Wydział Elektroniki i Technik Informacyjnych
Politechnika Warszawska

Projektowanie układów w sterowania (projekt grupowy)

Sprawozdanie z projektu i ćwiczenia laboratoryjnego nr 2, zadanie nr 1

Zespół Z01
Hubert Kozubek, Przemysław Michalczewski

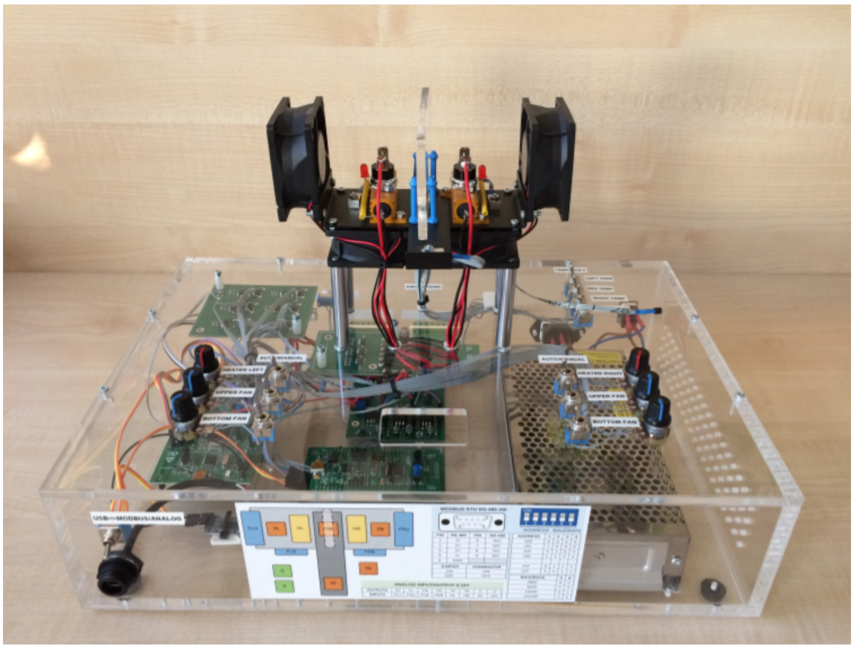
Warszawa, 2021

Spis treści

1. Laboratorium

1. Cel laboratorium

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



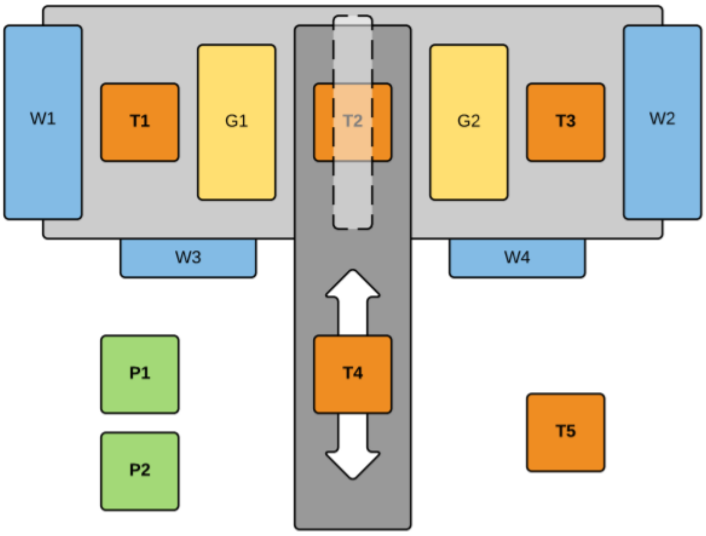
Rys. 1.1. Stanowisko grzejąco-chłodzące używane w trakcie laboratoriów.

2. Przebieg laboratorium

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

W ramach laboratorium należało wykonać 5 zadań:

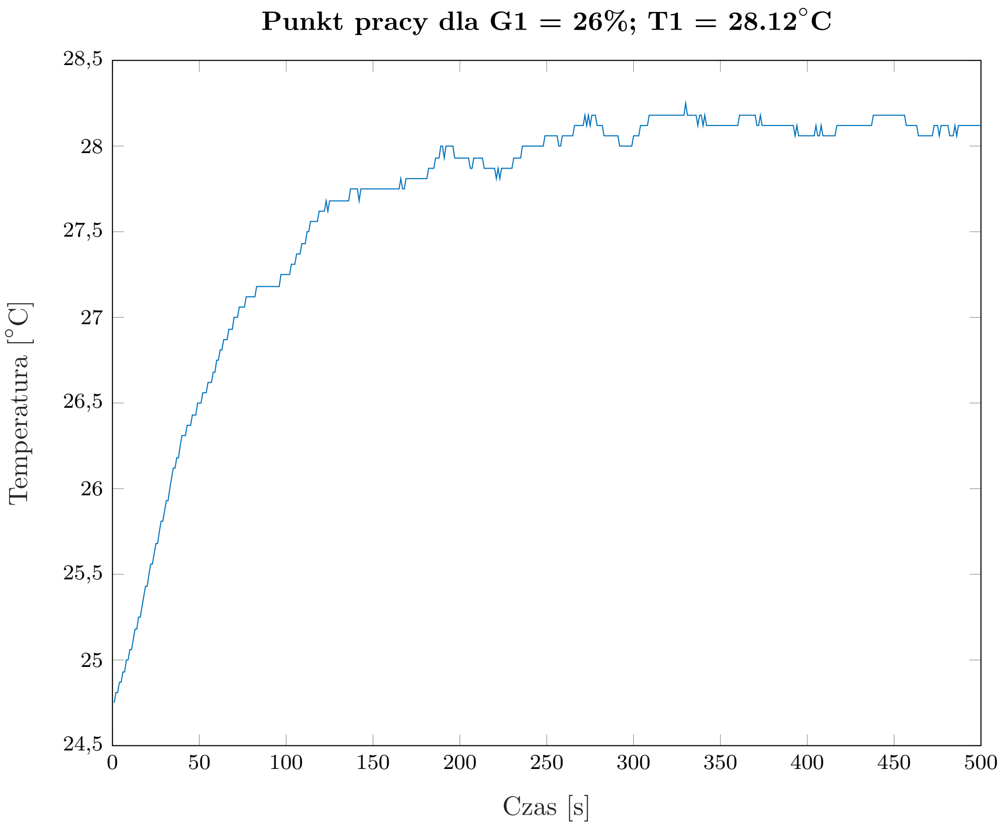
1. Odczytać wartość pomiaru temperatury dla termometru T1 dla mocy 26% grzałki G1 w stanie ustalonym (wyznaczyć punkt pracy).
2. Wyznaczyć odpowiedzi skokowe toru zakłócenie-wyjście dla trzechróżych zmian sygnału zakłócającego Z rozpoczynając z punktu pracy.
3. Przygotować odpowiedzi skokowe wykorzystywane w algorytmie DMC.
4. Zaimplementować algorytm DMC do regulacji procesu stanowiska w języku MATLAB.
5. Dobrać parametr D^z dla algorytmu DMC i przeprowadzić eksperymenty.



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

3. Punkt pracy stanowiska

W celu wyznaczenia punktu pracy stanowiska dla mocy grzałki G1=26% zadano tę wartość dla sygnału sterującego grzałką. Następnie poczekano, aż temperatura T1 ustali się. Wynik eksperymentu przedstawiono na rys. 1.3. Odczytana wartość temperatury dla termometru T1 wyniosła 28,12 °C.

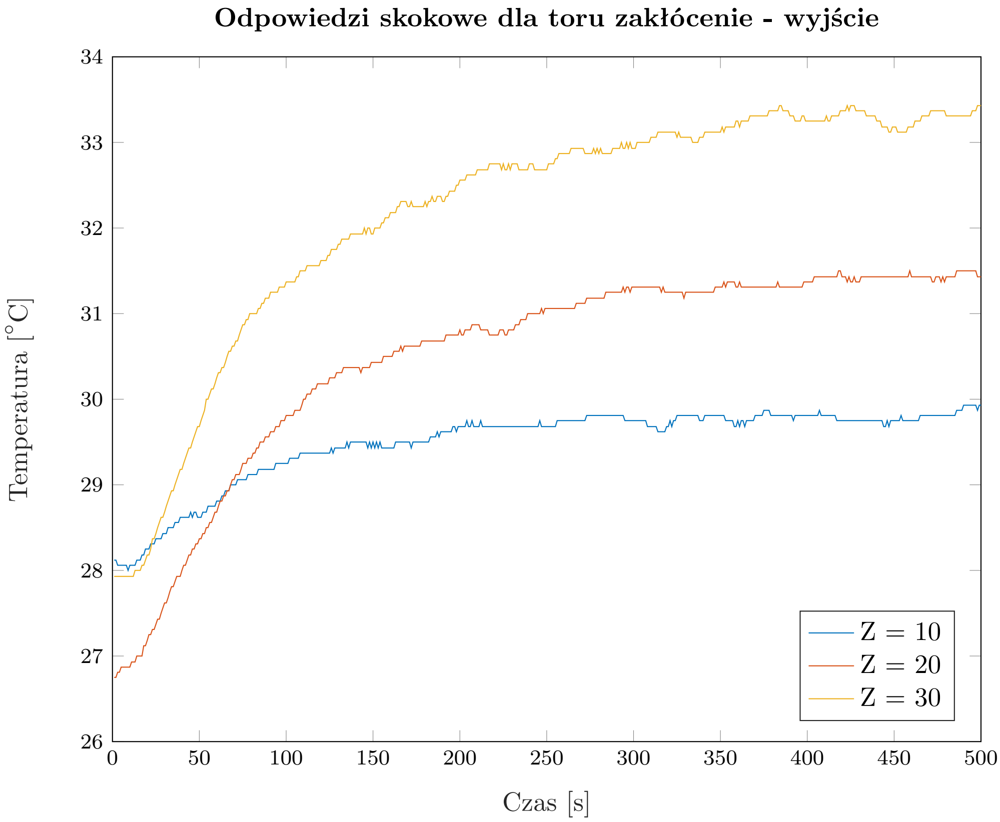


Rys. 1.3. Ustalanie się temperatury dla punktu pracy.

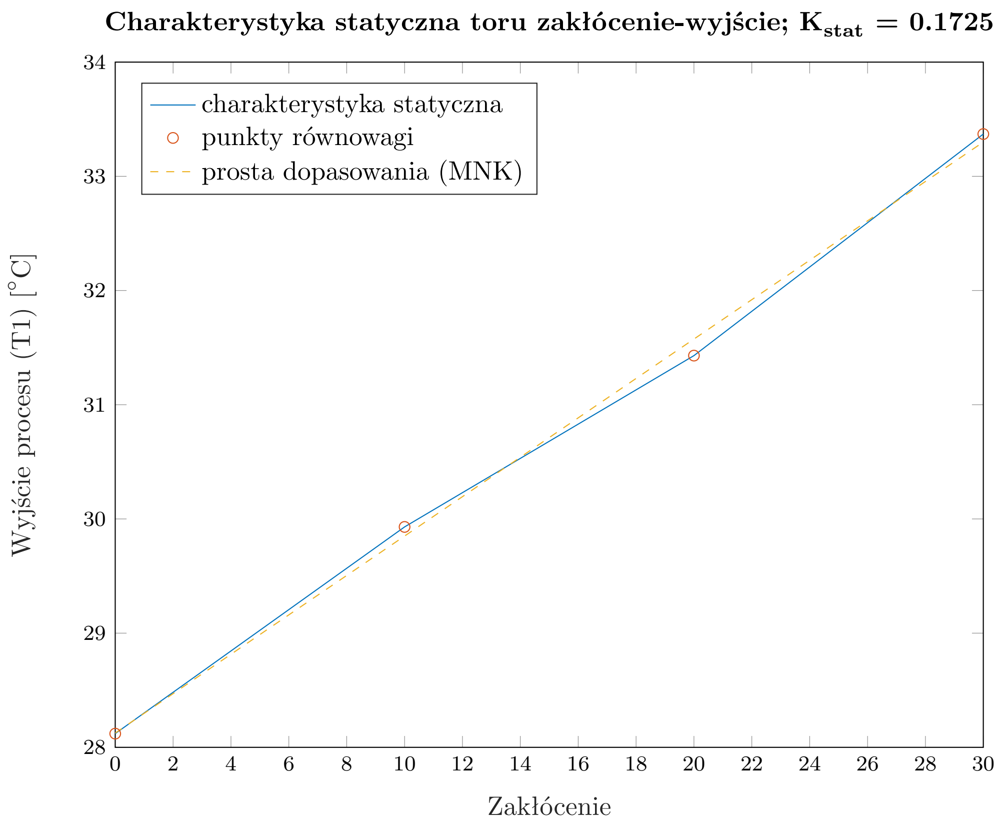
4. Odpowiedzi skokowe toru zakłócenie-wyjście

Dla stanowiska pracującego w ustalonym punkcie pracy (G1=26%; T1=28,12°C) zadano 3 różne wartości skoku zakłócenia. Eksperyment wykonano dla skoków sygnału zawsze z wartości Z=0 do kolejno wartości: Z=10, Z=20 oraz Z=30. Wyniki przedstawiono na rys. 1.4. Różnica w początkowych wartościach temperatury T1 dla poszczególnych skoków wynika z zakłóceń powodowanch przez zmianę temperatury w pracowni laboratoryjnej oraz nagrzewania się stanowiska grzejąco-chłodzącego.

Na podstawie wyznaczonej charakterystyki statycznej dla toru zakłócenie-wyjście przedstawionej na rys. 1.5 możemy stwierdzono, że właściwości statyczne są w przybliżeniu liniowe. Wzmocenienie statyczne dla tego toru procesu wyznaczono metodą najmniejszych kwadratów i otrzymano wartość $K_{\mathrm{stat}} = 0.1725$.



Rys. 1.4. Odpowiedzi skokowe dla toru zakłócenie - wyjście.



Rys. 1.5. Charakterystyka statyczna dla toru zakłócenie - wyjście.

5. Odpowiedzi skokowe dla algorytmu DMC

W celu wyznaczenia odpowiedzi skokowej toru zakłócenie-wyjście wybrano trzecią odpowiedź skokową przedstawioną na rys. 1.4, tj. skok zakłócenia do wartości Z=30. Do przekształcenia zebranej odpowiedzi skokowej skorzystano z przekształcenia:

$$S(i) = \frac{Y(i) - Y_{\text{pp}}}{Z_{\text{skok}} - Z_{\text{pp}}} \tag{1.1}$$

gdzie:

— $S(i)$ - odpowiedź skokowa potrzebna do algorytmu DMC,
— $Y(i)$ - odpowiedź skokowa przed przekształceniem,
— Y_{pp} - wartość wyjścia w chwili k=0 (tutaj Y_{pp} = 27,93),
— Z_{skok} - wartość sterowanie w chwili k=0 i później (tutaj $Z_{\text{skok}} = 30$),
— Z_{pp} - wartość sterowania przed chwilą k=0 (tutaj $Z_{\text{pp}} = 0$)

Otrzymana odpowiedź skokowa dla toru zakłócenie wyjście została przedsawiona na rys. 1.6. Następnie dokonano aproksymacji odpowiedzi skokowej poprzez przybliżenie używając w tym celu członu inercyjnego drugiego stopnia z opóźnieniem:

$$G(s) = \frac{K}{(sT_1 + 1)(sT_2 + 1)} e^{-T_{\text{d}}s} \tag{1.2}$$

Po dyskretyzacji danej transmitancji otrzymujemy:

$$G(z) = \frac{b_1 z^{-1} + b_2 z^{-2}}{1 + a_1 z^{-1} + a_2 z^{-2}} z^{-T_{\text{d}}} \tag{1.3}$$

gdzie:

$$\begin{aligned}
a_1 &= -\alpha_1 - \alpha_2 \\
a_2 &= \alpha_1 \alpha_2 \\
\alpha_1 &= e^{-\frac{1}{T_1}} \\
\alpha_2 &= e^{-\frac{1}{T_2}} \\
b_1 &= \frac{K}{T_1 - T_2}[T_1(1 - \alpha_1) - T_2(1 - \alpha_2)] \\
b_2 &= \frac{K}{T_1 - T_2}[\alpha_1 T_2(1 - \alpha_2) - \alpha_2 T_1(1 - \alpha_1)]
\end{aligned} \tag{1.4}$$

Dla otrzymanej w ten sposób transmitancji w postaci dyskretnej napisano funkcję `AproksSkokZak_DMC(X)`, która przyjmowała parametry T_1, T_2 oraz K. Wartość parametru $T_{\text{d}} = 12$ odczytano z odpowiedzi skokwej, ponieważ jest to opóźnienie. Funkcja ta zwracała sumaryczny błąd kwadratowy pomiędzy aproksymacją otrzymaną dla zadanych parametrów a eksperymentalnie wyznaczoną odpowiedzią skokową dla toru zakłócenie-wyjście po przekształceniu dla algorytmu regulacji DMC. Implementacja funkcji `AproksSkokZak_DMC(X)`:

```
% aproksymacja odpowiedzi skokowej toru zaklocenie-wyjscie

function ERR = AproksSkokZak_DMC(X)

    data = load('S_z.mat');
    S_z = data.S_z;

    time = length(S_z);
```

```
    T1 = X(1);
    T2 = X(2);
    K = X(3);
    Td = 12;
    z(1:time) = 0;

    alpha1 = exp(-1/T1);
    alpha2 = exp(-1/T2);
    a1 = -alpha1-alpha2;
    a2 = alpha1*alpha2;
    b1 = K*(T1*(1- alpha1)-T2*(1-alpha2))/(T1-T2);
    b2 = K*(alpha1*T2*(1-alpha2)-alpha2*T1*(1-alpha1))/(T1-T2);

    for k = Td+3:time
        z(k) = b1 + b2 - a1*z(k-1) - a2*z(k-2);
    end

    e = S_z - z';

    ERR = (norm(e))^2;
end
```

Do wyznaczenia współczynników transmitancji użyto funkcji `ga` dostępnej w programie MATLAB, która wykorzystuje algorytm generyczny do wyznaczenia minimum funkcji. Dla naszej funkcji jest to równoważne ze znalezieniem najlepiej dopasowanej aproksymacji odpowiedzi skokowej toru zakłócenie-wyjście. Paramerty związane z algorytmem generycznym to kolejno:

$$\begin{aligned} StallGenLimit &= 200 \\ PopulationSize &= 400 \\ FunctionTolerance &= 1\mathrm{e}^{-8} \end{aligned} \tag{1.5}$$

gdzie:

— *StallGenLimit* oznacza maksymalną ilość iteracji algorytmu, dla których wartość różnicy między wartościami wyniku funkcji optymalizowanej jest mniejsza niż *FunctionTolerance*.
— *PopulationSize* oznacza ilość ziaren dla algorytmu generytecznego w każdej iteracji.
— *FunctionTolerance* oznacza wartość wskaźnika toleracji, który determinuje dla jakiej dokładoności algorytm uznaje, że otrzymany wynik jest ostateczy.

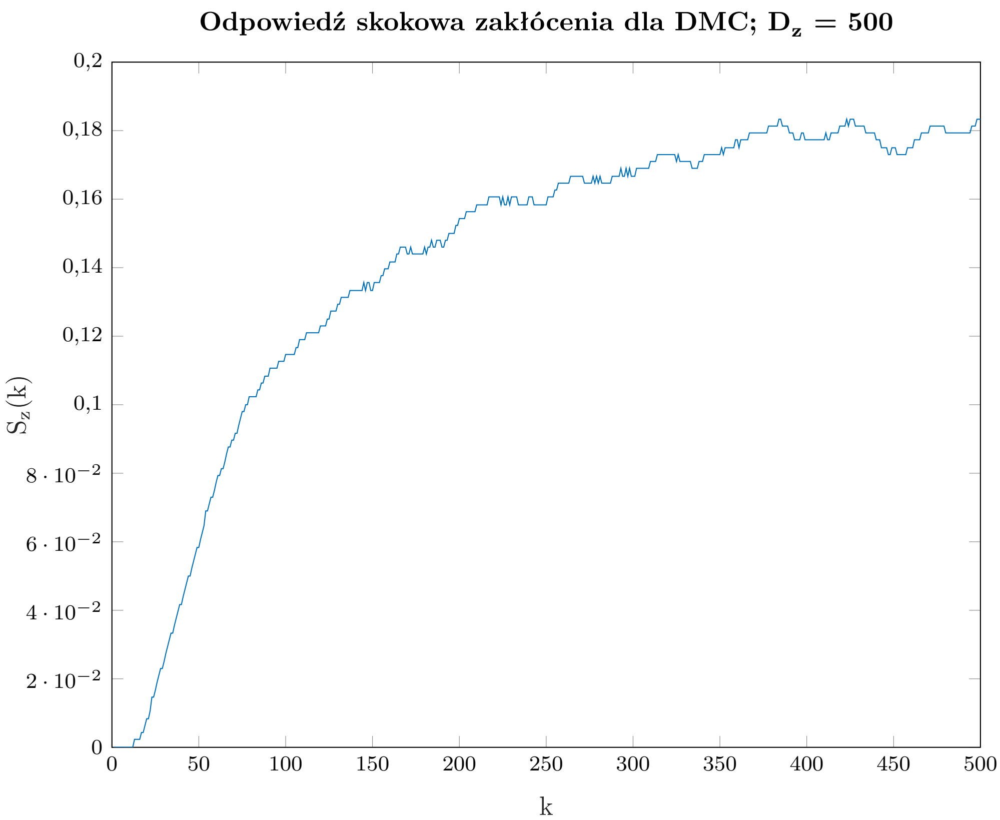
Wybrane wartości powyższych parametrów pozwalają na aproksymację z małym błędem, co sprawdzono poprzez kilkukrotne wywołanie algorytmu i dostrojenie eksperymentalne parametrów. Implementacja optymalizacji:

```
ERR = @(X)AproksSkokZak_DMC(X);

options = optimoptions('ga','StallGenLimit', 200, ...
    'PopulationSize', 400, 'FunctionTolerance', 1e-8);
[x_apro_zak, Err_apro_zak] = ga(ERR, 3, [], [], [],...
    [], [], [], [], [], options);

fprintf('\nT1 = %f; T2 = %f; K = %f;\n', x_apro_zak)
% T1=88.311400; T2=0.010351; K=0.177293;
```

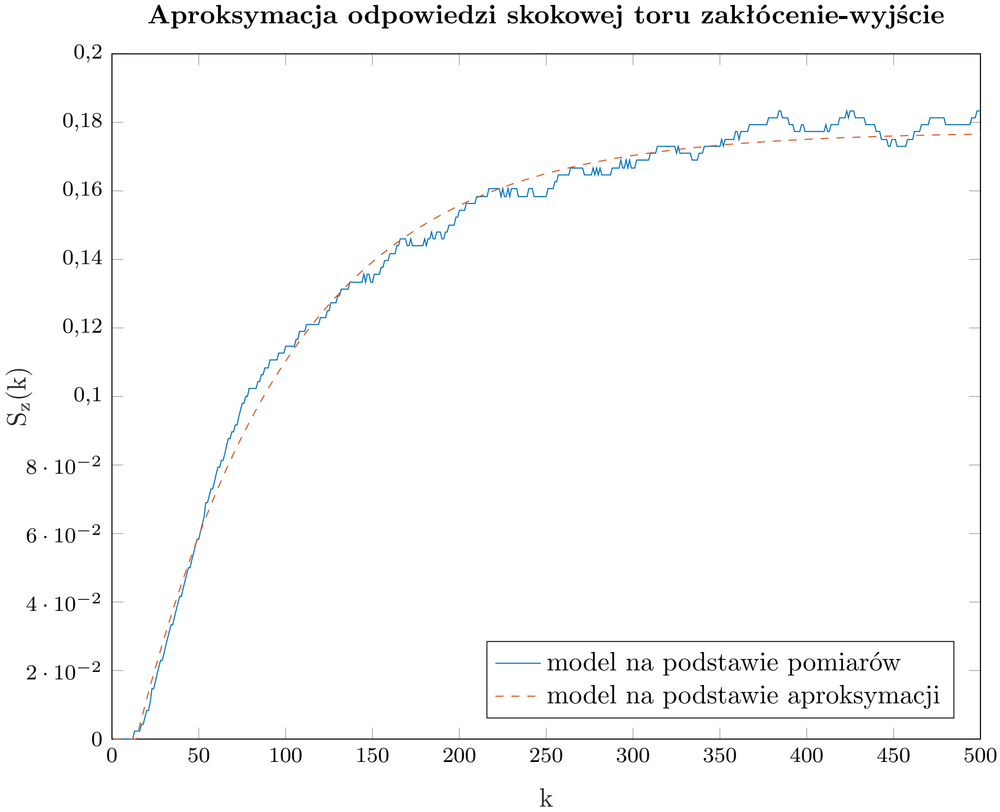
Ponieważ do minimalizacji funkcji użyto algorytmu generycznego to wyniki podczas kolejnych odtworzeń mogą się różnic. Jednak po wykonaniu testów, dla których błąd aproksymacji był bardzo zbliżony przyjęto wartości parametrów:

$$\begin{aligned} K &= 0.1772938 \\ T_1 &= 88.311400 \\ T_2 &= 0.010351 \\ T_\mathrm{d} &= 12 \end{aligned} \tag{1.6}$$

Otrzymaną aproksymowaną odpowiedź skokową dla toru zakłócenie-wyjście dla algorytmu regulacji DMC przedstawiono na rys. 1.7.

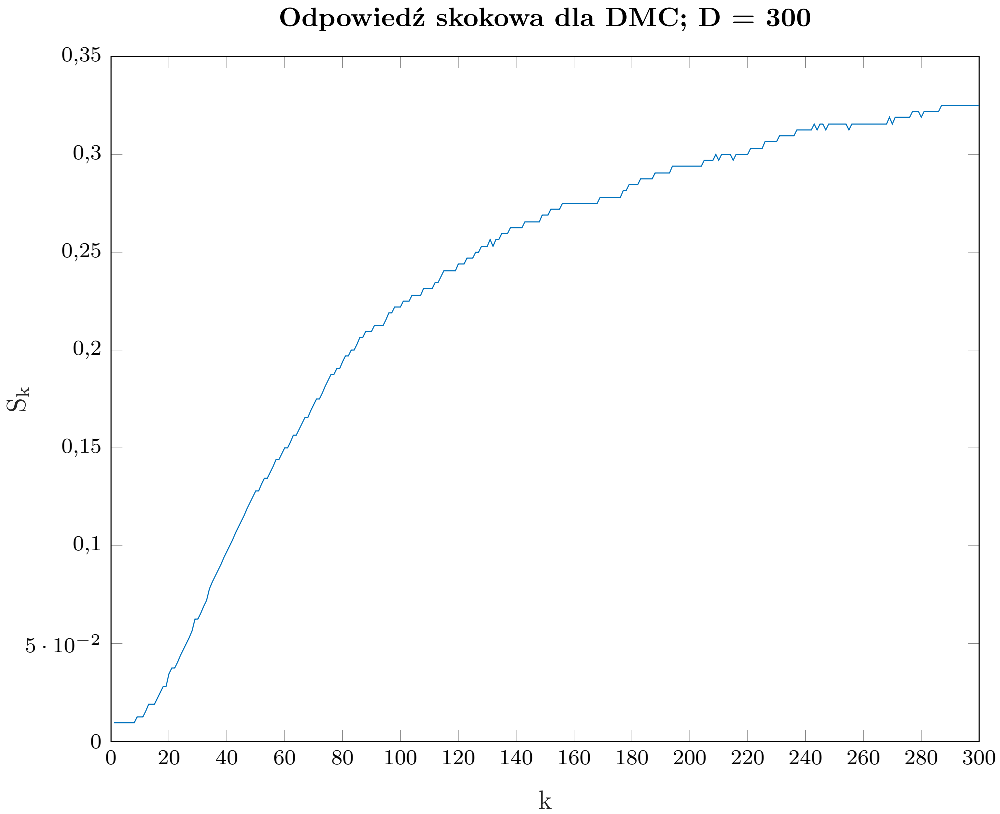


Rys. 1.6. Odpowiedź skokowa toru zakłócenie-wyjście dla algorytmu DMC.

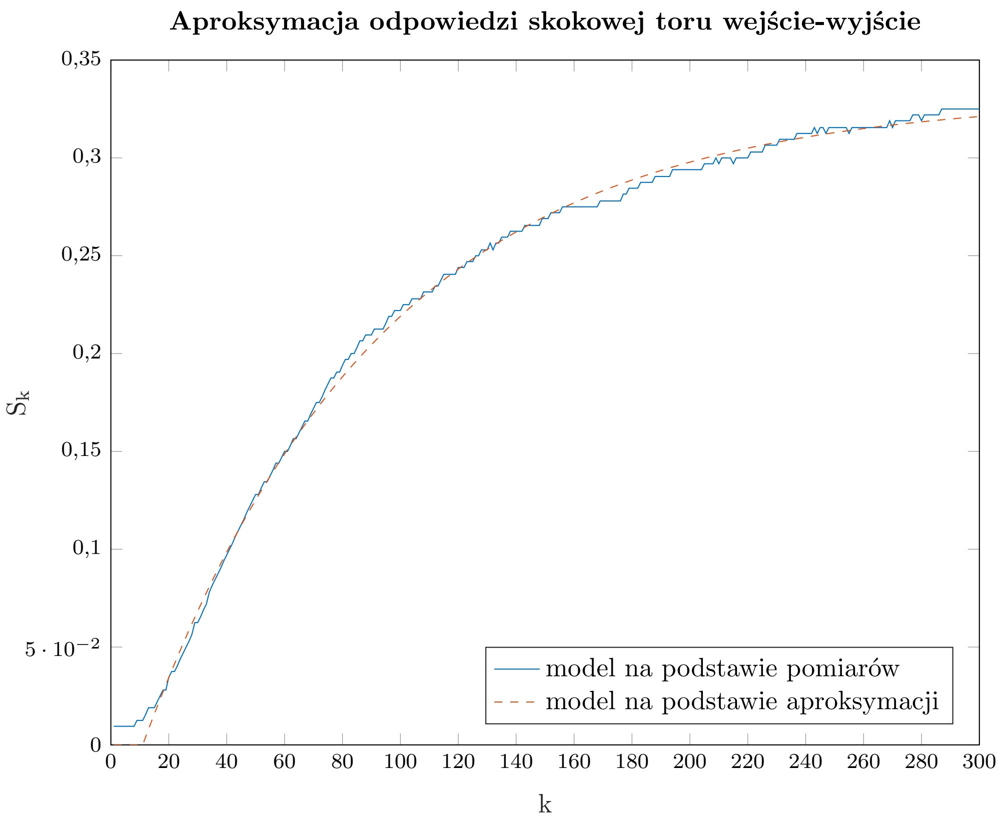


Rys. 1.7. Aproksymacja odpowiedzi skokowej toru zakłócenie-wyjście dla algorytmu DMC.

Odpowiedź skokowa toru wejście-wyjście dla algorytmu DMC została zaczerpnięta z laboratorium nr 1, przedstawiono ją na rys. 1.8. Analogicznie jak dla odpowiedzi skokowej toru zakłócenie-wyjście wykonano dla niej aproksymacje przedstawioną na rys. 1.9.



Rys. 1.8. Odpowiedź skokowa toru wejście-wyjście dla algorytmu DMC.



Rys. 1.9. Aproksymacja odpowiedzi skokowej toru wejście-wyjście dla algorytmu DMC.

6. Algorytm DMC z pomiarem zakłóceń

Do sterowania procesem zaimplementowano algorytm DMC z pomiarem zakłóceń w języku MATLAB. TODO(nie wiem bardziej omawiać implementacje, czy komentarze są wystarczające, bo w poleceniu jest aby oówić program)

```
%implementacja DMC z pomiarem zaklocen
function U = DMC_zak(yzad, y, D, z, Dz, N, Nu, lambda)

    persistent init
    persistent S
    persistent S_z
    persistent M
    persistent Mp
    persistent Mzp
    persistent K
    persistent dUP
    persistent dZP
    persistent zp
    persistent Upop


    if isempty(init)
        %liczone offline
        % Wczytanie macierzy S z pliku dane1.mat
        data = load('dane1.mat');
        S = data.S;

        % Wczytanie macierzy S_z
        data2 = load('S_z.mat');
        S_z = data2.S_Z;

%           data3 = load('S_z_apro.mat');
%           S_z = data3.S_Z_apro;


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

        % przedluzenie wektora S_z
        for i = Dz+1:Dz+N
            S_z(i) = S_z(Dz);
        end

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

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

        Mzp = zeros(N, Dz);
        Mzp(1:N, 1) = S_z(1:N);
        for i = 2:Dz
            Mzp(1:N, i) = S_z(i:N+i-1) - S_z(i-1);
        end

        I = eye(Nu);

        K = ((M'*M + lambda*I)^(-1))*M';
        dUP = zeros(D-1,1);
        dZP = zeros(Dz,1);
        Upop = 26;
        zp = 0;
        init = 1;
    end

    % Ograniczenia sterowania
    Gmax = 100;
    Gmin = 0;

    Y0 = zeros(N,1);
    dU = zeros(Nu,1);

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

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

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

    for n = Dz:-1:2
        dZP(n) = dZP(n-1);
    end
    dZP(1) = z - zp;
    zp = z;

    U = Upop + du;

    % ograniczenia sterowania
    if U > Gmax
```

```
        U = Gmax;
    end

    if U < Gmin
        U = Gmin;
    end

    Upop = U;
end
```

7. Dobór parametrów dla algorytmu DMC