

Zastosowanie algorytmu genetycznego do generacji reguł sterowania

Mariusz Podsiadło, e-mail: podsiadl@prz.rzeszow.pl

Marian Wysocki, e-mail: mwysocki@prz.rzeszow.pl

Politechnika Rzeszowska, Wydział Elektryczny, Katedra Automatyki i Informatyki
ul. W. Pola 2, 35-959 Rzeszów

1. Wstęp

W niniejszym artykule przedstawiono przykład zastosowania algorytmu genetycznego do syntezy regulatorów sterujących nieliniowymi obiektami dynamicznymi. Mimo, iż wiele obiektów nieliniowych może być z dobrym skutkiem sterowanych przy użyciu zwykłego regulatora PID, to spotyka się takie, dla których konieczne staje się użycie regulatora nieliniowego. Przykładem często stosowanego i opisywanego regulatora nieliniowego może być system z logiką rozmytą [1], [2], dla którego w zasadzie nie ma formalnych metod jego syntezy. Innym przykładem może być nieliniowa wielowarstwowa sieć neuronowa. Najczęstszą metodą projektowania nieliniowych sieci neuronowych jest ich uczenie za pomocą wstecznej propagacji błędów [3]. Jednak algorytm ten może być kosztowny obliczeniowo i nie zawsze łatwy do zastosowania. W [4] autorzy na podstawie doświadczeń zaprojektowali zestaw reguł sterowania obiektem, a następnie "wpisali" go do sieci za pomocą wyżej wymienionego algorytmu. Jest to oczywiście metoda bardzo złożona i wymagająca dużo czasu.

Poniżej przedstawiono przykład syntezy regulatorów rozmytego i neuronowego, zastosowanych do sterownia odwróconym wahadłem. Syntezy dokonano przy użyciu algorytmu genetycznego typu mikro (μ GA – Micro-Genetic Algorithm) [5].

2. Algorytm genetyczny typu mikro

W wielu zastosowaniach algorytmów genetycznych istnieje potrzeba obliczania funkcji dopasowania o bardzo skomplikowanej postaci, a co za tym idzie, dużym koszcie obliczeniowym. Dlatego istotny zysk, w postaci skrócenia czasu potrzebnego na uzyskanie wyniku, może przynieść zmniejszenie rozmiaru populacji. W pracy [6] szacowano "optymalny" rozmiar populacji na 30 do 200 osobników, co daje bardzo duży koszt obliczeniowy. W [5] przytoczono efektywny algorytm Goldberga wykorzystujący małą populację.

Algorytm operujący na małej populacji (μ GA) musi jednak być zmodyfikowany w stosunku do prostego algorytmu genetycznego (SGA – Simple Genetic Algorithm). μ GA wygląda następująco:

- I. Wygenerować losowo małą populację.
- II. Zastosować algorytm genetyczny aż do uzyskania określonej zbieżności (w sensie określonego wskaźnika, np. dopasowania najlepszego osobnika w populacji).

III. Wygenerować nową populację poprzez zachowanie najlepszego osobnika i wylosowanie pozostałych ponownie.

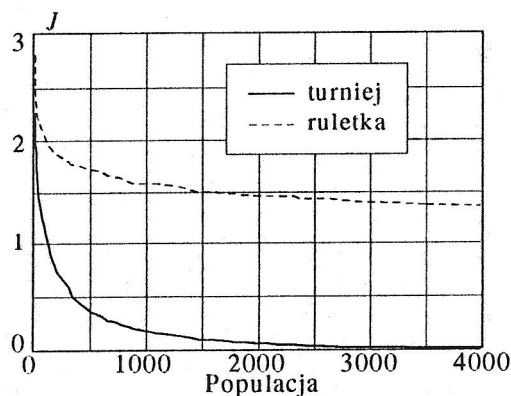
IV. Jeżeli nie jest spełniony warunek stopu, to powrót do kroku II, w przeciwnym wypadku koniec.

Algorytm genetyczny użyty w kroku drugim, nie może być zwykłym algorytmem SGA. Wynika to ze specyficznych właściwości μ GA. Po pierwsze nie ma konieczności stosowania mutacji, ponieważ co pewien czas losowana jest prawie cała nowa populacja. Po drugie, w przypadku małej populacji, w czasie selekcji nie jest dobrze zachowane prawo średniej, dlatego może się zdarzyć, że najlepsze genotypy nie przejdą do następnej populacji. W związku z tym należy używać selekcji turniejowej.

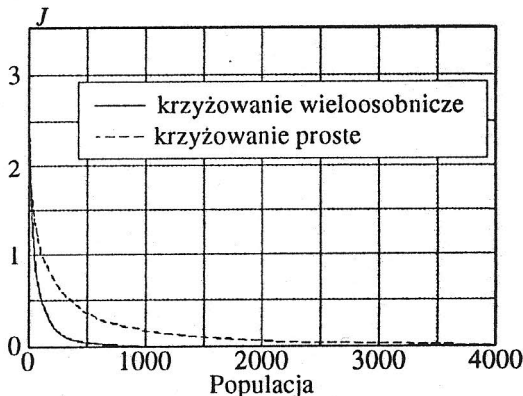
Działanie przedstawionego algorytmu autorzy przetestowali na szeregu trudnych do optymalizacji funkcji zaczerpniętych z pracy [7]. Testy przeprowadzono dla algorytmu operującego na populacji złożonej z pięciu osobników. Potwierdziły one efektywność algorytmu oraz potrzebę stosowania selekcji turniejowej. Na rysunku 1 pokazano porównanie zbieżności μ GA z selekcją ruletkową i turniejową; funkcja testowa określona jest równaniem:

$$J(x) = 1 + \sum_{i=1}^n x_i^2 / 40 - \prod_{i=1}^n \cos(x_i / \sqrt{i}); \quad -6 \leq x_i \leq 6$$

Do testów przyjęto $n = 10$.



Rys. 1. Porównanie zbieżności μ GA dla różnych metod selekcji

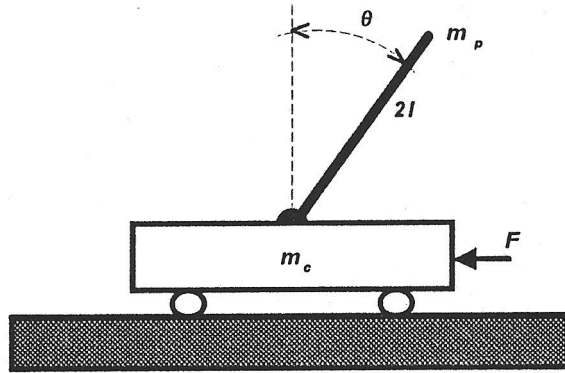


Rys. 2. Porównanie zbieżności μ GA dla różnych metod krzyżowania

Testy wykazały również wzrost efektywności algorytmu przy zastosowaniu krzyżowania wieloosobniczego. Był on zauważalny głównie dla złożonych funkcji. Krzyżowanie wieloosobnicze polega na tym, że fragmenty kodu genetycznego wymieniane są między więcej niż dwoma osobnikami (tu czterema). Na rysunku 2 pokazano porównanie efektywności obu sposobów krzyżowania. Obliczenia wykonano dla wyżej wymienionej funkcji.

3. Obiekt sterowania

Obiektem sterowania będzie wahadło odwrócone – rysunek 3. Jest to często wykorzystywany obiekt w pracach nad sterowaniem w układach nieliniowych. Składa się ze sztywnego pręta zamocowanego swobodnie (jeden stopień swobody) na wózku.



Rys. 3. Wahadło odwrócone.

Wózek może poruszać się w jednym kierunku (w płaszczyźnie swobody pręta), w wyniku przyłożonej siły. Zadanie sterowania polega na ustabilizowaniu wahadła w pozycji pionowej, przy czym sterowaniem jest siła oddziałująca na wózek.

Zakładając, że moment bezwładności pręta wynosi $\frac{1}{3}m_p l^2$ oraz zanedbując opory ruchu, otrzymuje się następujące równanie dynamiki pręta:

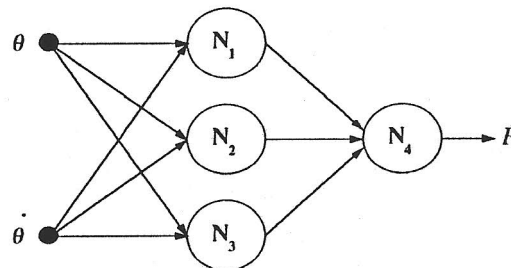
$$\ddot{\theta} = \frac{g \sin \theta - \cos \theta \left[\frac{F + m_p l \dot{\theta}^2 \sin \theta}{m_c + m_p} \right]}{l \left[\frac{4}{3} - \frac{m_p \cos \theta^2}{m_c + m_p} \right]}$$

gdzie: θ – kąt wychylenia pręta,
 m_c – masa wózka,
 m_p – masa pręta,
 l – połowa długości wahadła,
 g – przyspieszenie ziemskie,
 F – siła przyłożona do środka masy wózka.

Do obliczeń przyjęto wartości: $m_c = 1.0$ kg, $m_p = 0.1$ kg, $l = 0.5$ m, $|F_{\max}| = 10$ N.

4. Synteza regulatora neuronowego

Do sterowania obiektem zastosowano regulator neuronowy, w postaci dwuwarstwowej sieci neuronowej pokazanej na rysunku 4.



Rys. 4. Nieliniowa dwuwarstwowa sieć neuronowa zastosowana do sterowania wahadłem.

Węzłami sieci są nieliniowe neurony o logistycznej funkcji aktywacji. Funkcja przejścia neuronu opisana jest zależnościami:

$$\varepsilon = \sum_{i=1}^n x_i w_i + b_0$$

$$y = -1.0 + \frac{2.0}{1.0 + \exp(-\beta \varepsilon)}$$

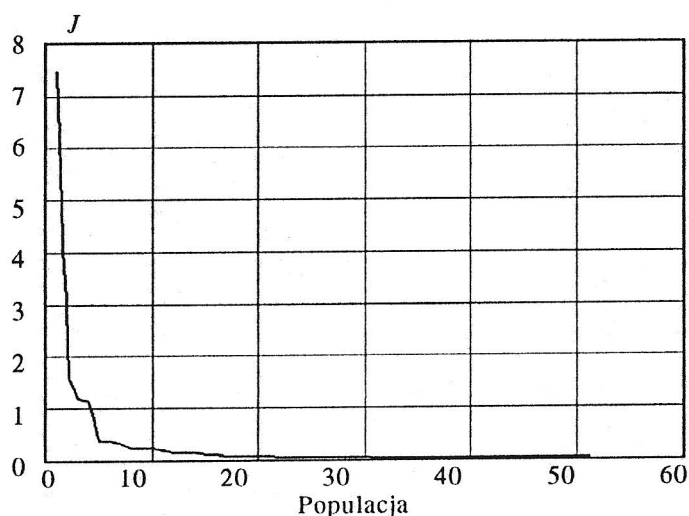
gdzie: x_i – sygnały wejściowe,
 w_i – wagi wejść,
 b_0 – przesunięcie,
 β – współczynnik,
 y – sygnał wyjściowy.

Zadanie syntezy regulatora sformułowano w postaci zadania optymalizacji, w którym zmiennymi decyzyjnymi są wartości wag sieci neuronowej, a funkcją kosztu jest wartość oczekiwana całki kwadratu uchybu regulacji dla różnych wartości początkowych:

$$\min_{w_{i,j}} (J = E \int_{x_0}^{\infty} e^2(t) dt)$$

gdzie: e – uchyb sterowania,
 E – średnia po wartościach początkowych,
 x_0
 $w_{i,j}$ – wagi sieci neuronowej.

Do rozwiązania problemu zastosowano algorytm genetyczny typu mikro. Wartości wag neuronów zakodowano w chromosomach w postaci liczb binarnych. Wartość funkcji dopasowania wyznaczana była poprzez symulację układu zamkniętego. Na rysunku 5 przedstawiono średni przebieg zbieżności algorytmu dla pięćdziesięciu pełnych procesów optymalizacji wag. Wskaźnikiem zbieżności jest dopasowanie najlepszego osobnika w populacji.

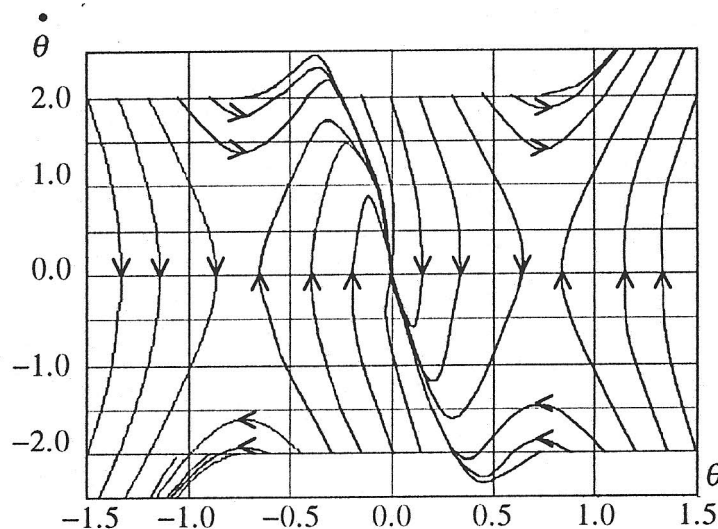


Rys. 5. Wykres średniej zbieżności μ GA dla problemu parametryzacji sieci neuronowej.

W tablicy 1 zestawiono wyniki – wagi sieci neuronowej z rysunku 4.

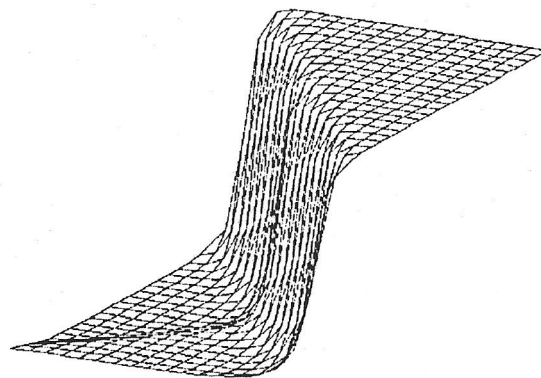
Tab. 1. Wagi sieci neuronowej

Neuron	w_1	w_2	w_3	b_0
N_1	-2.773	6.172	–	-8.691
N_2	9.492	-9.882	–	-1.992
N_3	9.003	7.656	–	0.84
N_4	-1.23	7.031	4.63	-2.558



Rys. 6. Portret fazowy układu zamkniętego z regulatorem neuronowym.

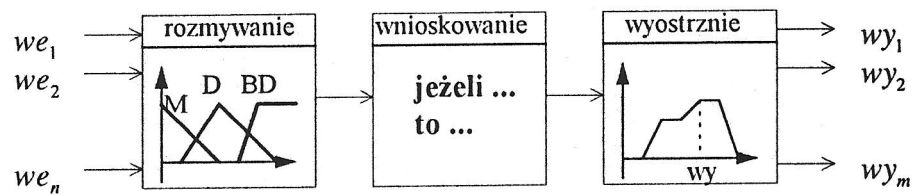
Rysunki 6 i 7 przedstawiają odpowiednio portret fazowy układu zamkniętego i nieliniową funkcję przejścia regulatora.



Rys. 7. Funkcja przejścia regulatora neuronowego.

5. Synteza regulatora rozmytego

Drugim regulatorem zastosowanym do sterowania obiektem był regulator rozmyty. Składa się on z trzech podstawowych bloków: bloku romywania, bloku wnioskowania i bloku wyostrzania, co przedstawiono schematycznie na rysunku 8.



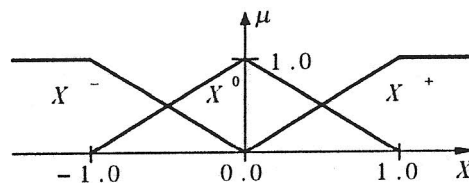
Rys. 8. Struktura regulatora rozmytego.

Rozmywanie polega na odwzorowaniu ciągłych zmiennych decyzyjnych (stan obiektu, wyjście obiektu, itp.), przy użyciu funkcji przynależności, na odpowiednie kategorie lingwistyczne (np. w odniesieniu do uchybu sterowania – “duży”, “mały”, itp.). Wnioskowanie odbywa się za pomocą reguł **jeżeli...to...** wyrażonych w kategoriach lingwistycznych. Otrzymane wyniki, są następnie odwzorowywane na wartości ciągłe zmiennych sterujących.

Podobnie jak w przypadku sieci neuronowej regulator rozmyty posiada dwa wejścia i jedno wyjście. Dla wejść i wyjść zdefiniowano następujące zbiory rozmyte:

- $X_1 = \{X_1^-, X_1^0, X_1^+\}$ – uniwersum wejścia pierwszego – kąt wychylenia wahadła,
- $X_2 = \{X_2^-, X_2^0, X_2^+\}$ – uniwersum wejścia drugiego – prędkość kątowa wahadła,
- $U = \{U^-, U^0, U^+\}$ – uniwersum wyjścia – siła sterująca.

Na rysunku 9 przedstawiono przykładowe funkcje przynależności zbiorów rozmytych. Funkcje są takie same dla odpowiadających sobie zbiorów w poszczególnych uniwersach.



Rys. 9. Funkcje przynależności zbiorów rozmytych.

	X_2^-	X_2^0	X_2^+
X_1^-	U^-	U^0	U^0
X_1^0	U^+	U^-	U^-
X_1^+	U^+	U^+	U^-

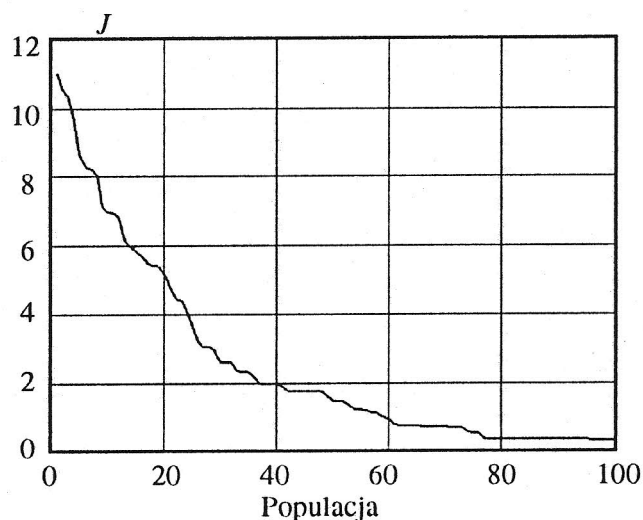
$IF (x_1 = X_1^-) \& (x_2 = X_2^+)$
 $THEN (u_1 = U^-)$

Rys. 10. Tablica reguł sterowania regulatora rozmytego.

Poszukujemy tablicy reguł sterowania, za pomocą których odbywać się będzie wnioskowanie. Rysunek 10 przedstawia przykład tablicy reguł i sposób jej interpretacji. Proces syntezy regulatora w tym przypadku został sformułowany jako problem optymalizacji, w którym funkcja kosztu określona jest jak poprzednio, lecz tym razem zmiennymi decyzyjnymi są następni reguł sterowania:

$$\min_{u(x_1, x_2)} (J = E \int_0^{\infty} e^2(t) dt)$$

gdzie: $u_{(x_1, x_2)}$ – następni reguł sterowania.



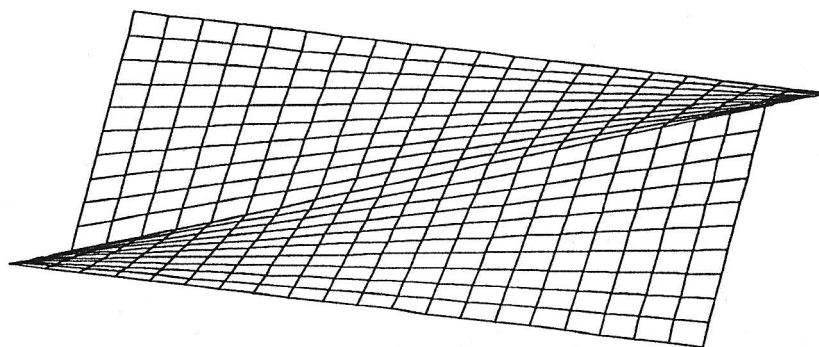
Rys. 11. Wykres średniej zbieżności μ GA dla problemu poszukiwania tablicy reguł.

Jak poprzednio, do rozwiązania problemu zastosowano algorytm genetyczny typu mikro. Przy czym tym razem w chromosomach zakodowano symbole następników reguł sterowania. Wartość funkcji dopasowania wyznaczana była poprzez symulację układu zamkniętego. Na rysunku 11 przedstawiono średni przebieg zbieżności algorytmu dla pięćdziesięciu pełnych procesów poszukiwania reguł.

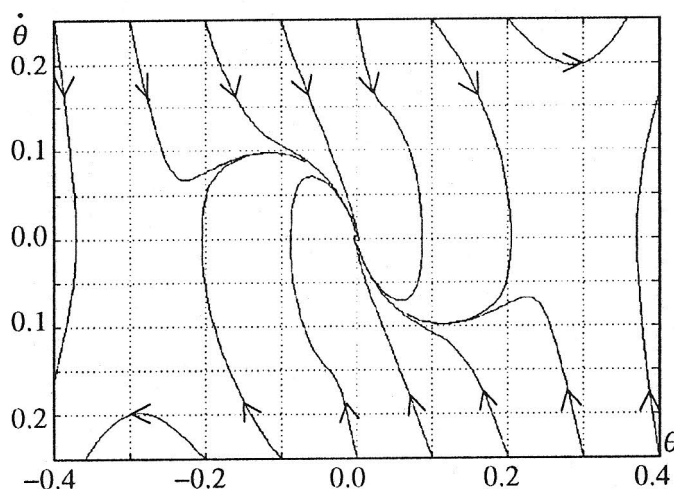
Tab. 2. Reguły sterowania.

	X_2^-	X_2^0	X_2^+
X_1^-	U^-	U^-	U^-
X_1^0	U^-	U^0	U^+
X_1^+	U^+	U^+	U^+

Otrzymane reguły zestawiono w tablicy 2. Zachowanie się układu zamkniętego z zaprojektowanym regulatorem ilustruje portret fazowy przedstawiony na rysunku 12. Rysunek 13 przedstawia funkcję przejścia regulatora.



Rys. 13. Funkcja przejścia regulatora rozmytego.



Rys. 12. Portret fazowy układu zamkniętego z regulatorem rozmytym.

6. Podsumowanie.

Przedstawione wyniki pokazują efektywność algorytmu genetycznego w zastosowaniu do projektowania układów automatycznej regulacji. Parametry regulatorów o różnej strukturze mogą być za jego pomocą w bardzo łatwy sposób optymalizowane pod względem dowolnego kryterium oceny jakości sterowania. Zapewniają to właściwości algorytmu genetycznego. Zastosowanie μ GA pozwala na znaczące przyspieszenie procesu projektowania regulatora.

Literatura

- [1] SUGENO M. (ed.), "Industrial Applications of Fuzzy Control", North-Holland, Elsevier, 1995.
- [2] KLUSKA J. "Sterowanie z logiką rozmytą", Wyd. Politechniki Rzeszowskiej, Rzeszów 1993.
- [3] TADEUSIEWICZ R. "Sieci neuronowe", Akademicka Oficyna Wydawnicza RM, Warszawa 1993.
- [4] GRANT E., BING ZHANG, "The Learned Control of Complex Dynamic Systems" in Final Report on the Learned Control of Complex Dynamic Systems, Turing Institute & University of Strathclyde, Glasgow 1990.
- [5] KALMANJE KRISHNAKUMAR, "Micro-Genetic Algorithms for Stationary and Non-Stationary Function Optimization", SPIE Vol. 1196 "Intelligent Control and Adaptive Systems", 1989.
- [6] GREFENSTETTE J. J., "Optimization of Control Parameters for Genetic Algorithms", IEEE Transactions on Systems, Man, and Cybernetics, SMC-16(1), 1986.
- [7] MUHLENBEIN H., SCHOMISCH M., BORN J., "The Parallel Genetic Algorithm as Function Optimizer", Parallel Computing 17, North-Holland Elsevier, 1991.

Podziękowanie:

Autorzy dziękują Panu Profesorowi Jackowi Klusce za dyskusje dotyczące regulatorów rozmytych.