Wydział Elektroniki i Technik Informacyjnych
Politechnika Warszawska

Projektowanie układów sterowania (projekt grupowy)

Sprawozdanie z ćwiczenia laboratoryjnego nr 2

Mateusz Koroś, Ksawery Pasikowski, Mateusz Morusiewicz

Warszawa, 2017

Spis treści

1. Zad. 1

Możliwość sterowania i pomiaru w komunikacji ze stanowiskiem została sprawdzona poprzez funkcję readMeasurements() oraz sendControlsToG1AndDisturbance(u, z). Sygnały sterujące, które były obsługiwane to moc na grzałce G1 oraz moc wiatraka $W1$, gdzie moc na grzałce była sumą mocy podanej jako sterowanie oraz zakłócenie, stąd $G1 = u + K * z$ natomiast mierzona była temperatura $T1$ w otoczeniu grzałki $G1$. W punkcie pracy, tzn. dla $(W1, G1, Z) = (50, 29, 0)$ pomiar temperatury wyniósł $35°C$.

2. Zad. 2

Odpowiedzi skokowe toru zakłócenie-wyjście dla trzech różnych zmian zakłócenia zostały przedstawione na wykresie **??**

Właściwości statyczne obiektu można określić jako liniowe, gdyż zmiana zakłócenia powoduje liniową zmianę sygnału wyjściowego. Wzmocnienie statyczne procesu zostało obliczone ze wzoru:

$$K_{st} = \frac{\Delta y}{\Delta z} = 0{,}3$$

3. Zad. 3

Najlepiej nadającą się odpowiedzią skokową sygnału sterującego była ta dla skoku sterowania $\Delta u = 10$. Odpowiedź ta została przekształcona do postaci wykorzystywanej w algorytmie DMC w następujący sposób:

```
S = (S - 35) / 10;
S = S(32:end);
```

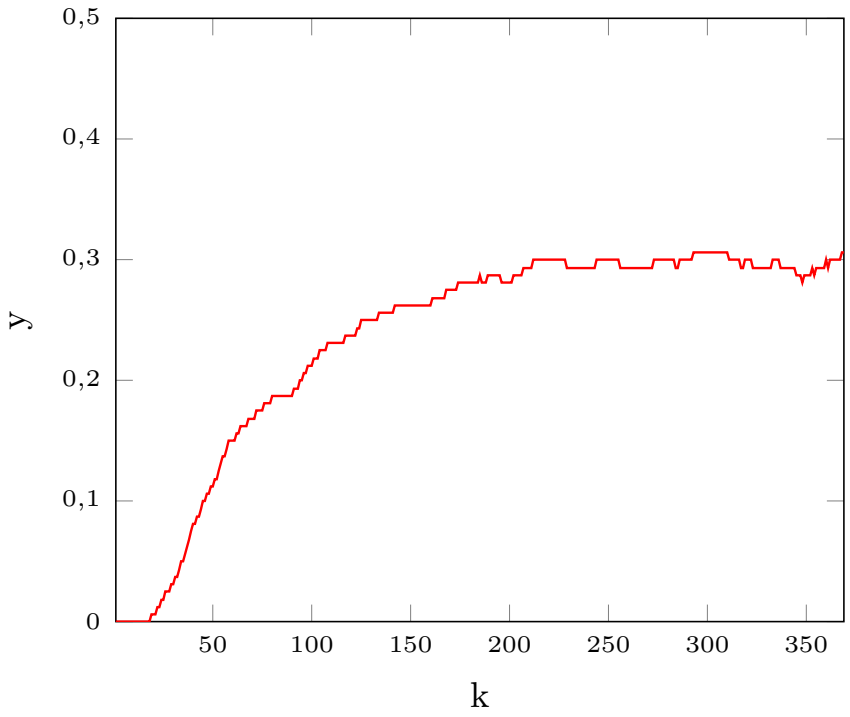
Odpowiedź ta została przedstawiona na wykresie 3.1. Analogicznie została przekształcona odpowiedź skokowa zakłócenia:

```
S = (S - 35) / 30;
S = S(32:end);
```

Dokonano tu dzielenia przez 30, gdyż o taką wartość zostało zmienione zakłócenie. Odpowiedź skokową zakłócenia widać na wykresie **??**

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

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



Rys. 3.1. Odpowiedź skokowa sygnału sterującego

Gdzie:

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

Po przekształceniu powyższego równania otrzymujemy równanie różnicowe postaci:

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

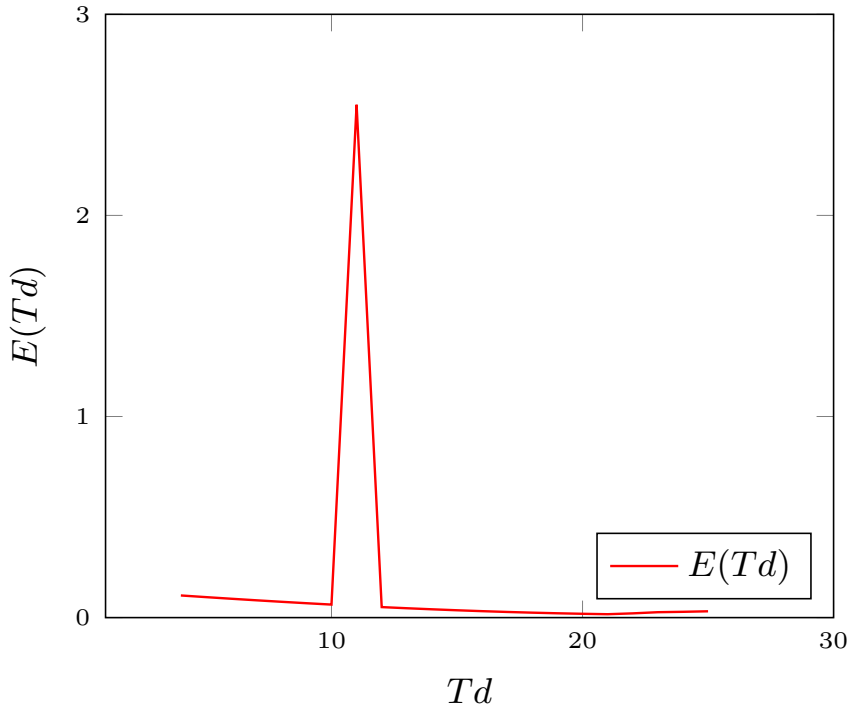
W celu doboru parametrów modelu została użyta funkcja wewnętrzna środowiska MATLAB: fmincon(). Parametry modelu, zwrócone przez ową funkcję to

$$T1 = 3{,}609\,17 \quad T2 = 61{,}238\,79 \quad K = 0{,}301\,64$$

dla odpowiedzi skokowej sterowania oraz

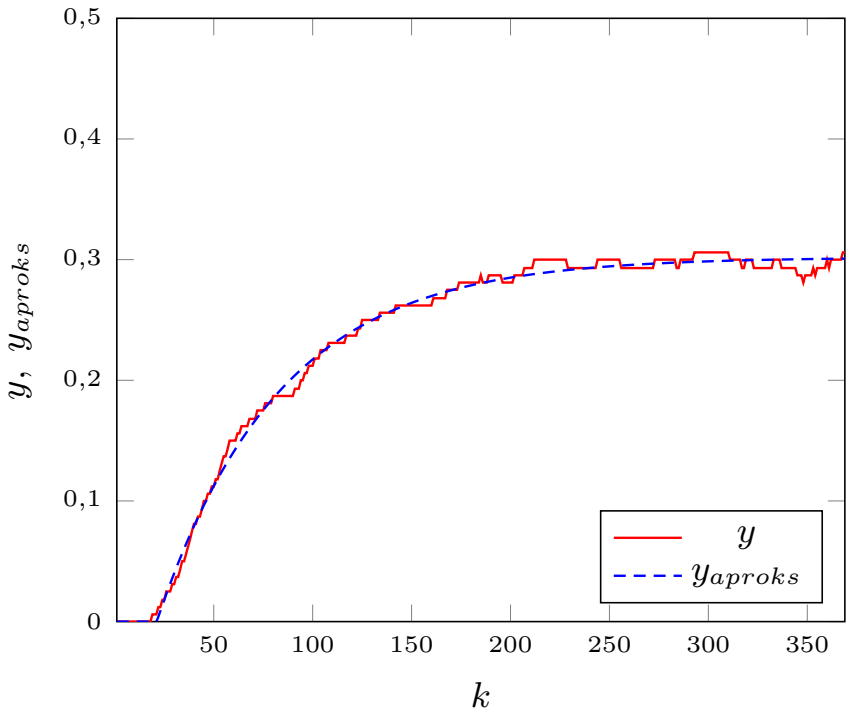
$$T1 = 81{,}442\,47 \quad T2 = 0{,}608\,11 \quad K = 0{,}150\,47$$

dla odpowiedzi skokowej zakłócenia. 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 skokowej sterowania widać na rysunku 3.2, natomiast dla odpowiedzi skokowej zakłócenia na rysunku **??**. Dla powyższych parametrów wyniósł on 0,0137 dla odpowiedzi skokowej sterowania oraz 0,0173 dla odpowiedzi



Rys. 3.2. Błąd średniokwadratowy odpowiedzi skokowej sterowania w zależności od T_d

skokowej zakłócenia. Porównanie odpowiedzi skokowej oryginalnej oraz wersji aproksymowanej widać na rysunku 3.3 dla odpowiedzi skokowej sterowania oraz 3.3 dla odpowiedzi skokowej zakłócenia. Jak widać na wykresach, funkcja aproksymująca jest bardzo dobrym przybliżeniem oryginalnego przebiegu. Sumaryczny błąd jest niewielki.



Rys. 3.3. Odpowiedź skokowa sygnału sterującego: oryginalna i aproksymowana

4. Zad. 4

Implementacja algorytmu DMC została dokonana za pomocą funkcji, której argumentami są horyzont dynamiki dla obu torów, horyzonty predykcji oraz sterowania, parametr λ, długość trwania regulacji w sekundach, wektor wartości zakłóceń w czasie oraz flaga, której ustawienie powoduje uwzględnienie zakłóceń w regulacji. Funkcja zwraca przebieg sterowania, wyjście obiektu oraz błąd, będący wskaźnikiem jakości regulacji:

```
function [ y, u, E ] = policzDMC( D_, Dz_, N_, Nu_, lambda_, Kk_, z, dist_mea
Upp = 29;
Zpp = 0;
Ypp = 34.5;

Kk = Kk_;

D = D_;
Dz = Dz_;
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('odpskok_y_apr');
s = s.Sapr;
s(length(s) : 400) = s(length(s));

s_z = load('odpskok_z_apr');
s_z = s_z.Sapr;
s_z(length(s_z) : 400) = s_z(length(s_z));

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

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

du = (zeros(1,Kk))' ;
dz = (zeros(1,D-1))' ;

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

Mpz = zeros(N, Dz-1) ;
for i = 1:N
for j = 1:(Dz-1)
if(i+j <= N)
Mpz(i,j) = s_z(i+j) - s_z(j) ;
else
Mpz(i,j) = s_z(N) - s_z(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;

kz = K(1,:)*Mpz;

addpath ('F:\SerialCommunication'); % add a path
initSerialControl COM16 % 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 skladowej 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

dz(k) = z(k) - z(k-1);
sum2 = 0;
for j = 1:Dz-1
if(k-j > 0)
sum2 = sum2 + kz(j)*dz(k-j);
end
end

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

if dist_measure == 1
du(k) = du(k) - sum2;
end

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

sendControlsToG1AndDisturbance(u(k)+Upp, z(k));

stairs(y);
pause(0.01);

waitForNewIteration();

end
```

```
% wska?nik jako?ci regulacji
E = (yzadVec - y)*(yzadVec - y)';
end
```

Wartością parametru λ, używaną do symulacji było 3, gdyż dawała akceptowalne wyniki a na eksperymentalne dobieranie parametru zabrakło czasu. Horyzonty natomiast miały wartość $D = D_z = N = N_u = 200$. Ograniczenia zostały uwzględnione poprzez rzutowanie po obliczeniu wartości sterowanie

5. Zad. 5

TODO W następnej próbie regulatora dokonano najpierw skoku wartości zadanej z punktu pracy do TODO, następnie w TODO chwili nastąpił skok zakłócenia z 0 do 15. Wyniki tego eksperymentu widać na rysunku **??**. Jak widać, regulator bardzo dobrze poradził sobie z zakłóceniem, powrót do wartości zadanej był szybki i wyniósł ok. $D^z = 50$ sekund. Po wyłączeniu uwzględnienia sterowania w regulatorze, zmiana temperatury była zauważalnie wyższa. Świadczy to o tym, że pomiar i uwzględnienie zakłócenia prowadzi do lepszej regulacji niż w przypadku braku tych czynności. Niestety na wzbudzenie zakłócenia do innych wartości zabrakło czasu, co było spowodowane błędami ze sprzętem, który odmawiał poprawnej komunikacji. Duże opóźnienie czasowe było spowodowane podejrzeniem, że winę za błędy w sterowaniu ponosił algorytm. Szukanie błędu w kodzie kosztowało sporo czasu, jednak poprawa nastąpiła dopiero po wyłączeniu i włączeniu obiektu oraz dociśnięciu kabli. Niestety było już za późno na dokonanie wszystkich eksperymentów, jednak powyższy jest wystarczającym dowodem na to, iż warto uwzględniać zakłócenia w regulatorze, gdyż regulacja jest dzięki temu dużo lepsza.