delta u_2} \approx 0,21$$



Rys. 2.3. Skrośne odpowiedzi skokowe czujnika T_2 dla różnych skoków sterowania grzałki G_1



Rys. 2.4. Skrośne odpowiedzi skokowe czujnika T_2 dla różnych skoków sterowania grzałki G_2



Rys. 2.5. Charakterystyki statyczne

3. Zad. 3

Najlepiej nadającymi się odpowiedziami skokowymi sygnału sterującego były te dla skoku sterowania $\Delta u = 40$. Odpowiedzi te zostały przekształcone do postaci wykorzystywanej w algorytmie DMC w następujący sposób:

```
Ysk = (s.Y1 - 36.5) / 40;
S = Ysk((Td+3):end);
```

Analogicznie została przekształcona odpowiedź skokowa drugiego czujnika:

```
Ysk = (s.Y2 - 38.5) / 40;
S = Ysk((Td+3):end);
```

Następnie została wykonana aproksymacja odpowiedzi skokowych, 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:

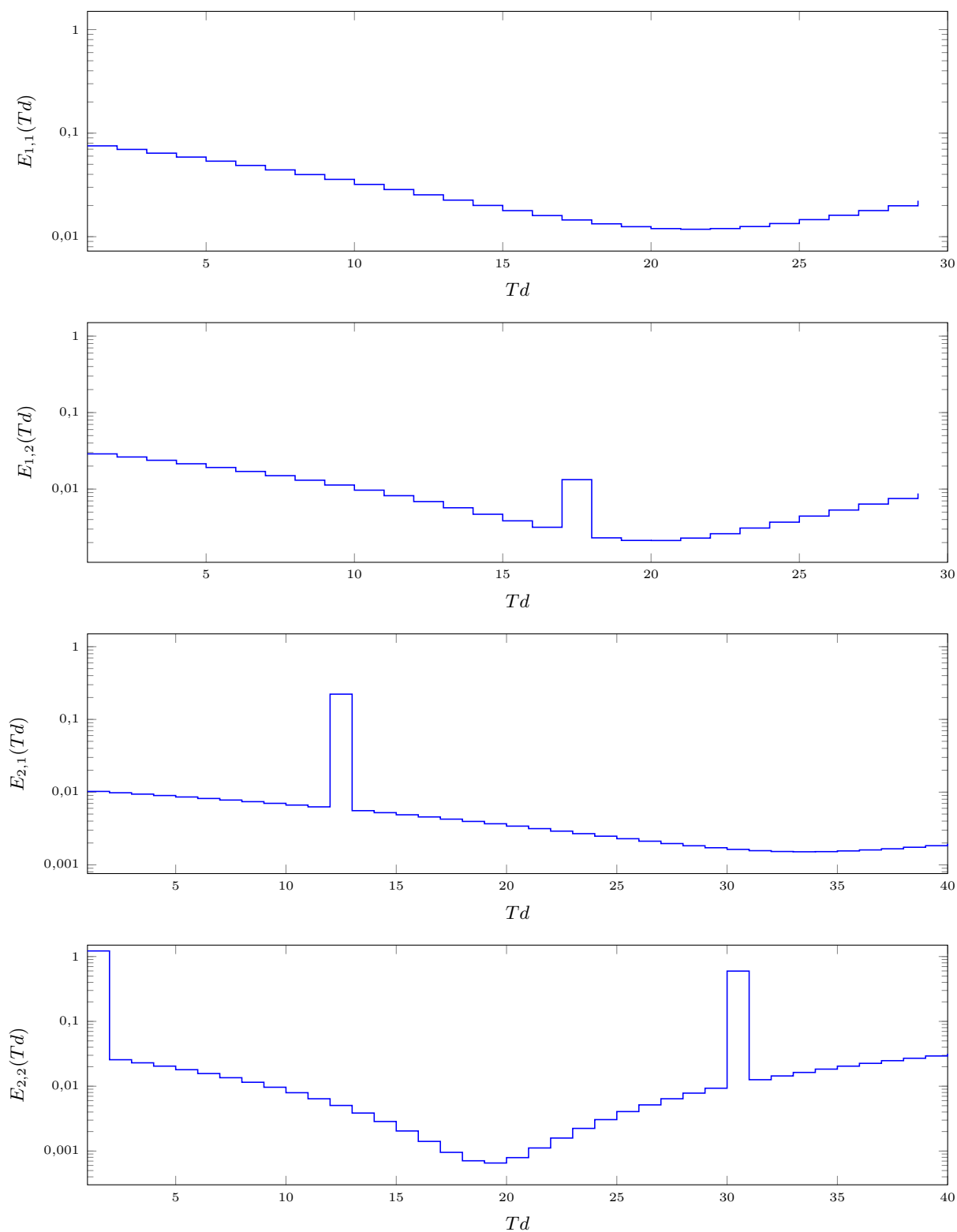
$$y(k) = b_1 u(k - T_D - 1) + b_2 u(k - T_d - 2) - a_1 y(k - 1) - a_2 y(k - 2)$$

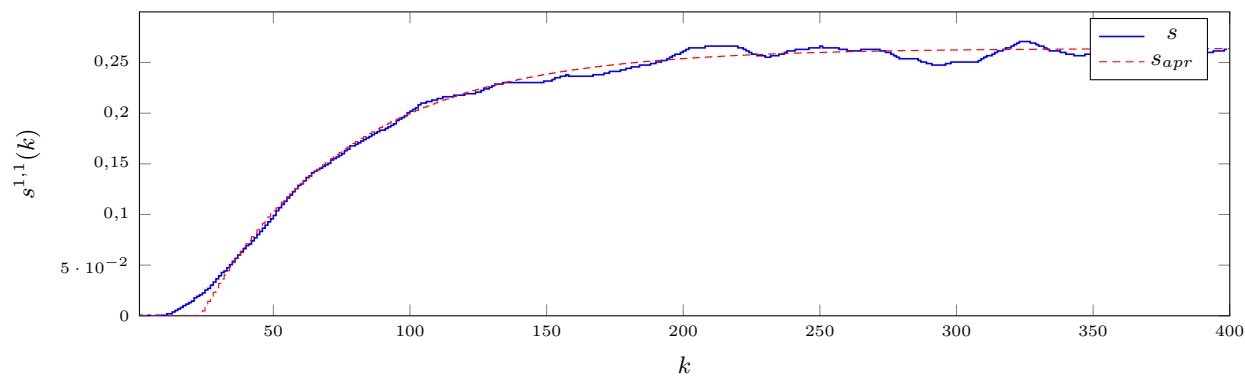
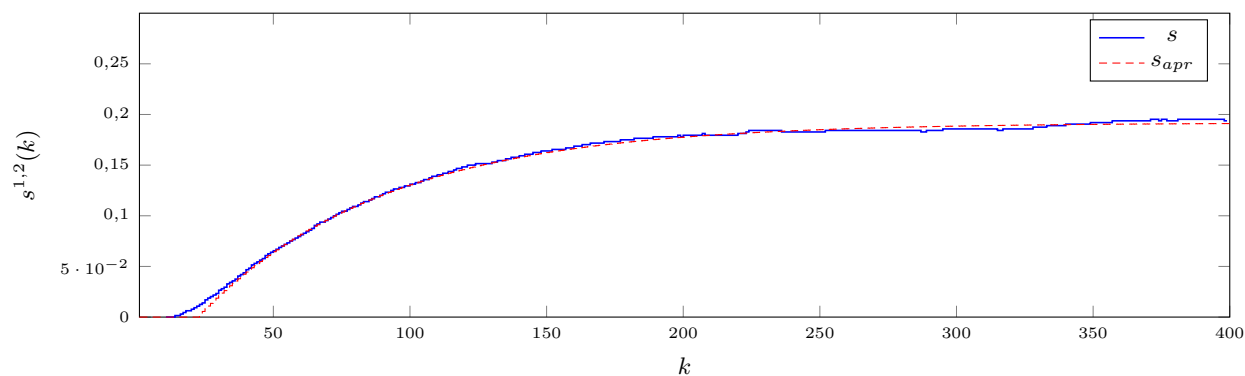
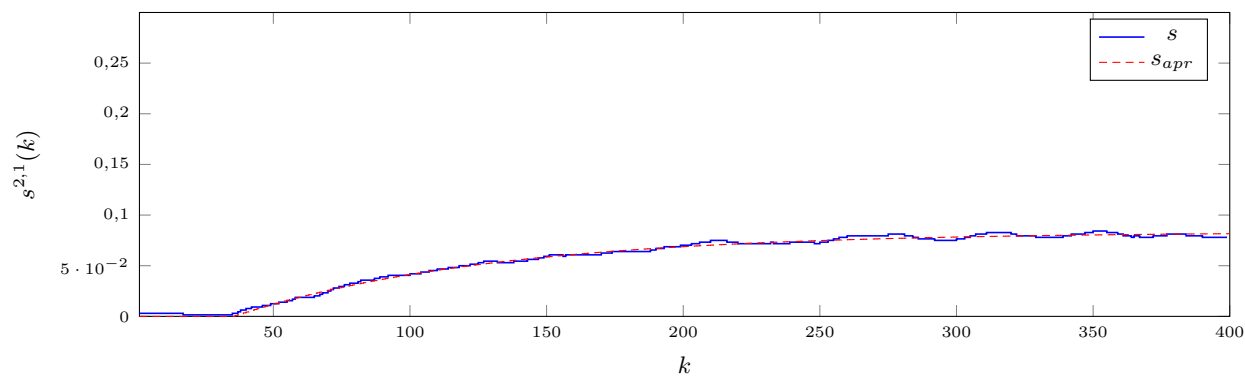
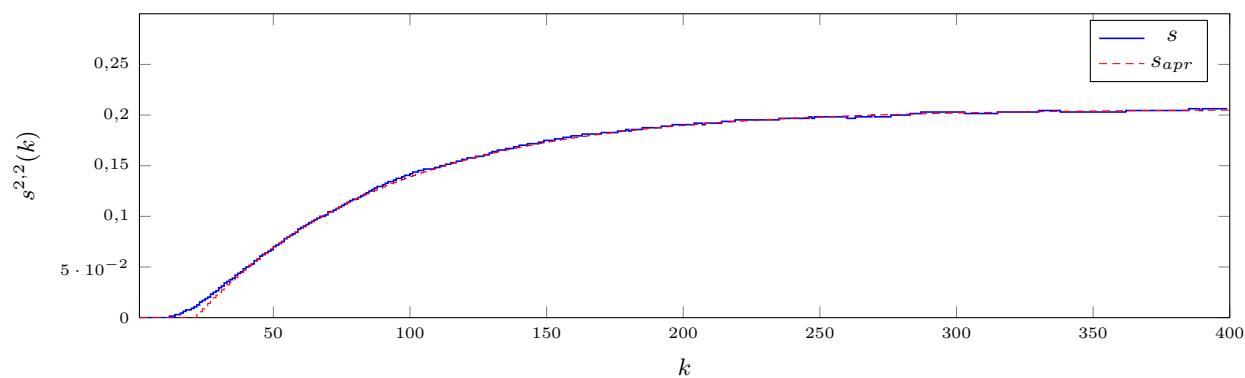
W celu doboru parametrów modelu została użyta funkcja wewnętrzna środowiska MATLAB: `fmincon()`. Parametry modelu, zwrócone przez ową funkcję dla różnych odpowiedzi skokowych:

- $G1, T1 \rightarrow T_1 = 0,1077, \quad T_2 = 54,7428, \quad T_3 = 0,2446$
- $G1, T2 \rightarrow T_1 = 0,1906, \quad T_2 = 377,9330, \quad T_3 = 0,8156$
- $G2, T1 \rightarrow T_1 = 0,1733, \quad T_2 = 232,1743, \quad T_3 = 0,1750$
- $G2, T2 \rightarrow T_1 = 0,1778, \quad T_2 = 149,0973, \quad T_3 = 0,3670$

Parametry te zostały dobrane w taki sposób, aby błąd średniokwadratowy między odpowiedzią aproksymowaną, a tą rzeczywistą był jak najmniejszy. Wartość błędu w zależności od parametru T_d dla odpowiedzi skokowych sterowania (w kolejności zgodnej z powyższym wypunktowaniem) widać na rysunku 3.1

Odpowiedzi skokowe obu czujników dla skoków sygnałów sterujących wraz z ich aproksymacjami widać na wykresach 3.2, 3.3, 3.4, 3.5. Jak widać na wykresach, funkcja aproksymująca jest bardzo dobrym przybliżeniem oryginalnego przebiegu. Sumaryczny błąd jest niewielki.

Rys. 3.1. Błędy średniokwadratowe odpowiedzi skokowych sterowania w zależności od T_d

Rys. 3.2. Odpowiedź skokowa czujnika $T1$ dla skoku sterowania na grzałce $G1$ Rys. 3.3. Odpowiedź skokowa czujnika $T2$ dla skoku sterowania na grzałce $G1$ Rys. 3.4. Odpowiedź skokowa czujnika $T1$ dla skoku sterowania na grzałce $G2$ Rys. 3.5. Odpowiedź skokowa czujnika $T2$ dla skoku sterowania na grzałce $G2$

4. Zad. 4

4.1. PID

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

yzads = [40, 37];
Upp1 = 29;
Ypp1 = 36.5;
Upp2 = 34;
Ypp2 = 38.5;
Kk = 500;
U_min = 0;
U_max = 100;

% nastawy regulatora PID
% Kp = 3 ;
% Ti = 10 ;
% Td = 3.2 ;
Kp1 = 3;%5.94 ;
Ti1 = 25;%5.64 ;
Td1 = 0.7;%3.16 ;
Tp = 1;

Kp2 = 5;%5.94 ;
Ti2= 50;%5.64 ;
Td2= 0;%3.16 ;
Tp2= 1;

r2_1 = (Kp1 * Td1) / Tp ;
r1_1 = Kp1 * ( (Tp/(2*Ti1)) - 2*(Td1/Tp) - 1 ) ;
r0_1 = Kp1 * ( 1 + Tp/(2*Ti1) + Td1/Tp ) ;

r2_2 = (Kp2 * Td2) / Tp ;
r1_2 = Kp2 * ( (Tp/(2*Ti2)) - 2*(Td2/Tp) - 1 ) ;
r0_2 = Kp2 * ( 1 + Tp/(2*Ti2) + Td2/Tp ) ;

% warunki poczatkowe
u1(1:31) = Upp1 ;
U1(1:31) = Upp1 ;
y1(1:31) = Ypp1 ;
y2_1(1:31) = Ypp1 ;
e1(1:31) = 0 ;
```

```

delta_u1 = 0;
u2(1:31) = Upp2 ;
U2(1:31) = Upp2 ;
y2(1:31) = Ypp2 ;
y2_2(1:31) = Ypp2 ;
e2(1:31) = 0 ;
delta_u2 = 0;

index = 1;

yzad = yzads(1);    %skok wartosci zadanej
yzad2 = yzads(2);
yzadVec(1:30) = Ypp1;
yzadVec2(1:30) = Ypp2;
yzadVec(31:Kk) = yzad;
yzadVec2(31:Kk) = yzad2;

figure;
% glowna petla symulacji
for k = 3 : Kk

tmp = readMeasurements([1,2]);
y1(k) = tmp(1);
y2(k) = tmp(2);

e1(k) = yzadVec(k) - y1(k) ;
e1(k);
e2(k) = yzadVec2(k) - y2(k) ;

% regulator dla T1
u1(k) = r2_1 * e1(k-2) + r1_1 * e1(k-1) + r0_1 * e1(k) + u1(k-1);

if u1(k) > U_max - Upp1
u1(k) = U_max - Upp1;
elseif u1(k) < U_min - Upp1
u1(k) = U_min - Upp1;
end

U1(k) = u1(k) + Upp1;

% regulator dla T2
u2(k) = r2_2 * e2(k-2) + r1_2 * e2(k-1) + r0_2 * e2(k) + u2(k-1) ;

if u2(k) > U_max - Upp2
u2(k) = U_max - Upp2;
elseif u2(k) < U_min - Upp2
u2(k) = U_min - Upp2;
end

U2(k) = u2(k) + Upp2;

```

```

sendControls([ 1,2,5,6], [ 50, 50, U1(k), U2(k)]);

plot(y1); hold on;
plot(y2); hold off;
pause(0.01);

waitForNewIteration();
end

E1 = (yzadVec - y1)*(yzadVec - y1)';
E2 = (yzadVec2 - y2)*(yzadVec2 - y2)';
E = E1+E2;

```

4.2. DMC

```

addpath ('F:\SerialCommunication'); % add a path
initSerialControl COM14 % initialise com port
N = 200;
Nu = 50;
D=200;
kk=250;
lambda = 2;
Upp1 = 29;
Upp2 = 34;
U_max = 100;
U_min = 0;

Sapru1y1 = load('Sapr_T1_G1');
Sapru2y1 = load('Sapr_T1_G2');
Sapru1y2 = load('Sapr_T2_G1');
Sapru2y2 = load('Sapr_T2_G2');
s11=Sapru1y1.Sapr;
s12=Sapru2y1.Sapr;
s21=Sapru1y2.Sapr;
s22=Sapru2y2.Sapr;

ny=2;
nu=2;
y=zeros(ny, kk);
yzad=zeros(ny, kk);
yzad(1, 1: kk)=40;
yzad(2, 1: kk)=37;
u=zeros(nu, kk);
du=zeros(nu, kk);
dUP=cell(D-1, 1);

```

```

dUP(1:D-1)={zeros(2,1)};
M=cell(N,Nu);

for i=1:N
for j=1:Nu
if (i>=j)
M(i,j)=[s11(i-j+1) s12(i-j+1); s21(i-j+1) s22(i-j+1)];
else
M(i,j)={zeros(nu,ny)};
end
end
end
MP=cell(N,D-1);
for i=1:N
for j=1:D-1
if i+j<=D
MP(i,j)=[s11(i+j)-s11(j) s12(i+j)-...
s12(j); s21(i+j)-s21(j) s22(i+j)-s22(j)];
else
MP(i,j)=[s11(D)-s11(j) s12(D)-s12(j); ...
s21(D)-s21(j) s22(D)-s22(j)];
end
end
end
K=(cell2mat(M)'*cell2mat(M)+...
diag(ones(1,Nu*ny)*lambda))^-1*cell2mat(M)';
ku=K(1:ny,:)*cell2mat(MP);
ke1=sum(K(1,1:2:(N*ny)));
ke2=sum(K(1,2:2:(N*ny)));
ke3=sum(K(2,1:2:(N*ny)));
ke4=sum(K(2,2:2:(N*ny)));
figure;
for k=2:kk

y(1,k) = readMeasurements(1);
y(2,k) = readMeasurements(2);
du(:,k)=[ke1 ke2;ke3 ke4]*(yzad(:,k)-y(:,k))-ku*cell2mat(dUP);
for i=D-1:-1:2
dUP(i)=dUP(i-1);
end
dUP(1)={du(:,k)};
u(:,k)=u(:,k-1)+du(:,k);
if u(1,k) > U_max - Upp1
u(1,k) = U_max - Upp1;
elseif u(1,k) < U_min - Upp1
u(1,k) = U_min - Upp1;
end
if u(2,k) > U_max - Upp2
u(2,k) = U_max - Upp2;
elseif u(2,k) < U_min - Upp2
u(2,k) = U_min - Upp2;

```

```
end

sendControls ([1,2,5,6],[50, 50, u(1,k) + Upp1, u(2,k) + Upp2]);

plot(y(1,:)); hold on;
plot(y(2,:)); hold off;
pause(0.01);

waitForNewIteration();
end

E1=0;
E2=0;
for k=1:kk
E1=E1+((yzad(1,k)-y(1,k))^2);
E2=E2+((yzad(2,k)-y(2,k))^2);
end
E=E1+E2;
```

5. Zad. 5

Regulatory zostały przetestowane poprzez zmianę wartości zadanej z punktu pracy do pewnej wartości. Dla obu regulatorów była to zmiana najpierw na jednym czujniku (skok $T1$ do 39), następnie na drugim (skok $T2$ do 41) a ostateczną próbą był skok jednej wartości do większej niż punkt pracy, natomiast drugiej do mniejszej ($T1_{zad} = 40, T2_{zad} = 37$). W regulatorze PID na pierwszych dwóch wykresach widać uchyb ustalony, który został wyeliminowany w ostatecznej próbie poprzez zwiększenie wpływu członu całkującego, co spowodowało znaczną poprawę. Regulator DMC poradził sobie bardzo dobrze we wszystkich przypadkach. Ostateczne parametry obu regulatorów:

PID: $K_p = 3 \quad T_i = 25 \quad T_d = 0,7$

DMC: $D = 200 \quad N = 200 \quad N_u = 50$.

Wskaźniki jakości regulacji dla poszczególnych skoków:

— PID

$$T1_{zad} = 39 \rightarrow E = 2018,9$$

$$T2_{zad} = 41 \rightarrow E = 1609,2$$

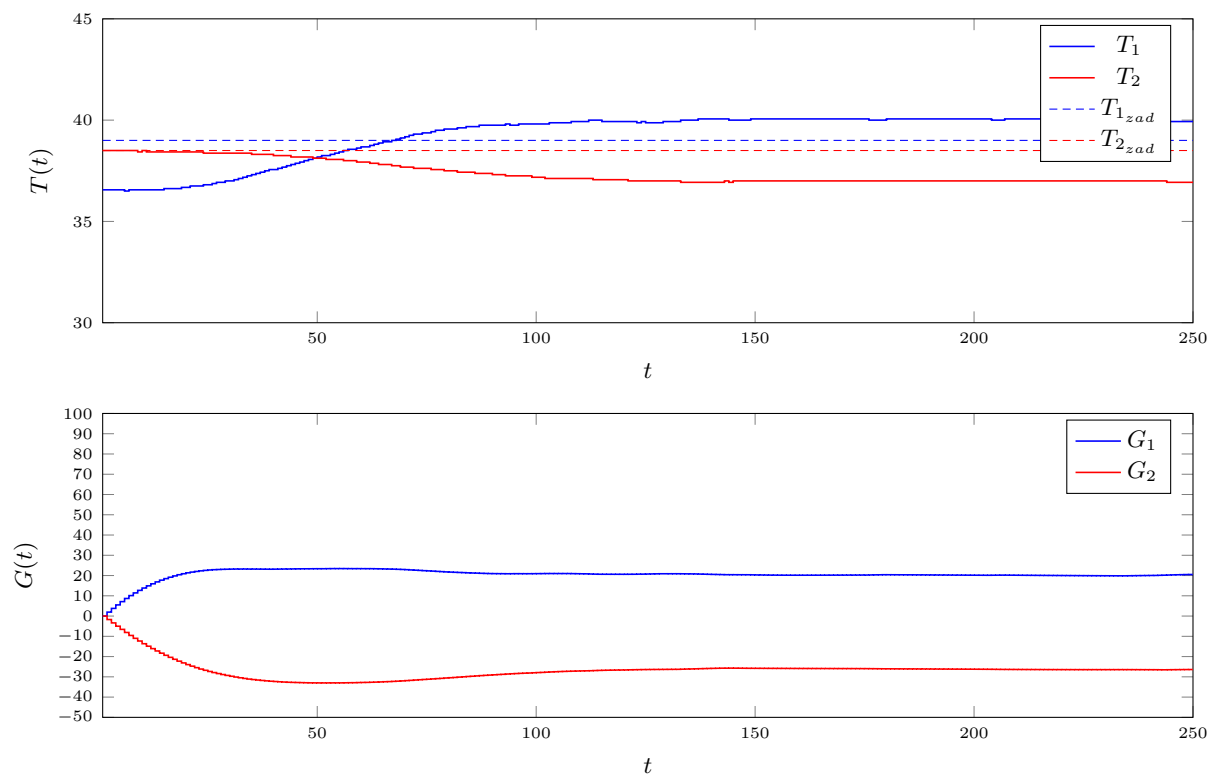
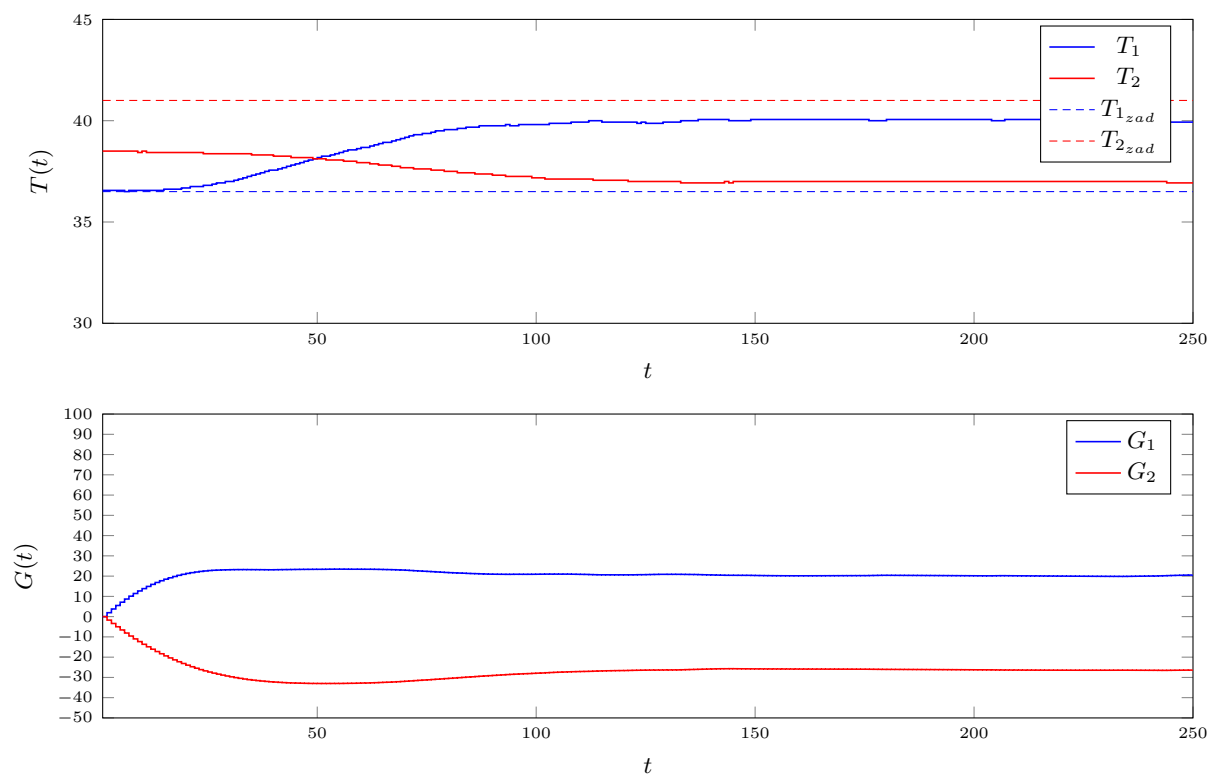
$$T1_{zad} = 40, T2_{zad} = 37 \rightarrow E = 3109,7$$

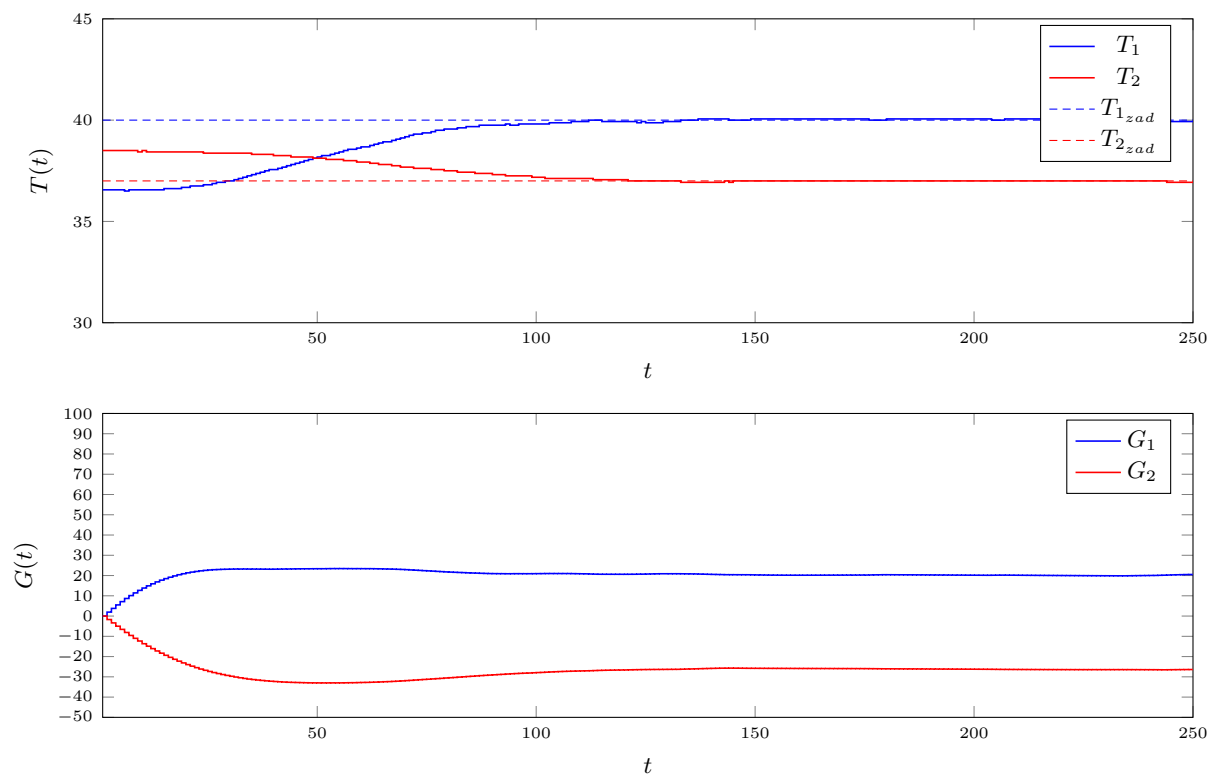
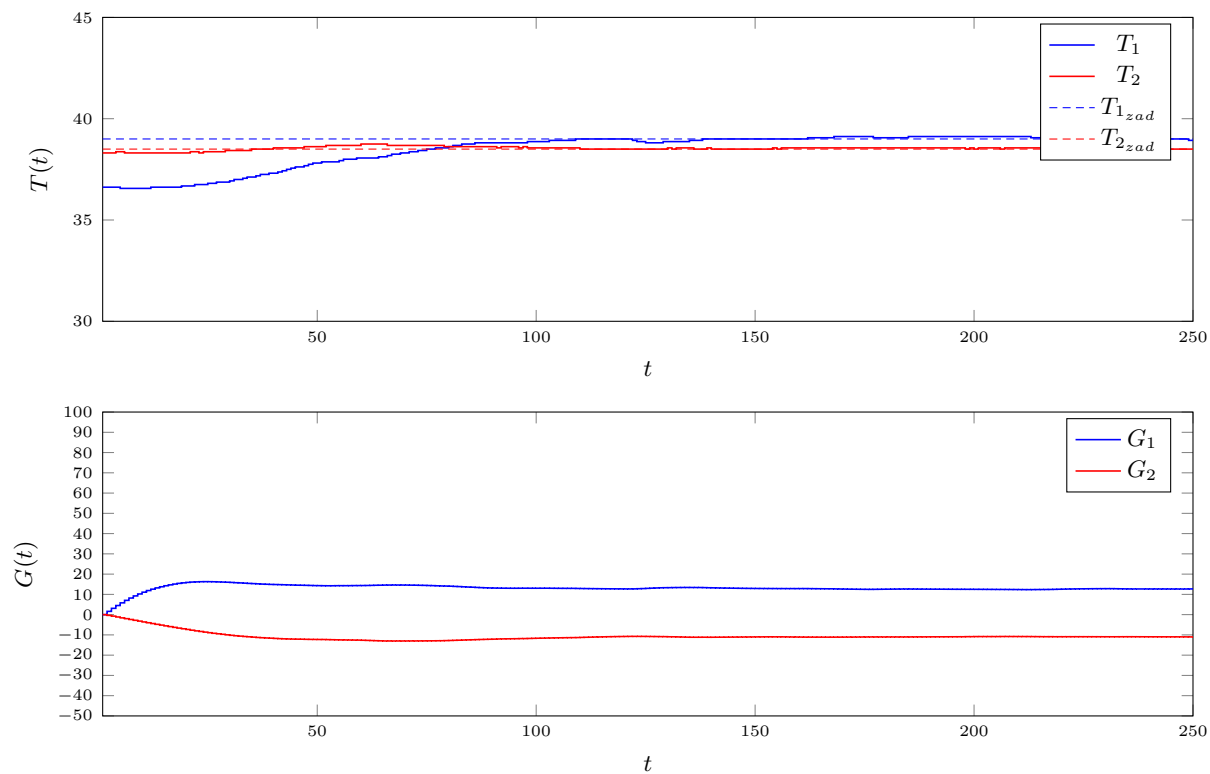
— DMC

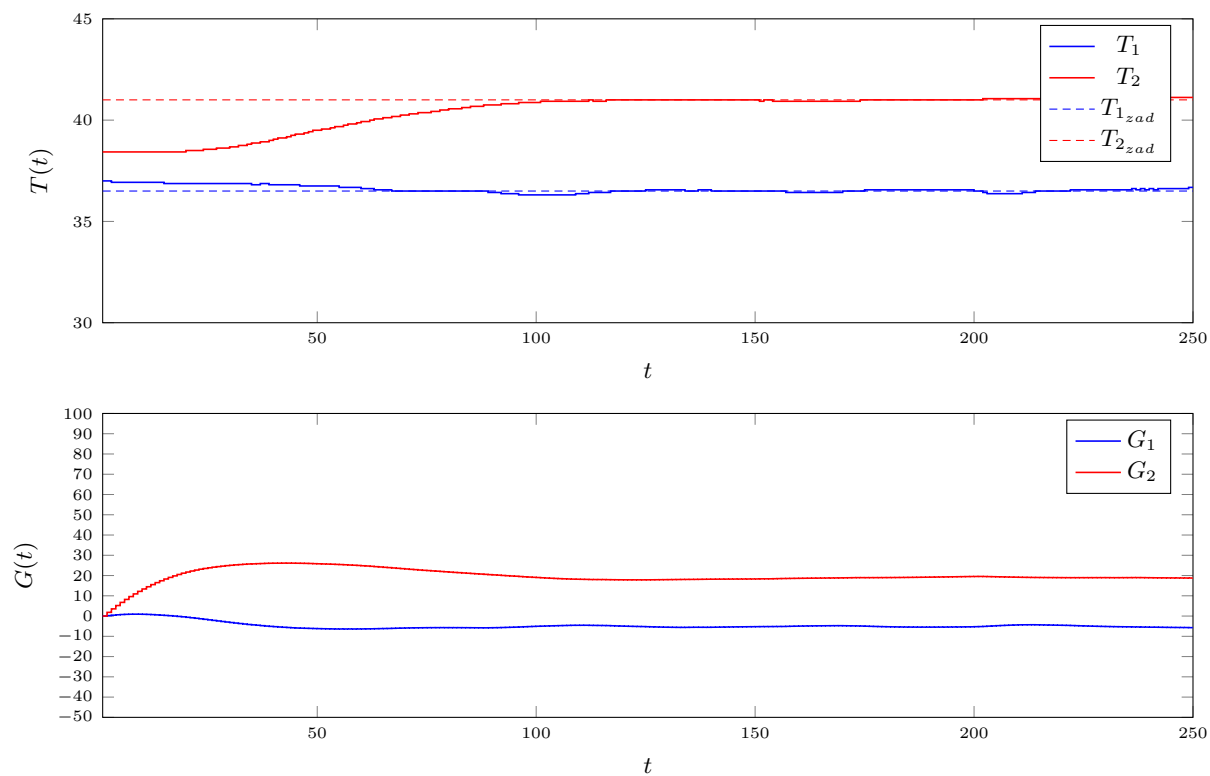
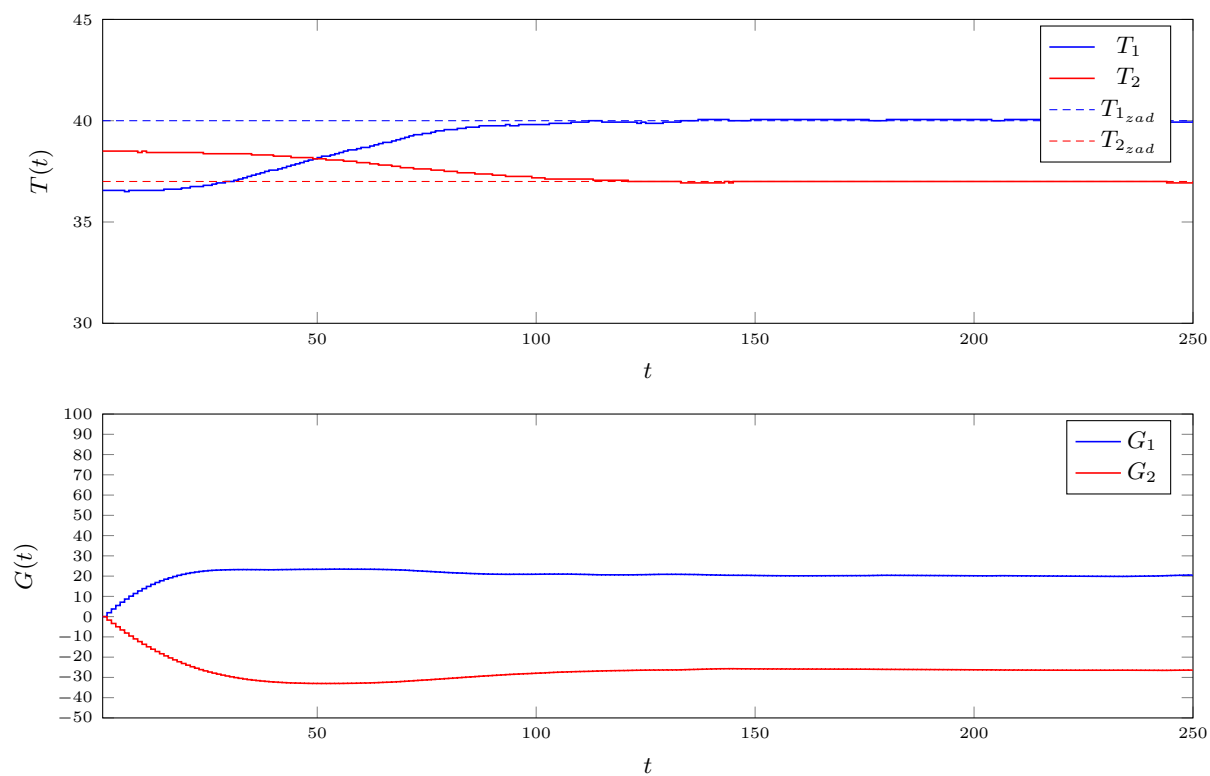
$$T1_{zad} = 39 \rightarrow E = 3240,8$$

$$T2_{zad} = 41 \rightarrow E = 3310$$

$$T1_{zad} = 40, T2_{zad} = 37 \rightarrow E = 3561,3$$

Rys. 5.1. Regulator PID przy skoku wartości zadanej T_1 do 39Rys. 5.2. Regulator PID przy skoku wartości zadanej T_2 do 41

Rys. 5.3. Regulator PID przy skoku wartości zadanej T_1 do 40 oraz T_2 do 37Rys. 5.4. Regulator DMC przy skoku wartości zadanej T_1 do 39

Rys. 5.5. Regulator DMC przy skoku wartości zadanej T_2 do 41Rys. 5.6. Regulator DMC przy skoku wartości zadanej T_1 do 40 oraz T_2 do 37