

dr inż. Bogumiła Mrozek
Instytut Modelowania Komputerowego
Wydział Fizyki, Matematyki i Informatyki Stosowanej
Politechnika Krakowska
31-155 Kraków, ul. Warszawska 24

ALGORYTM SELEKCJI KLONALNEJ W ZASTOSOWANIU DO STROJENIA REGULATORÓW ROZMYTYCH

Algorytmy immunologiczne odwzorowują procesy adaptacji i zróżnicowane możliwości działania naturalnego systemu immunologicznego. Selekcja klonalna jest jednym z naturalnych procesów naturalnego systemu immunologicznego, które najczęściej są odwzorowywane w algorytmach. Algorytm selekcji klonalnej jest stosowany w zadaniach optymalizacji. Opisano implementację algorytmu selekcji klonalnej CLONAG w wersji dla rozwiązywania zadań optymalizacji. Przedstawiono wnioski z wykonanych testów symulacyjnych porównania algorytmu selekcji klonalnej i algorytmu genetycznego dostępnej biblioteki MATLAB-a Genetic Algorithm and Direct Search Toolbox w zastosowaniu do doboru parametrów regulatora PID.

CLONAL SELECTION ALGORITHM FOR TUNING FUZZY CONTROLLER

In this paper, the application of the clonal selection has been used to solve optimization problems. A computational implementation of the algorithm of clonal selection, named CLONALG [1] is adapted to solve optimization tasks. This algorithm was employed to design PID fuzzy logic controller for chosen testing object, described by the transfer function. The genetic algorithm, implemented in Genetic Algorithm and Direct Search Toolbox was used also for tuning the tree PID gains for fuzzy controller. The results of applications both algorithms used for tuning the designed fuzzy controller, based on defined quality index is compared.

1. WPROWADZENIE

Regulatory rozmyte umożliwiają tworzenie adaptacyjnych układów sterowania dla dużych zmian parametrów obiektu regulowanego, bez potrzeby korzystania ze skomplikowanych lub trudnych do uzyskania modeli tego obiektu.

Do strojenia regulatorów rozmytych algorytmami uczenia sieci neuronowych oraz metodami klasteryzacji są niezbędne odpowiednie (wiarygodne) zestawy danych, najlepiej pomiarowych. Jakość działania rozmytego regulatora można opisać wskaźnikiem jakości, określonym przez użytkownika. Dla zdefiniowanego wskaźnika jakości można dobrać strukturę i parametry regulatora rozmytego przy użyciu technik przeszukiwań.

Algorytmy genetyczne są to uniwersalne techniki optymalizacji oparte na przeszukiwaniu przestrzeni rozwiązań. Naśladują one genetyczną adaptację występującą w naturalnych procesach ewolucyjnych, zachodzących w przyrodzie [4].

W odróżnieniu od klasycznych metod optymalizacji, algorytmy te nie wymagają szczegółowej wiedzy o strukturze optymalizowanego problemu. Konieczne jest jedynie określenie funkcji celu. Dla regulatorów rozmytych, jako funkcję celu można przyjąć wskaźnik jakości, zdefiniowany przez użytkownika.

Populacja ewentualnych rozwiązań jest tworzona z upływem czasu za pomocą operatorów genetycznych takich jak: mutacja, krzyżowanie i selekcja. Jakość działania regulatora rozmytego można poprawić poprzez dostrojenie funkcji przynależności oraz wartości współczynników funkcji konkluzji, dla modelu Sugeno. Algorytm genetyczny można użyć do określenia zbioru wybranych parametrów, dla których regulator rozmyty działa optymalnie, względem przyjętego wskaźnika jakości.

Podstawową wadą algorytmów genetycznych jest ich wolna zbieżność i związany z tym długi czas obliczeń [10].

Algorytmy immunologiczne również działają w oparciu o przeszukiwanie przestrzeni rozwiązań i mogą być stosowane w zadaniach optymalizacji. Algorytmy te odwzorowują procesy adaptacji i zróżnicowanych możliwości działania naturalnego systemu immunologicznego. Są one stosunkowo nową techniką optymalizacji i jak podają [3] i [9] są jej efektywnym narzędziem. W porównaniu z algorytmami genetycznymi, charakteryzują się wysoką wydajnością [9].

Algorytm selekcji klonalnej jest stosowany w zadaniach optymalizacji, ale może być używany także w zagadnieniach eksploracji danych, klasyfikacji i klasteryzacji.

1.1. Regulatory rozmyte PID typu Sugeno

Regulatory rozmyte typu Sugeno w konkluzji reguł mają funkcje $f(x)$, a nie zbiory rozmyte.

Typowa reguła Sugeno dla modelu (regulatora) rozmytego z dwoma wejściami ma postać [6]:

$$\text{JEŚLI } (x \text{ jest } A) \text{ I } (y \text{ jest } B) \text{ TO } z = f(x,y)$$

gdzie: x i y są zmiennymi wejściowymi, A i B są zbiorami rozmytymi, $f(x,y)$ jest funkcją, z jest zmienną wyjściową.

W modelu typu Sugeno pierwszego rzędu, funkcja $f(x,y)$ jest wielomianem rzędu pierwszego. W takim przypadku, konkluzję i -tej reguły można zapisać w postaci ogólnej:

$$z_i = f(x,y) = p_i x + q_i y + r_i$$

gdzie: p_i, q_i, r_i – stałe.

Dla regulatora rozmytego PID i -tą regułą można zapisać następująco:

$$\text{JEŚLI } (U \text{ jest } A_i) \text{ TO } u_{st} = KP_i e_1 + KI_i e_2 + KD_i e_3$$

gdzie: A_i – zbiór rozmyty, U – zmienna wejściowa, KP_i, KI_i, KD_i – współczynniki wzmocnienia części proporcjonalnej, całkującej i różniczkującej klasycznego regulatora PID, e – błąd regulacji, $e_1 = e$, $e_2 = \int e \cdot dt$, $e_3 = de/dt$.

1.2. Algorytmy genetyczne i ewolucyjne

W algorytmach genetycznych i ewolucyjnych stosuje się pojęcia zapożyczone z genetyki naturalnej. Tworzy się *populację* rozwiązań. Każde rozwiązanie jest nazywane *osobnikiem*. Najczęściej jest on reprezentowany przez jeden *chromosom*, który zawiera zakodowane potencjalne rozwiązanie problemu. Elementy składowe *chromosomu* to uporządkowane ciągi *genów*. W *populacji* dokonuje się *selekcji* (słabe osobniki giną) i stosuje się operatory genetyczne *krzyżowania* i *mutacji*.

W każdej *generacji* (iteracji algorytmu genetycznego) jest oceniane przystosowanie każdego osobnika w danej populacji, za pomocą *funkcji przystosowania*. Na tej podstawie jest tworzona nowa populacja osobników. Reprezentują oni zbiór potencjalnych rozwiązań problemu. Reprodukacja skupia się na osobnikach o najwyższym stopniu przystosowania.

Operatory genetyczne takie jak krzyżowanie i mutacja mieszają owe osobniki, realizując w ten sposób eksplorację przestrzeni rozwiązań.

Kryteria zatrzymania algorytmu genetycznego (ewolucyjnego) to: wykonanie zadanej liczby generacji albo uzyskanie odpowiedniej wartości przystosowania osobników.

Główne różnice pomiędzy algorytmami genetycznymi i ewolucyjnymi [1], [3] to:

- sposób kodowania chromosomu
- algorytmy realizujące operatory genetyczne.

1.3. Algorytmy immunologiczne

Algorytmy immunologiczne używają pojęć zapożyczonych z immunologii, która zajmuje się badaniem reakcji odpornościowo-obronnej organizmów na *patogeny*, czyli wirusy, bakterie, toksyny i inne czynniki chorobotwórcze. Naturalny system immunologiczny jest systemem bardzo skomplikowanym i nie do końca poznany. Przyjmując pewne uproszczenia, buduje się algorytmy imitujące działanie niektórych jego mechanizmów [1], [9].

Algorytmy immunologiczne posługują się też pojęciami zapożyczonymi z genetyki naturalnej. Istnieją *populacje* składające się z *osobników*, reprezentujących rozwiązania lub dane wejściowe. Każdy osobnik populacji jest reprezentowany identycznie jak w chromosomie dla algorytmu genetycznego.

Osobniki reprezentują *przeciwciała* lub *antygeny* zakodowane w postaci łańcuchów binarnych, liczb rzeczywistych albo są to bardziej rozbudowane struktury, opisujące kompletnie pojedynczy obiekt. Każdy algorytm immunologiczny działa na *populacji przeciwciał* i *antygenów* lub samych przeciwciał.

Antygeny (ang. *antibody generator*) wywołują reakcję układu odpornościowego. *Przeciwciała* (ang. *antibody*) rozpoznają *antygeny* i wiążą się z nimi, co z kolei umożliwia zachodzenie innych procesów, takich jak neutralizowanie toksyn, wirusów itp.

W zadaniach optymalizacji jest zazwyczaj stosowana jedynie *populacja przeciwciał*. Reprezentuje ona zbiór potencjalnych rozwiązań zadania. Populacja antygenów nie jest używana (nie musi być używana).

Podobnie jak w algorytmach genetycznych, najpierw jest generowana populacja początkowa przeciwciał. Najczęściej proces ten ma charakter losowego doboru elementów tworzących przeciwciała, niekiedy ograniczanego przez określone warunki.

W porównaniu z algorytmem genetycznym, algorytmy immunologiczne w zdecydowanie bardziej naturalny sposób równoważą możliwości eksploracyjne i eksploatacyjne algorytmu, co prowadzi do powstania i stabilnego podtrzymania zróżnicowanych populacji osobników [9].

2. NATURALNY SYSTEM IMMUNOLOGICZNY

Ogólna zasada działania układu odpornościowego *polega na rozpoznawaniu antygenów* na zasadzie "swoj-obcy" i eliminowaniu tych ostatnich. W procesie tym uczestniczy kilka typów komórek, są to tzw. *limfocyty*, czyli białe ciała krwi. Limfocyty można podzielić z grubsza na *limfocyty typu T* i *limfocyty typu B*. Limfocyty T (tzw. *wspomagające*) dają sygnał do podjęcia akcji obronnej, a limfocyty B rozpoczynają niszczenie *patogenów*.

W naturalnym systemie immunologicznym wyróżnia się dwie warstwy odporności: *nieswoistej (wrodzonej)* i *swoistej (adaptacyjnej)*. Odporność nieswoista nie podlega rozwojowi i modyfikacjom, stąd ten rodzaj odporności nie jest dalej rozpatrywany.

2.1. System odporności adaptacyjnej

System ten ma zdolność do ciągłej adaptacji procesu rozpoznawania nowo pojawiających się antygenów oraz przechowywania informacji o już rozpoznanych antygenach. Te procesy, ze względu na wysoką efektywność, stanowią *główny przedmiot zainteresowań informatyków*, jako wzór algorytmów bardzo wydajnych obliczeniowo.

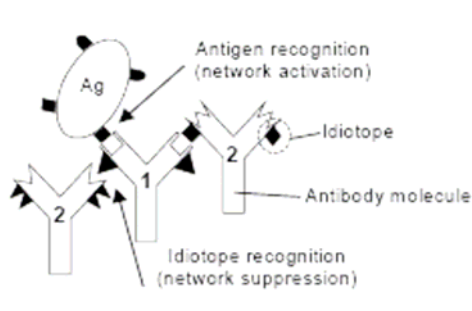
Proces, w którym system uczy się rozpoznawania nowego antygeny, nazwano *pierwotną odpowiedzią immunologiczną*. System odporności adaptacyjnej ma tzw. *pamięć immunologiczną*, dzięki której przechowuje przez pewien czas informacje o rozpoznanych antygenie. Stąd, przy kolejnym ataku, antygen jest szybciej rozpoznawany. Skutkuje to natychmiastową reakcją w postaci intensywnego wytwarzania specjalizowanych przeciwciał. Proces ten określa się jako *wtórna odpowiedź immunologiczną*. Limfocyty B i T są głównymi elementami warstwy adaptacyjnej systemu immunologicznego.

Limfocyty B są *przygotowywane do produkcji przeciwciał* o określonej strukturze molekularnej, przystosowanej do rozpoznawania określonej grupy antygenów.

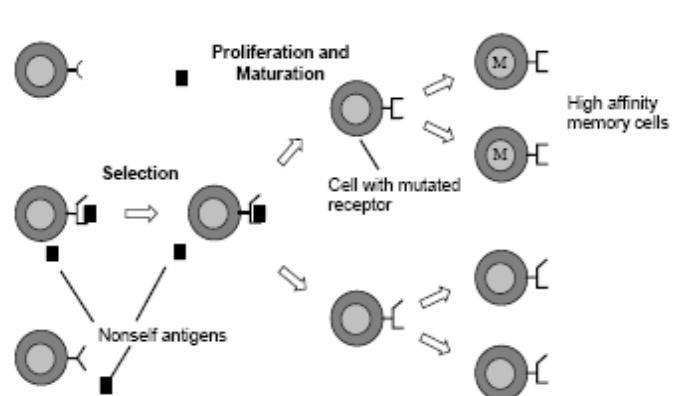
Limfocyty T realizują *mechanizm rozpoznawania "swój-obcy"*. Uczestniczą one w tzw. odpowiedzi typu komórkowego, wywołując bezpośrednią reakcję zwalczania antygeny. Rozróżnia się, co najmniej 4 typy limfocytów T.

Zarówno limfocyty B oraz T, jak i antygeny, wyposażone są w trójwymiarowe struktury chemiczne otaczające ich powierzchnię. Struktury te skonstruowane są z różnych sekwencji protein, które odgrywają główną rolę w wiązaniu przeciwciała z antygenem. *Fragment powierzchni antygeny* zawierający taką strukturę jest określany jako *epitop*.

Limfocyty B (komórki typu B – ang. *B-cells*) są otoczone przez zespół receptorów (**Ab**) – *przeciwciał*. Każde przeciwciało jest zbudowane ze specyficznego rodzaju białek wydzielanych przez pobudzone limfocyty B, tworząc kształt litery Y (rys. 1). Na każdym z ramion znajduje się struktura, nazywana *paratopem*, pozwalającą na dopasowanie do *epitopów* antygeny. Przeciwciało ma również epitopy. Siła wiązania epitop-paratop jest określana jako *stopień dopasowania*. Reprezentuje on stopień powinowactwa przeciwciała w stosunku do antygeny.



Rys. 1. Schematyczna budowa receptora komórki typu B i antygeny [1]



Rys. 2. Mechanizm selekcji klonalnej [1]

Duże zróżnicowanie limfocytów pozwala na rozpoznawanie szerokiej gamy różnych antygenów, lecz liczba komórek pierwotnie rozpoznających antygen jest zbyt mała by go zwalczyć.

Stąd, system immunologiczny został wyposażony w *mechanizm selekcji klonalnej*. Jego zadaniem jest namnożenie odpowiednich przeciwciał, uczestniczących w walce z antygenem. W systemie immunologicznym istnieje także *mechanizm selekcji negatywnej*, który działa na limfocyty T, rozpoznające własne komórki.

2.2. Mechanizm selekcji klonalnej

Aktywacja limfocyty B inicjuje reakcję systemu immunologicznego, powodującą namnożenie przeciwciał rozpoznających określony antygen.

Uaktywnione limfocyty B zaczynają intensywnie dzielić się, produkując wiele krótko żyjących klonów [1], [9]. Następnie klony te przechodzą proces *hipermutacji somatycznej*, celem wytworzenia lepiej dopasowanych przeciwciał (ang. *high affinity cells*). Proces ten przypomina mutację genetyczną, jednak jego intensywność jest o wiele większa. Etap ten określa się jako *prolifерacja* (klonowanie) [9] (rys. 2).

W rezultacie powstaje populacja zmutowanych klonów. Jest ona poddawana ocenie stopnia dopasowania do antygeny. Klony słabo wiążące antygen są usuwane z organizmu (*apoptoza*). Natomiast klony o wysokim stopniu dopasowania przekształcają się po pewnym czasie w komórki plazmatyczne albo komórki pamięciowe (*proces dyferencjacji*) [9] (rys.2).

Komórki plazmatyczne produkują wolne przeciwciała, które łącząc się z antygenem wskazują go komórkom żernym do eliminacji.

Komórki pamięciowe pozostają w organizmie przez dłuższy czas i biorą aktywny udział we wtórnej odpowiedzi immunologicznej.

3. ALGORYTM GENETYCZNY I SELEKCJI KLONALNEJ

Algorytm genetyczny i selekcji klonalnej zastosowano do strojenia regulatora rozmytego, dla którego określono wskaźnik jakości działania. Projektowanie i testowanie regulatora rozmytego wykonano w środowisku pakietu MATLAB-Simulink. Stąd, implementację algorytmu selekcji klonalnej wykonano w MATLAB-ie.

3.1. Implementacja algorytmu genetycznego

Algorytm genetyczny zaimplementowany w *Genetic Algorithm and Direct Search Toolbox* jest w zasadzie algorytmem ewolucyjnym. Udostępniono go poprzez funkcję *ga* oraz interfejs. Opcje tej funkcji dla poszczególnych etapów (kroków) działania algorytmu umożliwiają wybór implementacji kilku metod selekcji, mutacji, krzyżowania, generowania populacji początkowej i in. Poniżej wymieniono niektóre z algorytmów, możliwych do zastosowania:

- ✓ Tworzenie populacji początkowej – losowo z rozkładem równomiernym lub własna opcja.
- ✓ Skalowanie funkcji przystosowania – liniowe, rangowe, proporcjonalne i in.
- ✓ Selekcja – metoda ruletki, turniejowa i in.
- ✓ Krzyżowanie – równomierne, heurystyczne, jedno- i dwupunktowe, arytmetyczne i in.
- ✓ Mutacja – z rozkładem Gaussa, równomierna wielopunktowa, gradientowa i in.
- ✓ Wizualizacja procesu optymalizacji – wykresy dla wartości funkcji przystosowania, najlepszego osobnika, kryteriów zatrzymania i inne.

Domyślnie przyjęto, że geny w chromosomach osobników są liczbami rzeczywistymi. Można też wybrać opcję kodowania binarnego albo zdefiniować własną strukturę populacji.

Jest możliwa optymalizacja hybrydowa – optymalizację wstępną realizuje algorytm genetyczny, a końcową wskazana funkcja z *Optimization Toolbox*.

Schemat działania algorytmu genetycznego (ewolucyjnego) pokazano na rys. 3.

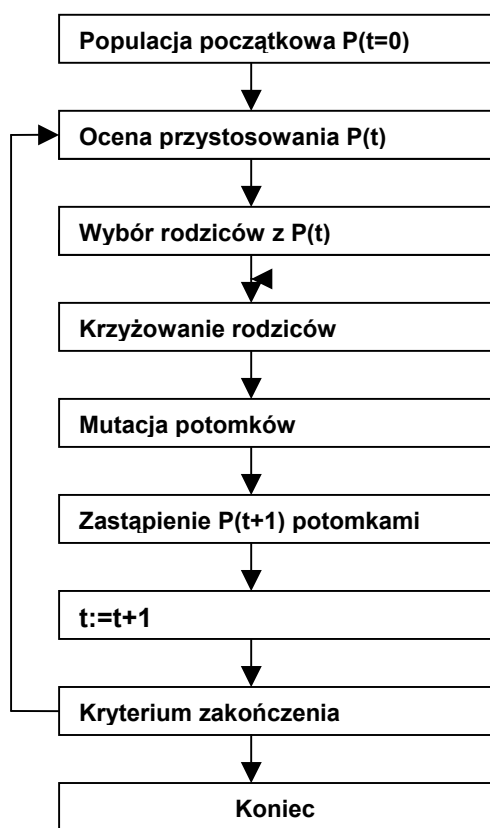
3.2. Implementacja algorytmu selekcji klonalnej

Algorytm selekcji klonalnej można podzielić na dwa etapy: *etap ekspansji klonalnej*, bazujący na naturalnym mechanizmie selekcji klonalnej oraz *etap hipermutacji*.

Ekspansja klonalna jest odpowiedzialna za wyselekcjonowanie najlepiej dopasowanych przeciwciał i namnożenie ich proporcjonalnie (ale niekoniecznie) do ich stopnia dopasowania.

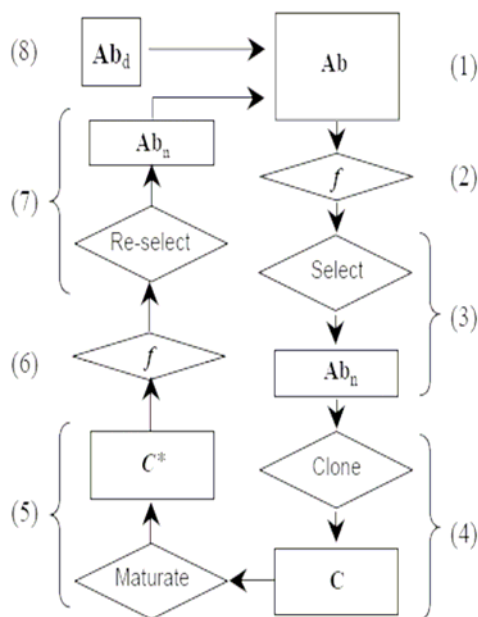
Hipermutacja realizuje proces dojrzewania przeciwciał, przekształcając namnożone klony w taki sposób, aby niektóre z nich osiągnęły lepszy stopień dopasowania niż ich poprzednicy.

W literaturze opisano kilka implementacji algorytmu selekcji klonalnej jak na przykład: AINE (Timmis 2000), aiNET (Castro i Zuben 2001), CLONAG [1, 2, 3] oraz [8]. Do dalszych rozważań przyjęto algorytm CLONAG. Istnieje on w dwóch wersjach, tj. dla zagadnień dotyczących rozpoznawania wzorców oraz dla rozwiązywania zadań optymalizacji [1, 2].



Rys. 3. Schemat działania algorytmu genetycznego

CLONALG: Optimization



Rys. 4. Schemat działania algorytmu selekcji klonalnej

Schemat działania algorytmu CLONAG, w wersji dla zadań optymalizacji [2] pokazano na rys. 4. Wprowadza się następujące oznaczenia: **Ab** – zbiór przeciwciał (reprezentuje rozwiązania), **Ab_{m}** – zbiór przeciwciał pamięci, **Ab_{r}** – reszta ze zbioru przeciwciał.

1. W wersji dla zadań optymalizacji nie ma populacji antygenów, która ma być rozpoznana, i cała populacja **Ab** tworzy zbiór pamięci. Dlatego nie ma potrzeby utrzymywania

- odrębnego zbioru pamięci $\mathbf{Ab}_{\{m\}}$. Natomiast pojawia się funkcja celu, którą się optymalizuje, tj. wyznacza się jej minimum albo maksimum. W tym przypadku, współczynnik dopasowania przeciwiła odpowiada obliczaniu wartości funkcji celu dla danego przeciwiła. Każde przeciwiła \mathbf{Ab}_i reprezentuje element przestrzeni rozwiązań.
- Określa się wektor \mathbf{f}_j , który zawiera podobieństwo (dopasowanie) do wszystkich N przeciwił w zbiorze \mathbf{Ab} . Stopień dopasowania przeciwiła (ang. *affinity*) odpowiada wartości funkcji celu dla danego przeciwiła.
 - Wybór n przeciwił o najwyższym stopniu dopasowania ze zbioru \mathbf{Ab} , które tworzą nowy zbiór $\mathbf{Ab}_{\{n\}}$ przeciwił z wysokim stopniem dopasowania.
 - Wybór n przeciwił, które będą klonowane (reprodukowane) niezależnie i proporcjonalnie do ich stopnia dopasowania (podobieństwa) do funkcji celu. Generowanie zbioru klonów \mathbf{C}^j : im wyższy stopień dopasowania do funkcji celu tym większa liczba klonów jest generowana, dla każdego z n wybranych przeciwił.
 - Zbiór \mathbf{C}^j jest poddawany ocenie w procesie dojrzewania podobieństwa (ang. *affinity maturation process*) odwrotnie proporcjonalnie do podobieństwa z funkcją celu, generując populację \mathbf{C}^{j*} dojrzałych klonów (hipermutacja): wyższy stopień podobieństwa, to mniejszy współczynnik mutacji.
 - Określa się wektor podobieństwa \mathbf{f}_j^* dla dojrzałych klonów \mathbf{C}^{j*} (populacja po procesie hipermutacji), w relacji do funkcji celu.
 - Ze zbioru dojrzałych klonów \mathbf{C}^{j*} ponownie wybiera się n przeciwił \mathbf{Ab}_j^* o najwyższym podobieństwie w relacji do funkcji celu. Są to kandydaci do wprowadzenia ich do zbioru komórek pamięci $\mathbf{Ab}_{\{m\}} = \mathbf{Ab}$. Jeśli stopień dopasowania wybranego przeciwiła jest większy niż jakiegoś przeciwiła z pamięci systemu, wtedy \mathbf{Ab}_j^* zastępuje to przeciwiła w zbiorze \mathbf{Ab} .
 - W zbiorze \mathbf{Ab} określa się d przeciwił z najniższym stopniem podobieństwa względem funkcji celu i zastępuje się je przez nowe osobniki generowane losowo (supresja).
- Po wykonaniu 8 kroków jak powyżej, można powiedzieć, że *generacja* (iteracja) jest kompletna.

4. BADANIA SYMULACYJNE

Środowisko pakietu oprogramowania MATLAB-Simulink umożliwia szybkie projektowanie i testowanie regulatorów rozmytych. Ponadto regulatory rozmyte o strukturze i parametrach zaprojektowanych z wykorzystaniem bibliotek MATLAB-a można stosować w aplikacjach przemysłowych.

Testowano możliwości strojenia regulatorów rozmytych z wykorzystaniem algorytmu selekcji klonalnej i algorytmu genetycznego dla regulatora rozmytego typu Sugeno, który jest implementacją dwóch różnych regulatorów PID dla obiektu opisanego transmitancją [11]:

$$G(s) = \frac{\omega^2}{s^2 + 2\xi\omega s + \omega^2} \quad \text{dla } \omega = 1, \xi = 0,1$$

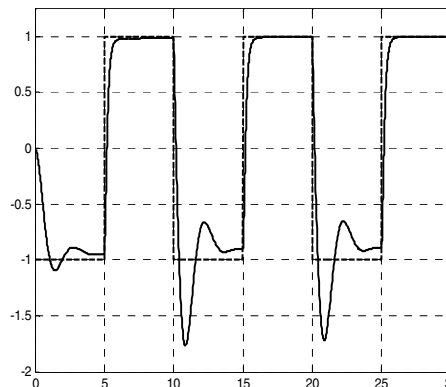
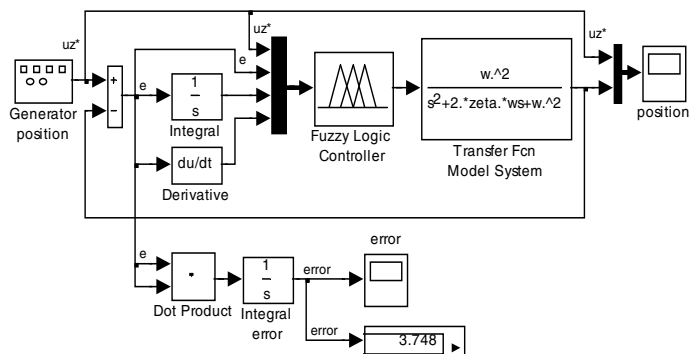
Badany układ zapisano jako model Simulinka pokazany na rys. 5a. Na wejście regulatora rozmytego wprowadzono prostokątny sygnał zadawania z generatora (rys. 5b).

Regulator rozmyty typu Sugeno zaprojektowano z wykorzystaniem biblioteki *Fuzzy Logic Toolbox* i jest on dostępny poprzez blok Simulinka. Sposób i zakres działania tego regulatora

opisuje zmienna lingwistyczna *position*, określona zbiorami rozmytymi A i B z trapezowymi funkcjami przynależności oraz dwie reguły jak poniżej:

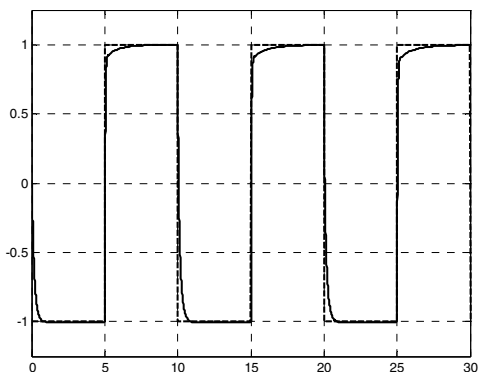
JEŚLI (*position* jest A) *TO* $z=k1x_2+k2x_3+k3x_4$

JEŚLI (*position* jest B) *TO* $z=k4x_2+k5x_3+k6x_4$



Rys. 5. Model Simulinka badanego obiektu z regulatorem rozmytym (a). Przebiegi sygnału zadawania z generatora i odpowiedzi wielkości regulowanej dla wstępnych parametrów regulatora rozmytego (b)

Funkcję przystosowania (celu) określono jako sumę błędu kwadratowego odpowiedzi badanego układu w czasie 30 s symulacji (rys. 5a). Zadaniem algorytmów selekcji klonalnej i genetycznego było znalezienie takich współczynników $[k1 \ k2 \ k3 \ k4 \ k5 \ k6]$ w konkluzjach regulatora typu Sugeno, aby osiągnąć minimum tak zdefiniowanej funkcji błędu.



Rys. 6. Przebiegi sygnału zadawania z generatora i odpowiedzi wielkości regulowanej dla parametrów określonych za pomocą algorytmu selekcji klonalnej

Najlepiej przystosowanego osobnika określał jeden chromosom, który składał się z genów reprezentujących poszukiwane współczynniki. Wartości wstępne parametrów regulatora rozmytego zakodowano jako wektor podwójnej precyzji (przebiegi z rys. 5b), który można zapisać: $k = [5 \ 1 \ 2 \ 60 \ 4 \ 14]$ (wartość błędu wynosiła 3,748).

Przebiegi sygnału zadawania i odpowiedzi wielkości regulowanej, które są wynikiem procesu optymalizacji z wykorzystaniem algorytmu selekcji klonalnej, pokazano na rys. 6. Wektor $k = [285.2 \ 3.5 \ 39.2 \ 27.8 \ 10 \ 19]$ zawiera optymalne parametry regulatora rozmytego, które uzyskano po około 120 generacjach, błąd regulacji miał wartość 0,690. Przyjęto rozmiar populacji 30, liczbę klonów 10 i współczynnik supresji 0,2.

Podobne rezultaty optymalizacji uzyskano po około 35 generacjach, przy użyciu algorytmu genetycznego z biblioteki *Genetic Algorithm and Direct Search Toolbox*, ale dla parametrów innych niż domyślne (za wyjątkiem rozmiaru populacji równego 20). Dobór odpowiedniego algorytmu krzyżowania i mutacji oraz ich współczynników, celem uzyskania właściwej zbieżności był dość pracochłonny.

5. UWAGI KOŃCOWE

Testowany algorytm selekcji klonalnej oraz algorytm genetyczny (ewolucyjny) zaimplementowany w *Genetic Algorithm and Direct Search Toolbox* umożliwia strojenie rozmytych regulatorów typu Sugeno na podstawie zdefiniowanego kryterium błędu, bez potrzeby stosowania danych uczących.

Algorytm selekcji CLONAG w wersji dla zadań optymalizacji jest stosunkowo łatwy w użyciu tzn. ma stosunkowo niewielką liczbę parametrów do określenia. Mimo tego, iż do uzyskania podobnego rezultatu potrzebuje większej liczby generacji niż algorytm genetyczny, to uzyskanie tego wyniku tzn. właściwej zbieżności algorytmu, jest zdecydowanie mniej pracochłonne.

LITERATURA

1. de Castro L. N. and Timmis J. I., *Artificial Immune Systems: A Novel Paradigm to Pattern Recognition*, in *Artificial Neural Networks in Pattern Recognition*, J. M. Corchado, L. Alonso, and C. Fyfe (eds.), SOCO-2002, University of Paisley, UK, 2002
2. de Castro L. N. and Von Zuben F. J., *Learning and Optimization Using the Clonal Selection Principle*, IEEE Transactions on Evolutionary Computation, Special Issue on Artificial Immune Systems, 6(3) 2002
3. de Castro L. N. and Timmis J. I., *Artificial Immune Systems as a Novel Soft Computing Paradigm*, Soft Computing Journal, 7(7), 2003
4. *Fuzzy Logic Toolbox User's Guide*, The MathWorks, Inc.
5. Robous H., Setnes M., *Compact fuzzy models and classifiers through model reduction and evolutionary optimization*, in *The practical handbook of genetic algorithms: Applications*, Chapman&Hall/CRC, 2001
6. Mrozek B., *Projektowanie regulatorów rozmytych w środowisku MATLAB/Simulink*, Pomiary Automatyka Robotyka, nr 11/2006.
7. Ross T. J., *Fuzzy logic with engineering applications*, John Wiley&Sons, 2004
8. Wierzchoń S.T., Kuźelewska U., *Stable clusters formation in an artificial immune system*, Conference on AIS, University of Kent at Canterbury, UK 9-11.09.2002,
9. Wierzchoń S. T., *Sztuczne systemy immunologiczne. Teoria i zastosowania*, Akademicka Oficyna Wydawnicza EXIT, Warszawa 2001
10. Wnuk P., *Zastosowanie algorytmu PSO do optymalizacji sposobu rozmywania w modelach TSK*, Pomiary Automatyka Kontrola, nr 9 /2005
11. *Fuzzy Logic Toolbox User's Guide*, The MathWorks, Inc.