

doi:10.15199/48.2024.10.38

Wieloetapowy algorytm naprowadzania na cel obiektu BSP

Streszczenie. Artykuł przedstawia kluczowe cechy i wyniki testów autorskiego wieloetapowego algorytmu naprowadzania na cel powietrzny obiektu BSP. Kluczowymi cechami opracowanej metody naprowadzania są: możliwość jej wykorzystania dla różnych platform BSP, elastyczność pod względem wykorzystanego sprzętu komputerowego oraz skuteczność działania potwierdzona poprzez pozytywne rezultaty testów w warunkach symulowanych oraz rzeczywistych.

Abstract. The following text presents the key features and test results of the developed multistage algorithm for homing an airborne target of a BSP object. The key features of the developed homing method, are: the possibility of its use for different BSP platforms, flexibility in terms of the computer hardware used, and the effectiveness of operation confirmed by positive test results in simulated and real conditions. (**Hybrid BSP object target guidance algorithm**).

Słowa kluczowe: BSP, naprowadzanie, kamera stereo, przewidywanie DIS

Keywords: UAV, guidance, stereo camera, dead reckoning

Wstęp

W niniejszym artykule przedstawiono kluczowe cechy algorytmu naprowadzania na manewrujący [1] cel powietrzny obiektu BSP (Bezzałogowego Systemu Powietrznego).

Przedstawiony algorytm składa się z 3 kluczowych etapów, którymi są: etap pościgu, etap przejściowy oraz etap przechwycenia. Algorytm jest określany algorytmem wieloetapowym, ponieważ od startu BSP z naziemnej platformy do przechwycenia wrogiego celu manewrującego wykorzystuje określone etapy.

Etap fazy pościgu zakłada dostęp do danych o położeniu naprowadzanego BSP oraz celu np. danych z radaru [2]. Efektem działania tego etapu algorytmu jest doprowadzanie naprowadzanej jednostki do celu w tylnej półsfery tak, aby umożliwić wykorzystanie danych z kamery w kolejnych etapach.

Kolejnym etapem jest etap fazy przejściowej, w którym zostanie skalibrowana kamera przy wykorzystaniu techniki odwróconej kamery stereo, co umożliwi szacowanie odległości pomiędzy obiektami, wykorzystując dane jedynie z kamery monoskopowej [3]. Takie rozwiązanie zapewnia uniwersalność metody pod kątem implementacji jej na różnych platformach typu BSP.

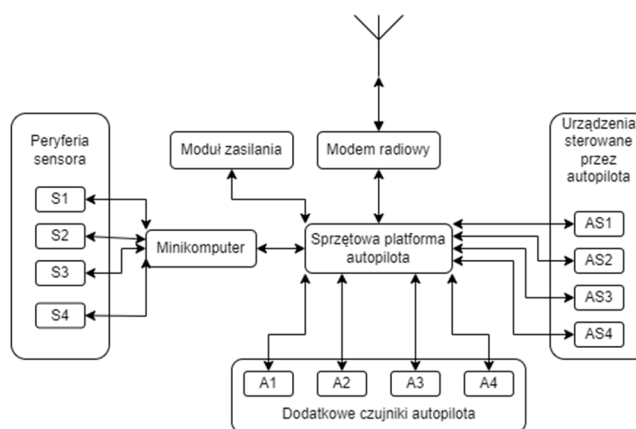
Ostatnim etapem algorytmu jest etap fazy przechwycenia, w którym to naprowadzane BSP przy wykorzystaniu zebranych danych pomiarowych z poprzedniego etapu jest kierowane na cel jedynie za pomocą danych z kamery.

Takie podejście, które zostanie szczegółowo opisane w dalszej części pracy, umożliwi wykorzystanie różnych efektorów na pokładzie naprowadzanej jednostki, mogących posłużyć do przejścia kontroli nad celem, wyprowadzenia go ze strefy kontrolowanej oraz jego neutralizacji w bezpiecznych i kontrolowanych warunkach [4]. Dzięki możliwości naprowadzania jedynie na podstawie danych z kamery efektorami mogą być wszelkiego rodzaju metody walki elektronicznej, za sprawą niezależności naprowadzanej jednostki od sygnału GPS (Global Positioning System) i RC (Remote Control) ze stacji dowodzenia.

Wprowadzenie do Bezzałogowych Systemów Powietrznych

Bezzałogowe Systemy Powietrzne, znane również jako drony, są to platformy, które umożliwiają wykonywanie zadań w powietrzu bez udziału załogi na pokładzie. BSP można traktować w kategorii systemów, gdyż składają się z

dwóch powiązanych funkcjonalnie komponentów: statku powietrznego (lub wielu statków zwanych rojem) oraz stacji dowodzenia. Statki powietrzne mogą być różnych rozmiarów i kształtów – od małych dronów po duże zaawansowane technologicznie konstrukcje. Ze względu na fakt, że BSP nie posiadają załogi na pokładzie, potrzebują systemu autonomicznego lotu [5] oraz stacji dowodzenia w roli kluczowego komponentu niezbędnego do kontroli urządzenia w powietrzu. Ponadto oprócz systemu autonomicznego lotu BSP na pokładzie posiadają również oddzielną lub wbudowaną w system autopilota jednostkę odpowiedzialną za realizację misji. Może ona być wyodrębniona fizycznie [6] bądź logicznie, natomiast na potrzeby niniejszego artykułu został przyjęty następujący model połączeń urządzeń na pokładzie naprowadzanego BSP przedstawiony na rysunku 1.



Rys. 1. Schemat połączenia urządzeń elektronicznych na pokładzie platformy bezzałogowej.

Wedle schematu, sprzętowa platforma autopilota odpowiada za sterowanie aktuatorami, czyli urządzeniami mającymi bezpośredni wpływ na lot bezzałogowca oraz komunikację ze stacją naziemną. Bezpośrednio do niej podłączone są również czujniki i urządzenia pomiarowe takie jak GPS, rurka Pitota, bądź urządzenia mierzące poprawność pracy urządzeń na pokładzie np. miernik napięcia baterii. Z jednostką autopilota połączony jest mikrokomputer służący do realizacji misji, mogący bezpośrednio wywoływać funkcję autopilota tak, aby dostosować lot do zamierzonych rezultatów. Do niego podłączone są periferia sensora, którymi podczas misji

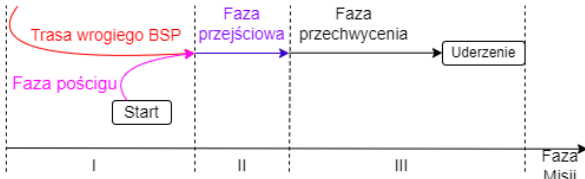
naprowadzania, będą obligatoryjnie: kamera monoskopowa [7] oraz efekторы radiowe lub akulatory efektorów fizycznych.

Plan misji realizowanej przez algorytm naprowadzania

Tytułowy algorytm zaprojektowano w sposób umożliwiający realizację misji przechwycenia i kontroli wrogich jednostek BSP. Jak wspomniano we wstępie, na misję przechwycenia składają się następujące etapy:

1. faza pościgu
2. faza przejściowa
3. faza przechwycenia

Etapy działania algorytmu zilustrowane są na rysunku 2, który służy jedynie jako ilustracja pogładowa, w rzeczywistości dopuszczalne jest ciągłe dynamiczne manewrowanie wrogiego BSP.



Rys. 2. Etapy działania algorytmu.

I etap algorytmu – faza pościgu

Naprowadzanie na cel obiektu w fazie pościgu zakłada dostęp do informacji na temat położenia naprowadzanego BSP oraz celu. Pochodzenie tych informacji w omawianym przypadku nie jest znane i nie jest istotne [2]. Zakładamy, że dane na wejściu algorytmu są w formacie (x_c, y_c, z_c) , gdzie $x_c, y_c, z_c \in U$, gdzie $U \subseteq R^+$ ponad to dane zawarte w U , są to dane przekształcone, wyrażone w metrach, po wykonaniu operacji projekcji kartograficznej.

Dodatkowo dla modelu matematycznego projektowanego algorytmu wyspecyfikowano następujące dane:

- $(C_{x_0}, C_{y_0}, C_{z_0}) \in U$ – położenie celu w czasie t-3;
- $(C_{x_1}, C_{y_1}, C_{z_1}) \in U$ – położenie celu w czasie t-2;
- $(C_{x_2}, C_{y_2}, C_{z_2}) \in U$ – położenie celu w czasie t-1;
- $(C_{x_3}, C_{y_3}, C_{z_3}) \in U$ – położenie celu w czasie t;
- $(Z_x, Z_y, Z_z) \in U$ – położenie naprowadzanego obiektu BSP w czasie t;

r_{min} – minimalny promień wirażu naprowadzanego obiektu BSP;

Δt – interwał czasu odświeżania;

d – rozdzielczość;

W prezentowanym modelu matematycznym jako zmienne decyzyjne wyznaczono:

(P_x, P_y, P_z) – punkt w przestrzeni określający wymagany azymut i elewację naprowadzanego BSP $P_x, P_y, P_z \in U$;

Jako wskaźniki modelu przyjęto:

α, LOS – kąt linii wzroku oraz długość linii wzroku [1] [8], czyli linii pomiędzy naprowadzonym BSP, a celem;

r_A – promień wirażu obiektu BSP, czyli jego promień skrętu;

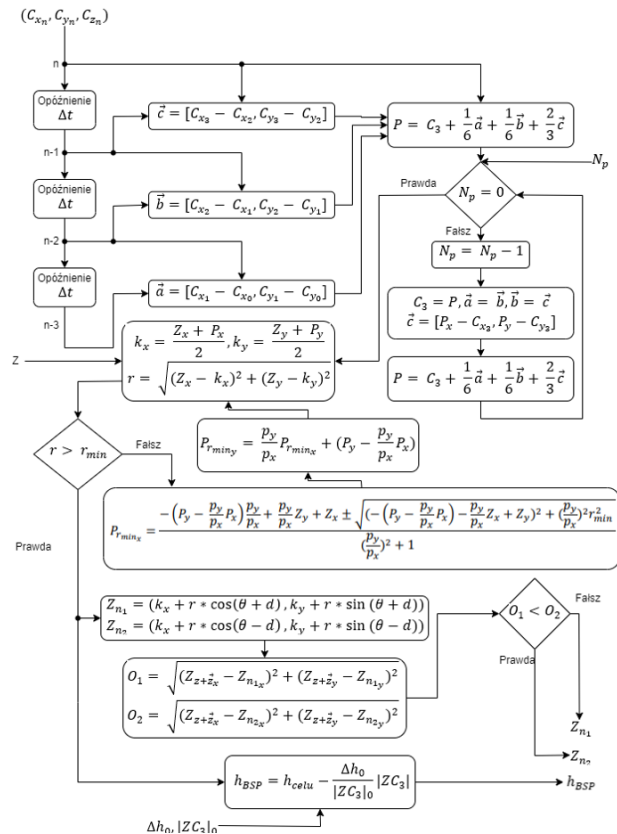
$r_{A_{min}}$ – minimalny promień wirażu, który osiąga dana konstrukcja BSP (istotne w przypadku płatowców);

Metoda doboru optymalnej trasy dla BSP, która pozwoli na wykonanie podejścia do celu z tylnej półsfery, w celu przechwycenia musi spełniać następujący warunek:

$$(1) \quad r_A > r_{A_{min}}$$

oraz zaplanowana trasa powinna być najkrótsza. W tym celu algorytm z założenia będzie dobierał trasę tak, aby wraz ze zmniejszającą się odległością pomiędzy obiektami zmniejszał się też r_A . Podobne podejście zostało naznaczone w [8].

W celu spełnienia założeń algorytm w fazie pościgu porusza się w pętli, w której na początek, za pomocą standardu symulacji DIS (Dead Reckoning Model), powszechnie wykorzystywanego w zastosowaniach wojskowych, będzie przewidywał kolejne położenie celu na podstawie informacji otrzymanych z zewnętrznego systemu kontroli przestrzeni powietrznej. Następnie, wynikowy punkt przewidywania będzie wykorzystany do wykonania obliczeń [9] mających na celu znalezienie odpowiedniego punktu pośredniego, będącego punktem trasy, którą BSP musi pokonać, aby osiągnąć zamierzony rezultat. Schemat blokowy algorytmu został przedstawiony na rysunku 3.



Rys. 3. Etap algorytmu naprowadzania fazy pościgu.

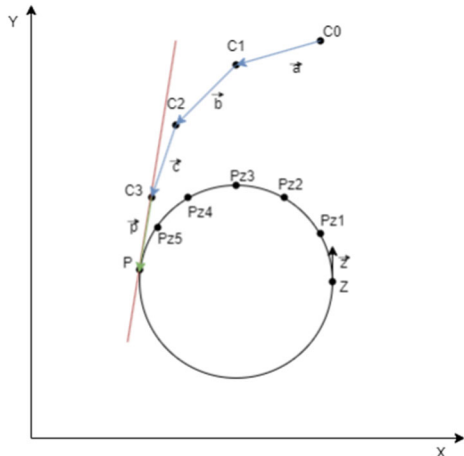
Powyższy algorytm wykonywany jest w pętli do momentu, w którym naprowadzany dron zbliży się do celu na odległość mniejszą, niż ustawiona przez użytkownika.

Na początku wykorzystana została standardowa formuła modelowania DRM (FVW) [10] przeznaczona do modelowania obiektów manewrujących poruszających się z wysoką prędkością:

$$(2) \quad P = P_0 + \vec{V}_0 \Delta t + \frac{1}{2} \vec{A}_0 \Delta t^2 + \frac{1}{6} \vec{J} \Delta t^3$$

Powyższa formuła jest natomiast wariacją formuły DRM(FVW), ponieważ jest dodany do niej człon $\frac{1}{6} \vec{J} \Delta t^3$ określający pochodną przyśpieszenia zwaną zrywem. Ten zabieg, co pokazały testy, pozwolił w bardziej dokładny sposób modelować ruch celu. Formułę po przekształceniu można wyrazić w formie wygodniejszej obliczeniowo, bazując na wektorach $\vec{a}, \vec{b}, \vec{c}$, określonych na rysunku 4, na którym punkty C0, C1, C2 i C3 odpowiadają kolejno punktom $(C_{x_0}, C_{y_0}, C_{z_0})$ $(C_{x_1}, C_{y_1}, C_{z_1})$ $(C_{x_2}, C_{y_2}, C_{z_2})$ i $(C_{x_3}, C_{y_3}, C_{z_3})$. Formuła uzyskana w wyniku przekształceń to:

$$(3) \quad P = C_3 + \frac{1}{6}\vec{a} + \frac{1}{6}\vec{b} + \frac{2}{3}\vec{c}$$



Rys. 4. Wizualizacja procesu wyboru kolejnego punktu trasy.

Kolejnym etapem tej fazy jest wyznaczenie punktów, na które naprowadzane BSP powinno obrać azymut i elewację. Metoda doboru tych punktów jest następująca:

1. Na rysunku 4 wyznaczono położenie celu P oraz położenie naprowadzanego BSP jako punkt Z;
2. Wyznaczana jest prosta pomiędzy obecnym położeniem celu C_3 oraz przewidywanym położeniem celu P;
3. Wyznaczany jest okrąg styczny do prostej w punkcie P przechodzący przez punkt Z;
4. Wyznaczane są punkty $P_{z1}, P_{z2}, P_{z3}, \dots$ na wyznaczonym okręgu z określoną rozdzielczością d (rozdzielczość jest zmniejszana wprost proporcjonalnie do odległości pomiędzy nimi);
5. Sprawdzany jest warunek $r_A > r_{A_{min}}$, jeżeli jest spełniony to następuje kontynuacja, w przeciwnym przypadku, wybierany jest inny punkt na wyznaczonej prostej;
6. Wybierany jest najbliższy z wyznaczonych punktów punktowi Z;
7. Wyznaczana jest wysokość na jakiej naprowadzane BSP powinno się znaleźć w wyznaczonym punkcie tak aby zachować odpowiednią proporcję odległości pomiędzy obiektami, a wysokością [1].

Podczas tego etapu prędkość naprowadzanego BSP jest domyślnie ustawiona na najszybszą możliwą.

II etap algorytmu – faza przejściowa

Algorytm fazy przejściowej ma na celu dobranie odpowiedniej prędkości względem celu oraz kalibrację kamery na pokładzie naprowadzanego BSP, tak aby możliwe było przejście z fazy pościgu do fazy przechwycenia. Zaprojektowany algorytm będzie modyfikacją fazy pościgu, który rozpoczyna pracę w momencie przechwycenia [8], czyli wtedy, kiedy zostaną wywołane procedury zmiany okręgu w celu spełnienia kryterium minimalnego promienia oraz kąt pomiędzy wektorami ruchu obu obiektów α będzie mniejszy niż kąt α_{max} , który jest bliski zeru.

Algorytm nie dobiera żadnej określonej trajektorii. Jego działanie polega na obieraniu azymutu i elewacji bezpośrednio na położenie celu na podstawie danych pochodzących z zewnętrznego systemu kontroli przestrzeni powietrznej. Z uwagi na możliwy kształt celu, będącego np. płowcem, należy określić w algorytmie również pewien poziom przewyżki, ponieważ płowce są lepiej widoczne pod kątem niż podczas lotu bezpośrednio za nimi. W trakcie

tego etapu kontrolowana jest również prędkość naprowadzanego BSP [1] na podstawie poniższej formuły:

$$(4) \quad v_{BSP} = v_{celu} + \frac{\Delta v_0}{|ZC_3|_{min}} |ZC_3|$$

gdzie,

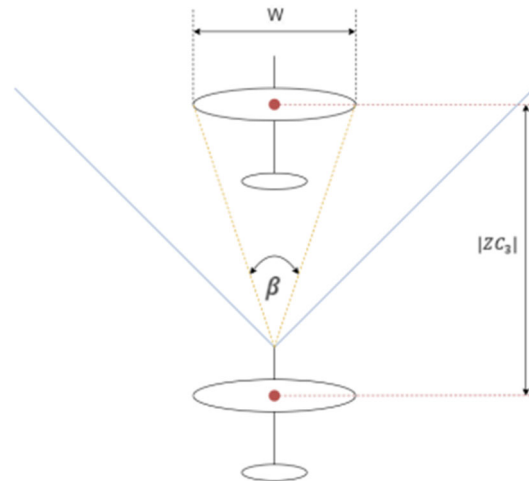
Δv_0 – różnica prędkości między obiektami, kiedy odległość wynosi $|ZC_3|_{min}$;

v_{celu} – prędkość celu w punkcie C_3 .

Do skalibrowania kamery należy na początku zidentyfikować cel w kamerze. Zakładając, że innymi jednostkami są wrogie BSP, będące płowcami, można przyjąć za cel najszerzy kształt na obrazie z kamery w jednolitym kolorze.

W celu zidentyfikowania wrogiej jednostki zostanie wykorzystany algorytm detekcji krawędzi Canny'ego [11] co umożliwi szybką identyfikację obiektu na obrazie z kamery oraz skuteczne określanie jego konturów. Celem jego użycia, obraz zostanie przekształcony z barw BGR na barwy HSV [12]. Następnie, zostanie odczytany kolor najszerzego kształtu, który będzie uznawany za skrzydło wrogiej BSP. Ta barwa zostanie zapisana i na podstawie szerokości obiektu o tym kolorze na obrazie z kamery zostanie dokonana kalibracja.

Kamera kalibrowana w obecnym etapie algorytmu naprowadzania będzie w kolejnym etapie wykorzystana do określania odległości pomiędzy obiektami za pomocą odwróconej kamery stereo [13], jak na rysunku 5.



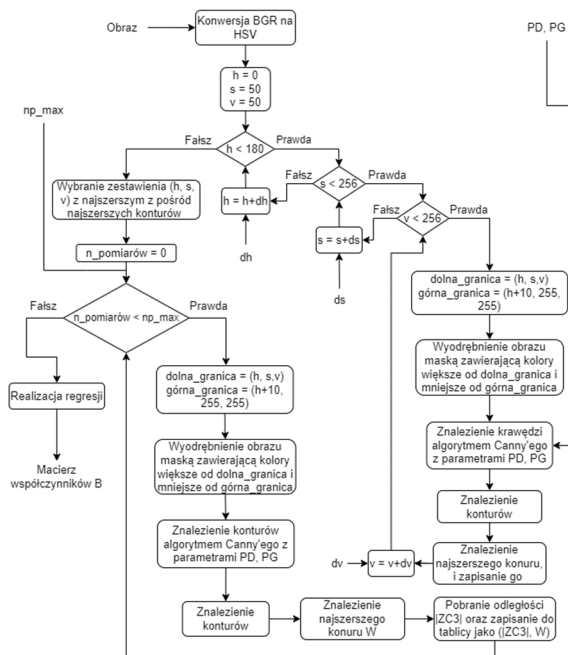
Rys. 5. Wizualizacja założenia odwróconej kamery stereo.

Założenie odwróconej kamery stereo [13] zakłada, że wraz ze wzrostem odległości pomiędzy dronami, oznaczonej jako $|ZC_3|$, zmniejsza się szerokość celu widziana w kamerze oznaczona jako W poprzez zmniejszenie kąta β . Zmniejszenie to odbywa się według następującej zależności.

$$(5) \quad 2|ZC_3| = \operatorname{tg}\left(\frac{\beta}{2}\right)W$$

W związku z powyższym, kalibracja będzie polegała na zebraniu odpowiednio dużego zbioru danych tak, aby za pomocą metod regresji, w czasie rzeczywistym, określić zmianę odległości między obiektami na podstawie zmiany szerokości. Podczas przeprowadzonych testów, dobre rezultaty dała regresja liniowa [14], natomiast podczas implementacji jest możliwość dostosowania metody regresji w zależności od potrzeb.

Część algorytmu naprowadzania na cel obiektu BSP w etapie II, została przedstawiona na rysunku 6.



Rys. 6. Algorytm kalibracji kamery fazy przejściowej.

III etap algorytmu – faza przechwycenia

Ostatnim etapem jest faza przechwycenia, w której to algorytm na podstawie modelu uzyskanego w wyniku regresji podąża za wrogą jednostką wykorzystując jedynie dane z kamery niezależnie od danych z zewnętrznego systemu kontroli przestrzeni powietrznej. Dzięki temu, na pokładzie naprowadzanej jednostki można wykorzystać efekty walki elektronicznej, które nie będą miały wpływu na skuteczność opracowanego algorytmu naprowadzania BSP na cel. Takie podejście pozwoli również na podążanie za wrogą jednostką do czasu aż nie opuści strefy kontrolowanej, a nawet do momentu powrotu do jej operatorów celem ich lokalizacji. Sytuacja spowodowana pracą algorytmu pozwala również na bezpieczną neutralizację wrogiego BSP poprzez przejście nad nim kontroli i dokonanie neutralizacji w wybranym miejscu.

Algorytm ostatniej fazy również wykorzystuje algorytm Canny'ego każdorazowo wybierając już wcześniej ustalony kolor z przestrzeni barw HSV. Algorytm rozpoznaje obiekt na obrazie z kamery oraz dostosowuje azymut i elewację tak, aby utrzymać go na środku obrazu z kamery. W tym celu zostały wyprowadzone następujące formuły:

- Dla kierunku poziomego:

$$(6) \quad \varepsilon_{poz} = \arctg\left(\frac{e_{poz}}{h}\right)$$

- Dla kierunku pionowego:

$$(7) \quad \varepsilon_{pion} = \arctg\left(\frac{e_{pion}}{h}\right)$$

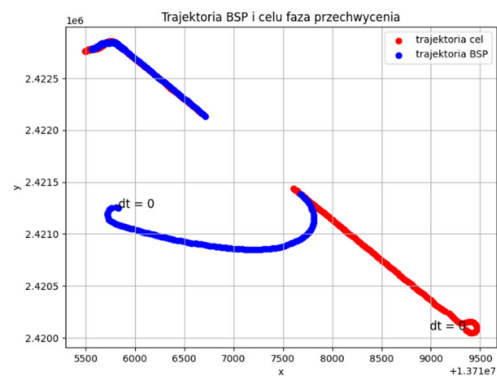
gdzie: ε_{poz} – wyrażona w stopniach wartość, o którą powinno naprowadzane BSP zmienić kąt w płaszczyźnie poziomej (azymut); ε_{pion} – wyrażona w stopniach wartość, o którą powinno naprowadzane BSP zmienić kąt w płaszczyźnie pionowej (elewacja); e_{poz} – odległość wyrażona w pikselach celu na kamerze w płaszczyźnie poziomej od środka obrazu; e_{pion} – odległość wyrażona w pikselach celu na kamerze w płaszczyźnie pionowej od środka obrazu; h - odległość pomiędzy obiektami oszacowana na podstawie szerokości celu na obrazie po

wykorzystaniu zależności z regresji wyznaczonej w etapie II.

Wyniki działania algorytmu w symulacji i w warunkach rzeczywistych

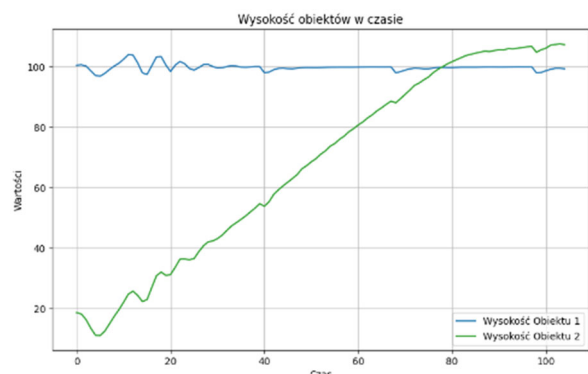
Testy symulacyjne [9] zostały przeprowadzone za pomocą oprogramowania „X-Plane 11” [15] wykorzystanego na 2 komputerach w tej samej sieci lokalnej, co umożliwiło zasymulowanie 2 samolotów na tej samej mapie. Oba są kontrolowane przez maszynę wirtualną z uruchomionym oprogramowaniem „ArduPilot”, którego połączenie umożliwiło oprogramowanie „Mission Planner” [16]. Jeden z samolotów jest sterowany przez implementację algorytmu w języku „Python”, poprzez wykorzystanie biblioteki „DroneKit”.

Na rysunku 7 przedstawiono trajektorię naprowadzanego BSP (niebieski) i celu (czerwony). Z rysunku wynika że BSP poprawnie przechwycił cel. Przerwa pomiędzy kolejnymi lokalizacjami obiektów na wykresie wynika z właściwości symulacji jednowątkowej, która w tym momencie analizowała zdjęcie z kamery celem kalibracji.



Rys. 7. Trajektoria BSP i celu podczas działania wszystkich etapów algorytmu.

Na rysunku 8 przedstawiono zmianę wysokości naprowadzanego BSP w trakcie etapu pościgu. Wysokość BSP w poprawny sposób zmienia się względem wysokości celu proporcjonalnie do promienia odległości. W około 100. s symulacji BSP znajdują się około 10 m nad celem co jest ustawioną wartością przewyżki potwierdzając poprawne działanie algorytmu.



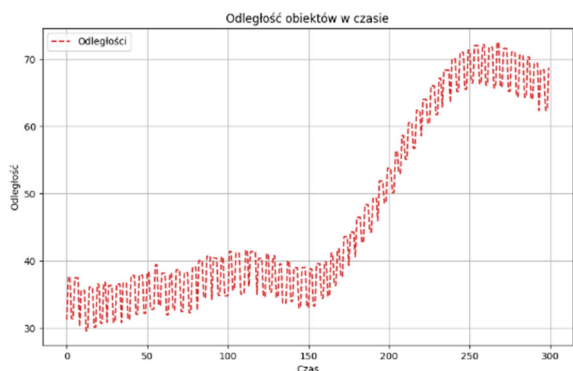
Rys. 8. Zmiana wysokości naprowadzanego BSP względem celu.

Na rysunku 9 przedstawiono wynik działania etapu przejściowego na którym rozpoznano cel. Rozpoznany obiekt został oznaczony czerwoną ramką. Kształtem jest główne skrzydło celu. Poniższy rysunek potwierdza poprawne działanie rozpoznawania celu.



Rys. 9. Rozpoznany obiekt celu w symulatorze X-Plane 11.

Na rysunku 10. przedstawiono zmianę odległości pomiędzy obiektami podczas etapu przechwycenia. Na wykresie można zauważyć, że algorytm zareagował w sposób prawidłowy, o czym świadczy tendencja spadkowa w okolicach 270. sekundy.



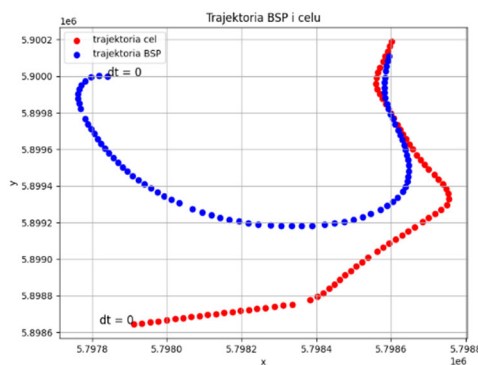
Rys. 10. Zmiana odległości pomiędzy obiektami podczas etapu przechwycenia.

Algorytm został również przetestowany w warunkach rzeczywistych za pomocą samolotu zdalnie sterowanego „Multiplex Blizzard” oraz BSP „Interceptor”, czyli zwycięskiego projektu konkursu MON na zaprojektowanie BSP/BSL/BSM (Bezzałogowy System Powietrzny/Lądowy/Morski), w kategorii bojowej, pokazany na rysunku 11.



Rys. 11. BSP „Interceptor”, zwycięski projekt konkursu MON na BSP/BSL/BSM 2023 r.

Na rysunku 12 przedstawiono trajektorie lotu naprowadzanego BSP oraz celu podczas przejścia przez wszystkie etapy algorytmu. Algorytm poprawnie doprowadził do przechwycenia oraz kontynuacji lotu za celem.



Rys. 12. Trajektoria lotu naprowadzanego BSP i celu w warunkach rzeczywistych.

Na rysunku 13. przedstawiono wynik rozpoznania celu w fazie przejściowej w warunkach rzeczywistych. Podczas tej fazy działania algorytmu została analizowana jeszcze możliwość wykorzystania sztucznej inteligencji, która w tym przypadku okazała się znacznie bardziej zawodna niż zaproponowana autorska metoda rozpoznania, z uwagi na niską jakość nagrania z kamery w różnych warunkach oświetleniowych. Dzięki temu zaproponowana metoda rozpoznania pozwala wykorzystać sprzęt gorszej jakości co przekłada się bezpośrednio na uniwersalność jej wykorzystania.



Rys. 13. Rozpoznany obiekt celu w warunkach rzeczywistych.

Zaprezentowane wyniki działania algorytmu, zarówno w symulacji, jak i w warunkach zbliżonych do rzeczywistych, świadczą o skutecznym działaniu zaimplementowanego algorytmu. Szczególnie istotnym wynikiem jest możliwość detekcji wrogiego bezzałogowca na obrazie z kamery bez potrzeby wykorzystywania algorytmów sztucznej inteligencji, które przy niskiej jakości obrazu z kamery mogłyby nie dać zadowalających rezultatów.

Podsumowanie

Celem artykułu było przybliżenie kluczowych cech wieloetapowego algorytmu naprowadzania na cel obiektu BSP, umożliwiającego przechwytywanie i neutralizowanie wrogich jednostek bezzałogowych w kontrolowanych obszarach. W ramach pracy opracowano trzy etapy tytułowego algorytmu: pierwszy pozwalający na optymalizację trajektorii lotu BSP do celu, tak aby umożliwić przechwycenie, drugi i trzeci wykorzystujące rozwiązanie odwróconej kamery stereo dla uniezależnienia przechwycenia od zewnętrznego systemu kontroli przestrzeni powietrznej. Algorytm pozwala na jego łatwą rozbudowę i modernizację, oraz sprawne dostosowywanie jego kluczowych parametrów do zmiennych warunków i naprowadzanych platform BSP, co stanowi solidną bazę dla dalszego rozwoju w zakresie przeciwdziałania bezzałogowym systemom powietrznym.

Podziękowania dla dr. inż. Pawła Skokowskiego za opiekę naukową oraz dla Wydziału Elektroniki WAT.

~ inż. Maciej Kawka

Autorzy: inż. Maciej Kawka, Wojskowa Akademia Techniczna, Wydział Cybernetyki, Instytut Systemów Informatycznych, ul. gen. Sylwestra Kaliskiego 2, 00-908 Warszawa, E-mail: maciej.kawka01@student.wat.edu.pl;

LITERATURA

- [1] Al-Qudsi YS., El-Bayoumi GM: *Intercept Algorithm for Maneuvering Targets Based on Differential Geometry and Lyapunov Theory*, INCAS BULLETIN, 10, 175-192, 01.03.2018.
- [2] Rahman S., Robertson, D.A., *Radar micro-Doppler signatures of drones and birds at K-band and W-band*. Sci Rep 8, 17396, 2018
- [3] Jana S., Tony LA., Bhise AA., Varun VP., Ghose D.: *Interception of an aerial maneuvering target using monocular vision*, Robotica, 40, 4535-4554, 03.08.2022.
- [4] Razzoqi O.D., Ali A.H.: *Drones neutralized by utilize electromagnetic pulse (EMP) system*. Proceedings of the 2022 5th International Conference on Engineering Technology and its Applications (IICETA), 31.05.2022
- [5] Gan Z. F. T., Feng K., Khouli F.: *Development and Optimization of Flight Dynamics, Control Laws and Avionics System for UAV with a Multi-scale Optimized Blended Wing Body Configuration*. CASI AERO 2019, Laval (Quebec), 14.06.2019
- [6] <https://ardupilot.org/dev/docs/companion-computer-nvidia-tx2.html> [dostęp: 30.03.2024]
- [7] Grzyb M., Stefański K.: *Naprowadzanie bomby kierowanej z wykorzystaniem specjalnego algorytmu dla identyfikacji i śledzenia celu*. Pomiary Automatyka Robotyka, 15 (2), 668-676, 06.04.2011
- [8] Looker JR.: *Minimum Paths to Interception of a Moving Target When Constrained by Turning Radius*, Defence Science and Technology Organization Air Operations Division, Edinburgh (Australia), 01.12.2008.
- [9] Li K., Su W., Chen L.: *Performance analysis of three-dimensional differential geometric guidance law against low-speed maneuvering targets*. Astrodynamics, 2, 233-247, 21.06.2018
- [10] <https://open-dis.github.io/dis-tutorial/DeadReckoningStateUpdate> [dostęp: 22.01.2024]
- [11] Zhang X., Kusri K., *Autonomous long-range drone detection system for critical infrastructure safety*, Multimedia Tools and Applications, 80(9), 23723-23743, 12.02.2021
- [12] Hearn D., Baker M.P., Carithers W.: *Computer Graphics with OpenGL*. Wydawnictwo Prentice Hall, USA, 4th Edition, 2010.
- [13] O'Riordan A., Newe T., Toal D.J.F., Dooly G.: *Stereo Vision Sensing: Review of existing systems*. Proceedings of the International Conference on Sensing Technology, University of Limerick, 04.12.2024
- [14] Maulud D., Abdulazeez A.M.: *A Review on Linear Regression Comprehensive in Machine Learning*. Journal of Applied Science and Technology Trends, 1(4), 140-147, 31.12.2020
- [15] Gan Z. F. T., Feng K., Khouli F.: *Development and Optimization of Flight Dynamics, Control Laws and Avionics System for UAV with a Multi-scale Optimized Blended Wing Body Configuration*. CASI AERO 2019, Laval (Quebec), 14.06.2019
- [16] <https://ardupilot.org/dev/docs/sitl-with-xplane.html> [dostęp: 30.03.2024]