

Wydział Elektroniki i Technik Informacyjnych
Politechnika Warszawska

Projektowanie układów sterowania
(projekt grupowy)

Sprawozdanie z projektu i ćwiczenia laboratoryjnego nr
1, zadanie nr 7

Stanisław Borkowski, Piotr Czajkowski, Maciej Grudziąż

Warszawa, 2018

Spis treści

1. Wstęp	2
2. Podstawy teoretyczne	3
2.1. Charakterystyka skokowa	3
2.2. Algorytm PID	3
2.3. Algorytm DMC	3
2.4. Ograniczenia	4
3. Obiekt symulowany - część projektowa	5
3.1. Wyznaczenie odpowiedzi skokowych procesu	5
3.2. Programy do symulacji algorytmów	5
3.3. Ręczny dobór odpowiednich wartości parametrów	7
3.4. Optymalizacja wskaźnika jakości	7
4. Sterowanie rzeczywistym obiektem - część laboratoryjna	9
4.1. Odpowiedzi skokowe	9
4.2. Programy do symulacji algorytmów	11
4.3. Regulator PID	13
4.4. Algorytm DMC	15
5. Wnioski	17

1. Wstęp

Celem projektu była implementacja, weryfikacja poprawności działania i dobór parametrów algorytmów regulacji jednowymiarowego procesu laboratoryjnego. W pierwszej jego części zajęliśmy się funkcją symulującą działanie procesu zrealizowaną w środowisku MATLAB, natomiast później zajmowaliśmy się rzeczywistym obiektem (stanowiskiem grzejąco-chłodzącym).

W obu przypadkach naszym zadaniem była analiza zachowań obiektów - badaliśmy m.in. takie parametry jak ich odpowiedzi skokowe i wartości sygnałów w punkcie pracy. Następnie zajęliśmy się implementacją oraz optymalizacją algorytmów regulacji.

2. Podstawy teoretyczne

2.1. Charakterystyka skokowa

Podstawowym opisem działania każdego układu regulacji są trzy charakterystyki: *skokowa*, *impulsowa* i *częstotliwościowa*. W przypadku tego zadania badaliśmy jedynie charakterystykę skokową. Definiuje się ją jako odpowiedź układu na wymuszenie w postaci skoku jednostkowego przy zerowych warunkach początkowych. Innymi słowy, jest to przebieg wyjścia $y(t)$ układu przy sterowaniu:

$$u(t) = \begin{cases} 0 & \text{gdy } t < 0 \\ 1 & \text{gdy } t \geq 0 \end{cases}$$

Metoda umożliwia proste wyznaczenie charakterystyki statycznej $y(u)$ dla danej wartości sterowania u .

2.2. Algorytm PID

Algorytm PID (w tym przypadku dyskretny) jest jednym z prostszych i najbardziej znanych sposobów na wyznaczenie wartości sterowania w automatycznym układzie regulacji. Taki regulator oblicza wartość uchybu $e(k)$ jako różnicę pomiędzy wartością zadaną $y^{\text{zad}}(k)$, a rzeczywistą zmienną procesu $y(k)$ i dąży do tego, aby ten uchyb zminimalizować. Regulator składa się z trzech członów (proporcjonalnego, całkującego i różniczkującego).

Parametrami algorytmu dostrajanymi przez projektanta są:

- wzmocnienie k
- stała całkowania T_i
- stała różniczkowania T_{mathrmd}

Zasada działania regulatora PID jest bardzo prosta, największy problem sprawia dobór odpowiednich parametrów w taki sposób, aby znaleźć zadowalający nas balans między szybkością regulacji, przeregulowaniem i uchybem ustalonym. Jest on prosty w implementacji i nie wymaga dużych nakładów obliczeniowych. Niestety, sprawdza się on dobrze tylko w przypadku układów z łagodną dynamiką i niewielkimi opóźnieniami.

2.3. Algorytm DMC

W przypadku obiektów z dużym opóźnieniem znacznie lepsze efekty przynosi zastosowanie algorytmów predykcyjnych, tzn. takich, które nie tylko wykorzystują obecną wartość wyjścia układu, ale także przewidują jego przyszłe zachowanie. Przykładem takiego algorytmu jest algorytm DMC. W każdej kolejnej dyskretniej chwili na podstawie obecnego stanu układu i w zależności od dobranych parametrów oblicza wektor przyszłych wartości sterowania i wykorzystuje pierwszy jego element $\Delta u(k|k)$ jako właściwą wartość wysyłaną na sterowanie obiektem.

Parametrami algorytmu dostrajanymi przez projektanta są:

- horyzont sterowania N_u
- horyzont predykcji N
- parametr λ

2.4. Ograniczenia

W przypadku sterowania obiektem rzeczywistym bardzo ważną rolę pełnią ograniczenia narzucane na projektanta przez fizyczne właściwości sprzętu. Ograniczeń takich są dwa typy - na wartość sterowania oraz na jej przyrost. Niezastosowanie się do tych restrykcji może spowodować zniszczenie sprzętu, ważna jest więc ich implementacja w kodzie algorytmu. Na szczęście jest to bardzo proste - wystarczy obliczone wartości rzutować na dopuszczalny ich zbiór przed przekazaniem ich na obiekt.

3. Obiekt symulowany - część projektowa

Pierwszym etapem naszej pracy była analiza dostarczonego przez prowadzącego programu symulującego działanie obiektu. Był to obiekt dyskretny z opóźnieniem zależny od zmiennych procesu $U(k-11)$, $U(k-10)$, $Y(k-1)$, $Y(k-2)$.

Program do obsługi całego zadania symulacji został zaimplementowany w pliku `Projekt1.m`. Wykonywane są w nim wszystkie wymagane polecenia niezbędne do wykonania zadania, tj. sprawdzenie poprawności wartości w punkcie pracy U_{pp} i Y_{pp} , wyznaczenie odpowiedzi skokowych procesu, wywołanie funkcji implementujących algorytmy PID i DMC, a także optymalizacja wskaźnika jakości regulacji E .

3.1. Wyznaczenie odpowiedzi skokowych procesu

Rozpoczynając od punktu pracy $U_{pp} = 0,8$ i $Y_{pp} = 2$ wyznaczaliśmy różne odpowiedzi skokowe dla kilku zmian sygnału sterującego. Wyniki symulacji przedstawione są na Rys. 3.1. Jak widać, symulowany obiekt jest stabilny (charakterystyki po pewnym czasie ustalają się na stałym poziomie) oraz liniowy (charakterystyki statyczne $Y(U)$ leżą w przybliżeniu na linii prostej, co przedstawione jest na Rys. 3.1).

Na podstawie tej charakterystyki możemy obliczyć wzmocnienie statyczne procesu. Definiuje się je jako

$$K = \frac{\Delta Y}{\Delta U} \quad (3.1)$$

Parametry ΔY i ΔU są stałe dla dowolnie wybranych punktów na wykresie $Y(U)$, ponieważ charakterystyka jest liniowa. Należy więc wybrać dowolne punkty (U_i, Y_i) i (U_j, Y_j) i na ich podstawie obliczyć wzmocnienie statyczne K . Wynik:

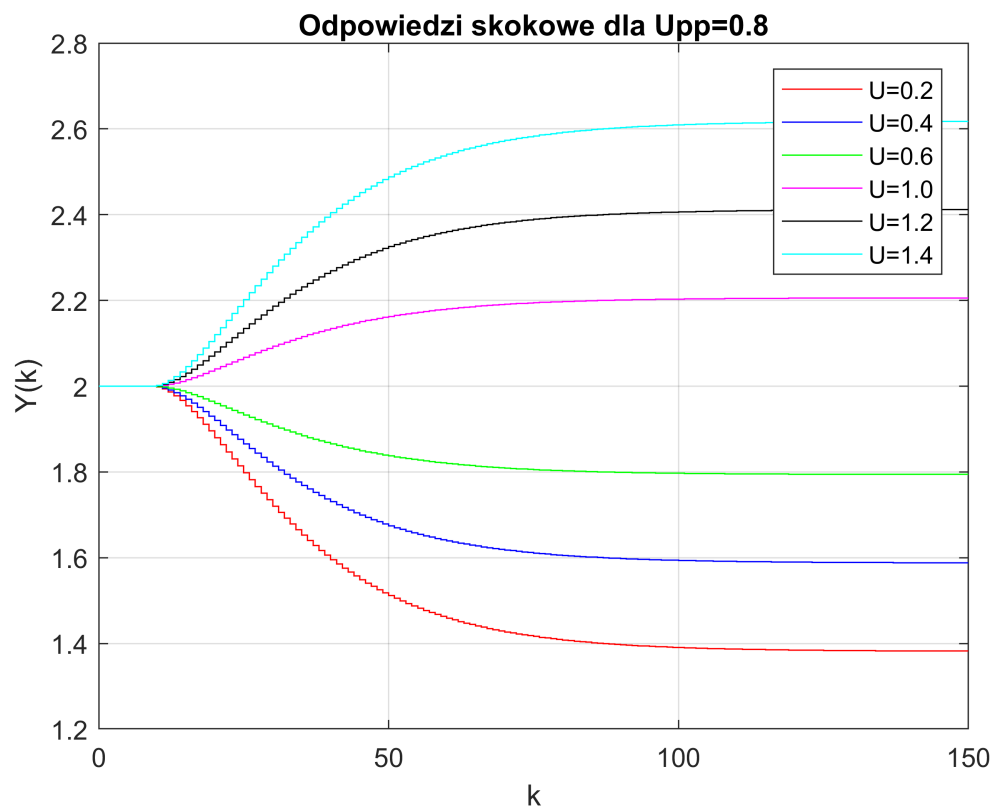
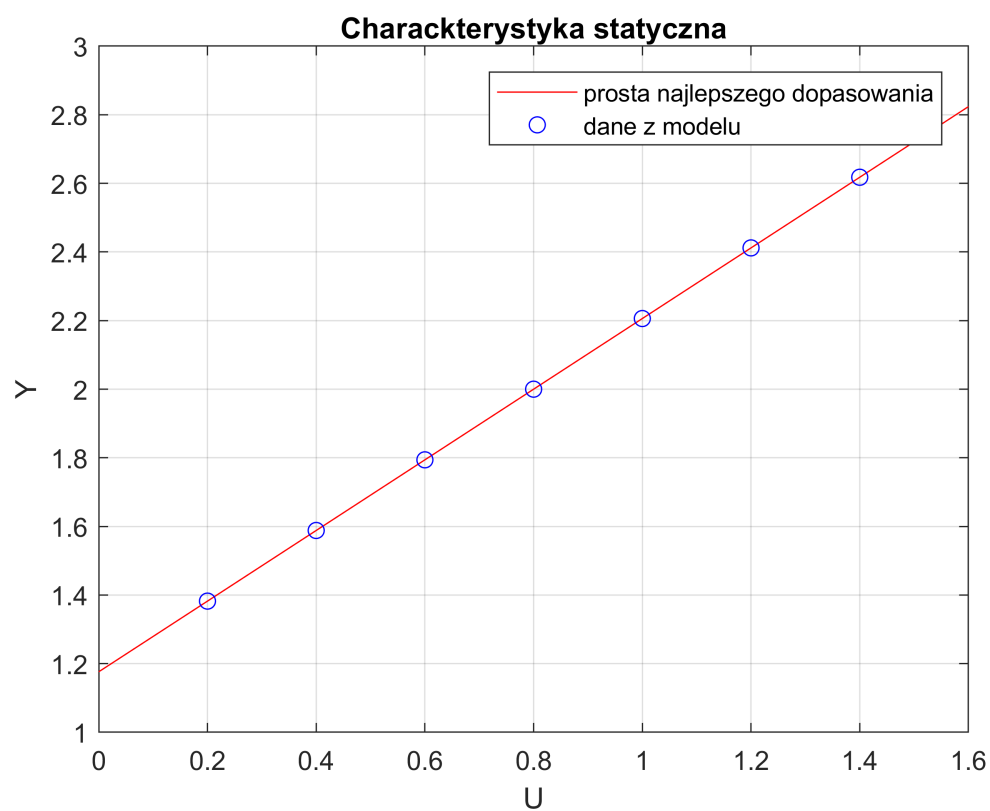
$$K = 1 \quad (3.2)$$

Aby otrzymać odpowiedź skokową wykorzystywaną w algorytmie DMC należy pobudzić obiekt skokiem jednostkowym, gdzie od chwili zerowej sygnał sterujący ma wartość 1, a w przeszłości jest zerowy. Wynik tej odpowiedzi skokowej jest przedstawiony na rys //TODO

3.2. Programy do symulacji algorytmów

Programy do symulacji algorytmów zostały zaimplementowane w plikach `PID.m` i `DMC.m`. Funkcje te są wywoływane w każdej kolejnej dyskretnych chwili k i obliczają wartość sterowania jaką należy przesłać na obiekt. Na wejście przyjmują więc obecne wartości zmiennych (PID - uchyb E , numer dyskretnych chwili k , parametry regulatora dyskretnego r_0 , r_1 , r_2 i wartość punktu pracy U_{pp} , a DMC dodatkowo jeszcze macierze K , M_P , wektor ΔU_P). Nakładają one także omówione wcześniej ograniczenia na sygnał sterujący. Funkcje te wywoływane są przez funkcje przeprowadzające symulacje `PID_simulation.m` i `DMC_simulation.m`, które to z kolei zwracają wartości wyjścia, sterowania i wskaźnika jakości i przekazują je do głównego programu.

Rys. 3.1. Odpowiedzi skokowe dla procesu symulowanego

Rys. 3.2. Charakterystyki statyczne $Y(U)$ 

//TODO

3.3. Ręczny dobór odpowiednich wartości parametrów

Nastawy regulatora PID i parametry regulatora DMC dobierano metodą eksperymentalną, tj. powoli i cierpliwie zmieniając ich wartości oraz obserwując rysunki przedstawiające przebiegi wyjścia procesu. Jak się okazuje optymalnie działający regulator PID przy nastawach ręcznych otrzymaliśmy przy następujących wartościach: $K = 0,01$, $T_i = 10000$, $T_d = 0$, natomiast wskaźnik jakości przyjął w tym wypadku wartość $E = 122,243$. Dla algorytmu DMC jakościowo dobre okazały się parametry: $N = 50$, $N_u = 1$, $\lambda = 1,5$. Wskaźnik jakości dla tych parametrów przyjął wartość $E = 60,614$. Wyniki symulacji z takimi parametrami znajdują się na rysunkach 3.3 oraz ...

3.4. Optymalizacja wskaźnika jakości

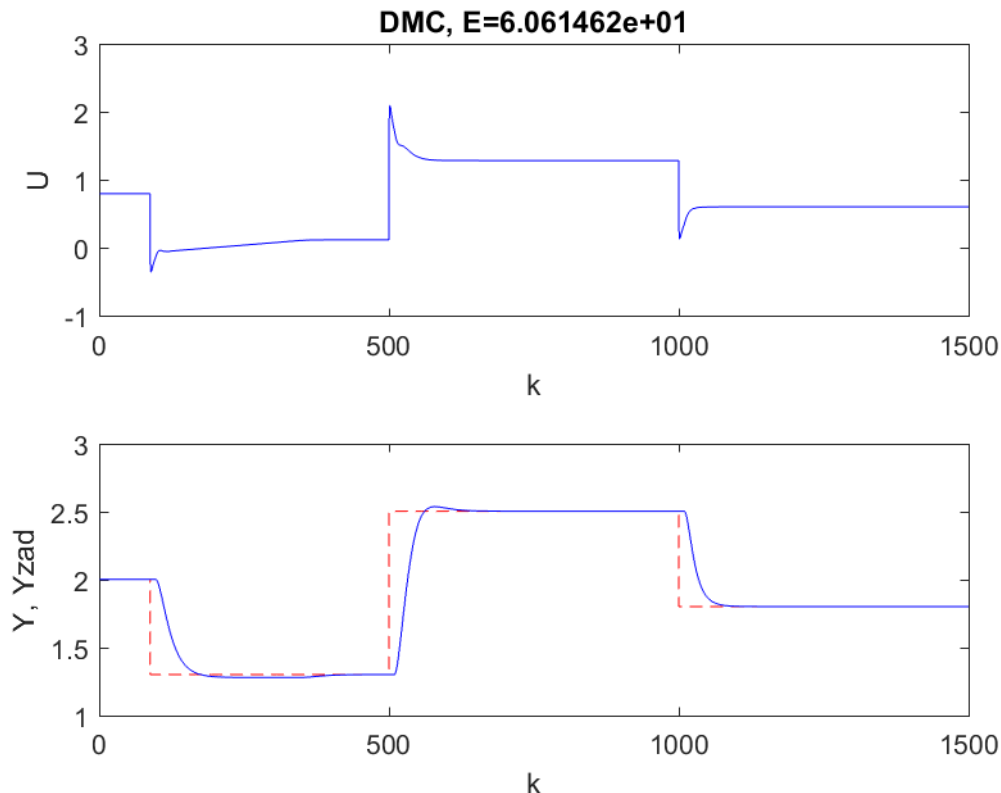
Następnie dokonano optymalizacji wskaźnika jakości regulatorów PID i DMC dla tej samej trajektorii z wykorzystaniem funkcji wbudowanych w środowisku Matlab. Za przyjęty wskaźnik jakości przyjęto sumę kwadratów różnic pomiędzy wartością zadaną, a wartością wyjścia symulowanego obiektu:

$$E = \sum_{k=1}^N (Y^{zad}(k) - Y(k))^2 \quad (3.3)$$

Dla regulatora PID do doboru optymalnych parametrów wykorzystano funkcję `fmincon`.
//TODO: PID do opisanja

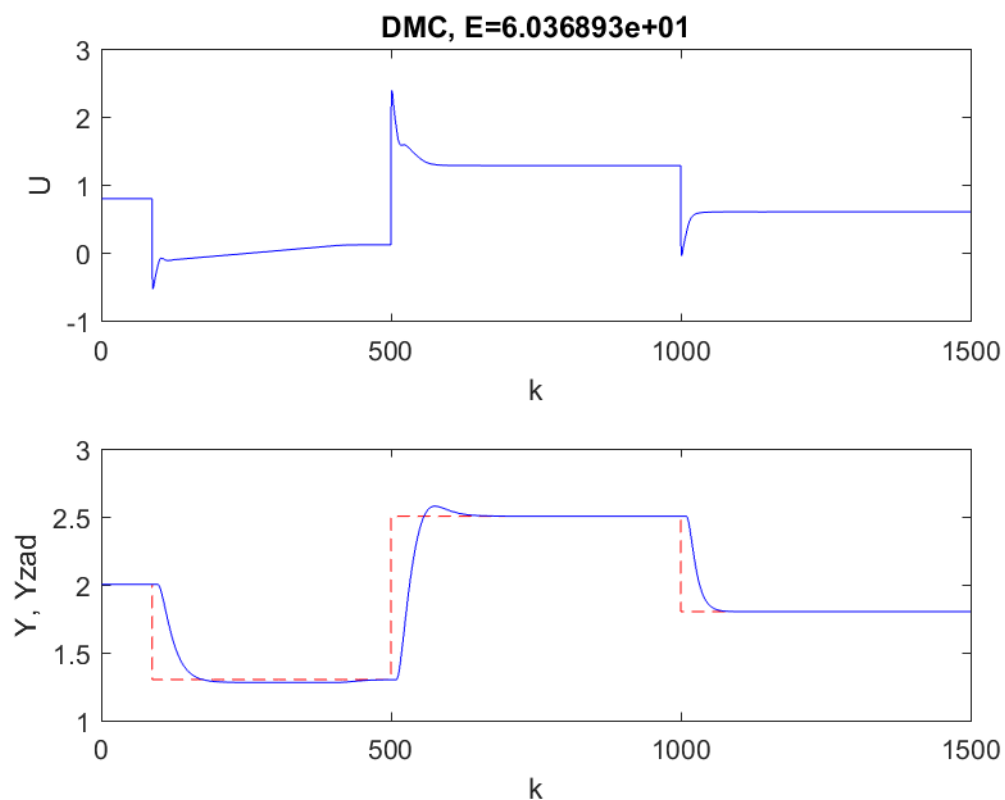
W przypadku regulatora DMC ze względu na parametry N , N_u przyjmujące wartości całkowite należało zastosować funkcję umożliwiającą optymalizację wskaźnika jakości dla parametrów o wartościach całkowitych. W tym celu wykorzystano funkcję `ga`, czyli funkcję wykorzystującą

Rys. 3.3. Przebieg symulacji DMC dla parametrów dobranych metodą eksperymentalną



algorytm genetyczny do optymalizacji. W ten sposób otrzymano parametry $N = 43$, $N_u = 1$, $\lambda = 1,0555$. Wskaźnik jakości dla tych parametrów wynosi $E = 60,369$. Wyniki symulacji z takimi parametrami znajdują się na rysunkach 3.4

Rys. 3.4. Przebieg symulacji DMC dla parametrów dobranych poprzez optymalizację wskaźnika jakości

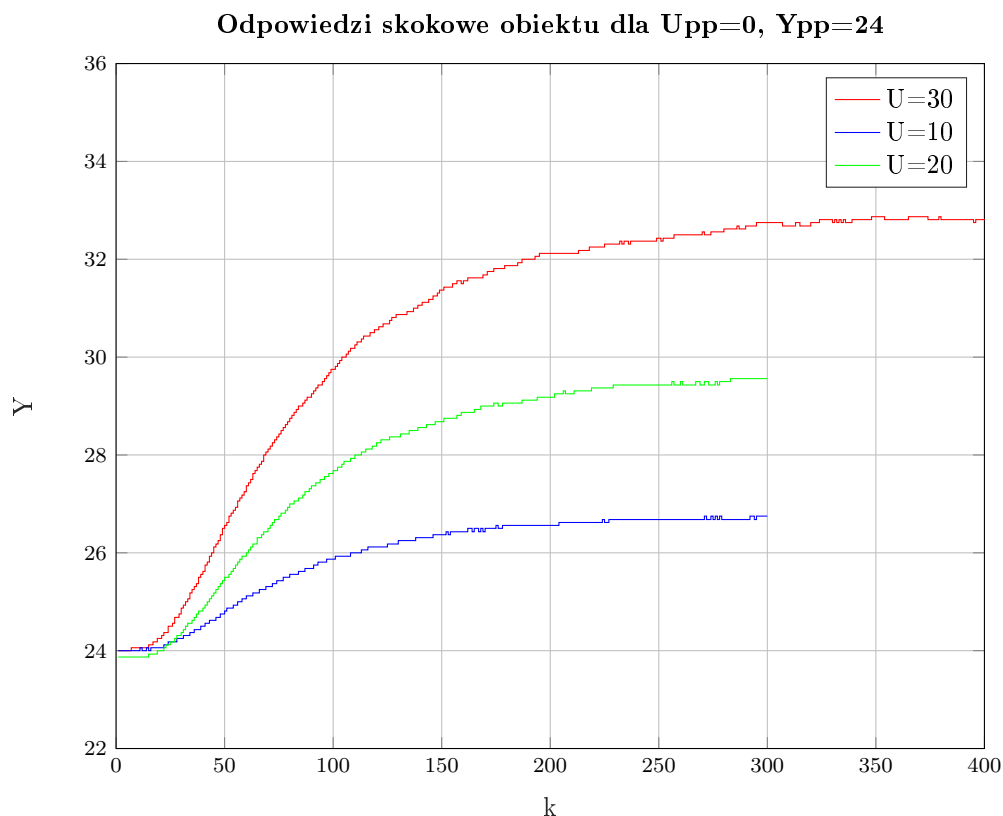


4. Sterowanie rzeczywistym obiektem - część laboratoryjna

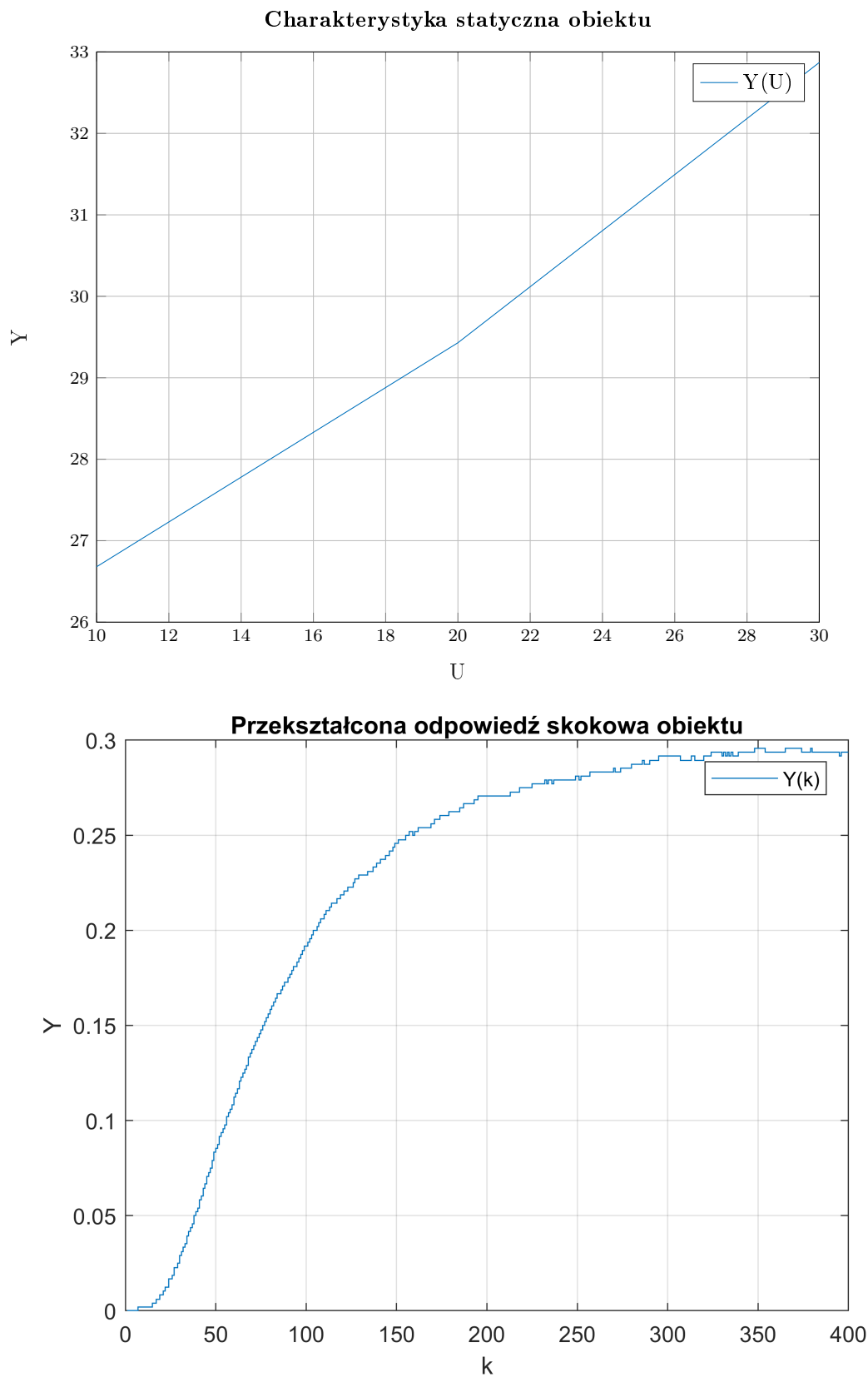
Zadanie laboratoryjne polegało na zebraniu danych z rzeczywistego obiektu oraz implementacji na nim algorytmów regulacji PID i DMC. Badanym obiektem było stanowisko grzewczo-chłodzące. Obiekt ten charakteryzuje się opóźnieniem na wejściu oraz wolną dynamiką. Taka charakterystyka obiektu ma znaczący wpływ na jakość danych regulatorów. Do zadania sterowania, jako wejście posłużyła moc grzałki sterowana w granicach 0-100% (G1), wyjściem była temperatura mierzona przy grzałce (T1). Dodatkowo zostało wprowadzone stałe zakłócenie w postaci wentylatora (W1) którego moc była ustawiona na 50% (traktowane jako cecha otoczenia).

4.1. Odpowiedzi skokowe

Pierwszą częścią zadania było pobranie odpowiedzi skokowych z obiektu zaczynając z ustalonego punktu pracy. Wybrany przez nas punktem pracy było $U_{pp} = 0$. Dla takiej wartości, wartość wyjścia w punkcie pracy wynosiła około $Y_{pp} = 24$. Wykonano 3 eksperymenty dla różnych skoków sterowania. Ich efekty przedstawione są poniżej: Doświadczenie dla $U = 30$ zostało przeprowadzone na dłuższej próbie czasowej ze względu na dłuższy czas ustalenia się wyjścia obiektu dla zadanego sterowania. Już na pierwszy rzut oka można stwierdzić, że obiekt ten jest



liniowy co potwierdza tylko jego charakterystyka statyczna. Tak zebrane odpowiedzi skokowe posłużyły następnie do wyznaczenia dopowiedzi skokowej dla skoku jednostkowego, która następnie posłużyła do implementacji algorytmu DMC.



4.2. Programy do symulacji algorytmów

Lsitingi programów do symulacji przedstawiają się następująco:

Algorytm PID:

```
function [Y,U,Yzad] = PID(K,Ti,Td)
addpath('F:\SerialCommunication'); % add a path to the functions
initSerialControl COM3 % initialise com port

simTime=400;

T=1.0;
r0=K*(1+T/(2*Ti)+Td/T);
r1=K*(T/(2*Ti)-2.0*Td/T-1);
r2=(K*Td)/T;

min_U=0.0;
max_U=100.0;
Upp=0;
Ypp=24;
y_zad=26;

E=zeros(2,1);
U=Upp*ones(2,1);
Y=Ypp*ones(2,1);
Yzad=y_zad*ones(2,1);

k=3;

time_buf=200;
U_buf=zeros(time_buf,1);
Y_buf=Ypp*ones(time_buf,1);
Yzad_buf=y_zad*ones(time_buf,1);

%while(1)
for k=3:simTime

if k<200
Yzad(k)=26;
else
Yzad(k)=28;
end

E(k)=Yzad(k)-Y(k-1);

U(k)=r2*E(k-2)+r1*E(k-1)+r0*E(k)+U(k-1);

if U(k) > max_U
U(k)=max_U;
elseif U(k)<min_U
U(k)=min_U;
end

sendControls([ 1, 5], ...
[ 50, U(k)]);

Y(k) = readMeasurements(1); % read measurements from 1 to 7
```

```

Y_buf(1:time_buf-1)=Y_buf(2:time_buf);
Y_buf(time_buf)=Y(k);
Yzad_buf(1:time_buf-1)=Yzad_buf(2:time_buf);
Yzad_buf(time_buf)=Yzad(k);
U_buf(1:time_buf-1)=U_buf(2:time_buf);
U_buf(time_buf)=U(k);

figure(1);
subplot(2,1,1);
stairs(U_buf,'b');
ylabel('U');
xlabel('k');
title('U(k)');
subplot(2,1,2);
stairs(Yzad_buf,'--r');
hold on;
stairs(Y_buf,'b');
ylabel('Y');
xlabel('k');
title('Y(k)');
hold off;
drawnow;

%k=k+1;

%% synchronising with the control process
waitForNewIteration(); % wait for new batch of measurements to be ready
end
end

```

Alogrytm DMC:

```

function [Uout,DUP] = DMC(K,MP,DUP,U,Y,Yzad,k,Upp)

%global limits
limits=1;

Ypp = 2;
Umin = 0;
Umax = 100;
dUmax = 10;

u = U - Upp;
y = Y - Ypp;
yzad = Yzad - Ypp;
umin = Umin - Upp;
umax = Umax - Upp;
dumax = dUmax;

D = length(DUP) + 1;
N = size(MP,1);

DUP = [u(k-1) - u(k-2) ; DUP(1:D-2)];

vec_Y0 = y(k)*ones(N,1)+MP*DUP;
vec_Yzad = yzad(k)*ones(N,1);

du = K*(vec_Yzad-vec_Y0);

```

```

if limits ~= 0
if du(1) < -dumax
du(1) = -dumax;
end;
if du(1) > dumax
du(1) = dumax;
end;
u(k) = du(1) + u(k-1);
if u(k) < umin
u(k) = umin;
end;
if u(k) > umax
u(k) = umax;
end;

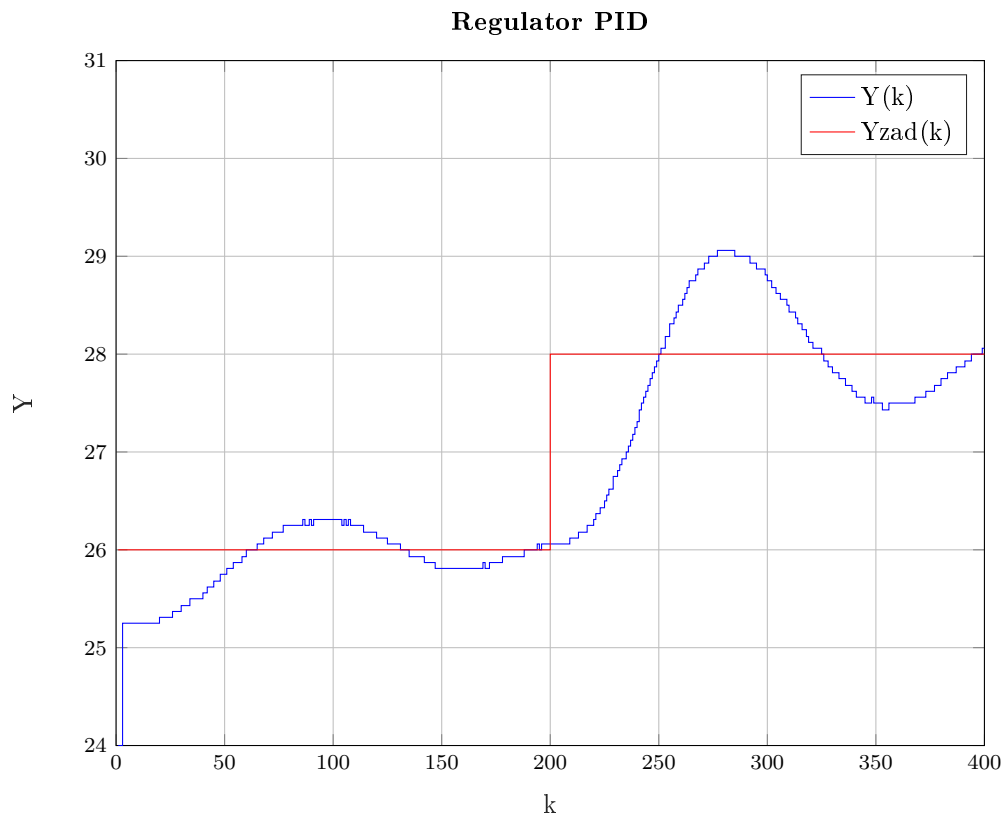
else
u(k) = du(1)+u(k-1);
end;

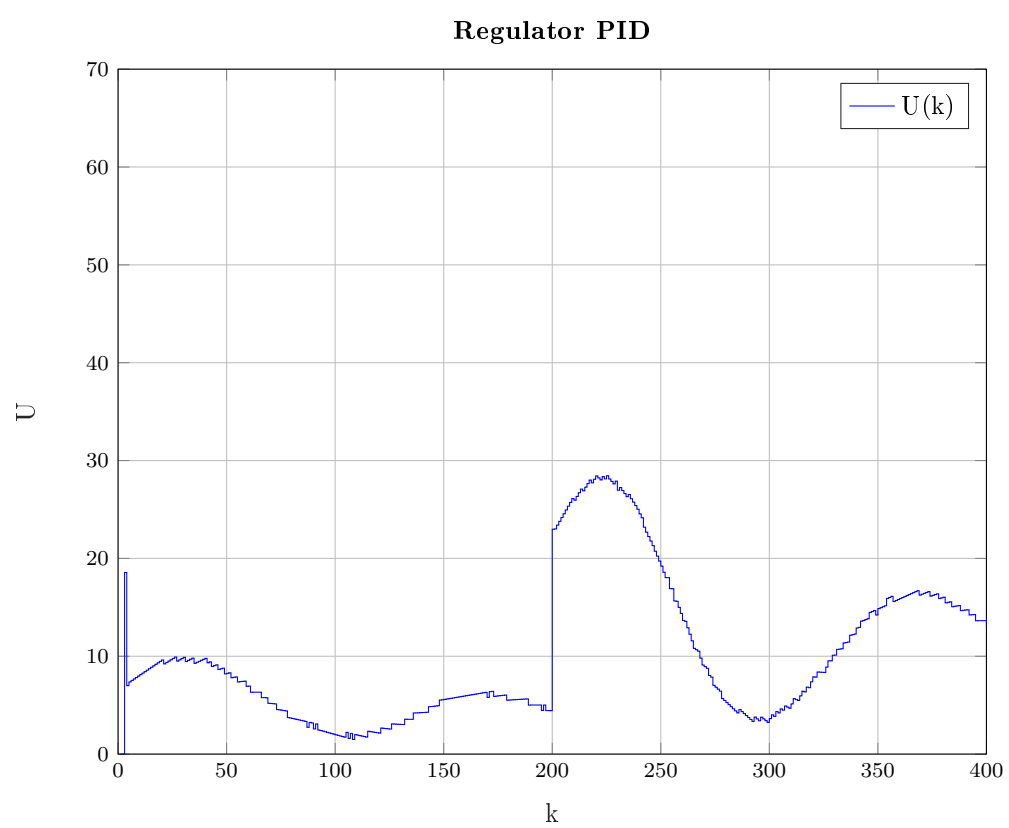
Uout = u(k)+Upp;
end

```

4.3. Regulator PID

Poniższy wykres przedstawia działanie zaimplementowanego algorytmu PID na obiekcie. Nastawy zostały wybrane dla niego metodą inżynierską i są następujące: $K = 9$, $T_i = 45$, $T_d = 0.02$. Wynik eksperymentu dla dwóch skoków wartości zadanej są przedstawione na poniższym wykresie:

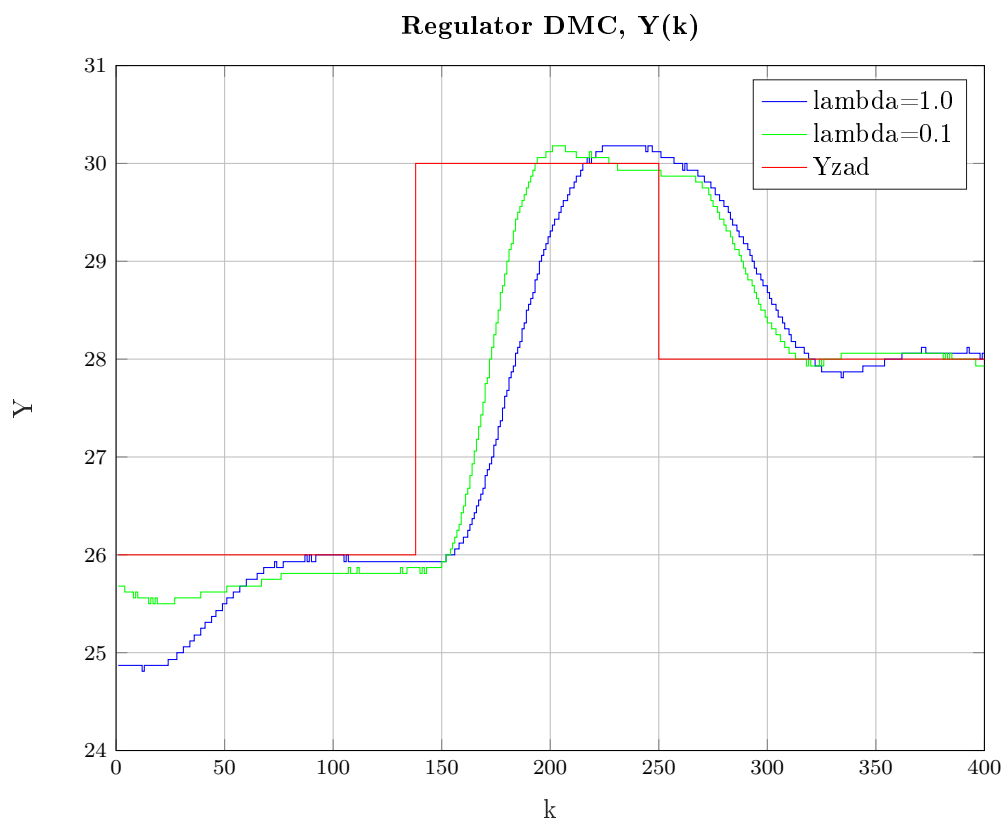


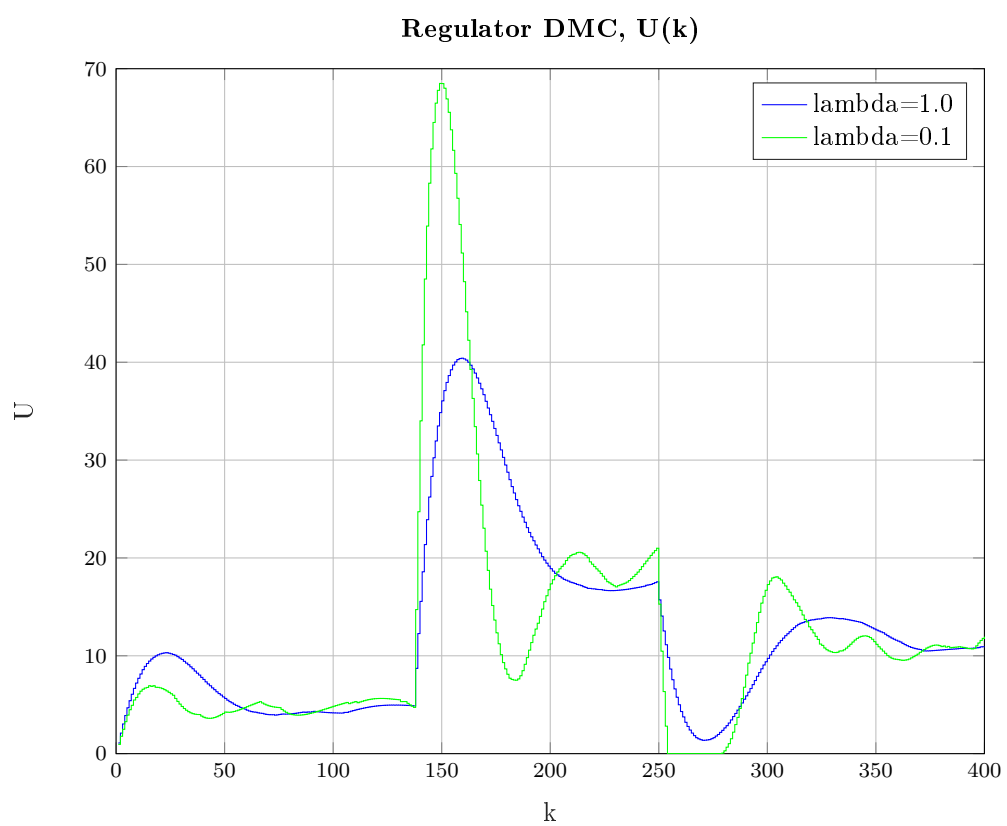


Jak widać regulator PID słabo radzi sobie z regulacją obiektu o tak wolnej dynamice. Wynika to z faktu że operuje on tylko na danych pochodzących bezpośrednio z obiektu i nie może w żaden sposób przewidzieć jego zachowania w przyszłości, dla którego mógłby dostosować wartość sterowania. W takim przypadku lepiej radzi sobie algorytm DMC.

4.4. Algorytm DMC

Poniższy wykres przedstawia przebiegi regultora DMC dla danego obiektu. W obu przypadkach użyto identycznych horyzontów dynamiki i sterowania wynoszących odpowiednio: $N_D = 400$, $N_U = 200$. Przeprowadzono dwa eksperymenty dla różnych wartości λ : 0.1 i 1.0. Oto efekty przeprowadzonych doświadczeń:





Jak widać na powyższych wykresach algorytm DMC radzi sobie znacznie lepiej przy sterowaniu obiektem o wolniejszej dynamice, niż algorytm PID. Wynika to z faktu iż DMC posługuje się modelem obiektu w postaci odpowiedzi skokowej, dzięki czemu może on przewidzieć zachowanie się obiektu w danej sytuacji i odpowiednio wcześniej reagować, zmieniając wartości sterowania przez co osiąga on wartość zadaną w znacznie szybszy i stabilniejszy sposób niż PID.

5. Wnioski

Przeprowadzone eksperymenty dowodzą, że w rozważanym typie procesu jednowymiarowego z opóźnieniem oba algorytmy radzą sobie ze sterowaniem - nie pojawiają się znaczące przeregulowania ani rosnące oscylacje. Niemniej jednak, zgodnie z hipotezą postawioną na początku, algorytm DMC przeprowadza regulację o wiele sprawniej i dokładniej dzięki predykcjom przyszłego sterowania wykonywanym na każdym kroku i to właśnie tego algorytmu powinniśmy używać w przypadku projektowania regulatora dla obiektów podobnych do testowanego przez nas stanowiska grzejąco-chłodzącego.

Warto zwrócić uwagę na fakt, że obiekt zarówno symulowany, jak i rzeczywisty był bardzo uproszczony - pomijaliśmy wpływ jakichkolwiek zakłóceń na jego działanie i tak naprawdę nie miało się co tam popsuć. Aby lepiej odwzorować procesy realizowane w prawdziwym świecie (np. przemyśle), należałoby jeszcze zbadać zachowanie układu w warunkach trudniejszych (z różnego rodzaju zakłóceniami) i dopiero na tej podstawie dobrać. Takie eksperymenty i rozważania przeprowadzać będziemy na kolejnych zadaniach laboratoryjnych.