

**Wydział Elektroniki i Technik Informacyjnych
Politechnika Warszawska**

**Projektowanie układów sterowania
(projekt grupowy)**

Sprawozdanie z laboratorium 5

Kamil Gabryjelski, Paweł Rybak, Paweł Walczak

Warszawa, 2017

Spis treści

1. Opis problemu	2
1.1. Stanowisko grzejące	2
1.2. Obiekt Inteco	2
2. Zadanie 1	3
2.1. Sprawdzenie połączenia	3
2.2. Punkt pracy	3
3. Zadanie 2	5
4. Zadanie 3	6
5. Zadanie 4	9
6. Zadanie 5	13
7. Zadanie 6	14

1. Opis problemu

1.1. Stanowisko grzejące

Pierwszym z celów piątego laboratorium było zbadanie obiektu, a następnie implementacja, weryfikacja poprawności działania i dobór parametrów algorytmów regulacji obiektu na sterowniku firmy Mitsubishi, przy pomocy oprogramowania GX Works. Badania dotyczyły obiektu wielowymiarowego, o dwóch wejściach i dwóch wyjściach. Obiekt składał się z:

- sygnał wejściowy pierwszy – grzałka G1
- sygnał wejściowy drugi – grzałka G2
- sygnał wyjściowy pierwszy – czujnik temperatury T1
- sygnał wyjściowy drugi – czujnik temperatury T2
- zakłócenie pierwsze – wentylator W1
- zakłócenie drugie – wentylator W2.

Sygnały wejściowe G1 i G2 mogą przyjmować wartości w zakresie 0 - 100. Sygnały wyjściowe, zwracają pomiary temperatury w °C. Wentylatory W1 oraz W2 należy traktować jako cechę otoczenia. Mają one stałe wysterowanie na 50% swojej mocy. Ich użycie pozwala na bardziej dynamiczne schładzanie obiektu. Czas próbkowania obiektu wynosi 4s.

1.2. Obiekt Inteco

Drugim celem laboratorium było stworzenie regulatora dla obiektu firmy Inteco. W naszym przypadku obiekt składał się z trzech zbiorników umieszczonych, jeden nad drugim, wypełnianych wodą. Każda kolejna para zbiorników była połączona ze sobą zaworami, przez które woda mogła przepływać z jednego zbiornika do drugiego. Zawór z najniższego zbiornika był połączony ze zbiornikiem, będącym buforem nadmiarowej wody. Z owego bufora woda mogła być pompowana do najwyższego zbiornika. Z punktu widzenia zadania, które mieliśmy wykonać obiekt prezentował się następująco:

- pompa P1 – stale wysterowane wejście wody do zbiornika pierwszego
- zawór V1 – ujście wody z pierwszego zbiornika i jednocześnie wejście wody do zbiornika drugiego (pierwszy sygnał sterujący)
- zawór V2 – ujście wody z drugiego zbiornika i jednocześnie wejście wody do zbiornika trzeciego (drugi sygnał sterujący)
- zawór V3 – ujście wody z trzeciego zbiornika (trzeci sygnał sterujący)
- pomiar M1 – stan wody w pierwszym zbiorniku (pierwszy sygnał wyjściowy)
- pomiar M2 – stan wody w drugim zbiorniku (drugi sygnał wyjściowy)
- pomiar M3 – stan wody w trzecim zbiorniku (trzeci sygnał wyjściowy)

Wszelkie sygnały wejściowe sterowane były za pomocą PWM. Czas próbkowania obiektu wynosił 100ms.

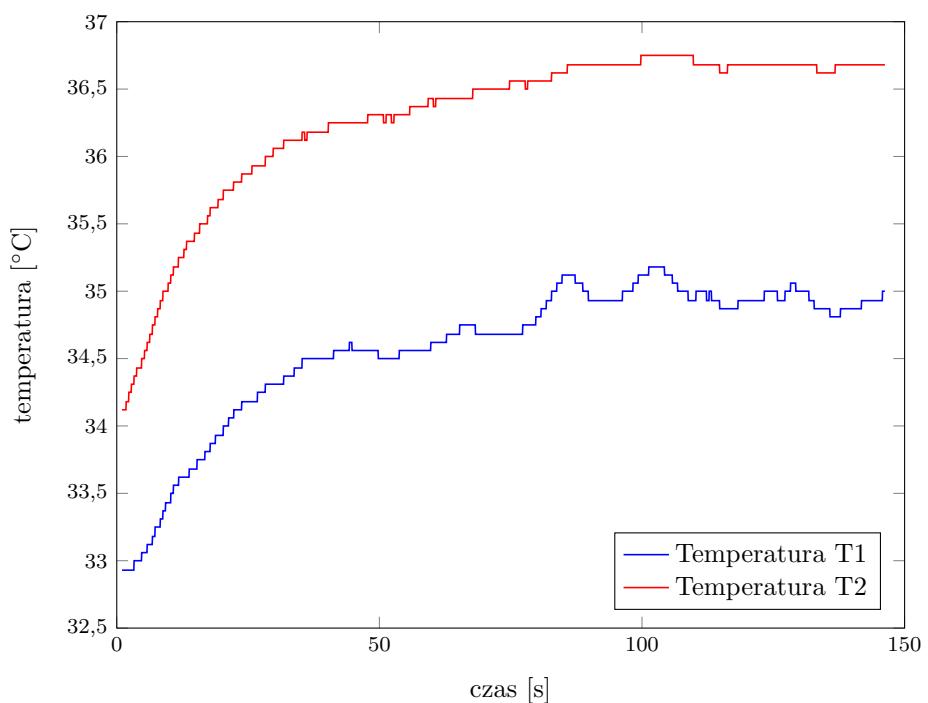
2. Zadanie 1

2.1. Sprawdzenie połączenia

Zadanie pierwsze polegało na sprawdzeniu możliwości komunikacji ze stanowiskiem. Sprawdziliśmy to, modyfikując wybrane wejścia obiektu, oraz obserwując zmiany na obiekcie. Przy modyfikacji wejścia wiatraków widać i słyszać było, iż wiatraki kręcą się wolniej lub szybciej, w zależności od wejścia. Przy sprawdzeniu działania grzałki polegaliśmy na diodach elektroluminescencyjnych, które świeciły mocniej lub słabiej w zależności od mocy grzania odpowiedniej grzałki. Nasza pewność w tej sprawie jest oparta na zaufaniu do konstruktora obiektu. Możliwość pomiaru wyjść obiektu pozostała sprawdzona w oparciu o trendy pomiaru, w zależności od wystęrowania wcześniej wspomnianych wejść. Otóż, przy zwiększeniu mocy grzałki lub zmniejszeniu mocy wiatraka temperatura na czujniku bliższym danej grzałce i wiatrakowi rośnie szybko, a na dalszym rośnie wolniej. Przy zwiększaniu mocy wiatraka lub zmniejszaniu mocy grzałki pomiar temperatur spadał. Podobnie jak wcześniej działało to z szybkością odwrotnie proporcjonalną do odległości czujnika od wiatraka i grzałki których sterowanie jest zmieniane.

2.2. Punkt pracy

Punkt pracy naszego stanowiska wynosił $G1 = 27$, $G2 = 32$, $W1 = 50$, $W2 = 50$. Przy tak ustawionych wejściach pozwoliliśmy obiektyowi się ustabilizować. Z powodów różnych zakłóceń pomiary temperatury nieustannie się wały. Oszacowaliśmy arbitralnie iż, pomiary ustabilizowały się na wartościach $T1 = 35$, $T2 = 36.68$. Pomiary wyjść przy takich ustawieniach sygnałów wejściowych przedstawia wykres 2.1.



Rys. 2.1. Punkt pracy obiektu.

3. Zadanie 2

W tym zadaniu należało stworzyć mechanizm zabezpieczający stanowisko przed przegrzaniem. Miał on polegać na tym, iż wyłączał on grzałkę przy czujniku, jeżeli pomiar na tym ostatnim przekroczył 150°C . Osiągnęliśmy ten rezultat, poprzez sprawdzenie temperatury na każdym z czujników z osobna oraz, jeżeli któryś z nich spełniał wcześniej wymieniony warunek, wyzerowanie wartości odpowiedniego rejestru, z którego sterowanie jest wysyłane do obiektu. Przy okzji była podnoszona flaga, oznajmiająca, iż jesteśmy w stanie awaryjnym. Służyła ona do tego, aby po powrocie temperatury do stanu akceptowalnego wrócić do punktu pracy. W późniejszym czasie, gdy obiekt będzie regulowany owa sekwencja powrotu jest zbędna, gdyż sterowanie będzie wyznaczał regulator. Wspomniany kod znajduje się w listingu 3.1. Ów fragment kodu miał w zamyśle znaleźć się zaraz za kodem regulatora, tak aby rejestrty były ustawione na odpowiednie wartości zanim zostaną wysłane do obiektu. Zamysł ten był oparty na założeniu, iż każdy z plików kodu w sterowniku wykonuje się sekwencyjnie w całości, bez wykonywania kodu wysyłania do sterownika w połowie. Gdyby okazało się, iż tak nie jest należałoby zapisywać wartość sterowania, nie bezpośrednio do rejestrów, ale do zmiennych tymczasowych, w których byłoby sterowanie wyliczone, i w razie wystąpienia sytuacji awaryjnej zerowane, a dopiero po tej sekwencji zapisywane w odpowiednim rejestrze.

Listing 3.1. Kod mechanizmu zabezpieczającego przed przegrzaniem

```
IF (D100 > 15000) THEN
    ERROR_FLAG1 := TRUE;
    D114 := 0;
END_IF;

IF (D101 > 15000) THEN
    ERROR_FLAG2 := TRUE;
    D115 := 0;
END_IF;

IF (ERROR_FLAG1 AND (D100 < 15000)) THEN
    D114 := 270;
    ERROR_FLAG1 := FALSE;
END_IF;

IF (ERROR_FLAG2 AND (D101 < 15000)) THEN
    D115 := 320;
    ERROR_FLAG2 := FALSE;
END_IF;
```

4. Zadanie 3

Celem zadania trzeciego było zaprojektowanie regulatora PID. Kod zaprojektowanego przez nas regulatora znajduje się w listingu 4.1. Przy odczycie aktualnych wartości rejestrów są normalizowane do zakresów wartości opisanych we wstępnie. Przy wysyłaniu wartości są wymnażane razy 10, aby zakres wartości był zgodny z zakresem używanym przez obiekt. Biorąc pod uwagę fakt, iż obiekt jest symetryczny, postanowiliśmy użyć dwóch regulatorów o takich samych nastawach. Wyszliśmy od nastaw używanych do tego obiektu w projekcie trzecim, gdzie zadanie było bardzo podobne, czyli:

$$K = 5 \quad T_i = 30 \quad T_d = 2,5 \quad (4.1)$$

Okazało się jednak, iż te nastawy nie zdają egzaminu. Uważamy, iż w dużej mierze wpływ na to ma okres próbkowania obiektu, który wydłużył się z 1 sekundy w projekcie trzecim do 4 sekund w aktualnym projekcie. Działanie regulatorów o takich nastawach przedstawia wykres 4.1. Widząc słabe działanie regulatora postanowiliśmy nie prowadzić dalej tego eksperymentu. Następnie testowane były inne nastawy. Aby zdążyć z wykonaniem laboratorium w czasie kolejne nastawy były oceniane dosyć szybko i w oparciu o znajomość chwilowego stanu obiektu, dlatego aby nie zmylić czytelnika wykresy przejściowe nie zostały zamieszczone. Finalne nastawy są następujące:

$$K = 4 \quad T_i = 45 \quad T_d = 0 \quad (4.2)$$

Ich działanie przedstawia wykres 4.2. Początkowo obiekt nie był w stanie równowagi, stał zakłócenie na początku. Widać na wykresie jednak, iż udało mu się dosyć szybko ustabilizować na wartości zadanej oraz wykonać skok owej wartości.

Listing 4.1. Kod regulatora PID

```
//Regulator PID na podstawie równania różnicowego
SV_PID3 := Zadana_PID3;
PV_PID3 := D100 / 100;

//Nastawy regulatorow
K_PID3 := 4;
Ti_PID3 := 45;
Td_PID3 := 0;
//Wyliczenie parametrow
R0_PID3 := K_PID3*( 1+(4/(2*Ti_PID3))+Td_PID3/4 );
R1_PID3 := K_PID3*( (4/(2*Ti_PID3))-(2*Td_PID3/4)-1 );
R2_PID3 := K_PID3*Td_PID3/4;
//Wyliczenie uchybu regulacji i przesunięcie historii
E2_PID3 := E1_PID3;
E1_PID3 := E0_PID3;
E0_PID3 := SV_PID3 - PV_PID3;
//Obliczenie sterowania
U_PID3 := R2_PID3*E2_PID3 + R1_PID3*E1_PID3 + R0_PID3*E0_PID3 + U_PID3;

//Ograniczenia
IF (U_PID3 > 100.0) THEN
    U_PID3 := 100.0;
ELSIF (U_PID3 < 0.0) THEN
    U_PID3 := 0.0;
ENDIF;
```

```

D114 := REAL_TO_INT(U_PID3 * 10);

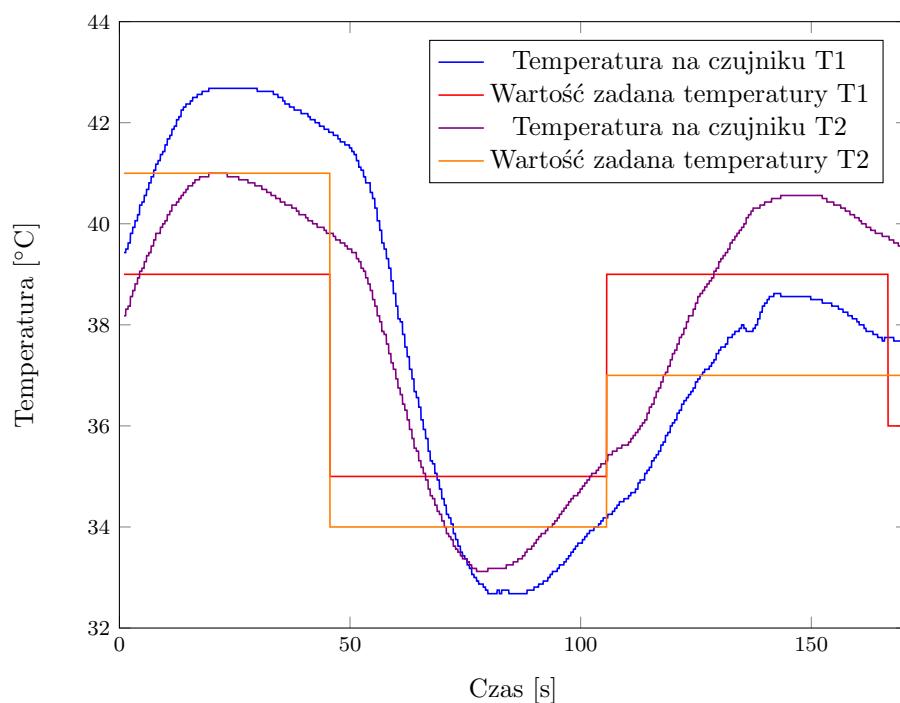
// PID NUMMER ZWEI, SEHR STARK
SV_PID2 := Zadana_PID2;
PV_PID2 := D101 / 100;

// Wyliczenie parametrow
R0_PID3 := K_PID3*( 1+(4/(2*Ti_PID3))+Td_PID3/4 );
R1_PID3 := K_PID3*( (4/(2*Ti_PID3))-(2*Td_PID3/4)-1 );
R2_PID3 := K_PID3*Td_PID3/4;
// Wyliczenie uchybu regulacji i przesuniecie historii
E2_PID2 := E1_PID2;
E1_PID2 := E0_PID2;
E0_PID2 := SV_PID2 - PV_PID2;
// Obliczenie sterowania
U_PID2 := R2_PID3*E2_PID2 + R1_PID3*E1_PID2 + R0_PID3*E0_PID2 + U_PID2;

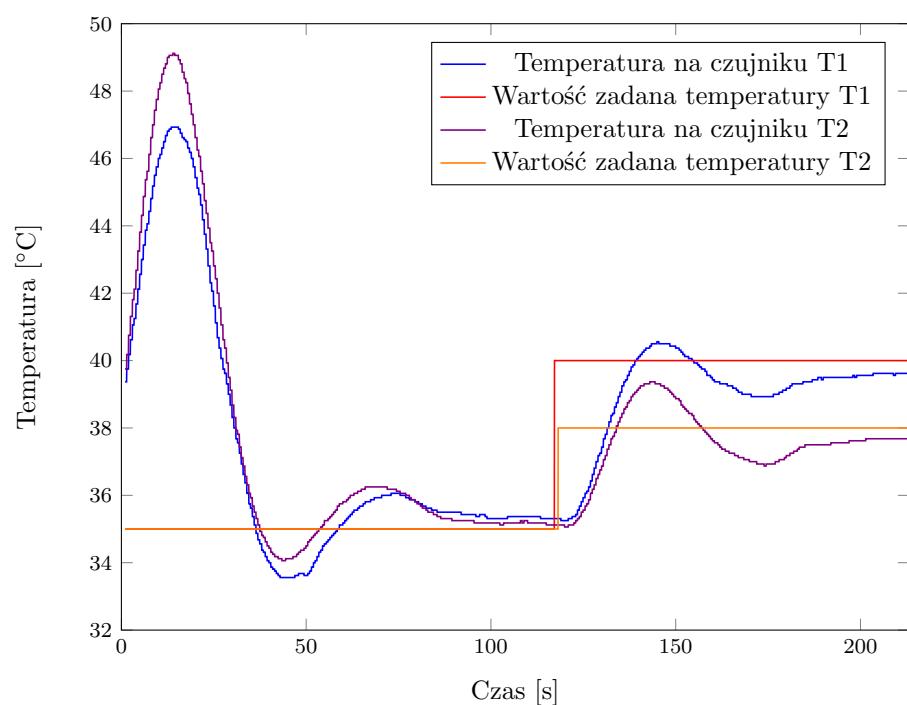
// Ograniczenia
IF (U_PID2 > 100.0) THEN
    U_PID2 := 100.0;
ELSIF (U_PID2 < 0.0) THEN
    U_PID2 := 0.0;
ENDIF;

D115 := REAL_TO_INT(U_PID2 * 10);

```



Rys. 4.1. Regulacja regulatorem PID, nastawy pierwsze.



Rys. 4.2. Regulacja regulatorem PID, nastawy drugie.

5. Zadanie 4

Kolejne zadanie polegało na przygotowaniu regulatora DMC dla stanowiska grzejącego. Aby to osiągnąć należało zebrać odpowiedzi skokowe czterech torów. Zmierzony został wpływ skoku każdej z dwóch grzałek, na każdy z dwóch pomiarów, a następnie wykresy zaproksymowano obiektem dwuinercyjnym z opóźnieniem. Wyniki zostały zaprezentowane na wykresach 5.1, 5.2, 5.3, oraz 5.4. Zebrane odpowiedzi zdają się nie być najwyższej jakości, jednak pod presją czasu postanowiliśmy zostawić je takimi jakie są. Patrząc na odpowiedzi skokowe widać, iż horyzont dynamiki obiektu wynosi 49 próbek. Horyzont predykcji, oraz horyzont sterowania przyjęliśmy taki sam jak horyzont dynamiki. Natomiast jako wartość wyjściową parametru Λ przyjęliśmy 1. Na podstawie tych próbek wyznaczone zostały cztery parametry, używane w algorytmie - wektory k_1^u i k_2^u , oraz wartości k_1^e i k_2^e . Są one definiowane jako:

$$k_1^e = \sum_{j=1}^N k_{1,j}, \quad k_2^e = \sum_{j=1}^N k_{2,j}, \quad k_1^u = \bar{K}_1 M^P, \quad k_2^u = \bar{K}_2 M^P \quad (5.1)$$

gdzie $k_{1,j}$ jest j -tym elementem pierwszego rzędu macierzy \mathbf{K} , $k_{2,j}$ j -tym elementem drugiego rzędu macierzy \mathbf{K} , \bar{K}_1 jest pierwszym rzędem macierzy \mathbf{K} , a \bar{K}_2 drugim rzędem owej macierzy. Sam algorytm wykorzystujący owe parametry został zaprezentowany w listingu 5.1. Działanie tak dobranego algorytmu przedstawia wykres 5.5. Uważamy, iż działanie tego algorytmu zdecydowanie mógłby polepszyć mniejszy współczynnik kary Λ , gdyż przy tak niskim czasie próbowania skoki sterowania powinny być wyższe, jednak nie udało nam się przeprowadzić testu.

Listing 5.1. Kod regulatora DMC

```
//Wejscia oraz wartosci zadane
SV_DMC3 := Zadana_DMC3;
PV_DMC3 := D100 / 100;
SV_DMC2 := Zadana_DMC2;
PV_DMC2 := D101 / 100;

//Uchyby
ek1 := SV_DMC3 - PV_DMC3;
ek2 := SV_DMC2 - PV_DMC2;

//Wyliczenie zmiany sterowania
dU1 := ke1 * ek1;
dU2 := ke2 * ek2;
FOR cnt:= 0 TO 95 BY 1 DO
    dU1 := dU1 - ku1[cnt] * dup[cnt];
    dU2 := dU2 - ku2[cnt] * dup[cnt];
END_FOR;

//Przesuniecie wektora poprzednich zmian sterowania
FOR cnt := 93 TO 0 BY -1 DO
    dup[cnt + 2] := dup[cnt];
END_FOR;
dup[0] := dU1;
dup[1] := dU2;

//Wyliczenie nowej wartosci sterowania
```

```

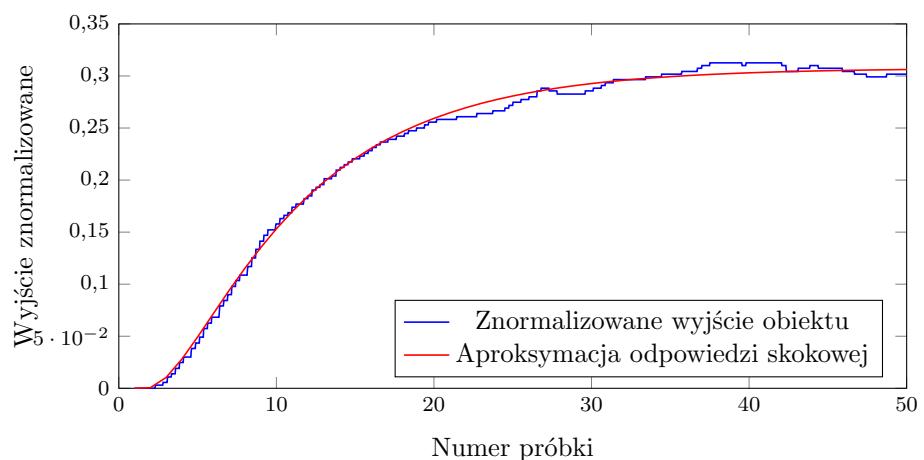
U_DMC2 := U_DMC2 + dU1;
U_DMC3 := U_DMC3 + dU2;

//Ograniczenia
IF (U_DMC2 > 100.0) THEN
    U_DMC2 := 100.0;
ELSIF (U_DMC2 < 0.0) THEN
    U_DMC2 := 0.0;
END_IF;

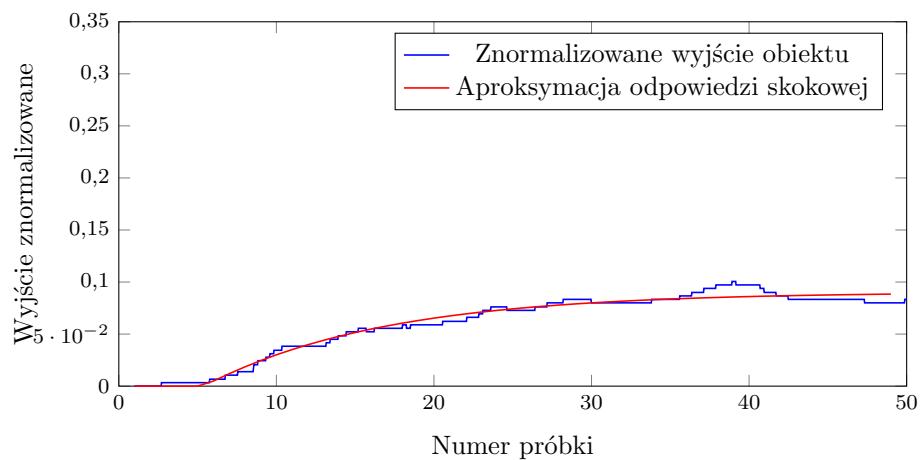
IF (U_DMC3 > 100.0) THEN
    U_DMC3 := 100.0;
ELSIF (U_DMC3 < 0.0) THEN
    U_DMC3 := 0.0;
END_IF;

//Wyslanie obliczonych wartosci do obiektu
D114 := REAL_TO_INT(U_DMC2 * 10);
D115 := REAL_TO_INT(U_DMC3 * 10);

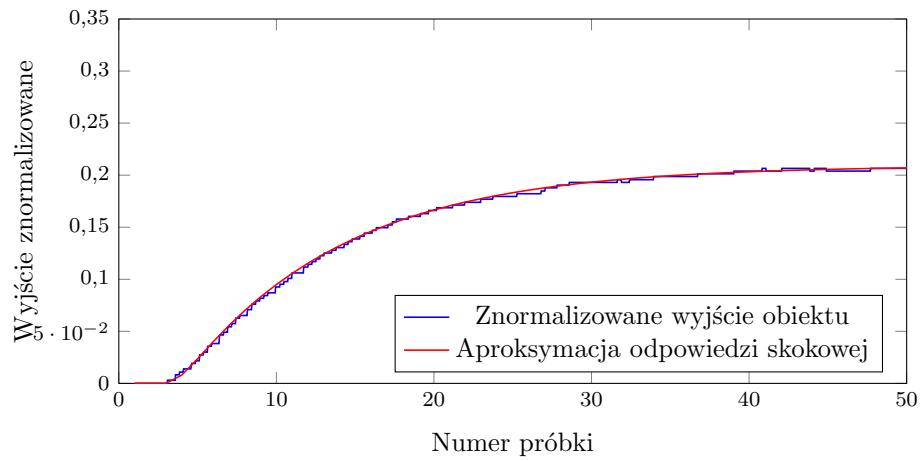
```



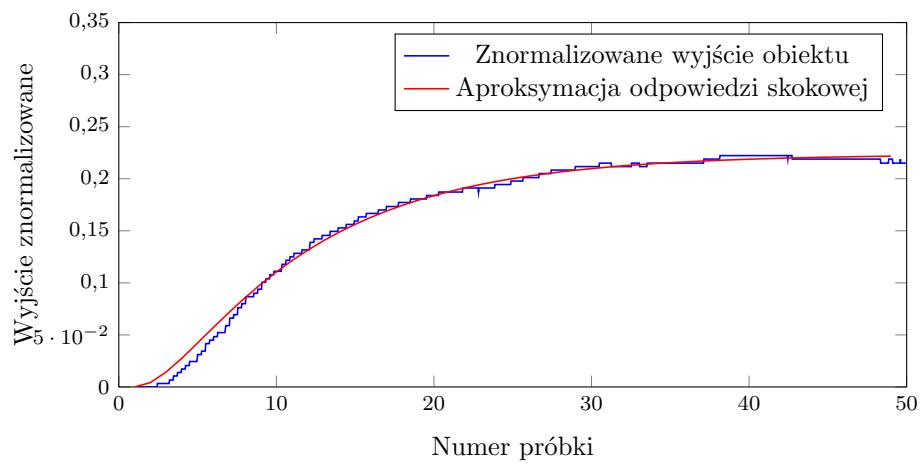
Rys. 5.1. Odpowiedź skokowa, pomiaru T1, na skok sygnału G1.



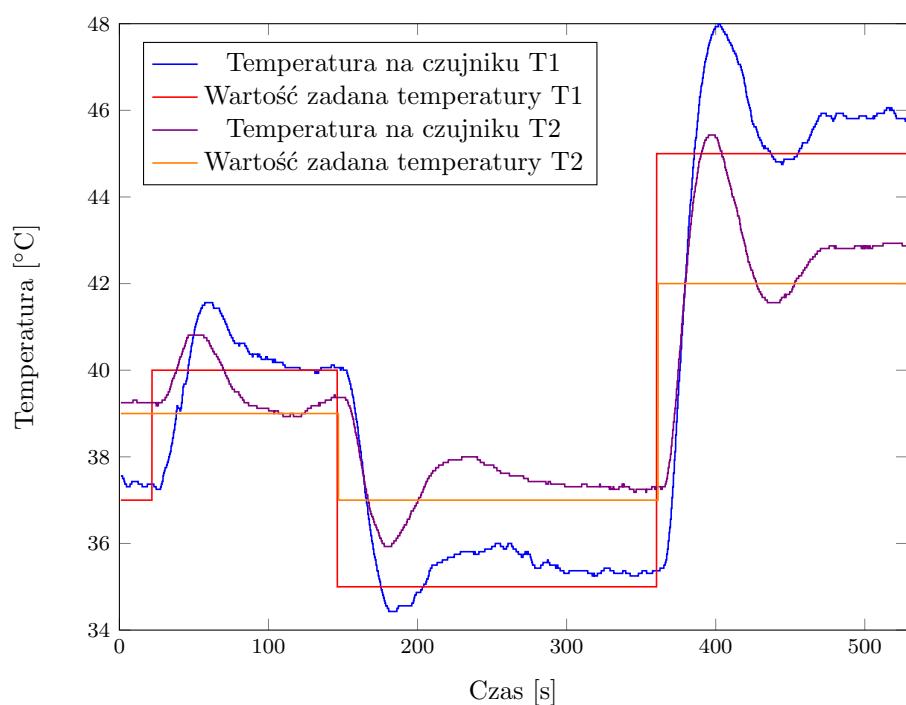
Rys. 5.2. Odpowiedź skokowa, pomiaru T1, na skok sygnału G2.



Rys. 5.3. Odpowiedź skokowa, pomiaru T2, na skok sygnału G1.



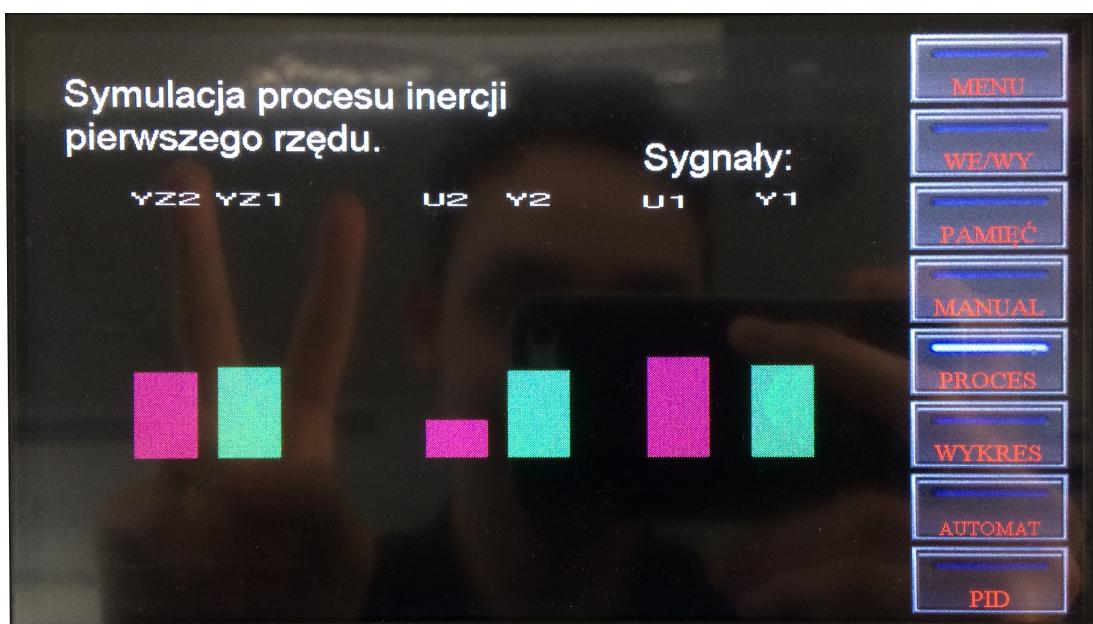
Rys. 5.4. Odpowiedź skokowa, pomiaru T2, na skok sygnału G2.



Rys. 5.5. Regulacja regulatorem DMC.

6. Zadanie 5

Kolejnym krokiem było stworzenie prostego graficznego panelu operatora, przy pomocy oprogramowania **GT Designer**. Składa się on z tytułu, oraz sześciu słupków, gdzie każdy reprezentuje poziom innego sygnału. Każdy ze słupków jest podpisany, tak, aby było wiadomo jaki sygnał reprezentuje. Wygląd panelu przedstawia rysunek 6.1. Ułożenie owych słupków nie było szczególnie planowane i należałoby je jakoś rozsądnie posortować, na przykład zestawiając trzy parametry każdego sygnału obok siebie. Należałoby również podać wartości liczbowe sygnału pod każdym ze słupków. Biorąc pod uwagę, iż jest to jedynie wersja poglądowa, nie przywiązywaliśmy do tego zbyt dużej wagi, jednak jesteśmy świadomi usprawnień jakie należy dokonać, gdyby taki system faktycznie miał być użytkowany.



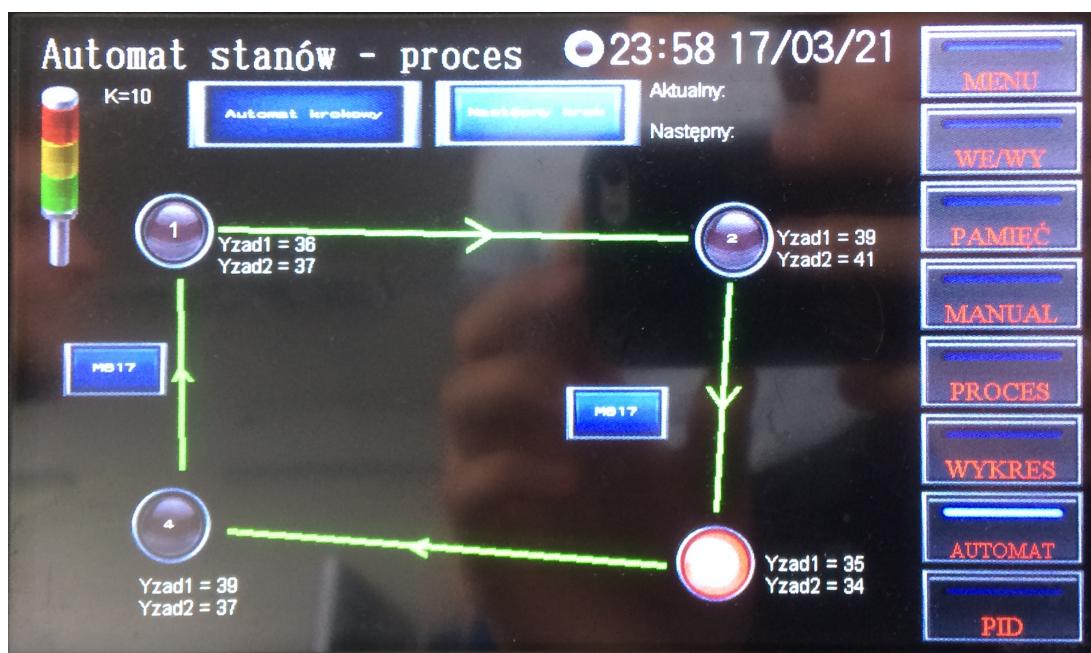
Rys. 6.1. Panel operatora

7. Zadanie 6

Dalej zaprojektowany został automat stanów dla obiektu. Jego zadanie było proste — co określony okres czasu zmieniał wartości zadane obiektu. Jego implementacja polegała na dodaniu zmiennej, w której trzymany był licznik, zwiększany z każdym wywołaniem programu. Jeżeli licznik zawierał się w zadanym przedziale wartości ustawiane były wartości zadane przypisane temu przedziałowi. Gdy licznik wyszedł poza przewidziane przedziały, był zerowany. Przy okazji w każdym z przedziałów ustawiana była inna zmienna, w której zapisany był numer aktualnego stanu. Służyło to jedynie do wyświetlania na panelu operatorskim aktualnego stanu. Czas przewidziany na jeden stan był łatwy do wyznaczenia. Mając świadomość, iż program wywoływany jest raz na 4 sekundy, długość przedziału pomnożona razy cztery była długością przebywania w danym stanie w sekundach. Kod tego programu jest zawarty w listingu 7.1. Panel operatorski, przedstawiający automat stanów przedstawia rysunek 7.1.

Listing 7.1. Kod automatu stanów

```
cntr := cntr+ 1;
IF (cntr < 60) THEN
    stan := 1;
    Zadana_DMC2 := 36;
    Zadana_DMC3 := 37;
ELSIF (cntr< 120) THEN
    stan := 2;
    Zadana_DMC2 := 39;
    Zadana_DMC3 := 41;
ELSIF (cntr< 180) THEN
    stan := 3;
    Zadana_DMC2 := 35;
    Zadana_DMC3 := 34;
ELSIF (cntr< 240) THEN
    stan := 4;
    Zadana_DMC2 := 39;
    Zadana_DMC3 := 37;
ELSE
    cntr := 0;
ENDIF;
```



Rys. 7.1. Automat stanów