

Wydział Elektroniki i Technik Informacyjnych
Politechnika Warszawska

Projektowanie układów sterowania
(projekt grupowy)

Sprawozdanie z projektu i ćwiczenia laboratoryjnego
nr 3, zadanie nr 2

Eva Reszka, Mateusz Roszkowski, Dominika Zając

Warszawa, 2021

Spis treści

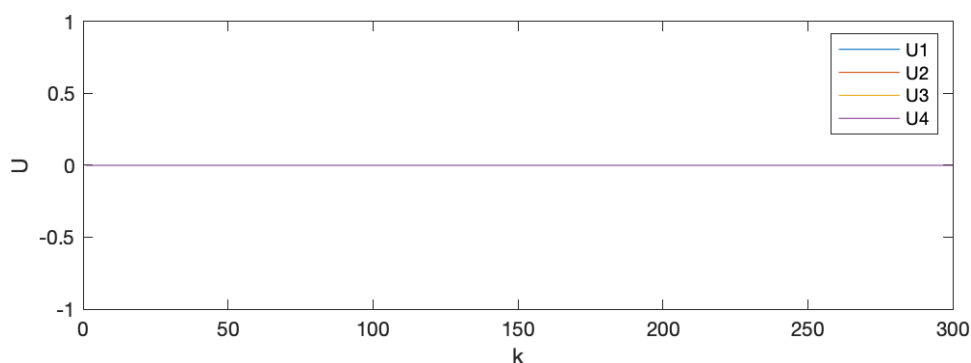
1. Projekt	2
1.1. Sprawdzenie poprawności punktu pracy	2
1.2. Wyznaczenie odpowiedzi skokowych procesu	2
1.3. Algorytmy PID i DMC	2
1.4. Rozmyty algorytm PID	3
1.5. Rozmyty algorytm DMC	3
2. Ćwiczenie laboratoryjne	5
2.1. Przygotowanie do wykonania ćwiczenia	5
2.2. Mechanizm zabezpieczający przed uszkodzeniem	5
2.3. Regulator dwupętlowy PID	6
2.4. Dobór parametrów PID	8
2.5. Dwuwymiarowy regulator DMC	9
2.6. Wyświetlanie danych na panelu operatora	9

1. Projekt

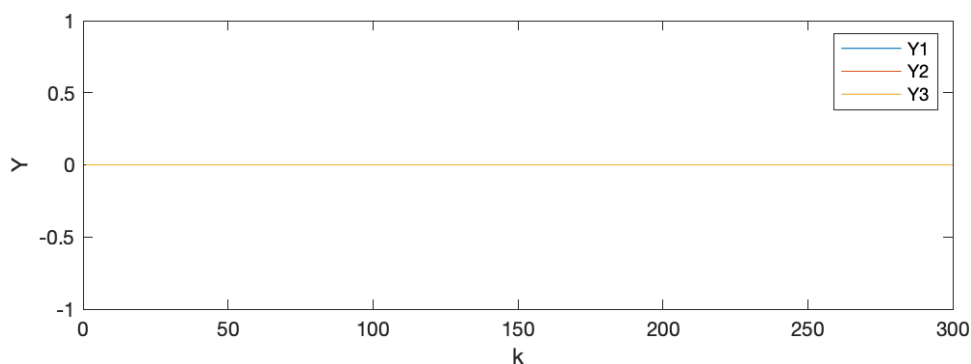
1.1. Sprawdzenie poprawności punktu pracy

Implementacja zadania znajduje się w pliku `zad1_2.m`.

Punkt pracy równy jest $U_{pp} = 0$, $Y_{pp} = 0$, co zostało przedstawione na wykresach 1.1 i 1.2.



Rys. 1.1. Wejście układu w punkcie pracy



Rys. 1.2. Wyjście układu w punkcie pracy

1.2. Wyznaczenie odpowiedzi skokowych procesu

Układ został pobudzony sygnałami o wartościach $U = [-0, 8; -0, 3; 0, 2; 0, 6; 1, 0]$.

Otrzymane zostały w ten sposób odpowiedzi skokowe:

Na wykresie ?? widoczna jest charakterystyka statyczna obiektu.

Właściwości dynamiczne oraz statyczne nie są liniowe. Do charakterystyki statycznej nie może zostać dopasowana prosta.

1.3. Algorytmy PID i DMC

Obiekt został poddany regulacji za pomocą algorytmów PID i DMC z Projektu 2.

Jakość sterowania zależy od wartości Y_{zad} . Różni się intensywność oscylacji oraz wartość przeregulowania. Wynika to z nieliniowości charakterystyki statycznej.

Wskaźnik jakości wyniosł $E = 388.88$.

Wskaźnik jakości wyniosł $E = 515.34$. Jego wartość jest większa, niż dla PID. Regulator DMC sprawdził się gorzej od pierwszego algorytmu. Wstępuje znacznie większe przeregulowanie oraz oscylacje. Dla $Y_{zad} = 4.5$ sygnał wyjściowy nie osiąga tej wartości.

1.4. Rozmyty algorytm PID

1.5. Rozmyty algorytm DMC

Następnie zaimplementowano algorytm rozmytego regulatora DMC dla różnej liczby regulatorów lokalnych (2, 3, 4 lub 5). Dla każdego regulatora z inną liczbą regulatorów lokalnych dobrano parametry oddzielnie.

Zestaw liczb s_1, s_2, s_3, \dots obliczany jest oddzielnie dla każdego z regulatorów lokalnych poprzez wykorzystanie funkcji `licz.s.m.`

```
function s=licz_s(u1, u2)
    y_temp = wynik_y(u1);
    Y=zeros(1,200);
    U=zeros(1,200);
    Y(1:15)=y_temp;
    U(1:10)=u1;
    U(11:200)=u2;

    for k=7:1:200
        Y(k)=symulacja_obiektu2y_p3(U(k-5),U(k-6),Y(k-1),Y(k-2));
    end
    for j=1:1:190
        s(j)=(Y(j+10)-y_temp)/(u2-u1);
    end
end
```

Przyjmuje ona jako parametry wartości $u1$ i $u2$. Parametr $u1$ przekazywany jest następnie funkcji `wynik_y`, która przeprowadza symulację dla wartości sterowania $u = u1$ i zwraca ostatnią wartość odpowiedzi skokową y_temp .

```
function y = wynik_y(u)
    U_sym(1:200)=u;
    Y_sym(1:200)=0;
    for k=7:200
        Y_sym(k)=symulacja_obiektu2y_p3(U_sym(k-5),U_sym(k-6),
        Y_sym(k-1),Y_sym(k-2));
    end
    y=Y_sym(200);
end
```

Wartość ta została następnie wykorzystana do przeprowadzenia kolejnej symulacji, dla skoku wartości sterowania z $u1$ do $u2$. Otrzymana odpowiedź skokowa została przeskalowana, w celu wyliczenia zestawu liczb s_1, s_2, s_3, \dots . Przekształcenie to przebiegło zgodnie ze wzorem 1.1, gdzie $Y_{pp} = y_temp$. Zamiast Y_i użyto Y_{i+10} , ponieważ zmiana wartości sterowania z $u1$ na $u2$ wystąpiła w chwili $k = 10$.

$$S_i = \frac{Y_{i+10} - Y_{pp}}{\Delta U}, i = 1, 2, 3, \dots \quad (1.1)$$

Po otrzymaniu przekształconych odpowiedzi skokowych dla każdego z regulatorów lokalnych przystąpiono do liczenia macierzy $M, M^P i K$. Wartości horyzontów dynamiki D zostały dobrane oddzielnie dla każdego z regulatorów. Dla ułatwienia przyjęto, że $N = N_u = D$. $\lambda = 1$ w przypadku każdego regulatora.

Współczynniki przynależności (wagi) każdego z regulatorów obliczone zostały w funkcji rozkład. Wykorzystana została funkcja przynależności dzwonowej `gbellmf`.

Regulator przetestowano dla kilku skoków wartości zadanej y_{zad} . Przebiegi dla różnej ilości regulatorów lokalnych, przy wartości parametru $\lambda = 1$ przedstawiono poniżej.

Dla dwóch regulatorów lokalnych wskaźnik jakości regulacji wyniósł $E = 227,3882$.

Dla trzech regulatorów lokalnych wskaźnik jakości regulacji wyniósł $E = 203,6199$.

Dla czterech regulatorów lokalnych wskaźnik jakości regulacji wyniósł $E = 844,2965$

2. Ćwiczenie laboratoryjne

Podczas tego zadania laboratoryjnego wykorzystano:

- grzałki G1 i G2 (sygnał sterujący U),
- wentylatory W1 i W2 (wartość zadana Y_{zad}),
- czujniki temperatury T1 i T3 (sygnał wyjściowy Y)

2.1. Przygotowanie do wykonania ćwiczenia

Przed rozpoczęciem pomiarów sprawdzono możliwość sterowania i pomiaru w komunikacji ze stanowiskiem. Punkty pracy grzałek G1 i G2 dla zespołu obliczone zostały wg. wzoru 2.1:

$$G1 = 25 + Z \quad G2 = 30 + Z \quad (2.1)$$

gdzie Z to numer zespołu %5, zatem dla grupy Z02 punkt pracy wynosi:

$$G1 = 25 + 2\%5 = 25 + 2 = 27 \quad G2 = 30 + 2\%5 = 30 + 2 = 32 \quad (2.2)$$

Następnie określono wartości pomiarów temperatur T1 i T3 dla obliczonych punktów pracy. W tym celu moc wentylatorów W1 i W2 ustawiono na 50% za pomocą rejestrów:

```
D110 := 500; // W1
D111 := 500; // W2
```

Moc wentylatorów, zarówno jak i grzałek przesyłana jest w postaci wartości 10 razy większych niż oczekiwane (np. 50% reprezentowane jest przez wartość 500). Odczytywana temperatura jest za to pomnożona razy 100 (np. 25°C reprezentowane jest przez wartość 2500)

Wartości mocy grzałki zadawane są poprzez rejestry:

```
D114 := 270; // G1
D115 := 320; // G2
```

Wartość G1 została ustawiona na 27%, a G2 na wartość 32%.

Temperatury T1 i T3 odczytano korzystając odpowiednio z rejestrów

```
T_1 := D100;
T_3 := D102;
```

Przy $G1 = 27$ i $G2 = 32$ temperatura T1 ustabilizowała się na wartości **20,37°C**, a temperatura T3 na wartości **21,00 °C**

2.2. Mechanizm zabezpieczający przed uszkodzeniem

Zaimplementowane zostało zabezpieczenie przed uszkodzeniem stanowiska w przypadku uszkodzenia czujnika temperatury. Sytuacja taka może doprowadzić do wzrostu temperatury stanowiska do niebezpiecznego stopnia, zatem, gdy czujnik wskaże temperaturę przekraczającą 250 °C, grzałka znajdująca się przy tym czujniku zostaje wyłączona. Implementacja zabezpieczenia znajduje się poniżej. Uwzględnia ona również ograniczenia sterowania grzałkami (minimalnie 0% i maksymalnie 100% mocy).

```
IF (T_1 > 25000.0) THEN
    U1_PID1 := 0;
    ELSIF(U1_PID1 > 1000) THEN
        U1_PID1 := 1000;
    ELSIF(U1_PID1 < 1) THEN
        U1_PID1 := 0;
END_IF;
```

```
F (T_3 > 25000.0) THEN
    U1_PID2 := 0;
    ELSIF(U1_PID2 > 1000) THEN
        U1_PID2 := 1000;
    ELSIF(U1_PID2 < 1) THEN
        U1_PID2 := 0;
END_IF;
```

2.3. Regulator dwupętłowy PID

Do realizacji zadania wykorzystano podstawowy plik do komunikacji ze stanowiskiem grzejąco - chłodzącym PUST_PLC.gx3.

Zaimplementowany został algorytm dwupętlowego regulatora PID.

```
//PID 1

Y_zad1 := 3000; // 25°C
T_1 := D100; // (x100)
G_1 := D114; // (x10)

E2_PID1 := E1_PID1;
E1_PID1 := E_PID1;

E_PID1 := (Y_zad1 - T_1)/100;

r0_PID1:= 15.9769734;
r1_PID1 := -14.58552;
r2_PID1 := 0.1875;

delta_U_PID1 := r2_PID1 * E2_PID1 + r1_PID1 * E1_PID1 + r0_PID1 * E_PID1;

U1_PID1 := delta_U_PID1 + U1_PID1;

// ograniczenia

F (T_1 > 25000.0) THEN
    U1_PID1 := 0;
    ELSIF(U1_PID1 > 1000) THEN
        U1_PID1 := 1000;
    ELSIF(U1_PID1 < 1) THEN
        U1_PID1 := 0;
END_IF;

// send U1

D114 := REAL_TO_INT(U1_PID1); // G1;

u_k := D114; // do wysyłania wykresów G1 w SocketComm

D110 := 500; // W1
D111 := 500; // W2
```

```
y_k := D100; // do wysylania wykresow T1 w SocketComm
y_k2 := D102; // do wysylania wykresow T3 w SocketComm

////////////////////////////////////

//PID 2
Y_zad2 := 3000; // 30*C

T_3 := D102; // (x100)
G_2 := D115; // (x10)

E2_PID2 := E1_PID2;
E1_PID2 := E_PID2;

E_PID2 := (Y_zad2 - T_3)/100;

// r0_PID2, r1_PID2, r2_PID2 mozemy obliczyc w MATLABIE

r0_PID2 := 15.9769734;
r1_PID2 := -14.58552;
r2_PID2 := 0.1875;

delta_U_PID2 := r2_PID2 * E2_PID2 + r1_PID2 * E1_PID2 + r0_PID2 * E_PID2;

U1_PID2 := delta_U_PID2 + U1_PID2;

// ograniczenia
IF (T_3 > 25000.0) THEN
    U1_PID2 := 0;
    ELSIF(U1_PID2 > 1000) THEN
        U1_PID2 := 1000;
    ELSIF(U1_PID2 < 1) THEN
        U1_PID2 := 0;
END_IF;

D115 := REAL_TO_INT(U1_PID2); // do wysylania wykresow G2 w SocketComm

u_k2 := D115;

D110 := 500; // W1
D111 := 500; // W2

D117 := REAL_TO_INT(Y_zad1);
D118 := REAL_TO_INT(Y_zad2);

y_k := D100; // do wysylania wykresow T1 w SocketComm
y_k2 := D102; // do wysylania wykresow T3 w SocketComm

// wizualizacja
D117 := REAL_TO_INT(Y_zad1);
D118 := REAL_TO_INT(Y_zad2);

y_k := D100;
y_k2 := D102;

// automat stanow

IF (T_1 < 3000) THEN
    D121 := 1;
    ELSIF (T_1 = 2500) THEN
        D121 := 2;
```



```
        ELSIF (T_1 > 2500) THEN
            D121 := 3;
END_IF;

IF (T_3 < 3000) THEN
    D122 := 1;
    ELSIF (T_3 = 2500) THEN
        D122 := 2;
    ELSIF (T_3 > 2500) THEN
        D122 := 3;
END_IF;
```

2.4. Dobór parametrów PID

Parametry regulatorów obliczane były w MatLabie, w pliku PID_wyliczanie_parametrow.m. W obu regulatorach stosowano te same nastawy.

Zespół zdecydował rozpoczęcie dobierania parametrów od przetestowania parametrów regulatora PID z bloku 3 ($K = 0,132199$; $T_i = 3,637091$; $T_d = 0,107170$). Wyniki przedstawione zostały na Rys.??.

Jak widać na Rys.??, nastawy te powodowały znaczące oscylacje, powodujące amplitudę zmian temperatury T1 o wartości około 9,18 °C, oraz 1,33 °C w przypadku temperatury T3. Należy zaznaczyć, że wartość zadana dla obydwu temperatur wynosiła wówczas 25 °C. Przy tych nastawach temperatura T3 nie osiągała nawet tej wartości, a w przypadku temperatury T1 występowało znaczące przesterowanie. Na wykresie widoczne są również chwilowe zaniki sterowania, reprezentowane jego spadkiem do wartości około 0. Nie wynikały one ze sposobu implementacji regulatora - zakładamy, że źródłem były chwilowe problemy komunikacyjne między sterownikiem a stanowiskiem grzejąco-chłodzącym.

Jako, że powyższy wynik był tak niezadowalający, a ograniczenie czasowe laboratoriów i długi czas przeprowadzania symulacji ograniczał zespołowi możliwość zrealizowania prawidłowego strojenia metodą inżynierską, zdecydowano wypróbować skrajnie inne nastawy o wartościach $K = 15$; $T_i = 38$; $T_d = 0,05$. Znacznie zwiększono wzmocnienie K , jak i człon całkujący T_i , aby umożliwić temperaturze T3 osiągnięcie wartości zadanej. Przebiegi dla tych nastaw przedstawione zostały na Rys.?? . Należy zaznaczyć, że poniższe testy przeprowadzane były innego dnia i wartość zadana dla obu temperatur została zmieniona na 30 °C, przy kolejnych testach powrócono jednak do wartości 25 °C.

Taki dobór nastaw spowodował, że obie temperatury zaczęły oscylować wokół wartości zadanej, jednak amplitudy ich zmian wzrosły.

Aby zmniejszyć oscylacje, zmniejszono wartości wzmocnienia K i członu całkującego T_i , przy jednoczesnym zwiększeniu wartości członu różniczkującego T_d , który miał za zadanie wypłaszczyć oscylacje. Następnie przetestowane nastawy to zatem: $K = 13,8$; $T_i = 35$; $T_d = 8,75$. Przebiegi przedstawiono na Rys. ?? .

Oscylacje znacząco się zmniejszyły - dla obu temperatur wynosiły około 4 °C. Były to najlepsze wyniki, jakie udało się osiągnąć zespołowi w czasie trwania laboratorium.

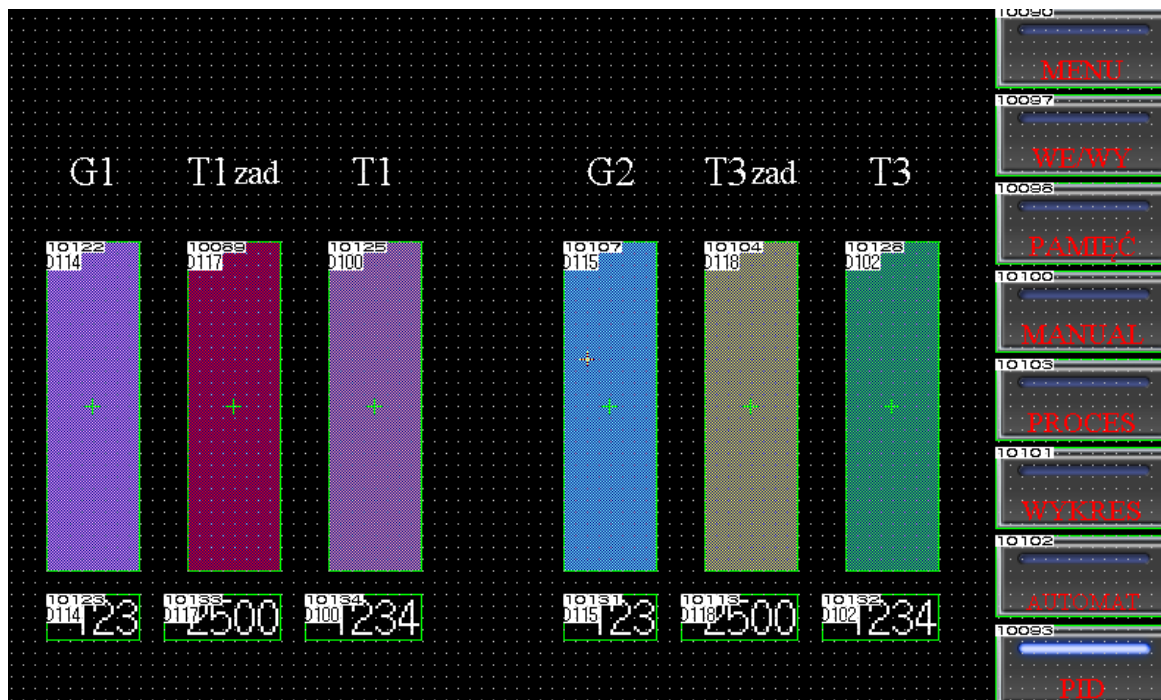
Jeśli zespół miałby więcej czasu na dostrajanie regulatora, najprawdopodobniej jeszcze bardziej zostałaby zmniejszona wartość wzmocnienia K i członu całkującego T_i , które powodowały widoczne na wykresach przesterowanie i oscylacje. Zwiększona zostałaby za to wartość członu różniczkującego T_d , który widocznie przyczynił się do spadku poziomu oscylacji.

Najbardziej optymalnym podejściem byłoby zastosowanie metody inżynierskiej - znalezienie wzmocnienia krytycznego i obliczanie parametrów na jego podstawie, a następne ich dostrajanie, jednak ograniczenia czasowe wobec ilości zadań do wykonania nie pozwoliłyby zespołowi na przeprowadzenie jej w całości.

2.5. Dwuwymiarowy regulator DMC

2.6. Wyświetlanie danych na panelu operatora

Stworzony został interfejs graficzny na podstawie przykładowego pliku `sample.GOT.GTX`. Implementacja interfejsu, jak i automatu stanów opisanego w następnej sekcji znajduje się w pliku `wizualizacja.GTX`. Interfejs zaprojektowany został w programie GT Designer3 (Rys. 2.1). W projekcie uwzględniono wartości mierzone, zadane oraz sterowania. W celu zaimplementowania logiki wyświetlania się wartości na panelu stworzono dwa nowe rejestry (D117 i D118) przetrzymujące wartości zadane. Następnie podpięto odpowiednie rejestry do odpowiednich słupków w wizualizacji ($G1$ - D114, $G2$ - D115, $T1_{zad}$ - D117, $T3_{zad}$ - D118, oraz $T1$ - D100 i $T3$ - D102).



Rys. 2.1. Projekt wizualizacji w programie GT Designer3

```
// wizualizacja
D117 := REAL_TO_INT(Y_zad1);
D118 := REAL_TO_INT(Y_zad2);

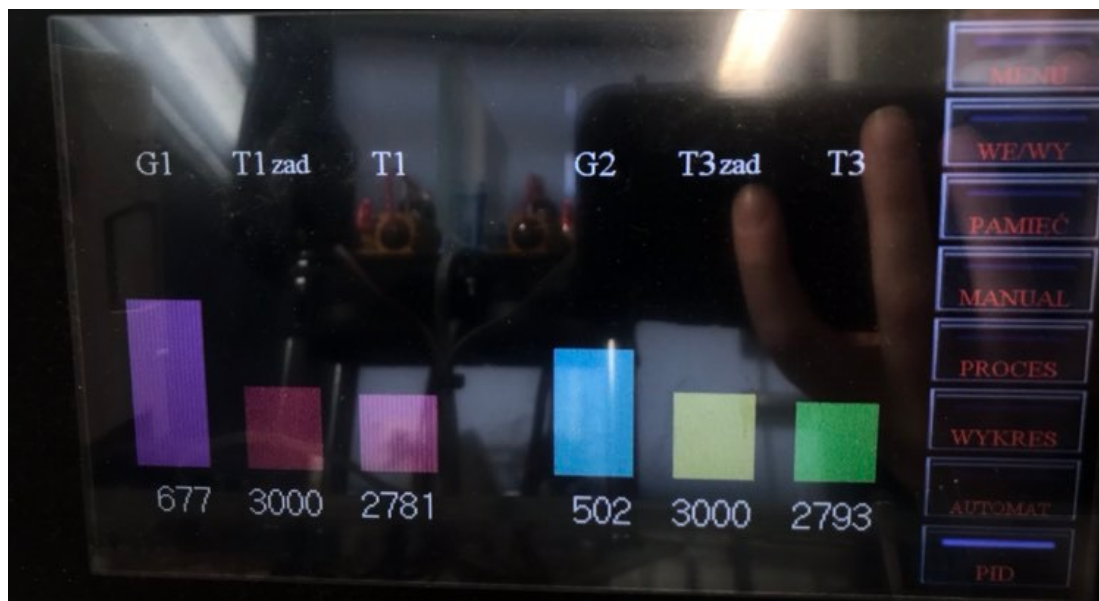
y_k := D100;
y_k2 := D102;

// automat stanow

IF (T_1 < 3000) THEN
D121 := 1;
ELSIF (T_1 = 2500) THEN
D121 := 2;
ELSIF (T_1 > 2500) THEN
D121 := 3;
END_IF;

IF (T_3 < 3000) THEN
D122 := 1;
ELSIF (T_3 = 2500) THEN
```

```
D122 := 2;  
ELSIF (T_3 > 2500) THEN  
D122 := 3;  
END_IF;
```



Rys. 2.2. Wizualizacja na panelu operatora