

Sterowanie Procesami - Algorytmy PID oraz DMC

Natan Orzechowski

May 2020

Spis treści

1	Wstęp	4
2	Wyznaczanie transmitancji dyskretnej	4
2.1	Transmitancja dyskretna z ekstrapolatorem zerowego rzędu . . .	4
2.2	Porównanie odpowiedzi skokowej obu transmitancji	5
2.3	Porównanie współczynników wzmocnienia statycznego	6
2.4	Wnioski	7
3	Wyznaczanie równania różnicowego	7
4	Regulator ciągły PID - metoda Zieglera-Nicholsa	8
4.1	Regulator PID - podstawy teoretyczne	8
4.2	Parametry ciągłego regulatora PID	8
4.3	Algorytm Zieglera-Nicholsa	12
4.4	Regulator PID ciągły	13
4.5	Regulator PID dyskretny	13
5	Symulacja algorytmów cyfrowego PID oraz DMC wersja analityczna	14
5.1	Realizacja regulatora dyskretnego PID	14
5.1.1	Implementacja oraz wykres	14
5.1.2	Wnioski	15
5.2	Realizacja dyskretnego regulatora DMC w wersji analitycznej bez ograniczeń	15
5.2.1	Wstęp	15
5.2.2	Oznaczenia przyjęte w skrypcie	16
5.2.3	Implementacja	17
6	Dobieranie parametrów algorytmu DMC	18
6.1	Horyzont dynamiki D	18
6.2	Horyzont predykcji N	20
6.2.1	Zdefiniowanie zakresu szukania optymalnej wartości . . .	21
6.2.2	Wyznaczenie optymalnej wartości za pomocą skryptu . .	24
6.2.3	Wnioski	25
6.3	Horyzont sterowania N_u	25
6.3.1	Badanie wpływu horyzontu sterowania na proces regulacji	26
6.3.2	Wnioski	29
6.4	Współczynnik dostrajany λ	29
6.4.1	Badanie wpływu współczynnika λ	29
6.4.2	Wnioski	32
6.5	Podsumowanie	33
6.5.1	Wpływ zmian horyzontu predykcji N na odpowiedź skokową	33
6.5.2	Wpływ zmian horyzontu sterowania N_u na odpowiedź skokową	33

6.5.3	Wpływ zmian λ na odpowiedź skokową	34
6.5.4	Końcowa wersja regulatora DMC	34
7	Porównanie jakości obu metod regulacji	35
7.1	Porównanie parametrów jakości	35
7.2	Wyznaczanie obszarów stabilności	37
7.2.1	Algorytm PID	37
7.2.2	Algorytm DMC	39
7.2.3	Wnioski	40
8	Podsumowanie	41

1 Wstęp

Projekt ten będzie polegał na symulowaniu rozmaitych procesów związanych z zadaniem obiektem regulacji w całości w pakiecie MATLAB. Obiekt ten jest opisany transmitancją:

$$G(s) = \frac{K_0 e^{-T_0 s}}{(T_1 s + 1)(T_2 s + 1)} \quad (1)$$

Gdzie $K_0 = 4.6, T_0 = 5, T_1 = 2.13, T_2 = 4.67$. Dokument ten skupiać się będzie na obliczeniach prowadzonych w pakiecie MATLAB i wykorzystywanych w ich celu funkcjach, toteż oprócz stosownych opisów wykonywanych zadań będą prezentowane również wybrane fragmenty kodu.

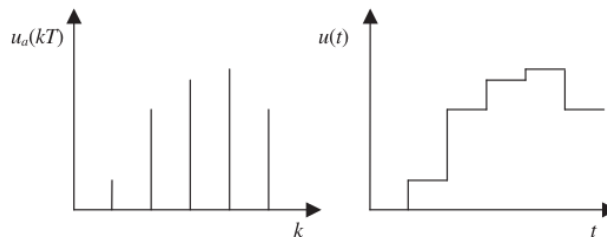
2 Wyznaczanie transmitancji dyskretnej

W tej sekcji zaprezentowano sposób wyznaczania transmitancji dyskretnej przy zadanych parametrach oraz porównano odpowiedzi skokowe transmitancji ciągłej oraz dyskretnej.

2.1 Transmitancja dyskretna z ekstrapolatorem zerowego rzędu

Transmitancja dyskretna będzie opisywać obiekt tylko w dyskretnych chwilach $T_p, 2T_p, 3T_p$ itd. Sygnał, który wyznaczono, będzie miał postać impulsów Diraca, a więc będą to słupki na wykresie napięcia w zależności od czasu. W celu przekształcenia impulsów na sygnał sterujący ciągły, należy zastosować ekstrapolator (w tym przypadku zerowego rzędu), który działa w ten sposób, że utrzymuje ustaloną przez impuls wartość napięcia w czasie dopóki nie nadejdzie kolejny impuls, niosący nową wartość. Zasada działania ekstrapolatora zerowego rzędu (zwanego też ZOH - Zero Order Holder) została przedstawiona na poniższym schemacie.

Rysunek 1: Zasada działania ZOH.



W celu wyznaczenia transmitancji dyskretnej układu, można przekształcić transmitancję ciągłą na iloczyn:

$$G(s) = G_0(s)e^{-T_0 s} \quad (2)$$

... a następnie rozłożyć $G_0(s)$ na ułamki proste. Przygotowaną tak postać transmitancji należy poddać transformacji Z wg następującego wzoru:

$$G(z) = \left[\frac{z-1}{z} \right] Z \left[\frac{G_0(s)}{s} \right] z^{-T_0} \quad (3)$$

W pakiecie MATLAB został stworzony skrypt, który korzysta z funkcji *c2d*, która podaje transmitancję dyskretną dla argumentów wejściowych w postaci licznika i mianownika transmitancji ciągłej, czasu próbkowania oraz rzędu ekstrapolatora. W tym przypadku zastosowano $T_p = 0.5s$ oraz ekstrapolator zerowego rzędu (ZOH) zgodnie z wytycznymi.

Rysunek 2: Fragment kodu zastosowanego w zadaniu.

```
%% Obliczenie licznika i mianownika transmitancji dyskretnej

% Funkcja c2d pozwala na wyznaczenie transmitancji dyskretnej z wybranym
% ekstrapolatorem (domyślnie zoh)
[N, D] = c2d(H, Tp); % ekstrapolator zerowego rzędu
Hd = tf(N, D, Tp); %transmitancja dyskretna
```

Transmitancja dyskretna wyznaczona za pomocą MATLABa wyniosła:

$$G(z) = \frac{0.05164z + 0.04608}{z^2 - 1.689z + 0.7105} z^{-10} \quad (4)$$

2.2 Porównanie odpowiedzi skokowej obu transmitancji

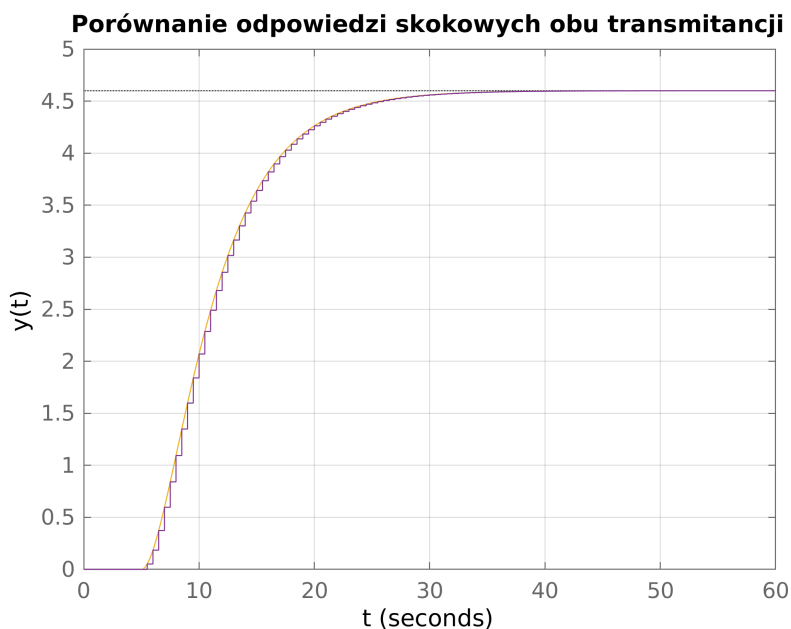
W celu weryfikacji poprawności wyznaczenia transmitancji dyskretnej, zasy-mulowano dla obu przypadków odpowiedź układu na skok jednostkowy przy zerowych warunkach początkowych (tzw. odpowiedź skokową). W tym celu zastosowano funkcję *step()*.

Rysunek 3: Fragment kodu - rysowanie odpowiedzi skokowych.

```
%% Wyznaczenie odpowiedzi skokowej transmitancji ciągłej
step(H);
stepI = stepinfo(H); %parametry tej odpowiedzi

%% Wyznaczenie odpowiedzi skokowej transmitancji dyskretnej
step(Hd);
stepII = stepinfo(Hd);
```

Odpowiedzi obu układów zaprezentowano na przebiegu poniżej. Widać wy-



raźnie, iż oba przebiegi mają ten sam charakter. Parametry obu przebiegów wyznaczone w MATLABie są bardzo do siebie zbliżone. Na przykład parametry RiseTime dla modelu ciągłego i dyskretnego wynoszą odpowiednio 11.83 oraz 12, zaś Peak pokazał wyniki 4.6 w obu przypadkach.

2.3 Porównanie współczynników wzmocnienia statycznego

Współczynniki wzmocnienia statycznego wyznaczamy ze wzorów:

$$\lim_{s \rightarrow 0} G(s) = 4.6 \quad (5)$$

$$\lim_{z \rightarrow 1} G(z) = 4.545 \quad (6)$$

Odchylenie transmitancji dyskretniej względem ciągłej wyniósł 1,2 procenta. Biorąc pod uwagę, iż proces obliczeń był czysto numeryczny, można przyjąć, iż obie te wartości są sobie równe.

2.4 Wnioski

Proces wyznaczania transmitancji dyskretnej w pakiecie MATLAB jest znacznie prostszy od wyznaczania analitycznego, które może okazać się niekomfortowe przy takich danych (obliczenia z wieloma miejscami po przecinku). Odpowiedzi skokowe, jeżeli są poprawnie zasymulowane, będą się pokrywać. Wzmocnienie statyczne dla obu transmitancji powinno być równe sobie, co faktycznie ma miejsce. Modele zatem są zgodne z teorią.

3 Wyznaczanie równania różnicowego

Równanie różnicowe pozwala na obliczenie wartości na wyjściu układu w konkretnych chwilach czasowych na podstawie historycznych danych.

Wiemy, że transmitancja to iloraz wyjścia i wejścia układu:

$$G(z) = \frac{Y(z)}{U(z)} \quad (7)$$

Możemy zatem podstawić znaną nam transmitancję dyskretną oraz wymnożyć równanie stronami:

$$\frac{Y(z)}{U(z)} = \frac{0.05164z + 0.04608}{z^2 - 1.689z + 0.7105} z^{-10} \quad (8)$$

$$Y(z) [z^2 - 1.689z + 0.7105] = U(z) [0.05164z^{-9} + 0.04608z^{-10}] \quad (9)$$

Wiadomo, że z^{-n} odpowiada opóźnieniu o n próbek, a zatem:

$$0.05164u(k-9) + 0.04608u(k-10) = y(k+2) - 1.689y(k+1) + 0.7105y(k) \quad (10)$$

Szukamy równania takiego, które dla wskazanej chwili czasowej, da wartość wyjścia na podstawie wartości wcześniejszych. Konieczne jest zatem przesunięcie wyznaczonych wartości w równaniu (10) o dwie do tyłu. Ostatecznie:

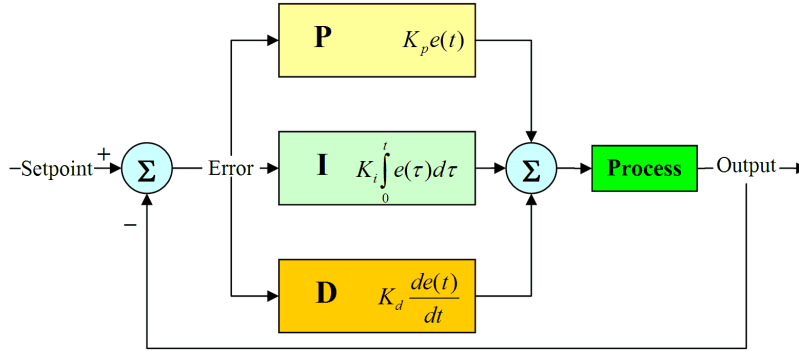
$$y(k) = 1.689y(k-1) - 0.7105y(k-2) + 0.05164u(k-11) + 0.04608u(k-12) \quad (11)$$

4 Regulator ciągły PID - metoda Zieglera-Nicholsa

4.1 Regulator PID - podstawy teoretyczne

Regulator PID pracuje w pętli sprzężenia zwrotnego, oblicza wartość uchybu jako różnicę pomiędzy pożądaną wartością zadaną i zmierzoną wartością zmiennej procesu i działa w taki sposób, by zredukować uchyb poprzez odpowiednie dostosowanie sygnału podawanego na wejście regulowanego obiektu. Składa się z członu wzmacniającego, różniczkującego i całkującego.

Rysunek 4: Schemat obiektu z regulacją PID.



Wyjście regulatora można przedstawić jako:

$$u(t) = K \left[1 + \frac{1}{T_i} \int_0^t e(\tau) d\tau + T_d \frac{de(t)}{dt} \right] e(t) \quad (12)$$

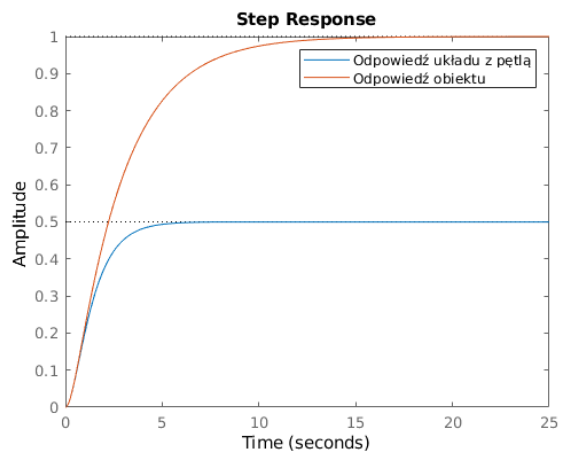
4.2 Parametry ciągłego regulatora PID

Wpływ parametrów regulatora na obiekt regulacji został przedstawiony na przykładzie powyższego schematu, gdzie obiekt demonstracyjny ma transmitancję (13).

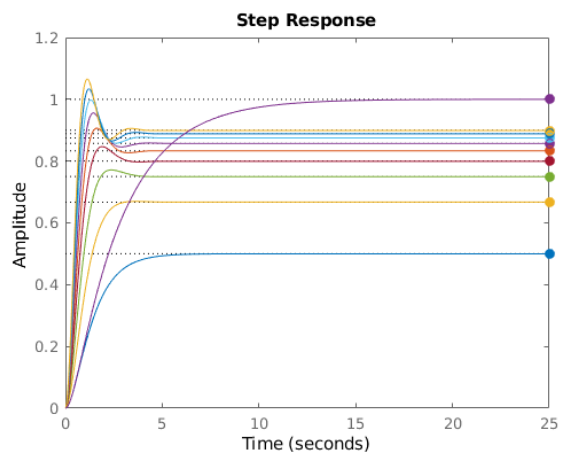
$$G_{dem}(s) = \frac{1}{s^2 + 3s + 1} \quad (13)$$

Na poniższym schemacie przedstawiono odpowiedź skokową transmitancji demonstracyjnej oraz odpowiedź skokową układu z regulatorem PID o nastawach nie wpływających na obiekt ($Kr = 1$, Ti oraz Td brak).

Rysunek 5: Porównanie odpowiedzi skokowych obiektu oraz układu.

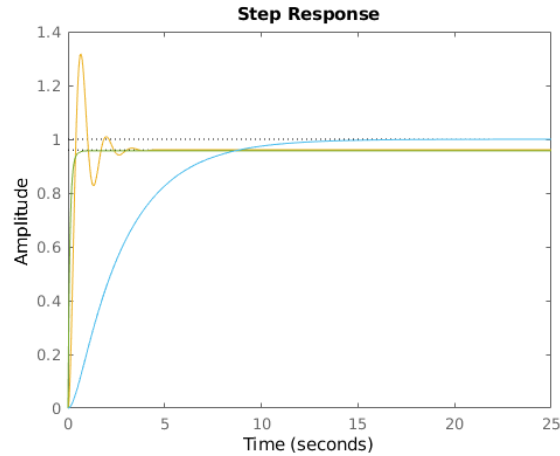


W miarę zwiększania parametru K_r odpowiadającego za wzmocnienie w regulatorze, obserwujemy szybsze osiągnięcie wartości granicznej, ale też znaczny "pik".

Rysunek 6: Nowe przebiegi dla kolejnych wartości całkowitych K_r od 1 do 9.

W celu spłaszczenia owych przebiegów, konieczne jest dodanie członu różniczkującego, który będzie odpowiadać za zmniejszenie fluktuacji odpowiedzi skokowej przy wartości granicznej, jednocześnie zachowując dobry czas dążenia do niej.

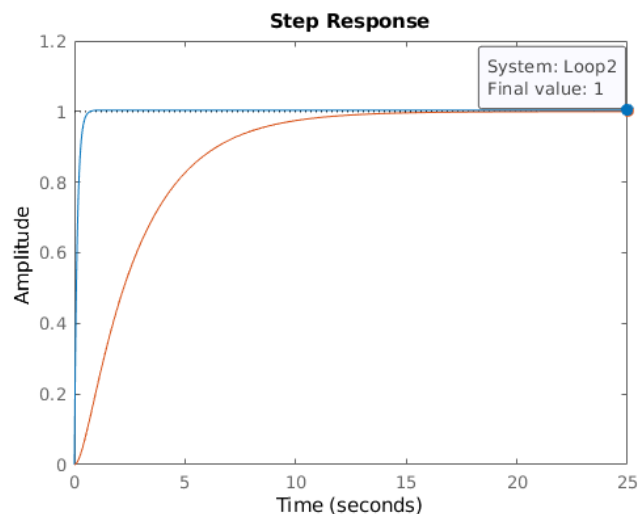
Rysunek 7: Przebiegi dla wartości $T_d = 0$ i $T_d = 10$



Na powyższym wykresie widać wyraźnie, iż w miarę zwiększania parametru T_d przebieg się spłaszcza. Zielony przebieg oznacza odpowiedź skokową dla układu o nastawach $K_r = 24, T_d = 8$.

Na tym etapie otrzymano układ, który szybko zmierza do zadanej wartości oraz osiąga to bez fluktuacji. Jego mankamentem jest natomiast to, iż nigdy on nie osiąga zadanej wartości 1. Można to osiągnąć, dodając człon całkujący. Na poniższym wykresie zaprezentowano efekt końcowy układu o parametrach regulatora PID $K_r = 24, T_d = 8, T_i = 9$.

Rysunek 8: Odpowiedź skokowa układu z nastrojonym regulatorem PID.



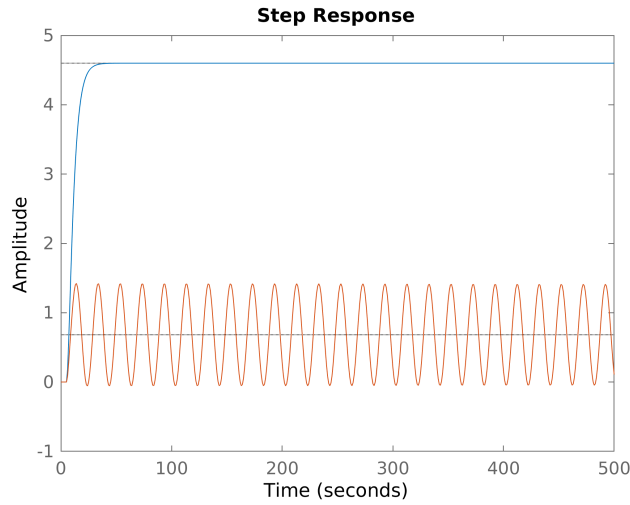
Jest to proces bardzo czasochłonny i nie dający stu procent pewności sukcesu. Należy pamiętać, iż mianownik transmitancji prezentowanego obiektu był wielomianem drugiego stopnia, co znacznie wpływa na prostotę regulacji pracy tego obiektu. W praktycznych zastosowaniach, mamy do czynienia z wielomianami wyższych rzędów, stąd zastosowania znajdują metody takie jak metoda Zieglera-Nicholsa, które pozwalają na stosunkowo proste i skuteczne strojenie regulatorów PID.

4.3 Algorytm Zieglera-Nicholsa

Algorytm ten polega na zastosowaniu do regulacji obiektu tylko członu wzmacniającego, a następnie na znalezieniu takich wartości parametrów K_k (wzmocnienia krytycznego) oraz T_k (okresu krytycznego), dla których układ wykazuje niegasnące, stałe oscylacje.

Znaleziono takie wartości metodą prób i błędów dość szybko i wyniosły one $K_k = 0.46615$ oraz $T_k = 20s$. Poniżej porównanie odpowiedzi skokowych obiektu oraz układu z regulatorem P.

Rysunek 9: Odpowiedź skokowa układu z regulatorem P w stanie oscylacji.



Na tym etapie można wyznaczyć poprawne nastawy regulatora zgodnie z zadanymi wartościami zmiennych:

$$K_r = 0.6K_k = 0.2797 \quad (14)$$

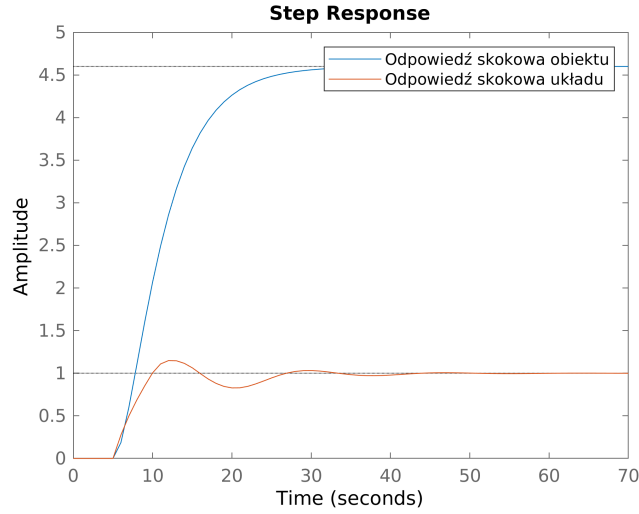
$$T_i = 0.5T_k = 10 \quad (15)$$

$$T_d = 0.12T_k = 2.4 \quad (16)$$

4.4 Regulator PID ciągły

Celem sprawdzenia działania regulatora, przeprowadzono symulację z nowymi nastawami, które potem posłużą do wyznaczenia współczynników dla regulatora dyskretnego.

Rysunek 10: Odpowiedź skokowa układu z nastrojonym regulatorem PID.



Regulator stabilizuje pracę układu w ten sposób, aby jego odpowiedź wynosiła stałe 1 (ma to związek ze skokiem jednostkowym i faktem, iż nie występuje wzmocnienie). Pokrywa się to ze wzmocnieniem statycznym układu zaprezentowanego na Rysunku 4., wyznaczonym z obliczonej transmitancji na podstawie modelu w przestrzeni stanu.

4.5 Regulator PID dyskretny

Analogicznie do układów ciągłych, możemy wyznaczyć regulator PID działający dla dyskretnych sygnałów. Wyjście takiego regulatora, a wejście obiektu, można przedstawić następująco za pomocą całkowania metodą trapezów.

$$u(k) = K_r \left[1 + \frac{T_p}{2T_i} \left[\frac{1 + z^{-1}}{1 - z^{-1} + \frac{T_d}{T_p}(1 - z^{-1})} \right] \right] e(k) \quad (17)$$

Niektóre zestawy zmiennych możemy zapisać jako parametry, co skutkuje uproszczeniem postaci wyjścia układu do:

$$u(k) = \frac{r_2 z^{-2} + r_1 z^{-1} + r_0}{1 - z^{-1}} e(k) \quad (18)$$

Gdzie parametry r_x oznaczają:

$$r_2 = \frac{K_r T_d}{T_p}, r_1 = K_r \left[\frac{T_p}{2T_i} - 2\frac{T_d}{T_p} - 1 \right], r_0 = K_r \left[1 + \frac{T_p}{2T_i} + \frac{T_d}{T_p} \right] \quad (19)$$

W przypadku omawianego w tym artykule modelu, parametry te przyjęły wartość:

$$r_2 = 1.3425, r_1 = -2.9577, r_3 = 1.6292 \quad (20)$$

Z tą wiedzą, możemy ustalić wartości wejścia obiektu w kolejnych chwilach czasu. Wyjścia tego obiektu z kolei, zdeterminowane są przez równanie różnicowe (11), wyznaczone w rozdziale nr 3. Implementacja i działanie dyskretnego regulatora PID zostało opisane w kolejnym rozdziale.

5 Symulacja algorytmów cyfrowego PID oraz DMC wersja analityczna

5.1 Realizacja regulatora dyskretnego PID

5.1.1 Implementacja oraz wykres

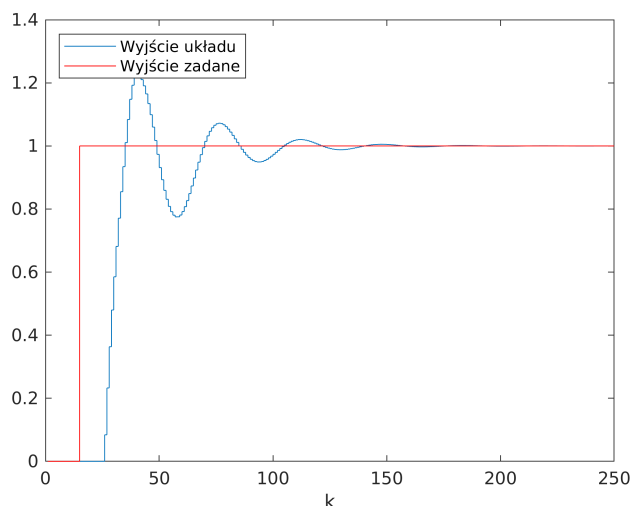
Z wyznaczonymi w poprzednim punkcie parametrami potrzebnymi do realizacji obliczania wartości wyjścia i wejścia układu, a także uchybu, przystąpiono do napisania pętli obliczającej kolejne wartości wychodzące z regulowanego układu.

Rysunek 11: Implementacja regulatora w pakiecie MATLAB.

```
%%                               Symulacja regulatora PID
for k = 13:simend
    y(k) = -a1*y(k-1)-a0*y(k-2)+b1*u(k-11)+b0*u(k-12);
    e(k) = yzad(k)-y(k);
    u(k) = r2*e(k-2)+r1*e(k-1)+r0*e(k)+u(k-1);
end
```

Działanie tego fragmentu skryptu jest bardzo proste - dla kolejnych chwil czasowych począwszy od $T = 13$ (jest to kwant czasu poprzedzający podanie logicznej jedynki na wyjście zadane układu), skrypt oblicza wartość wyjścia układu na podstawie równania różnicowego, wartość uchybu, a więc różnicę pomiędzy obliczoną wartością, a wartością zadaną (od chwili $T = 14$ jest to stała jedynka) oraz wartość sygnału sterującego, wyznaczoną na podstawie wartości uchybu regulacji w chwili bieżącej k oraz chwilach poprzednich $k - 1$, $k - 2$. Wyznaczone wartości są nanoszone na wykres. Symulacja zaimplementowana w ten sposób dała widoczne poniżej efekty (Rysunek 12).

Rysunek 12: Odpowiedź skokowa układu z nastrojonym regulatorem PID.



5.1.2 Wnioski

Z opóźnieniem równym największemu opóźnieniu pochodzącemu z równania różnicowego równym 12, wyjście układu zaczęło nadążać za wyjściem zadaniem. Wyjście układu wyraźnie fluktuuje wokół wartości zadanej, zaś stabilizacja wymaga dość dużo czasu. Działanie regulatora dało poprawny rezultat.

5.2 Realizacja dyskretnego regulatora DMC w wersji analitycznej bez ograniczeń

5.2.1 Wstęp

Algorytm DMC (Dynamic Matrix Control) jest algorytmem predykcyjnym, a więc uwzględniającym nie tylko historyczne dane, ale też prognozowaną przyszłość. Charakteryzuje się możliwością ograniczenia sygnałów wejściowych i wyjściowych procesu (w niniejszym artykule taka wersja algorytmu nie będzie implementowana), efektywnością regulacji procesów wielowymiarowych, uwzględnianiem wpływu zakłóceń na proces. Korzysta on z zasady przesuwanego horyzontu (zasada sterowania repetycyjnego), a więc dla każdej chwili działania algorytmu, oblicza on na podstawie aktualnego wyjścia i poprzednich wejść (a także prognoz tego sygnału) sekwencję przyszłych wartości sygnału sterującego. W niniejszym artykule, implementowany będzie algorytm DMC ze stałą trajektorią zadaną na całym horyzoncie predykcji (co ma sens w kontekście stałej wartości zadanej, potrzebnej w wielu procesach przemysłowych), toteż prawo

regulacji (21) prezentowanego regulatora będzie miało postać:

$$\Delta u(k) = \Delta u(k|k) = \sum_{p=1}^N k_{1,p}(y^{zad}(k) - y(k)) - \sum_{j=1}^{D-1} k_j^u \Delta u(k-j) \quad (21)$$

Funkcja celu algorytmu przyjmie jeden współczynnik dostrajany λ i będzie miała postać:

$$J(k) = \sum_{p=1}^N (y^{zad}(k+p|k) - \tilde{y}(k+p|k))^2 + \lambda \sum_{p=0}^{N_u} (\Delta u(k+p|k))^2 \quad (22)$$

W praktyce nie korzysta się $\lambda = 0$, gdyż skutkuje to dużymi amplitudami i przyrostami przebiegu sygnału sterującego. Wskazaną funkcję celu algorytm będzie miał za zadanie minimalizować.

5.2.2 Oznaczenia przyjęte w skrypcie

$$M^P = \begin{bmatrix} s_2 - s_1 & s_3 - s_2 & \dots & s_D - s_{D-1} \\ s_3 - s_1 & s_4 - s_2 & \dots & s_{D+1} - s_{D-1} \\ \dots & \dots & \dots & \dots \\ s_{N+1} - s_1 & s_{N+2} - s_2 & \dots & s_{N+D-1} - s_{D-1} \end{bmatrix} \quad (23)$$

$$\Delta U^P(k) = \begin{bmatrix} \Delta u(k-1) \\ \dots \\ \Delta u(k-(D-1)) \end{bmatrix} \quad (24)$$

$$M = \begin{bmatrix} s_1 & 0 & \dots & 0 \\ s_2 & s_1 & \dots & 0 \\ \dots & \dots & \dots & \dots \\ s_N & s_{N-1} & \dots & s_{N-N_u+1} \end{bmatrix} \quad (25)$$

- (23) - Macierz predykcji wyjścia (oznaczenie w kodzie: Mp) - zawiera różnice rzędnych
- (24) - Wektor przeszłych przyrostów sterowania (oznaczenie w kodzie: $dupk$)
- (25) - Macierz dynamiczna zawierająca rzędne (oznaczenie w kodzie: M)
- rzędne - wektor wartości odpowiedzi skokowej obiektu dla poszczególnych kwantów czasu (oznaczenie w kodzie: s)
- $\Delta U(k)$ - wektor optymalnych przyrostów sygnału sterującego na horyzoncie sterowania (oznaczenie w kodzie: duk)

- D - horyzont dynamiki - chwila czasu, w której odpowiedź skokowa obiektu stabilizuje się na wartości wzmocnienia statycznego
- N - horyzont predykcji - mówi o tym, ile chwil do przodu regulator przewiduje wyjście układu
- N_u - horyzont sterowania - mówi o tym, ile chwil do przodu regulator przewiduje wartości zmiany sterowania
- λ - współczynnik, który występuje w funkcji celu (będący przedmiotem optymalizacji); inaczej: współczynnik kary (oznaczenie w kodzie: *lambda*)

5.2.3 Implementacja

Skrypt implementujący regulator DMC wykonuje następujące działania:

- Definicja parametrów używanych w skrypcie.
- Przygotowanie macierzy M , M_p oraz sygnałów u , y_{mod} , y_{zad} .
- Wyznaczanie macierzy K .
- Symulacja algorytmu.
- Rysowanie wykresów.

Symulacja algorytmu została zaimplementowana w następujący sposób:

Rysunek 13: Symulacja DMC.

```
%%                               Symulacja algorytmu DMC
for k = 13:simtime
    ymod(k) = -a1*ymod(k-1)-a0*ymod(k-2)+b1*u(k-11)+b0*u(k-12);
    for j = D-1:-1:2
        dupk(j) = dupk(j-1);
    end
    e(k) = yzad(k)-ymod(k);
    duk = Ke*e(k)-Ku*dupk';
    dupk(1) = duk;
    u(k) = u(k-1)+duk;
end
```

Odpowiedź skokowa obiektu na potrzeby implementacji DMC została wyznaczona na podstawie równania różnicowego (11). Dobór nastaw oraz działanie algorytmu zostało przedstawione w kolejnym rozdziale.

6 Dobieranie parametrów algorytmu DMC

6.1 Horyzont dynamiki D

Początkowa wartość horyzontu dynamiki powinna być równa ilości próbek, która jest potrzebna, by odpowiedź skokowa obiektu ustabilizowała się na wartości wzmocnienia statycznego (w omawianym przykładzie jest to wartość (6) $K_{stat} = 4.545$). Rysunek 14 przedstawia funkcję, która pobiera wektor kolejnych wartości odpowiedzi skokowej obiektu, zaś zwraca ilość próbek potrzebnych, by odpowiedź skokowa ustabilizowała się. Wymagana różnica co do wartości kolejnych próbek została ustalona na 0.0001 co odpowiada fluktuacjom mniejszym niż 0.01 % wartości K_{stat} , co zostało ocenione jako dostatecznie małe wahanie, aby uznać odpowiedź skokową za stabilną. Pierwszą próbką, która spełnia zadane wymagania, jest próbka nr 95.

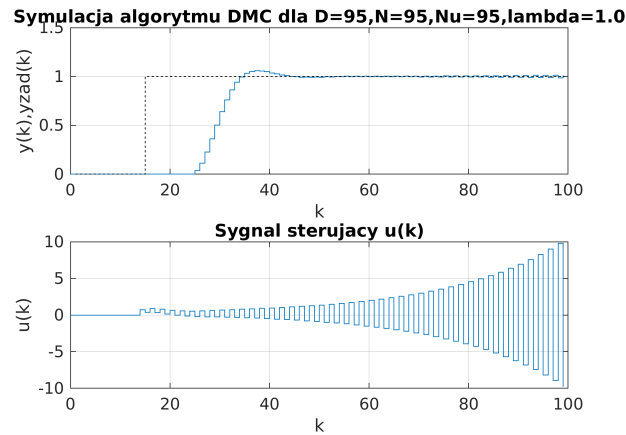
Rysunek 14: Funkcja zwracająca ilość próbek.

```
function [point] = stablePoint(s)
    differenceRequired = 0.0001;
    for i = 1:length(s)-1
        if s(i+1) < 1
            continue;
        end

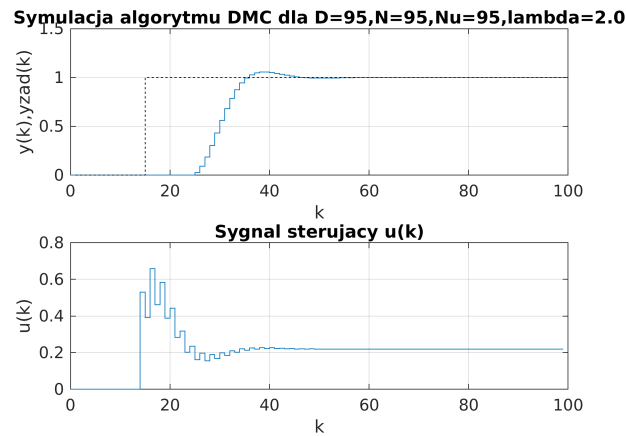
        dif = s(i+1)-s(i);
        display(dif);
        if dif < differenceRequired
            point = i;
            break;
        end
    end
end
```

Można zatem przyjąć początkową wartość horyzontu dynamiki $D = 95$. Następnie, założono początkową wartość współczynnika $\lambda = 1$, a także długości horyzontów predykcji i sterowania takie same, jak horyzontu dynamiki ($N = N_u = D$).

Rysunek 15: Działanie układu.



Widoczne jest działanie regulatora. Sygnal sterujący nie jest stabilny, to też zwiększono wartość współczynnika λ , aby nieco "zdławić" możliwość zmiany wartości sygnału sterującego.

Rysunek 16: Działanie układu - skorygowana wartość $\lambda = 2$.

W tym przypadku zarówno wyjście układu jak i sygnał sterujący wykazują cechy pożądane - oba stabilizują się na odpowiednich wartościach. W związku z tym, horyzont dynamiki D pozostaje bez zmian (w przeciwnym razie, gdyby regulator nie spełniał swojego zadania, należałoby ten horyzont zwiększyć).

6.2 Horyzont predykcji N

Horyzont predykcji wyznaczamy stopniowo go skracając, poczynawszy od wartości $N = D$. Horyzont sterowania w każdym z przypadków winien być równy co do wartości horyzontowi predykcji. Celem badań jest znalezienie takiej wartości N , aby osiągnąć możliwie jak najlepszy rezultat regulacji. Przez dobry rezultat regulacji, rozumie się minimalizację funkcji celu (22), a także możliwie krótki czas regulacji. W tym celu, napisano dwie funkcje, które mają za zadanie zwracać wartości rzeczywiste J oraz *regulationTime*, które mówią odpowiednio o sumie błędów regulacji oraz o czasie regulacji. Najlepsze nastawy horyzontu predykcji, będą zatem charakteryzować się możliwie małą wartością obu tych parametrów.

Rysunek 17: Implementacja funkcji sprawdzających czas i jakość regulacji.

```
function [point] = stablePoint(s)
    differenceReq = 0.01;
    lenReq = 5;
    result = 0;
    for i = 26:length(s)-lenReq
        for j = 0:lenReq-1
            x = s(i+j)
            if abs(x) < differenceReq
                result = result + 1;
            end
        end
        if result == lenReq
            point = i;
            break;
        end
        result = 0;
    end
end

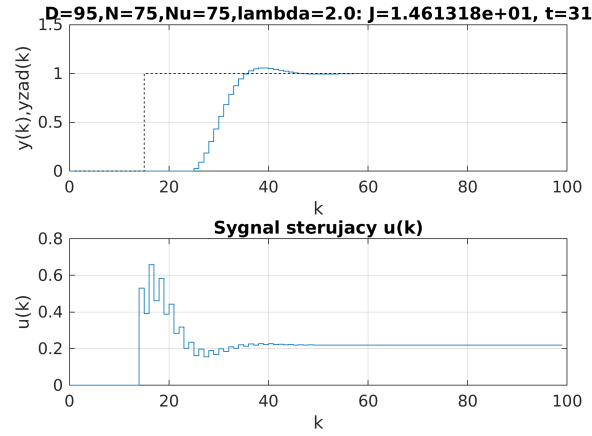
function [J] = quality(J, k, yzad, ymod)
    J = J+(yzad(k)-ymod(k))^2;
end
```

Za moment ustabilizowania się odpowiedzi skokowej układu przyjmuje się próbkę, poczynawszy od której 5 kolejnych próbek co do wartości nie przekracza 1% wartości zadanej $y^{zad} = 1$ (algorytm sprawdza tylko te próbki, dla których nastąpiło już zarówno wyzwolenie wartości zadanej, jak i działanie układu, mającego swoje opóźnienie).

6.2.1 Zdefiniowanie zakresu szukania optymalnej wartości

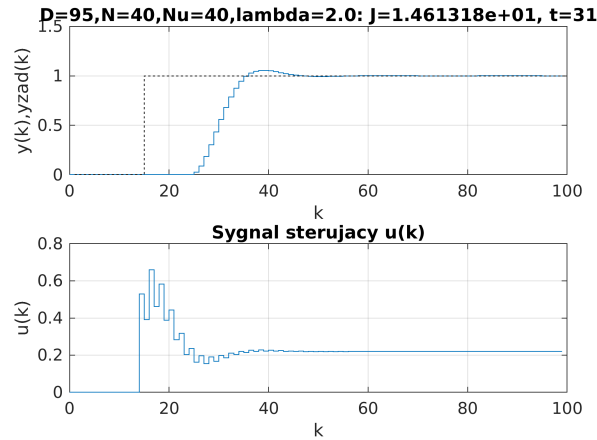
W celu znalezienia zakresu, w którym znajduje się optymalna wartość parametru N , przeprowadzono kilka symulacji. Każda z nich opatrzona jest odpowiednim opisem zarówno nastaw regulatora, jak i wynikiem pary zmiennych J oraz t , definiujących jakość regulacji.

Rysunek 18: Wyniki dla $N = 75$.

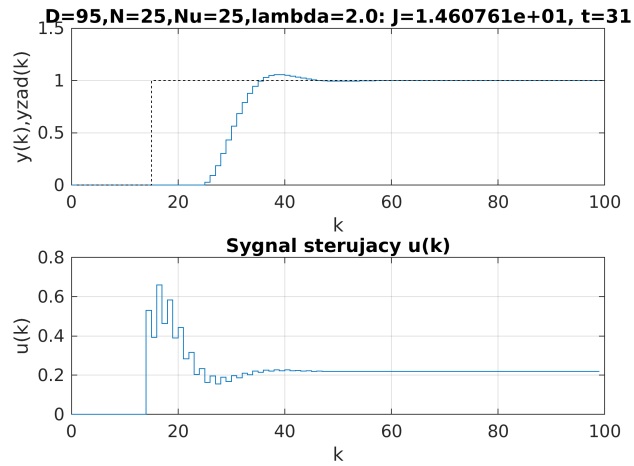


Oba przebiegi wyglądają tak samo jak dla maksymalnego $N = 95$. Brak poprawy jakości regulacji. (Rys. 18)

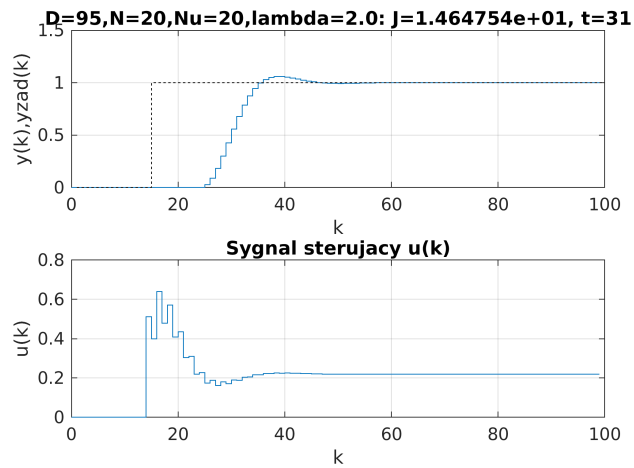
Rysunek 19: Wyniki dla $N = 40$.



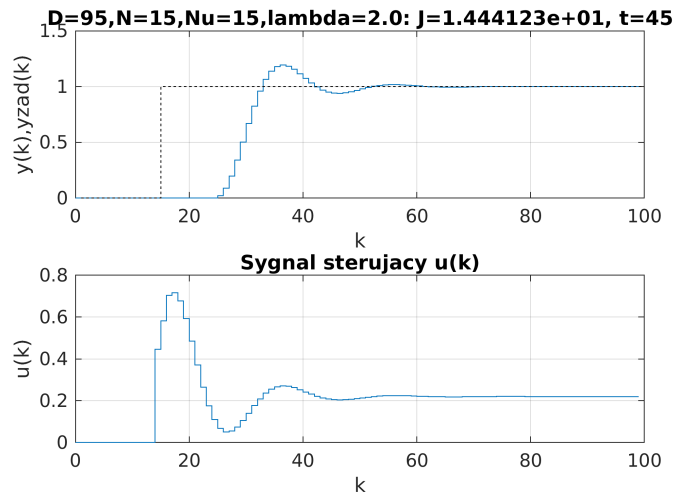
Nie odnotowano żadnej zmiany zarówno w kształcie przebiegów jak i parametrach jakości. (Rys. 19.)

Rysunek 20: Wyniki dla $N = 25$.

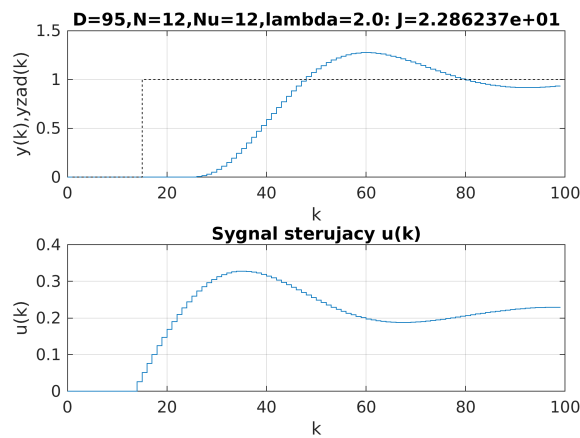
Choć charakter przebiegów pozostał ten sam, nastąpiła niewielka zmiana parametru J . Sugeruje to właściwy kierunek zmiany horyzontu predykcji. (Rys. 20)

Rysunek 21: Wyniki dla $N = 20$.

Błąd nieznacznie wzrósł, przebiegi pozostają takie same. (Rys. 21)

Rysunek 22: Wyniki dla $N = 15$.

Parametry jakości regulacji uległy znacznej zmianie: suma błędów regulacji wykazała poprawę, natomiast czas regulacji uległ znacznemu wydłużeniu (Rys. 22). Przebieg sygnału sterującego nie ma już tak gwałtownego charakteru. Zmiany sugerują, iż optimum dla tego układu znajduje się w zakresie 15 – 20, lecz gwoli precyzji pokazano jeszcze jeden przebieg dla krótszego horyzontu predykcji.

Rysunek 23: Wyniki dla $N = 12$.

Odpowiedź układu nie zdążyła się ustabilizować w wymaganym czasie symulacji równym 100 sekund. Sygnał sterujący również fluktuuje. W celu wygenerowa-

nia tych przebiegów, musiały zostać wyłączone wymagania co do czasu regulacji, stąd brak tego parametru na prezentowanym schemacie. Wskaźnik jakości regulacji J uległ znacznemu powiększeniu. Regulator nie działa prawidłowo.

6.2.2 Wyznaczenie optymalnej wartości za pomocą skryptu

W celu wyłonienia takiej wartości horyzontu predykcji N , dla której para (J, t) znajdzie swoje optimum, napisano skrypt, który sprawdza kolejne wartości wskazanych wektorów oraz oblicza moduł każdej z par. Jak łatwo zauważyć, parametr J ma dużo mniejsze wahania co do wartości niż parametr t , toteż wprowadzono odpowiednią wagę, mającą na celu zniwelowanie różnic w dziedzinie obu wartości.

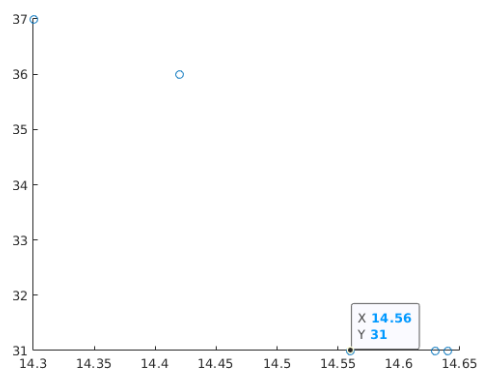
Rysunek 24: Funkcja szukająca optimum pary (J, t) z wagą.

```
J = [14.3 14.42 14.56 14.63 14.64 14.64];
t = [37 36 31 31 31 31];
wage = 30;

scatter(J, t);
w = checkBest(J, t, wage);
print('screeny/5bscatter.png', '-dpng', '-r400');
display(w)

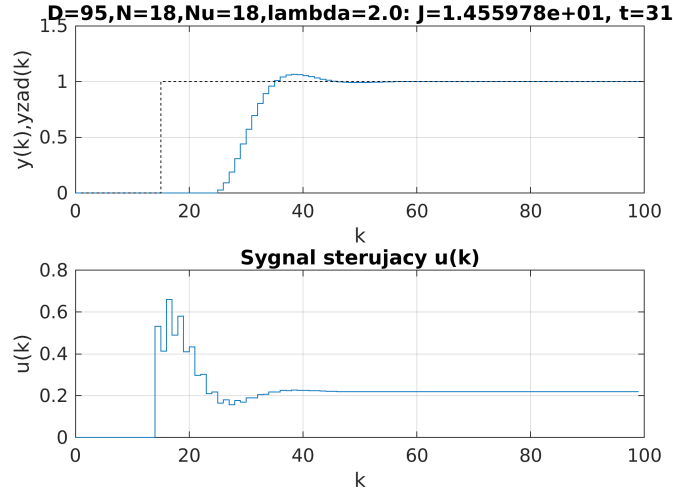
function [w] = checkBest(J, t, wage)
w = [0 0];
c = 10000;
for i = 1:length(J)
    squaresSum = wage*J(i)^2 + t(i)^2;
    module = sqrt(squaresSum);
    if module < c
        c = module;
        w(1) = J(i);
        w(2) = t(i);
    end
end
end
```

Rysunek 25: Położenie wartości parametrów na płaszczyźnie.



Najbardziej korzystnym horyzontem predykcji jest wartość $N = 18$, odpowiadająca parze $(14.56, 31)$.

Rysunek 26: Wyniki dla $N = 18$.



6.2.3 Wnioski

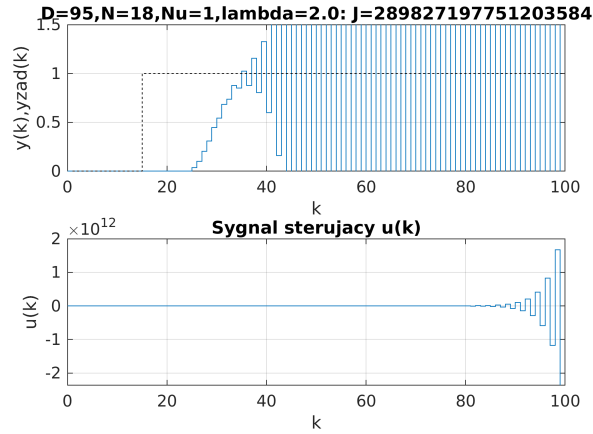
Horyzont predykcji N zdaje się mieć decydujący wpływ głównie na szybkość czasu regulacji i to w ograniczonym zakresie. Logika podpowiada, iż im dalej w przyszłość sięgają prognozy, tym lepiej da się dopasować sygnał sterujący, stąd widoczne pogorszenie jakości regulacji dla zbyt krótkiego horyzontu. Stąd też brak większych zmian w działaniu regulatora dla odpowiednio długich horyzontów.

6.3 Horyzont sterowania N_u

W celu dobrania odpowiedniej wartości horyzontu sterowania N_u oraz zbadania jego wpływu na pracę regulatora, zbadano jakość regulacji dla nastaw $D = 95, N = 18, N_u = 1, 2, 3, 4, 5, \dots, N$.

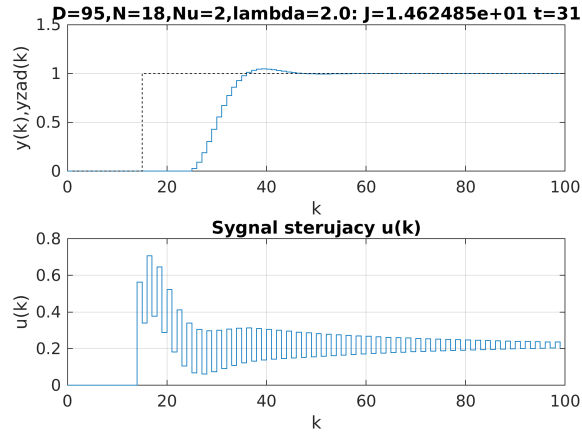
6.3.1 Badanie wpływu horyzontu sterowania na proces regulacji

Rysunek 27: Wyniki dla $N_u = 1$.

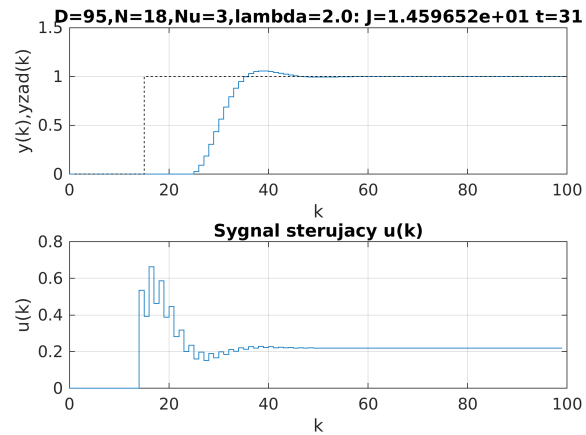


Układ jest niestabilny, nie działa prawidłowo. Nie ma mowy o regulacji obiektu. (Rys. 27.)

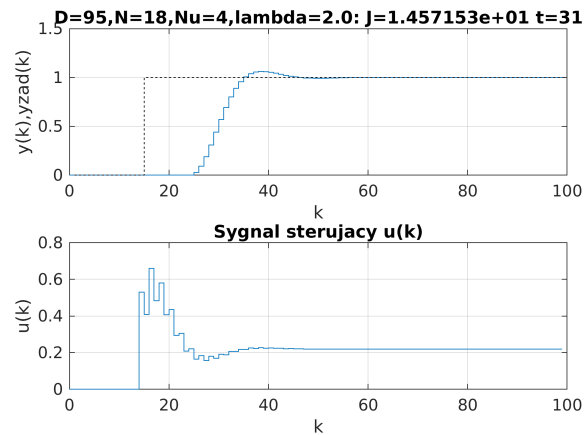
Rysunek 28: Wyniki dla $N_u = 2$.



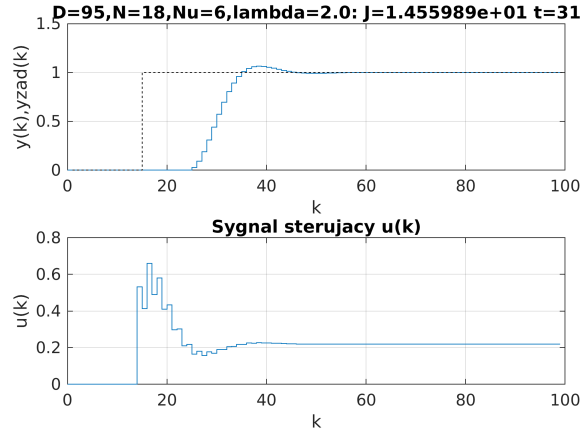
Regulator zaczął poprawnie pracować. Układ stabilizuje się w dobrym czasie, zaś suma błędów regulacji jest normalnej wielkości. Sygnał sterujący jest bardzo gwałtowny i, choć zbiega do wartości 0.2, nie jest on dobrej jakości. Wykres ten sprawia wrażenie, iż regulator zna kierunek zmian wyjścia, lecz nie ma informacji o ich wielkości - stąd gwałtowne poprawki w każdym kroku T . (Rys. 28)

Rysunek 29: Wyniki dla $N_u = 3$.

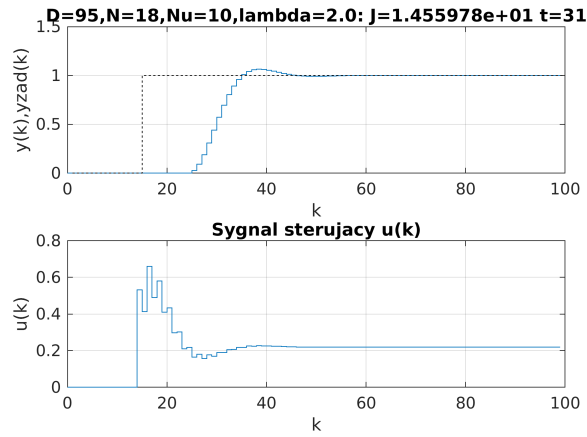
Jakość J uległa poprawie, zaś sam sygnał sterujący już nie wykonuje tak gwałtownych zmian. Są one na akceptowalnym poziomie. (Rys. 29)

Rysunek 30: Wyniki dla $N_u = 4$.

Jedyną zauważalną zmianą jest niższy parametr J - minimalizacja postępuje. Dla wszystkich dotychczasowych wartości N_u , czas regulacji pozostaje ten sam. (Rys. 30)

Rysunek 31: Wyniki dla $N_u = 6$.

Malejące wartości J zachowują kierunek i tempo zmian. Przebiegi zarówno wyjścia układu jak i sygnału sterującego nie ulegają widocznym zmianom. Para (J, t) ma bardzo zbliżone wartości, co w przypadku nastaw $N = 18, N_u = 18$, toteż spodziewany jest brak dalszych zmian obu parametrów wraz ze wzrostem długości horyzontu sterowania. (Rys. 31)

Rysunek 32: Wyniki dla $N_u = 10$.

Nie zaobserwowano żadnych znaczących zmian. Przebiegi dla $N_u = 10$ oraz $N_u = 18$ są takie same.

6.3.2 Wnioski

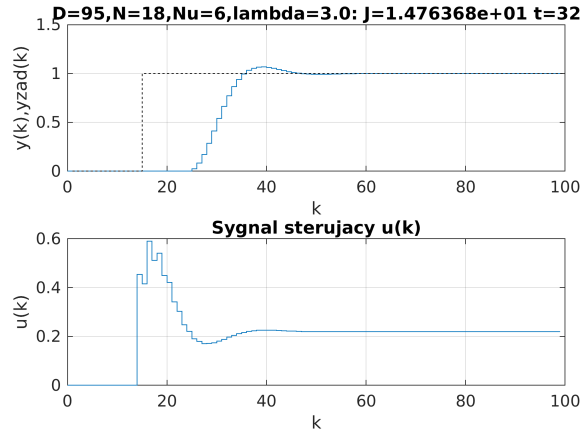
Regulator o zbyt niskim horyzoncie sterowania działa nieprawidłowo lub gwałtownie steruje obiektem. Wystarczył natomiast stosunkowo krótki horyzont $N_u = 3$, aby sterowanie obiektem przyniosło pożądane efekty. Dalsze wydłużanie go przyniosło niewielkie zmiany jeszcze tylko do wartości $N_u = 6$, zaś potem zmiany były kosmetyczne lub żadne. Z racji potrzeby minimalizowania funkcji celu J , a także dobrania możliwie małej wartości horyzontu sterowania, zdecydowano się na wartość $N_u = 6$. Do tej pory regulacja odbywa się szybko i sprawnie, natomiast problemem pozostaje wciąż relatywnie gwałtowny sygnał sterujący $u(k)$.

6.4 Współczynnik dostrajany λ

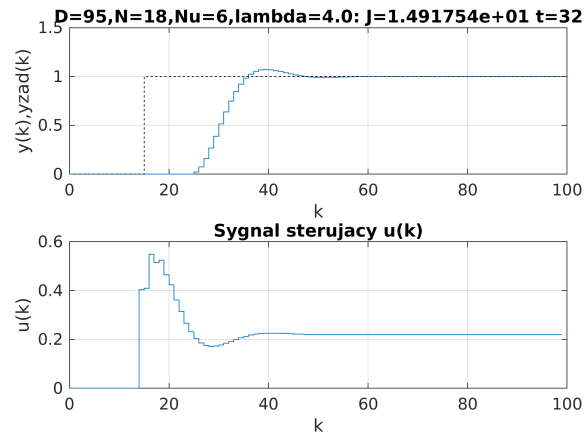
6.4.1 Badanie wpływu współczynnika λ

Jest to współczynnik kary za gwałtowność zmian sygnału sterowania. Powoduje on płynniejsze sterowanie obiektem kosztem czasu regulacji, a w konsekwencji również i jakością dopasowania J . Zbadano wpływ tego parametru na układ dla następujących wartości (zaczęto od wartości 3 i większych).

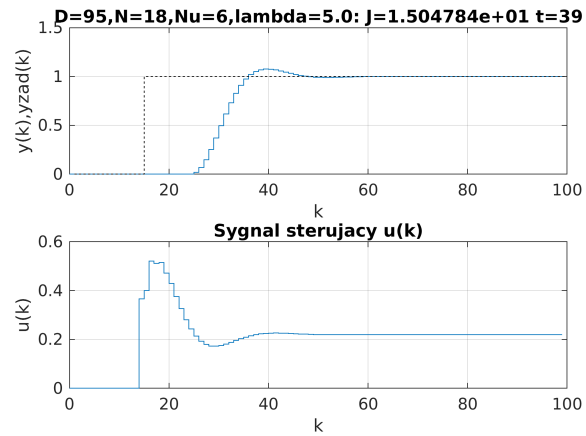
Rysunek 33: Wyniki dla $\lambda = 3$.



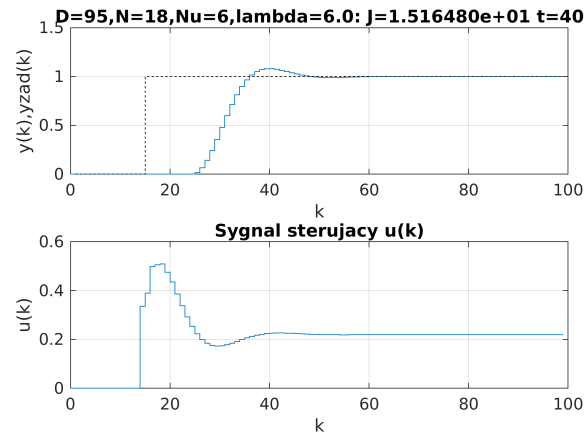
Widać wyraźnie poprawę płynności sterowania względem Rysunku 32. Piki $u(k)$ nie są już tak mocne. Nieznacznie wzrasta czas trwania regulacji. (Rys. 33)

Rysunek 34: Wyniki dla $\lambda = 4$.

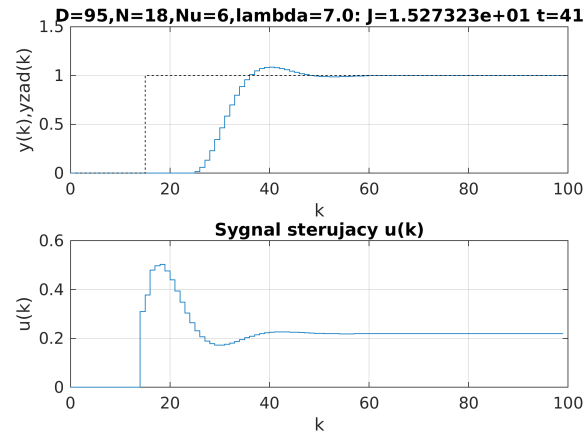
W dalszym ciągu różnice pomiędzy kolejnymi wartościami sygnału $u(k)$ się zmniejszają. Zaczyna on przypominać łagodny sygnał schodkowy reprezentowany przez $y(k)$. (Rys. 34)

Rysunek 35: Wyniki dla $\lambda = 5$.

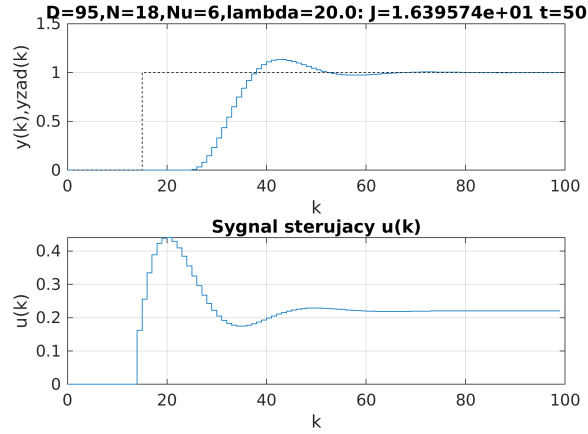
Tendencja łagodzenia kolejnych kroków jest kontynuowana. Wzrósł bardzo znacznie czas trwania regulacji do 39 sekund. (Rys. 35)

Rysunek 36: Wyniki dla $\lambda = 6$.

Sygnal sterujący niewiele różni się od poprzedniej nastawy. Czas regulacji wzrasta. (Rys. 36)

Rysunek 37: Wyniki dla $\lambda = 7$.

Nie zaobserwowano znaczących zmian.

Rysunek 38: Wyniki dla $\lambda = 20$.

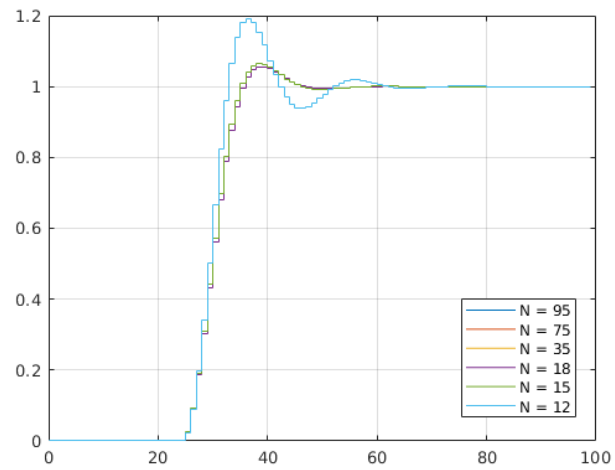
Dalsze zwiększanie tego parametru daje niewielkie korzyści, a straty w postaci długiego czasu regulacji są znaczne.

6.4.2 Wnioski

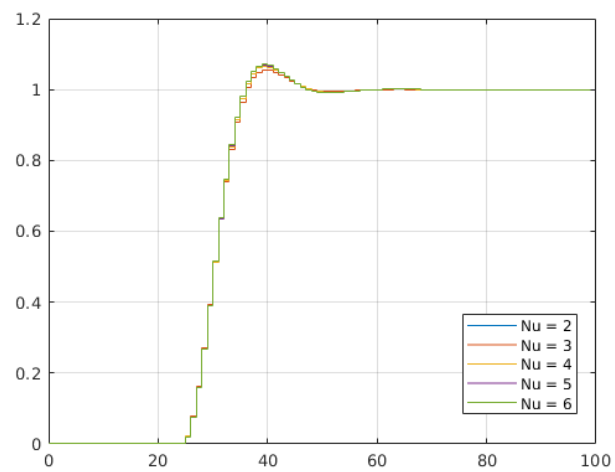
Parametr ten jest bardzo użyteczny, gdyż jako jedyny z prezentowanych parametrów ma tak znaczący wpływ na sygnał sterujący, który ma tendencje do gwałtownych zmian. Należy wybierać małe wartości tego parametru, gdyż największe korzyści w postaci wygładzenia $u(k)$ i najmniejsze straty w postaci wydłużenia czasu regulacji, przynosi on w zakresie 2–4 (zależnie od rozpatrywanego układu, ten zakres może się oczywiście różnić). Chcąc zachować kompromis pomiędzy postacią sygnału sterującego, a szybkością regulacji, zdecydowano się dobrać współczynnik $\lambda = 4$, ponieważ jest to największa jego wartość przed znacznym wzrostem czasu regulacji.

6.5 Podsumowanie

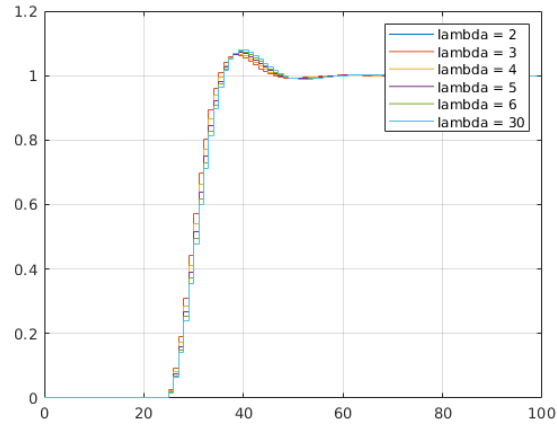
6.5.1 Wpływ zmian horyzontu predykcji N na odpowiedź skokową



6.5.2 Wpływ zmian horyzontu sterowania N_u na odpowiedź skokową



6.5.3 Wpływ zmian λ na odpowiedź skokową



6.5.4 Końcowa wersja regulatora DMC

Regulator ostatecznie otrzymał nastawy o wartościach

$$D = 95 \quad (26)$$

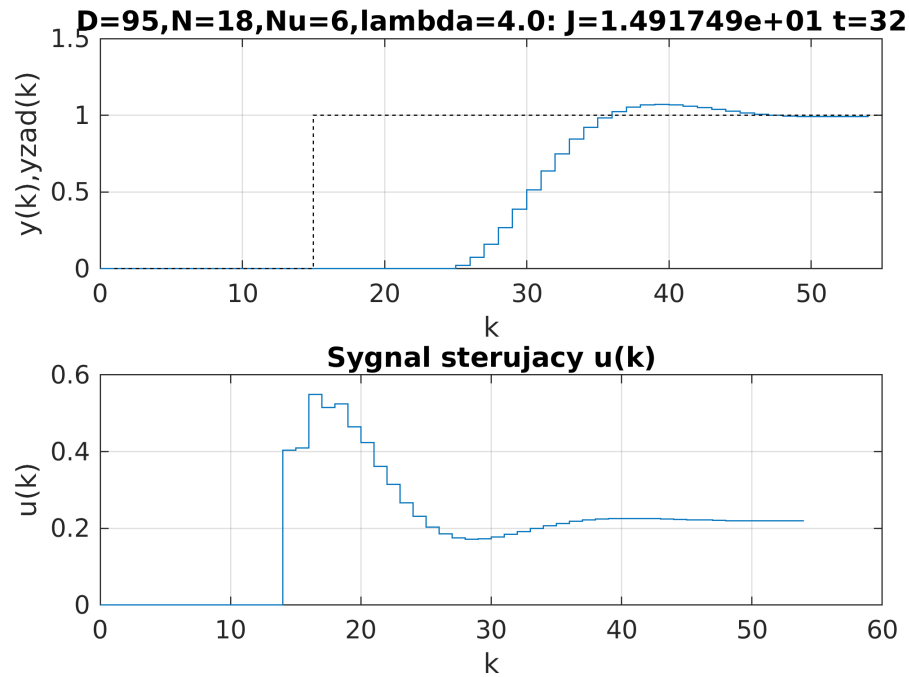
$$N = 18 \quad (27)$$

$$N_u = 6 \quad (28)$$

$$\lambda = 4 \quad (29)$$

... z wynikiem $J = 14.917$, czasem regulacji $t = 32s$. Przebiegi wyjścia układu oraz sygnału sterującego zaprezentowano na Rysunku 39.

Rysunek 39: Ostateczna forma regulatora DMC.

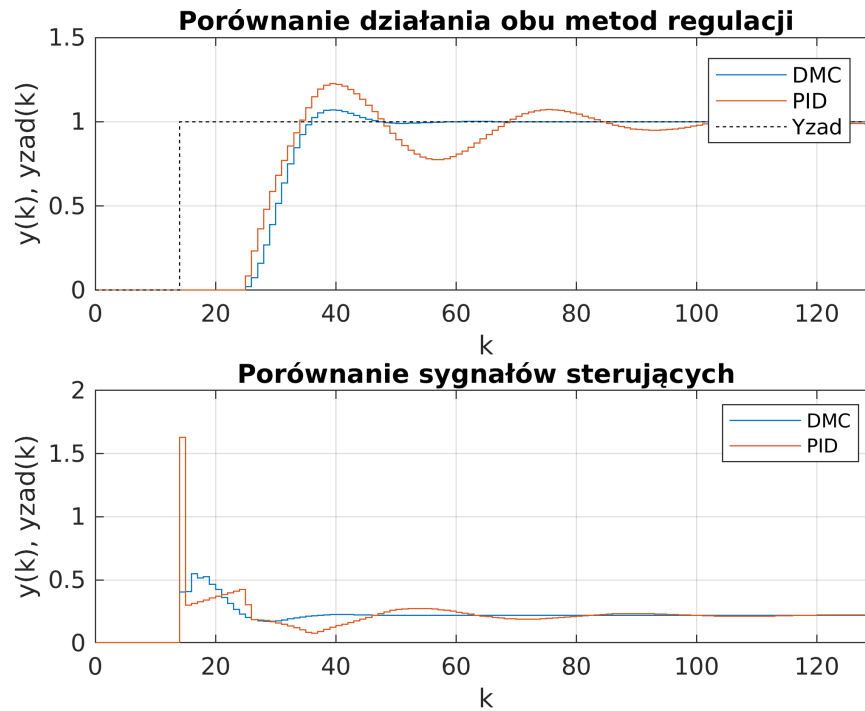


7 Porównanie jakości obu metod regulacji

7.1 Porównanie parametrów jakości

W celu porównania efektów regulacji obu regulatorów, napisano nowy skrypt, który stanowi połączenie skryptów zastosowanych w rozdziałach 5 i 6. Oba regulatory są już dostrojone, a więc warunki regulacji są najlepsze z możliwych. Przebiegi wyjść układów oraz sygnałów sterujących przedstawiono poniżej (Rysunek 40).

Rysunek 40: Porównanie regulatorów.



Zauważono znaczną różnicę pomiędzy zastosowanymi algorytmami, polegającą zarówno na szybkości wygaszania oscylacji jak również gwałtowności skoków sygnału sterującego. Obiekt regulowany algorytmem DMC osiąga de facto stabilność w momencie, gdy algorytm PID stara się uporać z pierwszym przeregulowaniem. Parametry jakości J oraz $simulationTime$ (Wzór 22 oraz Rysunek 14) dla obu algorytmów są następujące:

	PID	DMC
J	14.9177	14.9175
$regulationTime$ [s]	103	32

Wątpliwość budzi tak mała różnica co do wartości sumy błędów regulacji - regulator DMC wypadł lepiej tylko o 0.0002, co, biorąc pod uwagę tak różne trajektorie przebiegów, wydaje się być nieprawdłowe. Trzymając się jednak samych przebiegów, bez wątpienia można stwierdzić, iż regulator DMC jest szybciej zbieżny do wartości zadanej układu. Czas regulacji wyniósł w tym przypadku 32 sekundy, co wobec 103 sekund dla algorytmu PID jest wynikiem świetnym.

7.2 Wyznaczanie obszarów stabilności

W tej sekcji, manipulowano parametrami T_0 oraz K_0 . Stopniowo zwiększając okres opóźnienia, badano dla jakich wartości wzmocnienia K_0 , układ znajduje się na granicy stabilności (pożądane przebiegi były identyczne z tymi, które należało osiągnąć, szukając wzmocnienia krytycznego K_k w metodzie Zieglera-Nicholsa). Zważywszy na to, że układ musiał odpowiadać stałymi, niegasnącymi oscylacjami, zwiększono czas symulacji do 200 sekund.

Celem owych działań jest znalezienie kolejnych wartości ilorazów wyznaczonych parametrów oraz parametrów nominalnych, będących nastawami układu, dla których wyznaczono parametry regulatorów

$$T_{nom} = 5, K_{nom} = 4.6 \quad (30)$$

... dzięki czemu będzie można wyznaczyć krzywą $\frac{K_0}{K_{nom}}$ w funkcji $\frac{T_0}{T_{nom}}$, stanowiącą granicę na płaszczyźnie, pomiędzy obszarem stabilności regulatora, a obszarem niestabilności. W obu przypadkach, wektor T_0 , a więc wektor wartości czasu opóźnienia, był równy:

$$T_0 = \begin{bmatrix} 1 \\ 1.1 \\ 1.2 \\ 1.3 \\ 1.4 \\ 1.5 \\ 1.6 \\ 1.7 \\ 1.8 \\ 1.9 \\ 2 \end{bmatrix} T_{nom} \quad (31)$$

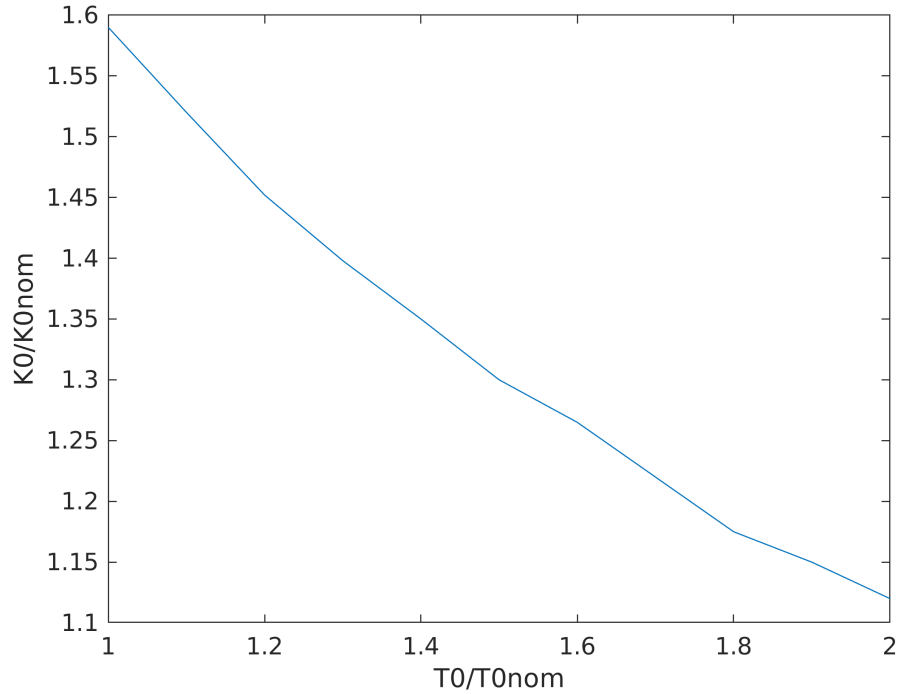
7.2.1 Algorytm PID

Dla elementów wektora T_0 , znaleziono odpowiadające im elementy wektora K_0 , które stanowią kolejne wartości wzmocnienia, dla których układ z regulatorem PID znajduje się na granicy stabilności.

$$K_0 = \begin{bmatrix} 1.59 \\ 1.52 \\ 1.452 \\ 1.398 \\ 1.35 \\ 1.3 \\ 1.265 \\ 1.22 \\ 1.175 \\ 1.15 \\ 1.12 \end{bmatrix} K_{nom} \quad (32)$$

Osiągnięto to dzięki niewielkiej modyfikacji skryptu wyznaczającego parametry dyskretnego regulatora PID oraz funkcji *tfdata*, która pozwala na ekstrakcję poszczególnych współczynników transmitancji dyskretnego obiektu. Krzywa $\frac{K_0}{K_0^{nom}}$ w funkcji $\frac{T_0}{T_0^{nom}}$ dla regulatora PID, wygląda następująco:

Rysunek 41: Krzywa stabilności regulatora PID.



Obszar pod krzywą oznacza płaszczyznę parametrów, dla której układ będzie stabilny. W celu stworzenia układu odniesienia przy analizie stabilności układu z regulatorem DMC, obliczono pole obszaru pod prezentowaną krzywą za pomocą funkcji do tego napisanej, która pobiera jako argumenty wyznaczone wektory (31) oraz (32), a następnie oblicza pole trójkąta i prostokąta, których szerokości wyznaczają odcinki poprowadzone od $i - tego$ oraz $i + 1$ punktu na płaszczyźnie do osi X.

Rysunek 42: Funkcja zwracająca pole obszaru stabilności.

```

function [area] = computeStabilityArea(KOnom, TOnom, Kmnoznik, Tmnoznik)
    area = 0;
    Kmnoznik = Kmnoznik * KOnom;
    Tmnoznik = Tmnoznik * TOnom;
    for i = 1:length(Kmnoznik)-1
        %Triangle:
        heightT = Kmnoznik(i) - Kmnoznik(i+1);
        widthT = Tmnoznik(i+1) - Tmnoznik(i);
        triangleArea = widthT*heightT*0.5;
        %Square
        heightS = Kmnoznik(i+1);
        widthS = widthT;
        squareArea = heightS * widthS;
        area = area + triangleArea + squareArea;
    end
end

```

Obszar stabilności dla układu z regulatorem PID wyniósł

$$PIDStabilityArea = 30.3255 \quad (33)$$

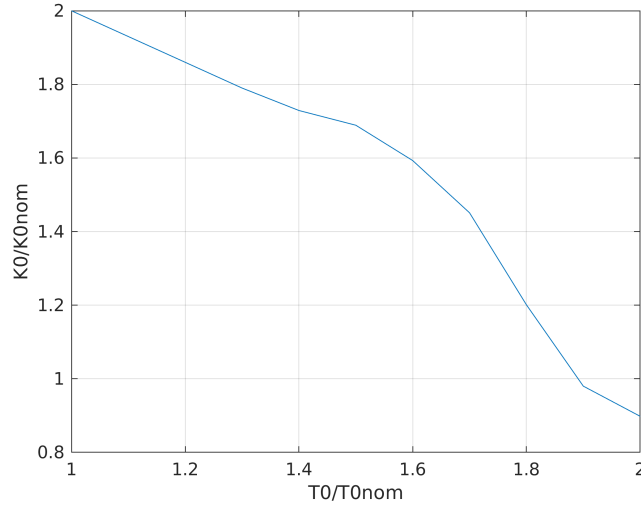
7.2.2 Algorytm DMC

Operację szukania wartości wektora K_0 powtórzono dla regulatora DMC z następującym skutkiem:

$$K_0 = \begin{bmatrix} 2 \\ 1.93 \\ 1.86 \\ 1.79 \\ 1.729 \\ 1.689 \\ 1.593 \\ 1.451 \\ 1.201 \\ 0.98 \\ 0.898 \end{bmatrix} K_{nom} \quad (34)$$

Trajektoria krzywej reprezentującej $\frac{K_0}{K_0^{nom}}$ w funkcji $\frac{T_0}{T_0^{nom}}$ została pokazana na Rysunku 43.

Rysunek 43: Krzywa stabilności regulatora DMC.



Pole obszaru położonego pod krzywą wynosi

$$DMCStabilityArea = 36.0456 \quad (35)$$

Można więc wnioskować, iż w prezentowanej dziedzinie ($100 - 200 \% T_0$), układ z regulatorem DMC oferuje więcej parametrów, dla których zachowa on stabilność. Należy pamiętać, iż fakt znajdowania się danego punktu pod krzywą, nie oznacza jeszcze efektywnej pracy.

7.2.3 Wnioski

Granica stabilności została badana na wydłużonym czasie symulacji, lecz dla każdej wartości wzmocnienia, oscylującego wokół wskazanej wartości granicznej, zarówno w algorytmie PID jak i DMC, możliwe było takie wydłużenie czasu symulacji, aby dostrzegalne było stopniowe obniżanie się przebiegu. Wnioskować można, że:

- Znalezienie dokładnej wartości wzmocnienia K_0 jest w praktyce bardzo trudne lub niemożliwe ze względu na możliwość wydłużania czasu symulacji rzędu godzin oraz dodawania kolejnych wartości po przecinku do wzmocnienia, co koniec końców skutkowałoby zaburzeniem wyników błędami numerycznymi. Inną kwestią pozostaje wątpliwy sens takiego działania.
- Fakt znajdowania się wybranego punktu pod krzywą nie gwarantuje efektywnej pracy - kwestią umowną pozostaje czas, w którym układ regulowany ma zniwelować uchyb do zera. Im dalej od krzywej, tym układ będzie szybciej zbieżny.

8 Podsumowanie

Zaprezentowane algorytmy zostały zbadane pod kątem trudności implementacji, szybkości oraz jakości regulacji, zakresem stabilnej pracy. O ile w warunkach pakietu MATLAB algorytm DMC jest nieco trudniejszy w implementacji, o tyle daje on lepsze rezultaty. Jego główną zaletą i przewagą nad algorytmem PID jest szybkość zbieżności do wartości zadanej. Regulacja zaawansowana pod postacią regulacji predykcyjnej algorytmem Dynamic Matrix Control, okazuje się dziedziną, w której warto poszerzać swoją wiedzę, chcąc efektywnie kontrolować zmiany w środowisku obiektu.