

Wojciech TARNOWSKI

POLITECHNIKA KOSZALIŃSKA, KATEDRA SYSTEMÓW STEROWANIA

Poliptymalizacja procesów nieliniowych**Prof. dr hab. inż. Wojciech TARNOWSKI**

Ukończył studia na Wydz. Mechaniki Precyzyjnej PW, doktorat na Wydz. Automatyki P. Śląskiej, habilitacja na Wydz. SiMR PW, profesura w 1990r.. W latach 1963-86 pracował na Wydz. Automatyki P. Śląskiej, potem na Wydz. Mechanicznym P. Koszalińskiej, gdzie utworzył Katedrę Systemów Sterowania. Zajmuje się polioptymalizacją, projektowaniem układów automatyki przemysłowej i komputerowym wspomaganie projektowania oraz metodologią procesu projektowo – konstrukcyjnego.



e-mail: wojciech.tarnowski@tu.koszalin.pl

Streszczenie

Sformułowano zadanie jednoczesnego poszukiwania polioptymalnej funkcji sterowania procesu oraz parametrów konstrukcyjnych obiektu, w którym ten proces przebiega. Zaproponowano dyskretyzację zmiennych niezależnych jako sposób zamiany na zadanie polioptymalizacji statycznej, znacznie łatwiejsze do rozwiązania. Podkreślono rolę sposobu dyskretyzacji. Przedstawiono przykład sterowania samochodem, i rozwiązano go za pomocą algorytmu genetycznego. Innym przykładem jest układ sterowania procesem sterowania nieliniowego. Przedstawiono oprogramowanie zadania. Omówiono otrzymane wyniki. Przedyskutowano metodykę postępowania.

Słowa kluczowe: polioptymalizacja procesu; dyskretyzacja; zamiana zadania optymalizacji dynamicznej na zadanie optymalizacji statycznej

Polyoptimization of non-linear processes**Abstract**

An optimization problem of control functions and object parameters simultaneously, is formulated. Discretization of independent variables is to convert an original problem into a static optimization problem, much easier to be solved. A way of discretization is important on the adequacy and accuracy of a solution. Two examples are included: a control of a car, and an automatic control by the bi-stable regulator. Computer codes are given. A general methodology is proposed.

Keywords: polyoptimization, discretization, conversion of dynamic to static optimization problem

1. Wstęp

W praktyce inżynierskiej i ekonomicznej polioptymalizacja ciągle nie „trafiła pod strzechy”. Przyczyn jest wiele, prawdopodobnie główna to ta, że potrzebny jest duży wysiłek, a z trudem uzyskany rezultat pozostaje wątpliwy ze względu na dyskusyjność zastosowanych kryteriów i ograniczeń i niepewność modelu matematycznego.

Typowym źródłem rezerwy praktyków wobec różnych form matematycznej optymalizacji jest to, że optymalizuje się parametry obiektu a nie proces, gdyż to drugie prowadzi do sformułowania bardzo trudnego zadania optymalizacji dynamicznej, które nie ma inżynierskich metod rozwiązania. Jeśli problem jest ciągły, metodą jest analiza wariancyjna, dla której jednak brak metod ogólnych; natomiast jeśli problem jest dyskretny – pracochłonne metody kombinatoryczne, z różnymi heurystykami.

W tej pracy lansuje się dyskretyzację zmiennych niezależnych i sformułowanie statycznego zadania optymalizacji. Jak pokażemy, takie ujęcie jest w pełni praktyczne i zapewnia wystarczającą dokładność.

2. Terminologia

Dana jest *przestrzeń zmiennych niezależnych* $Z = \{z_1, z_2, z_3, t\}$ gdzie z_i są zmiennymi przestrzennymi a t jest czasem, na przykład

trzy wymiary zbiornika i czas zegarowy. Na tej przestrzeni definiuje się zbiór *wielkości wyjściowych* procesu, $y = \{y_i, i = 1, \dots, n\}$ – to są te wielkości, które charakteryzują przebieg procesu, na przykład temperatura, ciśnienie i trzy składowe prędkości cieczy w zbiorniku. Każda z nich może być funkcją wszystkich lub niektórych zmiennych niezależnych:

$$y_i = f_i(z_1, z_2, z_3, t).$$

Zauważmy, że funkcje f_i są także zależne od parametrów konstrukcyjnych p obiektu oraz od funkcji wymuszeń zewnętrznych $u_j = u_j(z_1, z_2, z_3, t)$

$$y_i = f_i(u_j, p, z_1, z_2, z_3, t) \quad (1)$$

Wymuszeniami nazywamy łącznie zmienne sterujące oraz zakłócenia.

Stan procesu określa zbiór zmiennych wyjściowych, zatem stanem procesu s_0 w danym punkcie przestrzeni i w danej chwili czasu $Z_0 = [z_1, z_2, z_3, t]$ nazwiemy wektor wartości zmiennych wyjściowych:

$$s_0 = s|_{(z_1, z_2, z_3, t)} = [y_1, y_2, y_3, \dots, y_n]. \quad (2)$$

Procesem nazwiemy ciąg stanów obiektu, np. kolejne wartości konkretnej wielkości wyjściowej wzdłuż konkretnej trajektorii w przestrzeni zmiennych niezależnych, np. rozkład temperatury wzdłuż określonej krawędzi zbiornika, albo zmiany temperatury w czasie w konkretnym miejscu zbiornika, albo wysokości punktów geodezyjnych drogi w funkcji współrzędnych geograficznych, albo wysokość poziomu falującej wody itp.

Kryterium cząstkowym nazwiemy wielkość, która będzie charakteryzowała jakość procesu w określonym obszarze przestrzeni zmiennych niezależnych.

Z reguły to będzie funkcjonal określany na zdefiniowanych przedziałach zmiennych niezależnych, na przykład wartość średnia, albo wartość maksymalna itp. Na przykład dla procesu wymiany ciepła może to być temperatura wymiennika, albo jego moc cieplna, albo energia cieplna, dla procesu regulacji to może być przeregulowanie, czas regulacji, zapas stabilności itp.

$$k_i = \int_{z_1 \min}^{z_1 \max} dz_1 \int_{z_2 \min}^{z_2 \max} dz_2 \int_{z_3 \min}^{z_3 \max} dz_3 \int_{t \min}^{t \max} f(\cdot) dt \quad i=1, \dots, r \quad (3)$$

gdzie $f = f(y_i, i = 1, \dots, n, u_j, j = 1, \dots, m, p)$

Liczba kryteriów zwykle nie jest duża ($r = 2 - 4$).

Kryteria cząstkowe są niezależnymi cząstkowymi miarami jakości procesu, na przykład jest to sprawność, niezawodność, efektywność, dokładność itp. [1].

Ograniczenia funkcyjne

$$og_i = og_i(y_i, i = 1, \dots, n, u_j, j = 1, \dots, m, p) \leq 0 \quad i = 1, \dots, g \quad (4)$$

Ograniczenia przedziałowe na zmienne niezależne są zdefiniowane przez dopuszczalne przedziały ich wartości.

Parametry charakteryzują najczęściej obiekt w którym zachodzi proces. Nie zależą od czasu, najczęściej nie zależą także od pozo-

stałych zmiennych niezależnych. One także mogą być zmiennymi decyzyjnymi.

3. Sformułowanie zadania polioptymalizacji

Sformułowanie zadania polioptymalizacji dynamicznej

Należy znaleźć takie funkcje u_i i takie parametry p , aby kryteria cząstkowe miały wartości ekstremalne dla określonych przedziałów zmiennych niezależnych, przy spełnieniu ograniczeń na zmienne wyjściowe oraz na zmienne sterujące (model jest dany w [5]).

Sformułowanie zadania polioptymalizacji statycznej

Ponieważ zadanie optymalizacji dynamicznej jest z reguły trudne do rozwiązania (choćby ze względu na nieliniowość modelu matematycznego oraz równania różniczkowe (nierządki cząstkowe), zawsze nieliniowe, dokonuje się dyskretyzacji zmiennych niezależnych i poszukuje się kolejnych wartości zmiennych sterujących jako skończonych ciągów ich wartości.

4. Przykład sterowania samochodem

1. Obiekt optymalizacji (proces)

Auto przejeżdża z nieruchomego startu do celu odległego o L , po płaskiej poziomej drodze, i tam ma się zatrzymać. Jest napędzane zmienną siłą P i hamowane zmienną siłą H .

2. Zmienne niezależne

Jest to czas t i długość s , ale związane są oczywistą zależnością:

$$s = \int_0^t \dot{s}(t) dt, \text{ zatem jest tylko jedna zmienna niezależna.}$$

3. Warunki początkowe

Dla $t = 0$ $s = 0$ oraz $ds/dt = 0$

4. Szukane funkcje sterujące

Poszukuje się siły napędzającej $P(s)$ i siły hamującej $H(s)$

5. Dane

Maksymalna dopuszczalna siła napędzająca, rozwijana przez silnik lub hamulce $P_{max} = 1500$ N; droga do przejechania od startu do celu $L = 300$ m; masa auta $M = 1000$ kg; współczynnik doskonałości kształtu (oporów powietrza) $c = 0,3$.

6. Kryteria

Kryteria cząstkowe: T – całkowity czas przejazdu;
Praca = $EP + EH$ EP – praca siły P ; EH – praca siły H
Kryterium skalarnie (globalne):

$$F = (1-w)T + w \cdot (EP + EH) \quad w \in \langle 0, 1 \rangle \quad (5)$$

7. Ograniczenia przedziałowe

Siła napędowa nie może przekroczyć wartości maksymalnej:

$$P \in [0, P_{max}] \quad (6)$$

Siła hamująca (ujemna) nie może przekroczyć wartości maksymalnej (założono, że jej maksymalna wartość bezwzględna jest taka sama jak siły napędowej):

$$H \in [-P_{max}, 0] \quad (7)$$

Przejechana droga po czasie T :

$$s|_{t=T} \geq L \quad (8)$$

8. Ograniczenia funkcjonalne (model matematyczny procesu)

W tej wersji przyjęto skrajnie uproszczony model, w tym:

- pominięto siłę tarcia suchego;
- pominięto sprawność silnika i jego nieliniowe charakterystyki, przyjmując jako kryterium pracę 'netto', czyli na kołach, a nie zużycie paliwa;

$$\ddot{s} = \frac{1}{M} (P(s) - c \cdot \dot{s}^2) \cdot \text{sign}(\dot{s}) \quad (9)$$

Prędkość po przejechaniu drogi L na mecie nie powinna być większa niż 0,05 m/s.

Ponieważ można założyć, iż w danej chwili nie będzie przyłożona równocześnie siła P i siła H , zatem można oznaczyć je obie symbolem $P(t)$ (pamiętając, że ujemna wartość oznacza hamowanie), i wówczas łączna praca $EP + EH$ wyrazi się wzorem:

$$EP + EH = \int_0^L |P(s)| \cdot ds = \int_0^T |P(t) \cdot \dot{s}(t)| \cdot dt \quad (10)$$

W modelu pominięto także dynamikę sterowania silnika. Pominięto również ograniczenie na przyspieszenie d^2s/dt^2 (komfort pasażerów i przeciążenia auta).

9. Dyskretyzacja

Kwantujemy zmienną niezależną: długość s , po to aby zadanie optymalizacji **dynamicznej** zamienić na zadanie optymalizacji **stacyjnej**. Dzięki temu teraz zamiast szukania funkcji: siła napędzająca $P(s)$ i siła hamująca $H(s)$ należy znaleźć ich wartości $P(k)$ gdzie $k = 1, 2, \dots, n$, jest numerem odcinka drogi.

Zatem szukane zmienne decyzyjne są: $P(1), P(2), \dots, P(n)$.

Ponieważ problem jest ciągły, konieczna jest aproksymacja funkcji sterującej $P(t)$ lub $P(s)$ między dyskretnymi punktami $k = 1, \dots, n$.

Może być to przykład liniowa:

$$P(s) = \frac{P(k+1) - P(k)}{s(k+1) - s(k)} (s - s(k)) \quad (11)$$

10. Intuicyjne rozwiązanie

Wydaje się, że gdyby kryterium optymalizacji był czas przejazdu T (tzn. dla współczynnika wagi $w = 1$), to przy pominięciu tarcia rozwiązaniem byłby następujący przebieg sił: w przedziale drogi $< 0, L/2 >$ siła $P = P_{max}$, a w przedziale $< L/2, L >$ siła hamowania $H = -P_{max}$.

Natomiast dla b. dużych wartości w wartości sił dążą do zera a czas T do nieskończoności. Gdyby wprowadzić siłę tarcia suchego, siła hamowania powinna być wówczas równa zero.

Te spostrzeżenia można wykorzystać do weryfikacji oprogramowania.

11. Rozwiązanie

Wykorzystano algorytm genetyczny¹. Utworzono następującą funkcję która oblicza dwa kryteria: czas końcowy T oraz łączną pracę siły napędowej i siły hamowania.

¹ Wykorzystano pakiet optymalizacji jedno- i wielokryterialnej, napisany w środowisku MATLAB. Jest dostępny w <http://www.tu-harburg.de/~rtsap>; autorem jest Andrey Popov.

Funkcja obliczania kryteriów:

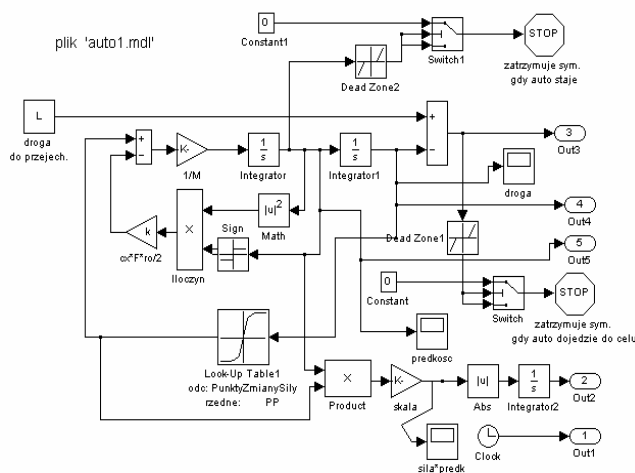
```
function F = fcel_auto_gen(x)
global PP kara yy
PP=x;
[t, xx, yy]= sim('auto1', 200);
% ograniczenia:
[a b]=size(yy);
kara=0;
og(1)= yy(a,5)-0.05; % prędkość
% na mecie nie może być większa od 0.05 m/s
waga=1e3; if og(1) >0 kara=og(1)^2*waga;
else kara=0;
end
og(2)= yy(a,3); % pozostało do przejechania
% auto musi dojechać do mety
waga=3*1e3; if og(2) >0 kara=kara+og(2)^2*waga;
end
F = [];
F(1) = yy(a,1)+kara/100; % czas przejazdu
F(2) = yy(a,2)*1e-4+kara/100; % skalowana praca
```

Plik uruchomieniowy:

```
% plik uruchamia optymalizację skalarną
% procesu
% sterowania autem za pomocą siły
% od silnika i od hamulców
clear all
global PP kara yy %global t yy L PP PunktyZmianySily M k skala
L=300; % odległość do przejechania
PunktyZmianySily=[0 0.2*L 0.4*L 0.6*L 0.8*L L];
M=1000; % masa samochodu
cx=0.3; % wsp. doskon. kształtu
Fp = 2; % pole pow. czołowej samochodu
ro = 1; % gęstość powietrza
k = cx*Fp *ro/2;
skala = 1; % wsp. skali jednostek
Pmax=1500; % max siła rozwijana przez silnik/hamulec
opt = GAopt(-1);
opt.MaxIter = 130;
opt.Select = 1;
opt.Best.Rate = 0.2;
opt.TolX=-1;
n=6;
bounds=[];
granica=[-Pmax Pmax 1e-4];
for i=1:n
    bounds = [bounds; granica];
end
%granica=[-Pmax 0 1e-4];
%%for i=n/2+1:n
    % bounds = [bounds; granica];
%end
[RGenes, RFit, RecGenes, RecFit] = GAMOminBC
('fcel_auto_gen_Pareto', bounds, opt);
PP = RGenes(1,:); % dla 1-go rozwiązania Pareto
figure(2)
[tt,xx,yy]=sim('auto1',100);
subplot(131)
plot(tt,yy(:,4))
```

```
title('droga')
grid
subplot(132)
plot(tt,yy(:,2))
title('praca')
grid
subplot(133)
plot(tt,yy(:,5))
title('predkosc')
grid
figure(3)
plot(RFit(:,1),RFit(:,2),'o');
title('warianty poli-optymalne')
xlabel('kryt.1: czas przejazdu, sek')
ylabel('kryt.2: praca, kJ')
```

Model procesu w Simulinku pokazuje rysunek 1.



Rys. 1. Model procesu przejazdu samochodu w Simulinku; Out1=T,

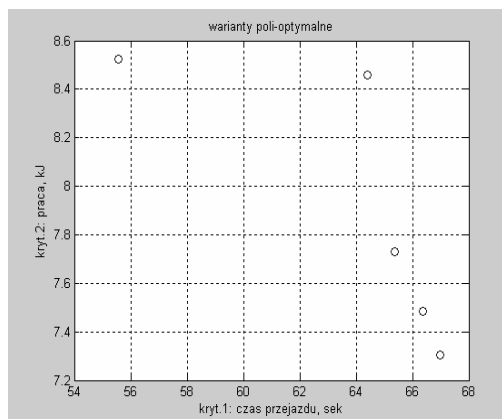
Out2=EP+EH

Fig. 1. Graphical Simulink model of the process of a car run; Out1=T,

Out2=EP+EH

Wyniki polioptymalizacji

Po 130 generacjach otrzymano zbiór wariantów polioptymalnych pokazany na rysunku 2.

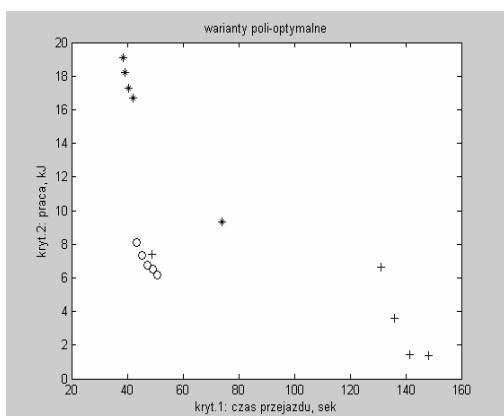


Rys. 2. Warianty polioptymalne znalezione po 130 iteracjach (generacjach)

Fig. 2. Polyoptimal variants set found by the Genetic Algorithm in 130 generations

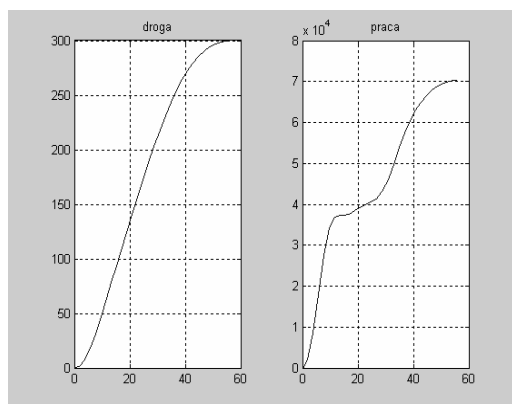
Tab. 1. Przykładowe znalezione optymalne funkcje sterowania $P(i)$, $i = 1, \dots, 6$ oraz wartości kryteriów $F(1)$ i $F(2)$ dla 5 wariantów polioptymalnych
 Tab. 1. Values of criteria and control functions $P(i)$, $i = 1, \dots, 6$ for 5 Polyoptimal variants

Wariant	Czas przejazdu	Skalowana praca	Kolejne wartości siły (zdyskretyzowane wzdłuż drogi)					
			$F(1)$, s	$F(2)$, kJ	$P(1)$	$P(2)$	$P(3)$	$P(4)$
1	66.9799	7.3046	1.1854	0.0552	-0.0535	-0.1335	-0.3566	-0.0648
2	66.3779	7.4835	1.1854	0.0553	-0.0535	-0.1333	-0.3566	-0.0648
3	64.4006	8.4585	1.2554	0.0455	-0.0532	-0.1345	-0.3737	-0.0646
4	65.3676	7.7300	1.2554	0.0445	-0.0532	-0.1346	-0.3737	-0.0646
5	55.5629	8.5260	1.1655	0.0554	-0.0535	-0.0336	-0.3566	-0.2348



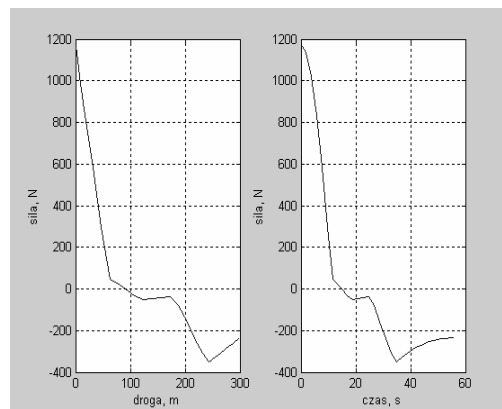
Rys. 3. Warianty poli-optymalne znalezione po 65 iteracjach (generacjach) (+), po 130 iteracjach (*), i po 200 iteracjach (o)
 Fig. 3. Poly-optimal variants set found by the Genetic Algorithm in 65 generations (+), in 130 generations (*) and in 200 generations (o)

Na rysunkach 4a i 4b pokazano rezultaty sterowania dla chromosomu nr 5 (Tablica).



Rys. 4a. Rezultat sterowania genem nr 5 (Tabl. 1) o wartościach $P(i) = 1.1655$ 0.0554 -0.0535 -0.0336 -0.3566 -0.2348
 Fig. 4a. Time characteristics of the control by the chromosom no 5 (Table 1)

Rysunek 4 pokazuje rozkład funkcji sterującej (siła P) w funkcji drogi s i w funkcji czasu t . Tu widać wygładzające działanie interpolacji (opcja *Interpolation-Extrapolation* w bloku Look-Up Table w Simulinku). Użytkownik może wprowadzić inny sposób, np. funkcję Beziera).



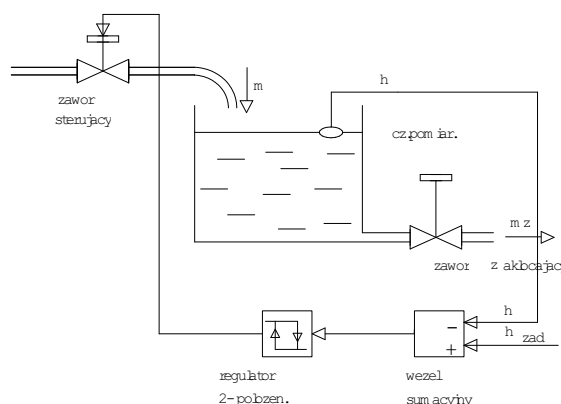
Rys. 4b. Funkcja sterującej (siła P) w funkcji drogi s i w funkcji czasu t (wariant nr 5)
 Fig. 4b. Drive force P as a function of the distance (s) and time (t) for the control by the chromosom no 5 (Table 1)

Uwagi

Każde kolejne uruchomienie algorytmu genetycznego – nawet przy tych samych parametrach – przynosi na ogół nieco inne wyniki, ponieważ jest to algorytm losowy! Natomiast wydłużanie procesu optymalizacji (zwiększanie liczby generacji) przynosi poprawę zbioru Pareto (patrz rysunek 3).

5. Przykład: układ regulacji dwu-położeniowej [2]

Rozważmy silnie nieliniowy układ regulacji pokazany na rysunku 5. Otwarty zbiornik napełniany jest strumieniem cieczy m , który jest sterowany dwupołożeniowym przełącznikowym regulatorem z histerezą. Wielkością regulowaną jest poziom cieczy h w zbiorniku: należy utrzymać stałą wartość $h = h_{zad}$ pomimo wypływu cieczy ze zbiornika (strumień mz).



Rys. 5. Obiekt optymalizacji: proces regulacji poziomu cieczy w zbiorniku za pomocą regulatora dwupołożeniowego: m – strumień napełniania (wielkość sterująca), h – chwilowy poziom cieczy (wielkość sterowana), h_{zad} – sygnał wartości zadanej, mz – strumień wypływu (zakłócenie)
 Fig. 5. Control process: stabilization of the level h in the tank by an on-off controller: m – supply stream (the control signal), h – level (the controlled variable), mz – outlet stream (disturbance)

Zakłóceniami są: ciśnienie cieczy zasilającej $p(t)$ (przed zaworem) i strumień wypływu $mz(t)$.

Z teorii regulacji wiadomo, że regulator dwupołożeniowy powoduje oscylacyjny przebieg procesu regulacji, nawet w stanie ustalonym.

Celem optymalizacji jest znalezienie takich nastaw regulatora (są to tutaj zmienne decyzyjne), aby częstość oscylacji była jak najmniejsza i aby odchylenia regulacji były jak najmniejsze (zatem są to tutaj kryteria optymalizacji).

Zarówno częstość oscylacji jak i odchyłki regulacji zależą od przebiegów czasowych zakłóceń: $mz(t)$ i $p(t)$. Dlatego optymalizację możemy przeprowadzić tylko dla zdeterminowanych przebiegów tych zakłóceń, np. dla stałego ciśnienia zasilania $p(t) = \text{const}$ oraz dla stałego położenia zaworu zakłócającego.

Sformułowanie zadania optymalizacji

1. Kryteria optymalizacji

$$\text{kryt}(1) = \int_0^t e(t) dt = \int_0^t |h_{\text{zad}}(t) - h(t)| dt \rightarrow \min - \text{całka z bez-}$$

względnej wartości odchyłki regulacji, oraz

$\text{kryt}(2) \rightarrow \min - \text{liczba oscylacji w określonym czasie;}$

2. Zmienne decyzyjne

$\text{zmdec}(1) = \text{strmasy}$ – wartość liczbową strumienia zasilania zbiornika,
 $\text{zmdec}(2) = \text{prog}$ – wartość sygnału wejściowego, przy której regulator otwiera zawór gdy odchyłka rośnie,
 $\text{zmdec}(3) = \text{hist}$ – histereza regulatora.

Zatem wartość sygnału wejściowego, przy której regulator zamyka zawór gdy odchyłka maleje wyniesie: $\text{prog} - \text{hist}$.

Zwróćmy uwagę, że **nie są** to funkcje czasu.

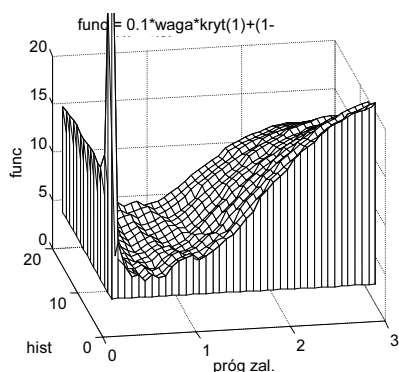
3. Ograniczenia

- Wszystkie zmienne decyzyjne powinny być dodatnie, co implikuje logikę zadania,
- $\text{zmdec}(1) < \text{wyd}$ – z przyczyn konstrukcyjnych strumień nie może być większy niż maksymalna wydajność instalacji zasilającej.

Analiza modelu matematycznego [2], prowadzi do następujących wniosków (intuicyjnie zrozumiałych):

- aby $\text{kryt}1$ było jak najmniejsze, prog powinien być minimalny przy małej histerezie, lecz przy większej histerezie ma minimum wewnątrz zakresu natomiast hist powinna być minimalna przy małej wartości prog , lecz przy większej wartości prog ma minimum wewnątrz zakresu;
- aby $\text{kryt}2$ było jak najmniejsze, $\text{prog} \rightarrow \max$ i jednocześnie $\text{hist} \rightarrow \max$, zatem jest to kryterium monotoniczne ze względu na te dwie zmienne decyzyjne;
- z przedstawionych wykresów nie wynika, jaka jest charakterystyka obu kryteriów ze względu na pierwszą zmienną decyzyjną: strmasy .

Kompromis leży gdzieś wewnątrz przeanalizowanego obszaru.



Rys. 6. Obraz skalarnej funkcji celu dla $waga=0,5$
 Fig. 6. Scalar objective function for weighting factor $waga=0,5$

Ten wniosek potwierdza analiza rysunku 6, na którym pokazano przebieg skalarnej funkcji celu będącej kombinacją liniową obu kryteriów:

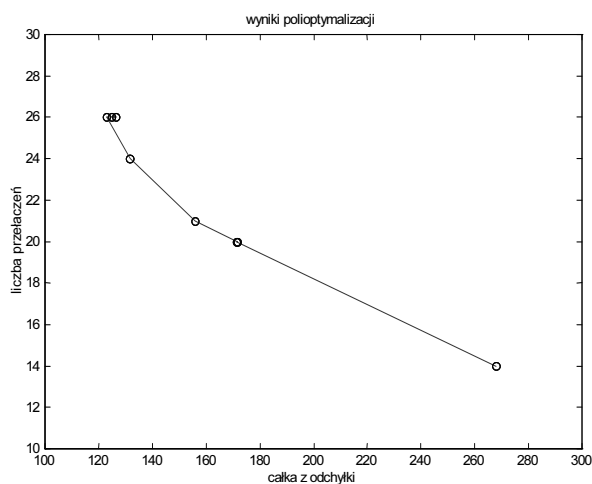
$$f = sk \cdot waga \cdot \text{kryt}(1) + (1 - waga) \cdot \text{kryt}(2)$$

Współczynnik skali $sk = 0.1$ dobrano tak, aby udział obu kryteriów był w przybliżeniu ten sam.

Minimum leży w okolicy punktu $[\text{prog}, \text{hist}] = [1, 2]$. Stąd wniosek konstrukcyjny, że wartość prog nie powinna być zbyt duża, natomiast hist znacznie większa, to znaczy wyłączenie powinno następować symetrycznie na lewo przy ujemnej wartości odchyłki regulacji.

Aby ocenić sytuację kompromisową, przeprowadzono polioptymalizację, wykorzystując klasyczny gradientowy algorytm optymalizacji, ze skalarą funkcją celu f .

Wyniki pokazano na rysunku 7.



Rys. 7. Zbiór rozwiązań polioptymalnych dla kolejnych wartości: $waga = 0.25, 0.35, \dots, 0.95$ (niektóre punkty niemal pokrywają się)

Fig. 7. Polyoptimal variants for the weight coefficient sequentially changed (some points are very close each other)

6. Procedura formułowania i rozwiązywania zadania polioptymalizacji

1. Sformułowanie zadania polioptymalizacji: kryterium nadrzędne, kryteria cząstkowe
2. Określenie funkcji sterujących;
3. Ustalenie zakresu optymalizacji; ustalenie czy parametry procesu (i obiektu) też są zmiennymi decyzyjnymi;
4. Zdefiniowanie ograniczeń;
5. Dyskretyzacja zmiennych niezależnych;
6. Uruchomienie modelu matematycznego procesu, symulacje dla różnych sterowań, weryfikacja modelu (w całym zakresie zmienności; nie musi być dokładnie);
7. Eksperymenty optymalizacyjne z jednym kryterium (pozostałe ustawione jako ograniczenia) – to weryfikacja procedury optymalizacyjnej;
8. Zgrubne określenie przedziałów zmienności poszczególnych kryteriów, odpowiednie ich wyskalowanie, tak aby ich wartości były współmierne;
9. Wybór metody optymalizacji:
 - a) jeśli algorytm genetyczny, utworzyć funkcję kary od ograniczeń i dodać do wszystkich kryteriów
 - b) jeśli metoda gradientowa, utworzyć zastępcze kryterium skalarne, na przykład jako ważoną sumę kryteriów cząstkowych;
10. Bardzo starannie przetestować działanie algorytmu.

7. Inne problemy: dyskretyzacja i wygładzanie poszukiwanych funkcji

Jeśli poszukujemy funkcji sterowania w przestrzeni zmiennych niezależnych – gdy są ciągłe – powinny zostać dyskretyzowane, jeśli zadanie optymalizacji (czy polioptymalizacji) ma być sformułowane jako zadanie optymalizacji statycznej. Aby zmniejszyć liczbę przedziałów (co przyspieszy obliczenia) bez dotkliwego pogorszenia dokładności, można rozważyć możliwość nierównomiernego podziału: zagęścić w tych obszarach, gdzie kryteria są najbardziej wrażliwe na sterowanie.

Innym bardzo skutecznym sposobem drastycznego zmniejszenia liczby zmiennych decyzyjnych jest zastosowanie funkcji wygładzających: np. funkcji splajnow lub funkcji Bezier'a [4].

W tym drugim przypadku dodatkowo należy zwrócić uwagę na ograniczenia przedziałowe wartości funkcji sterowania.

Warto zwrócić uwagę na grupę funkcji interpolacyjnych oprogramowanych w MATLAB'ie:

- INTERP1,
- INTERP1Q,
- INTERPFT,
- SPLINE,
- INTERP2,
- INTERP3,
- INTERPN.

Przykład

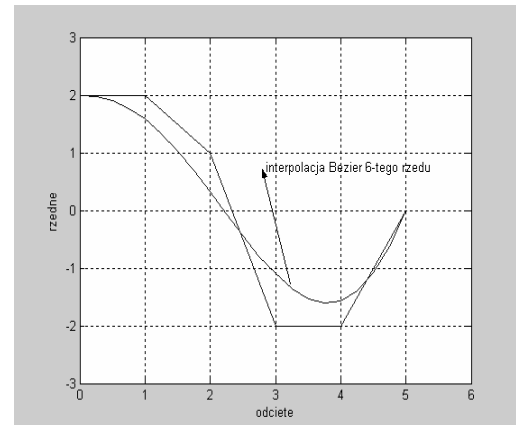
Dana jest funkcja za pomocą tablicy odciętych i rzędnych. Napisać funkcję w MATLAB'ie o nazwie *bezier6*.

```
function [odciete,rzedne]=bezier6(odc,rz)
% obliczanie 20 odciętych i rzędnych funkcji Bezier'a
% dla danych 6 wierzchołków: P0 do P5 o następn. współrz.:
% odc - wektor odciętych p-któw węzłowych
% rz - wektor rzędnych p-któw węzłowych
% t - parametr 0,00 ... 1,00

P0=[odc(1) rz(1)];
P1=[odc(2) rz(2)];
P2=[odc(3) rz(3)];
P3=[odc(4) rz(4)];
P4=[odc(5) rz(5)];
P5=[odc(6) rz(6)];
w = [1 1.2 1.2 1.4 1 1]; %wagi kolejnych wierzchołków
p=0:0.05:1;
for j=1:21,
    t=p(j);
    odciete(j)=(1-t)^5*P0(1) + 5*t*(1-t)^4*P1(1) + 10*t^2*(1-t)^3*P2(1)
    + 10*t^3*(1-t)^2*P3(1) + 5*t^4*(1-t)*P4(1) + t^5*P5(1);
    rzedne(j)=(1-t)^5*P0(2) + 5*t*(1-t)^4*P1(2)*w(1)
    + 10*t^2*(1-t)^3*P2(2)*w(2) + 10*t^3*(1-t)^2*P3(2)*w(3)
    + 5*t^4*(1-t)*P4(2)*w(4) + t^5*P5(2)*w(5);
end
clf
hold off
plot(odciete,rzedne);
hold
plot(odc,rz);
```

Przykładowy rezultat pokazuje rysunek 8.

Należy podkreślić, że w procesie symulacji (a więc obliczania funkcji celu podczas optymalizacji) wykorzystuje się funkcję interpolowaną, a nie dyskretyzowaną, nie ma więc utraty dokładności optymalizacji.



Rys. 8. Przykład aproksymacji funkcji dyskretnej: $rz = [2 \ 2 \ 1 \ -2 \ -2 \ 0]$, danej w punktach: $odc = [0 \ 1 \ 2 \ 3 \ 4 \ 5]$ za pomocą funkcji Bezier'a

Fig. 8. Approximation of a function given by co-ordinates: $odc = [0 \ 1 \ 2 \ 3 \ 4 \ 5]$, $rz = [2 \ 2 \ 1 \ -2 \ -2 \ 0]$, by the 6-th order Bezier function (example)

8. Wnioski

1. Zakresem optymalizacji powinien być jednocześnie i proces i obiekt w którym ten proces przebiega, oznacza to że należałoby rozwiązywać zadanie optymalizacji dynamicznej, co jest w praktyce niezwykle trudne;
2. Przez dyskretyzację zmiennych niezależnych można takie zadanie sformułować jako zadanie (poli-)optymalizacji statycznej, stosunkowo łatwe do rozwiązania;
3. Istotne dla dokładności takiego uproszczonego zadania jest sposób dyskretyzacji zmiennych niezależnych;
4. Bardzo uniwersalnym algorytmem (poli-)optymalizacji statycznej – choć obliczeniowo czasochłonnym – jest algorytm genetyczny.

9. Literatura

- [1] Tarnowski W.: Model procesu wyboru w projektowaniu technicznym. Zeszyt Naukowy Politechniki Śląskiej, Automatyka, Gliwice 1984.
- [2] Tarnowski W.: Static optimisation of non-linear processes by simulation. Modelling and simulation: a tool for the next millennium. 13th European simulation multicongress. Warszawa, 01–04.06.1999r., s. 513-518.
- [3] Tarnowski W.: Symulacja i optymalizacja w MATLAB'ie. Wydawnictwo Fundacja Wyższej Szkoły Morskiej w Gdyni, Gdynia 2001.
- [4] Foley J. D. I inni: Wprowadzenie do grafiki komputerowej. Warszawa 2001.
- [5] Tarnowski Wojciech: Optymalizacja statyczna procesów nieliniowych. Materiały XV Ogólnopolskiej Konf. „Poli-optymalizacja i Komputerowe Wspomaganie Projektowania” Mielno '97. str. 279–290.