

WOJSKOWA AKADEMIA TECHNICZNA

im. Jarosława Dąbrowskiego

WYDZIAŁ ELEKTRONIKI



PRACA DYPLOMOWA

ELEKTRONICZNY EDYTOR PLANU LOTU

.....
(temat pracy dyplomowej)

sierż. pchor. inż. Bartłomiej ZĄBEK

.....
(stopień wojskowy, tytuł zawodowy, imiona i nazwisko, imię ojca dyplomanta)

ELEKTRONIKA I TELEKOMUNIKACJA

(kierunek studiów)

RADIONAWIGACJA

.....
(specjalność)

STACJONARNE STUDIA DRUGIEGO STOPNIA – MAGISTERSKIE

.....
*(forma i rodzaj studiów)**

dr inż. Andrzej WITCZAK

.....
(stopień wojskowy, tytuł i stopień naukowy, imię i nazwisko promotora pracy dyplomowej)

Zadanie d

Zadanie dyp

Spis treści

1	Wstęp	8
2	Analiza problematyki planowania lotu oraz planowania misji	11
2.1	Procedura planowania lotu cywilnego statku powietrznego	11
2.2	Przegląd dostępnych rozwiązań	20
2.3	Planowanie misji	22
3	Założenia.....	26
3.1	Założenia do aplikacji przeznaczonej do planowania lotu.....	26
3.2	Założenia do aplikacji wspomagającej planowanie misji	28
4	Instrukcja obsługi aplikacji	29
4.1	Aplikacja PLAN LOTU	29
4.2	Aplikacja wspomagającej planowanie misji	35
5	Algorytmy aplikacji planowania lotu i planowania misji	39
5.1	Algorytmizacja aplikacji do planowania lotu	39
5.1.1	Okno wyboru wyposażenia.....	40
5.1.2	Narzędzie do planowania trasy	42
5.1.3	Okno Formularza	49
5.2	Algorytmizacja aplikacji do planowania misji.....	51
6	Podsumowanie i wnioski	57

Wykaz akronimów

AFTN - Aeronautical Fixed Telecommunication Network, Sieć Stałej Telekomunikacji Lotniczej

AIS – Air Information Service, Służba informacji Lotniczej

ATCU – Air Traffic Control Unit, Jednostka Służby Kontroli Ruchu Lotniczego

AO – Aircraft Operator, Operator statku powietrznego

ARO - Airport Reservation Office, Biuro Odpraw Załóg

ATS – Air Traffic Management, Zarządzenie ruchem lotniczym

DTED – Digital Terrain Elevation Data, Cyfrowe dane o elewacji terenu

EOBT - Estimated off-Block Time, Czas odblokowania

ETE – Estimated Time En Route, Przewidywany czas przelotu

ETFMS - Enhanced Tactical Flow Management System, System zarządzania przepływem ruchu lotniczego

ERC – En Route Chart, Mapa trasowa

FIR – Flight Information Region, Rejon Informacji Powietrznej

FL – Flight Level, Poziom Lotu

FPL - Filed Flight Plan, Złożony Plan Lotu

GAT - General Air Traffic, Ogólny ruch lotniczy

GMTED - Global Multi-Resolution Terrain Elevation Data, Zestaw globalnych, wieloskalowych zmiennych topograficznych

ICAO – International Civil Aviation Organization, Organizacja Międzynarodowego Lotnictwa Cywilnego

ILS – Instrumental Landing System, System lądowania wg przyrządów

IFPS - Integrated Initial Flight Plan Processing System, System wstępnego przetwarzania planów lotu

IFPZ – IFPS Zone, Strefa operacyjna systemu IFPS

NOTAM – Notice To AirMen, Depesza informacyjna dla pilotów

ORM - Operational Risk Management, Operacyjne zarządzanie ryzykiem

RPL - Repetitive Flight Plan, Powtarzalny plan lotu

SP – Statek Powietrzny

TAC - Terminal Area Chart, Plan tarminala

IFR – Instrumental Flight Rules, Lot wykonywany zgodnie z przepisami dla lotów według wskazań przyrządów.

WAC - World Aeronautical Chart, Mapa Lotnicza Świata

VFR – Visual Flight Rules Zasady lotu z widocznością

VNC - VFR Navigation Charts, Mapa lotnicza do lotów VFR

VOR - VHF Omnidirectional Radio Range, Radiolatarania ogólnokierunkowa VHF

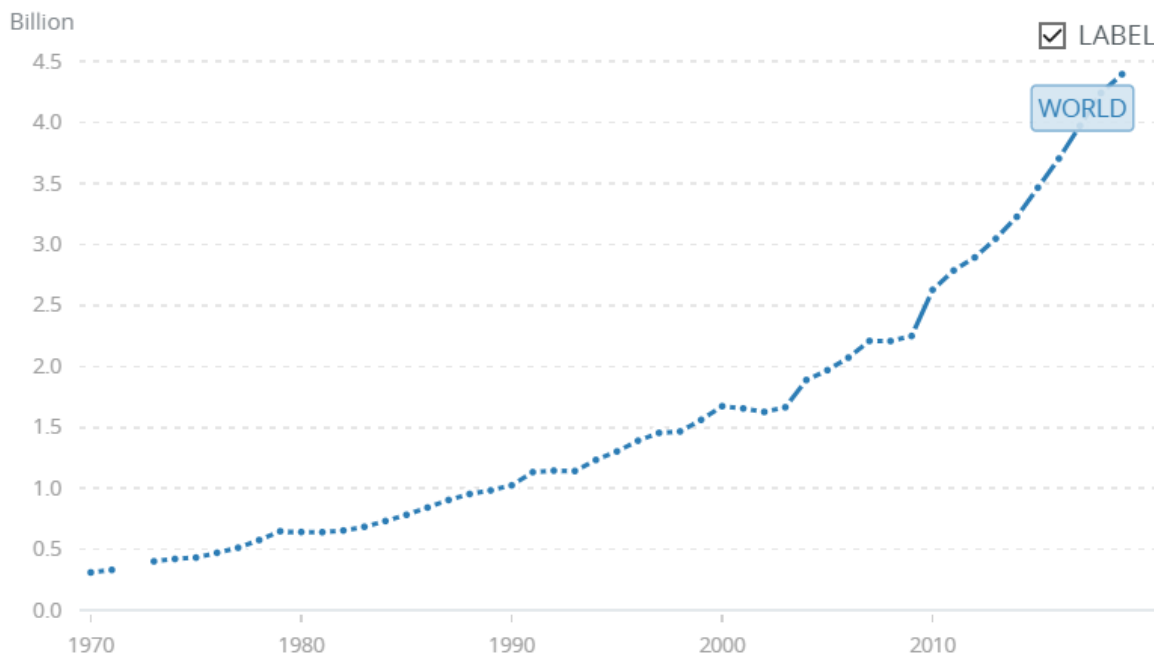
VTC - Visual Terminal Charts, Mapa terminala lotniczego dla lotów VFR

SARTIME - 'time search action required', Czas do podjęcia akcji ratowniczej

1 Wstęp

Lotnictwo, zarówno cywilne jak i wojskowe stały się w XX wieku istotnym elementem rozwoju cywilizacji. Jego kluczowy udział w transporcie osób i towarów powoduje istotne zagęszczenie ruchu lotniczego w wielu miejscach świata. Zapewnienie odpowiedniego bezpieczeństwa żeglugi powietrznej wymaga stosowania coraz bardziej rozwiniętych narzędzi technicznych wspierających nadzór nad ruchem i jego bezkolizyjnym przebiegiem. Urządzenia te instaluje się zarówno na pokładach statków powietrznych jak i na ziemi. Informacje dostarczane przez te systemy są podstawą działania służb Kontroli Ruchu Lotniczego (ang. ATC – Air Traffic Control). Elementem wyjściowym dla poprawnego działania całego systemu transportu lotniczego jest właściwe jego planowanie. Uczestnik ruchu lotniczego, aby móc wykonać lot, powinien zgłosić taki zamiar odpowiednim służbą, a zgłoszenie to ma postać znormalizowanego dokumentu jakim jest tzw. Plan Lotu. Wzrost ilości statków powietrznych uczestniczących w ruchu powoduje wzrost ilości informacji jaki jest przetwarzany w systemach planowania oraz nadzoru. Zgodnie z danymi banku światowego przedstawionymi na rysunku 1 [12] widać bardzo szybki wzrost liczby pasażerów na przestrzeni lat. Wynika to z postępującej globalizacji oraz rozwoju nowoczesnej techniki. Nawet wydarzenia takie jak atak z 11 września oraz kryzys finansowy z lat 2007 – 2009 nieznacznie i tylko chwilowo spowolniły rozwój omawianej branży. Z pewnością najgłębsze od lat zaburzenie dynamiki wzrostu spowodowane światową pandemią wywołaną wirusem Covid 19 nie zmieni wieloletniej tendencji wzrostowej.

Sytuacja taka powoduje, iż koniecznym staje się rozwój wszelkiego rodzaju rozwiązań technicznych usprawniających przepływ informacji w systemach nadzoru ruchu lotniczego. Celem niniejszej pracy była analiza systemu planowania misji w lotnictwie cywilnym i stworzenie aplikacji ułatwiającej przygotowanie planu lotu, a w efekcie również uzyskanie jego elektronicznej wersji, możliwej do przetwarzania w systemach planowania ruchu lotniczego.



Rysunek 1 Ilość pasażerów na przestrzeni lat [12]

Drugie zagadnienie jakie poddano analizie to wspomaganie planowania misji bojowych wojskowych statków powietrznych. Problem jest zdecydowanie bardziej szeroki i obejmuje różne kwestie, od zadbania o odpowiednie zabezpieczenie logistyczne, zaplanowanie przelotu nad rejon działań aż do operowania nad terytorium przeciwnika. Współczesne militarne systemy planowania misji są złożonymi systemami informatycznymi budowanymi przez wyspecjalizowane instytucje. Uwzględniając ograniczone możliwości realizacyjne, w niniejszej pracy ograniczono się do jednego z modułów takich systemów jakim jest moduł wspomagający minimalizację prawdopodobieństwa wykrycia naszego statku powietrznego przez radar przeciwnika w tym w szczególności powiązany z systemami obrony przeciwlotniczej. Opracowana aplikacja wyznacza strefy terenu w których samolot wykorzystując ukształtowanie terenu będzie mógł przelecieć unikając wykrycia przez radar naziemny. Jest to szczególnie przydatne w sytuacji prowadzenie operacji bojowych w terenie o urozmaiconym profilu elewacyjnym. Działania wojenne również toczą się w trudnym górzystym terenie jak np. wojny w Armenii czy w Afganistanie, co stanowiło dodatkowy impuls do rozpatrzenia wspomnianego problemu.

W drugim rozdziale przedstawiono wyniki analizy procedur związanych z planowaniem cywilnego lotu oraz dokonano przeglądu istniejących już rozwiązań

technicznych. Rozpatrzono również główne elementy jakie należy rozważyć planując misję wojskowego statku powietrznego.

W trzecim rozdziale przedstawiono założenia dotyczących funkcjonalności jakie mają spełniać obydwie aplikacje. Opracowano je dzięki wcześniejszej analizie problemu przedstawionej w pierwszym rozdziale oraz uwzględniając ograniczenia techniczne oraz czasowe związane z dużym nakładem pracy poświęconemu na szukanie rozwiązań technicznych i samemu pisaniu oprogramowania.

Czwarty rozdział zawiera instrukcję obsługi aplikacji oraz szczegółowo opisuje ich funkcjonalności. Piąty rozdział poświęcono szczegółowemu wyjaśnieniu działania oprogramowania.

Praca została zakończona podsumowaniem oraz wnioskami wyciągnięte podczas pracy nad projektem.

2 Analiza problematyki planowania lotu oraz planowania misji

W niniejszym rozdziale opisana została analiza zagadnień jakimi są planowanie lotu cywilnego statku powietrznego oraz planowania misji bojowej. Ponieważ tematy planowania misji i planowania lotu oparte są na zróżnicowanych wymaganiach oraz przepisach stąd zagadnienia te zostały omówione osobno. Pierwsza część skupia się na problemie planowania lotu cywilnego statku powietrznego, drugą poświęcono tematyce planowania misji.

2.1 Procedura planowania lotu cywilnego statku powietrznego

W pierwszej kolejności należy się przyjrzeć i dobrze zrozumieć procedury związane z planowaniem lotu. Niniejszy podrozdział przedstawi jakie czynności musi zrealizować, do czego jest zobowiązana, jakie informacje musi otrzymać przed rozpoczęciem pracy i z jakimi problemami się spotyka osoba planująca lot. Osobą taką może być pilot lub wyspecjalizowana komórka przewoźnika lotniczego.

Do zaplanowania lotu niezbędne jest wiele informacji. Konieczne jest sprawdzenie aktualnych warunków pogodowych w miejscu startu i miejscu lądowania a także na całej trasie przelotu, dostęp do depesz NOTAM, wgląd w mapę lotniczą, znajomość dostępnych tras lotniczych, punktów radionawigacyjnych, szczegółowe informacje dotyczące lotnisk, informacje o przeszkodach terenowych i wiele innych. Wypełniając formularz planu lotu należy dysponować wiedzą o wyposażeniu statku powietrznego, czy też ilości zabieranych pasażerów. Konieczne jest obliczenie zapotrzebowania na paliwo oraz czas przelotu. Formularz planu lotu przedstawiono na rysunku 2, jak widać liczba pól do wypełnienia jest duża. Ze względu na złożoność procesu powstało wiele elektronicznych rozwiązań ułatwiających to zadanie. Szczegółowa instrukcja w języku polskim jak wypełniać poszczególne pola formularza znajduje się w dokumencie - „WYTYCZNE NR 5 PREZESA URZĘDU LOTNICTWA CYWILNEGO z dnia 13 kwietnia 2017 r. w sprawie ogłoszenia wymagań ustanowionych przez Organizację Międzynarodowego Lotnictwa Cywilnego (ICAO) – Doc 4444” [2] oraz w języku angielskim w dokumencie o nazwie: „Appendix A. ICAO FLIGHT PLANS”. Poniżej opisano procedury wypełniania każdego z pól formularza planu lotu, bardziej szczegółowo jest to opisane w wymienionych wyżej dokumentach.

U S Department of Transportation Federal Aviation Administration		International Flight Plan	
PRIORITY <=FF	ADDRESSEE(S) <div style="border: 1px solid black; height: 15px; width: 100%;"></div>		
FILING TIME <div style="border: 1px solid black; width: 100px; height: 15px;"></div> ORIGINATOR <div style="border: 1px solid black; width: 100px; height: 15px;"></div> <=			
SPECIFIC IDENTIFICATION OF ADDRESSEE(S) AND / OR ORIGINATOR			
3 MESSAGE TYPE <=(FPL		7 AIRCRAFT IDENTIFICATION <div style="border: 1px solid black; width: 100px; height: 15px;"></div> <=	
9 NUMBER <div style="border: 1px solid black; width: 100px; height: 15px;"></div>		8 FLIGHT RULES <div style="border: 1px solid black; width: 100px; height: 15px;"></div> TYPE OF FLIGHT <div style="border: 1px solid black; width: 100px; height: 15px;"></div> <=	
10 EQUIPMENT <div style="border: 1px solid black; width: 100px; height: 15px;"></div> <=		13 DEPARTURE AERODROME <div style="border: 1px solid black; width: 100px; height: 15px;"></div> TIME <div style="border: 1px solid black; width: 100px; height: 15px;"></div> <=	
15 CRUISING SPEED <div style="border: 1px solid black; width: 100px; height: 15px;"></div> LEVEL <div style="border: 1px solid black; width: 100px; height: 15px;"></div> ROUTE <div style="border: 1px solid black; width: 100px; height: 15px;"></div>		16 DESTINATION AERODROME <div style="border: 1px solid black; width: 100px; height: 15px;"></div> TOTAL EET <div style="border: 1px solid black; width: 100px; height: 15px;"></div> <=	
18 OTHER INFORMATION <div style="border: 1px solid black; width: 100px; height: 15px;"></div>		ALTN AERODROME <div style="border: 1px solid black; width: 100px; height: 15px;"></div> 2ND ALTN AERODROME <div style="border: 1px solid black; width: 100px; height: 15px;"></div> <=	
SUPPLEMENTARY INFORMATION (NOT TO BE TRANSMITTED IN FPL MESSAGES)			
19 ENDURANCE <div style="border: 1px solid black; width: 100px; height: 15px;"></div> E/		PERSONS ON BOARD <div style="border: 1px solid black; width: 100px; height: 15px;"></div> P/	
SURVIVAL EQUIPMENT <div style="border: 1px solid black; width: 100px; height: 15px;"></div> D/		EMERGENCY RADIO <div style="border: 1px solid black; width: 100px; height: 15px;"></div> R/	
POLAR DESERT MARITIME JUNGLE <div style="border: 1px solid black; width: 100px; height: 15px;"></div> A/		JACKETS <div style="border: 1px solid black; width: 100px; height: 15px;"></div> N/	
DINGHIES <div style="border: 1px solid black; width: 100px; height: 15px;"></div> C/		LIGHT FLUORES UH VHF <div style="border: 1px solid black; width: 100px; height: 15px;"></div>)<=	
NUMBER CAPACITY COVER <div style="border: 1px solid black; width: 100px; height: 15px;"></div> <=		REMARKS <div style="border: 1px solid black; width: 100px; height: 15px;"></div> <=	
AIRCRAFT COLOR AND MARKINGS <div style="border: 1px solid black; width: 100px; height: 15px;"></div>		PILOT-IN-COMMAND <div style="border: 1px solid black; width: 100px; height: 15px;"></div> <=	
FILED BY <div style="border: 1px solid black; width: 100px; height: 15px;"></div>		ACCEPTED BY <div style="border: 1px solid black; width: 100px; height: 15px;"></div> ADDITIONAL INFORMATION <div style="border: 1px solid black; width: 100px; height: 15px;"></div>	

FAA Form 7233-4 (7-93)

Rysunek 2 Formularz planu lotu [2]

W początkowej części formularza jest miejsce na podanie adresatów do których miałby trafić plan lotu. Pierwsze trzy pola poprzedzające pole nr 7 wypełniają służby kontroli ruchu lotniczego.

W punkcie 7 należy podać znak rozpoznawczy statku powietrznego składający się z nie więcej niż 7 znaków alfanumerycznych. W przypadku polskiego statku powietrznego może to mieć postać SP-ARO.

W punkcie 8 podaje się przepisy wykonywania lotu i rodzaj lotu. Przepisy wykonywania lotu dzielą się na przepisy z widzialnością - VFR lub bez widzialności - IFR. W tym punkcie należy wstępnie zadeklarować czy w trakcie lotu będzie się zmieniało przepisy. Zasady według których wykonuje się lot można zmienić w określonym wcześniej w planie lotu punkcie na trasie przelotu. Rodzaj lotu podaje się jeżeli wymaga tego odpowiednia władza ATS i oznacza odpowiednimi literami.

Punkt 9 wpisuje się liczbę i typ statków powietrznych oraz kategoria turbulencji w śladzie aerodynamicznym.

Punkt 10 informuje służby ruchu lotniczego o wyposażeniu statku w urządzenia radiokomunikacyjne, radionawigacyjne, w pomoce podejścia do lądowania oraz o systemach dozoru przestrzeni powietrznej jakie są zainstalowane na statku powietrznym oraz dostępne mody pracy transpondera.

Punkt 13 lotnisko i godzina odlotu. Należy podać kod ICAO lotniska lub jeżeli lotnisku nie przydzielono takiego kodu wpisać ZZZZ i PODAĆ w punkcie 18 nazwę lotniska i lokalizację poprzedzoną skrótem DEP/ lub pierwszy punkt trasy jeżeli statek powietrzny nie wystartował z lotniska lub gdy plan lotu został złożony w locie. Jako czas podać przewidywaną godzinę odblokowania (EOBT) tzn. przewidywany czas rozpoczęcia przez statek powietrzny ruchu związanego z odlotem [2].. Jeżeli plan otrzymano od statku w locie, wpisać rzeczywisty lub przewidywany czas nad pierwszym punktem trasy, do której plan lotu się odnosi.

Punkt 15 – Trasa. Podać pierwszą prędkość przelotową tj. rzeczywistą prędkość przelotową i pierwszy poziom przelotu.

Dalej w polu „ROUTE” podaje się wybraną, zdefiniowaną odgórnie drogę lub drogi lotnicze, którymi będzie poruszał się statek powietrzny oraz inne punkty znajdujące się na zaplanowanej trasie przelotu opisane poprzez podanie nazwy pomocy nawigacyjnej, współrzędnych geograficznych lub namiaru i odległości na najbliższą pomoc nawigacyjną.

Pomoc nawigacyjną obok której planuje się przelot podaje się poprzez wpisanie jej kodu np. KORUB, TAMEG.

Jeżeli punkt na trasie przelotu opisuje się wykorzystując współrzędne geograficzne, podając stopnie i minuty, wykorzystuje się do tego 11 znaków pisarskich. Pierwsze 4 cyfry opisują szerokość w stopniach (2 cyfry) i minutach (również 2 cyfry), piąty znak określa czy punkt znajduje się na północnej (N) czy południowej (S) półkuli, szósty znak to zawsze „0”, natomiast ostatnie pięć znaków definiują długość geograficzną analogicznie jak w przypadku definiowania szerokości. Przykład: 1230N04512E. Jest również możliwość podania położenia według samych stopni, opis punktu jest wtedy o cztery cyfry krótszy.

Metoda opisu punktu trasy poprzez podanie namiaru na najbliższy punkt radionawigacyjny jest obecnie rzadziej spotykana. Podaje się nazwę punktu np. „BUG”, a następnie azymut magnetyczny w stopniach i odległość w milach morskich do tego punktu. Przykładowo punkt znajdujący się na kierunku magnetycznym 90° i w odległości 30 mil morskich od punktu radionawigacyjnego o nazwie BUG opisuje się w następujący sposób: BUG 090030.

W celu podania trasy ATS należy wpisać jej oznacznik kodowy (np. BNC1, A1, KODAP2A). Przed podaniem trasy jaką statek powietrzny będzie się przemieszczał należy wymienić punkt w jakim rozpocznie się przelot daną trasą, chyba że od momentu startu planuje się przelot daną trasą. Również należy podać punkt opuszczenia trasy, podobnie jak w przypadku deklarowania rozpoczęcia lotu po danej trasie lotniczej.

Zmiana poziomu lotu lub prędkości rzeczywistej. Punkt w którym planowana jest zmiana prędkości rzeczywistej o więcej niż 5% lub o 0.01 Macha lub planuje się zmianę poziomu, musi być podany w planie lotu. Podaje się zawsze obydwa wyżej wymienione parametry nawet gdy tylko jeden z nich ulegnie zmianie. Przykłady:

- HADDY/N0420F330
- 4602N07805W/N0500F350
- DUB180040/N0350M0840

Po ukośniku podaje się nową rzeczywistą prędkość przelotową oraz nowy poziom lotu.

Litery DCT odpowiadające angielskiemu słowu *Direction* wpisuje się pomiędzy punktami, z wyjątkiem gdy punkty te są opisane za pomocą współrzędnych namiarowych

lub geograficznych. Instrukcja jasno określa wymóg stosowania tego symbolu lecz można się spotkać, że służby lotnicze nie wymagają wpisywania DCT.

W punkcie, w którym planowana jest zmiana przepisów wykonywania lotu wpisuje się dodatkowo IFR lub VFR, zależy według jakich przepisów statek powietrzny w dalszym ciągu będzie kontynuował lot np. KODAK IFR.

Wznoszenie w przelocie opisuje się w następujący sposób np.:

- C/48N050W/M082F290F350
- C/48N050W/M082F290PLUS
- C/52N050W/M220F580F620.

Zaczyna się od litery C, po ukośniku podaje się punkt, według jednej z wcześniej opisanych formuł, po drugim ukośniku podaje się prędkość rzeczywistą statku powietrznego, dalej dwa poziomy (ang. *Flight Level*) określającą warstwę jaka będzie zajęta w trakcie manewru wznoszenia w przelocie. PLUS oznacza, że powyżej danego poziomu nastąpi wznoszenie w przelocie.

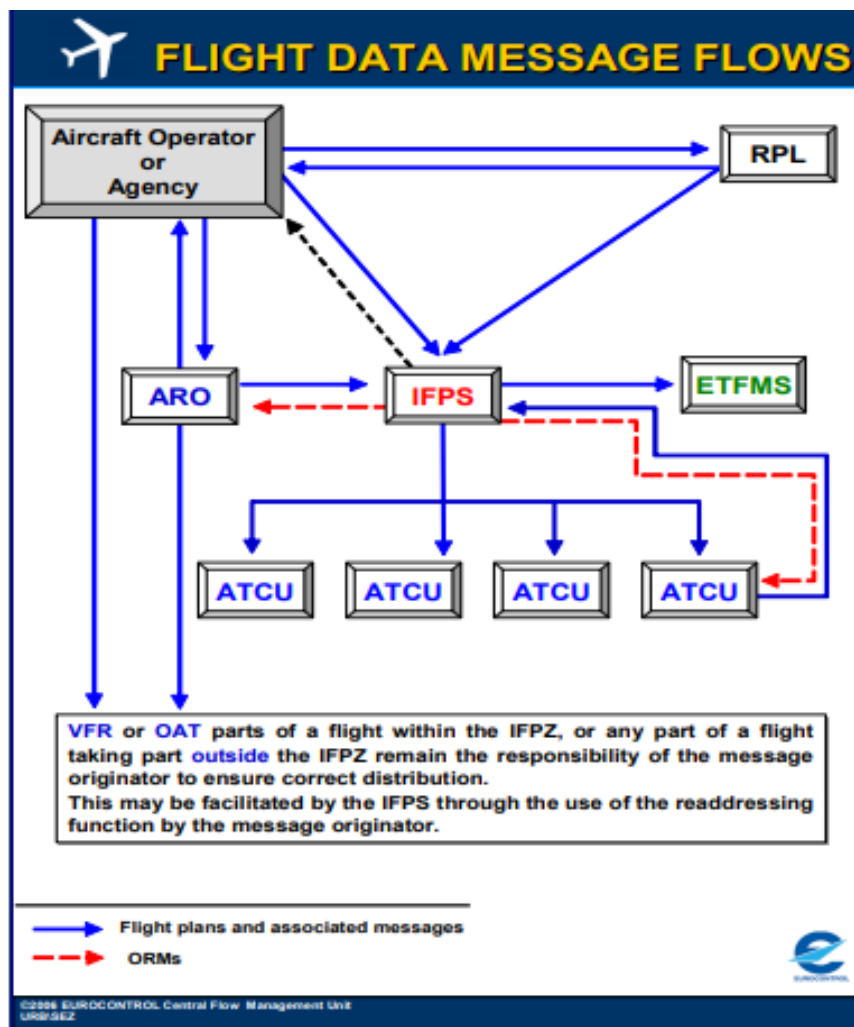
W punkcie 16 definiuje się lotnisko docelowe i całkowity przewidywany czas przelotu, oraz lotnisko/lotniska zapasowe W tym punkcie opisuje się lotnisko docelowe z wykorzystaniem czteroliterowego kodu ICAO nadanego lotnisku. Jeżeli lotnisko nie ma przypisanego kodu ICAO, należy wpisać ZZZZ, a w punkcie 18 podać nazwę i lokalizację lotniska poprzedzone znakiem DEST/. Analogicznie postępuje się przy deklarowaniu lotniska zapasowego, lecz w punkcie 18 wpisuje się znak ALTN/ zamiast DEST/ czy DEP/. Podaje się również planowany czas lotu.

Punkt 18. Można w nim określić inne bardziej szczegółowe informacje odnośnie trasy lotu. Wypełnienie tego punktu nie jest obligatoryjne.

Punkt 19. Określa zapas paliwa wyrażony 4 cyfrową liczbą czasu (godzin i minut) lotu na jaki wystarczy paliwa, ilość osób na pokładzie, informacje o kamizelkach ratunkowych, o radiowych urządzeniach ratowniczych, ilość, pojemność, rodzaj (przykryte czy nie), kolor tratw ratunkowych, kolor i charakterystyczne cechy wyglądu statku powietrznego, ewentualne uwagi oraz nazwisko kapitana statku powietrznego.

Po wypełnieniu formularza należy go przesłać do odpowiednich służb. W Polsce w tym celu utworzono Biura Odpraw Załóg – BOZ. Można to zrobić telefonicznie lub wysyłając wypełniony plan faxem. Polska Agencja Żeglugi Powietrznej utworzyła również elektroniczny system Integrated Web Briefing (IWB) który umożliwia wypełnienie oraz złożenie planu lotu z poziomu aplikacji. Obecnie większość planów lotu składana jest poprzez dedykowane do tego celu aplikacje lub poprzez wydzieloną sieć AFTN. W Stanach Zjednoczonych plan lotu również można wysłać faksem lub podać potrzebne informacje telefonicznie. Serwisy, które umożliwiają złożenie formularza online w USA to Flight Service 1800WxBrief oraz Lockheed Martin Flight Services [13].

W Europie zaimplementowany został system IFPS autoryzowany przez agencje EUROCONTROL [6]. Wszystkie plany lotu dla lotów IFR/GAT, które mają się odbyć w strefie IFPZ muszą zostać przetworzone przez IFPS. Każda informacja związana z planem lotu musi wyjść od operatora statku powietrznego (pilot lub odpowiednia agencja), który ma możliwość zgłoszenia tej informacji bezpośrednio do IFPS, do Biura Odpraw Załóg lub za pośrednictwem systemu do przetwarzania powtarzalnych planów lotu (RPL), tak aby dalej został automatycznie przesłany do IFPS. Plan musi zostać złożony 20 godzin przed czasem odblokowania. Obieg dokumentów przedstawiono na rysunku 3. Niebieskimi strzałkami oznaczono na nim przepływ wszelkich rodzajów depech dotyczących lotu. Czarne i czerwone strzałki pokazują przepływ wiadomości zwrotnych (ORM) wysłanych przez system IFPS do odpowiednich służb i operatora statku powietrznego (patrz rysunek 3). Strefa IFPZ obejmuje cały kontynent z wyjątkiem Łotwy, Białorusi i Rosji.



Rysunek 3 Przepływ danych w systemie IFPS [6]

Plan lotu i inne wiadomości związane z lotem może być przez systemy przetwarzany w dwóch formatach. W formacie ICAO oraz bardziej rozbudowanym formacie ADEXP utworzonego przez EUROCONTROL. Poniżej przedstawiono przykłady depeszy FPL w dwóch formatach.

(FPL-ACID-Flt Rules-Flight Type
 -AC Type/Wake Cat-Equip.&Capability
 -Departure EOBT
 -Speed/Altitude [sp] Route
 -Destination ETE [sp] Alternate(s)
 -Other Information)

Example:
 (FPL-TTT123-IS
 -C550/L-SDE1E2GHIJ3J5RWZ/SB1D1
 -KPWM1225
 -N0440F310 SS0XS5 SS0XS DCT BUZRD
 DCT SEY DCT HTO J174 ORF J121
 CHS EESNT LUNNI1
 -KJAX0214 KMCO
 -PBN/A1B1C1D1L1 DAT/1FANS2PDC
 SUR/260B RSP180 DOF/180217
 NAV/RNP2 REG/N123A SEL/BPAM
 CODE/A05ED7

Rysunek 4 Depesza FPL w formacie ICAO [4]

Opis Formatu ICAO:

- Format ICAO rozpoczyna „(FPL” oznaczające, że wiadomość to plan lotu
- ACID – Aircraft Identifier, kod identyfikujący dany statek powietrzny punkt 7 planu lotu
- Flt Rules – Flight Rules, przepisy według których wykonywany jest lot punkt 8
- Flight Type – rodzaj wykonywanego lotu
- AC Type – Type of Aircraft, Typ Statku powietrznego punkt 9a
- Wake Cat – Wake Turbulence, Kategoria turbulencji, punkt 9b
- Equip&Capability – Wyposażenie, punkt 10
- Departure – Lotnisko odlotu, punkt 13
- EOBT – punkt 13a
- Speed – prędkość przelotowa punkt 15
- Altitude – poziom podany jako FL punkt 15
- Route – Trasa punkt 15
- Destination – lotnisko docelowe
- ETE – całkowity czas przelotu
- Alternate(s) – lotniska zapasowe
- Other Information – Inne Informacje oraz informacje dodatkowe, punkt 18 i 19
- [sp] – odstęp, spacja

Format ADEXP (rysunek 5) dedykowany jest do wspomnianego wcześniej europejskiego systemu IFPS. Po słowie ROUTE podano trasę tak jak w planie lotu. Depesza FPL w formacie ADEXP zawiera bardziej szczegółowe informacje. Największą różnicą jest podanie szczegółowej listy punktów na trasie po słowie RTEPTS, wraz z wysokością przelotu nad danym punktem oraz czasem ETO (moment, w którym statek powietrzny znajdzie się przy określonym punkcie) podanym z datą i czasem określonym z dokładnością do jednej sekundy. W poniższym przykładzie po słowie RTEPTS jest więcej punktów niż po słowie ROUTE ponieważ dodatkowo wspomniana została każda pomoc nawigacyjna jaka leży na drodze lotniczej. Dana jest procedura odlotu (SID - Standard Instrument Departure). W linii rozpoczętej oznaczeniem ATSRT dane są pierwszy i ostatni punkt drogi lotniczej.

TITLE IFPL	-SRC FPL
-BEGIN ADDR	--TTLEET 0053
-FAC CFMUTACT	-RFL F350
-FAC EGZYTTE	-SPEED N0433
-FAC EGTZGZP	-FLTRUL I
-FAC EGZCCTE	-FLTYP S
-FAC EIAAYVYX	-ROUTE N0433F350 LIFFY5A LIFFY UL975 WAL
-FAC EIDWZPZX	M16 DOLAS L603 LAMSO
-FAC EIDWZQZX	-ALTRNT1 EHRD
-FAC EHAAZQLR	-EETFIR EGTT 0008
-FAC EHAAZQZX	-EETFIR EHAA 0042
-FAC EGZYADEX	-BEGIN RTEPTS
-FAC EGZYTTF0	-PT -PTID EIDW -FL F002 -ETO 170728093200
-FAC EGZYTAD	-PT -PTID OE -FL F050 -ETO 170728093402
-FAC EGZYPXAD	-PT -PTID DUB -FL F089 -ETO 170728093521
-FAC EGZYPXFO	-PT -PTID LIFFY -FL F194 -ETO 170728094000
-END ADDR	-PT -PTID IDEXA -FL F222 -ETO 170728094145
-ADEP EIDW	-PT -PTID GINIS -FL F243 -ETO 170728094319
-ADES EHAM	-PT -PTID NATKO -FL F256 -ETO 170728094427
-AOARCID EIN	-PT -PTID LYNAS -FL F273 -ETO 170728094556
-AOOPR EIN	-PT -PTID ROLEX -FL F280 -ETO 170728094741
-ARCID EIN60T	-PT -PTID OSLO -FL F295 -ETO 170728094855
-ARCTYP A320	-PT -PTID MALUD -FL F301 -ETO 170728094926
-CEQPT SDE2E3FGHIJ1RWXY	-PT -PTID EMLIP -FL F314 -ETO 170728095040
-EOBD 170728	-PT -PTID WAL -FL F326 -ETO 170728095146
-EOBT 0920	-PT -PTID MCT -FL F350 -ETO 170728095603
-FILTIM 272031	-PT -PTID XAPOS -FL F350 -ETO 170728095916
-IFPLID AT00424641	-PT -PTID NAPEX -FL F350 -ETO 170728100309
-ORGNID EINAOC	-PT -PTID DOLAS -FL F350 -ETO 170728101239
-ORIGIN -NETWORKTYPE AFTN -FAC	-PT -PTID ENITO -FL F294 -ETO 170728102005
EIDWEINU	-PT -PTID DIBAL -FL F290 -ETO 170728102014
-SEQPT L	-PT -PTID BUKUT -FL F252 -ETO 170728102149
-WKTRC M	-PT -PTID LAMSO -FL F227 -ETO 170728102251
-ARCADDR 4CA5C8	-PT -PTID ETPOS -FL F164 -ETO 170728102540
-IFP MODESASP	-PT -PTID SUGOL -FL F090 -ETO 170728102913
-OPR EIN	-PT -PTID *SPL -FL F007 -ETO 170728103744
-PBN A1B1D1L1S2	-PT -PTID EHAM -FL F000 -ETO 170728103838
-REG EIDVE	-END RTEPTS
-RMK TCAS	-SID LIFFY5A
-RVR 075	-ATSRT UL975 LIFFY WAL
-SEL EHPR	-ATSRT M16 WAL DOLAS
	-ATSRT L603 DOLAS LAMSO

Rysunek 5 Depesza FPL w formacie ADEXP [5]

2.2 Przegląd dostępnych rozwiązań

Istnieje szereg dostępnych aplikacji wspomagających [11, 15]:

- wyznaczenie trasy,
- złożenie planu lotu,
- przechowujących historię wykonanych przez użytkownika przelotów,
- umożliwiających szybki dostęp do szczegółowych informacji na temat wybranego lotniska,
- łatwy dostęp do depeszy NOTAM,
- pokazujących ostrzeżenia w razie zaplanowania zbyt niskiej wysokości przelotowej,
- zawierające inne pomocne funkcjonalności.

Grupą tego typu aplikacji są aplikacje EFB (ang. *Electronic Flight Bag*). Nazwa ta wzięła się z faktu zastąpienia ciężkich teczek z dokumentami papierowymi urządzeniami elektronicznymi. Z definicji [1] jest to elektroniczny system przeznaczony do użytku na pokładzie statku powietrznego składający się z urządzeń elektronicznych oraz oprogramowania przeznaczonego do wykonywania określonych funkcji. Urządzenia EFB mogą zapewnić pilotowi różnorodne informacje zaczynając od podstawowych obliczeń (np. zapasu paliwa, parametrów lotu). W przeszłości czynności realizowane przez systemy EFB wykonywał pilot na podstawie dokumentów lub na podstawie danych dostarczonych przez dyspozytora linii lotniczej. Aplikacja EFB może wykorzystywać różne technologie, urządzenia, formaty danych i sposoby komunikacji. Aby system można było nazwać systemem EFB musi mieć zaimplementowane oprogramowanie typu A i/lub typu B.

Federalna agencja lotnictwa (FAA – Federal Aviation Administration) w swoich wytycznych [1] wyróżnia trzy klasy sprzętu elektronicznego na których zainstalowane jest oprogramowanie oraz trzy typy oprogramowania. Klasy sprzętu:

Klasa 1 Są to komercyjne szeroko dostępne urządzenia jak np. tablety, komputery osobiste i różnego rodzaju przenośne urządzenia elektroniczne które są pozbawione aprobaty FAA. Nie są one zintegrowane z systemami samolotu, mogą być ładowane z sieci elektrycznej statku powietrznego. Urządzenia na

których zainstalowano aplikacje typu B muszą być odpowiednio zabezpieczone w trakcie krytycznych faz lotu i nie mogą mieć wpływu na system sterowania statkiem powietrznym.

Klasa 2 Podobnie jak w przypadku klasy 1 są to szeroko dostępne przenośne urządzenia elektroniczne których konstrukcja, proces produkcji, instalacji nie został zaaprobowany przez FAA. Zwykle są one w pewien sposób zamontowane, jednak muszą mieć możliwość łatwego zamontowania i usunięcia przez załogę. Mogą być czasowo podłączone do sieci elektrycznej samolotu w celu naładowania baterii. Mogą być połączone bezprzewodowo lub przewodowo do portów danych i do zainstalowanych anten. Nie są one częścią projektu statku powietrznego.

Klasa 3 EFB zainstalowane zgodnie z regulacjami określającymi zdolność do lotu.

Typy oprogramowania:

- Typ A Aplikacje używane podczas planowania lotu, na ziemi lub podczas niekrytycznych faz lotu.
- Typ B Zawierają informacje lotnicze niezbędne pilotowi podczas lotu, przeznaczone szczególnie do użytku na etapie planowania lotu i we wszystkich fazach lotu.
- Typ C Software zatwierdzony przez FAA.

Istnieją liczne przykładowe aplikacje wspomagające pracę pilotów. Poniżej przedstawiono kilka przykładów:

AvPlan jest australijską aplikacją spełniającą wymogi rządowej organizacji do spraw bezpieczeństwa lotów *Civil Aviation Safety Authority*. Aplikacja ma następujące funkcjonalność [11]:

- Zawiera bazę danych o wszystkich australijskich lotniskach, pomocach nawigacyjnych, wzrokowych pomocach nawigacyjnych
- Dostęp do map lotniczych WAC, VTC, VNC, TAC, ERC
- Połączenie z AIS
- Wyświetlanie aktualnej pozycji na mapie
- Możliwość automatycznego pobrania depeszy NOTAM
- Wgląd w informacje meteorologiczne
- Ostrzeżenia o przeszkodach terenowych
- Bazę danych samolotów, wraz z ich parametrami technicznymi
- Obliczenie wyważenia samolotu
- Ostrzeżenia SARTIME, jest to czas po jakim zostanie rozpoczęta akcja ratownicza, ostrzeżenie przypomina pilotowi o zatrzymaniu licznika

- Elektroniczna edycja wybranej trasy lotu
- Złożenie planu lotu
- Import istniejącego już planu lotu
- Eksport trasy planu lotu do podręcznego urządzenia GPS firmy Garmin

SkyVector jest aplikacją zintegrowaną z serwisem xbrief1800 co umożliwia złożenie wcześniej wypełnionego planu lotu. Aplikacja wyświetla lotniczą mapę świata, z oznaczonymi na niej lotniskami, punktami nawigacyjnymi, stacjami meteorologicznymi itd. W prosty sposób można wybrać punkty na trasie i automatycznie wygenerować trasę przelotu w formie wymaganej w 15 punkcie formularza planu lotu. Możliwe jest wypełnienie całego formularza. Pozwala sprawdzić warunki pogodowe, oraz strefy z wprowadzonymi ograniczeniami dotyczącymi wykonywania lotu, a nawet ceny paliwa na poszczególnych lotniskach [16].

Inne przykłady tego typu aplikacji to **SkyDemon**, utworzona przez Polską Agencję Żeglugi Powietrznej, Integrated Web Briefing i AOPA Flight Planner.

2.3 Planowanie misji

Planowanie misji bojowej statku powietrznego składa się z wielu aspektów [9]. Jednym z nich może być np. przelot przez przestrzeń powietrzną, w której będzie musiał wykonać lot przestrzegając cywilnych przepisów. W dalszej części podrozdziału zostaną przedstawione zagadnienia związane z planowaniem misji wojskowego statku powietrznego.

Wyróżnia się trzy rodzaje rozkazów związanych z misją, jakie otrzymuje jednostka:

- 1) Zarządzenie przygotowawcze, angielskie Warning Order (WORNO) – pomaga w przygotowaniach do operacji powietrznej. Zaznajamia uczestników misji o szczegółach i z czasem w jakim mają nastąpić poszczególne działania. Po otrzymaniu rozkazu WORNO dowódca misji przekazuje go załogą i personelowi naziemnemu. Poniżej przedstawiono elementy zarządzenia przygotowawczego:

- I. Określenie typu misji;
- II. Określenie czasu kiedy zostanie wykonana misja;
- III. Zadania dla załogi i statku powietrznego;
- IV. Określenie przygotowań jakie należy podjąć;
- V. Harmonogram czasowy do przestrzegania;
- VI. Indywidualne zadania i obowiązki.

- 2) Rozkaz Bojowy, angielskie Operation Order (OPORD) – zawiera pięć punktów w których są zawarte dokładne informacje niezbędne podczas wykonywania misji:
- I. Sytuacja;
 - II. Zadanie;
 - III. Realizacja;
 - IV. Zabezpieczenie logistyczne;
 - V. Dowodzenie i łączność.
- 3) Zarządzenie bojowe, Fragmentary Order (FRAGO) – wprowadza zmiany do rozkazu bojowego już po wydaniu tego rozkazu.

W trakcie analizowania sposobu wykonania misji należy rozważyć aspekty takie jak cel misji, zadania do zrealizowania, dostępne informacje o rozmieszczeniu i środkach obrony przeciwnika, ukształtowaniu i pokryciu terenu, dostępności wsparcia przez siły własne, ograniczenia czasowe oraz względy cywilne, zgodnie z akronimem stosowanym przez siły zbrojne USA określamy je odpowiednio: METT-TC (Mission, Enemy, Terrain&Weather, Troops, Time Available, Civilian Considerations). Ze względu na charakter projektu zrealizowanego w ramach pracy, w dalszych rozważaniach skupiono się jedynie na aspekcie związanym z terenem oraz przeciwnikiem.

Zadanie do zrealizowania (Mission) – dowódca i podwładni muszą dobrze znać swoje cele i zadania. W tym punkcie analizy rozpatruje się krytyczne zadania jakie należy wykonać oraz podział zadań pomiędzy uczestnikami misji.

Przeciwnik (Enemy) – na podstawie dostępnym źródeł informacji w szczególności należy rozpatrzyć siły przeciwnika pod względem:

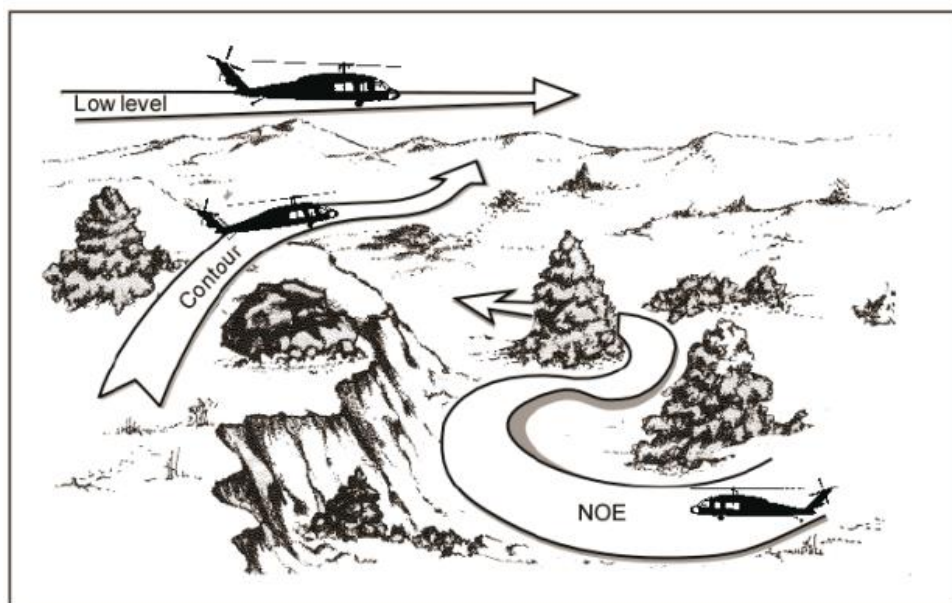
- a) sposób identyfikacji,
- b) lokalizacja,
- c) dyspozycja,
- d) siła,
- e) morale,
- f) możliwości,
- g) skład,
- h) prawdopodobne kierunki działania.

Teren i warunki atmosferyczne (Terrain&Weather)

- a) dominujące punkty terenowe,
- b) naturalne cechy,
- c) punkty z których może nastąpić atak przeciwnika,
- d) zabudowania,
- e) ważne skrzyżowania,
- f) **zagrożenia dla lotu,**
- g) siły własne,
- h) siły przeciwnika,
- i) pogoda i widoczność.

Problemem koniecznym do rozwiązania podczas planowania misji jest ograniczenie możliwości wykrycia naszego statku powietrznego przez przeciwnika. W tym celu można wykorzystać między innymi ukształtowanie terenu. Różne ukształtowanie terenu może pozwolić ukryć się przed wzrokiem lub przed wiązką radarową przeciwnika jednak również stanowi zagrożenie dla bezpieczeństwa lotu. Niski pułap lotu zmniejsza ryzyko wykrycia statku powietrznego i co za tym idzie zmniejsza również ryzyko użycia przeciwko niemu środków ogniowych przeciwnika. Zwiększa się jednak ryzyko zderzenia z przeszkodą terenową. Lot na niskim pułapie utrudnia także nawigację, wymusza poruszanie się z mniejszą prędkością oraz jest bardziej stresujący dla pilota. Instrukcja sił powietrznych USA [9] dotycząca planowania misji lotniczych wyróżnia trzy modele lotu w terenie (rysunek 6):

- Low Level Flight – wykonywany ze stałą prędkością, wysokością, wykonywany na wysokości od 80 do 200 stóp nad najwyższą przeszkodą terenową.
- Contour Flight – stała prędkość, pułap lotu jest zmienny, zależny od ukształtowania terenu i przeszkód. Lot wykonywany na wysokości od 25 do 80 stóp.
- Nap of The Earth Flight – wykonywany ze zmienną prędkością, możliwie najbliżej ziemi przynajmniej 25 stóp ponad roślinnością.



Rysunek 6 Modele lotu w terenie [9]

3 Założenia

Na podstawie przeprowadzonej analizy dostępnych informacji o systemach planowania lotu oraz planowania misji przyjęto, iż celem dalszych działań będzie stworzenie dwóch niezależnych aplikacji programowych. Ich przeznaczeniem jest głównie wykorzystanie w celach dydaktycznych i w związku z powyższym mają posiadać głównie charakter poglądowy. Założono również, iż aplikacje mogą być w przyszłości rozwijane, jednak ze względu na sposób ich tworzenia, wynikający z ograniczonego czasu realizacji oraz jednoosobowego wykonania, będą z pewnością mniej zaawansowane w porównaniu do aplikacji napisanych przez wyspecjalizowane firmy zatrudniające wieloosobowe zespoły programistów.

W niniejszym rozdziale przedstawiono funkcjonalności jakie mają spełniać aplikacje oraz ograniczenia jakie dla nich przyjęto.

Do pracy nad projektem zdecydowano się na wybór środowiska Matlab ponieważ daje ono możliwość skorzystania z zestawu narzędzi dedykowanych do pracy z mapami tj. Mapping Toolbox oraz Antenna Toolbox ułatwiający obliczenia związane z propagacją sygnału radarowego w przestrzeni. Sam język programowania jest pod wieloma względami bardzo przystępny oraz był wielokrotnie wykorzystywany w toku studiów magisterskich i inżynierskich. Matlab posiada także wygodne narzędzia do pracy nad interfejsem graficznym aplikacji. Drugim rozważanym środowiskiem był QtCreator bazujący na języku C++.

3.1 Założenia do aplikacji przeznaczonej do planowania lotu

Zasadniczym celem aplikacji jest przyspieszenie procedury składania planu lotu poprzez umożliwienie intuicyjnego i szybkiego wyznaczenia trasy lotu oraz poprzez dokonanie przydatnych obliczeń. Oprogramowanie zawiera kilka przydatnych podczas planowania lotu funkcjonalności:

1. Interfejs identyczny jak papierowa wersja formularza planu lotu
2. Specjalne okno do łatwego wyboru wyposażenia statku powietrznego (punkt 10 planu lotu)

3. Interaktywny dostęp do mapy, na której oznaczono znaczące pomoce nawigacyjne. Posługując się myszką od komputera możliwy jest wybór kolejnych punktów trasy klikając na mapę.
4. Nakreślenie trasy przelotu na mapie.
5. Możliwość wybrania lotniska startowego i końcowego z listy.
6. Generowanie wykresu elewacji terenu dla trasy przelotu umożliwiające unikanie przeszkód terenowych i pozwalający na określenie minimalnego bezpiecznego poziomu lotu.
7. Edycję tabeli z informacjami odnoszącymi się do każdego z wybranych punktów trasy: współrzędne geograficzne punktu lub nazwę znaczącego punktu nawigacyjnego jeśli statek powietrzny nad takowym przelatuje, odległość między punktami, kurs jakim porusza się statek powietrzny, prędkość statku powietrznego podaną w kilometrach, czas przelotu pomiędzy punktami, poziom lotu. Informacje o czasie trwania przelotu pomogą obliczyć pilotowi ilość paliwa jakie musi zatankować. Posiadany zapas paliwa także należy podać w formularzu planu lotu w punkcie 18.
8. Opisanie trasy zgodnie z wytycznymi nr 5 prezesa urzędu lotnictwa cywilnego².
9. Wygenerowanie na podstawie zebranych danych depeszy FPL w formacie zgodnym z wymaganiami ICAO.

Przyjęto również ograniczenia upraszczające, które umożliwia zakończeniu projektu w limitowanym okresie czasu przeznaczonym na realizację pracy dyplomowej.

1. Aplikacja została opracowana jedynie dla obszaru podległego pod FIR Warszawa. Opracowanie jej dla większego obszaru wymagałoby odpowiednio większego nakładu pracy, pozyskania danych o położeniu lotnisk i punktów radionawigacyjnych innych państw oraz pobrania większych cyfrowych map elewacji DTED do których brakuje dostępu.
2. Nie są dostępne wszystkie możliwości opisanie trasy lotu. Brak opcji wyboru drogi lotniczej. Pominięto opcje podania współrzędnych punktu według namiaru na najbliższy punkt radionawigacyjny, ponieważ już rzadziej się ten opis stosuje. W przypadku wznoszenia w przelocie pilot musi opisać je ręcznie. Punkt zmiany przepisów według których wykonuje się lot także należy wprowadzić ręcznie.
3. Baza danych obejmuje tylko 14 największych lotnisk w Polsce.

4. Cyfrowa mapa elewacji terenu nie pokrywa południowej części Polski poniżej równoleżnika 50°N.

3.2 Założenia do aplikacji wspomagającej planowanie misji

Aplikacja ma celu dobranie trasy statku powietrznego i określenie wysokości przelotu, która będzie optymalna pod względem ryzyka zderzenia z przeszkodą terenową i ryzykiem wykrycia przez radar przeciwnika. W związku z powyższym do realizacji tak postawionego zadania koniecznym jest:

- 1 Dostęp do mapy 3D dokładnie zobrazowująca ukształtowanie terenu;
- 2 Obliczenia i zobrazowanie terenu wykonywane dla jednej zadanej wysokości przelotu. W celu sprawdzenia obszaru, w którym statek powietrzny zostanie wykryty dla innej wysokości przelotu, należy powtórzyć obliczenia;
- 3 Możliwość rozmieszczenia dowolnej ilości radarów w różnych miejscach;
- 4 Wyznaczenie dla zadanej wysokości przelotowej żółtym kolorem przeszkód terenowych, a czerwonym obszaru w którym występuje ryzyko wykrycia przez radar. Jeżeli obszar jest bezpieczny nie jest wtedy wyróżniony żadnym kolorem i w tym rejonie na zadanej wysokości statek powietrzny bezpiecznie (bezpieczniej) może wykonać misję;
- 5 Wyznaczenie punktów trasy statku powietrznego.

Planowanie całości misji to rozwinięty proces angażujący nie jedną osobę i obejmujący wiele zagadnień, jak chociażby widoczność wzrokowa, obraz radarowy terenu, zabezpieczenie logistyczne misji, zaplanowanie tankowania w powietrzu, określenie trasy przelotu do celu, uzbrojenie, łączność itd. W pracy podjęto się rozwiązywania problemu jaki może wystąpić w potencjalnie najniebezpieczniejszym etapie misji, w którym statek powietrzny będzie zagrożony oddziaływaniem obrony przeciwlotniczej przeciwnika. Ze względu na trudności natury technicznej i ograniczony czas skupiono się na samym problemie uniknięcia wykrycia przez radar. Uniknięcie opromieniowania przez wiązkę radarową praktycznie uniemożliwi działanie sporej części środków ogniowych przeciwnika i pozwoli wykorzystać element zaskoczenia, stąd priorytet jaki nadano temu zagadnieniu.

4 Instrukcja obsługi aplikacji

Na następnych stronach zostaną dokładnie przedstawione funkcjonalności, sposób zrealizowania założeń oraz instrukcja obsługi poszczególnych aplikacji.

4.1 Aplikacja PLAN LOTU

Aplikacja PLAN LOTU składa się z trzech części. Głównym elementem jest okno wzorowane na papierowej wersji formularza planu lotu (rysunek 7). Wszystkie dane można wprowadzić ręcznie jednak część pól może zostać uzupełnionych dzięki wykorzystaniu dwóch dodatkowych narzędzi, które dalej zostaną szczegółowo omówione. Poniżej przedstawiono wygląd formularza planu lotu w wersji elektronicznej. Trzy górne pola pozostawiono puste, ponieważ te pola wypełnia odpowiednia służba kontroli ruchu powietrznego do której plan został zgłoszony. Pole 7 „AIRCRAFT IDENTIFICATION” wypełnia wpisując właściwy identyfikator lotu.

The screenshot shows a MATLAB App window titled "MATLAB App" with a standard Windows-style title bar (minimize, maximize, close buttons). The application contains a form for creating a flight plan, organized into several sections:

- Top Section:** Includes fields for "ADDRESSEE(S)", "FILING TIME", "ORIGINATOR", "7. AIRCRAFT IDENTIFICATION" (containing "klm343"), "FLIGHT RULES" (dropdown with "I"), and "TYPE OF FLIGHT" (dropdown with "S").
- Second Section:** Includes "9 NUMBER" (containing "1"), "TYPE OF AIRCRAFT" (containing "B707"), "WAKE TURBULENCE CAT" (dropdown with "L"), and "10 EQUIPMENT" (containing "SCDE1J1/LB1G2" with a "Select eq." button).
- Third Section:** Includes an "Open route planning tool" button, "13 DEPARTURE AERODROME" (containing "EPSC"), and "TIME hhmm" (containing "1400").
- Fourth Section:** Includes "15 CRUISING SPEED" (containing "K600") and "LEVEL" (containing "210").
- ROUTE Section:** A large text field containing the route "BAVGO 5208NO1852E POBOK DCT GOVR/K0500F190".
- Bottom Section:** Includes "16 DESTINATION AERODROME" (containing "EPRZ"), "TOTAL EET" (containing "0110"), "ALTN AERODROME" (containing "EPLB"), and "2ND ALTN AERODROME" (empty). Below this is "18 OTHER INFORMATION" (containing "PBN/B6").
- 19 ADDITIONAL INFORMATION:** This section is divided into several sub-sections:
 - ENDURANCE hhmm:** "E/" containing "0200".
 - PERSONS ON BOARD:** "P/" containing "220".
 - EMERGENCY RADIO:** Includes checkboxes for "UHF" (checked), "VHF" (checked), and "ELT" (checked).
 - SURVIVAL EQUIPMENT:** Includes checkboxes for "POLAR" (unchecked), "DESERT" (unchecked), "MARITIME" (checked), and "JUNGLE" (unchecked).
 - DINGHIES:** Includes "NUMBER" (containing "13"), "CAPACITY" (containing "330"), "COVER" (checked), and "COLOUR" (containing "YELLOW").
 - AIRCRAFT COLOUR AND MARKINGS:** "A/" containing "WHITE".
 - REMARKS:** "N/" (empty).
 - PILOT IN COMMAND:** "C/" containing "DENKE".

At the bottom right of the form, there is a button labeled "Generate ICAO format FL message" and an "OK" button.

Rysunek 7 Elektroniczny formularz planu lotu

7a „FLIGHT RULES” oraz 7b “TYPE OF FLIGHT” zostają wypełnione jedną z liter z rozwijanej listy (rysunki 8 i 9).

7. AIRCRAFT IDENTIFICATION: klm343
 9 NUMBER: 1
 TYPE OF AIRCRAFT: B707
 WAKE TURBULENCE CAT: L
 TIME: 1400
 13 DEPARTURE AERODROME: EPSC
 10 EQUIPMENT: SCDE1J1/LB1G2
 Select eq.
 Open route planning tool

Rysunek 8 Lista rozwijana "FLIGHT RULES"

7. AIRCRAFT IDENTIFICATION: klm343
 9 NUMBER: 1
 TYPE OF AIRCRAFT: B707
 WAKE TURBULENCE CAT: L
 TIME: 1400
 13 DEPARTURE AERODROME: EPSC
 15 CRUISING SPEED: 15
 LEVEL: 15
 TYPE OF FLIGHT: S
 10 EQUIPMENT: SCDE1J1/LB1G2
 Select eq.
 Open route planning tool

Rysunek 9 Lista rozwijana "TYPE OF FLIGHT"

MATLAB App

EQUIPMENT

- ☐ N: NIL
- ☒ S: VHF RTF/VOR/ILS
- ☐ A: GBAS
- ☐ B: LPV (APV/SBAS)
- ☒ C: LORAN
- ☒ D: DME
- ACARS Data Link
- ☒ E1: FMC Waypoint Reorting (FMC WPR)
- ☐ E2: Flight Information services (D-FIS)
- ☐ E3: Pre-departure Clearance (PDC)
- ☐ F: ADF - Automatic Direction Finder
- ☐ G: GNSS - Global Navigation Satellite System
- ☐ H: HF RTF - High Frequency Radio Telefon
- ☐ I: Inertial Navigation
- CPDCL - Controller Pilot Data Link
- ☒ J1: ATN VDL Mode 2
- ☐ J2: FANS 1/A HFDL
- ☐ J3: FANS 1/A VDL Mode A
- ☐ J4: FANS 1/A VDL Mode 2
- ☐ J5: FANS 1/A INMARSAT
- ☐ J6: FANS 1/A MTSAT
- ☐ J7: FANS 1/A IRIDIUM
- ☐ K: Microwave Landing System
- ☒ L: Instrumental Landing System
- ☐ M1: ATC RTF INMARSAT
- ☐ M2: ATC RTF MTSTAT
- ☐ M3: ATC RTF Iridium
- ☒ O: VOR
- ☐ R: RNAV/RNP Capable (PBN)
- ☐ T: TACAN
- ☐ U: UHF Radio Comm
- ☒ V: VHF Radio Comm
- ☐ W: RVSM
- ☐ X: MNPS
- ☐ Y: VHF 8.33 kHz Radio Channel Spacing
- ☐ Z: Other

SURVEILLANCE

- ☐ NIL
- ☐ A: MODE A
- ☐ C: MODE A/C
- ☐ E: MODE S, ID, Alt and Squitter
- ☐ H: Mode S, ID, Alt and Enhanced Squitter
- ☐ I: Mode S, ID, no Alt
- ☒ L: Mode S, ID, Alt, Squitter and Enhanced Surv
- ☐ P: Mode S, Alt, no ID
- ☐ S: Mode S, ID and Alt
- ☐ X: Mode S, no ID no Alt
- ☐ None

ADS-B/1090 MHZ

- ☒ B1: ADS-B, 1090 MHz Out
- ☐ B2: ADS-B, 1090 MHz Out and In
- ☐ None

ADS-B/UAT

- ☐ U1: ADS-B, 978 MHz UAT Out
- ☐ U2: ADS-B, 978 MHz UAT Out and In
- ☒ None

ADS-B/VDL M4

- ☐ V1: ADS-B, VDL Mode 4 Out
- ☐ V1: ADS-B, VDL Mode 4 Out and In
- ☒ None

Equipment: SCDE1J1

Surveillance: LB1G2

Confirm

Clear

Rysunek 10 Narzędzie do wyboru wyposażenia jakie znajduje się na statku powietrznym

Punkt 10 czyli wyposażenie w urządzenia radiokomunikacyjne, nawigacyjne, pomoce podejścia oraz wyposażenie i możliwości dozoru wypełnia się za pośrednictwem dodatkowego okna aplikacji (rysunek 10), które otwiera się po wciśnięciu przycisku „Select eq.”. Narzędzie tego typu bardzo ułatwia wybór wyposażenia, jakie znajduje się na statku powietrznym ze względu na wiele rodzajów urządzeń i systemów w jakie może być wyposażony.

Wyposażenie i możliwości statku powietrznego dzieli się na dwie kategorie. Pierwsza z nich to „Urządzenia radiokomunikacyjne i nawigacyjne oraz pomoce podejścia oraz możliwości” po lewej stronie okna, pod napisem „Equipment”, natomiast druga to „Wyposażenie i możliwości dozoru” po prawej stronie określone jednym słowem „SURVEILLANCE”. W polu 10 formularza planu lotu obydwie grupy oddzielone są ukośnikiem „/”, z lewej strony ukośnika są urządzenia z grupy pierwszej. Zaznaczając pole „S: VHF RTF/VOR/ILS” jednocześnie zaznaczane są trzy inne pola tj. „O: VOR”, „V: VHF”, „L: ILS”. Litera „S” oznacza, że statek powietrzny posiada standardowe wyposażenie jakim jest radiostacja VHF oraz instrumenty pozwalające na współpracę z systemem ILS i VOR. Pole o „N:NIL” oznacza, że statek powietrzny nie posiada wyposażenia z danej grupy (np. w wyniku awarii) i w formularzu wpisana zostanie litera „N”. W grupie urządzeń służących do dozoru przestrzeni powietrznej pole „N: NIL” działa w ten sam sposób. Znajdują się tam cztery prostokątne panele spośród których można wybrać tylko jeden typ sprzętu lub jego brak oraz dwa pola które można zaznaczać w dowolny sposób. Po wciśnięciu przycisku „Confirm” dane są przekazywane do formularza planu lotu.

Pole „TIME” określa czas odblokowania statku powietrznego, informacja jest wpisywana ręcznie w odpowiednim formacie tj. dwie litery określające godzinę i następne dwie określające minuty (hhmm).

Do wypełnienia pola 15 i pola 16 służy narzędzie uruchamiane po wciśnięciu przycisku „Open route planning tool”, wyjątkiem są pola odnoszące się do lotnisk zapasowych („DEPARTURE AERODROME”), które wypełnia się ręcznie. Narzędzie pomaga wygenerować trasę lotu według norm ICAO. Podgląd trasy możliwy jest na bieżąco w tabeli, na której w pierwszej kolumnie pokazana jest nazwa pomocy nawigacyjnej lub „-” jeżeli trasa nie przebiega nad taką pomocą, w drugiej i trzeciej kolumnie szerokość oraz długość geograficzna, czwartej odległość między punktami lub między lotniskiem

startowym a pierwszym punktem trasy, a w kolejnych kurs, prędkość przelotową, czas lotu pomiędzy punktami a w ostatniej kolumnie poziom lotu. Wygląd okna widać na rysunku nr 11.

Input Fields:

- Lon: 50.12, Lat: 21, No: 1
- Cruising speed [km/h]: 500, Altitude [FL]: 190
- Departure aerodrome: Szczecin Goleniów SZZ
- Destination aerodrome: Rzeszów Jesionka RZE

Route Table:

No	RNAV	Lat.	Lon	Distance[km]	Course	Cruise speed	Time	A
1	BAVGO	53.2412	18.2376	224.2387	98.5484	600	00:22	2
2	-	52.1414	18.8644	129.3768	160.6927	600	00:12	2
3	POBOK	50.6388	20.184	190.5213	150.7585	600	00:19	2
4	GOVRI	50.1157	21.0011	82.0976	134.7963	500	00:09	1
5	EPRZ	50.1111	22.0228	72.8568	270.7905	ładowanie	00:08	-

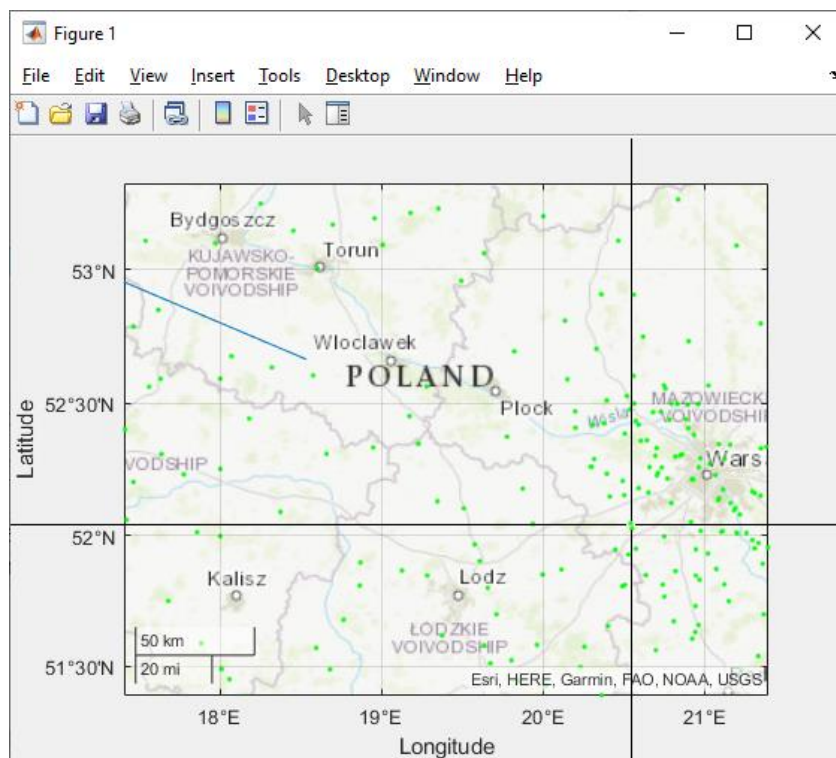
Summary: EET hhmm: 0110, Total distance [km]: 699.1

Route: BAVGO 5208N01852E POBOK DCT GOVRI/K0500F190

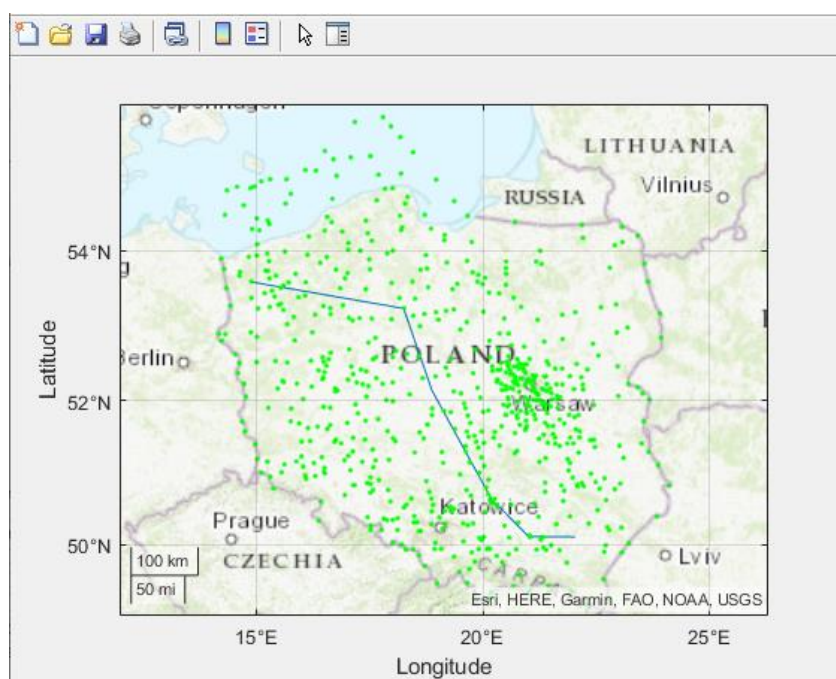
Rysunek 11 Narzędzie do planowania trasy

W celu wygenerowania trasy należy po otwarciu okna:

- 1) w pierwszej kolejności podać prędkość przelotową podawaną w kilometrach na godzinę (pole „Cruising speed [km/h]”) oraz poziom lotu (pole „Flight level [FL]”). Obydwa pola umieszczone w prawym górnym rogu aplikacji.
- 2) Drugim krokiem jest wybór lotniska startowego i docelowego z dwóch list. Na każdej liście do wyboru jest 14 największych lotnisk w Polsce.
- 3) Aby dodać punkt trasy należy kliknąć na „Add point”, a następnie wybrać punkt na mapie. Zielone kropki oznaczają pomoce nawigacyjne. Prędkość przelotową lub poziom lotu można zmienić przed wyborem każdego z punktów w polach wspomnianych w punkcie pierwszym, a zmiana ta zostanie uwzględniona w opisie trasy.



Rysunek 12 Wybór punktu na trasie



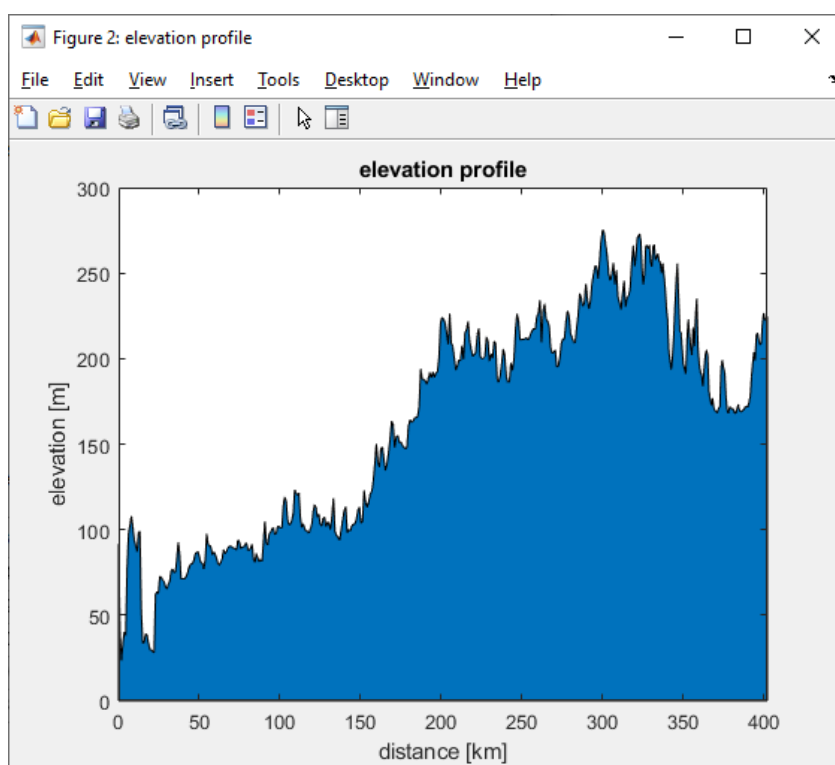
Rysunek 13 Całość trasy ze Szczecina do Rzeszowa

- 4) Czwartym krokiem jest obliczenie długości trwania całego lotu. Wciskając przycisk „Summary” obliczony zostanie całkowity czas trwania lotu, oraz dystans jaki pokona samolot. Obliczenie dystansu ma charakter poglądowy, pomocny przy późniejszym

wglądzie w wykres elewacji terenu, gdyż dane te nie są wymagane przy wypełnianiu formularza planu lotu.

- 5) Przycisk „Route” powoduje wygenerowanie opisu trasy.
- 6) Na koniec należy zatwierdzić wyniki przyciskiem „Confirm” w dolnej części okna, po czym nastąpi przesłanie danych do formularza.
- 7) Usunięcie punktu z listy możliwe jest poprzez wybranie jego liczby porządkowej i wciśnięciu klawisza „Delete point”.

Inną ważną funkcjonalnością omawianej części aplikacji jest możliwość wygenerowania profilu terenu na trasie przelotu poprzez wciśnięcie przycisku pod napisem „Generate terrain profile”. Wyświetlony wykres daje pogląd osobie planującej lot na którym etapie trasy może spodziewać się przeszkód terenowych, jeżeli poziom lotu byłby realizowany w bezpośredniej bliskości terenu. Wykres generowany jest z wykorzystaniem numerycznego modelu terenu GMTED2010.



Rysunek 14 Wykres elewacji terenu

Informacje w punkcie 18 i 19 formularza należy wprowadzić ręcznie. Jeżeli zadeklarowany zapas paliwa będzie mniejszy niż czas trwania lotu, aplikacja zasygnalizuje błąd. Po wprowadzeniu wszystkich danych można wygenerować depeszę

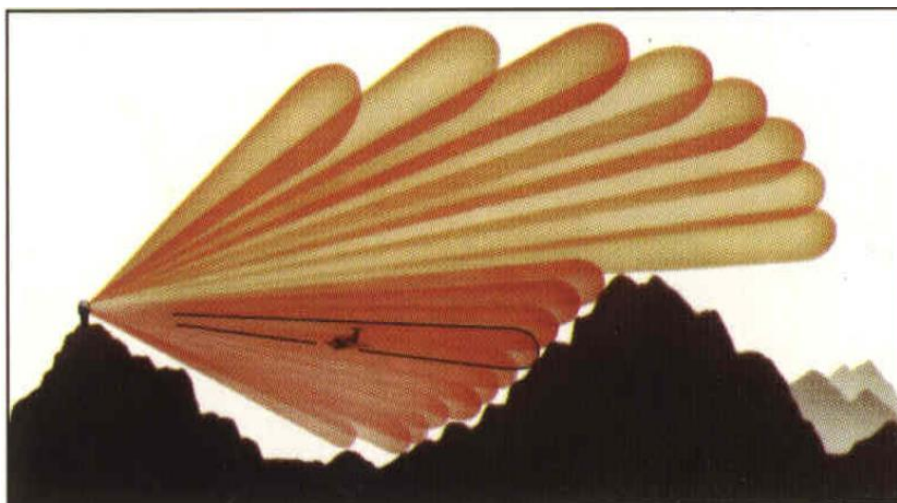
FPL w formacie ICAO, używając przycisku „OK” w prawym dolnym rogu elektronicznego formularza. Depesza zostanie zapisana w pliku tekstowym.



Rysunek 15 Depesza FPL zapisana w pliku tekstowym

4.2 Aplikacja wspomagającej planowanie misji

Aplikacja pomaga uniknąć wiązki radarowej przez statek powietrzny dzięki wykorzystaniu ukształtowania terenu na swoją korzyść. Lecąc na odpowiedniej wysokości statek powietrzny może ukryć się za nierównościami terenu i znaleźć się w cieniu radiolokacyjnym co uniemożliwi jego wykrycie. Przykładową sytuację przedstawiono na rysunku 16.

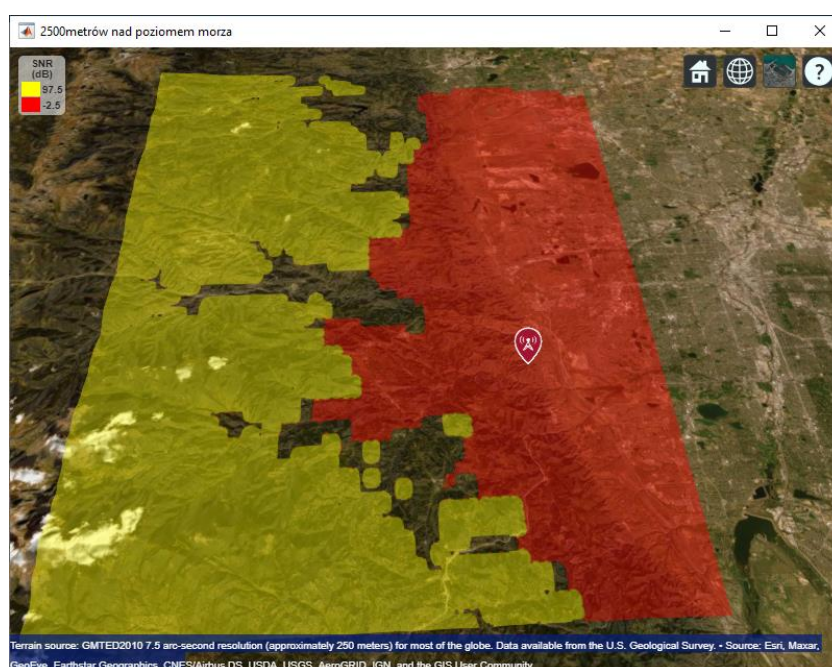


Rysunek 16 Ograniczenie zasięgu radaru przez teren

Aplikacja daje możliwość rozmieszczenia w terenie dowolnej ilości stacji radiolokacyjnych (dodatkowa ilość radarów spowolni działanie programu ze względu na konieczność wykonania dla każdego z nich osobnych obliczeń) oraz ich usunięcie. Drugą funkcjonalnością jest możliwość zaznaczenia punktów trasy samolotu na mapie 3D. Natomiast zasadniczą funkcją programu jest wygenerowanie pokrycia terenu przez radar. Aplikacja działa w ściśle określonym terenie pomiędzy współrzędnymi geograficznymi od 39,5N do 40N i od 105,1W do 105,6W. Jest to teren w centralnej części Stanów Zjednoczonych gdzie zaczyna się pasmo górskie – obszar idealny do zademonstrowania

działania programu. Poniżej (rysunek 17) przedstawiono interfejs aplikacji oraz przykładowy sposób działania na mapie 3D (rysunek 18). W niniejszym przykładzie obliczenia przeprowadzono dla przypadku w którym statek powietrzny poruszałby się na wysokości 2500 m.n.p.m.. Żółtym kolorem oznaczono przeszkody terenowe, natomiast czerwonym rejon w którym istnieje ryzyko wykrycia przez radar, pozostały obszar zakłada się, że jest bezpieczny i tamtędy powinien przemieszczać się statek powietrzny na zadanej wysokości lub mniejszej pod warunkiem, że teren na to pozwala.

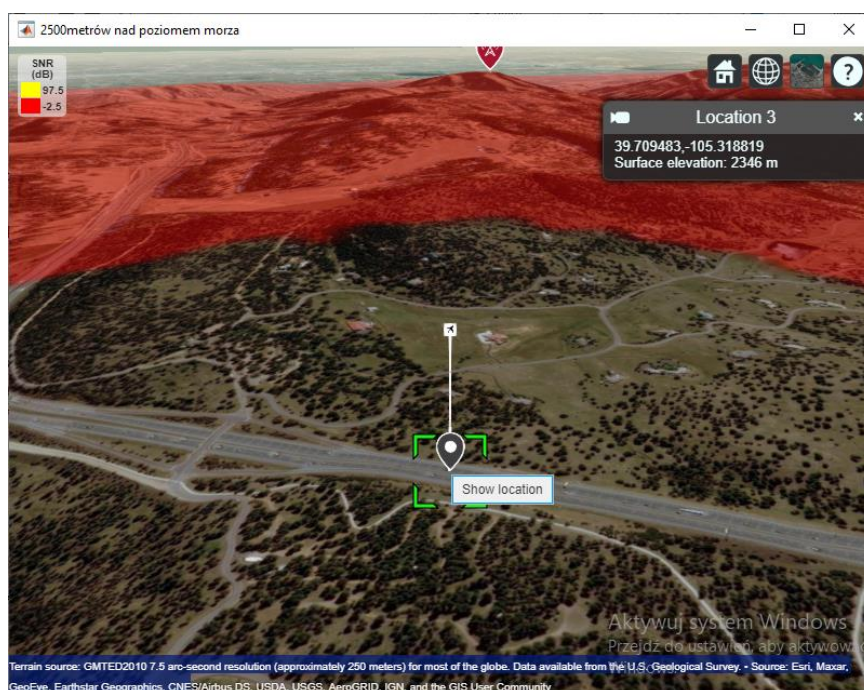
Rysunek 17 Interfejs aplikacji wspomagającej planowanie misji



Rysunek 18 Pokrycie obszaru wiązką radarową dla 2500 m.n.p.m

Po wygenerowaniu pokrycia radarowego można wybrać punkt na trasie pamiętając, że obliczenia zostały wykonane dla jednej wybranej wysokości, lub wykonać obliczenia dla innej wysokości w celu znalezienia optymalnej trasy. Optymalny może się okazać lot na różnych wysokościach, co wymaga wykonanie wielokrotnych obliczeń dla różnych wysokości. Dalej zostanie przedstawiony przykład takiego działania.

Aby dodać punkt na trasie należy prawym przyciskiem myszy kliknąć w pożądane miejsce i wybrać opcję „show location” (rysunek 18). Następnie należy skopiować współrzędne geograficzne punktu do okna aplikacji, wpisać względną wysokość nad terenem i nacisnąć dolny przycisk „zatwierdź”. Pojawi się wtedy ikona samolotu obrazująca również wysokość na jakiej statek powietrzny będzie się znajdował. Na następnych stronach zaprezentowana zostanie przykład wyboru optymalnej trasy.



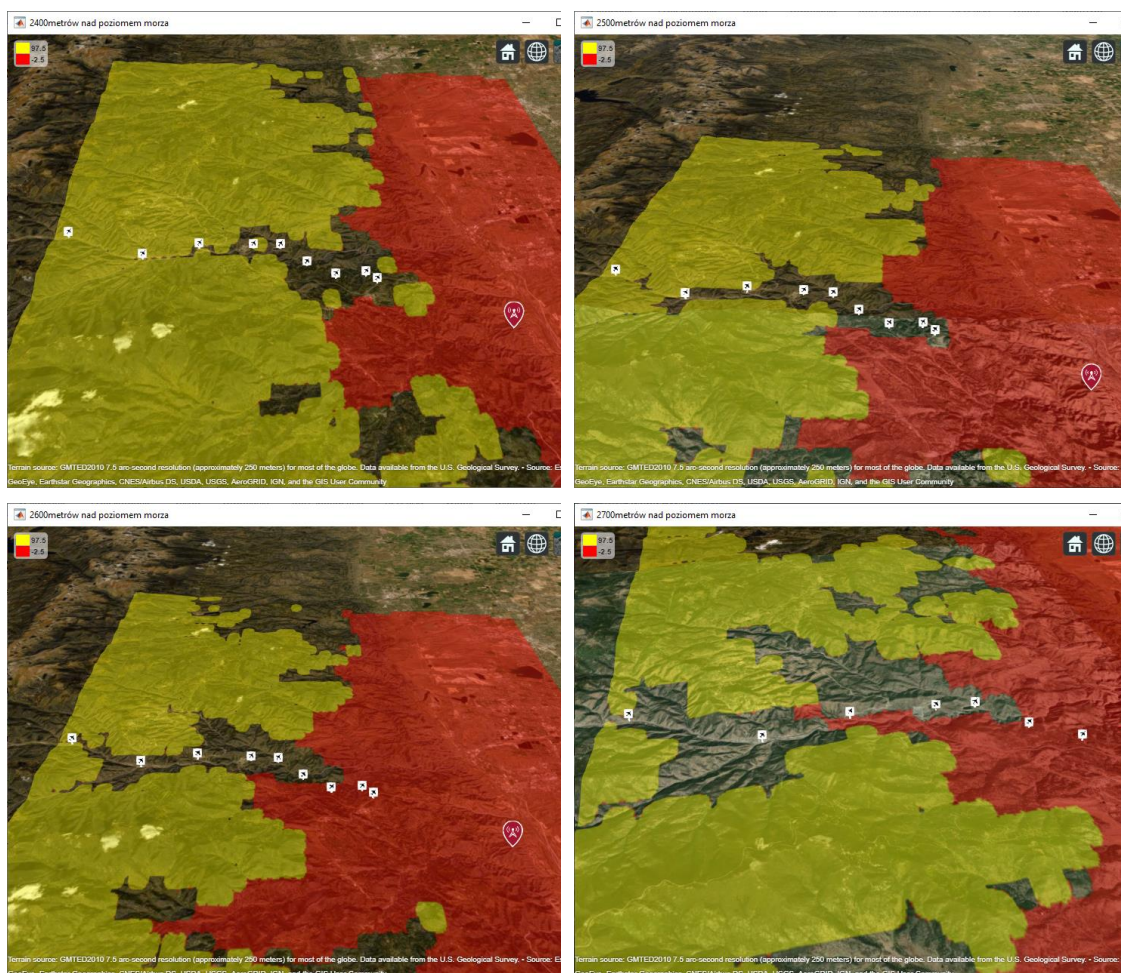
Rysunek 19 Dodanie punktu na trasie

Obliczenia wykonano po kolei dla wysokości 2400, 2500, 2600 i 2700 m n.p.m. Wraz ze wzrostem wysokości można zauważyć, że zwiększa się obszar oddziaływania radaru oraz zmniejsza obszar występowania niebezpiecznych przeszkód terenowych. Pierwsze trzy punkty licząc od prawej strony zostały wybrane dla wysokości 2500 m, gdyż była ona bardziej optymalna niż 2400 m, ryzyko wykrycia przez radar było praktycznie takie samo natomiast ryzyko zderzenia z górami zmniejszało się. Następne pięć punktów dobrano dla wysokości 2600 m, natomiast ostatni dla 2700 m. Pokrycie radarowe dla

wymienionych wysokości pokazano na rysunku 21, w lewym górnym rogu dla 2400 m, w prawym górnym dla 2500 m, w dolnym lewym 2600 m i 2700 m w prawym dolnym. Na rysunku 20 przedstawiono tabelę skopiowaną z okna aplikacji, a pokazującą zestawienie wybranych punktów. Aby uniknąć wykrycia statek powietrzny powinien poruszać się pomiędzy wyznaczonymi punktami.

Trasa lotu			
Lp.	Szer. geogr.	Długość geogr.	Wys. nad terenem
1.0000	39.7095	-105.3188	154.0000
2.0000	39.7159	-105.3272	150.0000
3.0000	39.7159	-105.3526	150.0000
4.0000	39.7275	-105.3760	100.0000
5.0000	39.7458	-105.3982	100.0000
6.0000	39.7446	-105.4218	100.0000
7.0000	39.7481	-105.4693	200.0000
8.0000	39.7407	-105.5172	220.0000
9.0000	39.7639	-105.5866	200.0000

Rysunek 20 Tabela z punktami trasy



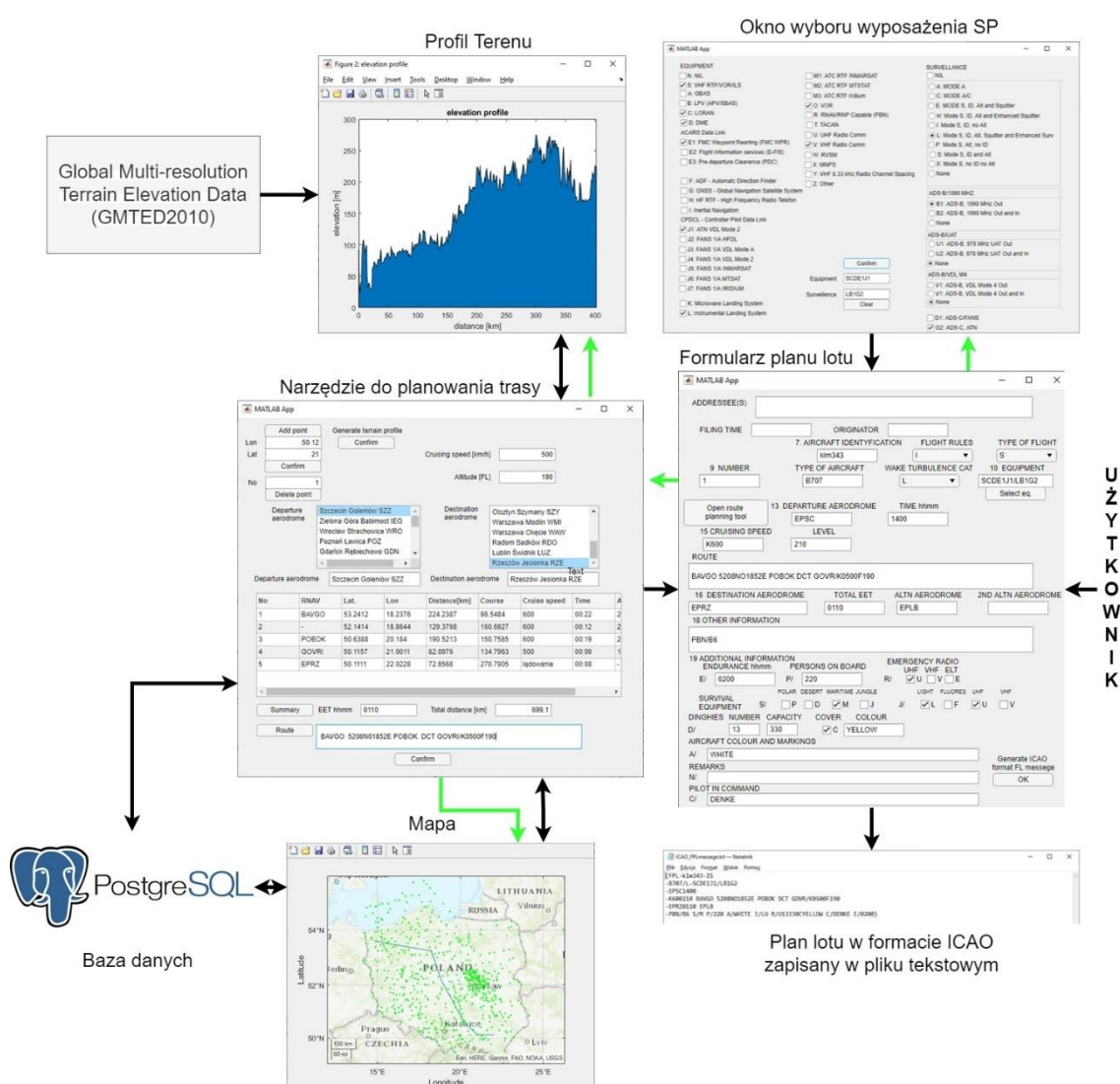
Rysunek 21 Efekt działania aplikacji dla różnych wysokości przelotu

5 Algorytmy aplikacji planowania lotu i planowania misji

W niniejszym rozdziale opisano jak zrealizowano obydwie aplikacje, jakie napotkano problemy techniczne i ograniczenia, a także szczegółowo omówiono działanie oprogramowania.

5.1 Algorytmizacja aplikacji do planowania lotu

Całość aplikacji składa się z wielu współdziałających ze sobą elementów co przedstawiono na diagramie (rysunek 21).



Rysunek 22 Diagram pokazujący wszystkie elementy aplikacji i przepływ danych pomiędzy nimi

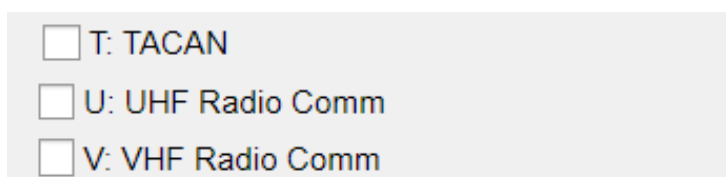
Czarne strzałki pokazują przepływ danych pomiędzy poszczególnymi elementami, natomiast zielone pokazują z poziomu, którego narzędzia otwierane są poszczególne okna (np. okno wyboru wyposażenia jest otwierane z poziomu formularza planu lotu).

Elementem spajającym całość jest formularz planu lotu. Do niego spływają ostatecznie wszystkie dane z innych okien oraz bezpośrednio od użytkownika - nie wszystkie informacje można wprowadzić automatycznie i część pól formularza musi być uzupełnionych ręcznie. Każde pole może być również zmieniane przez użytkownika aplikacji.

5.1.1 Okno wyboru wyposażenia

Każdemu z pól okna (rys. 10) przypisano jedną zmienną typu string oraz każde pole odpowiada określonemu rodzajowi sprzętu jaki może znaleźć się na statku powietrznym. Wszystkie te zmienne po wciśnięciu przycisku „Confirm” są sumowane i tworzą dwa ciągi znaków mówiących o tym jak wyposażony jest statek powietrzny. Jeden ciąg znaków wpisywany jest w polu „Equipment” i odpowiada wyposażeniu radiokomunikacyjnemu oraz radionawigacyjnemu, natomiast drugi ciąg w pole „Surveillance” odpowiada za sprzęt do dozoru przestrzeni powietrznej. Obydwie grupy są wyraźnie oddzielone w interfejsie graficznym.

Jeżeli dane pole jest zaznaczone to zmiennej przypisuje się wartość pojedynczej litery, jeżeli nie, to wartość zmiennej jest pusta. Wstępnie w ustawieniach aplikacji wszystkie zmienne zadeklarowane są z nieprzypisaną wartością np.: eqT = „”.



☐ T: TACAN
☐ U: UHF Radio Comm
☐ V: VHF Radio Comm

Rysunek 23 Pola wyboru danego typu wyposażenia

```
% Value changed function: TTACANCheckBox
function TTACANCheckBoxValueChanged(app, event)
    if app.TTACANCheckBox.Value
        app.eqT="T";
    else
        app.eqT="";
    end
end
```

Rysunek 24 Obsługa pola wyboru

Pola „S: VHF RTF/VOR/ILS”, „V: VHF Radio Comm O”, „VOR, „L: Instrumental Landing System” są ze sobą powiązane. Zaznaczenie pola S jest równoznaczne zaznaczeniu pól V, O i L, zatem w obu przypadkach w pole „Equipment” wpisana zostanie litera „S”. Co wymagało dodatkowych linii kodu przy każdym z czterech pól.

Na rysunku 24 opisano działanie przycisku Confirm spajającego cały program. Dla obydwóch pól tekstowych czyli pola „Equipment” i pola „Surveillance” sposób działania jest taki sam. Wszystkie zmienne przypisane polom wyboru (tzw. CheckBox) są po wciśnięciu przycisku sumowane i wyświetlane w polach tekstowych. Jeżeli żadne pola nie zostały zaznaczone lub jeżeli zaznaczono pole „NIL” wyświetlana jest litera „N”. Na samym końcu obydwie ciągi znaków są sumowane, oddzielane ukośnikiem i przesyłane do formularza planu lotu. Oprogramowanie interfejsu okna jest rozległe ze względu na dużą ilość elementów jednak stosunkowo proste i oparte na instrukcjach warunkowych. Całość, wraz z kodem wygenerowanym automatycznie przez środowisko składa się z ponad 1000 linijek kodu.

```

228 %WYPOSAŻENIE RADIOKOMUNIKACYJNE I RADIONAWIGACYJNE
229 app.EquipmentEditField.Value = ... %podstawienie danych do pola Equipment
230 app.eqN+app.eqS+app.eqA+app.eqB+app.eqC+app.eqD+app.eqE1+...
231 app.eqE2+app.eqE3+app.eqF+app.eqG+app.eqH+app.eqI+app.eqJ1+app.eqJ2+app.eqJ3+...
232 app.eqJ4+app.eqJ5+app.eqJ6+app.eqJ7+app.eqK+app.eqL+app.eqM1+ app.eqM2+app.eqM3+...
233 app.eqO+app.eqR+app.eqT+app.eqU+app.eqV+ app.eqW+app.eqX+app.eqY+app.eqZ;
234 if isempty(app.EquipmentEditField.Value) %Przypadek gdy nie wprowadzono żadnych danych
235 app.EquipmentEditField.Value="N"; %domyślną wartością jest "N" czyli brak sprzętu
236 end
237 if app.EqNILCheckBox.Value %Przypadek gdy oznaczone jest pole NIL
238 app.EquipmentEditField.Value="N";
239 end
240 %SPRZĘT DOZOROWANIA PRZESTRZENI POWIETRZNEJ
241 app.SurveillanceEditField.Value = ... %podstawienie danych do pola Surveillance
242 app.survN+app.survA+app.survC+app.survE+...
243 app.survH+app.survI+app.survL+app.survP+app.survS+app.survX+app.survB1+...
244 app.survB2+app.survU1+app.survU2+app.survV1+app.survV2+app.survD1+app.survG2;
245 if isempty(app.SurveillanceEditField.Value) %Przypadek gdy nie wprowadzono żadnych danych
246 app.SurveillanceEditField.Value="N"; %domyślną wartością jest "N" czyli brak sprzętu
247 end
248 if app.SurvNILCheckBox.Value %Przypadek gdy oznaczone jest pole NIL
249 app.SurveillanceEditField.Value="N";
250 end
251 EquipmentAndSurveillance = app.EquipmentEditField.Value+"/"+app.SurveillanceEditField.Value;
252 passData(app.CallingApp,EquipmentAndSurveillance); %przesłanie danych do formularza
253 end

```

Rysunek 25 Obsługa przycisku "Confirm"

5.1.2 Narzędzie do planowania trasy

Narzędzie to jest najbardziej zaawansowanym fragmentem aplikacji. Ta część projektu współpracuje z bazą danych „PostgreSQL” w której zapisano nazwy i współrzędne pomocy nawigacyjnych. Po pobraniu tych danych z bazy danych możliwe jest ich wyświetlenie na mapie Polski. Zastosowanie bazy danych pozwala na łatwiejsze przechowywanie informacji, a co najważniejsze pozwala łatwo sortować i wybierać informacje według pożądaných kryteriów, co było niezbędne aby wybrać pomoc nawigacyjną z mapy, co zostanie omówione podczas omawiania kodu. Kolejną zaawansowaną rzeczą jest możliwość współdziałania aplikacji z danymi typu GMTED2010, co umożliwiło wygenerowanie wykresu elewacji terenu.

Pierwszą częścią programu są ustawienia czyli „properties” (rysunek 26). Są to zmienne do których można się dostać z poziomu każdej funkcji. Część ma przypisaną wartość już w momencie deklaracji zmiennej, wartości innych są przypisywane dopiero w funkcji *startupFcn* która jest wykonywana przy starcie aplikacji lub w trakcie działania programu (rysunek 27).

```

41 properties (Access = public)
42 - lat=0; %do tej zmiennej przypisana jest wartosc z pola edycyjnego "Lat"
43 - lon=0; %do tej zmiennej przypisana jest wartosc z pola edycyjnego "Lon"
44 - gx; %później zadeklarowany jako obiekt klasy geoxaxes czyli mapa
45 - tablica_pomocnicza= [];
46 - Lp = 0;
47 %wstępna deklaracja wektora przechowującego
48 %współrzędne geograficzne pomocy nawigacyjnych
49 - rnav_lats =0;
50 - rnav_lons=0;
51 %stworzenie połączenia z bazą danych
52 - conn = database('PostgreSQL30','postgres','z');
53 %cyfrowe dane o elewacji terenu Z - wysokości poszczególnych komorek R - opcje dotyczące pliku
54 - Z=[];
55 - R;
56 %kolumny tabeli, pozwoli łatwo zmieniać kolejność kolumn i odczytywać dane
57 - Lpkolumna = 1;
58 - RNAVKolumna = 2;
59 - LatKolumna = 3;
60 - LonKolumna = 4;
61 - OdlegloscKolumna = 5;
62 - KursKolumna = 6;
63 - PredkosczPrzelotowaKolumna =7;
64 - CzasPrzelotuKolumna =8; %liczony między punktami
65 - WysokoscKolumna =9; %opoznienie przy danym miejscu
66 %ostatni wiersz znacznie różni się od poprzednich
67 - wiersz_docelowy = [ " " " " " " " " " " " " " " " " " "-"];
68 %struktura danych przechowująca dane lotniskach
69 - lotniskoStartowe = struct('kodicao','----','nazwalotniska','-','lat',50,'lon',50);
70 - lotniskoDocelowe = struct('kodicao','----','nazwalotniska','-','lat',50,'lon',50);
71 %Przekazywanie danych pomiędzy aplikacjami
72 CallingApp;
73 - end

```

Rysunek 26 Ustawienia - properties

W funkcji *startupFcn* otwierane jest okno z interaktywną mapą. Jest ona obiektem o nazwie *gx*, co w dalszej części działania programu pozwala wprowadzać zmiany graficzne na mapie. Od linii 83 do 94 pobierane są dane o współrzędnych geograficznych punktów nawigacyjnych a następnie zostają one narysowane na mapie w postaci zielonych punktów. W linii 97 przetwarzane są dane o elewacji terenu z pliku o rozszerzeniu *tiff* i zapisywane są jako zmienne *app.Z* i *app.R*. Dalej wybierane jest domyślne lotnisko startowe, a ostatnia linia jest niezbędna aby umożliwić połączenie z oknem formularza.

```

79 % Code that executes after component creation
80 function startupFcn(app, formularz, DepartureAerodrome, Route, CruisingSpeed, FlightLevel, DestinationAerodrome)
81 app.gx = geoaxes('Basemap','topographic'); %Otworzenie okna z mapą Polski
82
83 query = "select (lat) from pomoce_r"; %Zapytanie do bazy danych
84 app.rnav_lats = select(app.conn,query); %Pobranie danych o szerokościach geog.
85 app.rnav_lats = table2array(app.rnav_lats); %Zmiana typu danych
86 query = "select (lon) from pomoce_r"; %Te same czynności lecz dla długości geog.
87 app.rnav_lons = select(app.conn,query);
88 app.rnav_lons = table2array(app.rnav_lons);
89 %Wyświetlenie pomocy nawigacyjnych na mapie
90 hold on;
91 for i=1:length(app.rnav_lats)
92     geoplots(app.gx,app.rnav_lats(i),app.rnav_lons(i),'Marker','.', 'color','green');
93 end
94 geobasemap(app.gx,'topographic');
95
96 %--cyfrowe dane o elewacji terenu Z - wysokości poszczególnych komerek R - opcje dotyczące pliku
97 [app.Z,app.R] = readgeoraster('D:\Praca_Magisterska_robota\Aplikacja_Plan_Lotu2\50n000e_20101117_gated_bin300.tif','OutputType','double');
98
99 %--lotnisko startowe wybrane domyślnie
100 query="select * from lotniska where nazwalotniska='"+app.LotniskaStartoweListBox.Value+"'";
101 app.lotniskoStartowe = select(app.conn,query);
102
103 query="select * from lotniska where nazwalotniska='"+app.LotniskaDoceloweListBox.Value+"'";
104 app.lotniskoDocelowe = select(app.conn,query);
105
106 app.LotniskoStartoweEditField.Value = string(app.lotniskoStartowe.nazwalotniska);
107 app.LotniskoDoceloweEditField.Value = string(app.lotniskoDocelowe.nazwalotniska);
108 %Przekazywanie danych pomiędzy aplikacjami
109 app.CallingApp = formularz;
110 end

```

Rysunek 27 Funkcja startowa – *startupFcn*

Następną funkcją jest funkcja wywołana wciśnięciem przycisku „Add point”. Funkcja *ginput()* powoduje pojawienie się na mapie krzyża (patrz rysunek 12) i dalej po kliknięciu pobiera współrzędne punktu które są przetwarzane po wciśnięciu przycisku „Confirm”.

```

112 % Button pushed function: AddpointButton
113 function AddpointButtonPushed(app, event)
114 [app.lat, app.lon] = ginput(1); %obsługa wybrania punktu z mapy, zwraca współrzędne geograficzne punktu
115 app.LonEditField.Value = app.lat;
116 app.LatEditField.Value = app.lon;
117 end

```

Rysunek 28 Obsługa przycisku "Add point"

Najwięcej operacji jest wykonywanych po wciśnięciu przycisku „Confirm” (rysunek 29, 30) umieszczonym pod polami „Lat” i „Lon”. W pierwszej kolejności sprawdzane jest czy podano prędkość przelotową, jeżeli nie to wyświetli się komunikat przypominający o tym i dalsza część skryptu nie zostanie wykonana. Następnie przypisywana jest liczba porządkowa wybranemu punktowi, odległość od poprzedniego wybranego punktu lub od lotniska startowego. Między linią 139, a 173 sprawdzane jest

czy wybrany punkt jest położony w pobliżu punktu nawigacyjnego. W tym celu zostaje wysłane zapytanie do bazy danych czy w przedziale współrzędnych którego wielkość jest określona przez zmienną *zakres*, znajduje się pomoc nawigacyjna. Jeżeli nie, to w kolumnie RNAV zostanie wpisany „-”, jeśli tak to zostanie wpisana jego nazwa. Ta część skryptu wymagała rozbudowy ze względu na to, że w danym zakresie mogło znajdować się kilka punktów. Skrypt po kolei sprawdza odległości między punktem o współrzędnych pobranych z mapy, a pobliskimi pomocami nawigacyjnymi i ostatecznie wybiera tą najbliższą.

W kolejnych wierszach obliczane są: kurs jakim porusza się statek powietrzny, odległość i czas przelotu między punktami oraz wprowadzane są pozostałe dane do tabeli. Zapisywane są one w zmiennej *nowy_wiersz*, która jest dodawana do zmiennej *tablica_pomocnicza*, następnie dane z tablicy pomocniczej zapisywane są w obiekcie *UITable*. Takie same obliczenia są wykonywane osobno dla ostatniego wiersza tabeli i również dodawane są do obiektu *UITable* w wierszu 229. Na samym końcu napisany został skrypt kreślący trasę na mapie

```

120 function ZatwierdzDodaniePunktuButtonPushed(app, event)
121 if app.PredkoscPrzelotowaEditField.Value ==0
122 komunikat_predkoscprzelotowa;
123 else
124 %liczba porządkowa, nr na trasie
125 app.Lp = app.Lp + 1;
126 %----obliczanie dystansu od poprzedniego punktu -----
127 if app.Lp ==1
128 odleglosc = distance(app.lotniskoStartowe.lat,app.lotniskoStartowe.lon,app.lat, app.lon);
129 odleglosc = deg2km(odleglosc);
130 else
131 odleglosc = distance([str2double(app.tablica_pomocnicza((app.Lp-1),3)),...
132 str2double( app.tablica_pomocnicza((app.Lp-1),4))],[app.lat, app.lon]);
133 odleglosc = deg2km(odleglosc);
134 end
135 %---dodanie lub nie, nazwy znaczącego punktu radionawigacyjnego
136 % kod wybiera punkt umieszczony najbliżej wybranych współrzędnych
137 %pomoc nawigacyjna zostanie dodana jeżeli będzie się znajdowała w odpowiednim
138 % zakresie współrzędnych, czyli dostatecznie blisko wybranego z mapy miejsca
139 zakres = 0.1;
140
141 lat_zakres_dolny = app.lat - zakres;
142 lat_zakres_gorny = app.lat + zakres;
143
144 lon_zakres_dolny = app.lon - zakres;
145 lon_zakres_gorny = app.lon + zakres;
146 %pobranie z bazy danych punktów które znajdują się w pobliżu
147 query = "select * from pomoce_r where lat >="+lat_zakres_dolny+" AND lat <=" + lat_zakres_gorny+ ...
148 " AND lon>="+lon_zakres_dolny+" AND lon <= "+lon_zakres_gorny;
149 data = select(app.conn,query);
150 %Obsługa kliknięcia myszką na mapę. Kliknięcie lewym przyciskiem myszy w odpowiedniej
151 %odleglosci od pomocy radionaw. spowoduje uwzględnienie jej w tabelce
152 stop = size(data); %pętla for wykona się tyle razy ile w pobliżu jest punktów sprawdzając który punkt jest najbliżej
153 if stop(1)>1
154 najblizej = distance(app.LatEditField.Value, data{1,3},data{1,2}, app.LonEditField.Value);

```

Rysunek 29 Skrypt przycisku "Confirm

```

CODE BRIEF APP LAYOUT
156 - punkt = i;
157 - for i=1:i:stop(1)
158 -     najblizej2 = distance(app.LatEditField.Value, data{i,3},data{i,2}, app.LongEditField.Value);
159 -     if najblizej2 < najblizej
160 -         najblizej = najblizej2;
161 -         punkt = i;
162 -     end
163 - end
164 - RNAV = data[punkt,1];
165 - elseif stop(1)==1
166 -     RNAV = data{1,1};
167 - elseif stop(1)==0
168 -     RNAV = "-";
169 - end
170 - RNAV=[RNAV{:}];
171 - RNAV=string(RNAV);
172 - %---koniec obsługi kliknięcia na mapę
173 -
174 - %wyznaczenie kursu
175 - if app.Lp ==1
176 -     kurs = azimuth(app.lotniskoStartowe.lat, ...
177 -                    app.lotniskoStartowe.lon, ...
178 -                    app.lat, ...
179 -                    app.lon );
180 - else
181 -     kurs = azimuth(str2double(app.UITable.Data(app.Lp-1,app.LatKolumna)), ...
182 -                   str2double(app.UITable.Data(app.Lp-1,app.LonKolumna)), ...
183 -                   app.lat, ...
184 -                   app.lon );
185 - end
186 - %---koniec wyznaczenia kursu
187 -
188 - % ---dodawanie nowych wsp geograficznych
189 - nowy_wiersz = [ " " " " " " " " " " " " " " " " " " " " " " " " ];
190 -
GO APP LAYOUT
191 - nowy_wiersz(1,app.LpKolumna)= num2str(app.Lp);
192 - nowy_wiersz(1,app.RNAVKolumna)=RNAV;
193 - nowy_wiersz(1,app.LatKolumna)=num2str(app.lat);
194 - nowy_wiersz(1,app.LonKolumna)=num2str(app.lon);
195 - nowy_wiersz(1,app.OdlęgloścKolumna)=num2str(odleglosc); %policzony powyżej
196 - nowy_wiersz(1,app.KursKolumna)=num2str(kurs); %policzony powyżej
197 - nowy_wiersz(1,app.PredkoscPrzelotowaKolumna)=num2str(app.PredkoscPrzelotowaEditField.Value);
198 - nowy_wiersz(1,app.WysokoscKolumna) = num2str(app.WysokoscPrzelotowaEditField.Value);
199 -
200 - if app.Lp == 1
201 -     czas = datestr(hours(odleglosc/app.PredkoscPrzelotowaEditField.Value),'HH:mm');
202 - else
203 -     czas = datestr(hours(odleglosc/app.PredkoscPrzelotowaEditField.Value),'HH:mm');
204 - end
205 - nowy_wiersz(1,app.CzasPrzelotuKolumna)=num2str(czas);
206 -
207 - app.tablica_pomocnicza = [app.tablica_pomocnicza; nowy_wiersz];
208 -
209 - %%--wiersz z lotniskiem docelowym (ostatni wiersz tabeli)
210 - app.wiersz_docelowy(app.LpKolumna) = string(app.Lp+1);
211 - app.wiersz_docelowy(app.RNAVKolumna) = string(app.lotniskoDocelowe.kodicao);
212 - app.wiersz_docelowy(app.LatKolumna) = app.lotniskoDocelowe.lat;
213 - app.wiersz_docelowy(app.LonKolumna) = app.lotniskoDocelowe.lon;
214 -
215 - odleglosc = distance([str2double(app.tablica_pomocnicza((app.Lp),app.LatKolumna)),...
216 -                     str2double( app.tablica_pomocnicza((app.Lp),app.LonKolumna))],...
217 -                     [app.lotniskoDocelowe.lat, app.lotniskoDocelowe.lon]);
218 - odleglosc = deg2km(odleglosc);
219 - app.wiersz_docelowy(app.OdlęgloścKolumna) = odleglosc;
220 - app.wiersz_docelowy(app.KursKolumna) = azimuth(app.lotniskoDocelowe.lat, app.lotniskoDocelowe.lon,...
221 -                                                str2double(app.tablica_pomocnicza(app.Lp,app.LatKolumna)),...
222 -                                                str2double(app.tablica_pomocnicza(app.Lp,app.LonKolumna)));
223 - app.wiersz_docelowy(app.PredkoscPrzelotowaKolumna) = "ladowanie";
224 -
225 -
226 - czas = datestr(hours(odleglosc/app.PredkoscPrzelotowaEditField.Value),'HH:mm');
227 - app.wiersz_docelowy(app.CzasPrzelotuKolumna)= num2str(czas);
228 -
229 - app.UITable.Data = [app.tablica_pomocnicza; app.wiersz_docelowy];
230 - %narozowanie trasy na mapie
231 - hold off;
232 - geoplots(app.gx, ...
233 -          [app.lotniskoStartowe.lat; str2double(app.UITable.Data(:,app.LatKolumna))],...
234 -          [app.lotniskoStartowe.lon; str2double(app.UITable.Data(:,app.LonKolumna))] ...
235 -          );
236 - hold on;
237 - for i=1:length(app.rnav_lats)
238 -     geoplots(app.gx,app.rnav_lats(i),app.rnav_lons(i),'Marker','.', 'Color','green');
239 - end
240 - geobasemap(app.gx,"topographic");
241 - end %if
242 -
243 - end

```

Rysunek 30 Skrypt przycisku "Confirm cd."

```

245 % Button pushed function: UsunPunktButton
246 function UsunPunktButtonPushed(app, event)
247     wiersz_usun = app.LpPunktDoUsunieciaEditField.Value;
248     app.tablica_pomocnicza(wiersz_usun,:) = [];
249
250     stop = size(app.tablica_pomocnicza);
251     for i=1:stop(1)
252         app.tablica_pomocnicza(i,1) = num2str(i);
253     end
254
255     S = size(app.tablica_pomocnicza);
256     app.Lp = S(1);
257     app.UITable.Data = app.tablica_pomocnicza;
258
259     hold off;
260     geoplots(app.gx,[str2double(app.tablica_pomocnicza(:,3))], [str2double(app.tablica_pomocnicza(:,4))]);
261     hold on;
262     for i=1:length(app.rnav_lats)
263         geoplots(app.gx,app.rnav_lats(i),app.rnav_lons(i),'Marker','.', 'Color','green');
264     end
265
266     geobasemap(app.gx,"topographic");
267 end

```

Rysunek 31 Kod odpowiedzialny za usunięcie punktu

Na rysunku 32 widoczny jest skrypt odpowiedzialny za usunięcie niepożądanego punktu na trasie. Pierwszym krokiem jest usunięcie danego wiersza z tablicy pomocniczej, następnie nadaje się nowe numery porządkowe, zaktualizowane dane z tablicy pomocniczej ponownie są wczytywane do obiektu *UITable*. Na końcu aktualizowana jest trasa lotu na mapie.

```

269 % Button pushed function: ProfilTerenuZatwierdzButton
270 function ProfilTerenuZatwierdzButtonPushed(app, event)
271     [Zi,Ri,~,~] = mapprofile(app.Z,app.R,str2double(app.tablica_pomocnicza(:,3)),str2double(app.tablica_pomocnicza(:,4)));
272     %osi=nexttile;
273     Ri=deg2km(Ri);
274
275     figure('Name','elevation profile');
276     plot(Ri,Zi);
277     area(Ri,Zi);
278     xlim([0 Ri(length(Ri))]);
279     title('elevation profile');
280     xlabel('distance [km]');
281     ylabel('elevation [m]');
282 end

```

Rysunek 32 Generowanie wykresu elewacji terenu

Funkcja *mapprofile* jest odpowiedzialna za wykreślenie profilu terenu pomiędzy punktami podanymi jako argument tej funkcji. Jako argumenty funkcji podawane są także dane o elewacji terenu pobrane z pliku [50n000e_20101117_gmted_bln300.tif](#). Plik zawiera dane o elewacji terenu dla całego obszaru Polski na północ od równoleżnika 50N. Oznacza to, że nie możliwe jest wygenerowanie wykresu elewacji dla obszaru na południe od Krakowa i Rzeszowa, co jest ograniczeniem aplikacji wynikające z braku dostępu do innych map DTED. Rozdzielczość cyfrowych danych to 30 sekund geograficznych (kwadraty o szerokości około 900 metrów). Funkcja zwraca wektor wysokości *Zi* i wektor odpowiadający pokonanej odległości *Ri*, na podstawie danych pobranych z pliku. Kolejne linijki kodu to standardowe funkcje do tworzenia wykresów.

```

284 % Value changed function: LotniskaStartoweListBox
285 function LotniskaStartoweListBoxValueChanged(app, event)
286 - query="select * from lotniska where nazwalotniska = '"+app.LotniskaStartoweListBox.Value+"'";
287 - app.lotniskoStartowe = select(app.conn,query);
288 - app.LotniskoStartoweEditField.Value = string(app.lotniskoStartowe.nazwalotniska);
289 - end
290
291 % Value changed function: LotniskaDoceloweListBox
292 function LotniskaDoceloweListBoxValueChanged(app, event)
293 - query="select * from lotniska where nazwalotniska = '"+app.LotniskaDoceloweListBox.Value+"'";
294 - app.lotniskoDocelowe = select(app.conn,query);
295 - app.LotniskoDoceloweEditField.Value = string(app.lotniskoDocelowe.nazwalotniska);
296 - end
297

```

Rysunek 33 Sposób wyboru lotniska startowego i docelowego

Przy wyborze lotniska z jednej z list rozwijanych (rysunek 33), zostaje wysłane zapytanie do bazy danych z zapytaniem o informacje odnoszące się do niego. Pobrane z bazy danych informacje zostają zapisane w zmiennych *app.lotniskoStartowe* i *app.lotniskoDocelowe*, które są strukturami danych.

```

331 % Button pushed function: SummaryButton
332 function SummaryButtonPushed(app, event)
333 - iloscWierszy = size(app.UITable.Data);
334 - sum = 0;
335 - for i=1:1:(iloscWierszy(1))
336 - czas = char(app.UITable.Data(i,app.CzasPrzelotuKolumna));
337 - minuty = string(czas(4))+string(czas(5));
338 - minuty = str2double(minuty);
339 - godziny = string(czas(1))+string(czas(2));
340 - godziny=str2double(godziny);
341 - czas2 = godziny*60 + minuty;
342 - czas2 = czas2/60;
343 - sum = sum + czas2;
344 - end
345 - app.EETHmmEditField.Value=datestr(hours(sum),'HHMM');
346 - sum=0;
347 - for i=1:1:(iloscWierszy(1))
348 - dystans = str2double(app.UITable.Data(i,app.OdlegloscKolumna));
349 - sum = sum + dystans;
350 - end
351 - app.TotaldistancekmEditField.Value = sum;
352 - end

```

Rysunek 34 Obliczanie długości lotu

Kod programu odpowiadający za obliczenie czasu trwania lotu oraz pokonanego dystansu znajduje się na rysunku 34. Jest to kod obsługujący przycisk „Summary”. Dane pobierane są z tabeli.


```

299     function RouteButtonPushed(app, event)
300     Trasa = "";
301     koniec = size(app.UITable.Data);
302     for i = 1:1:koniec(1)-1
303         if app.UITable.Data(i,app.RNAVKolumna) == "-"
304             punktTrasy = punktGeograficzny(str2double(app.UITable.Data(i,app.LatKolumna)),str2double(app.UITable.Data(i,app.LonKolumna)));
305         else
306             punktTrasy = app.UITable.Data(i,app.RNAVKolumna);
307         end
308
309         if i>1
310             if app.UITable.Data(i,app.WysokoscKolumna) ~= app.UITable.Data(i-1,app.WysokoscKolumna)... %warunek
311                 || app.UITable.Data(i,app.PredkoscPrzelotowaKolumna) ~= app.UITable.Data(i-1,app.PredkoscPrzelotowaKolumna):
312
313                 punktTrasy = zmianaParametrowLotu(app.PredkoscPrzelotowaEditField.Value,app.WysokoscPrzelotowaEditField.Value,punktTrasy);
314             end
315         end
316
317         if i>1 && app.UITable.Data(i-1,app.RNAVKolumna) ~= "-" && app.UITable.Data(i,app.RNAVKolumna) ~= "-"
318             Trasa = Trasa + " DCT " + punktTrasy;
319         elseif i == koniec(1)
320             else
321                 Trasa = Trasa + " " + punktTrasy;
322             end
323
324     end
325     app.TrasaEditField.Value = Trasa;
326     end

```

Rysunek 35 Kod odpowiedzialny za wygenerowanie trasy zgodnie z wymaganiami ICAO

Na rysunku 35 widnieje kod generujący opis trasy lotu, który jest wpisywany w punkcie 15 planu lotu. Skrypt wykorzystuje samodzielnie napisane funkcje *punktGeograficzny* i *zmianaParametrowLotu*. Pierwsza funkcja pozwala n podstawie znanych współrzędnych geograficznych opisać punkt według norm ICAO np. 5208NO1852E. W tym celu konwertuje współrzędne geograficzne reprezentowane w formacie cyfrowym na współrzędne określone w tradycyjny sposób za pomocą stopni i minut, a następnie łączy odpowiedni ciąg znaków. Druga wymieniona funkcja pokazuje, że parametry lotu uległy zmianie, efektem działania jest np. GOVR/K0500F190. W połączeniu z odpowiednimi instrukcjami warunkowymi i danymi dostępnymi w tabeli, opisywany skrypt generuje całą trasę lotu np.: K600210 BAVGO 5208NO1852E POBOK DCT GOVR/K0500F190. Ostatnią funkcją jest przesłanie danych do formularza planu lotu. Operacja jest wykonywana po wciśnięciu przycisku na samym dole interfejsu (rysunek 36).

```

351     % Button pushed function: ConfirmButton
352     function ConfirmButtonPushed(app, event)
353         %Przesłanie danych do formularza
354         routePlanning(app.CallingApp,app.TrasaEditField.Value, ...
355             string(app.lotniskoStartowe.kodicao), ...
356             string("K"+app.UITable.Data(1,app.PredkoscPrzelotowaKolumna)), ...
357             string(app.UITable.Data(1,app.WysokoscKolumna)), ...
358             app.EETHmmEditField.Value, ...
359             string(app.lotniskoDocelowe.kodicao));
360     end

```

Rysunek 36 Przesłanie danych do formularza

5.1.3 Okno Formularza

Każda możliwa informacja jaką zawiera formularz jest przechowywana jako pole *Value* obiektu graficznego w którym została ta informacja podana. W przypadku informacji takich jak np. informacja o kamizelkach ratunkowych, została stworzona dodatkowa zmienna *app.LifeJackets* która łączy informacje z pól wyboru L, F, U i V. Jeśli pola nie są zaznaczone zmiennej przypisana jest pusta wartość natomiast gdy przynajmniej jedno z pól wyboru jest zaznaczone zmienna ta przyjmuje wartość np. „S/LU”. Pola wyboru są zaprogramowane (rysunek 37) tak samo jak te z okna do wyboru wyposażenia. Takie samo rozwiązanie zastosowano w przypadku radiowych urządzeń ratowniczych i wyposażenia do przetrwania.

```
% Value changed function: LifeJacketsLCheckBox
function LifeJacketsLCheckBoxValueChanged(app, event)
    if app.LifeJacketsLCheckBox.Value
        app.LifeJackets="L";
    else
        app.LifeJackets="";
    end
    app.LifeJackets = "J/"+app.LifeJacketsL+app.LifeJacketsF+app.LifeJacketsU+app.LifeJacketsV;
    if app.LifeJacketsLCheckBox.Value==0 && app.LifeJacketsFCheckBox.Value==0 && app.LifeJacketsUCheckBox.Value==0 && app.LifeJacketsVCheckBox.Value==0
        app.LifeJackets = "";
    end
end
```

Rysunek 37 Kod obsługi jednego z pól wyboru związanego z kamizelką ratunkową

W celu realizacji podstawowego celu całej aplikacji, czyli wygenerowania depeszy planu lotu zrozumiałej dla systemów komputerowych, dane z formularza są łączone w jeden długi ciąg znaków. Funkcja *newline* przesuwaa tekst liniijkę niżej. Kod przedstawiono na rysunku 38. Instrukcje warunkowe są odpowiedzialne za przypisanie wartości zmiennej do której przypisane są informacje z pola „Remarks” czyli uwagi do planu lotu. Każda linijka kodu między wierszem 170, a 175 odpowiada jednemu wierszowi w depeszy FPL, wiersze 176 i 177 to ostatnia linia zawierająca informacje z 18 i 19 punktu formularza. Ostatnie linie kodu zapisują utworzoną depesze w piku tekstowym.

```
165 -         if isempty(app.REMARKSEditField.Value)
166 -             Remarks = "";
167 -         else
168 -             Remarks = "N/"+app.REMARKS.Value;
169 -         end
170 -         FPLmessege = "(FPL-" + app.AIRCRAFTIDENTIFICATIONEditField_2.Value + "-" + app.FLIGHTRULESDropDown.Value+app.TYPEOFFLIGHTDropD
171 -             + "-" + app.TYPEOFFAIRCRAFTEditField.Value + "/" + string(app.WAKETURBULENCECATDropDown.Value) + "-" + app.EQUIPMENTEditField.Va
172 -             + "-" + app.DEPARTUREAERODROMEEditField.Value + app.TIMEhhmmEditField.Value + newline...
173 -             + "-" + app.CRUISINGSPEEDEditField.Value + app.LEVELEditField.Value + " " + app.ROUTEEditField.Value + newline...
174 -             + "-" + app.DESTINATIONAERODROMEEditField.Value + app.TOTALEETEditField.Value + " " + app.ALTNAERODROMEEditField.Value + " " + app
175 -             + "-" + app.OTHERINFORMATIONEditField.Value...
176 -             + " " + app.SurvEq+ " P/" + app.PersonsOnBoardEditField.Value+ " A/" + app.COLOURANDMARKINGS.Value+ " " + app.LifeJackets+ " " + app
177 -             + Remarks+ " C/" + app.PilotInCommand.Value + " E/" + app.EnduranceEditField.Value + ")";
178 -
179 -         fileId= fopen('D:\Praca_Magisterska_robota\Aplikacja_Plan_Lotu3\ICAO_FPLmessege.txt','w');
180 -         fprintf(fileID,'%s',FPLmessege);
```

Rysunek 38 Tworzenie depeszy FPL

Dodatkową funkcjonalnością aplikacji jest porównanie zadeklarowanego zapasu paliwa z czasem trwania lotu. Porównywana jest wartość z obydwóch pól. Jeżeli zapas paliwa jest niewystarczający pola zmienia kolor na czerwony oraz zablokowany zostanie przycisk „OK”. Funkcja sprawdzająca wywoływana jest po każdej zmianie danych w widocznych na rysunku 39 polach. Kod przedstawiono na rysunku 40 i jest taki sam dla obydwóch pól.

16 DESTINATION AERODROME	TOTAL EET	ALTN AERODROME	2ND ALTN AERODROME
EPRZ	0210	EPLB	
18 OTHER INFORMATION			
PBN/B6			
19 ADDITIONAL INFORMATION			
ENDURANCE hhmm	PERSONS ON BOARD	EMERGENCY RADIO	
E/ 0200	P/ 220	UHF VHF ELT	
		R/ <input checked="" type="checkbox"/> U <input type="checkbox"/> V <input type="checkbox"/> E	

Rysunek 39 Porównanie całkowitego czasu lotu i zapasu paliwa

```

339 function TOTALEETEditFieldValueChanged(app, event)
340 -   if str2double(app.TOTALEETEditField.Value) > str2double(app.EnduranceEditField.Value)
341 -       app.TOTALEETEditField.BackgroundColor = "red";
342 -       app.EnduranceEditField.BackgroundColor = "red";
343 -       app.OKButton.Enable = "off";
344 -   else
345 -       app.TOTALEETEditField.BackgroundColor = "white";
346 -       app.EnduranceEditField.BackgroundColor = "white";
347 -       app.OKButton.Enable = "on";
348 -   end
349 - end

```

Rysunek 40 Porównanie całkowitego czasu lotu i zapasu paliwa

5.2 Algorytmizacja aplikacji do planowania misji

W celu przedstawienia sposobu działania aplikacji wstawiono rysunek 40. Rozpatrywany teren jest ograniczony do pewnej wielkości. Całość podzielona jest na kwadraty, krawędź każdego z nich ma 90m szerokości w terenie. Rozdzielczość jest taka sama jak rozdzielczość danych z pliku DTED zawierającym dane pozwalające utworzyć cyfrową mapę terenu. Każdemu kwadratowi przypisana jest wysokość n.p.m. terenu pobrana z pliku o formacie *dt1*. Każdemu z kwadratów zostaje przypisany jeden nadajnik i odbiornik co umożliwia w dalszej części działania skryptu, obliczenie strat propagacji.



Rysunek 41 Rysunek poglądowy obrazujący idee działania programu

W ustawieniach początkowych aplikacji (rysunek 41) zdefiniowano dwie tablice *nadajniki_rlok=[]* i *odbiorniki_rlok=[]* przechowujące dane o obiektach klasy *txsite* oraz *rxsite*, będące nadajnikiem i odbiornikiem. W miejscu w którym znajduje się radar rozmieszczany jest zarówno odbiornik i nadajnik. W 42 wierszu otwierana jest mapa 3D dzięki funkcji *siteviewer()*, 44 i 45 linia to tablice zapamiętujące wybraną trasę lotu, gdzie *trasa_tabelaGUI* to obiekt graficzny. W linii 47 wskazano plik z rozszerzeniem *dt1*. z danymi o elewacji terenu. Plik na którym opiera się program przechowuje dane dla obszaru o współrzędnych geograficznych od 39N do 40N i od 106W do 107W.

```

39 - properties (Access = public)
40 -     nadajniki_rlok=[];
41 -     odbiorniki_rlok=[];
42 -     viewer = siteviewer();
43 -     %wyznaczenie trasy
44 -     trasa_tablica=[]; %tablica wykorzystana zostanie do zoobrazowania trasy na mapie
45 -     trasa_tabelaGUI = []; %tabela to obiekt GUI
46 -     liczbaPorzadkowaPktNaTrasie=0;
47 -     dtedtfile = "n39_w106_3arc_v2.dt1";
48 - end

```

Rysunek 42 Ustawienia początkowe

Do obliczenia i wyświetlenia pokrycia terenu przez wiązkę radarową niezbędne jest w pierwszej kolejności wybór położenia stacji radiolokacyjnych. Jest to realizowane poprzez część programu pokazaną na rysunku 42. Tworzone są dwa obiekty klasy *txsite* i *rxsite* czyli nadajnik i odbiornik, o określonej przez użytkownika nazwie, wysokości anteny oraz znajdujące się w miejscu zdefiniowanych w interfejsie aplikacji współrzędnych. Dalej obiekty dodawane są do tablicy przechowującej dane o wszystkich radarach. Ostatni wiersz kodu odpowiada za wyświetlenie położenia stacji na mapie. Kod funkcji umożliwiającej usunięcie stacji nie będzie szczegółowo omawiany, działa on poprzez usunięcie wybranego obiektu z tablicy.

```

54 - function ZatwierdButtonPushed(app, event)
55 -     nadajnik = txsite("Name",app.NazwaEditField.Value, ...
56 -         "AntennaHeight",app.WysokoantyEditField.Value, ...
57 -         "Latitude",app.SzergeogrEditField.Value, ...
58 -         "Longitude",app.DugogeogrEditField.Value);
59 -
60 -     odbiornik = rxsite("Name",app.NazwaEditField.Value, ...
61 -         "AntennaHeight",app.WysokoantyEditField.Value, ...
62 -         "Latitude",app.SzergeogrEditField.Value, ...
63 -         "Longitude",app.DugogeogrEditField.Value);
64 -
65 -     app.nadajniki_rlok = [app.nadajniki_rlok nadajnik];
66 -     app.odbiorniki_rlok = [app.odbiorniki_rlok odbiornik];
67 -     show(app.nadajniki_rlok);
68 - end

```

Rysunek 43 Dodawanie stacji radiolokacyjnej

Główna część aplikacji kryje się pod przyciskiem „Pokrycie radarowe”. W pierwszej kolejności podawane są parametry związane z działaniem radaru i konieczne do wykonania obliczeń (rysunek 44).

```

84 - pd = 0.5; % Prawdopodobieństwo detekcji
85 - pfa = 1e-6; % Prawdopodobieństwo fałszywego alarmu
86 - maxrange = 35000; % Maksymalny zasięg jednoznacznego pomiaru (m)
87 - rangeres = 5; % Rozróżnialność (m)
88 - tgtrcs = .1; % Powierzchnia skuteczna celu (m^2)
89 - numpulses = 100; % Ilość impulsów w sygnale sondującym

```

Rysunek 44 Podstawowe parametry wykorzystane do obliczeń

Następnym krokiem jest obliczenie minimalnego stosunku sygnał szum sygnału jaki odebrany zostanie przez odbiornik radaru, koniecznego do wykrycia statku powietrznego.

Wykorzystano do tego równanie Albersheim'a [10]. Na następnej stronie przedstawiono wzór Albersheim'a oraz fragment skryptu z zaimplementowaną funkcją wykorzystującą równanie nr 1.

$$A = \ln \frac{0.62}{P_{FA}} \quad B = \ln \frac{P_D}{1-P_D}$$

$$SNR = 5 \log_{10} N + [6.2 + 4.54/\sqrt{N + 44}] \log_{10}(A + 0.12AB + 1.7B) \quad (1)$$

N – ilość impulsów w sygnale sondującym

```
91 % Minimalny stosunek sygnał/szum niezbędny do wykrycia statku powietrznego [dB]
92 snrthreshold = albersheim(pd, pfa, numpulses);
```

Rysunek 45 Równanie Albersheim'a

W dalszym ciągu podano kolejne dane (rysunek 46). Wyznaczono szerokość impulsu na podstawie parametru rozróżnialności radaru (zmienna *rangeres*), według wzorów 2 i 3:

$$BW = \frac{c}{2S} \quad (2)$$

$$\frac{2S}{c} = \tau = \frac{1}{BW} \quad (3)$$

BW – szerokość pasma sygnału

τ - czas trwania impulsu sondującego

```
APP 94 - fc = 10e9; % Częstotliwość pracy radaru: 10 GHz
95 - antgain = 38; % Zysk antenowy 38 dB
96 - c = physconst('LightSpeed');
97 - lambda = c/fc; %długość fali|
98
99 - pulsebw = c/(2*rangeres);
100 - pulsewidth = 1/pulsebw;
101 - Ptx = radareqpow(lambda,maxrange,snrthreshold,pulsewidth,...
102 'RCS',tgtrcs,'Gain',antgain);
```

Rysunek 46 Kod programu z równaniem zasięgu radaru

W linii 101 (rysunek 46) obliczana jest moc nadajnika radaru (wzór 3), za odległości R przyjęto maksymalny zasięg jednoznacznego pomiaru.

$$P_t = \frac{P_r(4\pi)^3 k T_s R_t^2 R_r^2 L}{N \tau G_t G_r \lambda^2 \sigma} \quad (4)$$

P_t — Moc nadajnika Peak transmit power in watts

G_t — Zysk antenowy dla anteny nadajnika w dB

G_r — Zysk antenowy dla anteny odbiornika w

λ — Długość fali

σ —Skuteczna powierzchnia odbicia

L — Straty wynikające z propagacji fali jak i z niedoskonałości urządzeń

R_t — Odległość między celem a nadajnikiem

R_r — Odległość między celem a odbiornikiem. Jeżeli radar jest monostatyczny odległości są jednakowe. Tak samo

N — Szum

CODE BRI	APP LAYOUT
103	
104	% Zdefiniowanie obszaru
105 -	latlims = [39.5 40];
106 -	lonlims = [-105.6 -105.1];
107 -	[map,~] = dted(app.dtedtfile,10,latlims,lonlims);
108	
109	% Zdefiniowanie siatki kwadratów
110 -	tglatv = linspace(latlims(1),latlims(2),length(map));
111 -	tgtlonv = linspace(lonlims(1),lonlims(2),length(map));
112 -	[tgtlons,tgtlats] = meshgrid(tgtlonv,tglatv);
113 -	tgtlons = tgtlons(:);
114 -	tgtlats = tgtlats(:);
115 -	map = map(:);

Rysunek 47 Skrypt programu

Zmienne *latlims* i *lonlims* wyznaczają rejon dla którego zostaną przeprowadzone obliczenia (rysunek 47).

Część skryptu z rysunku 47 wyznacza rejon dla którego zostaną przeprowadzone obliczenia oraz tworzy zmienne które odpowiadają każdemu z kwadratów na jakie podzielony jest teren. Do wektora *map* przypisane dane o elewacji terenu pobrane z pliku *n39_w106_3arc_v2.dt1*. Zmienne *tgtlats* i *tgtlons* odpowiadają siatce kwadratów na jakie podzielono mapę.

CODE BRI	APP LAYOUT
110	
117 -	tgtalt = app.WysokoscNadPozMorza.Value; % Wysokość przelotowa dla której generowane jest pokrycie radarowe
118	%-----
119 -	app.viewer.Name = num2str(app.WysokoscNadPozMorza.Value) + "metrów nad poziomem morza";
120 -	clearMap(app.viewer) % wyczyszczenie mapy
121	% Przyjęty model propagacji fali w terenie
122 -	pm = propagationModel('longley-ricc');
123	
124	% Obliczenie strat w wyniku propagacji sygnału w terenie
125	% i obliczenie odległości pomiędzy każdym z punktów i radarem
126 -	[L, ds] = helperPathlossOverTerrainSeaLevel(pm, app.nadajniki_rlok, ...
127	app.odbiorniki_rlok, tgtlats, tgtlons, tgtalt, map);
128	
129	% Oblicz SNR dla każdego radaru i każdego punktu w terenie
130 -	numtgts = numel(tgtlats);
131 -	numrdrs = numel(app.nadajniki_rlok);
132 -	rawsnr = zeros(numtgts,numrdrs);
133 -	for tgtind = 1:numtgts
134 -	for rdrind = 1:numrdrs
135 -	rawsnr(tgtind,rdrind) = radareqsnr(lambda,ds(tgtind,rdrind),Ptx,pulsewidth, ...
136	'Gain',antgain,'RCS',tgtrcs,'Loss',L(tgtind,rdrind));
137 -	end
138 -	end
139	
140 -	snr = max(rawsnr(:,:),[],2);

Rysunek 48 Kod programu

Do obliczeń przyjęto model propagacji fali Longel-Rice (linia 122 rysunek 48), wybrany model propagacji nie ma większego wpływu na działanie programu. Funkcja *helperPathlossOverTerrainSeaLevel* oblicza straty wynikające z propagacji fali w przestrzeni uwzględniając ukształtowanie terenu. Wewnątrz funkcji każdemu skrawkowi terenu przypisano jeden nadajnik i jeden odbiornik niezbędny do wykonania obliczeń. Wynika to ze specyfiki funkcji *pathloss* znajdującej się wewnątrz funkcji *helperPathlossOverTerrainSeaLevel*, ponieważ argumentami jakie trzeba w niej podać są obiekty klasy *txsite* i *rxsite*. Funkcja *pathloss* oblicza straty pomiędzy odbiornikiem, a nadajnikiem. Funkcja *helperPathlossOverTerrainSeaLevel* zwraca odległości i straty wynikające z propagacji sygnału pomiędzy każdym punktem w terenie a radarem. Funkcja ta została napisana przez twórców środowiska Matlab, jednak w pierwotnej wersji rozmieszczała ona symulowane nadajniki i odbiorniki na powierzchni ziemi, aby program mógł zadziałać została ona zmodyfikowana, żeby wszystkie nadajniki i odbiorniki znajdowały się na zadanej przez użytkownika wysokości n.p.m. czyli takiej na jakiej przelatuje statek powietrzny. Zwrócone przez omawianą funkcję wartości są wykorzystywane w funkcji *radareqsnr* (linia 135 kodu). Funkcja ta znów wykorzystuje równanie zasięgu radaru (równanie nr 5), tym razem w celu obliczenia SNR pomiędzy każdym z punktów na mapie a stacjami radiolokacyjnymi. W równaniu nr 5 symbole oznaczają te same parametry co w równaniu nr 4.

$$\frac{P_r}{N} = \frac{P_t(4\pi)^3 k T_s R_t^2 R_r^2 L}{N \tau G_t G_r \lambda^2 \sigma} \quad (5)$$

Zmienna *rawsnr* przechowuje dane o stosunku sygnał/szum pomiędzy wszystkimi radarami a punktami. Jeżeli zadeklarowano by np. 5 radarów, wynik działania tej funkcji dałby tablicę 5 kolumn, z tyloma wierszami na ile kwadratów podzielono mapę. W linii 140 wybierane są największe wartości SNR i to na ich podstawie jest generowany czerwony kontur na mapie.


```

140 - snr = max(rawsnr(:,:),[],2);
141
142 % Wyświetl położenie radarów
143 - show(app.nadajniki_rlok);
144 - if isempty(app.trasa_tablica)
145 - else
146 - show(app.trasa_tablica);
147 - end
148
149 - snr_elewacja=snr;
150 - for i=1:length(snr)
151 -     if map(i,1)>tgtalt
152 -         snr_elewacja(i,1)=10000;
153 -     end
154 - end
155 - strefaBezpieczna = -1000;
156 - for i=1:length(snr)
157 -     if snr_elewacja(i,1)<snrthreshold
158 -         snr_elewacja(i,1)=strefaBezpieczna;
159 -     end
160 - end
161 % Wyświetlenie stref zagrożenia dla statku powietrznego
162 - rdrData = propagationData(tgtlats,tgtlons,"SNR",snr_elewacja);
163 - contour(rdrData,...
164     "Levels",[ snrthreshold snrthreshold+100 ], ...
165     "Colors",[ "red" "yellow"]);
166 - end

```

Rysunek 49 Kod programu

Tak samo jak do wektora *map* przypisano wartości elewacji dla każdego punktu, tak dla wektora *snr_elewacja* (rysunek 49) przypisano dane niezbędne do wygenerowania czerwonego i żółtego konturu pokrywającego mapę. Domyślnie każda wartość tego wektora wynosi tyle ile obliczony dla każdego kwadratu stosunek sygnał/szum. Jeżeli jednak w danym kwadracie wysokość terenu przekracza wysokość lotu, podstawiana jest w to miejsce wartość 10000, jeśli wartość SNR jest poniżej minimalnej wartości koniecznej do wykrycia statku powietrznego (przy zadanym prawdopodobieństwie detekcji i innych parametrach) wtedy kwadratowi przypisana zostanie wartość -1000 (minus tysiąc). Jako parametry funkcji *contour* podane zostały wartości powyżej których dany kwadrat ma zostać pokryty kolorem żółtym lub czerwonym. Kwadraty którym przypisano wartości między minimalną wartością SNR a 10000 zostaną pokryte czerwonym kolorem, natomiast te o wartości 10000 kolorem żółtym jako przeszkody terenowe.

6 Podsumowanie i wnioski

Informacje zawarte w planie lotu są niezbędne dla funkcjonowania służb kontroli ruchu lotniczego. Istnieje kilka metod przesłania planu lotu do odpowiednich służb, zaczynając od wykorzystania faksu, telefonicznie, przez sieć AFTN i poprzez specjalne do tego aplikacje komputerowe, często dedykowane dla obszaru konkretnego państwa jak np. amerykański SkyVector, polskie IWB czy australijski AvPlan. Najbardziej przystępnym sposobem jest wykorzystanie aplikacji.

Napisane w niniejszej pracy oprogramowanie spełnia podstawowe funkcjonalności niezbędne do opracowania planu lotu. W celu rozwoju aplikacji można by wprowadzić jeszcze bardzo wiele udoskonaleń, zaczynając od zoptymalizowania jej działania, dodając bazę danych o dostępnych drogach lotniczych i umożliwić ich wybór w interfejsie narzędzia do wyznaczania trasy, umożliwić eksport do formatu pdf oraz dodatkowo opracować jej wersję do której można by uzyskać dostęp przez Internet oraz wiele innych udoskonaleń.

Aplikacja do wspomagania planowania misji ma charakter demonstracyjny. Osiągnięto cel jakim jest wygenerowanie pokrycia terenu przez radar. Wyraźnie pokazuje problem jaki postanowiono rozwiązać pisząc niniejszą pracę. Aplikacja może być również wykorzystana do rozplanowania optymalnego rozstawienia stacji radiolokacyjnych.

Podobnie jak aplikacja do planowania lotu może być z powodzeniem rozwijana w przyszłości uwzględniając dostęp do baz danych terenu oraz ograniczenia środowiska programistycznego.

LITERATURA:

- [1] U.S. Department of Transportation Federal Aviation Administration, Guidelines for the Certification, Airworthiness, and Operational Use of Electronic Flight Bags, 120-76B, z dnia 01.06.2012
- [2] WYTYCZNE NR 5 PREZESA URZĘDU LOTNICTWA CYWILNEGO z dnia 13 kwietnia 2017 r. w sprawie ogłoszenia wymagań ustanowionych przez Organizację Międzynarodowego Lotnictwa Cywilnego (ICAO) – Doc 4444
- [3] IFPS and RPL Dictionary of Messages; ICAO 2012 Special edition
- [4] FAA ICAO FPL Quick Guide (2019), Version 7, 14 June 2019
- [5] EUROCONTROL Specification for ATS Data Exchange Presentation (ADEXP)
- [6] EUROCONTROL, IFPS USERS MANUAL, 07/05/2019
- [7] https://www.faa.gov/about/office_org/headquarters_offices/ato/service_units/air_traffic_services/flight_plan_filing/
- [8] <https://passyourmessage.com/2014/09/08/aftn-flight-planning-network/>, 26.05.2021
- [9] TACTICAL FLIGHT MISSION PLANNING AND MAP PREPARATION GUIDE, 7 December 2009
- [10] <https://uk.mathworks.com/help/phased/ref/albersheim.html>, , 26.05.2021
- [11] <https://www.avplan-efb.com/avplan/>, , 26.05.2021
- [12] <https://data.worldbank.org/indicator/IS.AIR.PSGR?end=2019&start=1970&view=mchart>, 26.05.2021
- [13] https://www.faa.gov/about/office_org/headquarters_offices/ato/service_units/air_traffic_services/flight_plan_filing/?source=post_page, 26.05.2021
- [14] <https://www.aircraftit.com/articles/what-is-an-efb/>, 26.05.2021
- [15] Polska Agencja Żeglugi Powietrznej, IWB Integrated Web Briefing Instrukcja użytkownika
- [16] <https://skyvector.com/Flight-Planning-and-Filing-Pilots-Guide>, 26.05.2021
- [17] Materiały dydaktyczne z przedmiotu Podstawy teledetekcji wykład pt. „Radary lądowe oraz morskie”, dr hab. inż. prof. WAT Jerzy Pietrański - <http://ire.wel.wat.edu.pl/index.php/pl/zaklad-teledetekcji/dzialalnosc-dydaktyczna/podstawy-teledetekcji/materiały-dydaktyczne>, 26.05.2021