Mostefa Mohamed-Seghir
Akademia Morska w Gdyni

PROGRAMOWANIE DYNAMICZNE W ROZMYTYM OTOCZENIU DO STEROWANIA STATKIEM

W artykule przedstawiono propozycję zastosowania programowania dynamicznego do rozwiązywania problemu wyznaczania bezpiecznej trajektorii statku w sytuacjach zagrożeń kolizyjnych w rozmytym otoczeniu. Podano przykłady symulacyjne bezpiecznej trajektorii statku w sytuacjach mijania się z ruchomymi obiektami.

1. WPROWADZENIE

W artykule przedstawiono algorytm programowania dynamicznego w rozmytym otoczeniu. Programowanie dynamiczne, wprowadzone przez Bellmana w połowie lat pięćdziesiątych, jest jedną ze standardowych metod rozwiązywania wielu klas zadań, przede wszystkim typu wieloetapowego podejmowania decyzji i sterowania [1, 2]. W pracy zaprezentowano sposób rozwiązywania procesu bezpiecznego sterowania statkiem w sytuacjach kolizyjnych w rozmytym otoczeniu metodą programowania dynamicznego. W procesie sterowania statkiem w sytuacjach kolizyjnych pojęcia takie jak „szybkość bezpieczna", „ryzyko kolizji" i określenie „działanie podjęte w celu zapobiegania zderzenia powinno być wykonane wystarczająco wcześnie" są niedokładnie i nieszczegółowo zdefiniowane. Nawigator wie, że są to określenia subiektywne i nieprecyzyjne oraz w dużym stopniu zależą od okoliczności i warunków realizacji procesu kierowania ruchem statku.

2. MODEL PROCESU BEZPIECZNEGO STEROWANIA STATKIEM W ROZMYTYM OTOCZENIU

Proces sterowania charakteryzujący się przedstawionymi powyżej cechami można opisać ogólnym modelem wieloetapowego podejmowania decyzji (sterowania) w rozmytym otoczeniu (rys. 1). Na wszystkie możliwe decyzje nałożone są pewne ograniczenia, a więc nie wszystkie decyzje są dopuszczalne i dlatego poszukuje się decyzji optymalnej wśród rozwiązań dopuszczalnych. Konieczne jest wprowadzenie pojęcia otoczenia rozmytego sformułowanego przez Bellmana i Zadeha [1, 3], w postaci uporządkowanej czwórki < ***G, C, D, U*** >, w której:

G – cel rozmyty,
C – ograniczenia rozmyty,
D – decyzja rozmyta,
U – zbiór decyzji.

Cel rozmyty określa się jako zbiór rozmyty $\boldsymbol{G} \subseteq \boldsymbol{U}$ o funkcji przynależności μ_G:

$$\mu_G : \boldsymbol{X} \times \boldsymbol{U} \to [0,1] \in \boldsymbol{R}.$$

Ograniczenie rozmyte określa się jako zbiór rozmyty $\boldsymbol{C} \subseteq \boldsymbol{U}$ o funkcji przynależności μ_C:

$$\mu_C : \boldsymbol{X} \times \boldsymbol{U} \to [0,1] \in \boldsymbol{R}.$$

Funkcja przejść stanu opisana jest następująco:

$$f(X_t, S_t) \to \boldsymbol{X},$$

$$X_{t+1} = f(X_t, S_t) \quad \text{...........}, \tag{1}$$

$$X_{t+1}, X_t \in \boldsymbol{X} = \{a_1, a_2, ..., a_j) \text{ – przestrzeń stanów,}$$

$$S_t \in \boldsymbol{U} = \{c_1, c_2, ..., j_.) \qquad \text{– przestrzeń sterowań,}$$

przy czym:

X_t – stan na etapie t, $t = 0, 1, 2, ..., N$,
S_t – sterowanie na etapie t, $t = 0, 1, 2, ..., N$.

Przejście procesu ze stanu początkowego X_0 do stanu końcowego X_N na skutek podejmowanych decyzji opisuje trajektoria tworzona przez ciąg kolejnych sterowań i stanów:

$$X_1 = f(X_0, S_0)$$

$$X_2 = f(X_1, S_1) = f(f(X_0, S_0), S_1)$$

$$.$$

$$\tag{2}$$

$$.$$

$$X_N = f(X_{N-1}, S_{N-1}) = f(f(f\ ...\ (f(X_0, S_0), S_1)\ ...\ )S_{N-1}).$$

Stan X_t określony jest przez trajektorię:

$$X_0, S_0, X_1, S_1, ..., X_{t-1}, S_{t-1}$$

$$t = 0, 1, 2, ... , N.$$

Załóżmy, że znany jest stan początkowy X_0. Jakość sterowania określona jest wówczas przez decyzję rozmytą będącą agregacją celów i ograniczeń rozmytych na kolejnych etapach sterowania w postać funkcji przynależności [3]:

$$\mu_D(S_0, S_1, \ldots, S_{N-1}, | X_0) = \mu_C{}^0(S_0) * \mu_G{}^1(X_1) * \mu_C{}^1(S_1)\ \mu_G{}^2(X_2) * \ldots$$

$$\ldots * \mu_C{}^{N-1}(S_{N-1}) * \mu_G{}^N(X_N). \tag{3}$$

Problem podejmowania optymalnej decyzji polega na wyznaczeniu optymalnego ciągu decyzji S_0^*, S_1^*, ..., S_{N-1}^* maksymalizującego funkcję przynależności decyzji rozmytej wynikającej z celu rozmytego i ograniczenia na poszczególnych etapach:

$$\mu_D(S_0^*, S_1^*, \ldots, S_{N-1}^*, | X_0) = \max_{S_0,\ldots S_{N-1}} (\mu_C{}^0(S_0) * \mu_G{}^1(X_1) * \mu_C{}^1(S_1) * \mu_G{}^2(X_2) * \ldots$$

$$\ldots * \mu_C{}^{N-1}(S_{N-1}) * \mu_G{}^N(X_N). \tag{4}$$

Operacjami agregacji (*), które mogą być stosowane do rozpatrywanego przypadku wieloetapowego procesu podejmowania decyzji w rozmytym otoczeniu, są operacje przecięcia, iloczynu, kombinacji wypukłej i sumy mnogościowej. Przejęcie tych operacji prowadzi odpowiednio do decyzji rozmytej typu minimum, iloczynu, kombinacji wypukłej i maksimum [3, 5].

3. ALGORYTM PROGRAMOWANIA DYNAMICZNEGO W ROZMYTYM OTOCZENIU

Aby pokazać ideę programowania dynamicznego, najpierw trzeba przedstawić zadanie w nieco pełniejszej postaci [3, 4]:

$$\mu_D(S_0^*, S_1^*, \ldots, S_{N-1}^*, | X_0) = \max_{S_0,\ldots S_{N-1}} (\mu_C{}^0(S_0) \wedge \mu_G{}^1(X_1) \wedge \mu_C{}^1(S_1) \wedge \mu_G{}^2(X_2) \wedge \ldots$$

$$\ldots \wedge \mu_C{}^{N-1}(S_{N-1}) \wedge \mu_G{}^N(f(X_{N-1}, S_{N-1}))). \tag{5}$$

Można zauważyć, że w tym wyrażeniu po prawej stronie znaku równości dwa ostatnie człony z prawej strony, tzn. $\mu_C{}^{N-1}(S_{N-1})$ i $\mu_G{}^N(f(X_{N-1}, S_{N-1}))$, zależą jedynie od sterowania S_{N-1}, a nie zależą od pozostałych sterowań S_0, S_1, ..., $S_{N-2.}$ Zatem maksymalizację względem ciągu sterowań S_0, S_1, ..., $S_{N\text{-}1}$ można rozdzielić na maksymalizację według sterowania S_{N-1}:

$$\mu_D(S_0^*,\ S_1^*,\ \dots,S_{N-1}^*,|\ X_0) = \max_{S_0,\dots S_{N-2}} (\mu_C^{\ 0}(S_0)\wedge\mu_G^{\ 1}(X_1)\wedge\mu_C^{\ 1}(S_1)\wedge\mu_G^{\ 2}(X_2)\wedge\dots$$

$$\dots \qquad \wedge\mu_C^{\ N-2}(S_{N-2})\wedge\mu_G^{\ N-1}(X_{N-1})\wedge$$

$$\wedge\max_{S_{N-1}} (\mu_C^{\ N-1}(S_{N-1})\wedge\mu_G^{\ N}(f(X_{N-1},\ S_{N-1})))) \tag{6}$$

oraz na maksymalizację względem ciągu sterowań S_0, S_1, ..., S_{N-2}. Postępując w ten sam sposób dla sterowania S_{N-2} i dla ciągu sterowań S_0, S_1, ..., S_{N-3}, otrzymuje się następujące równanie:

$$\mu_D(S_0^*,\ S_1^*,\ \dots,S_{N-1}^*,|\ X_0) = \max_{S_0,\dots S_{N-3}} (\mu_C^{\ 0}(S_0)\wedge\mu_G^{\ 1}(X_1)\wedge\mu_C^{\ 1}(S_1)\wedge\mu_G^{\ 2}(X_2)\wedge\dots$$

$$\dots\wedge\mu_C^{\ N-3}(S_{N-3})\wedge\mu_G^{\ N-2}(X_{N-2})\wedge\max_{S_{N-2}} (\mu_C^{\ N-2}(S_{N-2})\wedge\mu_G^{\ N-1}(X_{N-1})\wedge$$

$$\wedge\max_{S_{N-1}} (\mu_C^{\ N-1}(S_{N-1})\wedge\mu_G^{\ N}(f(X_{N-1},\ S_{N-1}))))) \tag{7}$$

Postępując analogicznie, otrzymuje się następujący układ równań rekurencyjnych:

$$\mu_G^{\ N-i}(X_{N-i}) = \max_{S_{N-1}} (\mu_C^{\ N-i}(S_{N-i})\wedge\mu_G^{\ N-i+1}(X_{N-i+1}))$$

$$X_{N-i+1} = f(X_{N-i},\ S_{N-i}) \tag{8}$$

W powyższym wzorze $\mu_G^{\ N-i}(X_{N-i})$ można traktować jako funkcję przyna–leżności celu rozmytego na etapie sterowania $N-i$, generowanego przez cel rozmyty na etapie sterowania $N-i+1$.

Rozwiązując równanie (8), otrzymuje się kolejno wartości optymalne sterowań S_{N-1}^*, i = 1, 2, ..., N, składające się na poszukiwany optymalny ciąg sterowań S_0^*, $S_1^*,\ \dots,S_{N-1}^*$. Wartość sterowania S_{N-1}^* otrzymuje się z równania:

$$S_{N-1}^* = p_{N-1}^*(X_{N-i}),\ \ i = 1,\ 2,\ \dots,\ N, \tag{9}$$

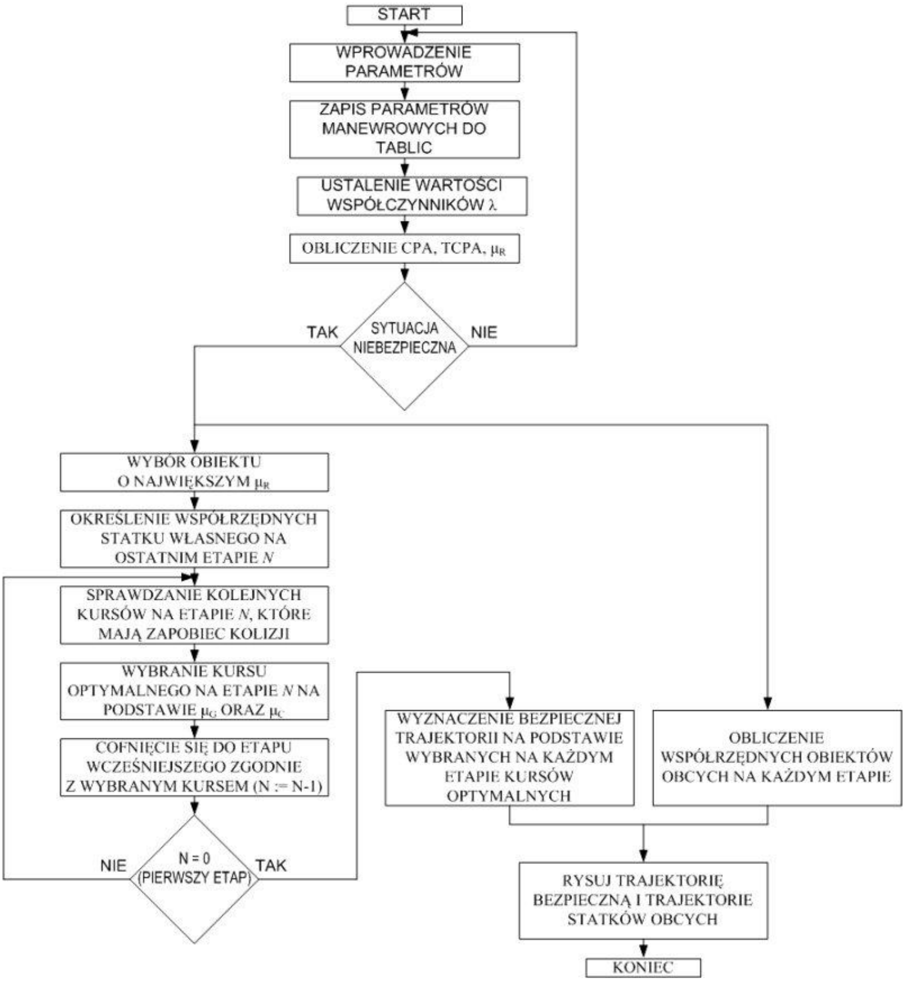
przy czym:

$$p^* = (\ p_0^*,\ \ p_1^*,\dots,\ \ p_{N-1}^*). \tag{10}$$

Ze względu na skończoność przestrzeni stanów i przestrzeni sterowań musi być spełniona zależność $\mu_D(S_0,\ S_1,\ \dots\ ,\ S_{N-1},|\ X_0) \neq 0$ dla przynajmniej jednego ciągu sterowań i każdego stanu początkowego X_0 – wówczas istnieje optymalna decyzja.

4. SYMULACYJNE WYNIKI WYZNACZANIA BEZPIECZNEJ TRAJEKTORII STATKU W SYTUACJACH KOLIZYJNYCH

W celu sprawdzenia prawidłowości działania algorytmu programowania dynamicznego w rozmytym otoczeniu opracowano wizualny program komputerowy DYNAMICPRO 1.0, do wyznaczania bezpiecznej trajektorii statku w sytuacjach kolizyjnych w rozmytym otoczeniu. Na rysunku 1 przedstawiono algorytm działania programu DYNAMICPRO 1.0.



CPA – najmniejsza odległość zbliżenia
TCPA – czas pozostający do osiągnięcia najmniejszej odległości zbliżenia
μ_R – wartość funkcji przynależności zbioru ryzyka kolizji
λ – współczynnik odzwierciedlający subiektywizm nawigatora w podejmowaniu decyzji

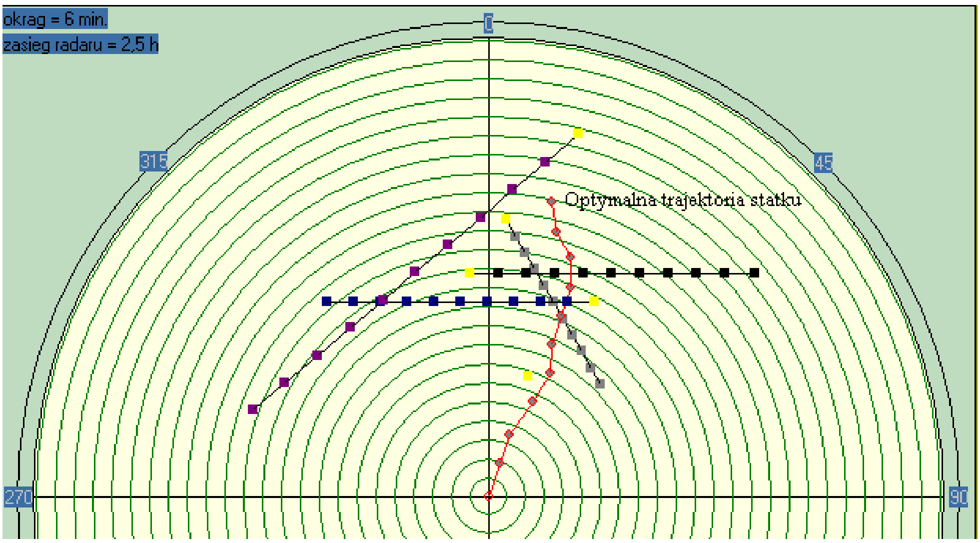
Rys. 1. Schemat blokowy algorytmu działania programu DYNAMICPRO 1.0

Przykład symulacji procesu bezpiecznego sterowania statkiem z wykorzystaniem algorytmu programowania dynamicznego do wyznaczania bezpiecznej trajektorii statku w rozmytym otoczeniu przedstawiono na rysunku 2, a parametry manewrowe wykorzystane w sytuacji nawigacyjnej umieszczono w tabeli 1.

Tabela 1

Parametry manewrowe statku własnego i obiektów

	Prędkość [w]	Kurs [°]	Odległość [Mm]	Namiar [°]
Statek własny	19	12		
Obiekt 1	18	270	22	50
Obiekt 2	0	0	8	18
Obiekt 3	17	90	16	320
Obiekt 4	27	50	16	290
Obiekt 5	12	330	10	45



Rys. 2. Fragment ekranu radarowego, w którym przedstawiono optymalną bezpieczną trajektorię statku w sytuacji mijania się z pięcioma spotkanymi obiektami

5. WNIOSKI

W artykule pokazano sposób przedstawienia problemu wyznaczania optymalnej trajektorii bezpiecznej statku w rozmytym otoczeniu, a następnie rozwiązania problemu sterowania statkiem metodą programowania dynamicznego. Wyniki opisanych sytuacji nawigacyjnych, dają obraz możliwości wyznaczania trajektorii bezpiecznej w sytuacjach kolizyjnych przy zastosowaniu metody programowania dynamicznego. W dalszych pracach badawczych będą przedstawione wyniki innych metod opartych na zbiorach rozmytych.

LITERATURA

1. Bellman R. E. Zadeh L.A., *Decision making in afuzzy environment*, Management Sci., 1970, 17.
2. Bubnicki Z., *Problem automatyki i informatyki*, PAN, Wrocław 1998.
3. Kacprzyk J., *Zbiory rozmyte w analizie systemowej*, PWN, Warszawa 1986.
4. Lisowski J., M. Mohamed-Seghir, *Safe Ship Control Methods Based on Fuzzy Set Theory*, Polish Journal of Environment Studies, 2008, vol. 17, no 3C.
5. Mohamed-Seghir M., *Fuzzy Set Theory in Safe Ship Automation Control*, 10th IEEE International Conference on Methods and Models in Automation and Robotics, MMAR 2004.

PROGRAMMING DYNAMIC IN A FUZZY ENVIRONMENT FOR SHIP'S CONTROL

Summary

The paper describes application of dynamic programming method to solve task of determination of safe ship trajectories in the collision situation in fuzzy environment. Moreover, examples of simulations of safe ship passing situations with moving targets are included.