

Wydział Elektroniki i Technik Informacyjnych
Politechnika Warszawska

Projektowanie układów sterowania
(projekt grupowy)

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

Imię i Nazwisko, Imię i Nazwisko, Imię i Nazwisko

Warszawa, 2017

Spis treści

1. Wstęp	2
1.1. Cel projektu	2
1.2. Opis algorytmów	2
1.2.1. PID	2
1.2.2. DMC	2
2. zad1	4
3. zad2	5
4. Odpowiedź skokowa	6
5. Implementacja PID	7
6. Implementacja DMC	9
7. zad5	11
8. zad6	12

1. Wstęp

1.1. Cel projektu

Celem projektu było zbadanie właściwości danego obiektu oraz próba regulacji z wykorzystaniem dyskretnych algorytmów PID oraz DMC w wersji analitycznej. Częścią zadania było również uwzględnienie ograniczeń sterowania narzuconych w treści projektu.

1.2. Opis algorytmów

1.2.1. PID

W zadaniu projektowym wykorzystany został regulator PID. Algorytm ten, na podstawie obliczonej wartości uchybu oraz dobranych nastaw, wyznacza wartość sterowania dla chwili k . Elementami struktury algorytmu są następujące stałe:

- K - stała proporcjonalna
- T_i - stała całkowania
- T_d - stała różniczkowania
- T - czas próbkowania

Dobranie nastaw algorytmu oznacza znalezienie możliwie optymalnych nastaw zapewniających najlepszą jakość regulacji.

Po wyznaczeniu parametrów, należy obliczyć współczynniki prawa regulacji używając następujących wzorów:

$$r_2 = \frac{KTd}{T} \quad (1.1)$$

$$r_1 = K\left(\frac{T}{2T_i} - \frac{2T_d}{T} - 1\right) \quad (1.2)$$

$$r_0 = K\left(\frac{T}{2T_i} + \frac{T_d}{T} + 1\right) \quad (1.3)$$

Prawo regulacji regulatora opisane jest równaniem:

$$u(k) = r_2e(k-2) + r_1e(k-2) + r_0e(k) + u(k-1) \quad (1.4)$$

1.2.2. DMC

Regulator DMC jest algorytmem predykcyjnym wyznaczającym trajektorię sygnału wyjściowego oraz przyszłe przyrosty sterowań. DMC potrzebuje wcześniejszej informacji o obiekcie w postaci odpowiedzi skokowej. Parametrami algorytmu są:

- D - horyzont dynamiki
- N - horyzont predykcji
- N_u - horyzont sterownia
- λ - kara za zmianę sterownia

Strojenie algorytmu polega na odpowiednim dobraniu parametrów tak, by zapewnić możliwie najlepszą jakość regulacji.

Aby otrzymać prawo regulacji, należy wyznaczyć szereg współczynników:
Macierz dynamiczną oraz macierz K :

$$M = \begin{bmatrix} s_1 & 0 & \cdots & 0 \\ s_2 & s_1 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ s_N & s_{N-1} & \cdots & s_{N-N_u+1} \end{bmatrix} \quad (1.5)$$

$$K = (M^T \Psi M + \Lambda)^{-1} M^T \Psi \quad (1.6)$$

Macierz M^P oraz wektor zmian sterowania ΔU^P :

$$M^P = \begin{bmatrix} s_2 - s_1 & s_3 - s_2 & \cdots & s_D - s_{D-1} \\ s_3 - s_1 & s_4 - s_2 & \cdots & s_{D+1} - s_{D-1} \\ \vdots & \vdots & \ddots & \vdots \\ s_{N+1} - s_1 & s_{N+2} - s_2 & \cdots & s_{N+D-1} - s_{D-1} \end{bmatrix} \quad (1.7)$$

$$\Delta U^P(k) = \begin{bmatrix} \Delta u(k-1) \\ \Delta u(k-2) \\ \vdots \\ \Delta u(k-(D-1)) \end{bmatrix} \quad (1.8)$$

Na podstawie powyższych macierzy oraz wektorów, można obliczyć parametry reguлятора:

$$k_e = \sum_{i=1}^N K_{1,i} \quad (1.9)$$

$$k_u = \overline{K}_1 M^P \quad (1.10)$$

a następnie wyznaczyć sterowanie z następującego prawa regulacji:

$$e(k) = y_{zad}(k) - y(k) \quad (1.11)$$

$$u(k|k) = u(k-1) + k_e e(k) - k_u \Delta U^P(k) \quad (1.12)$$

Ograniczenie wartości sygnału sterującego przez wartości maksymalną i minimalną wykonane jest w następujący sposób:

1. jeżeli $u(k|k) < u_{min}$ wtedy $u(k|k) = u_{min}$
2. jeżeli jeżeli $u(k|k) > u_{max}$ wtedy $u(k|k) = u_{max}$
3. $u(k) = u(k|k)$

2. zad1

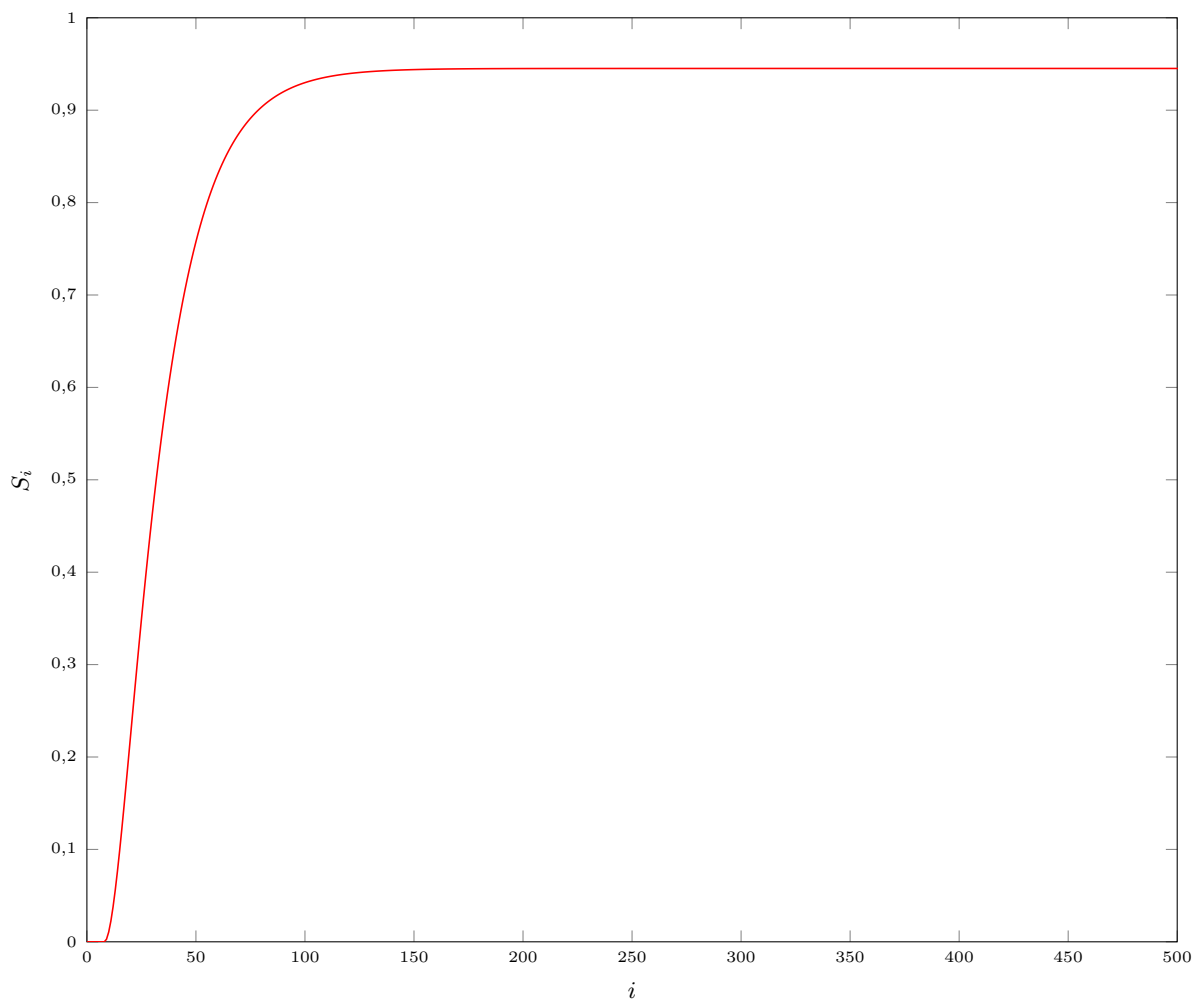
3. zad2

4. Odpowiedź skokowa

Odpowiedź skokowa to odpowiedź układu na wymuszenie w postaci skoku jednostkowego w chwili $k = 0$. Z powodu ograniczeń sterowania obiektu, skok jednostkowy jest niemożliwy, dlatego należy przeskalować odpowiedź skokową w następujący sposób:

$$\Delta S_i = \frac{S_i^0 - Y_{pp}}{\Delta U}, \text{ dla } i = 1, \dots, N \quad (4.1)$$

gdzie N to wybrana długość odpowiedzi skokowej. Odpowiedź skokowa prezentowana na wykresie 4.1 została otrzymana z przeskalowania odpowiedzi obiektu po skoku z $u_{pp} = 1,1$ do $u = 1,4$.



Rys. 4.1. Odpowiedź skokowa

5. Implementacja PID

Poniższy listing zawiera implementację dyskretnego regulatora PID w języku Matlab. W liniach 18, 19 i 20 obliczane są parametry r_0 , r_1 i r_2 , które następnie są wykorzystywane razem z uchybami z chwil k , $k-1$ i $k-2$ do obliczenia przyrostu sterowania ΔU w lini 47. W związku z istnieniem ograniczeń w postaci

$$-\Delta U^{\max} \leq \Delta U(k) \leq \Delta U^{\max} \quad (5.1)$$

gdzie $\Delta U^{\max} = 0,1$ oraz

$$0,6 \leq U(k) \leq 1,6 \quad (5.2)$$

należy przycinać sygnał wyjściowy regulatora. Dzieje się to w liniach 50-60. Aby ograniczyć liczbę instrukcji if najpierw ograniczany jest przyrost sygnału sterującego ΔU , a potem wartość sygnału sterującego U .

```
1 % Ograniczenia
2 du_max = 0.1;
3 u_min = 0.6;
4 u_max = 1.6;
5 % Punkt pracy
6 u_pp = 1.1;
7 y_pp = 2.5;
8 % Wskaznik jakosci - blad sredniokwadratowy
9 piderr = 0;
10 %Okres regulacji
11 T=1;
12 % Nastawy PID
13 params = [4.4817, 14.4945, 6.8265];
14 K = params(1);
15 Ti = params(2);
16 Td = params(3);
17
18 r0 = K * (1 + (T/(2*Ti)) + (Td/T));
19 r1 = K * ((T/(2*Ti)) - (2*Td/T) - 1);
20 r2 = K*Td/T;
21
22 % Czas symulacji
23 t_sim = 800;
24 % Zadana trajektoria
25 y_zad = ones(t_sim, 1) * 2.7;
26 y_zad(100:250) = 2.9;
27 y_zad(250:400) = 2.7;
28 y_zad(400:600) = 2.4;
29 % Inicjalizacja wektorow y, u, e
30 y = ones(t_sim, 1) * y_pp;
31 u = ones(t_sim, 1) * u_pp;
32 e = zeros(t_sim, 1);
```



```
33
34 % Petla, w ktorej odbywa sie symulacja
35 for k = 3:t_sim
36     % Symulacja obiektu
37     if k-11 <= 0
38         y(k) = symulacja_obiektu6Y(u_pp,u_pp,y(k-1),y(k-2));
39     else
40         y(k) = symulacja_obiektu6Y(u(k-10),u(k-11),y(k-1),y(k-2));
41     end
42     % Obliczenie uchybu
43     e(k) = y_zad(k) - y(k);
44     % Obliczenie wskaźnika jakości
45     piderr = piderr + e(k)^2;
46     % Obliczenie przyrostu sterowania
47     du = r2*e(k-2) + r1*e(k-1) + r0*e(k);
48
49     % Nałożenie ograniczeń
50     if du>du_max
51         du = du_max;
52     elseif du<-du_max
53         du = -du_max;
54     end
55     uk = u(k-1) + du;
56     if uk>u_max
57         uk = u_max;
58     elseif uk<u_min
59         uk = u_min;
60     end
61     u(k) = uk;
62 end
```

6. Implementacja DMC

Zaimplementowany regulator to DMC w wersji analitycznej „oszczędnej”, czyli w każdej chwili liczona jest tylko obecny przyrost sterowania, a nie cały wektor przewidywanych przyrostów. Poniższy fragment kodu zawiera inicjalizację potrzebnych parametrów D , N , N_u i λ oraz oblicza macierze, które są wyznaczone „offline”, czyli \mathbf{M} , \mathbf{M}_p i \mathbf{K} . Na ich podstawie obliczane są k_e i k_u . DMC również obowiązują ograniczenia, są zaimplementowane dokładnie tak samo jak w przypadku PID.

```
1  % Ograniczenia
2  du_max = 0.1;
3  u_min = 0.6;
4  u_max = 1.6;
5  % Punkt pracy
6  u_pp = 1.1;
7  y_pp = 2.5;
8  % Nastawy
9  D = 200;
10 N = 32;
11 Nu = 3;
12 lambda = 1;
13 % Macierz M
14 M = zeros(N,Nu);
15 for i = 1:size(M,1)
16     for j = 1:size(M,2)
17         if i>=j
18             M(i,j) = s(i-j+1);
19         end
20     end
21 end
22 % Macierz Mp
23 Mp = zeros(N,D-1);
24 for i = 1:size(Mp,1)
25     for j = 1:size(Mp,2)
26         if i+j<D
27             Mp(i,j) = s(i+j) - s(j);
28         else
29             Mp(i,j) = s(D) - s(j);
30         end
31     end
32 end
33 % Macierz K
34 K = ((M'*M + lambda*eye(Nu))^-1)*M';
35 ke = sum(K(1,:));
36 ku = zeros(D-1,1);
37 for i = 1:D-1
```

```

38     ku(i) = K(1,:) * Mp(:,i);
39 end

```

Poniższy fragment zawiera główną pętlę symulacji. W linii 20 obliczany jest przyrost sterowania, na który zostają nałożone ograniczenia. Zmienna `current_sum` obliczana w linii 16 i używana do obliczenia przyrostu sterowania to składnik sumy składającej się na przyrost sterowania

$$k_u \Delta U^P(k) \quad (6.1)$$

```

1  % Petla, w ktorej odbywa sie symulacja
2  for k = 3:t_sim4
3      % Symulacja obiektu
4      if k-11 <= 0
5          y(k) = symulacja_obiektu6Y(u_pp,u_pp,y(k-1),y(k-2));
6      else
7          y(k) = symulacja_obiektu6Y(u(k-10),u(k-11),y(k-1),y(k-2));
8      end
9      % Obliczenie uchybu
10     e(k) = y_zad(k) - y(k);
11     dmcerr = dmcerr + e(k)^2;
12     % Obliczenie sumy
13     current_sum = 0;
14     for i = 1:D-1
15         if k-i > 1
16             current_sum = current_sum + ku(i) * du(k-i);
17         end
18     end
19     % Obliczenie przyrostu sterowania
20     duk = ke*e(k) - current_sum;
21
22     % Nalozenie ograniczen
23     if duk>du_max
24         duk = du_max;
25     elseif duk<-du_max
26         duk = -du_max;
27     end
28     du(k) = duk;
29     uk = u(k-1) + duk;
30     if uk>u_max
31         uk = u_max;
32     elseif uk<u_min
33         uk = u_min;
34     end
35     u(k) = uk;
36 end

```

Działanie obu regulatorów zostanie zaprezentowane w kolejnych punktach.

7. zad5

8. zad6