Zdzisław KOWALCZUK[1], **Karol DUZINKIEWICZ**[2]
[1] POLITECHNIKA GDAŃSKA, WYDZIAŁ ELEKTRONIKI, TELEKOMUNIKACJI I INFORMATYKI
[2] INSTYTUT WDROŻEŃ TECHNICZNYCH „INTECH"

Planowanie trajektorii ruchu zespołu robotów mobilnych z zastosowaniem metody warstwicowej

Prof. dr hab. inż. Zdzisław KOWALCZUK

Profesor zwyczajny i kierownik Katedry Systemów Decyzyjnych, Wydział Elektroniki Telekomunikacji i Informatyki Politechniki Gdańskiej. Urodzony w Gdańsku 1953, mgr inż. 1978, dr 1986, dr hab.1993, profesor nauk technicznych 2003. Zakres badań naukowych: teoria i projektowanie komputerowych systemów sterowania i diagnostyki oraz adaptacja, modelowanie i identyfikacja, przetwarzanie sygnałów i sztuczna inteligencja. Laureat Nagrody Fundacji na Rzecz Nauki Polskiej (1999) w dziedzinie nauk techn.

e-mail: kova@eti.pg.gda.pl

Mgr inż. Karol DUZINKIEWICZ

W roku 2006 otrzymał tytuł magistra inżyniera kierunku Automatyka i Robotyka na Wydziale Elektroniki, Telekomunikacji i Informatyki Politechniki Gdańskiej. Obecnie jest pracownikiem Instytutu Wdrożeń Technicznych INTECH na stanowisku Specjalisty ds. Projektowania i Wdrożeń. Równocześnie kontynuuje prace nad doktoratem pod opieką prof. dr hab. Zdzisława Kowalczuka w Katedrze Systemów Decyzyjnych. Obszarem badawczym jest planowanie trajektorii ruchu pojazdów bezzałogowych.

e-mail: karol.duzinkiewicz@gmail.com

Streszczenie

Podstawową część artykułu stanowi opis zaproponowanej przez autorów metody planowania trajektorii ruchu dla zespołu autonomicznych robotów mobilnych poruszających się w określonym szyku. Zastosowano w tym celu zmodyfikowaną metodę sztucznego potencjału[1], nazwaną przez autorów metodą warstwicową. Przedstawiona procedura jest wieloetapowa. Każdy z etapów został opisany z zastosowaniem odpowiednich schematów proceduralnych, zaś wyniki działania każdego z algorytmów składowych zostały zilustrowane wykresami pochodzącymi z symulacji komputerowych.

Słowa kluczowe: planowanie trajektorii, zespoły robotów, metoda sztucznego potencjału.

Motion trajectory planning of multi-robot team using contour line method

Abstract

The purpose of the paper is to introduce a multi-stage trajectory planning algorithm for mobile robots. Unmanned Ground Vehicles (UGVs) and Unmanned Aerial Vehicles (UAVs) became very popular in the last few years. Many military and civil applications can be found in real life. One of the most challenging issues regarding UGVs and UAVs concern their autonomous control in environment with obstacles. The proposed algorithm is based on a contour line method, modification of the artificial potential field method designed to efficiently escape local minima of the artificial potential function.

Keywords: trajectory planning, multi-robot teams, artificial potential field method.

1. Wstęp

Coraz większą popularnością cieszą się zespoły bezzałogowych pojazdów, zdolnych do wykonywania różnorakich funkcji. Prowadzone są liczne badania w zakresie ich wykorzystania w różnych dziedzinach życia, od monitoringu dużych obszarów, poprzez misje transportowe, ratunkowe i eksploracyjne, po zadania militarne związane z niszczeniem wrogich celów, patrolowaniem stref przygranicznych oraz przestrzeni powietrznej [1, 2].

Prowadzone są intensywne prace nad opracowaniem efektywnych metod sterowania tego typu urządzeniami [1, 3-6]. Podstawowym celem w tego typu metodach jest minimalizacja roli człowieka w podejmowaniu decyzji.

W takim kontekście autonomiczny system sterowania pojazdami odpowiedzialny jest za generowanie odpowiednich, niskopoziomowych sygnałów sterujących pracą poszczególnych jednostek [3].

2. Aktualny stan wiedzy

Zdecydowanie najszerzej opisanym w literaturze problemem związanym ze sterowaniem zespołem autonomicznych pojazdów jest zagadnienie sterowania grupą urządzeń typu UAV[2] [1-3]. Wynika to przede wszystkim zaawansowanym stanem prac nad konstrukcją techniczną takich maszyn. Przykładowo, w armii amerykańskiej stosowanych jest w obecnej chwili wiele tego typu urządzeń. Wykonują one zwykle misje rekonesansowe i szpiegowskie, ale też coraz częściej wykorzystywane są z powodzeniem w misjach *stricte* bojowych, polegających na zniszczeniu określonych celów przy jednoczesnym unikaniu kontaktu z innymi obiektami wroga (takimi jak stacje radarowe, wyrzutnie rakiet) [2].

Rys. 1. Przykłady współczesnych pojazdów bezzałogowych (źródło: WWW)
Fig. 1. Examples of modern unmanned vehicles (source: WWW)

Dużym ułatwieniem przy projektowaniu algorytmów sterowania systemami UAV jest charakter środowiska, w jakim się one poruszają. Zwykle są to maszyny latające na dużych wysokościach, gdzie nie występują naturalne przeszkody w postaci nierówności terenu itp. Planowanie trajektorii ruchu tego typu pojazdów jest ułatwione, gdyż najczęściej możemy sprowadzić je do problemu składania ze sobą odcinków linii prostych oraz łuków o określonym promieniu [1, 7, 8].

Sytuacja komplikuje się w przypadku pojazdów naziemnych (UGV[3]). W związku z tym, że przeszkody naziemne zazwyczaj nie mogą zostać pominięte, należy opracować metody wyznaczające optymalne trajektorie ruchu omijające te przeszkody [5, 8-10].

Wybór metody planowania trajektorii zależy w znacznym stopniu od postaci mapy otoczenia, jaka jest dostępna. Najczęściej spotykane w praktyce rodzaje map środowiska, w którym poruszają się pojazdy możemy podzielić na dwie podstawowe kategorie: mapy rastrowe i mapy metryczne [11-13].

W przypadku map rastrowych przestrzeń, w której poruszają się pojazdy jest podzielona na szereg jednakowych sektorów.

[1] ang. Artificial Potential Field method

[2] ang. Unmanned Air Vehicle – bezzałogowy pojazd powietrzny
[3] ang. Unmanned Ground Vehicle – bezzałogowy pojazd naziemny

Każdy z takich sektorów można określić jako 'zajęty' lub 'wolny'. Znając postać tak zdefiniowanej mapy można wygenerować ścieżkę z punktu A do punktu B poruszając się jedynie pomiędzy sektorami określonymi jako 'wolne'. W celu projektowania trajektorii w ten sposób najczęściej wykorzystywane są algorytmy oparte na tzw. diagramach Voronoi [14] lub na propagacji sztucznej fali [11]. Oczywiście, jakość zaplanowanej na tej podstawie trajektorii zależy w znacznym stopniu od rozmiarów sektora. Wraz ze wzrostem rozdzielczości mapy rastrowej rośnie również złożoność obliczeniowa tych algorytmów.

Mapy metryczne nie dyskretyzują przestrzeni w taki sposób jak mapy rastrowe. Udostępniają one informację o rzeczywistym położeniu przeszkód, na przykład poprzez przechowywanie położeń wierzchołków brył aproksymujących te przeszkody. Metody sterowania oparte na wydobywaniu informacji z mapy otoczenia (lub jakiegokolwiek innego modelu otoczenia) nazywane są ogólnie mianem sterowania refleksyjnego [12].

Ciekawym zagadnieniem jest zadanie sterowania w przypadku zupełnego braku jakiejkolwiek mapy. Niekiedy jednak taka sytuacja jest wymuszona. Czasem jest to wynik świadomej rezygnacji z tego typu informacji ze względu na oszczędność zasobów (pojemność pamięci operacyjnej, moc obliczeniową). Sterowanie tego typu nosi nazwę sterowania reakcyjnego i charakteryzuje się dużą prostotą, połączona jednak ze sporą efektywnością w przypadku prostych zastosowań [8, 11, 13]. Znaczne osiągnięcia na polu sterowania reakcyjnego odnotowali uczeni czerpiący swoje pomysły ze świata zwierząt, szczególnie owadów [13]. Jedną z najpopularniejszych odmian sterowania reakcyjnego jest sterowanie behawioralne – oparte o predefiniowane schematy zachowań, które wykonuje każdy z robotów. Najczęściej są to proste reguły oparte o zasadę akcji i reakcji [3].

3. Struktura procedury planowania trajektorii zespołu robotów

Rozważono przypadek planowania trajektorii zespołu robotów mobilnych w środowisku z przeszkodami w przestrzeni 2D. Przeszkody są aproksymowane przez dwa rodzaje figur: wielokąty wypukłe oraz okręgi. Ruch zespołu będzie odbywać się pomiędzy punktem startowym o współrzędnych (x_{init}, y_{init}) i docelowym o współrzędnych (x_{goal}, y_{goal}). Zaprojektowana trajektoria powinna spełniać kryterium minimalnego dopuszczalnego promienia krzywizny R_{MIN}, aby umożliwić przejazd formacji w niezakłóconym szyku.

4. Planowanie trajektorii referencyjnej

Etap konstruowania trajektorii referencyjnej można podzielić na kilka podetapów: (1) wyznaczenie ścieżki pomiędzy punktem startowym i docelowym na podstawie zmodyfikowanej metody sztucznego potencjału (2) optymalizacja trajektorii za pomocą algorytmu Dijkstry, polegająca m.in. na eliminowaniu pętli (3) korekta trajektorii z uwzględnieniem rozmiarów formacji (4) wygładzenie trajektorii wyjściowej poprzez zastosowanie krzywych Béziera.

5. Modyfikowana metoda sztucznego potencjału i algorytm warstwicowy

Do określenia ścieżki łączącej punkt startowy z punktem docelowym wykorzystano zmodyfikowaną metodę sztucznego potencjału (APF) [9, 11], nazwaną *algorytmem warstwicowym*.

Metoda APF wykorzystuje pojęcie wirtualnej funkcji potencjalnej $U(x,y)$. Funkcja ta skonstruowana jest tak, aby jej minimum globalne leżało w punkcie (x_{goal}, y_{goal}). Wirtualne siły przyciągają punkty trajektorii do punktu docelowego, zaś odpychają od przeszkód.

Ogólnie rzecz biorąc funkcja $U(x,y)$ ma następującą postać [11]:

$$U(x,y) = U_{goal}(x,y) + U_{rep\_obs}(x,y), \tag{1}$$

gdzie $U_{goal}(x,y)$ jest składową odpowiedzialną za przyciąganie trajektorii w kierunku punktu docelowego, zaś składowa $U_{rep\_obs}(x,y)$ odpycha trajektorię od przeszkód. Te dwie składowe można przedstawić za pomocą wzorów [11]:

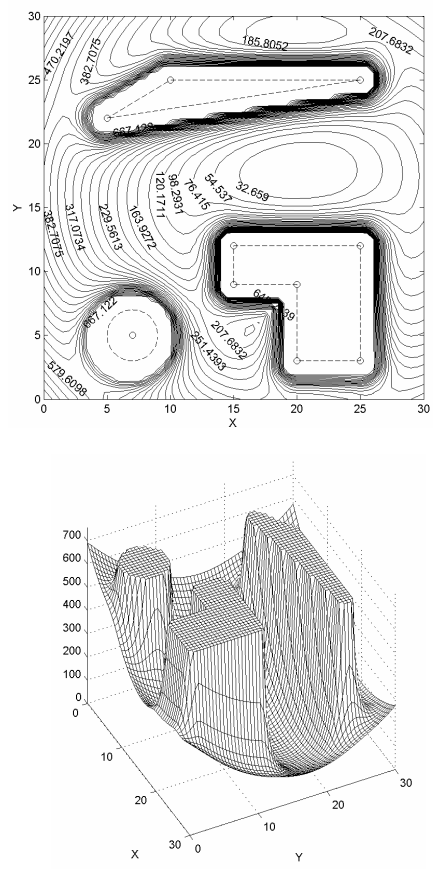
$$U_{goal}(x,y) = K_{goal} \cdot d_{goal}(x,y)^2 \tag{2}$$

$$U_{rep\_obs}(x,y) = \max\{U_{\max}, U_{rep}(x,y)\}, \tag{3}$$

gdzie K_{goal} jest stałym współczynnikiem, $d_{goal}(x,y)$ jest odległością pomiędzy obecnym położeniem (x,y) i punktem docelowym (x_{goal}, y_{goal}). Wartość $U_{rep}(x,y)$ jest wyznaczana zgodnie ze wzorem [11]:

$$U_{rep}(x,y) = \sum_{m=1}^{M} \frac{1}{2} \cdot \eta \cdot \left(\frac{1}{d_{obs}(x,y)} - \frac{1}{d_{range}} \right)^2, \tag{4}$$

gdzie M oznacza liczbę wszystkich przeszkód zamieszczonych na mapie, η jest stałym współczynnikiem, $d_{obs}(x,y)$ jest odległością do najbliższej przeszkody, $U_{\max}$ i d_{range} są stałymi współczynnikami. Przykład wygląd funkcji $U(x,y)$ przedstawia rys. 1, na którym linie przerywane określają pozycję przeszkód.



Rys. 2. Przykładowy wygląd funkcji $U(x,y)$ dla (x_{goal}, y_{goal})=(20,17)
Fig. 2. Exemplary shape of $U(x,y)$ function for (x_{goal}, y_{goal})=(20,17)

MOST WIEDZY Pobrano z mostwiedzy.pl

W standardowej metodzie APF ścieżka wyznaczana jest za pomocą gradientowej metody poszukiwania minimum funkcji $U(x,y)$. Metoda ta opiera się na iteracyjnym podążaniu od punktu początkowego $(x_0,y_0)=(x_{init},y_{init})$ do punktu docelowego (x_{goal},y_{goal}) – każdorazowo przesuwając się o wektor $\vec{d}_{step}$ zgodnie z kierunkiem danym przez wektor wirtualnej siły $\vec{F}(x,y)$ wyznaczanej jako gradient [11]:

$$\vec{F}(x,y)=-\nabla U(x,y)\,. \tag{5}$$

Wektor siły jest normalizowany i skalowany przez stały współczynnik δ w celu wyznaczenia aktualnego przesunięcia kolejnego punktu trajektorii (x_i,y_i) [11]:

$$\vec{d}_{step}(x,y)=[d_{step_x},d_{step_y}]=\delta\cdot\frac{\vec{F}(x,y)}{\left|\vec{F}(x,y)\right|}\,. \tag{6}$$

Algorytm kończy działanie po osiągnięciu określonej bliskości punktu docelowego. Wszystkie punkty otrzymywane w kolejnych iteracjach algorytmu są zapamiętywane w celu późniejszego odtworzenia trajektorii od punktu (x_{init},y_{init}) do (x_{goal},y_{goal}).

Wspomniany algorytm może zostać przedstawiony jako następująca procedura iteracyjna [11]:

Procedura 1. Standardowy algorytm metody APF

1. Ustaw: $(x_0,y_0)=(x_{init},y_{init})$; $i=0$
2. Jeżeli $d_{dis\tan ce}(x,y)>d_{\min\_dist}$ to
   $x_{i+1}=x_i+d_{step_x}$
   $y_{i+1}=y_i+d_{step_y}$
   w przeciwnym przypadku: KONIEC (algorytm osiągnął odpowiednią bliskość punktu docelowego)
3. Ustaw: $i=i+1$ → IDŹ DO KROKU 2

gdzie $d_{\min\_dist}$ to stała dokładności.

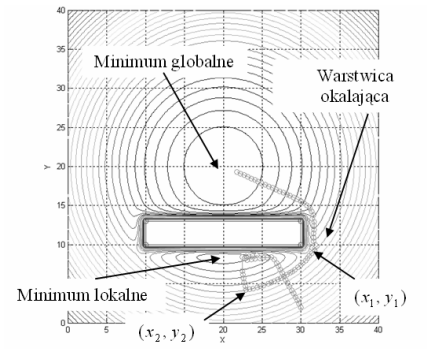
Podstawową wadą tego typu algorytmu jest niebezpieczeństwo zablokowania się algorytmu w lokalnym minimum funkcji $U(x,y)$. W celu uniknięcia tego problemu zaproponowano poniższą modyfikację – *metodę warstwicową.*

6. Metoda warstwicowa

Podstawą metody warstwicowej jest następujące spostrzeżenie. Jeżeli istnieje pewna zamknięta warstwica I, zwana dalej *ratunkową warstwicą okalającą*, okrążająca lokalne minimum funkcji $U(x,y)$ w położeniu $(x_{local\_\min},y_{local\_\min})$ oraz punkt docelowy (x_{goal},y_{goal}) oraz istnieją na tej warstwicy takie dwa *punkty ratunkowe* (x_1,y_1) oraz (x_2,y_2), które spełniają następujące warunki (wyjścia):

- dla punktu (x_1,y_1) istnieje ścieżka łącząca ten punkt z innym minimum lokalnym i można ją znaleźć za pomocą standardowego algorytmu metody APF
- dla punktu (x_2,y_2) istnieje ścieżka łącząca ten punkt z punktem $(x_{local\_\min},y_{local\_\min})$ i można ją wyznaczyć za pomocą standardowego algorytmu metody APF

to również można na jej podstawie określić ścieżkę łączącą punkty $(x_{local\_\min},y_{local\_\min})$ i (x_{goal},y_{goal}) przebiegającą w *wolnej przestrzeni operacyjnej*, którą definiujemy jako zbiór wszystkich punktów przestrzeni operacyjnej nie należących do zbioru rozważanych przeszkód. Ideę warstwicy okalającej prezentuje rys. 3.



Rys. 3. Przykład wyjścia algorytmu warstwicowego z lokalnego minimum funkcji *U*(*x,y*)
Fig. 3. Example of escape from a local minimum of *U*(*x,y*) found by contour line algorithm

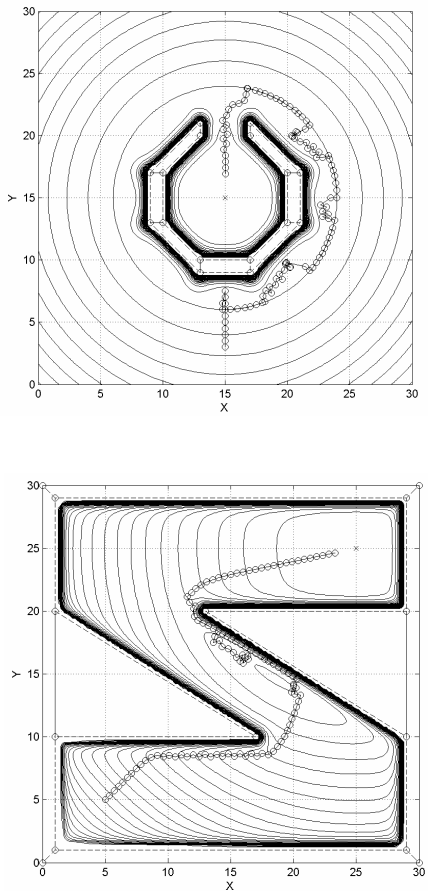
Procedura 2. Algorytm metody warstwicowej

1. Ustaw: $(x_0,y_0)=(x_{init},y_{init})$; $i=0$
2. Wykonaj krok standardowego algorytmu metody APF .
3. Jeżeli (x_i,y_i) jest lokalnym minimum funkcji $U(x,y)$ to zakończ wykonywanie algorytmu APF i:

- Sprawdź czy znalezione lokalne minimum jest minimum globalnym:
  jeżeli TAK → KONIEC (droga od punktu startowego do docelowego została znaleziona);
  jeżeli NIE → wykonaj poniższe kroki:
- Zapamiętaj pozycję wykrytego minimum lokalnego: $(x_{local\_\min_j},y_{local\_\min_j})=(x_i,y_i)$ (j to indeks lokalnego minimum).
- Znajdź najniższą warstwicę I, która okrąża zarówno (x_{goal},y_{goal}) jak i $(x_{local\_\min_j},y_{local\_\min_j})$.
- Znajdź najbliższy punkt $(x_{1,j},y_{1,j})$ na I, z którego można dostać się do innego (jeszcze nie odwiedzonego) lokalnego minimum przy użyciu metody APF.
- Znajdź najbliższy punkt $(x_{2,j},y_{2,j})$ na I, z którego można dostać się do nowo wykrytego lokalnego minimum $(x_{local\_\min_j},y_{local\_\min_j})$ przy użyciu metody APF.
- Dodaj do znalezionej trajektorii cześć warstwicy pomiędzy punktami (x_{1j},y_{1j}) i (x_{2j},y_{2j})
- WRÓĆ DO KROKU 3

MOST WIEDZY Pobrano z mostwiedzy.pl

Przykłady działania metody warstwicowej przedstawione są na poniższym rysunku.



Rys. 4. Przykłady działania algorytmu warstwicowego
Fig. 4. Exemplary results of contour line algorithm

Jak widać na powyższych rysunkach trajektoria znaleziona przy pomocy metody warstwicowej wymaga wygładzenia (m.in. usunięcia pętli) oraz zapewnienia odpowiedniego promienia krzywizny.

7. Optymalizacja trajektorii ratunkowej

Optymalizacja trajektorii będzie polegała na wyborze ze zbioru M wszystkich punktów trajektorii pewnego podzbioru N elementowego, który będzie reprezentował ścieżkę krótszą i pozbawioną pętli. W tym celu zastosowano algorytm Dijkstry [15]. Graf skierowany $G = (V, E)$, gdzie V to zbiór wierzchołków, zaś E to zbiór krawędzi jest skonstruowany na podstawie trajektorii znalezionej przez metodę warstwicową w taki sposób, że każdy wierzchołek grafu odpowiada punktowi trajektorii i jest połączony krawędzią tylko z tymi wierzchołkami, które odpowiadają punktom trajektorii następującym po nim.

Funkcja kosztu dla poszczególnych krawędzi grafu ma postać daną wzorem:

$$w(v_i, v_j) = K_L \cdot L(v_i, v_j) + K_U + MaxU(v_i, v_j) \tag{7}$$

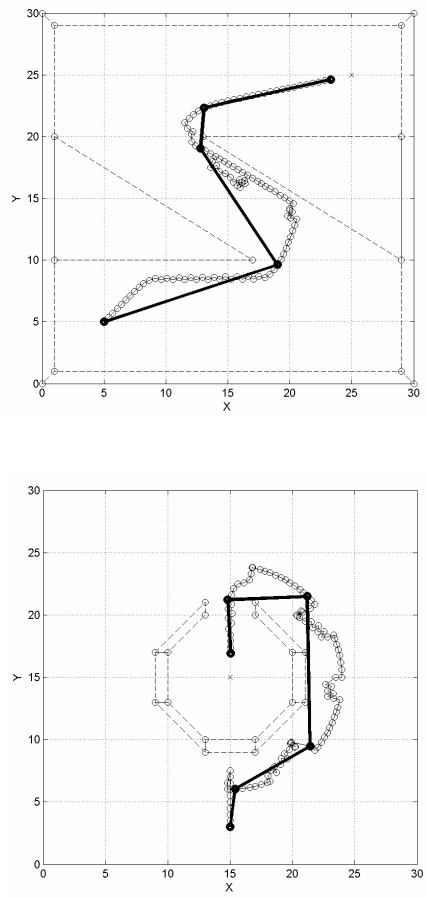
gdzie v_i i v_j to wierzchołki grafu G, K_L i K_U to stałe współczynniki oraz

$$L(v_i, v_j) = \begin{cases} \sqrt{(x[v_i] - x[v_j])^2 + (y[v_i] - y[v_j])^2} & if \quad \lambda \cap P = \varnothing \\ \infty & if \quad \lambda \cap P \neq \varnothing \end{cases} \tag{8}$$

gdzie $x[v_i]$ i $y[v_i]$ to współrzędne punktów trajektorii odpowiadającemu wierzchołkowi v_i, λ to odcinek łączący punkty trajektorii reprezentowane przez wierzchołki v_i i v_j, P to zbiór wszystkich punktów należących do przeszkód oraz

$$MaxU(u, v) = \max_{(x,y)\in\lambda} \{U(x, y)\} \tag{9}$$

Poniższa ilustracja przedstawia przykład działania zastosowanego algorytmu Dijsktry.



Rys. 5. Przykłady optymalizacji trajektorii wyjściowej z metody warstwicowej
Fig. 5. Examples of optimization of the resulting trajectory from contour line method

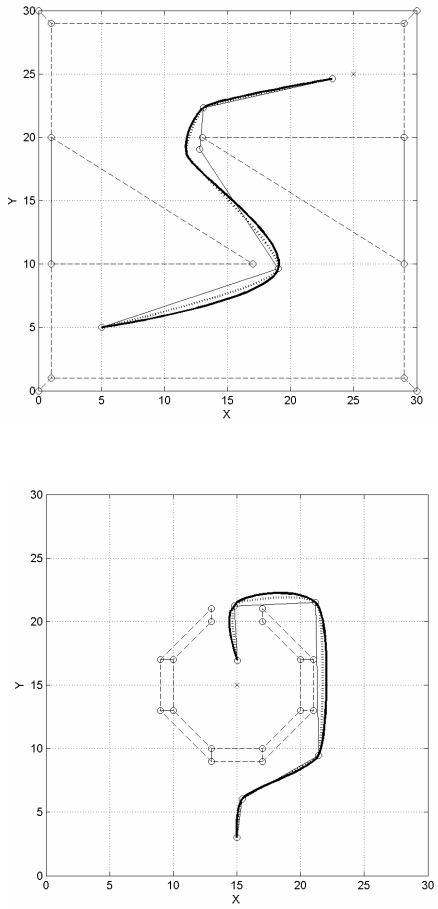
8. Wygładzenie trajektorii

Trajektoria zoptymalizowana przy pomocy algorytmu Dijkstry nie spełnia zazwyczaj ograniczeń nałożonych na minimalny promień krzywizny, który jest pożądany. W tym celu wykorzystywane są parametryczne krzywe Béziera trzeciego stopnia dane wzorem [16, 17]:

$$P = (1-t)^3 P_1 + 3(1-t)^2 t P_2 + 3(1-t) t^2 P_3 + t^3 P_4, \tag{10}$$

MOST WIEDZY Pobrano z mostwiedzy.pl

gdzie $t \in <0,1>$ to parametr krzywej, P_1, P_2, P_3 i P_4 to tzw. punkty kontrolne. Promień krzywizny trajektorii złożonej z krzywych Béziera może być łatwo zmieniany poprzez zmianę położenia punktów kontrolnych krzywych składowych. Zaprojektowany algorytm iteracyjnie „nagina" trajektorię w istotnych punktach aż do momentu, gdy trajektoria wynikowa będzie spełniała wymagania na wartość minimalnego promienia krzywizny R_{MIN}.



Rys. 6. Wynik działania algorytmu składającego trajektorię z krzywych Béziera dla wartości R_{MIN}=0.5 (linia kropkowana) i R_{MIN}=1.2 (linia ciągła)
Fig. 6. Results of algorithm responsible for constructing trajectory using Bézier curves for R_{MIN}=0.5 (dotted line) i R_{MIN}=1.2 (solid line)

9. Wnioski

W pracy porusza się problem konstruowania algorytmów autonomicznego sterowania bezzałogowymi pojazdami. Uwagę skupiono na planowaniu trajektorii ruchu w przestrzeni 2D na podstawie informacji o położeniu przeszkód znajdujących się w ograniczonym obszarze analizowanej przestrzeni operacyjnej, którą wydobywa się z mapy terenu pokonywanego przez zespół robotów. Procedura wytyczania trajektorii referencyjnej dla zespołu pojazdów oparta została na metodzie warstwicowej. Działanie algorytmów składowych zostało zilustrowane wynikami pochodzącymi z symulacji komputerowych.

Prezentowana metoda trajektorii referencyjnej jest jedynie pierwszym krokiem wieloetapowego zadania sterowania grupą pojazdów autonomicznych. Inne ograniczenia i czynniki, takie jak rozmiar grupy pojazdów i zmiana jej kształtu na trajektorii optymalnej są przedmiotem dalszych badań [18].

10. Literatura

[1] B.T. Clough: Unmanned Aerial Vehicles: Autonomous Control Challenges, A Researcher's Perspective. In: Cooperative Control and Optimization (R. Murphey, P. Parandalos), Kluwer Academic Publishers, Hingham, MA, USA, 2002.
[2] D.P. Gillen, D.R. Jacques: Cooperative Behavior Schemes form Improving the Effectiveness of Autonomous Wide Area Munitions. Cooperative Control and Optimization (R. Murphey, P. Parandalos), Kluwer Academic Publishers, Hingham, MA, USA, 2002.
[3] T.Balch, R.C. Arkin: Behavior-based formation control for multirobot teams. IEEE Transactions on Robotics and Automation, Vol. 14, No. 6, 1998.
[4] Y. Guo., L.E. Parker: A distributed and optimal motion planning approach for multiple mobile robots. Proceedings of the 2002 IEEE International Conference on Robotics and Automation (ICRA), Washington (USA), 2002.
[5] E. Bicho, S. Monteiro: Formation control for multiple mobile robots: a non-linear attractor dynamics approach. Proceedings of the IEEE/RSJ Intern. Conference on Intelligent Robots and Systems, 2003.
[6] A.J. Healey: Application of Formation Control for Multi-Vehicle Robotic Minesweeping. IEEE CDC Conference, paper no. CDC 01-INV3103, 2001.
[7] L. Loo, E. Lin, M. Kam, P. Varshney: Cooperative Multi-Agent Constellation Formation Under Sensing and Communication Contraints. In: Cooperative Control and Optimization (R. Murphey, P. Parandalos), Kluwer Academic Publishers, Hingham, MA, USA, 2002.
[8] R. Fierro, P. Song, A. Das, V. Kumar: Cooperative Control of Robot Formations. In: Cooperative Control and Optimization (R. Murphey, P. Parandalos), Kluwer Academic Publishers, Hingham, MA, USA, 2002.
[9] O. Khatib: Real-time obstacle avoidance for manipulators and mobile robots. Proceedings of the IEEE International Conference on Robotics and Automation, 1985.
[10] T.D. Barfoot T.D., C.M. Clark: Motion Planning for Formations of Mobile Robots. Robotics and Autonomous Systems, vol. 46, no. 2, 2003.
[11] J. Latombe: Robot Motion Planning, Chapter 7, Kluwer Academic Publishers, Amsterdam, 1991.
[12] F. Cuesta, A. Ollero: Intelligent Mobile Robot Navigation. Springer Tracts in Advanced Robotics, Vol. 16, Berlin, 2005.
[13] T. Niemuller, S. Widyadharma: Artificial Intelligence – An Introduction to Robotics, 2003.
[14] Franz Aurenhammer: Voronoi Diagrams - A Survey of a Fundamental Geometric Data Structure. ACM Computing Surveys, 1991.
[15] T.H. Carmen, Ch. E. Leiserson, R.L. Rivest.: Wprowadzenie do Algorytmów. Wydawnictwa Naukowo-Techniczne, Warszawa, 1997.
[16] R. Duda, P.Hart: Pattern Classification and Scene Analysis. John Wiley and Sons, New York, 1973,
[17] Paul Bourke: Bézier Curves. 1993.
[18] Z. Kowalczuk, K. Duzinkiewicz: Wyznaczanie trajektorii ruchu zespołu robotów mobilnych w środowisku z przeszkodami. W: Inteligentne wydobywanie informacji w celach diagnostycznych (Z. Kowalczuk, B. Wiszniewski), PWNT, Gdańsk, 2007.

Artykuł recenzowany

MOST WIEDZY Pobrano z mostwiedzy.pl