

**Wydział Elektrotechniki, Automatyki, Informatyki i Inżynierii Biomedycznej**

Projekt dyplomowy

*Optymalizacja trajektorii dronów*

*Drone trajectory optimization*

Autor: *Maciej Morgalla*

Kierunek studiów: *Automatyka i Robotyka*

Opiekun pracy: *dr inż. Piotr Kadłuczka*

Kraków, 2022

Spis treści

[1. Wstęp 3](#_Toc91760549)

[1.1. Wprowadzenie 3](#_Toc91760550)

[1.2. Cel i zakres pracy 5](#_Toc91760551)

[2. Rozważany problemy i metody jego rozwiązania 6](#_Toc91760552)

[2.1. Sposoby użycia dronów 6](#_Toc91760553)

[2.2. Problemy optymalizacyjne w sterowaniu dronami 8](#_Toc91760554)

[2.3. Stosowane algorytmy 9](#_Toc91760555)

[3. Formalizacja modelu matematycznego 11](#_Toc91760556)

[3.1. Podstawowe parametry symulacji 11](#_Toc91760557)

[3.2. Wyznaczanie kolejnych pozycji dronów 12](#_Toc91760558)

[3.3. Kolizje 20](#_Toc91760559)

[3.4. Funkcja celu 20](#_Toc91760560)

[4. Algorytm przeszukiwania z zabronieniami 22](#_Toc91760561)

[4.1. Schemat algorytmu 22](#_Toc91760562)

[4.2. Pamięć krótko- i długoterminowa 24](#_Toc91760563)

[5. Opis aplikacji 25](#_Toc91760564)

[5.1. Środowisko programistyczne 25](#_Toc91760565)

[5.2. Proces powstawania i testowania kodu 26](#_Toc91760566)

[6. Scenariusze testowe 28](#_Toc91760567)

[6.1. Optymalna liczba dronów 28](#_Toc91760568)

[6.2. Prędkość maksymalna dronów 28](#_Toc91760569)

[6.3. Wyrównywanie czasu lotu 28](#_Toc91760570)

[6.4. Maksymalizacja oddalenia dronów 28](#_Toc91760571)

[7. Podsumowanie 28](#_Toc91760572)

[8. Bibliografia 28](#_Toc91760573)

# Wstęp

## Wprowadzenie

Bezzałogowe statki powietrzne (szerzej znane jako drony) to, jak sama nazwa wskazuje, statek powietrzny niewymagający obecności załogi na jego pokładzie oraz niemogący zabrać na swój pokład pasażerów [1]. W zależności od przeznaczenia mogą być duże jak samoloty pasażerskie lub tak małe, że można schować je do kieszeni. Dzięki ich znakomitej mobilności mogą być wykorzystywane do wielu zadań (więcej w rozdziale 2.1).

Pomimo, iż drony dopiero od niedawna są powszechnie spotykane, to pierwsze konstrukcje pojawiły się jeszcze w pierwszej połowie XX wieku. W 1920 roku Etienne Oehmichen stworzył kwadrokopter, który przeleciał rekordowe na tamte czasy 360 metrów [2]. Nie był to jednak dron, bowiem Oehmichen osobiście nim sterował. Konstrukcja ta przypominała rozwiązaniami technicznymi obecne modele dronów. Wraz z rozwojem technologii ich produkcja stawała się coraz łatwiejsza. W ostatniej dekadzie stały się szczególnie popularne, czemu sprzyjała duża różnorodność modeli dostępnych na rynku.

Obraz zawierający tekst, samolot, transport

Opis wygenerowany automatycznie

*Rysunek 1.1 Kwadrokopter Etienne’a Oehmichena z 1922r.* [3]

Rosnąca z roku na rok liczba wyprodukowanych dronów pozwala na wykorzystanie ich na dużą skalę. Przykładem mogą być ceremonie otwarcia Igrzysk Olimpijskich w Pjongczangu w 2018 roku oraz w Tokio w 2021r. W samym Pjongczangu użyto 1 218 dronów do stworzenia spektaklu świetlnego, którego tłem było nocne niebo.



*Rysunek 1.2 Ceremonia otwarcia Letnich Igrzysk Olimpijskich w Tokio w 2021 r.* [4]

Wykorzystanie większej liczby dronów do celów artystycznych wymaga rozwiązania szeregu problemów. Należy odpowiednio wytyczyć trajektorie wszystkich dronów, aby nie dopuścić do kolizji oraz aby każdy dron znalazł się w swoim punkcie docelowym. Ponadto na drony wpływają warunki atmosferyczne, a czas ich lotu bez ładowania akumulatora jest ograniczony. W związku z tym aby prawidłowo zaprojektować taki spektakl trzeba wziąć pod uwagę wiele kryteriów i zmiennych. Duża czasochłonność obliczeń jakie należałoby wykonać wymusza użycia do tego celu jednostek obliczeniowych. W ramach niniejszego projektu zaproponowany zostanie sposób na rozwiązanie tego problemu.

## Cel i zakres pracy

Niniejsza praca ma na celu określenie trajektorii roju dronów pomiędzy dwoma zadanymi konfiguracjami, przy zadanych ograniczeniach przestrzennych i technicznych oraz optymalizację przydziału punktów docelowych do dronów pod kątem ustalonej funkcji kryterialnej. Na potrzeby pracy zdefiniowano model matematyczny problemu w przestrzeni trójwymiarowej. Opracowano również algorytm przybliżony, z uwzględnieniem specyfiki problemu.

Zakres pracy obejmuje:

* przegląd informacji związanych z dronami i algorytmami,
* formalizację modelu matematycznego na podstawie uzyskanych informacji,
* opracowanie algorytmu przybliżonego określającego optymalny przydział punktów docelowych do dronów pod kątem funkcji celu,
* napisanie programu realizującego w ramach symulacji model matematyczny oraz programu realizującego algorytm,
* przeprowadzenie eksperymentów obliczeniowych dla zbioru reprezentatywnych zadań testowych,
* sformułowanie wniosków na temat efektywności algorytmu oraz własności problemu.

# Rozważany problemy i metody jego rozwiązania

## Sposoby użycia dronów

Pędzący rozwój technologiczny pozwala w dzisiejszych czasach na realizowanie projektów, które do tej pory były niemożliwe, bądź skrajnie trudne do wykonania. Pomimo bogatej historii bezzałogowych statków powietrznych, dopiero w ubiegłej dekadzie zyskały one na popularności. Ich powszechność z pewnością jest katalizatorem wielu inspiracji do wykorzystania ich w praktyce, a nie tylko do zabawy.

Najczęściej wykorzystuje się zdolność dronów do obserwowania obiektów z dużej wysokości. Jest to szczególnie ważne, gdy obserwowany obszar ma znaczną powierzchnię, bądź jest niezwykle trudno dostępny. Dlatego idealnie nadają się do pomocy w chociażby leśnictwie. Umożliwiają wtedy m. in. obserwację migracji zwierząt, badanie wilgotności, tropienie kłusowników oraz lokalizację martwych drzew. W przypadku straży pożarnej stają się doskonałym środkiem do kontrolowania pożarów (np. lasów). [5]

Docieranie do trudno dostępnych miejsc można wykorzystać także w inny sposób. Rozwój technologii pozwala obecnie na przesyłanie dronami niewielkich przesyłek, tam gdzie dotarcie drogą lądową byłoby niemożliwe lub niezwykle czasochłonne. Ponownie należy przytoczyć przykład leśnictwa, gdyż szczepienie zwierząt zrzucanymi na ziemię szczepionkami ukrytymi w pożywieniu jest doskonałym na to przykładem. Jednym z bardziej niecodziennych rozwiązań jest użycie drona do rzucenia koła ratunkowego tonącemu. Skuteczność takiego ratunku byłaby co najmniej wątpliwa. [5]

Niektóre wielkie koncerny nie zamierzają poprzestać na użyciu bezzałogowych statków powietrznych jedynie w ekstremalnych sytuacjach. Firma Amazon w 2013 roku ogłosiła plany co do wprowadzenia poczty lotniczej. Polegałaby na dostarczeniu przez drona produktu z magazynu pod drzwi zamawiającego. Podobnie swoje usługi w tej materii zaczyna wprowadzać firma Google. Wygląda więc na to, że nieuniknionym elementem przyszłości będzie przesyłanie paczek drogą powietrzną. Można jedynie zastanowić się jak bardzo zatłoczona będzie wtedy przestrzeń powietrzna. [2]

Jednakże nie tylko cele praktyczne muszą determinować wykorzystanie dronów. Jednymi z najchętniej oglądanymi na całym świecie spektaklami są ceremonie otwarcia igrzysk olimpijskich. Nie dziwi więc, że w każdej kolejnej edycji organizatorzy stają na głowie aby jak najbardziej je uatrakcyjnić, pomagając sobie najnowszymi nowinkami technologicznymi. Nie inaczej było na ceremoniach otwarcia igrzysk w Pjongczangu, czy też w Tokio. W samym Pjongczangu do niezwykłego spektaklu świetlnego użyto 1 218 dronów, które formując przeróżne konstelacje, za pomocą zainstalowanych źródeł światła, formowały kształty, które z pewności zachwycały zgormadzonych widzów.



*Rysunek 2.1 Ceremonia otwarcia zimowych igrzysk olimpijskich w Pjongczang* [6]

## Problemy optymalizacyjne w sterowaniu dronami

Sterowanie dronami nie jest prostym zadaniem. Jak powszechnie wiadomo ich lot jest możliwy dzięki zamontowanymi na nich wirnikom. Ich liczba zależy głównie od producenta, jednakże najczęściej drony posiadają cztery wirniki. Tutaj niestety pojawia się pewien problem, O ile lot tylko w osi *z* (wysokości) nie wymaga szczególnych działań, to już lot w płaszczyźnie poziomej wymaga, aby wirniki miały różne prędkości obrotowe. To zaś może powodować różnicę pomiędzy zadaną trajektorią drona, a rzeczywistą trajektorią [7]. Dlatego konieczne jest aby zachowywać określony minimalny dystans pomiędzy dronami.

Dodatkowym czynnikiem przemawiającym za zachowywaniem dystansu jest skutek pracy wirników. Wytwarzanie siły nośnej jest możliwe tylko dzięki szybkiemu „wypychaniu” powietrza, dającemu potrzebny efekt odrzutu. Może to powodować nietypowe (turbulentne) ruchy powietrza zarówno powyżej jak i poniżej drona. Turbulencje powodują zaburzenia w przypływie powietrza przez wirniki, co może skutkować utratą kontroli. Wynikiem jest to zagrożenie dla innych obiektów latających oraz obiektów naziemnych.

Innym czynnikiem przyrodniczym wpływającym na drony jest temperatura. Tak jak wszystkie inne maszyny, posiadają one pewien zakres temperaturowy, w którym powinny pracować. Szczególnie narażone na temperatury są akumulator, oraz elementy elektroniczne. W przypadku akumulatora ma to związek z jego zdolnością do magazynowania i przekazywania energii. Dlatego lot w bardzo niskich bądź bardzo wysokich na warunki ziemskie temperaturach jest niewskazany.

Kolejną przeszkodą jest ograniczona pojemność akumulatora. Im dłużej wirniki obracają się tym bardziej zapasy energii kurczą się, tym samym ograniczając maksymalny czas jaki może spędzić w powietrzu. W dodatku po przekroczeniu pewnej granicy poziomu energii maksymalna osiągalna prędkość obrotowa wirników może być nieosiągalna. Wymusza to wymianę bądź ładowanie akumulatora w odpowiednim czasie, a także ograniczenie czasu użycia drona.

Jeszcze innym problemem jest udźwig. Jest to szczególnie palący problem w użyciu dronów do celów transportowych. Technologia ta jest wprowadzana m. in. przez firmę Amazon [8]. Wymaga jednak użycia dronów, które są w stanie wytworzyć dużą siłę ciągu potrzebną do lotu, ze względu na masę oraz opór aerodynamiczny paczki.

We wspomnianych w rozdziale 2.1 spektaklach, drony, aby utworzyć kolejny kształt, zmieniały swoje pozycje. Generuje to jednak pewien problem. Dla pojedynczego drona, w zależności od doboru jego kolejnego punktu docelowego, zmieniają się parametry trajektorii, jaką musi pokonać. W związku z tym należy w przemyślany sposób przydzielić wspomniane punkty docelowe do odpowiednich dronów.

## Stosowane algorytmy

Problem przydziału jest powszechnie spotykanym problemem optymalizacyjnym. Pojawia się on przy m.in. masowej produkcji, logistyce, a nawet zwykłym planowaniu dnia. W wielu przypadkach człowiek potrafi samodzielnie rozwiązać taki problem, szczególnie gdy nie jest zbyt skomplikowany. Są też sytuacje, w których złożoność jest na tyle trudna, że poza najtęższymi umysłami na Ziemi, nikt nie jest w stanie szybko podać rozwiązania. Wtedy z pomocą przychodzą właśnie algorytmy.

Wiele z tych problemów można przedstawić za pomocą tabel, grafów czy też funkcji liniowych. Popularnymi algorytmami i sposobami do ich rozwiązywania są w zależności od problemu: algorytm Dijkstry, algorytm węgierski, metoda Symplex, programowanie liniowe, algorytm aproksymacyjny itp.. [9]

Istnieją też tak bardzo skomplikowane problemy, że konieczne jest użycie algorytmów heurystycznych. Stosowane są szczególnie w przypadku problemów, których rozwiązanie wymagałoby ogromnej złożoności obliczeniowej, czy też dużej ilości czasu. Ich właściwością jest bowiem skrócenie czasu potrzebnego na obliczenia, możliwym kosztem jakości rozwiązania. Wśród algorytmów heurystycznych możemy znaleźć takie algorytmy jak: algorytm przeszukiwania z zabronieniami (Tabu Search - TS), algorytmy genetyczne oraz algorytm symulowanego wyżarzania. [10]

Przykładami użycia jednego z algorytmów heurystycznych – algorytmu przeszukiwania z zabronieniami – są następujące problemy:

* planowanie tras [11],
* niepermutacyjny problem przypływowy z kryterium sumacyjnym [12],
* uogólniony problem przydziału [13].

# Formalizacja modelu matematycznego

## Podstawowe parametry symulacji

Utworzona na potrzeby projektu symulacja dzieje się w pewnej przestrzeni. Tak jak w rzeczywistości jest ona trójwymiarowa. Jej długość, szerokość oraz wysokość jest ograniczona. Zakłada się, że przestrzeń ta jest jednorodnie wypełniona powietrzem atmosferycznym o uśrednionych proporcjach poszczególnych jego składników. Ponadto ciśnienie atmosferyczne oraz gęstość powietrza uznaje się za jednakowe niezależnie od wysokości. Zakłada się, że temperatura nie wpływa negatywnie na pracę dronów i nie powoduje dodatkowych ruchów powietrza, a odpowiadająca jej gęstość powietrza wynosi 1,2.

Dodatkowym czynnikiem uwzględnionym w symulacji jest ruch powietrza czyli wiatr. Określana jest jego prędkość względem osi *x* oraz *y* (względem osi *z* zawsze wynosi 0), która w trakcie trwania symulacji jest stała. Co więcej zakłada się stały kierunek wiatru. Turbulencje związane są tylko z przelotem drona. Uwzględniono je we wzorze na opór aerodynamiczny. Nie wpływają jednak w żaden sposób na inne drony (kluczowa jest tutaj strefa bezpieczeństwa drona, o której więcej informacji w rozdziale 3.3).

Pozycję drona w przestrzeni symulacji określają współrzędne (*x*,*y*,*z*). Jest to pozycja jego środka ciężkości. Symulacja polega na określaniu kolejnych pozycji co podany okres czasu (interwał czasowy). Czas ten także jest wyznaczany przez operatora symulacji w zależności od oczekiwanej dokładności. Należy tutaj wspomnieć, iż w pewnym zakresie obliczeń rozwiązuje się równanie różniczkowe metodą Eulera. Zbieżność uzyskanego wyniku z wynikiem rzeczywistym jest tym lepsza im mniejszy jest zadany interwał czasowy.

Wyznacza się również siłę nośną oraz maksymalną prędkość drona. Siła to nic innego jak maksymalna siła jaką są w stanie wytworzyć wirniki „wypychając” powietrze. Maksymalna prędkość to ograniczenie zakresu prędkości jaką mogą rozwinąć drony.

Zależnie od rodzaju testów oraz potrzeb symulowany jest lot *n* dronów. Liczba ta oczywiście powinna być liczbą naturalną. Aby symulacja mogła być użyta przez algorytm wymagane są minimum 2 drony.

## Wyznaczanie kolejnych pozycji dronów

W zaimplementowanej symulacji drony pomiędzy zadanymi punktami poruszają się po trajektoriach prostoliniowych. Ma to na celu skrócenie do minimum odległości jakie pokonują drony oraz uproszczenie problemów związanych ze sterowaniem.

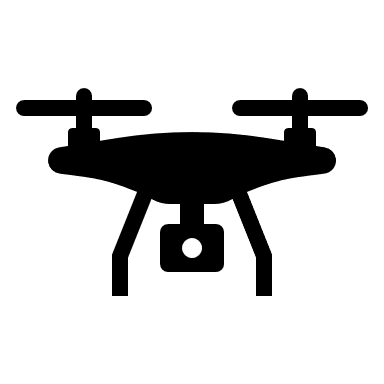
Podstawą sterowania dronami jest wyznaczenie interwałów czasowych, w których przyspieszają, hamują bądź lecą ze stałą prędkością oraz osiągniętej pozycji. Ruch drona po trajektorii można opisać jednym z trzech wariantów ruchu.

Pierwszym wariantem jest lot na dystansie pozwalającym rozwinąć prędkość maksymalną. Dron przyspiesza do tejże prędkości, a następnie leci zachowując jej wartość. Gdy znajdzie się w odległości od punktu docelowego równej wyznaczonej drodze hamowania, dron rozpoczyna hamowanie i zatrzymuje się w zadanej pozycji.

Drugim wariantem jest lot na dystansie zbyt krótkim aby rozwinąć maksymalną prędkość, ale na tyle długim aby przemieszczać się dłużej niż jeden interwał czasowy. Dron przelatuje wtedy całą odległość z przyspieszeniem bądź z opóźnieniem. Dla potrzeb sterowania obliczany jest przybliżony dystans przyspieszania i hamowania. Następnie jest wykonywany zgodnie z uzyskanymi wartościami. Ze względu na możliwe różnice pomiędzy faktycznym miejscem zatrzymania, a punktem docelowym (wynikające z zastosowanego przybliżenia) wprowadza się pewien margines błędu. Jego istnienie jest dopuszczalne, bowiem dron zawsze znajdzie się dostatecznie blisko, aby punkt docelowy został objęty co najmniej strefą bezpieczeństwa drona. Uznaje się więc, że w rzeczywistości dron delikatnie skorygowałby swoją pozycję do zadanej. Nie wpływa to znacznie na kryteria funkcji celu, dlatego pomija się ten fakt w symulacji i lot drona zostaje sklasyfikowany jako ukończony w punkcie docelowym.

Ostatnim wariantem jest najprostszy wariant: dron osiąga punkt docelowy w jednym interwale czasowym. Nie wymaga to żadnych dodatkowych obliczeń.

opór powietrza



siła wytworzona przez wirniki drona

siła odziaływania wiatru na drona

siła ciężkości

kierunek lotu

wiatr

*Rysunek 3.1 Siły działające na drona podczas lotu.*

Aby zająć się kwestią przyspieszania i hamowania należy najpierw pochylić się nad siłami wpływającymi na lot drona. Wynikiem dodania tych sił wektorowo jest wektor wypadkowy siły , której kierunek powinien być identyczny do kierunku zadanej trajektorii. Do celów obliczeniowych wygodniejsze jest operowanie na jej składowych: , i . Pierwszą ważną siłą wpływającą na wektor wypadkowy jest siła ciężkości drona, którą oblicza się za wzoru 3.1.

(3.1)

– siła ciężkości drona

– masa drona

– przyspieszenie ziemskie = 9.8

Jej zwrot jest zawsze przeciwny do zwrotu osi *z* i jest prostopadły do innych osi, więc będzie wpływała jedynie na składową . Inaczej jest z siłą oporu aerodynamicznego. Oblicza się ją ze wzoru 3.2.

(3.2)

– opór aerodynamiczny

-współczynnik oporu aerodynamicznego drona

– gęstość powietrza =1,2

– pole przekroju poprzecznego drona

-prędkość drona

Na siłę oporu składa się kilka zmiennych. Współczynnik oporu aerodynamicznego oraz pole przekroju drona zależą od kierunku przemieszczania się powietrza względem drona. Dla uproszczenia uznaje się wartości te za takie same (różnice nie są spektakularnie duże).

Największy problem stanowi jednak prędkość, gdyż nie jest to prędkość drona ale powietrza. Chociaż w bezwietrznej sytuacji wektor prędkości powietrza różniłby się od wektora prędkości drona jedynie przeciwnym zwrotem, to przy wietrze należy dodać jeszcze jeden wektor. Dopiero wartość tego wektora może zostać wykorzystana we wzorze 3.2. Podstawiając wszystkie zmienne do wzoru w wyniku otrzymujemy wartość wektora siły oporu, który ma kierunek i zwrot zgodny z wektorem prędkości powietrza. Do celów obliczeniowych wektor ten zostaje rozłożony na wektory *Dx, Dy* i *Dz*.

Teraz należy przyjrzeć się samej sile . Kierunek wektora siły wypadkowej musi być zgodny z wyznaczoną trajektorią prostoliniową. Tylko wtedy trajektoria drona będzie tożsama z zadaną trajektorią. W związku z tym należy odpowiednio skierować siłę drona . Znając współrzędne punktu początkowego (x1,y1,z1) oraz punktu docelowego (x2,y2,z2) można, opierając się na właściwościach rozkładu wektora na składowe, wyznaczyć równanie:

(3.3)

Mając obliczone wszystkie znane siły można zapisać następujący układ równań:

(3.4)

Gdzie *,* ito składowe wektora . Po przekształceniu układu równań 3.4 otrzymuje się:

(3.5)

Korzystając z twierdzenia pitagorasa można zapisać siłę drona F jako:

(3.6)

Pamiętając, że wszystkie wektory zostały pogrupowane względem równoległości do danej osi oraz uwzględnia się ich zwrot względem niej można zapisać:

(3.7)

Za pomocą wiedzy o proporcjonalności składowych siły wypadkowej można podstawić jedną ze składowych sił w miejsce pozostałych jeśli nie jest równa 0. Przykładowo dla składowej siły wzór prezentowałby się następująco:

(3.8)

Otrzymane równanie ma jedynie jedną niewiadomą (). Po jego uporządkowaniu otrzymuje się równanie kwadratowe, które da się rozwiązać za pomocą tzw. delty. Otrzymaną wartość wstawia się do równania 3.3 i w rezultacie otrzymuje się brakujące składowe siły wypadkowej. Za pomocą twierdzenia Pitagorasa (podobnego do równania 3.6) otrzymuje się finalną wartość siły wypadkowej jaka powoduje przyspieszenie drona.

Do kolejnych obliczeń potrzebne jest przyspieszenie jakie nadaje siła . Z przekształcenia drugiej zasady dynamiki Newtona wynika:

(3.9)

– wektor przyspieszenia drona

– wypadkowa siła

– masa drona

Obliczone przyspieszenie w rzeczywistości zmieniałoby się w czasie, bowiem każda zmiana prędkości (czyli efekt przyspieszenia) powoduje zmianę oporu aerodynamicznego, co zmienia wartość siły wypadkowej powodującej przyspieszenie. Mamy więc do czynienia ze swoistym równaniem różniczkowym. Na potrzeby projektu oblicza się przyspieszenie co pewien interwał czasowy, tj. korzysta się z metody Eulera na obliczanie tego typu równań.

Obliczanie przyspieszenia jest kluczowe jedynie w przypadku hamownia oraz przyspieszania. Dron wytwarza wtedy maksymalną zadaną siłę ciągu . Inaczej jest podczas lotu z prędkością maksymalną. Wtedy siła ciągu drona jest redukowana do stopnia pozwalającego na zrównoważenie występujących oporów. Jej wartość jest co najwyżej równa , ponieważ wszystkie siły oporu przy stałej prędkości będą stałe. Dlatego w tym przypadku nie jest konieczne wykonywanie obliczeń związanych z siłami.

Do obliczania prędkości i drogi jaką pokonuje dron w zadanym przedziale czasowym stosuje się następujące wzory:

(3.10)

(3.11)

– prędkość chwilowa drona

– prędkość drona w poprzedniej próbce

– przyspieszenie drona

– zadany interwał czasowy [s]

droga pokonana przez drona w zadanym interwale czasowym [m]

Tutaj sprawa staje się banalnie prosta bowiem za pomocą twierdzenia Talesa oraz twierdzenia Pitagorasa można obliczyć dokładną pozycję drona w kolejnym kroku symulacji jak widać na rysunku 3.2.

*x1*

*x2*

*S*

*Sx*

*Rysunek 3.2 Graficzne przełożenie odległości w przestrzeni na odległość w osi x*

Rezultatem przeprowadzonej symulacji jest otrzymanie macierzy wypełnionej pozycjami dronów. Każdy kolejny rząd oznacza pozycję drona po upływie zadanego interwału czasowego *t*. Jeśli np. rozpatrywana jest pozycja w rzędzie 5 oraz *t*=1s, to jest to pozycja, którą zajmuje dron po 5 sekundach od rozpoczęcia symulacji (procedury zmiany pozycji). Odległości pomiędzy kolejnymi pozycjami mogą się różnić z powodu przyspieszania i hamowania. Kolumny macierzy odpowiadają poszczególnym dronom. Wszystkie pozycje pojedynczego drona leżą na prostej łączącej punkty początkowy i docelowy. Taka forma rozwiązania pozwala na obliczenie kryteriów potrzebnych do funkcji celu:

* czas symulacji to liczba rzędów pomnożona przez zadany interwał czasowy (3.12).
* możliwość wystąpienia kolizji jest sprawdzana poprzez porównanie odległości pomiędzy dwiema pozycjami z zadanymi wymiarami strefy bezpieczeństwa (rozdział 3.3) (3.13-15),
* średnie oddalenie podczas symulacji oblicza się jako iloraz sumy odległości pomiędzy wszystkimi kombinacjami dwóch pozycji we wszystkich rzędach i liczby wszystkich kombinacji

rows (3.12)

(pozycje dronów i oraz j w czasie t) ,

(zgodnie z 3.14) (3.17)

(3.18)

– czas potrzebny na zajęcie punktów docelowych przez wszystkie drony

– zadany interwał czasowy

rows – liczba rzędów jakie zajmują pozycje drona od rozpoczęcia lotu do osiągnięcia punktu docelowego

– liczba kolizji

– liczba rzędów macierzy wynikowej

– liczba dronów

– numery porządkowe dronów

, , – wymiary strefy bezpieczeństwa drona według poszczególnych osi [m]

– średnie oddalenie dronów od siebie podczas całej symulacji

– liczba wszystkich kombinacji dronów podczas całej symulacji

## Kolizje

Każdy dron posiada własną strefę bezpieczeństwa wyznaczoną względem wszystkich osi (np. . Pojawienie się innego drona w tej strefie uznaje się zawsze jako kolizję, niezależnie od tego czy fizyczna kolizja rzeczywiście mogła nastąpić. Należy bowiem brać pod uwagę także inne niebezpieczeństwa wynikające z bliskości dronów opisane w rozdziale 2.2.

Funkcja sprawdzająca czy dochodzi do kolizji otrzymuje pełną listę zawierającą kolejne pozycje dronów w każdej próbce symulacji. Następnie wszystkie pozycje w danym kroku sprawdza się pod kątem odległości we wszystkich trzech wymiarach. Celem uniknięcia powtarzania porównań między tymi samymi pozycjami, wpierw sprawdza się ewentualne kolizje pomiędzy pierwszym dronem i wszystkimi pozostałymi, później zaś pomiędzy drugim, a tymi których numer porządkowy jest wyższy itd.. Jeśli odległości we wszystkich wymiarach będą mniejsze od wyznaczonych dla strefy, to funkcja oznacza takie zbliżenie jako kolizję.

## Funkcja celu

Bardzo ważnym pod względem optymalizacji symulacji jest określająca ją funkcja celu. Jej wartość uzyskuje się z następującego równania:

– rozpatrywane rozwiązanie

– czas potrzebny na zajęcie punktów docelowych przez wszystkie drony

*k* – liczba kolizji pomiędzy dronami

*d* – średnie oddalenie dronów podczas symulacji [m]

*a,b,c –* współczynniki wagi poszczególnych zmiennych

Każde z wymienionych kryteriów ma swój własny współczynnik wagi. Jest on zadawany odgórnie w zależności od potrzeb symulacji i testów. Najważniejszym z punktu widzenia realizmu jest kryterium kolizji, które powinno praktycznie uniemożliwiać akceptację symulacji. Dlatego mnożnik tego kryterium musi być w porównaniu z pozostałymi dużo większy.

Kryterium czasu trwania zmiany pozycji wszystkich dronów jest szczególnie ważne w kontekście zużycia energii oraz estetyki pokazu. W symulacji zakłada się, że energia akumulatorów dronów zawsze pozwala na wybranie dowolnej trajektorii (zużycie nie jest w żaden sposób modelowane). Lecz w przypadku użycia wyników symulacji do rzeczywistego wykorzystania należy uwzględnić ograniczony czas lotu bez ładowania akumulatora. Wtedy kluczowa jest minimalizacja czasu spektaklu. Dlatego współczynnik wagi kryterium czasu należy dostosować w zależności od potrzeb.

Można bowiem wyżej cenić kryterium oddalenia dronów od siebie. Jest to szczególnie pożądane jeśli chodzi o estetykę, gdyż donioślej wygląda lot dronów daleko od siebie niż stłoczenie na bardzo małej przestrzeni. Współczynnik wagi tego kryterium musi więc być odpowiednio dostosowany względem mnożnika kryterium czasu trwania.

Funkcja celu jest minimalizowana. Zadanie znalezienia takiego przydziału dronów do punktów docelowych, aby wynikająca z cech symulacji funkcja celu była jak najniższa, należy do zastosowanego algorytmu.

# Algorytm przeszukiwania z zabronieniami

## Schemat algorytmu

Sterowanie dronami skutkujące osiągnięciem poszczególnych punktów końcowych w celu optymalizacji funkcji symulacji nie jest banalny. Dokonanie zmiany celu lotu pomiędzy zaledwie dwoma dronami może całkowicie zmienić wyniki symulacji, ze względu na możliwość zajścia kolizji. W dodatku przy wzroście liczby dronów zwiększa się również liczba możliwych kombinacji, a także nakład obliczeniowy przeanalizowania każdej z nich. Dlatego konieczne jest zastosowanie algorytmu heurystycznego w celu ograniczenia czasu kosztem jakości otrzymanego rozwiązania. Na potrzeby niniejszego projektu wybrano algorytm przeszukiwania z zabronieniami – Tabu Search (*TS*).

Algorytm Tabu Search ma swoje korzenie na przełomie lat 60-tych i 70-tych. Znaną dzisiaj formę algorytmu zawdzięczamy profesorowi Fredowi Gloverowi, który sformułował ją w latach 80-tych [14]. Idea działania opiera się na przeszukiwaniu określonego sąsiedztwa w poszukiwaniu najlepszego rozwiązania. W rozpatrywanym przypadku będzie to wymiana punktów docelowych pomiędzy dwoma dronami. Następnie algorytm wykonuje kolejne „ruchy” czyli zmiany sąsiedztwa na najlepsze, które mogło zostać osiągnięte z poprzedniego sąsiedztwa. Najlepszy wynik osiągnięty podczas działania algorytmu oraz odpowiadający mu przydział są finalnym rozwiązaniem problemu.

**Algorytm Tabu Search** (na podstawie pseudokodu z artykułu [13])

**Krok 1**:

Wyznacz rozwiązanie początkowe *xpocz* i podstaw: *x* := *xpocz*, *xts* := *xpocz*, *Qts* := Q(*xpocz*), *CA:*=*False*

**Krok 2**: dla *k*=1 do *K* wykonaj: (*K*- warunek stopu):

*x*(*i\*, j\**) = arg min;

*x*(*i’,j’*) = arg min ;

podstaw: *x* := *x*(*i\*, j\**);

Jeżeli < *Qts*  to *xts* := *x, Qts* := Q(*x*);

Jeżeli < *W\*Qts*  to *xts* := *x*(*i’,j’*)*, Qts* := Q(*x*(*i’,j’*)), *CA:*=*True* ;

Dla każdego (*i*,*j*) podstaw: ;

Jeżeli *CA*=*False*, to *KLT*(*i\**,*j\**) = *T*, inaczej *KLT*(*i’*,*j’*) = *T;*

Jeżeli *CA*=*False*, to *DLT(i\**,*j\**):= *DLT(i\**,*j\**)+1, inaczej *DLT(i’*,*j’*):= *DLT(i’*,*j’*)+1, *CA:*=*False.*

Zwróć (*xts*, *Qts*)

Opis oznaczeń w pseudokodzie:

*CA* – zmienna określająca czy zastosowano kryterium aspiracji,

– wartość współczynnika kary,

*KLT* – lista tabu zawierająca liczbę tur, w których określony krok jest zabroniony,

*DLT* – lista tabu zawierająca liczbę poszczególnych dokonanych kroków algorytmu,

*T* – liczba iteracji, w których dany krok znajduje się na liście *KLT*.

*W* – współczynnik służący do kontrolowania kryterium aspiracji

## Pamięć krótko- i długoterminowa

W zależności od potrzeb możliwa jest implementacja algorytmu przeszukiwania z zabronieniami na kilka sposobów. W tym przypadku użyte zostaną algorytmy z pamięcią krótkoterminową oraz algorytm z pamięcią zarówno krótko- jak i długoterminową.

Pamięć krótkoterminowa (*KLT*) to lista tabu, która posiada po jednej pozycji dla każdego możliwego ruchu algorytmu. Domyślnie zapełniona jest zerami. Po każdym kroku algorytmu, najpierw od wszystkich niezerowych wartości odejmuje się jeden, a następnie na pozycji odpowiadającej wykonanemu ruchowi pojawia się wartość *T,* czyli określony w liczbie iteracji okres znajdowania się ruchu na liście tabu. Algorytm sprawdza dla każdego ruchu czy nie znajduje się na *KLT* (czyli czy *KLT*(*i*,*j*)>0). Jeśli *KLT*(*i*,*j*)=0, to gdy ruch okaże się najlepszy, może być wykonany. Jeśli jednak *KLT*(*i*,*j*)>0, to ruch może zostać wykonany tylko jeżeli spełni warunek aspiracji.

Warunek aspiracji to swoisty wytrych, który umożliwia algorytmowi wykonanie niedozwolonego ruchu. Jeśli jego wynik okaże się lepszy od progowej wartości funkcji celu (*W\*Qts* ), to kryterium aspiracji zostaje spełnione.

Główną różnicą pomiędzy zastosowanymi dwoma typami algorytmu jest zastosowanie bądź pominięcie pamięci długoterminowej (*DLT*). Podobnie jak lista *KLT* zawiera po jednej pozycji dla każdego ruchu. Jednakże, jej przeznaczeniem jest zliczanie liczby wykonanych poszczególnych ruchów. Informacja ta jest wykorzystywana przy modyfikacji funkcji celu dla rozpatrywanego przydziału, bowiem do podstawowego wyniku dodaje się iloczyn wartości na odpowiedniej pozycji *DLT* i stałej wartości współczynnika kary, podzielonego przez ilość wykonanych iteracji algorytmu. Ma to na celu zapobieganiu wykonywaniu tych samych ruchów zbyt wiele razy. [13]

## 

# Opis aplikacji

## Środowisko programistyczne

Do implementacji opisanych w poprzednich rozdziałach modelu matematycznego oraz algorytmu przeszukiwania z zabronieniami wykorzystano program Spyder. Oprogramowanie to zostało wydane na otwartej licencji MIT, dzięki czemu jest ono darmowe i powszechnie dostępne [15]. Program ten został wybrany dzięki dużej łatwości sprawdzania wartości poszczególnych zmiennych bez potrzeby zamieszczania polecenia ich wypisania w kodzie. Pozwala to na szybsze znajdywanie błędów w implementacji oraz na wygodne analizowanie uzyskanych wyników.

Obraz zawierający tekst

Opis wygenerowany automatycznie

*Rysunek 5.1 Zrzut ekranu programu Spyder*

Implementacja została wykonana za pomocą języka Python, który jest obecnie bardzo często używany. Jest to niezwykle przyjazny dla użytkownika język, który automatycznie przypisuje typy zmiennym oraz pozwala na zapisanie w jednej linijce instrukcji, które w innych językach potrzebują dużo więcej miejsca.

## Proces powstawania i testowania kodu

Całość kodu została zawarta w trzech plikach Pythona: *Zmienne*, *Symulacja* oraz *Algorytm*. W pliku *Zmienne* zostały umieszczone wszystkie zadawane zmienne wpływające na wygląd symulacji. Są to m. in. wymiary przestrzeni symulacji, liczba dronów, wymiary strefy bezpieczeństwa drona, prędkość wiatru w poszczególnych osiach oraz maksymalna siła wytwarzana przez drona. Tutaj została również zdefiniowana klasa Drone, która przechowuje wszystkie informacje dotyczące pojedynczego drona (np. prędkość, punkt docelowy, akyualna pozycja itp.). Ponadto w liku tym znajdują się również funkcje wykorzystywane w klasie Drone (np. sprawdzanie kierunku lotu), oraz funkcje przygotowujące losowe punkty startowe i punkty docelowe.

Plik o nazwie *Symulacja* przechowuje funkcje obliczeniowe. Ich zadaniem jest wykrycie i realizacja odpowiedniego wariantu sterowania, obliczanie kolejnych pozycji dronów zgodnie z zadanym modelem matematycznym, oraz obliczanie kryteriów potrzebnych do funkcji celu.

Ostatni plik (*Algorytm*) jest najważniejszy, ze względu na swoją zawartość. To w nim znajduje się oraz jest wywoływana funkcja realizująca algorytm. Do jej działania wykorzystywana jest zawartość pozostałych plików.

Podczas programowania dokonywano po skończeniu kolejnego etapu prac testów zachowania programu. Wpierw sprawdzone zostało działanie funkcji losowo dobierającej punkty startowe oraz docelowe. Dopiero po jej pozytywnej weryfikacji możliwe było kodowanie modelu matematycznego.

Kodowanie modelu matematycznego było podzielone na dwa etapy ze względu na jego dużą złożoność. W pierwszym etapie stworzono funkcje obliczające kolejne pozycje, przy czym przyspieszenie zostało podane jako stała wartość. Dzięki takiemu rozwiązaniu weryfikacja prawidłowości lotu dronów była dużo prostsza. Tutaj pojawił się problem dotyczący niedokładności sterowania dronem (dokładnie chodziło o zbyt szybkie hamowanie). Został on rozwiązany poprzez wprowadzenie maksymalnego błędu odległości, zależnego od maksymalnej siły drona. Jeśli dron znajduje się wystarczająco blisko, to za jego kolejną pozycję uznaje się pozycję docelową. Jest to dopuszczalne, gdyż dron wykorzystuje do hamowania maksymalną siłę ciągu, więc w rzeczywistości jej częściowe zmniejszenie pozwoliłoby dronowi na zatrzymanie się dokładnie w wyznaczonym miejscu.

W drugim etapie zbudowane zostały funkcje obliczające przyspieszenie zgodnie ze wzorami z rozdziału 3.2. Ze względu na różniczkowy charakter równań przyspieszenie jest obliczane jedynie co krok symulacji (czyli co określony interwał czasowy w symulacji). Podczas testów zauważony został problem zbyt małej siły ciągu drona. W skrajnie trudnych przypadkach siła drona nie wystarczała do pokonania siły ciężkości i oporów ruchu dla danej prędkości maksymalnej. To zaś powodowało pojawienie się zespolonych rozwiązań równania kwadratowego. W celu wykrywania tego problemu stworzony został specjalny wyjątek zatrzymujący całą symulację i wskazujący na ten problem.

Ostatnim brakującym ogniwem była implementacja algorytmu. Została ona wykonana najpierw dla algorytmu TS z pamięcią krótkotrwałą. Później zaś na jej podstawie, po dodaniu kilku niezbędnych elementów, powstał algorytm TS z pamięcią krótko- i długotrwałą.

Podczas pisania całego kodu pojawiało się kilka mniejszych problemów, które zostały rozwiązane. Często wynikały one z błędów w składni kodu bądź z błędów literowych. Pojawiały się również błędy niszczące wyniki symulacji, poprzez błędne obliczenia w jednej z funkcji. Mimo to udało się uzyskać prawidłowo działający program realizujący model matematyczny oraz algorytm.

# Scenariusze testowe

## Optymalna liczba dronów

## Prędkość maksymalna dronów

## Wyrównywanie czasu lotu

## Maksymalizacja oddalenia dronów

# Podsumowanie

# Bibliografia

[1] P. Bukowski, G. Szala: *Bezzałogowe statki powietrzne – geneza, teraźniejszość i przyszłość* Postępy w Inżynierii Mechanicznej Tom 11(6): 5-11, 2018r.

[2] A. Miah *Drones : The Brilliant, the Bad and the Beautiful.* Wydawnictwo Emerald publishing, 2020r.

[3] [online] <https://commons.wikimedia.org/wiki/File:Oemichen2.jpg>

[4] MRT: *How many thousands of drones have they used in the balloon of the Tokyo 2020 ceremony?* [online] <https://marketresearchtelecast.com/how-many-thousands-of-drones-have-they-used-in-the-balloon-of-the-tokyo-2020-ceremony/111062/>

[5] M. Feltynowski, M. Zawistowski: *Możliwości wykorzystania bezzałogowych platform w służbach ratunkowo-porządkowych.* Bezpieczeństwo i technika pożarnicza Tom 51: 138-149, 2018r.

[6] N. Schwartz: *Watch 1,218 drones form the Olympics rings during Opening Ceremony* [online] <https://ftw.usatoday.com/2018/02/2018-winter-olympics-opening-ceremony-1218-drones-rings>

[7] R. Kowalik, T. Łusiak, A. Novak: *A mathematical model for controlling a quadrotor UAV.* Transactions on Aerospace Research Rocznik 2021 Tom 3 (264): 58-70, 2021r.

[8] D. Tobór, J. Barcik, P. Czech: *Legal aspects of air transport safety and the use of drones*. Scientific Journal of Silesian University of Technology. Series Transport Tom 97: 167-179, 2017r.

[9] S. Dasgupta, C. Papadimitriou, U. Vazirani: *Algorytmy* Wydawnictwo Naukowe PWN 2012r.

[10] *Algorytmy heurystyczne*, [Online]: <http://algorytmy.ency.pl/artykul/algorytmy_heurystyczne>

[11] J. Pempera: *Równoległy algorytm tabu z elementami inspirowanymi naturą dla problemu planowania tras.* Automatyka Tom 15 Zeszyt 2: 399-408, 2011r.

[12] J. Grabowski, J. Pempera: *Hybrydowy algorytm tabu dla niepermutacyjnego problemu przepływowego z kryterium sumacyjnym.* Automatyka Tom 13 Zeszyt 2: 289-296, 2009r

[13] K. Wala: *Algorytm Tabu w optymalizacji uogólnionego problemu przydziału.* Automatyka Tom 11 Zeszyt 1-2: 309-316, 2007r.

[14] F.Glover, E. Taillard, D. de Werra: *A user's guide to tabu search.* Annals of Operations Research 41: 3-28, 1993r.

[15] A. Damien: *Spyder, potężne interaktywne środowisko programistyczne dla języka Python* [online] <https://ubunlog.com/pl/spyder-entorno-desarrollo-python/>