

Sem vložte zadání Vaší práce.

ČESKÉ VYSOKÉ UČENÍ TECHNICKÉ V PRAZE
FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
KATEDRA SOFTWAREVÉHO INŽENÝRSTVÍ



Diplomová práce

Návrh IS pro plánování letu ultralehkých letadel

Bc. Stanislav Mayer

Vedoucí práce: Ing. Tomáš Levora

16. května 2014

Poděkování

Chtěl bych poděkovat svému vedoucímu práce Ing. Tomášovi Levorovi za cenné rady, které mi k této práci poskytl. Dále bych rád poděkoval své rodině a kamarádům, kteří mě podporovali v mém studiu.

Prohlášení

Prohlašuji, že jsem předloženou práci vypracoval(a) samostatně a že jsem uvedl(a) veškeré použité informační zdroje v souladu s Metodickým pokynem o etické přípravě vysokoškolských závěrečných prací.

Beru na vědomí, že se na moji práci vztahují práva a povinnosti vyplývající ze zákona č. 121/2000 Sb., autorského zákona, ve znění pozdějších předpisů. V souladu s ust. § 46 odst. 6 tohoto zákona tímto uděluji nevýhradní oprávnění (licenci) k užití této mojí práce, a to včetně všech počítačových programů, jež jsou její součástí či přílohou a veškeré jejich dokumentace (dále souhrnně jen „Dílo“), a to všem osobám, které si přejí Dílo užít. Tyto osoby jsou oprávněny Dílo užít jakýmkoli způsobem, který nesnižuje hodnotu Díla a za jakýmkoli účelem (včetně užití k výdělečným účelům). Toto oprávnění je časově, teritoriálně i množstevně neomezené. Každá osoba, která využije výše uvedenou licenci, se však zavazuje udělit ke každému dílu, které vznikne (byť jen zčásti) na základě Díla, úpravou Díla, spojením Díla s jiným dílem, zařazením Díla do díla souborného či spracováním Díla (včetně překladu), licenci alespoň ve výše uvedeném rozsahu a zároveň zpřístupnit zdrojový kód takového díla alespoň srovnatelným způsobem a ve srovnatelném rozsahu, jako je zpřístupněn zdrojový kód Díla.

V Praze dne 16. května 2014

.....

České vysoké učení technické v Praze
Fakulta informačních technologií

© 2014 Stanislav Mayer. Všechna práva vyhrazena.

Tato práce vznikla jako školní dílo na Českém vysokém učení technickém v Praze, Fakultě informačních technologií. Práce je chráněna právními předpisy a mezinárodními úmluvami o právu autorském a právech souvisejících s právem autorským. K jejímu užití, s výjimkou bezúplatných zákonných licencí, je nezbytný souhlas autora.

Odkaz na tuto práci

Mayer, Stanislav. *Návrh IS pro plánování letu ultralehkých letadel*. Diplomová práce. Praha: České vysoké učení technické v Praze, Fakulta informačních technologií, 2014.

Abstract

This thesis deals with design and implementation of mobile application for ultralight aircrafts. Application is able to find suitable route for emergency landing. Application supports navigation along computed route. Another part of this thesis is research of familiar applications and testing.

Keywords Ultralight aircrafts, route, computing, emergency, Android, Java

Abstrakt

Práce se zabývá návrhem a implementací mobilní aplikace, která dokáže najít vhodnou trajektorii pro nouzové přistání ultralehkých letadel a navigovat po ni. Součástí práce je řešení podobných aplikací a testování vytvořené aplikace.

Klíčová slova Ultralehká letadla, ULL, trajektorie, nouzová situace, Android, Java

Obsah

Úvod	1
Cíle práce	2
1 Analýza	3
1.1 Definice aviatických pojmů	3
1.2 Flight Management Systémy	8
2 Rešerše Flight Management Systémů	9
2.1 VP-400	9
2.2 Sky Demon	13
2.3 Air Navigation Pro	15
2.4 LK8000 – Tactical Flight Computer	18
2.5 Srovnání jednotlivých aplikací	20
3 Návrh	21
3.1 Požadavky na aplikaci	21
3.2 Případy užití	22
3.3 Návrh modulů a komunikačních API	23
3.4 Výběr platformy	25
3.5 Návrh uživatelského rozhraní	26
3.6 NMEA komunikační protokol	29
3.7 Databáze letišť	30
4 Implementace	31
4.1 Objektový návrh	31
4.2 Modul EmergencyLanding	31
4.3 Modul Client	46
4.4 Modul Position	48
4.5 Modul SharedPreferences	48

5	Testování	51
5.1	Unit testy	51
5.2	Manuální testování trajektorie na simulátoru	52
5.3	Manuální testování aplikace na simulátoru	54
5.4	Reálné testování aplikace v reálné prostředí	54
	Závěr	57
	Literatura	59
A	Seznam použitých zkratk	63
B	Obsah přiloženého CD	65
C	Zpráva z reálného testování	67
C.1	Prostředky a místo	67
C.2	Průběh testu	67
C.3	Hodnocení aplikace	68
C.4	Poznámky a návrhy ke zlepšení	68
D	Instalační příručka	71
E	Formáty používaných NMEA zpráv	73

Seznam obrázků

1.1	Závislost TAS na nadmořské výšce letadla	4
1.2	Dopředná rychlost a rychlost opadání	5
1.3	Rychlostní polára	6
1.4	Vliv větru na rychlost a směr letadla	7
1.5	Flight Management System používaný v moderních letadlech [8]	8
2.1	Přístroj VP-400 a jeho součásti [25]	10
2.2	GUI přístroje VP-400 [25]	11
2.3	Energy metr přístroje VP-400 [25]	12
2.4	Screenshot aplikace Sky Demon za letu	15
2.5	Vertikální profil trasy aplikace Sky Demon	15
2.6	Ukázka aplikace Air Navigation Pro za letu [36]	17
2.7	3D pohled aplikace Air Navigation Pro [36]	17
2.8	Aplikace LK8000 - letový režim [16]	18
2.9	Aplikace LK8000 - mód kroužení v termickém proudu [7]	19
3.1	Diagram případů užití	22
3.2	Komponent diagram aplikace	24
3.3	Navržené rozhraní pro komunikaci mezi moduly aplikace	24
3.4	Navržené GUI	27
3.5	Navržené GUI pro nastavení parametrů	28
3.6	Propojení jednotlivých aktivit	29
4.1	Class diagram hlavních tříd modulu EmergencyLanding	32
4.2	Ukázka třídy Point	33
4.3	Využití funkcí sin a cos ke zjištění zeměpisných souřadnic dalšího počítaného bodu.	34
4.4	Grafické znázornění vytvořené zatáčky shora	36
4.5	Grafické znázornění výpočtu délky rovné trasy v zatáčce	36
4.6	Rozhodovací proces tvoření zatáčky	37
4.7	Vliv větru na provedenou zatáčku	38

4.8	Grafické znázornění vytvoření zatáčky při neznámém poloměru . . .	39
4.9	Grafické znázornění výpočtu poloměru zatáčky	39
4.10	Grafické znázornění tvorby zatáčky	40
4.11	Grafické znázornění komplexního výpočtu poloměru zatáčky	40
4.12	Ukázka trajektorie letu FinaleFixedPath	43
4.13	Ukázka trajektorie letu FinaleFixedPath zobrazená ve 3D	43
4.14	Ukázka trajektorie ThreeTurnsPath	44
4.15	Ukázka trajektorie TwoTurnsPath	45
4.16	Class diagram modulu Client	46
4.17	SettingsActivity s příliš silným zadaným větrem	47
4.18	Class diagram modulu Position	48
4.19	Class diagram modulu SharedPreferences	49
5.1	Porovnání vypočítané trajektorie a letu kluzáku v simulátoru Con- dor podél trajektorie.	53
5.2	Ukázka testování vypočtené trajektorie za využití ducha v simulá- toru Condor	54
5.3	Ukázka spočtené trajektorie a reálného letu	55

Úvod

Doby, kdy bylo ultralehké letadlo (dále jen ULL) synonymem pro amatérsky vyrobený stroj z trubek, plátna a lan, jsou pryč. Díky moderním lehkým kompozitovým materiálům vypadá moderní ULL k nerozeznání od běžných letadel jako je například Cessna 182. Výkony ULL dokonce leckdy převyšují své větší vzory. Další výhodou ULL letadel je nižší pořizovací cena a úspora paliva. Díky těmto faktorům jsou ULL dostupná veřejnosti víc než kdy dříve. V posledních letech narůstá nejenom počet pilotních licencí na ULL, ale spolu s tím roste i počet nehod.

Nehody ULL jsou většinou způsobené kombinací pilotní chyby a technické závady na letadle. ULL nemusí podle zákona obsahovat certifikovanou techniku. Proto nejsou technické závady letounu nijak výjimečné. Mezi hlavní poruchovou jednotku patří motor. V případě výpadku motoru letadlo klouže klesavým letem a jeho dokluz je omezen. Pilot se zpravidla snaží rychle najít nejbližší letiště nebo alespoň vhodnou plochu pro přistání. V případě, že pilot nalezne přistávací plochu, snaží se v klouzavém režimu zaletět několik vyklesávacích zataček tak, aby dosedl na určené místo. Piloti ULL nejsou na klouzavý režim letu příliš zvyklí a jsou vlivem nouzové situace ve stresu. V takových případech mohou piloti dělat chyby častěji než obvykle. [20]

Druhou častou příčinou leteckých nehod ULL je špatné počasí a to hlavně při špatné viditelnosti. V takových situacích může mít pilot dohled tak malý, že nenajde vhodnou plochu nebo dokonce ani samotné letiště. Na takové nouzové situace se zaměřuje vyvíjená aplikace, která si klade za cíl pomoci pilotům ULL v nouzové situaci. Aplikace umožní omezit chyby špatného rozpočtu na přistání a pomocí navigace určí pilotům bezpečnou trasu pro přistání.

Cíle práce

Cílem diplomové práce je návrh a implementace prototypové FMS (Flight Management System) aplikace, která bude umožňovat vypočítat trasu letadla pro nouzové přistání. Aplikace bude podporovat navigaci letadla podél vypočítané trasy. Součástí práce je výběr vhodné technologie a řešerše podobných aplikací.

Analýza

Tato kapitola nejprve definuje základní aviatické pojmy nutné pro porozumění práci. Dále se kapitola věnuje Flight Management Systémům a popisuje jejich použití v ULL.

1.1 Definice aviatických pojmů

1.1.1 Proudnice

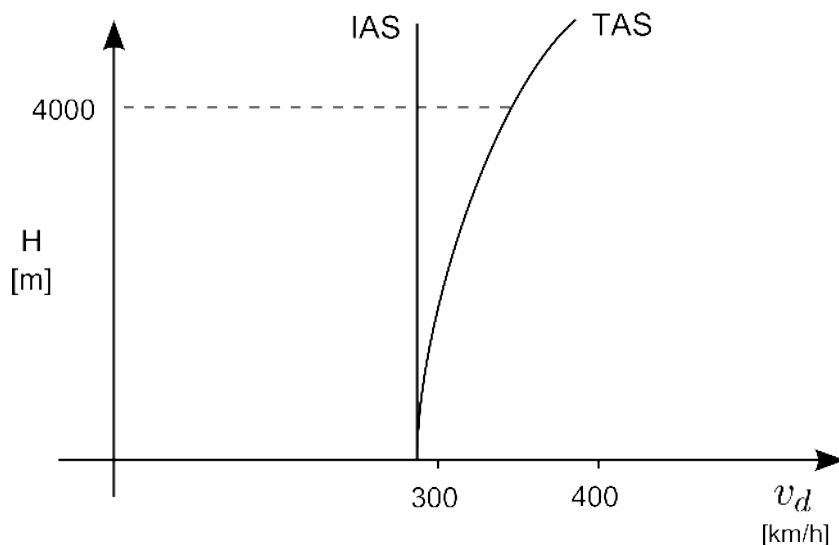
Proudnice je křivka, jejíž tečny mají v každém bodě směr rychlosti proudu vzduchu. V ustáleném proudu je to dráha jednotlivých částic nebo také molekul. Zhuštěné proudnice vyjadřují narychlení proudu, naopak pokud se proudnice od sebe vzdalují, bude rychlost proudění menší. [6]

1.1.2 Indikovaná vzdušná rychlost

Indikovaná vzdušná rychlost IAS (Indicated AirSpeed) je funkce dynamického tlaku působícího na snímač umístěný vně letounu. Díky různé hustotě vzduchu v různých výškách je to rychlost pouze zdánlivá, vyjadřující aerodynamické obtékání letadla. Ve vyšších nadmořských výškách je hustota vzduchu nižší, a proto bude rychlost IAS ukazovat nižší rychlost než u země. Znalost IAS je naprosto zásadní pro řízení letadla. Pilot je díky rychloměru IAS schopen udržovat letadlo ve stejném režimu letu v různých nadmořských výškách. [6]

1.1.3 Pravá vzdušná rychlost

Pravá vzdušná rychlost označována jako TAS (True AirSpeed) je rychlost, kterou se letadlo pohybuje vůči okolnímu vzduchu. Je obtížné ji změřit, lze ji ale vypočítat v závislosti na dalších leteckých rychlostech, barometrického tlaku a teploty vzduchu. Rychlost IAS je shodná s TAS ve výšce hladiny moře. Letadlo pohybující se stále stejnou rychlostí TAS bude mít ve výšce – vzhledem k řidšímu vzduchu – nižší IAS než u země, viz obrázek 1.1. [6]



Obrázek 1.1: Závislost TAS na nadmořské výšce letadla

1.1.4 Rychlost vůči zemi

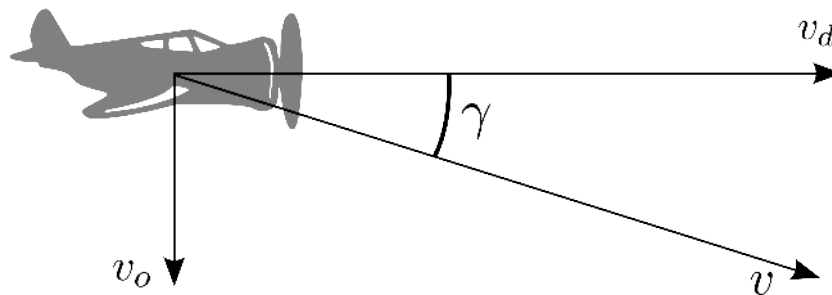
Rychlost vůči zemi, označována jako GS (Ground Speed), je součtem vektorů TAS a pohybu vzduchu (větru). Rychlost GS lze odhadovat složitými technikami. Jedna z možností je využití pozemních vysílačů a změny údajů DME (Distance Measuring Equipment) za daný čas. Měřit rychlost GS také umožňuje moderní technologie GPS, která odhaduje pozici na základně časové prodlevy mezi přijímanými signály z družic. Pokud fouká silný vítr, může se GS od TAS velmi lišit.

1.1.5 Pádová rychlost

Pádová rychlost letadla je tak malá rychlost letadla, při které dochází k odtržení proudnic od křídel a propadu letadla. Pádová rychlost je pro každý model letadla různá a v zatáčce je pádová rychlost letadla vyšší [6]. Pádová rychlost vyjádřená v IAS je zobrazena na každém rychloměru letadla a je v různých nadmořských výškách stejná. Pádová rychlost vyjádřená v rychlosti TAS roste s přibývajícím nadmořskou výškou. Pádová rychlost je uvedena v každé příručce konkrétního letadla.

1.1.6 Dopředná rychlost a rychlost opadání

Uvažujme bezmotorový ustálený režim letu v bezvětrí. TAS rychlost letounu si lze představit jako vektor v prostoru. Rozdělme vektor na dvě navzájem nezávislé složky. Svislou složku nazveme *rychlost opadání* v_o a vodorovnou složku *dopřednou rychlostí* v_d [23], viz obrázek 1.2.



Obrázek 1.2: Dopředná rychlost a rychlost opadání

1.1.7 Úhel klouzání

Úhel γ mezi horizontální rovinou a vektorem rychlosti se nazývá *úhlem klouzání* [23]. Úhel spočítáme pomocí dopředné rychlosti v_d a rychlosti opadání v_o , viz rovnice 1.1.

$$\gamma = \tan^{-1} \frac{v_o}{v_d} \quad (1.1)$$

1.1.8 Klouzavost

Klouzavost letadla je parametr, který se měří v ustáleném bezmotorovém režimu letu. Klouzavost letadla je poměr mezi dopřednou rychlostí a rychlostí opadání, viz rovnice 1.2.

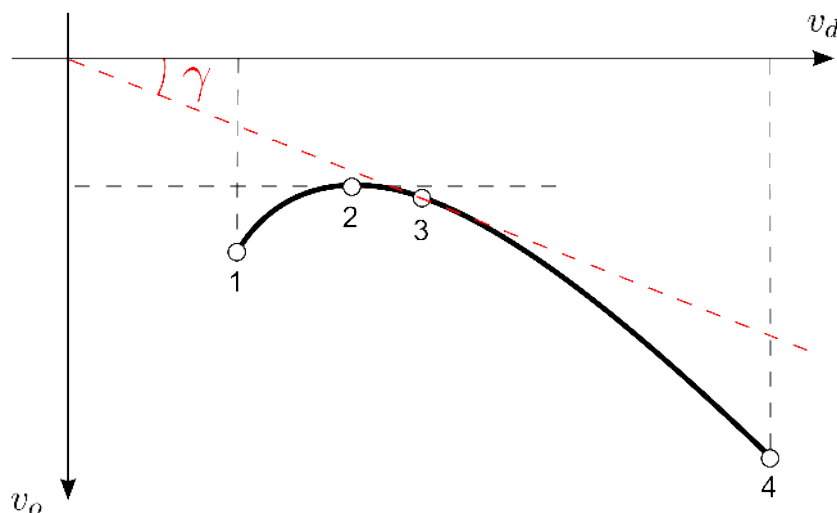
$$K = \frac{v_d}{v_o} \quad (1.2)$$

Klouzavost 20 znamená, že letadlo dokáže doletět v bezmotorovém režimu z výšky 1000 metrů na vzdálenost 20 kilometrů [31]. Klouzavost je zásadním parametrem kluzáků, ale měl by jí přibližně znát i pilot motorového letounu pro případ nouzového přistání po vysazení motoru.

1.1.9 Rychlostní polára

Rychlost v_o se mění v závislosti na v_d , z důvodu měnících se odporových a vztlakových sil, které působí na letadlo. Rychlostní polára je závislost v_d na v_o letounu. Je to funkce $f(\gamma)$, která každému úhlu klouzání přiřadí odpovídající rychlosti v_d a v_o , viz obrázek 1.3. Křivka určuje čtyři důležité body.[23]

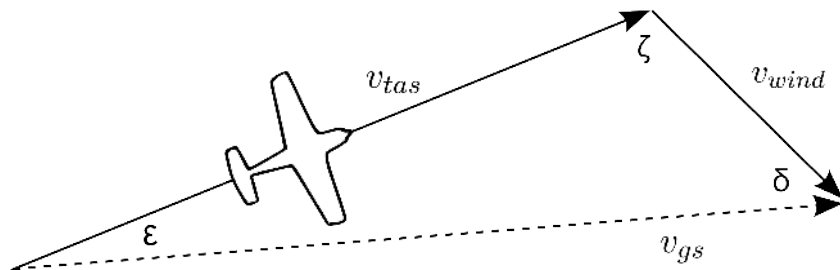
Bod 1 určuje pádovou rychlost letadla. Bod 2 určuje rychlost nejmenšího opadání. Rychlost určující bod 2 je vhodné dodržovat v případě, že chce pilot během bezmotorového letu strávit ve vzduchu nejdelší možný čas. Bod 3 určuje dopřednou rychlost letadla, při které letadlo doklouže na nejdelší vzdálenost. Bod 4 určuje maximální rychlost letadla.



Obrázek 1.3: Rychlostní polára

1.1.10 Vliv větru

Vliv větru na ULL je velmi významný a nelze ho zanedbat. Uvažujme letadlo letící rychlostí v_{tas} . V případě, že se okolní vzduchová hmota bude pohybovat, bude i letadlo unášeno s okolní vzduchovou hmotou. Čím pomaleji letadlo letí, tím je vliv větru působící na letadlo významnější. Obrázek 1.4 ukazuje jednotlivé vektory rychlosti a směru letadla. Další vektor v obrázku je vektor rychlosti a směru větru. Celkovou rychlost a směr letadla dostaneme sčítáním vektorů. K tomu nám pomůže Kosinová věta, viz rovnice 1.3.



Obrázek 1.4: Vliv větru na rychlost a směr letadla

$$v_{gs} = \sqrt{v_{tas}^2 + v_{wind}^2 + 2 \cdot v_{tas} \cdot v_{wind} \cdot \cos \zeta} \quad (1.3)$$

Uvažujme bezmotorový ustálený režim letu. Uvažujme letoun letící klouzavým letem proti větru. V takovém případě je rychlost GS zmenšená o rychlost větru. Pokud bychom v takovém případě uvažovali klouzavost letadla vztaženou vůči zemi, bude klouzavost nižší než za bezvětří. Zavedeme klouzavost K_{ground} , která bude vztažená vůči zemi, viz rovnice 1.4. Zavedeme si i pojem úhel klouzání vztažený vůči zemi φ_{ground} , viz rovnice 1.5.

$$K_{ground} = \frac{v_{gs}}{v_o} \quad (1.4)$$

$$\varphi_{ground} = \tan^{-1} \frac{v_o}{v_{gs}} \quad (1.5)$$

1.1.11 Ultralehké letadlo

Ultralehký letoun, označován zkratkou ULL, je takový letoun, který je konstruován maximálně pro dvě osoby, řízený přesouváním těžiště pilota nebo aerodynamickými prostředky, jehož pádová rychlost nepřevyšuje 65 km/h [11]. Limity pro maximální vzletovou váhu ULL se v každé zemi liší. V České republice je maximální povolená vzletová hmotnost 450 kilogramů [12].

1.1.12 Kluzák

Kluzák, nebo také větroň, je bezmotorové letadlo těžší vzduchu, s pevnými nosnými plochami. Kluzáky startují za pomoci tažného letounu nebo pomocí tažného lana a navijáku. Běžná klouzavost těchto letadel se pohybuje okolo hodnoty 40. Nejmodernější typy dosahují klouzavosti 65. [32]

1.2 Flight Management Systémy

Flight management system (dále jen FMS) je palubní počítač letadla, který umožňuje plánování a optimalizaci letu. Jeho hlavním úkolem je minimalizovat pracovní vytížení posádky letadla. FMS ukazuje během jednotlivých fází letu optimální nastavení letových parametrů většinou za účelem úspory paliva [26].

První FMS byl vyvinut pro letadlo Boeing 767 v osmdesátých letech 19. století. Původně uměly FMS pouze navigovat letadlo za využití pozemních vysílačů. Později umožňovaly FMS zobrazení směru a síly větru, počítaly čas příletu ETA (Estimated Time of Arrival) a mnoho dalšího.

V dnešní době jsou již FMS součástí všech dopravních letadel. Ukázka moderního FMS je na obrázku 1.5. Integrovaný FMS má v dnešní době i tak malé letadlo, jako je Cesna 182. Pro další letadla, která nemají integrovaný FMS, jsou k dispozici externí FMS. Externí FMS jsou v posledních letech stavěné na operačních systémech Android a iOS. FMS postavené na mobilních operačních systémech nemohou nahradit integrované FMS v dopravních letadlech, kde je nutná přesnost a spolehlivost. Mohou však být vhodně použity v menších až čtyřmístných letadlech a s použitím senzoru GPS implementovat funkce FMS.

Většina pilotů ULL nevlastní svoje letadlo. Piloti si běžně půjčují ULL z aeroklubu, ve kterém mají členství. Piloti ULL využívají zjednodušené FMS, které mají zpravidla implementovány méně funkcí. Piloti ULL používají tyto FMS především pro orientaci během letu a pro získání leteckých informací. FMS upravený pro ULL umožňuje pilotům také vytvořit letový plán, podle kterého piloti následně pilotují letadlo. Aplikace FMS běžně zobrazují letecké prostory, kterým se pilot musí vyhnout. Většina aplikací upozorní pilota několik kilometrů před vletnutím do letového prostoru. V následujícím textu bude uvažován pouze FMS pro ultralehká letadla, který je oproti běžným FMS omezen.



Obrázek 1.5: Flight Management System používaný v moderních letadlech [8]

Rešerše Flight Management Systémů

V této kapitole diplomové práce je zpracována rešerše stávajících řešení FMS pro ULL a bezmotorová letadla. Aplikace jsou zhodnoceny a jsou podrobně popsány prvky aplikace, které by byly použitelné pro implementovanou aplikaci. Pro názornost je popsán i palubní počítač pro kluzáky.

2.1 VP-400

2.1.1 Základní údaje

VP-400 je letecký přístroj vyvinutý americkou společností Vertical Power. Společnost se zabývá především tvorbou leteckého softwaru pro SLZ (Sportovní létající zařízení) a malá letadla. Společnost Vertical Power je zaměřena výhradně na americký trh [25].

Nejprve společnost vyvíjela přístroje řady VP-X, které sloužily k monitoringu a ovládání veškeré elektroniky v letadle. Tyto přístroje rozpoznají např. prasklou žárovku signalizačního světla, nebo umí odpojit obvod startéru po nastartování motoru. Pro zobrazení elektronických informací a pro ovládání přístroje používaly přístroje typu VP-X napojení na letový informační systém EFIS (Electronic Flight Information System) [25].

Nyní se společnost Vertical Power zaměřuje na vývoj přístrojů určených pro nouzové situace a vytváří k nim i vlastní palubní GUI (Graphical User Interface). Použití těchto přístrojů je vhodné při výpadku motoru, výpadku EFIS přístrojového vybavení, zhoršení meteorologických podmínek a také při náhlé indispozici pilota. Nejnovější přístroj zaměřený na nouzové situace je produkt VP-400. Tento přístroj navazuje na předchozí modely VP-X. Kromě toho přidává GADAHRS (GPS, Air Data, Attitude, Heading Reference System) jednotku, GUI a novou funkcionalitu. Přístroj je velmi sofistikovaný a do-

káže zcela bez zásahu pilota řídit letadlo a dokonce s ním i přistát. Výrobce stanovuje na svých webových stránkách cenu přístroje VP-400 na 8000 \$ [25].

Společnost Vertical Power později uzavřela smlouvu s firmou X-Avionics a umožnila přepsat aplikaci VP-400 pro iOS. Nově vzniklá aplikace dostala název Xavion. Rozdíl vůči původní aplikaci je markantní. Xavion aplikace umožňuje zobrazení navigace na displeji, ale nepodporuje ovládání letadla a není s letadlem nijak spojená. Výhodou aplikace Xavion je přenositelnost mezi letadly a možnost použití v letadlech s certifikací elektronického vybavení. Aplikace Xavion stojí na AppStoru 99 \$ [34]. Podrobné rozdíly aplikací budou podrobně popsány na konci rešerše.

2.1.2 Funkce aplikace

Přístroj VP-400 se skládá z tří hlavních komponent. Skládá se z displeje, PDU (Power Distribution Unit) jednotky a GADAHRS jednotky, viz obrázek 2.1.



Obrázek 2.1: Přístroj VP-400 a jeho součásti [25]

GADAHRS jednotka obsahuje integrovaná čidla pro určení polohy a náklonu letadla. Jednotka také obsahuje GPS senzor, tlakový senzor výšky a gyroskopický senzor. GADAHRS jednotku je možné připojit na rychlostní sondu letadla tzv. vzdušná rychlost.

PDU jednotka poskytuje propojení s ovládacími prvky letadla a dalšími elektro prvky jako jsou např. i světla letadla. Díky PDU jednotce může přístroj VP-400 ovládat celé letadlo. Výrobce doporučuje při přiblížení na přistání okolo 100 metrů nad zemí vypnout autopilota přístroje VP-400 a převzít řízení. Pokud pilot ovšem např. ze zdravotních důvodů nepřevzme řízení, letadlo přistane zcela automaticky samo. VP-400 v takovém případě udržuje letadlo

v rychlosti nejmenšího opadání, dokud se letadlo nedotkne země. Je reálné, že se letadlo poškodí vlivem tvrdého přistání, ale neměla by se stát závažnější nehoda [25].

2.1.3 GUI a funkce aplikace

Letecký přístroj VP-400 přepočítává během letu nejlepší možnou přistávací trasu na nejvhodnější letiště a runway. V průběhu celého letu je na obrazovce zobrazena aktuálně vypočítaná trasa.

Displej přístroje VP-400 ukazuje ve vrchní části 3D pohled krajiny. Přes 3D pohled umožňuje aplikace vykreslit náklon letadla, spolu s kompasem umístěným ve spodní části horní obrazovky [15]. Dále aplikace zobrazuje GPS výšku v pravé části a také výšku AGL (Above Ground Level). V levé části 3D pohledu vykresluje aplikace rychlost letadla a pod rychlostí zobrazuje směr větru. GUI aplikace VP-400 je zobrazeno na obrázku 2.2.



Obrázek 2.2: GUI přístroje VP-400 [25]

Zajímavostí 3D vrchní části je zobrazení nouzové trasy. Trasa je vykreslena

pomocí obrazců U , které jsou zobrazeny vždy po 100 metrech trasy. V 3D pohledu je pak dobře vidět zamýšlená zatáčka či vyklesání. Obrazce ve tvaru U mají bílou barvu, ale po stisknutí nouzového tlačítka pro ostrý běh aplikace, jsou U obrazce fialové. Výjimku tvoří poslední obrazec před letištěm umístěný 100 metrů nad zemí, který je zbarvený zeleně. Obrazce U jsou za sebou umístěny postupně v nižší výšce a to právě kvůli nouzové situaci, pro kterou je přístroj určen. Mezi těmito obrazci by mělo letadlo letět přímo a klouzavým letem se správným opadáním. Pokud je aplikace v automatickém módu, letadlo ovládá PDU jednotka a hladce prolétá mezi obrazci. Pokud ovšem chce pilot ovládat letadlo sám, tak se snaží při nouzové situaci proletět U obrazci co nejpřesněji. Pokud tak letí, navigace ho dovede těsně nad přistávací dráhu a tam umožní pilotovi přistát [15].

Ve spodní části displeje je vykreslená 2D mapa se zobrazenými letišti a s plánovanou trasou. Letiště jsou zobrazená jako kruhy s úsečkou uprostřed. Úsečka značí směr přistávací dráhy. Barva kruhu je určena podle vypočítané použitelnosti konkrétního letiště. Pokud je tedy letiště dobře použitelné, aplikace ho zobrazí žlutozeleně. Pokud není příliš vhodné pro přistání, je letiště zobrazeno dočervena [15]. Letiště, na které nejsme schopni v žádném případě doklouzat, zobrazeny nejsou.

Použitelnost letiště pro přistání aplikace propočítává z několika údajů, hlavní údaje jsou vzdálenost letiště a nadmořská výška. Z těchto údajů aplikace vypočítává, zdali a v jaké výšce přiletí letadlo k letišti. Další údaje pro výpočet použitelnosti letiště jsou směr přistávací dráhy vůči větru a také délka a povrch VPD. Letiště, na které uživatel chce nouzově přistát, může uživatel zadat také ručně kliknutím a potvrzením vybraného letiště [25].



Obrázek 2.3: Energy metr přístroje VP-400 [25]

Zajímavostí aplikace je, že po použití nouzového tlačítka pro navigaci na letiště, není již přepočítávána trasa na vybrané letiště. Vzhledem k pevně dané

trase vyvinuli vývojáři tzv. Energy metr, který je zobrazen vpravo od 2D mapy. Zobrazená energie letadla je vypočítávána ze vzdušné rychlosti a z výšky letadla, oproti vypočítané ideální trase na letišti. Pokud má letadlo nadměrnou výšku a vyšší rychlost než bylo vypočítáno, je energy metr zobrazen s plným zeleným sloupcem nad ideální energií, zobrazenou bílou ryskou. Pokud bude mít letadlo menší rychlost, případně menší výšku než plánovanou, tak bude energy metr ukazovat menší sloupec energie zbarvený do červena, viz obrázek 2.3. Je to jistě zajímavá funkce aplikace, ovšem nutno dodat, že přebytečnou energii letadlo vytrácí pomocí vztlakových klapek. Vztlakové klapky ovšem nemají některá ULL. V takovém případě by letadlo, které se drží tohoto přístroje, bylo na přistání s největší pravděpodobností dlouhé [25].

2.1.4 Mapy a lokalizace

Letecký přístroj VP-400 používá informace o letištích od firmy Seattle Avionics [25]. Jsou to velice kvalitní data, která obsahují nejenom přesné GPS souřadnice všech krajních bodů VPD, ale i informace o přiblížení na přistání, letecké IFR koridory. Přístroj VP-400 zobrazuje 3D mapu se zobrazeným terénem a s vykreslením vodních ploch včetně řek. V 3D mapě jsou zakresleny jednotlivá letiště a jejich VPD.

Přístroje společnosti Vertical Power jsou určeny zejména pro americký trh, je ale možné přikoupit si databázi letišť Evropy. Při srovnání s veřejně dostupnou českou databází letišť chybí v mapách od Seattle Avionics velká část přistávacích ploch, zejména ploch bez veřejného provozu a malých ULL letišť. Databáze firmy Seattle Avionics je aktualizována každý měsíc [18].

Aplikace Xavion používá informace o letištích ze simulátoru X-Plane. V této databázi jsou vedeny letiště z celého světa. Nutno však poznamenat, že letišť je zde zaznamenáno méně než v Seattle Avionics databázi. Informace o letištních plochách, jak výrobce X-Plane databáze uvádí, mohou být zastaralé a je vhodné si je ověřovat [25].

2.2 Sky Demon

2.2.1 Základní údaje

Mobilní aplikace SkyDemon je určená pro plánování a navigaci VFR (Visual Flight Rules) letů. Aplikaci SkyDemon vyvíjí tým programátorů ve Velké Británii. Aplikace je dostupná na trhu od roku 2009. Aplikaci je možné nainstalovat na celou řadu operačních systémů. Podporované jsou především mobilní operační systémy Android, iOS a Windows CE [19]. Je také možné systém nainstalovat na PC, laptop nebo tablet s operačním systémem Windows.

Cena aplikace SkyDemon je 210 €. Za tuto částku dostane uživatel od výrobce licenci, se kterou může používat dvě zařízení iOS, dvě zařízení Android

a dvě zařízení s operačním systémem Windows. Za používání aplikace je nutné platit roční poplatek aplikace ve výši 104 € [19].

2.2.2 Mapy a lokalizace

Výrobce umožňuje použití aplikace v Evropě a v USA. Veškeré mapy aplikace je možné stáhnout zdarma předem do aplikace. Mapy jsou spravovány výrobcem aplikace. Výrobce neuvádí, jak často mapy aktualizuje. Pro stažení mapy je vždy nutné zvolit typ mapy a vybranou zemi.

Aplikace využívá veřejně dostupných leteckých informací TAF (Terminal Aerodrome Forecast) a NOTAM (Notice To Airmen) a jiných. Informace o počasí TAF jsou podrobně zobrazeny v mapě, ale je možné si tyto informace podrobně zobrazit na určené obrazovce. Informace NOTAM, které upozorňují pilota na možné nebezpečí a omezení provozu, jsou důkladně zobrazeny na mapě. Aplikace zobrazuje na mapě letové prostory. Není nutné si prostory ani mapy stahovat. Aplikace si veškeré potřebné informace stáhne automaticky sama z dostupného internetového připojení. Výrobce ovšem uvádí možnost špatné kompatibility s informacemi NOTAM. Proto je možné vložit ručně textovou informaci NOTAM do aplikace [19].

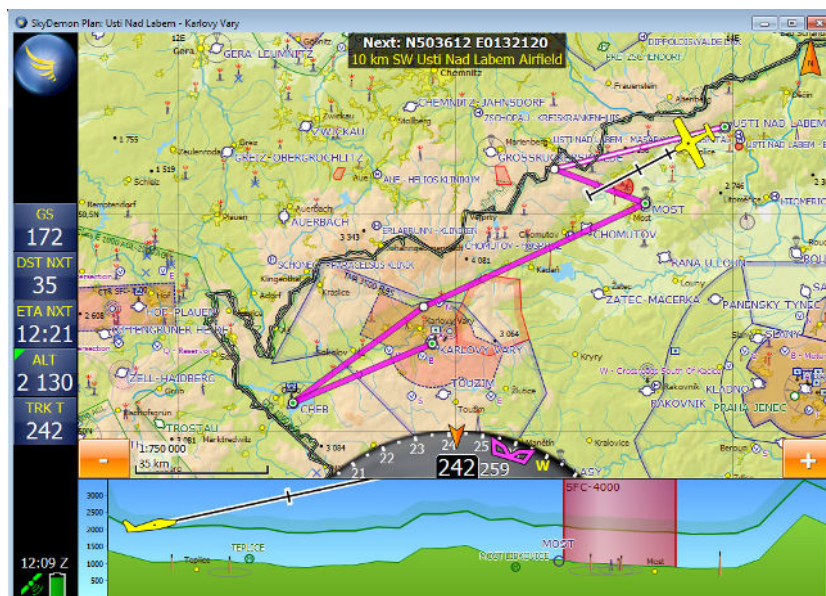
2.2.3 GUI a funkce aplikace

Aplikace Sky Demon je nejlépe hodnocená mobilní aplikace ve své kategorii navigace VFR. Funkce aplikace jsou více než dostačující i pro náročného pilota, přitom je zachována velká jednoduchost zobrazení i ovládání. Aplikace podporuje zobrazení aktuálních leteckých informací, a proto je téměř nutností, aby bylo zařízení připojené za běhu na internet.

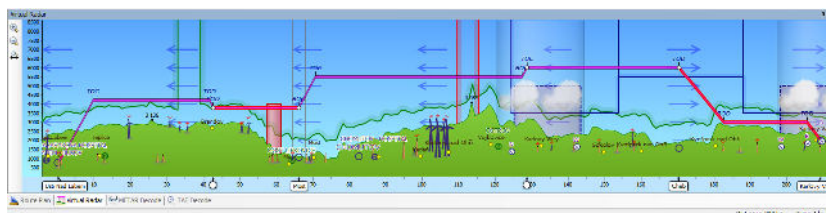
Hlavní částí GUI je 2D pozemní mapa, která je skromně zbarvena. Díky tomu je možné přehledně a snadně vyčíst nad vrstvou mapy další zobrazené prvky. Aplikace zobrazuje letecké prostory, letiště, leteckou trasu a další důležité informace pro let. Ovládání aplikace je intuitivní a není zapotřebí číst manuál.

Aplikace SkyDemon umožňuje snadné vytvoření letecké trasy. Při vytváření trasy je uživatel upozorněn aplikací na možné nebezpečí za letu. Pokud uživatel aplikace akceptuje upozornění, může aplikace automaticky trasu upravit.

Zajímavou částí GUI je zobrazený vertikální profil trasy ve spodní části obrazovky, viz obrázek 2.5. Z profilu trasy můžeme jednoduše vyčíst výšku terénu, výškové překážky, města a významné body v průběhu plánovaného letu. Nad terénem nám aplikace umožňuje zobrazit letecké prostory a také informace o větru a oblačnosti. V případě, že je plánovaný let nízko nad terénem aplikace označí ve vertikálním profilu trasy kritický úsek červeně. Výška plánované trasy letu jde snadno zvýšit kliknutím na červený úsek a tažením nahoru.



Obrázek 2.4: Screenshot aplikace Sky Demon za letu



Obrázek 2.5: Vertikální profil trasy aplikace Sky Demon

2.2.4 Nouzová navigace na letiště

Aplikace SkyDemon neumožňuje nouzovou navigaci na letiště. Je ale možné vyhledat si ručně nejbližší letiště a nechat se na něj navigovat pomocí funkce „direct to“. Trasa není počítaná se zatáčkami.

2.3 Air Navigation Pro

2.3.1 Základní údaje

Aplikace Air Navigation Pro se objevila poprvé na trhu v roce 2009. Aplikaci vytvořil a spravuje ji pan Jon Richardson [35]. Aplikaci je možné používat spolu s operačním systémem Android a iOS. Pro oba operační systémy stanovil výrobce stejnou cenu 49,99 EUR [10]. Aplikace Air Navigation Pro je zaměřená

na stejnou cílovou skupinu ULL a SLZ stejně jako aplikace SkyDemon.

Aplikace Air Navigation Pro je určena pro plánování letu v prostorech VFR a IFR (Instrument Flight Rules). Navigace využívá GPS technologie a integrovaného akcelerometru. GPS senzor je také možné připojit k zařízení externě např. pomocí bezdrátové technologie Bluetooth. Aplikace umožňuje naplánovat si a uložit leteckou trasu, pomocí které aplikace naviguje [35].

2.3.2 Mapy a lokalizace

Do aplikace je možné si stáhnout podkladové mapy Open Street maps z celého světa zdarma. Aplikace také obsahuje celosvětovou databázi letišť. Databáze obsahuje i letecké prostory mnoha zemí. Spolu s koupí aplikace uživatel dostane jednu ICAO mapu a jednu 3D mapu vybrané země. Většina map, které umožňuje aplikace stáhnout je placená. Je ale nutno říci, že jsou naprosto dostačující mapy bez poplatku, které aplikace doplní o veřejně dostupné letecké informace.

2.3.3 GUI a funkce aplikace

Air Navigation Pro umožňuje zobrazení pohyblivé 2D mapy shora se zobrazením plánované trasy, letecky významných prostorů a letišť [37]. Mapu lze přibližovat, posouvat a centrovat. Směrování mapy je možné nastavit na sever nebo ve směru letu. Uživateli je umožněno zobrazit si vedle mapy další zvolené informace nebo zobrazení napodobených leteckých přístrojů. Zajímavá je možnost zobrazení výškového profilu ve směrovaném letu. Pokud jsou v aplikaci stažené 3D mapy dané oblasti, umožňuje aplikace ve výškovém profilu zobrazit nejenom výšku plánované trasy, ale i terénní nerovnosti a jména měst na trase letu, viz obrázek 2.6. Pokud je trasa vedena přes vyšší hory, je automaticky plánováno stoupání letadla s dostatečným předstihem [35].

Vytvořit plánovanou trasu lze několika intuitivními způsoby. Je možné přidávat navigační body dotykem na význačná místa v mapě, ale lze i zadávat body přímo ze seznamu [37]. Aplikace počítá a umožňuje zobrazit časy ETA, ETE (Estimated Time Enroute)[35]. Samozřejmostí je možnost zobrazení aktuální rychlosti vůči zemi, nadmořskou výšku, kurz směru letu a ideální kurz na další otočný bod. V aplikaci je možné nechat zobrazit napodobené letecké přístroje VOR, rychloměr, výškoměr a kompas. Zajímavostí aplikace je možnost přepnutí do 3D módu, ve kterém je dobře vidět náklon letadla. Před použitím 3D módu je vhodné stáhnout do aplikace 3D mapy. V 3D mapě je možné zapnout tzv. Terrain Awareness [35]. Tato funkce obarvuje vrcholky hor nad úrovní letadla červeně. Pro bezpečnost letu je možné nechat si zobrazit informace o počasí, je ale nutné být připojený online na internet. Pomocí gest je možné psát prsty přímo na obrazovku s mapou, tato funkce je vhodná např. pro psaní důležitých vysílacích frekvencí letišť. Aplikace Air Navigation Pro umožňuje také nahrávat pozici na server online tzv. livetracking [35].



Obrázek 2.6: Ukázka aplikace Air Navigation Pro za letu [36]



Obrázek 2.7: 3D pohled aplikace Air Navigation Pro [36]

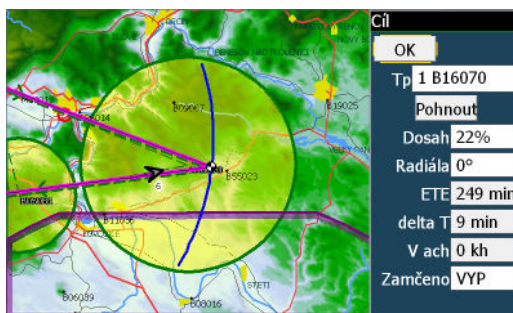
2.3.4 Nouzová navigace na letišti

Aplikace Air Navigation Pro podporuje pouze navigaci letadel a nepodporuje nouzové situace klouzavého letu letadla. Je ale možné si v aplikaci vybrat nejbližší nebo jiné letiště ručně a nechat se směrově navigovat na letiště.

2.4 LK8000 – Tactical Flight Computer

2.4.1 Základní údaje

LK8000 je software vytvořený zejména pro piloty bezmotorových sportovních letacích zařízení. Mezi ně patří piloti větroňů, rogalisté a paraglidisté. LK8000 je open source projekt, který spravuje skupina nadšenců bezmotorového létání. Software je možné nainstalovat pouze na zařízení s operačním systémem Windows CE a Windows 32 [14]. V praxi se nejlépe osvědčily navigace do auta PNA různých výrobců a také mobilní telefony PDA. Software LK8000 je zcela zdarma ke stažení přímo na oficiálních stránkách aplikace.



Obrázek 2.8: Aplikace LK8000 - letový režim [16]

2.4.2 Mapy a lokalizace

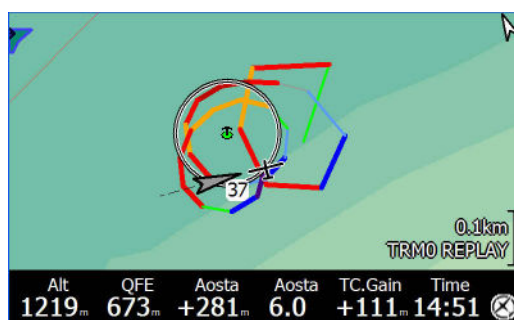
Aplikace je přeložena do 17 jazyků včetně českého [13]. LK8000 používá vlastní topologickou mapu s příponou LKM. K topologické mapě má uživatel možnost přidat soubor využívající standardu DEM (Digital Elevation Model). Soubor s příponou DEM obsahuje informace o terénních nerovnostech povrchu země. [24].

2.4.3 GUI a funkce aplikace

LK8000 je sofistikovaná aplikace umožňující vzdušnou navigaci zaměřující se na efektivnost klouzavého letu. Tuto aplikaci využijí hlavně sportovní nebo závodní piloti, kteří chtějí maximálně využít potenciál letového dne. Aplikace pracuje ve třech módech letu: klouzavý let, kroužení v termice a dokluz do

cíle [24]. Pro každý režim zobrazuje aplikace jiné údaje, důležité pro aktuální situaci. Aplikace LK8000 umožňuje zobrazit letecké prostory. Při přiblížení k zakázanému leteckému prostoru aplikace pilota upozorní.

Pro správné výpočty opadání letadla je nutné do aplikace nahrát údaje o poláře kluzáku [24]. Z těchto údajů je aplikace schopna vypočítat nejefektivnější vzdušnou rychlost. V módu kroužení aplikace počítá z jednotlivých posunutých zatáček rychlost větru. Tento výpočet může být v určitých situacích značně nepřesný, a proto umožňuje aplikace i ruční zadání rychlosti a směru větru. V praxi je ale tento výpočet dostačující a při delším točení v termice se výpočet stále zpřesňuje. V režimu kroužení, nebo také stoupání, zobrazuje aplikace pravděpodobný střed stoupavého proudu. Okolo středu zobrazeného stoupavého proudu aplikace vykreslí kružnici vhodnou pro zatáčení [24].



Obrázek 2.9: Aplikace LK8000 - mód kroužení v termickém proudu [7]

Aplikace LK8000 je navržena podle originálního konceptu *turbulence proof*, který by měl usnadňovat ovládání aplikace při turbulenci v letadle. Aplikace obsahuje minimum tlačítek. Na místo nich jsou na obrazovce určeny prostory pro kliknutí. Pro zoomování jsou zde určeny dvě oblasti. Tyto oblasti tvoří většinu z možného prostoru obrazovky. Pro přiblížení mapy je nutné kliknout do horní části obrazovky na mapu. Naopak pro oddálení mapy je zapotřebí kliknout do spodní části mapy [24]. Další, menší prostory pro kliknutí, jsou na krajích obrazovky. Je možné si zvolit, která akce se po kliknutí na určitou část obrazovky vykoná. Komunita vývojářů uvádí, že je tento koncept dobře využitelný a zmenšuje pravděpodobnost překlepu při turbulenci. Koncept *turbulence proof* je zajisté zajímavý nápad, nutno ovšem dodat, že pro uživatele není ovládání zcela intuitivní a je nutné si aplikaci před prvním použitím prostudovat.

2.4.4 Nouzová navigace na letiště

Aplikace LK8000 obsahuje pouze navigaci na konkrétní bod bez 3D plánování trajektorie trasy. Umožňuje tedy nouzovou navigaci v případě, že uživatel

v menu najde nejbližší letiště a nechá se na něj navigovat. V takovém případě bude pilot muset rozpočet na přistání a jednotlivé zatáčky vymyslet sám.

2.5 Srovnání jednotlivých aplikací

	VP-400	Xavion	Sky Demon	Air-Nav Pro	LK8000
Cena aplikace	8000 \$	99 \$	210 €	50 €	zdarma
Lokalizace	USA, EU	USA, EU	ESA, Evropa	World	World
Operační systém	Přístroj	iOS	Android, iOS	Android, iOS	Windows
Nouzová navigace	ano	ano	ne	ne	ne
Dokáže přístroj řídit letadlo	ano	ne	ne	ne	ne
Navigace na nejbližší letiště	ano	ano	ano	ano	ano
3D pohled aplikace	ano	ano	ne	ano	ne
Plánování trajektorie letu 3D	ano	ano	ne	ne	ne

Tabulka 2.1: Porovnání jednotlivých aplikací

Každá zmíněná aplikace má mírně odlišnou cílovou skupinu uživatelů a právě proto byly rozebrány v rešerši. Z rozebíraných programů by bylo vhodné inspirovat se klady jednotlivých aplikací. Jediné veřejně dostupné aplikace, které dokáží nouzově navigovat letadlo v 3D projekci letu jsou aplikace VP-400 a Xavion od společnosti Vertical Power.

Návrh

3.1 Požadavky na aplikaci

Cílem této práce je vytvoření prototypové FMS aplikace, která dokáže nouzově navigovat ultralehké letadlo na přistání. Aplikace spočítá pilotovi trajektorii letu, která umožní v klouzavém bezmotorovém letu přistát na nejbližší letiště v požadovaném směru.

3.1.1 Funkční požadavky

3.1.1.1 Požadavky na aplikaci

- FP01** Aplikace bude zobrazovat podkladovou mapu.
- FP02** Aplikace bude zobrazovat aktuální pozici letadla z GPS senzoru.
- FP03** Aplikace bude umožňovat vypočítat trajektorii pro nouzové přistání.
- FP04** Aplikace bude umožňovat vizualizaci trajektorie do podkladové mapy.
- FP05** Aplikace bude umožňovat nastavit parametry výpočtu trajektorie.
- FP06** Aplikace bude umožňovat nastavit základní parametry letounu.
- FP07** Aplikaci bude možné propojit se simulátorem tak, aby bylo možné snadno otestovat funkčnost aplikace.
- FP08** Aplikace bude podporovat navigaci podél spočítané trajektorie.
- FP09** Aplikace bude zobrazovat aktuální rychlost letadla.
- FP10** Aplikace umožní přepočet trajektorie v průběhu nouzové situace.

3.1.1.2 Požadavky na vypočítanou trasu

- FP11** Vypočtená trajektorie bude složena z jednotlivých zatáček a přímého letu ve 3D prostoru.
- FP12** Výpočet bude zahrnovat do nouzové trasy letadla vítr.
- FP13** Klesání letadla ve vypočítané trajektorii bude vycházet z reálné situace.

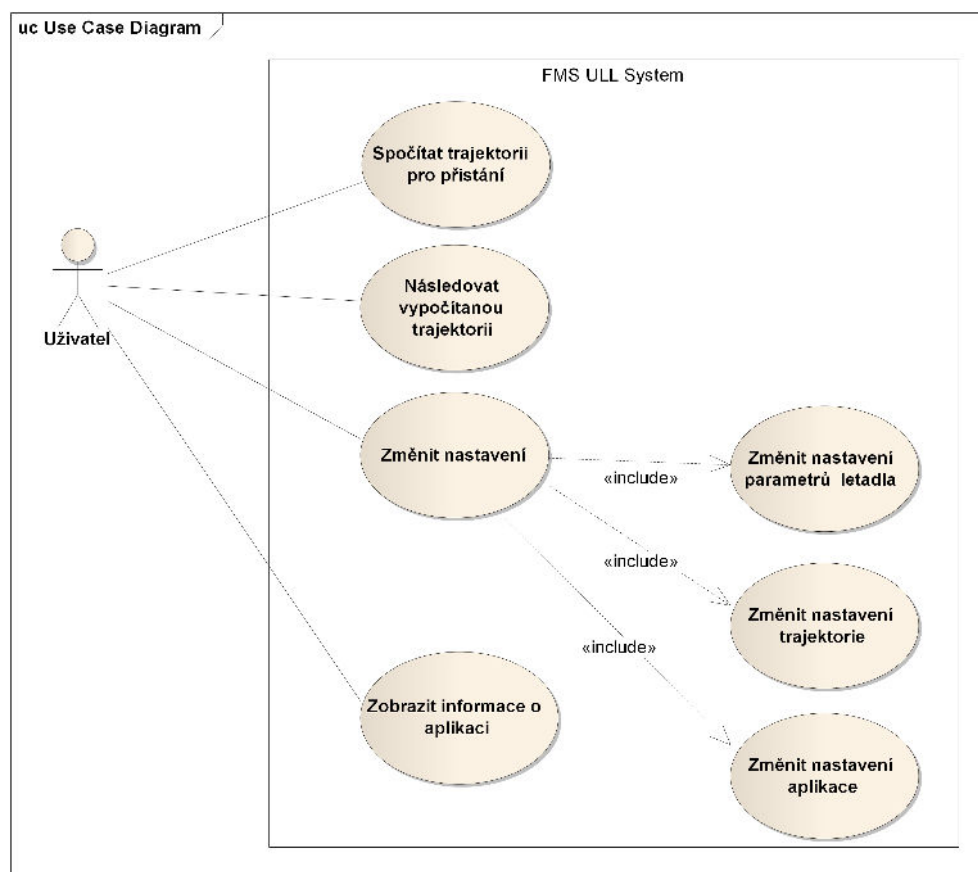
3.1.2 Nefunkční požadavky

NP01 Aplikace bude schopna spočítat trajektorii dostatečně rychle tak, aby výpočet neomezil bezpečnost přistání.

NP02 Aplikace bude složena z modulů tak, aby bylo možné modul spočtení nouzové trajektorie využít v jiných aplikacích.

3.2 Případy užití

Pomocí případů užití jsou zachyceny požadavky na FMS systém. Pro aplikaci je navržena jedna uživatelská role, které jsou umožněny následující interakce se systémem, viz obrázek 3.1.



Obrázek 3.1: Diagram případů užití

3.3 Návrh modulů a komunikačních API

Jednotlivé moduly systému by měly oddělovat logicky fungující celky tak, aby bylo možné tyto moduly použít v jiné aplikaci, případně je nahradit jinými. K tomu je nezbytné navrhnout komunikační API (Application Programming Interface) mezi moduly a správné rozvržení aplikace.

3.3.1 Popis návrhu

Hlavním modulem, který celou aplikaci zobrazuje a interaguje s uživatelem bude modul nazvaný *Client*. Modul *Client* bude platformě závislý a umožní komunikaci uživatele s celým systémem.

Nejobsáhlejším modulem aplikace bude *EmergencyLanding*. Tento modul bude vypočítávat správnou trajektorii pro přistání a pomocí rozhraní *INavigatable* bude poskytovat modulu *Client* jednotlivé body ve 3D prostoru, které modul vypočítal. Tyto body budou poskytovány dvěma metodami *getIGCPoints* a *getPoints*. Před voláním metod, které vracejí body je nutné zavolat metodu *computeRoute*, která dané body vytvoří. Metoda *getIGCPoints* bude vracet body, jejichž rozmístění je inspirováno IGC (International Gliding Commission) standardem. Body vrácené touto metodou jsou od sebe vzdáleny na určitý počet sekund předpokládané rychlosti letu. Metoda *getPoints* bude vracet pouze body, které jsou nutné k vytvoření dostatečně hladké trasy. Což zjednodušeně řečeno znamená, že pokud bude v trase část rovného letu, bude určena pouze dvěma body. Rozhraní *INavigatable* definuje další metody *setRoutePoints* a *deleteRoutePoints*, které jsou navrženy za účelem budoucího rozšíření modulem pro počítání jiné trajektorie.

Modulem, který bude poskytovat pozici a zajišťovat její aktualizaci, je modul *Position*. Tento modul bude komunikovat pomocí rozhraní *IPositionService*. Modul *Position* realizují dva moduly, které lze mezi sebou přepnout. Dostáváme tedy GPS souřadnice nebo simulované GPS souřadnice pomocí protokolu NMEA, viz obrázek 3.2.

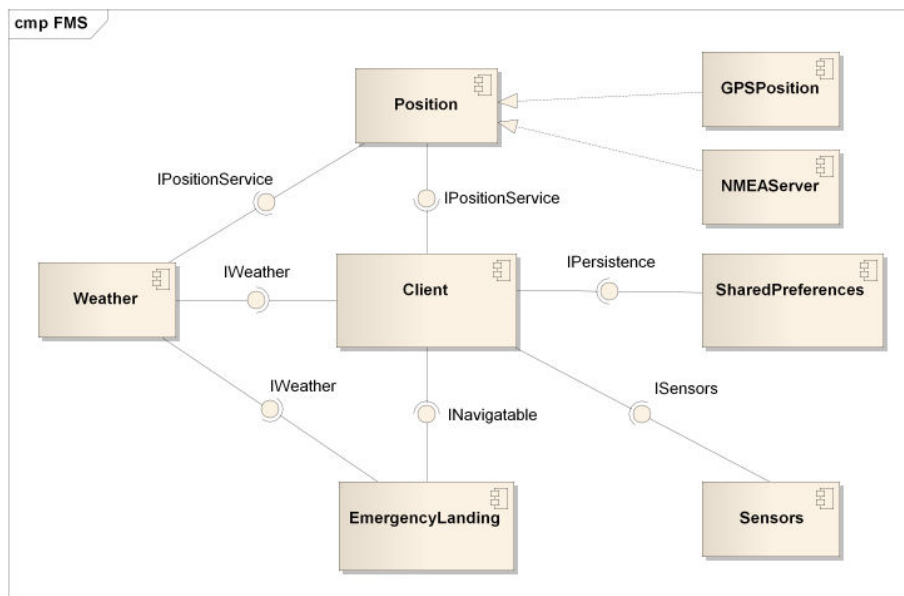
Modul, který se stará o načtení a uložení parametrů aplikace, je modul *SharedPreferences*. Modul umožňuje stejné nastavení aplikace bez uživatelské interakce z předchozího běhu. Modul komunikuje pomocí rozhraní *IPersistence*, které obsahuje metody *loadParameters* a *saveParameters*.

Dalším navrženým modulem je modul *Sensors*, který poskytuje informace o jednotlivých senzorech pomocí rozhraní *ISensors*. Rozhraní umožňuje získat třídu *Sensor* díky metodě *getSensor*, která potřebuje jako parametr textový identifikátor konkrétního senzoru.

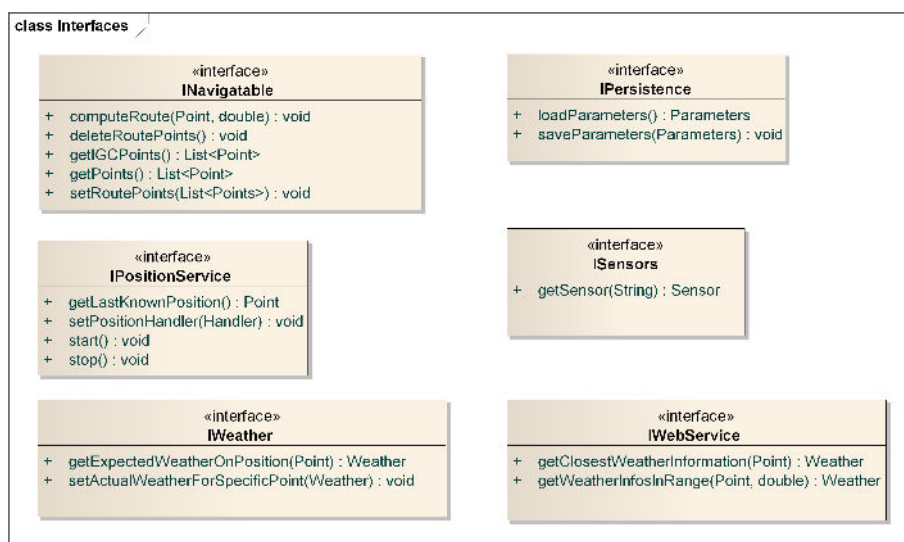
Aplikace je navržena také s modulem *Weather*, který by měl aplikaci poskytovat informace o počasí. Rozhraní umožňující tuto komunikaci se nazývá *IWeather*. Rozhraní *IWeather* obsahuje metody *getClosestWeatherInformation* a *getWeatherInfosInRange*. Metoda *getClosestWeatherInformation* najde informace o počasí z nejbližší meteostanice. Metoda *getWeatherInfosInRange*

3. NÁVRH

najde informace o počasí z meteostanic, které jsou v okolí aktuální pozice do maximální vzdálenosti, kterou určuje parametr *range*. Modul *Weather* se bude starat o načítání informací o počasí v okolí aktuální pozice. Proto je také vazba mezi modulem *Weather* a *Position*.



Obrázek 3.2: Komponent diagram aplikace



Obrázek 3.3: Navržené rozhraní pro komunikaci mezi moduly aplikace

3.4 Výběr platformy

Cílovou skupinou uživatelů aplikace jsou piloti ultralehkých letadel v České a Slovenské republice. Některá ULL mají v kokpitu přístrojové vybavení EFIS (Electronic Flight Information System). V České a Slovenské republice je letadel s EFIS vybavením méně než 10% [27]. I proto jsou v dnešní době aplikace pro malá letadla stavěné na mobilních operačních systémech. Mezi často používané systémy patří iOS, Android, Windows CE a Windows Phone.

Každý pilot je zvyklý na svoje zařízení, ve kterém již většinou má nainstalovanou aplikaci pro leteckou navigaci. Podle nejnovějších článků časopisu Pilot používají piloti ULL především zařízení iOS a Android [10]. Celosvětové statistiky prodejců mobilních zařízení říkají, že nejčastější mobilní platformou je operační systém Android. Velmi podobná je i bilance prodeje Android zařízení v České republice [22].

3.4.1 Android

Android je rozsáhlá open source platforma, která vznikla zejména pro mobilní zařízení, mezi něž patří chytré telefony, PDA, navigace a tablety. Platforma Android v sobě zahrnuje operační systém založený na jádru Linux. Vyvíjí ho konsorcium Open Handset Alliance, jehož cílem je progresivní rozvoj mobilních technologií, které budou mít výrazně nižší náklady na vývoj a distribuci a zároveň spotřebitelům přinese inovativní uživatelsky přívětivé prostředí [28].

Společnost Android Inc., která vyvíjí systém Android, byla založena v roce 2003. V roce 2005 společnost Google Inc. odkoupila v té době nepříliš známou „startup“ firmu Android Inc. a udělal z ní svoji dceřinou společnost [3].

Android se vyznačuje snadno rozšiřitelným programovacím jazykem Java, díky čemuž jsou aplikace pro Android snadněji přenositelné a dají se vyvíjet na různých operačních systémech [21]. Nevýhodou je naopak složitější testování na více zařízeních s rozdílnými vlastnostmi.

3.4.2 iOS

Platforma iOS obsahuje mobilní operační systém vytvořený společností Apple Inc. Původně byl určen pouze pro mobilní telefony iPhone, později se však začal používat i na dalších mobilních zařízeních společnosti Apple, jako jsou iPod Touch, iPad a nejnověji Apple TV [30].

Operační systém iOS vychází ze systému Mac OS X. Architektura iOS je v podstatě UNIX, který je zapouzdřen do několika vrstev pro zachování abstrakce celého vývoje. iOS také poskytuje unifikované uživatelské rozhraní oproti konkurenci. Pro vývoj aplikací pro iOS je třeba zaplatit minimální částku 99\$ na rok [30]. Vyvíjet aplikace je možné v jazyce C++ a to pouze na operačním systému Mac OS X.

3.4.3 Výběr platformy

Z důvodu vyššího zastoupení na trhu i z důvodu jednodušší implementace byla pro implementaci aplikace zvolena platforma Android.

3.5 Návrh uživatelského rozhraní

Aplikace implementované pro operační systém Android jsou postaveny ze stavebních bloků uživatelských rozhraní, jimiž jsou aktivity. Aktivitu si můžeme představit jako entitu systému Android analogickou k oknu nebo dialogu klasické desktopové aplikace pro počítače [1]. Aktivita je v podstatě programová entita, se kterou může interagovat uživatel. Aktivity většinou mají GUI, není to však povinná vlastnost aktivity.

3.5.1 Hlavní aktivita

Při startu aplikace se spustí hlavní aktivita. Základem hlavní aktivity bude podkladová mapa usnadňující orientaci pilota. Podkladová mapa bude zobrazena ze shora a nebude umožňovat natočení v 3D prostoru. Ikona letadla bude zobrazena na středu obrazovky přes podkladovou mapu. Při změně kurzu nebo polohy letadla bude ikona letadla stále na stejném místě. Posunutí a otáčení bude implementováno na samotné mapě. Bylo by vhodné zvýraznit, nebo jinak označit sever mapy, který se bude během letu měnit vůči kurzu letadla.

Mapa nebude umožňovat uživatelské akce jako posunutí mapy, natočení mapy apod. Pro možnost zobrazení celé plánované cesty bude umožněno měnit přiblížení mapy.

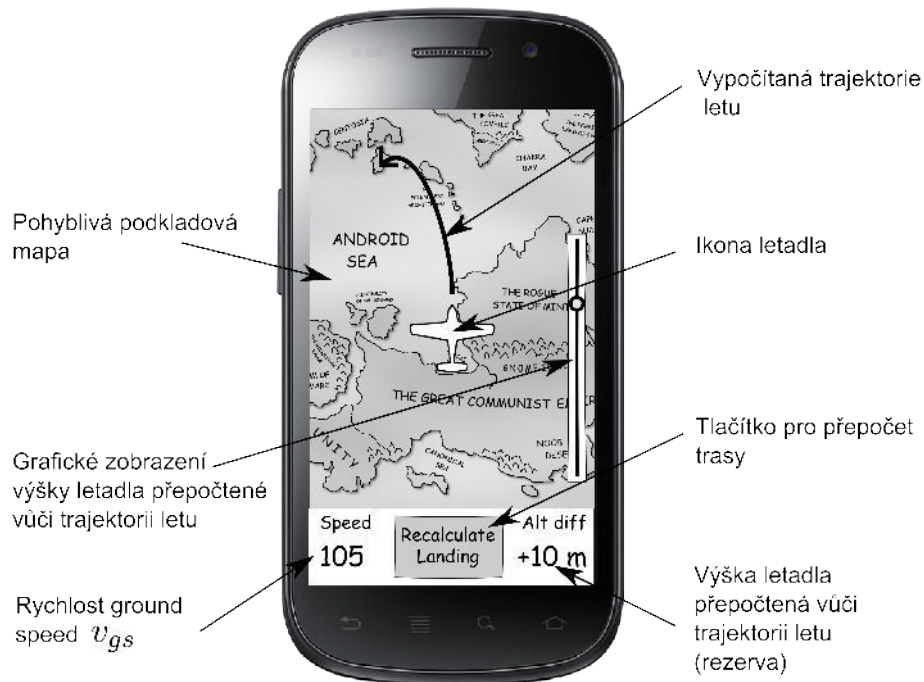
Aplikace bude zobrazovat výšku v metrech vztaženou k trajektorii letu. Pokud budeme uvažovat let v blízkosti řádově v desítkách metrů od vypočítané trasy, bude kladné číslo výškoměru znamenat, že jsme nad ideální trajektorií letu. Naopak pokud bude číslo záporné, bude se letadlo nacházet výškově pod ideální trajektorií letu. Výška přepočítaná vůči ideální trajektorii v metrech bude zobrazena v pravém dolním rohu aplikace.

Aplikace bude zobrazovat grafický výškoměr, který bude vizualizovat výšku letadla vztaženou k trajektorii letu, viz obrázek 3.4. Tento grafický prvek by měl být snáze viditelný v případě, že uživatel sleduje polohu letadla na mapě namísto spodní informační lišty aplikace.

Pro přehlednost a bezpečnost bude zobrazovat aplikace rychlost letu vůči zemi v_{gs} . To znamená, že pokud bude foukat vítr 20 km/h proti směru letu, bude zobrazené číslo letu o 20 km/h nižší než reálná rychlost letu vůči vzduchu v_{tas} .

Tlačítko *Recalculate landing* bude sloužit pro výpočet nebo přepočet trasy na přistání. Tlačítko nebude zobrazené v případě, že aplikace nemá známou polohu letadla. Tato situace může nastat v případě, že je například vypnutý senzor GPS, nebo senzor GPS stále hledá dostatečné množství satelitů pro

svou lokalizaci. Po stisknutí tlačítka na podkladové mapě bude zobrazena vypočítaná trasa pro přistání. Pilot by měl držet správný kurz podél spočítané trajektorie a také by měl udržovat výšku vůči spočítané trase.



Obrázek 3.4: Navržené GUI

3.5.2 Aktivita nastavení parametrů

Uživateli bude umožněno upravit parametry výpočtu a aplikace. Pro nastavení parametrů bude vytvořena další aktivita. Aktivita bude umožňovat měnit nastavení jednotlivých parametrů, viz obrázek 3.5. Z hlavní aktivity bude umožněno přepnout na aktivitu nastavení parametrů pomocí systémového tlačítka menu a výběru položky *Settings*.

V aktivitě nastavení parametrů bude umožněno uživateli měnit parametry výpočtu nouzové trasy pro přistání. Pro simulování GPS pozice bude aplikace umožňovat síťovou komunikaci pomocí protokolu NMEA 0183, viz oddíl 3.6. Aktivita nastavení parametrů bude umožňovat nastavení čísla síťového portu. Jednotlivé parametry budou spadat do předem určeného rozsahu hodnot, což zamezí uživateli zadat špatný vstup pro výpočet.

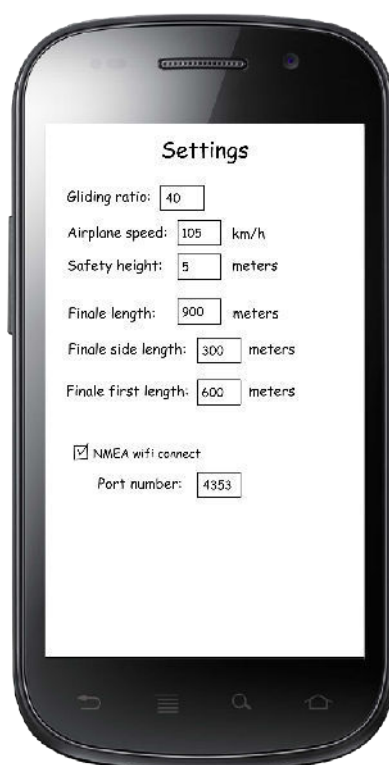
3.5.2.1 Popis hlavních parametrů aplikace

Parametr *gliding ratio*

Hodnota parametru *gliding ratio* specifikuje maximální klouzavost letadla. Hodnota maximální klouzavosti je běžně udávána v technických datech letounu. Hodnotu je možné zadávat v rozsahu 4 – 70. Hodnota klouzavosti nemá specifikovanou jednotku, protože je to pouze poměrové číslo, viz oddíl 1.1.8.

Parametr *airplane speed*

Hodnota parametru *airplane speed* udává vzdušnou rychlost v_{ias} letadla, při které letoun dosahuje maximální klouzavosti, viz oddíl refsubsec:polar.



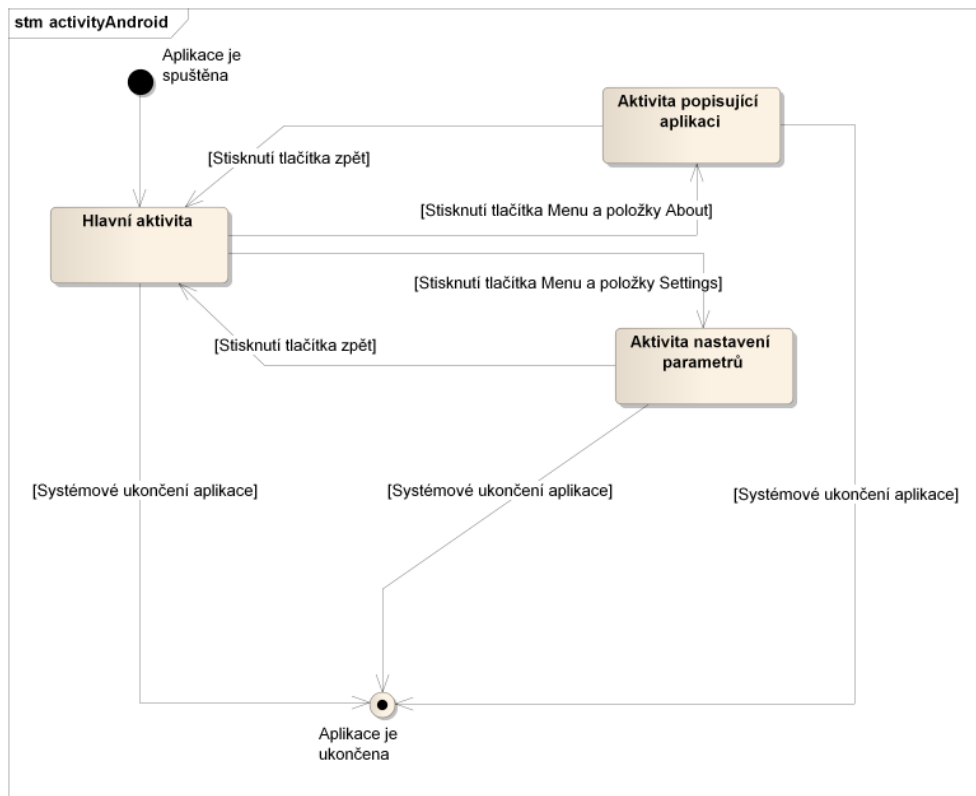
Obrázek 3.5: Navržené GUI pro nastavení parametrů

3.5.3 Aktivita popisující aplikaci

Pro zobrazení informací, které popisují aplikaci, byla navržena aktivita *About-Activity*. Aktivita popisující aplikaci zobrazuje text, který informuje uživatele o správném použití aplikace.

3.5.4 Propojení jednotlivých aktivit

Jednotlivé aktivity jsou navzájem propojeny. Navigace mezi aktivitami je navržena s využitím systémových tlačítek. Za použití systémového tlačítka menu v hlavní aktivitě, je možné navigovat do dalších aktivit. Pro navigaci aplikace zpět na hlavní aktivitu bude sloužit systémové tlačítko zpět, viz obrázek 3.6.



Obrázek 3.6: Propojení jednotlivých aktivit

3.6 NMEA komunikační protokol

The National Marine Electronics Association (NMEA) je nezisková organizace, která se zabývá certifikací elektronických zařízení určených pro námořní využití. NMEA 0183 je komunikační protokol, který definuje datovou komunikaci mezi námořní elektronikou jako jsou sonary, anemometry a GPS přijímače [33]. Tento protokol se běžně používá i v civilním světě např. při komunikaci mezi externím GPS přijímačem připojeným přes bluetooth k mobilnímu telefonu. Tento protokol se také často používá pro výstup z leteckých simulátorů, které jím nahrazují GPS přijímač. NMEA 0183 používá sériovou, textově oriento-

vanou komunikaci [33]. Pro tuto diplomovou práci postačí ukázat využívané NMEA zprávy a návrh komunikace.

3.6.1 Návrh komunikace pomocí NMEA 0183

Vyvíjená aplikace bude umožňovat získávání zeměpisných souřadnic ve dvou režimech. První režim využije GPS senzor vestavěný v mobilním zařízení. Druhý režim umožní použití externího GPS zdroje, který bude posílat souřadnice pomocí protokolu NMEA 0183. Díky režimu NMEA je možné aplikaci připojit k leteckému simulátoru běžící na počítači.

NMEA komunikace podporuje TCP/IP protokol a je navržena tak, že GPS přijímač nebo letecký simulátor je komunikační klient [9]. Serverovou část bude implementovat mobilní aplikace. Serverová část aplikace bude naslouchat na určeném komunikačním portu. Klientská část, což je v našem případě simulátor, musí znát IP adresu mobilního telefonu a určený komunikační port.

Aplikace bude přijímat ASCII zprávy od klienta. Jednotlivé zprávy bude třídit podle typů zpráv. Ze specifikace [9] vyplývá, že pro naše potřeby jsou potřeba pouze dva typy zpráv a to jsou GPGGA (Global Positioning System Fix Data) a GPRMC (Recommended Minimum Specific GPS/TRANSIT Data). Formáty obou zpráv jsou důkladně popsány v příloze E. Z ASCII znaků jednotlivých zpráv jsou vyhledány příslušné údaje o pozici letadla a rychlosti. Údaje jsou posléze podstrčeny aplikaci jako GPS souřadnice.

3.7 Databáze letišť

Jako databáze přistávacích drah byla zvolena Československá databáze letišť, kterou spravuje pan Patrik Sainer [17]. Databáze je veřejně přístupná a obsahuje nejenom běžné letiště, ale i menší plochy, na kterých je možné přistát. Pro Českou a Slovenskou republiku je to nejpodrobnější databáze, která je volně přístupná. Databáze lze získat ve formátu xml, kde jsou podrobně popsány nejenom body přistávací dráhy, ale také například frekvence rádiového spojení.

Implementace

Navržená aplikace je napsána v jazyce Java. Projekt je rozdělen do balíčků (packages), kde rozdělení balíčků na nejvyšší úrovni odděluje jednotlivé moduly aplikace. V následujícím textu budou popsány jednotlivé implementované moduly aplikace. Vytvořená prototypová aplikace má implementovány pouze některé navržené moduly. Těmito moduly jsou *EmergencyLanding*, *Client*, *Position* a *SharedPreferences*. V následujícím textu budou tyto jednotlivé moduly podrobně rozebrány.

4.1 Objektový návrh

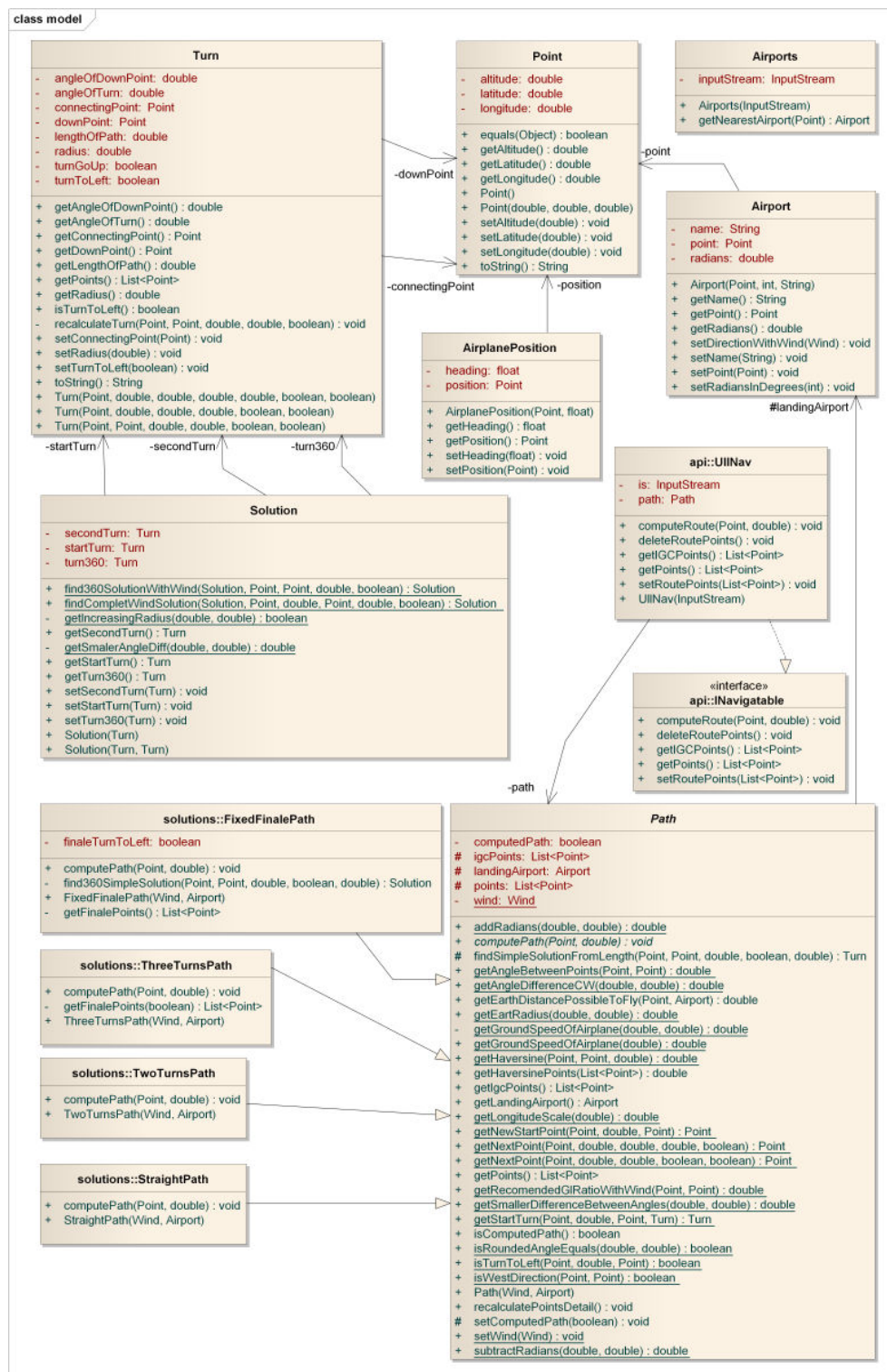
Vyvíjená aplikace je implementována podle architektonického vzoru Model-View-Controller. V modulu *Client* jsou implementovány části View a Controller. Část architektonického vzoru model je implementována v ostatních modulech.

4.2 Modul *EmergencyLanding*

Modul *EmergencyLanding* je nejrozsáhlejší vytvořený modul aplikace. Zajišťuje výpočet trajektorie pro nouzové přistání ULL. Je implementován tak, aby ho bylo možné použít i v jiných aplikacích. Na obrázku 4.1 je zobrazen diagram tříd nejdůležitější části modulu *EmergencyLanding*.

Modul *EmergencyLanding* umožňuje komunikaci s dalšími moduly pomocí rozhraní *INavigatable* poskytující kromě jiných také metodu *computeRoute*. Třída *UllNav* implementuje rozhraní *INavigatable* a uchovává si odkaz na třídu *Path*. Třída *Path* je abstraktní třída, od které dědí jednotlivá řešení *FixedFinalePath*, *ThreeTurnsPath*, *TwoTurnsPath* a *StraightPath*. Metoda *computeRoute* vybere vhodné řešení dané situace (oddíl 4.2.3) a zajistí výpočet jednotlivých bodů v trajektorii.

4. IMPLEMENTACE

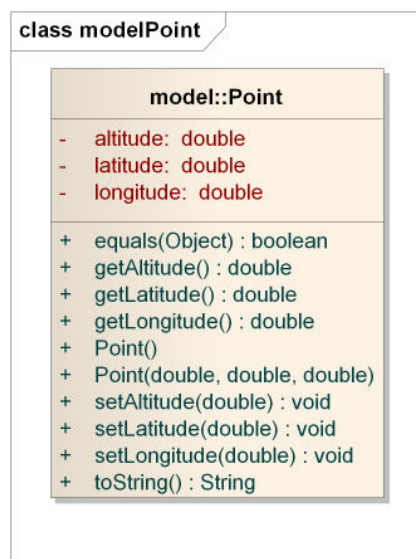


Obrázek 4.1: Class diagram hlavních tříd modulu EmergencyLanding

4.2.1 Datové typy

4.2.1.1 Bod ve 3D prostoru

Implementace aplikace by se neobešla bez třídy *Point*, kterou využívá výpočet aplikace i vizualizace trasy. Třída *Point* je hlavním stavebním blokem modulu *EmergencyLanding*, viz obrázek 4.2. Třída *point* obsahuje vlastnosti *longitude*, *latitude* a *altitude*. Vlastnosti třídy jednoznačně určují polohu na povrchu země a nadmořskou výšku konkrétního bodu.



Obrázek 4.2: Ukázka třídy *Point*

4.2.2 Výpočty

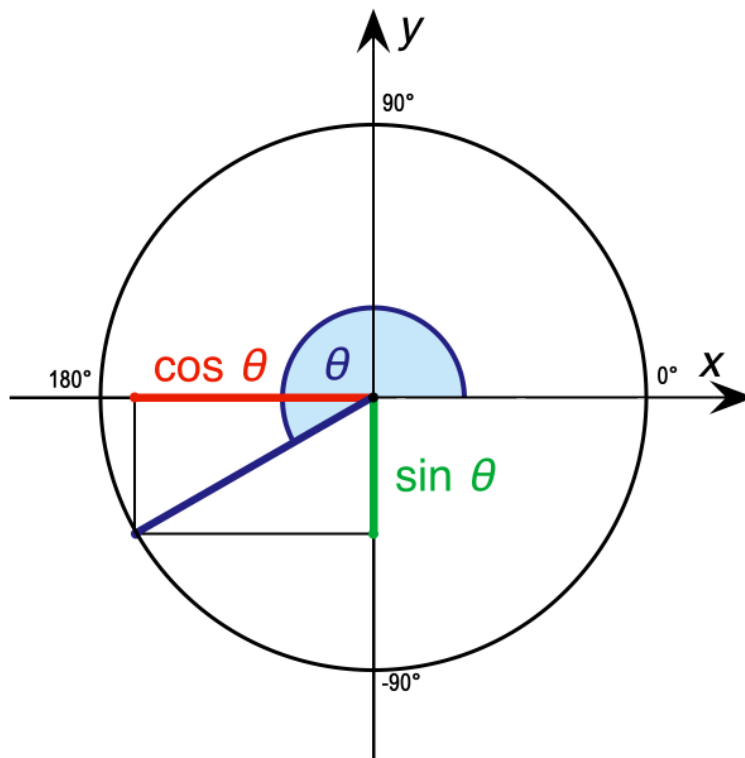
4.2.2.1 Počítání vzdálenosti

Pro výpočet trajektorie letu se ukázalo jako nezbytné počítat vzdálenost mezi dvěma body. Pro výpočet vzdálenosti je navržena metoda *getHaversine*, která vrátí vzdálenost v metrech mezi dvěma body. Metoda využívá vztahu definovaného jako Haversine formula [29]. Výpočet je zpřesněn tím, že používá různou hodnotu poloměru země v závislosti na zeměpisné šířce prvního bodu výpočtu. Navržená metoda *getHaversine* je obsažena v abstraktní třídě *Path*, která umožňuje základní výpočty nad body v prostoru.

4.2.2.2 Spočtení dalšího bodu

Základní metoda pro tvoření trajektorie letu je metoda *getNextPoint*, která je obsažena v abstraktní třídě *Path*. Metoda dokáže vytvořit nový bod, který vytvoří v určité vzdálenosti a směru od předchozího bodu. Metoda má na vstupu

předchozí bod, vzdálenost k dalšímu bodu, směr k dalšímu bodu, úhel klesání a informaci o tom, zda-li je to trasa počítaná ve směru letu nebo proti směru letu. Metoda v závislosti na směru letu a vzdálenosti k dalšímu bodu spočítá pomocí funkcí $\sin$ zeměpisnou šířku a pomocí funkce $\cos$ spočítá zeměpisnou délku, viz obrázek 4.3. Výšku dalšího bodu spočte metoda v závislosti na úhlu klesání letadla.



Obrázek 4.3: Využití funkcí $\sin$ a $\cos$ ke zjištění zeměpisných souřadnic dalšího počítaného bodu.

4.2.2.3 Výpočet klouzavosti v trajektorii

V případě, že se ULL dostane do nouzové situace, je nutno určit v jakém režimu letu poletí. Ve výpočtu trasy nerozlišujeme o jakou konkrétní nouzovou situaci se jedná. Může se jednat o nouzovou situaci výpadku motoru, či situaci špatné viditelnosti. Trajektorie letu vychází v obou případech z režimu nejlepší klouzavosti, viz oddíl 1.1.9. Číslo nejlepší klouzavosti letadla, je multiplikativní konstantou 0,85 zhoršena a posléze použita k výpočtům. Konstanta vyplynula z testování, viz oddíl 5.2.

Pro tvorbu zatáček bylo nutno určit opadání letadla v zatáčce. Podle zdroje [2] bylo spočítáno, že při běžných rychlostech ULL a při náklonu, který je

třeba držet v nejostřejších vytvořených zatáčkách, je celkové opadání v zatáčkách s poloměrem 200 metrů větší o 0,02 m/s. Minimální rychlost opadání v_o ULL se pohybuje okolo 1 m/s při dopředných rychlostech letounu v_d kolem 80 km/h. V navrhované trajektorii může být maximálně 540 stupňový obrat letadla s poloměrem 200m. V takovém krajním případě ztratí letadlo o 1,5 metru výšky víc, než při stejně dlouhé dráze letu rovným letem. Vzhledem k možnosti vlivu stoupavých a klesavých proudů nebudeme ve výpočtu trajektorie uvažovat zvýšené opadání v zatáčkách. V případě bezvětří uvažuje výpočet konstantní rychlost opadání letounu po celou vytvořenou trajektorii.

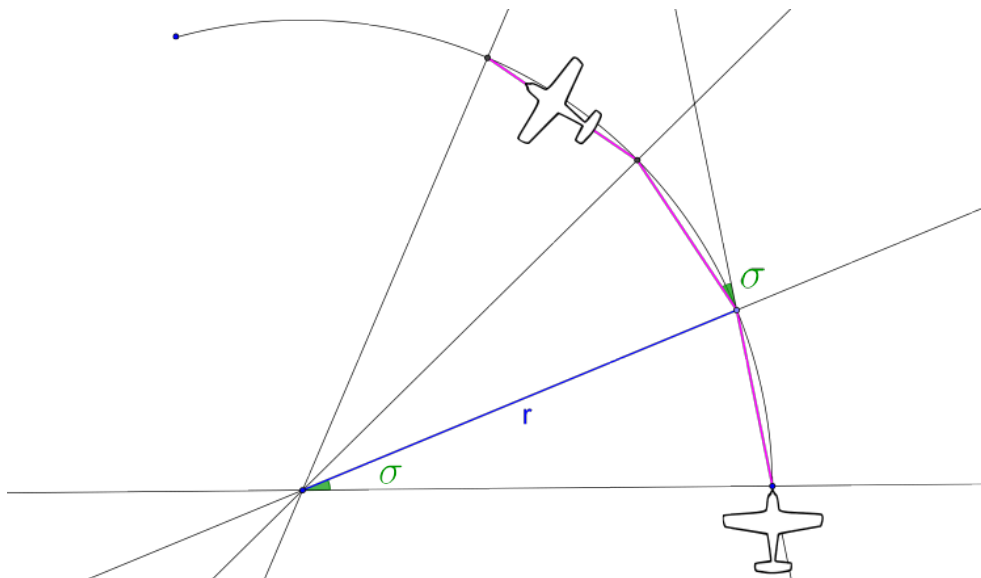
Při výpočtu klouzavosti letadla v trajektorii letu nelze zanedbat vliv větru, viz oddíl 1.1.10. Pro výpočet opadání v závislosti na směru větru byla vytvořena statická metoda *getGroundSpeedOfAirplane*, která vypočítá předpokládanou rychlost letadla z jeho pohybového vektoru a z vektoru rychlosti větru, viz rovnice 1.3. Předpokládaná rychlost letadla v_{gs} ve směru letu přímo určuje klouzavost letadla vůči zemi K_{gs} . Klouzavost se tedy v průběhu trajektorie mění v závislosti na směru a síle větru.

4.2.2.4 Výpočet bodů v zatáčce

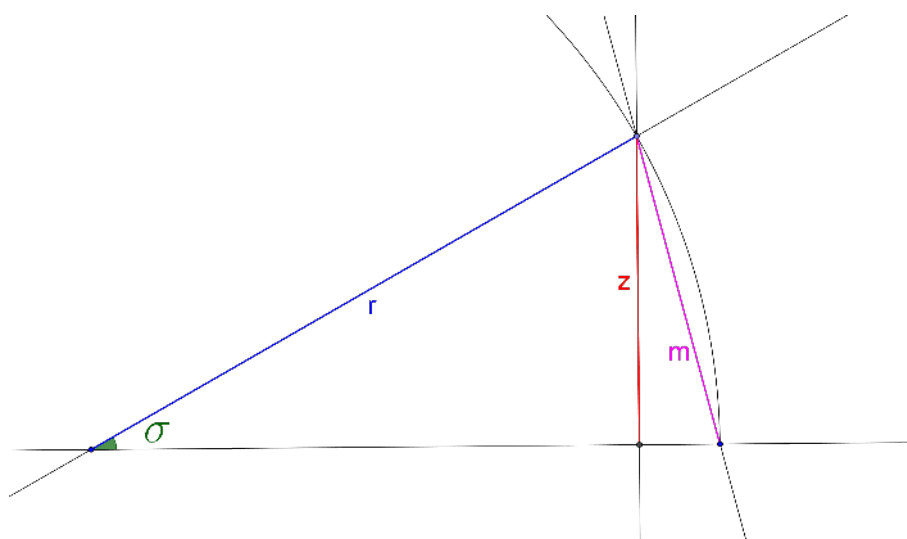
Jak již bylo nastíněno v předchozí kapitole, trajektorie letu se bude skládat z rovných částí letu a ze zatáček. Pro vizualizaci zatáčky v 3D prostoru je nutné vygenerovat zatáčku z jednotlivých bodů. Proto bylo vhodné vyvinout metodu, která takovou zatáčku vytvoří. V našem konkrétním případě byla vytvořena třída *Turn*, která uchovává informace o zatáčce, je schopná vypočítat jednotlivé body a předat je dalším výpočtům.

Způsob vytváření zatáček je zobrazen na obrázku 4.4. Pro vyšší přesnost zatáček je vhodné zvolit menší úhel σ , který je možné měnit jako konstantu výpočtu v třídě *Constants*. Pro výpočet jednotlivých bodů bylo nutné spočítat délku jedné rovné úsečky. Obrázek 4.5 ilustruje rovnice 4.1, která přímo určuje délku rovné části m .

Metoda *getPoints()* třídy *Turn* tak bude počítat jednotlivé body zatáčky bod za bodem pomocí metody *getNextPoint*, dokud nebude dosažena ukončovací podmínka. Ukončení algoritmu nastane ve dvou možných případech. První z případů nastane tehdy, když je dosažen potřebný úhel pro otočení. Druhý případ pro ukončení algoritmu nastane tehdy, když je dosažen potřebný směr letadla, viz obrázek 4.6.



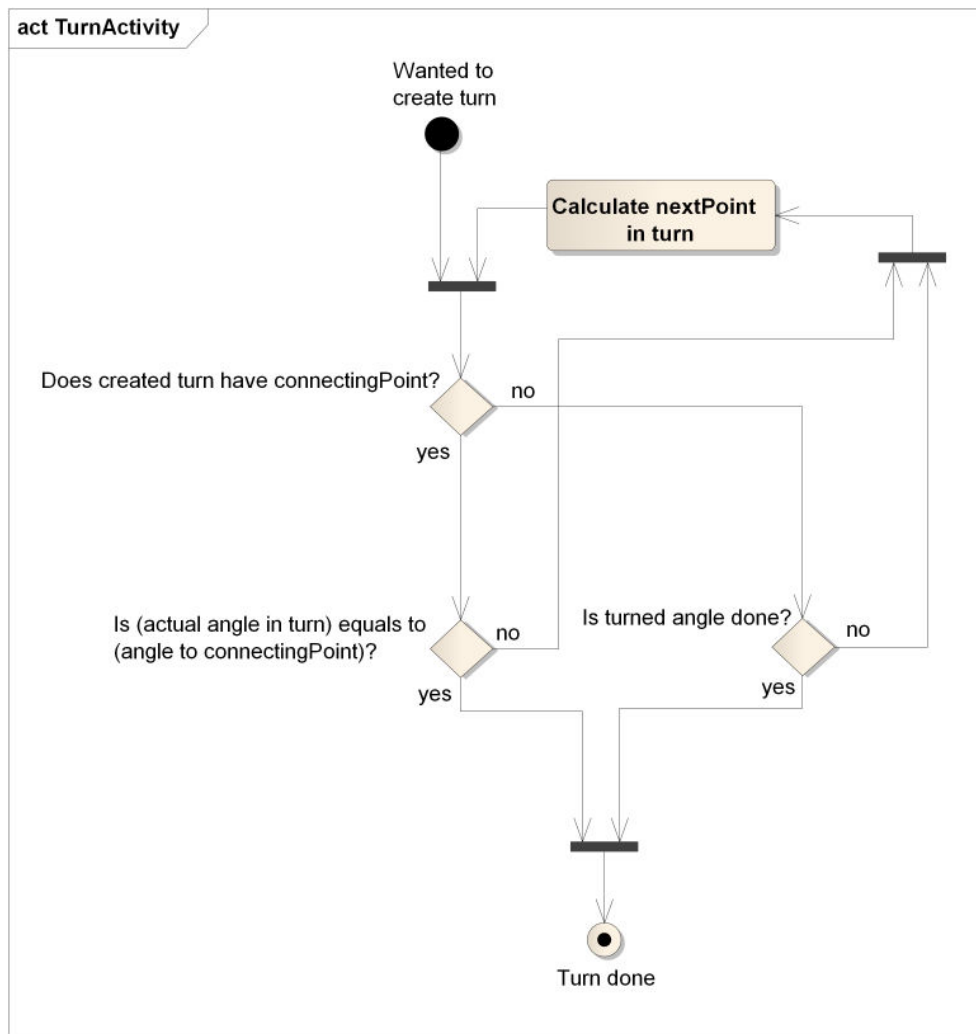
Obrázek 4.4: Grafické znázornění vytvořené zatáčky shora



Obrázek 4.5: Grafické znázornění výpočtu délky rovné trasy v zatáčce

Vyjádříme si délku jedné rovné části v zatáčce m z trojúhelníku

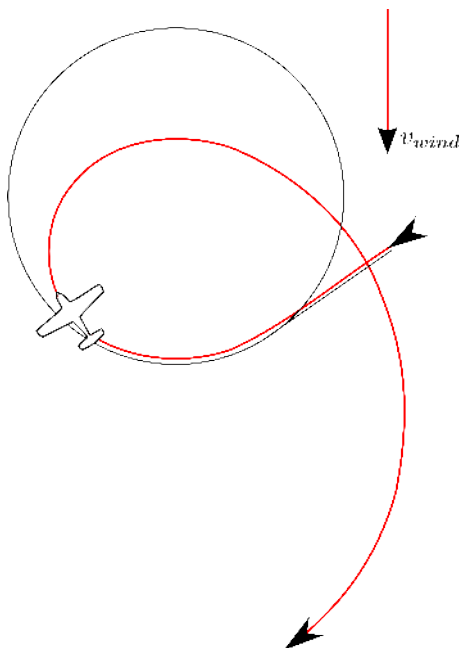
$$m = \frac{z}{\cos \frac{\sigma}{2}} \quad z = \sin \sigma \cdot r \quad (4.1)$$



Obrázek 4.6: Rozhodovací proces tvoření zatáčky

4.2.2.5 Vliv větru na tvorbu bodů v zatáčce

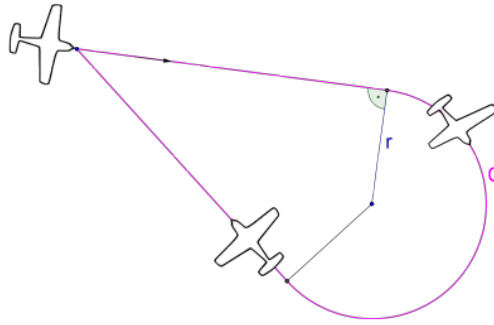
V případě, že má výpočet trajektorie nastavený vítr, dochází k přepočtu každé rovné části m v zatáčce. Přepočet je implementován na základě rovnice 1.3. V našem případě to znamená, že zatáčka bude posouvána ve směru větru, viz obrázek 4.7. Pilot letící ve větrném počasí se bude díky tomuto přepočtu snáze držet vypočtené trajektorie.



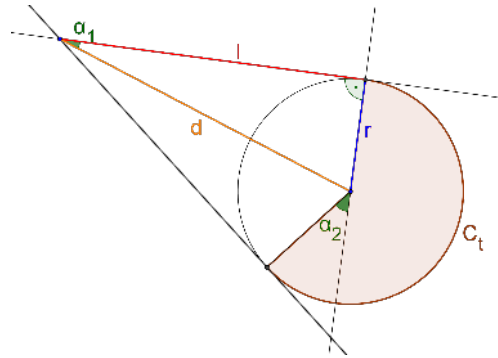
Obrázek 4.7: Vliv větru na provedenou zatáčku

4.2.2.6 Jednoduchý výpočet poloměru v zatáčce – bezvětří

Během vývoje aplikace bylo navrženo několik možných řešení způsobu přistání. Jednoduchý způsob, na kterém byl postaven složitější výpočet, můžeme vidět na obrázku 4.8. Počítaná trajektorie je na obrázku vyznačená růžovou barvou. Uvažujme, že letadlo přiletí přímo nad bod přistání. V takovém případě bude mít letadlo určitou výšku nad letištěm. Ze spočítané výšky nad letištěm zjistíme, jak daleko letadlo doklouže v bezvětří. Vzdálenost, kterou letadlo doklouže, máme vyjádřenou jako číslo v metrech označme ji C . Podle obrázku 4.9 byl vyjádřen poloměr zatáčky (rovnice 4.6) letadla, který určuje stejnou délku trajektorie, jako je vypočítaná délka doletu letadla. Úhel α_1 je v implementačním algoritmu omezen na rozsah $(0, \pi)$. Jednoduchý výpočet poloměru navede letadlo ve správné výšce a v bezvětří na přistávací dráhu, ale nerespektuje směr přistávací dráhy.



Obrázek 4.8: Grafické znázornění vytvoření zatáčky při neznámém poloměru



Obrázek 4.9: Grafické znázornění výpočtu poloměru zatáčky

Vyjádříme si délku trajektorie C

$$C = 2l + C_t \quad (4.2)$$

Vyjádříme si délku trajektorie v zatáčce C_t

$$C_t = (\pi + 2\alpha_1) \cdot r \quad (4.3)$$

Vyjádříme si délku trajektorie v rovné části letu před zatáčkou l

$$l = \frac{r}{\tan \alpha_1} \cdot C \quad \alpha_1 \neq k \cdot \pi, \quad k \in \mathbb{N} \quad (4.4)$$

Dosadíme vyjádřené l a C_t do rovnice 4.2

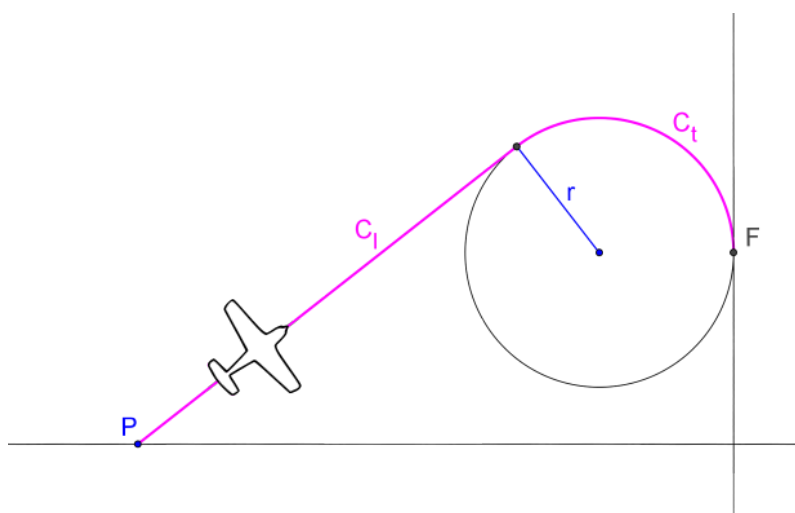
$$C = \frac{2r}{\tan \alpha_1} + (\pi + 2\alpha_1) \cdot r \quad \alpha_1 \neq k \cdot \pi, \quad k \in \mathbb{N} \quad (4.5)$$

Vyjádříme poloměr r z rovnice 4.5

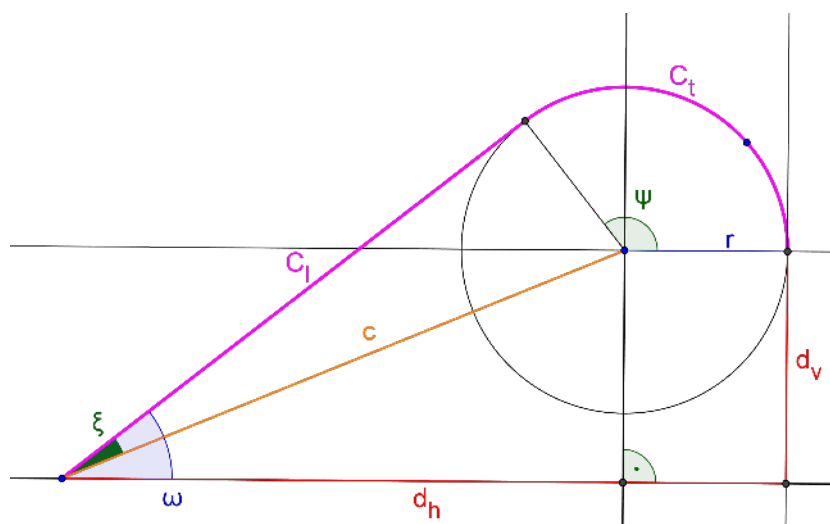
$$r = \frac{C}{\frac{2}{\tan \alpha_1} + \pi + 2\alpha_1} \quad \alpha_1 \neq k \cdot \pi, \quad k \in \mathbb{N} \quad (4.6)$$

4.2.2.7 Komplexní výpočet poloměru zatáčky – bezvětří

Při vytváření trajektorie letadla se nelze spolehnout pouze na jednoduchý výpočet (oddíl 4.2.2.6), ve kterém jsou dány stejné body počátku a konce trajektorie. V obecném případě je nutné řešit výpočet zatáčky tak, jak je zobrazeno na obrázku 4.10. Uvažujme trajektorii letadla C s přímým letem a zatáčkou. Podle obrázku 4.11 byla odvozena rovnice 4.13.



Obrázek 4.10: Grafické znázornění tvorby zatáčky



Obrázek 4.11: Grafické znázornění komplexního výpočtu poloměru zatáčky

Analyticky nelze vyjádřit poloměr vytvářené zatáčky z důvodu složitosti rovnice 4.13. Proto bylo nutné implementovat odhadování výsledku rovnice s předem zvolenou odchylkou, kterou je možné měnit ve třídě *Constants*. Odchylka je běžně nastavená na 0,01 metru a není tím pro výpočet výsledné trajektorie významná.

Vyjádříme si délku počítané trajektorie C

$$C = C_l + C_t \quad (4.7)$$

Vyjádříme si délku rovné části letu C_l

$$C_l = \frac{r}{\tan \xi} \quad (4.8)$$

Vyjádříme si z trojúhelníku úhel ξ

$$\xi = \sin^{-1} \frac{r}{\sqrt{(d_h - r)^2 + d_v^2}} \quad (4.9)$$

Vyjádříme si délku zatáčky C_t

$$C_t = r \cdot \psi \quad (4.10)$$

Vyjádříme si úhel ψ

$$\psi = \frac{\pi}{2} + \omega \quad (4.11)$$

Vyjádříme úhel ω

$$\omega = \tan^{-1} \frac{d_v}{d_h - r} \quad (4.12)$$

Dosadíme do rovnice 4.7

$$C = \frac{r}{\tan(\sin^{-1} \frac{r}{\sqrt{(d_h - r)^2 + d_v^2}})} + r \cdot \left(\frac{\pi}{2} + \tan^{-1} \frac{d_v}{d_h - r} \right) \quad (4.13)$$

Analyticky nelze vyjádřit poloměr zatáčky r , proto musíme v implementaci řešení odhadovat.

4.2.2.8 Přepočet poloměru zatáčky – s větrem

Vypočítání poloměru zatáčky v bezvětří slouží jako odhad celkového řešení. V případě nastaveného větru nebude vytvořená zatáčka opisovat tvar kružnice, ale bude mít tvar jako je zobrazen na obrázku 4.7. V takovém případě se délka vytvořené trajektorie nerovná C .

Proto dojde k přepočtu poloměrů zatáček tak, aby se délka trajektorie opět rovnala C . K tomu poslouží třída *Solution*, která zajistí přepočet poloměru díky metodě *find360SolutionWithWind*. Metoda upraví poloměry zatáček na základě délky trajektorie vypočítané z jednotlivých vygenerovaných bodů.

4.2.3 Druhy trajektorií

Z důvodu vyšší bezpečnosti přistání byly vytvořeny různé druhy počítaných trajektorií. Nejbezpečnější druh navržené trajektorie je *FinaleFixedPath*, ale zároveň potřebuje pilot pro tento druh vysokou výšku v blízkosti letiště. Další druhy počítaných trajektorií jsou *ThreeTurnsPath* a *TwoTurnsPath*. Díky těmto trajektoriím, se dostane pilot nad přistávací dráhu s minimální rezervou. V případě, že pilot s daným nastavením letadla nedoklouže na nejbližší letiště, je navržená trajektorie *StraightPath*. Tato trajektorie obsahuje pouze rovnou část letu na nejbližší letiště. Klouzavost u trajektorie *StraightPath* bude vyšší než je v parametrech nastavení. V takovém případě ovšem může pilot následovat trajektorii například s motorem na poloviční výkon a pokusit se nechat přístroj znovu přepočítat trasu v menší vzdálenosti od letiště.

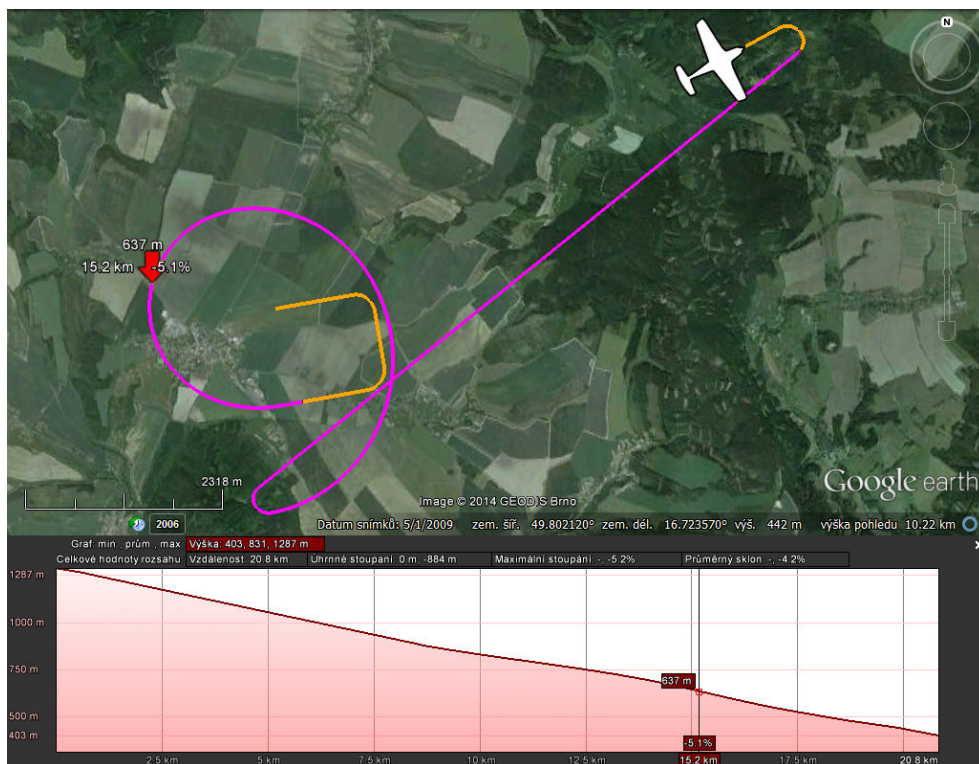
4.2.4 Trajektorie FinaleFixedPath

Pokud má pilot dostatek výšky, je navržená trajektorie *FinaleFixedPath*. Navržená trajektorie je zobrazena na obrázku 4.12 a na obrázku 4.13. Trajektorie respektuje základní přistávací pravidla a preferuje zatáčení doleva. Zatáčky doleva jsou o mnoho bezpečnější, protože piloti sedí zpravidla v kokpitu vlevo a díky tomu vidí v zatáčce lépe na přistávací plochu.

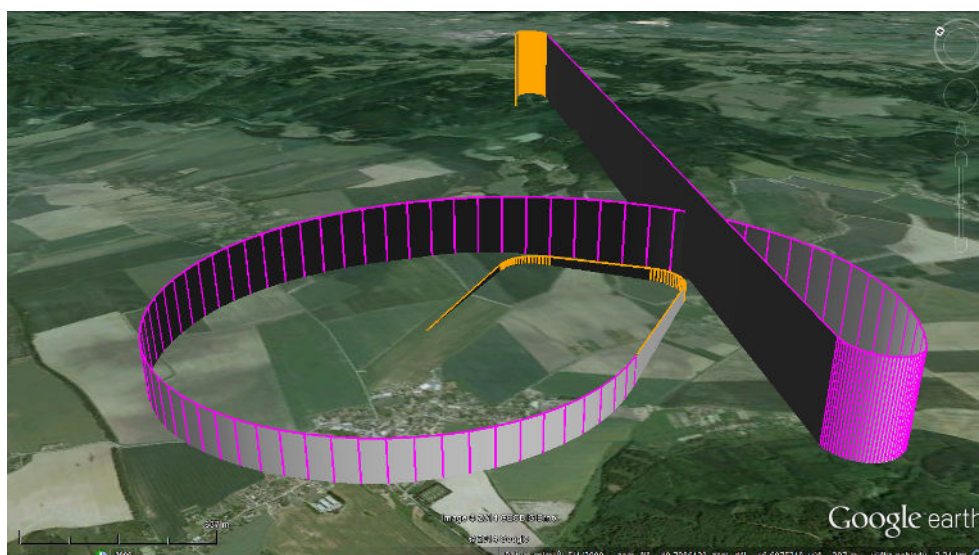
Vypočtená trajektorie zobrazená na obrázku 4.12 má nastavený vítr 25 km/h z jihu. Proto je na zatáčce s největším poloměrem vidět snos na sever. Na obrázku 4.12 můžeme ve spodní části je patrný vertikální profil trasy, který zobrazuje klouzavost letadla vůči zemi K_{gs} . Z vertikálního profilu trasy lze vyčíst zhoršenou klouzavost letadla v případě, že letadlo letí proti větru.

Žlutě vyznačená trajektorie je pevná ve výpočtu trajektorie se nemění. Úpravy poloměrů zatáček je nutné aplikovat pouze různě vyznačenou část letu. V metodě *find360SolutionWithWind* jsou měněny poloměry zatáček do té doby, dokud klesání ve všech bodech neodpovídá správnému K_{gs} .

4.2. Modul EmergencyLanding



Obrázek 4.12: Ukázka trajektorie letu FinaleFixedPath

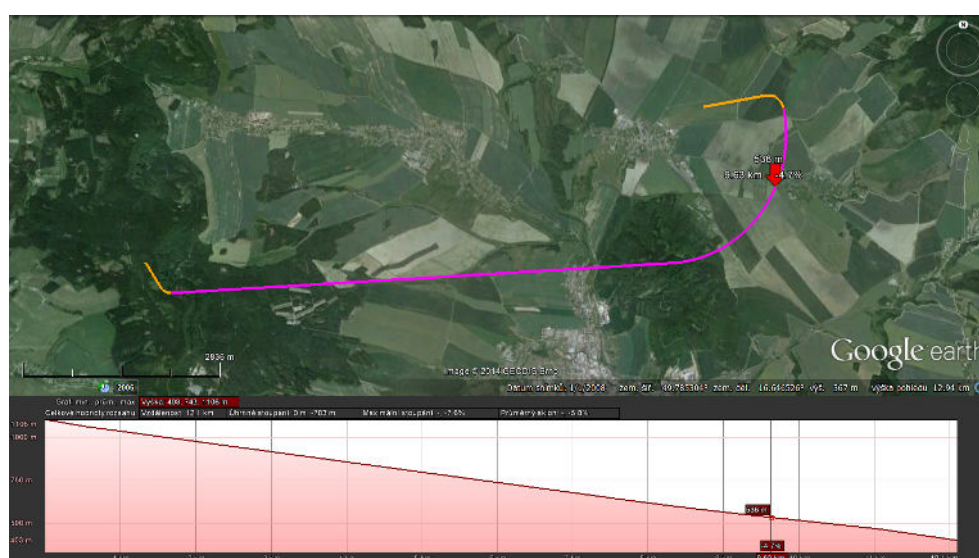


Obrázek 4.13: Ukázka trajektorie letu FinaleFixedPath zobrazená ve 3D

4.2.5 Trajektorie ThreeTurnsPath

Trajektorie *ThreeTurnsPath* je počítána v případě, že nelze spočítat trajektorii *FinaleFixedPath* tak, aby měly všechny zatáčky větší poloměr než 200 metrů. Obecně lze říci, že trajektorie *ThreeTurnsPath* je použita v případě, že letadlo doklouže s malou rezervou k letišti. Typická trajektorie *ThreeTurnsPath* je zobrazena na obrázku 4.14. Trajektorie navede na přistání vždy proti větru, ale jinak nerespektuje standardní přistávací manévry.

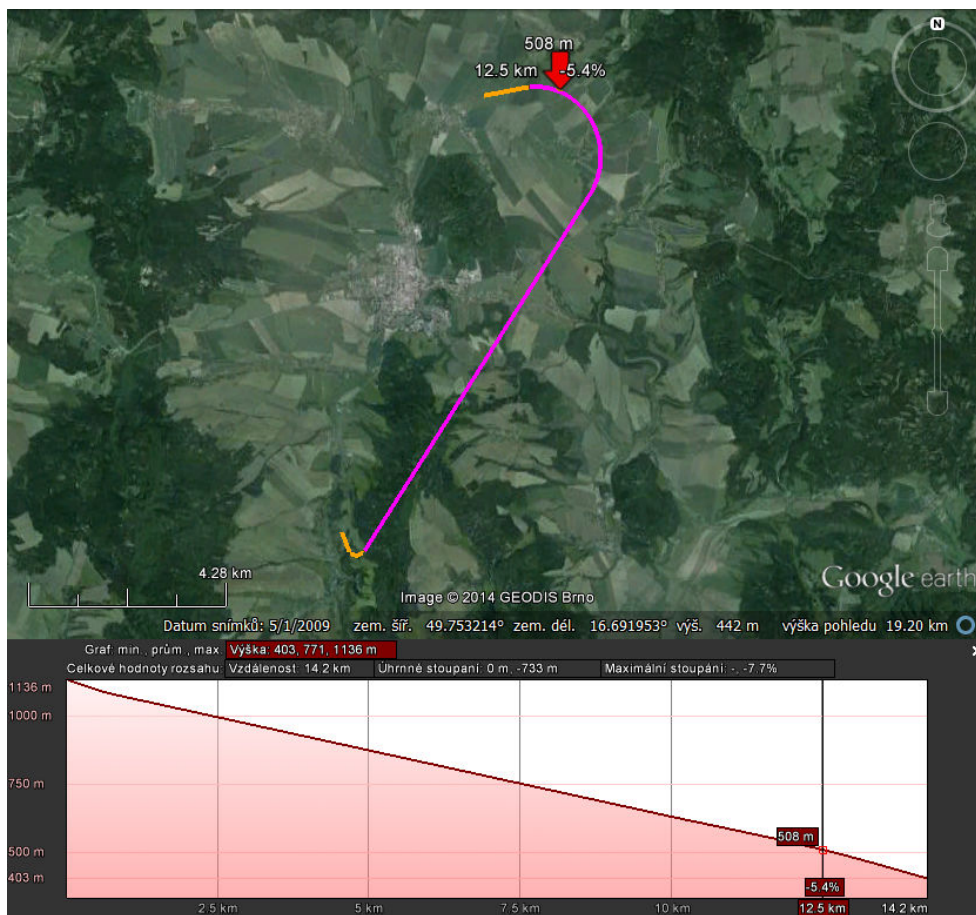
V trajektorii *ThreeTurnsPath* je třeba navrhnout takový poloměr zatáčky (vyznačena růžově), aby klouzání letadla v každém bodě odpovídalo K_{gs} .



Obrázek 4.14: Ukázka trajektorie ThreeTurnsPath

4.2.6 Trajektorie TwoTurnsPath

V případě, že nelze spočítat zatáčky v trajektorii *ThreeTurnsPath* s větším poloměrem než 200 metrů, je spočítána trajektorie *TwoTurnsPath*. Tato trajektorie neplýtvá výškou letadla a vede ho nejkratší možnou cestou na přistání, viz obrázek 4.15. V trajektorii je opět nutné spočítat správný poloměr zatáčky, jako v předchozích případech.



Obrázek 4.15: Ukázka trajektorie TwoTurnsPath

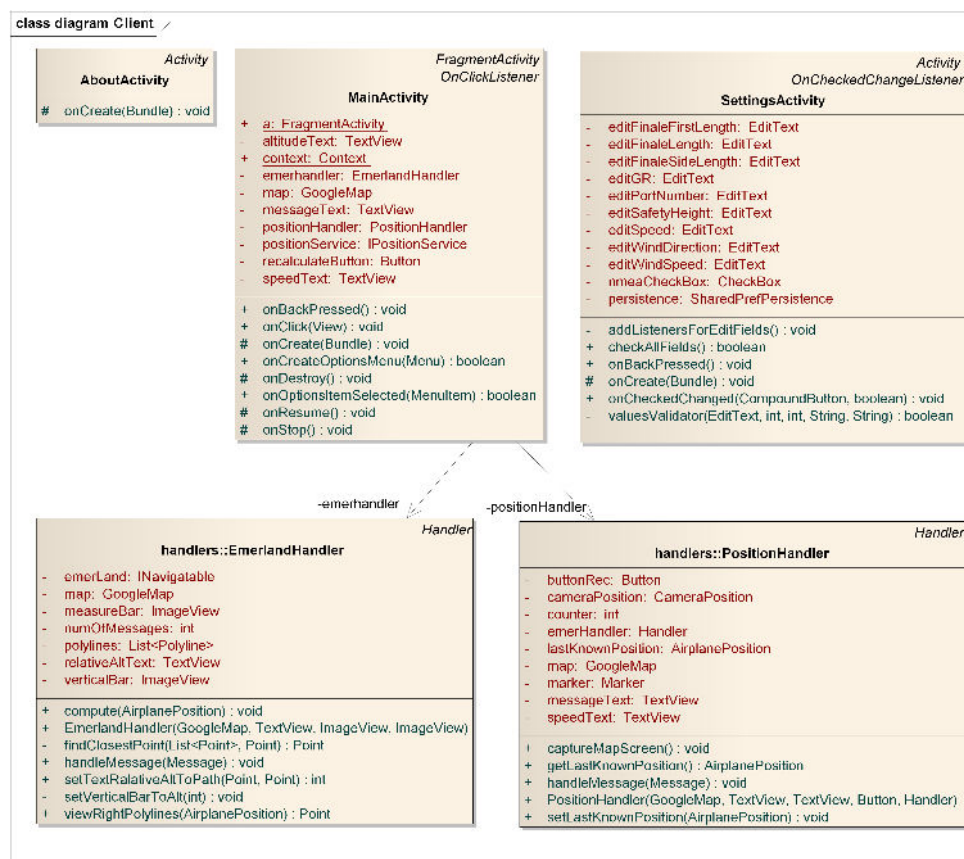
4.2.7 Trajektorie StraightPath

V případě, že nelze spočítat ani jednu z předchozích trajektorií s poloměry zatáček větší než 200m, je spočítána trajektorie *StraightPath*. Trajektorie *StraightPath* neobsahuje žádnou zatáčku a je pouze ukazatelem přímého směru na letiště. Trajektorie je navržena pouze dvěma body a klouzavost v trajektorii neodpovídá běžnému klouzání letadla.

V případě, že je trajektorie spočítána, je velmi pravděpodobné, že letadlo nedoklouže v bezmotorovém režimu na žádné letiště v databázi. V případě, že se jedná o výpadek motoru, měl by pilot upřednostnit přistání na neoznačené ploše, například na poli. V případě, že je motor stále funkční může se pilot pokusit letět co neefektivněji podél trajektorie a nechat trasu přepočítat v průběhu letu k letišti.

4.3 Modul Client

Modul *Client* je platformě závislý a je naprogramován pro mobilní platformu Android. Modul zajišťuje komunikaci s uživatelem a zobrazování dat. V modulu jsou vytvořeny tři aktivity *MainActivity*, *SettingsActivity* a *AboutActivity*. Class diagram modulu *Client* je zobrazen na obrázku 4.16.



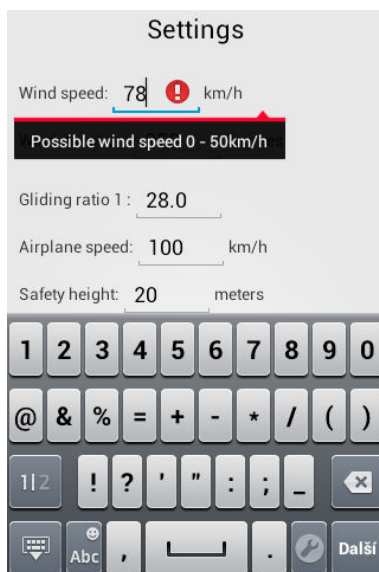
Obrázek 4.16: Class diagram modulu Client

4.3.1 MainActivity

Aktivita *MainActivity* je hlavní aktivitou, zobrazující uživateli letové údaje a navigaci. Aktivita vykresluje podkladovou mapu s využitím Google maps v2. Aktivita uchovává odkaz na dvě třídy *PositionHandler* a *EmerlandHandler*, které ji obsluhují změnu údajů o rychlosti letadla, výšce letadla, pozici a změnu navigačních křivek zobrazených na mapě.

4.3.2 SettingsActivity

Aktivita *SettingsActivity* umožňuje změnu parametrů výpočtu a aplikace. Aktivita je vytvořená podle návrhu, viz oddíl 3.5.2. Aktivita podporuje vložení rozsahu hodnot, který je specifikovaný v třídě *ParametersRange*, viz obrázek 4.17.



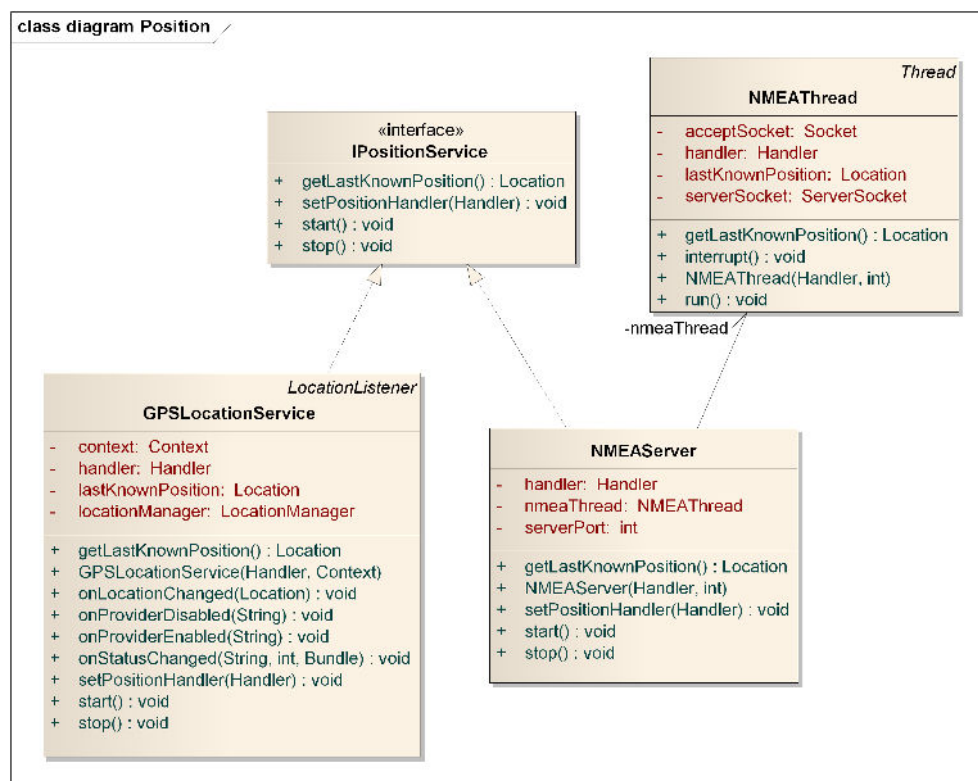
Obrázek 4.17: SettingsActivity s příliš silným zadaným větrem

4.3.3 AboutActivity

Aktivita *AboutActivity* zobrazuje pouze text popisující aplikaci. Text popisuje aplikaci a upravuje podmínky použití aplikace. Aktivitu je možné vyvolat stisknutím systémového tlačítka menu a výběr položky *About Application*.

4.4 Modul Position

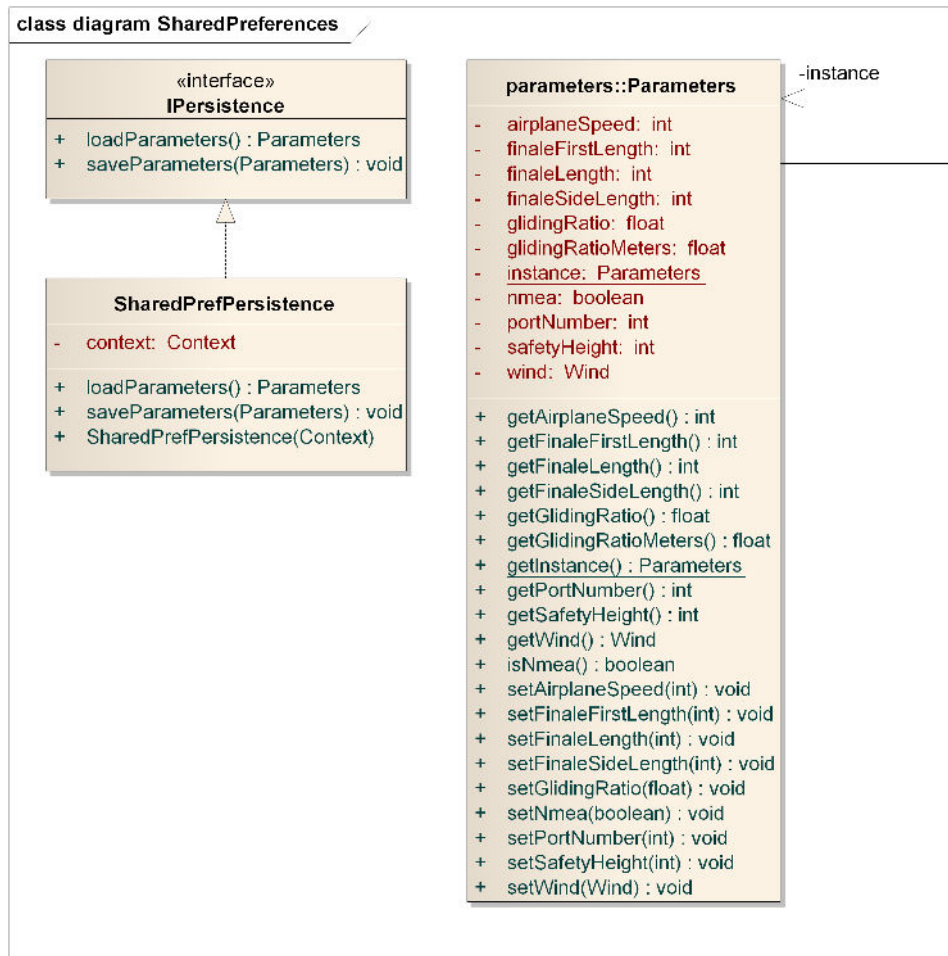
Modul *Position* zajišťuje předání pozice třídě *PositionHandler* pomocí posílání zpráv *android.os.Message*. Pozici je možné získat z GPS senzoru pomocí třídy *GPSLocationService* nebo ze simulátoru díky třídě *NMEA Server*. Class diagram modulu je zobrazen na obrázku 4.18.



Obrázek 4.18: Class diagram modulu Position

4.5 Modul SharedPreferences

Modul *SharedPreferences* zajišťuje načtení nastavených parametrů z posledního běhu aplikace a zároveň v případě změny parametrů ukládá parametry do souboru. Pro uložení a načtení parametrů používá modul Android třídu *SharedPreferences*, která poskytuje základní framework pro ukládání dat. Modul vytvoří novou instanci třídy *Parameters*, která je implementovaná jako singleton, viz obrázek 4.19. Interface *IPositionService* obsahuje metody *start* a *stop*, které je nutné volat za účelem pozastavení nebo ukončení vlákna při skrytí aplikace systémovým požadavkem.



Obrázek 4.19: Class diagram modulu SharedPreferences

Testování

Testování bylo provedeno na několika úrovních. Aplikace byla otestována Unit testy, manuálním testováním za pomoci simulátoru a manuálním testováním v reálné situaci. Tato kapitola podrobně popisuje jednotlivé testování.

5.1 Unit testy

Během vývoje aplikace byl vytvořený nový Java projekt, který testuje implementovanou aplikaci. K testování je použit framework *JUnit4*. Veškeré testovací metody jsou kompletně popsány v *JavaDoc* přiloženy na CD.

Tato diplomová práce je zaměřena především na použitelnost modulu *EmergencyLanding*. Proto je cíleně testován pouze tento nejrozsáhlejší modul aplikace. Všechny vytvořené *JUnit4* testy jsou *PASS*. Pokrytí implementovaného kódu v modulu *EmergencyLanding* je zobrazeno v tabulce 5.1.

Package	Pokrytí implmentovaného kódu
cz.cvut.fit.dip.fms.emerland.model	97,2%
cz.cvut.fit.dip.fms.emerland.solutions	97,1%
cz.cvut.fit.dip.fms.emerland.io	98,2%
cz.cvut.fit.dip.fms.emerland.api	90,1%
cz.cvut.fit.dip.fms.emerland.exceptions	100%
cz.cvut.fit.dip.fms.emerland	96,52%

Tabulka 5.1: Coverage implementovaného kódu v modulu *EmergencyLanding*

5.1.1 Testování vytvořené trajektorie

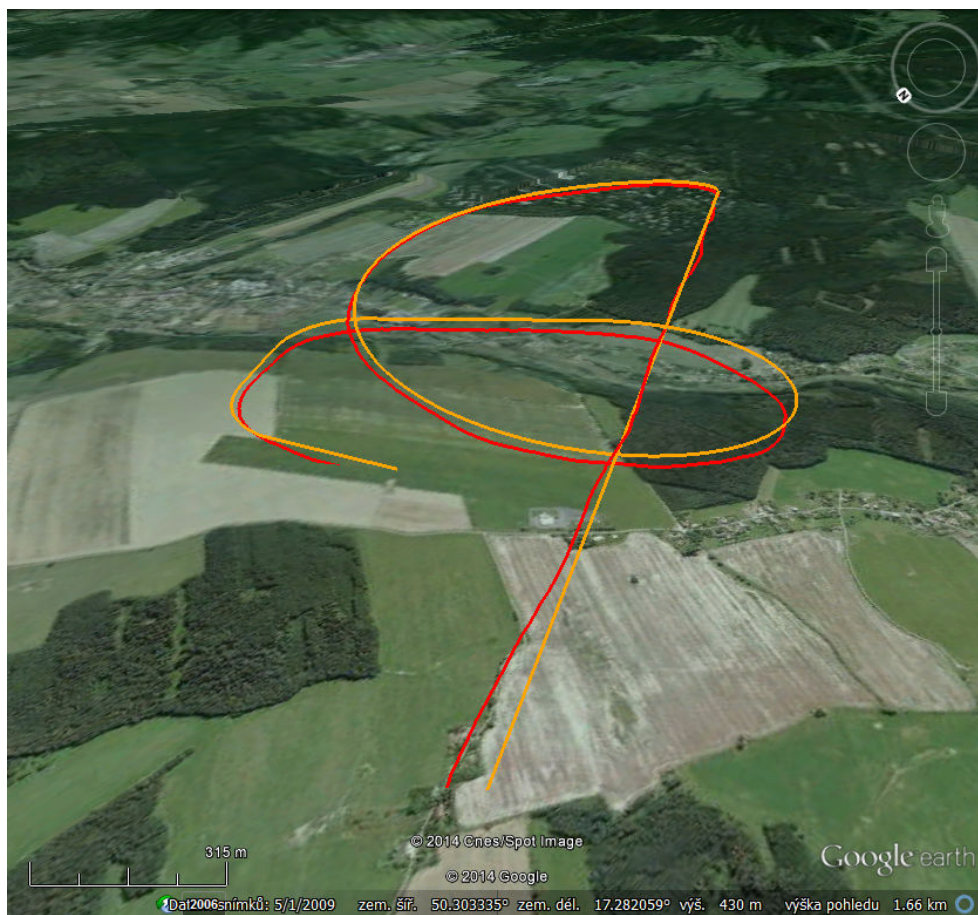
Nejzajímavějším testem je *JUnit4* test *TestNavigatableModule*, který testuje funkčnost celého modulu *EmergencyLanding*. Test vytváří ve smyčce náhodné pozice a kurzy letadla v okolí jednoho letiště. Pro každou náhodně vygenerovanou pozici a kurz letadla je spočítána trajektorie. Trajektorie je testována na spojitost v každém bodě. Dále je v každém bodě testována správná klouzavost letadla K_{gs} spočtena z pohybového vektoru letadla a větru. Nakonec test zjišťuje zda-li je poslední bod trajektorie na přistávací dráze letiště. Tento test ukládá vygenerované trajektorie do složek *results* a *resultsIGC*, díky čemuž je snadné vybrat si specifickou vypočítanou trajektorii pro další například manuální testování.

5.2 Manuální testování trajektorie na simulátoru

Při návrhu trajektorií bylo nutné testovat, zda-li je možné danou trajektorii následovat letadlem. Pro testování byl vybrán simulátor klouzavého letu Condor verze 1.1.5. Z výstupu *JUnit4* testu *TestNavigatableModule* získáme pro každou trajektorii dva soubory jeden s příponou *kml* a druhý s příponou *igc*.

Podle zdroje [4] bylo zjištěno, že lze zobrazit druhé letadlo v simulátoru jako tzv. ducha (ghost). Zobrazení ducha v simulátoru umožňuje vygenerovaný soubor *igc*, který je spočítán ve 4D, kde čtvrtý rozměr je čas. Vygenerovaný soubor *igc* byl pomocí webové stránky [4] převeden na specifický formát simulátoru Condor *ftr*. Po startu simulátoru v blízkosti vypočítané trajektorie, bylo nutno nalézt letící letadlo po trajektorii (ducha). Posléze srovnat letové parametry a snažit se letět stejně jako letící duch. Z počátku to byl problém, ale po změně letových parametrů bylo možné letícího ducha celkem snadno následovat až na přistávací dráhu. Testování na simulátoru provedl autor práce, který má zkušenosti z padákového kluzáku (nalétáno 400 hodin).

Na obrázku 5.1 je možné vidět srovnání spočítané a reálné trajektorie letu. Žlutě je zobrazené vypočítaná trajektorie a červeně je zobrazený reálný záznam letu letadla. Z obrázku 5.1 lze vyčíst, že trajektorie je správně navrhnutá a je možné ji klouzavým letem následovat. Obrázek 5.2 ilustruje, jakým způsobem bylo testování prováděno. Na obrázku 5.2 můžeme vidět druhé letadlo (ducha), které letí navrženou trajektorií. Jednotlivé testované trajektorie jsou přiloženy na CD.



Obrázek 5.1: Porovnání vypočítané trajektorie a letu kluzáku v simulátoru Condor podél trajektorie.



Obrázek 5.2: Ukázka testování vypočtené trajektorie za využití ducha v simulátoru Condor

5.3 Manuální testování aplikace na simulátoru

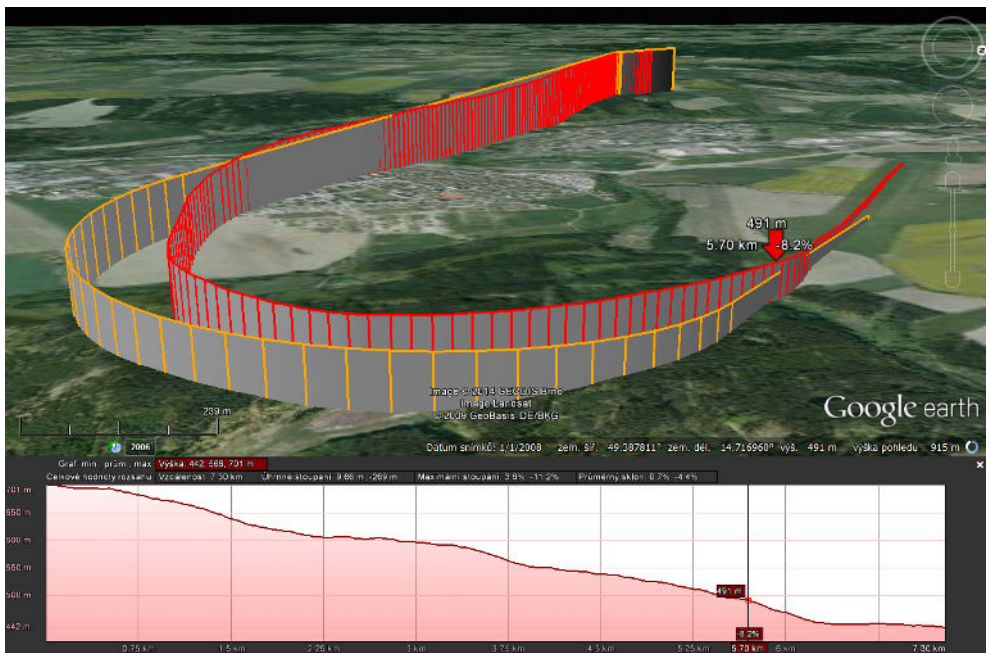
Před reálným testováním v ULL bylo nutné otestovat mobilní aplikaci pomocí simulátoru. Po předchozích zkušenostech a možnostech byl vybrán opět simulátor klouzavého letu Condor. Podle zdroje [5] lze spojit externí zařízení přes wifi se simulátorem pomocí NMEA 0183. Po nastavení aplikace a simulátoru bylo možno přeposílat polohu ze simulátoru do implementované aplikace. Tímto způsobem byly manuálně testovány desítky jednotlivých přistání. Bohužel nemá simulátor Condor zcela kalibrované mapy, proto bylo nutné zvolit pouze letiště v blízkosti středu mapy. Testování ukázalo několik nedostatků, které byly posléze upraveny. Z testování vyplynulo, že je aplikace použitelná v reálném prostředí. Video zachycující manuální testování aplikace je přiloženo na CD.

5.4 Reálné testování aplikace v reálném prostředí

Reálné testování implementované aplikace proběhlo dne 26. 4. 2014 na letišti Aeroklubu Tábor (LKTA). Cílem bylo zjistit funkčnost a koncepci aplikace v reálném prostředí letu. K testování byl zvolen motorový kluzák L-13SW Vivat, který svými letovými výkony odpovídá mnoha typům ULL a díky brzdícím klapkám umožňuje variabilní úpravu rychlosti opadání v_o . Testování se ujal zkušený pilot Bc. Jan Hruška, civilní a vojenský pilot s celkovým náletem 400 hodin na větroních, UL a motorových letounech. Provoz probíhal na

Před samotným testováním byly do telefonu zadány parametry letounu: klouzavost 25 a rychlost při nejlepší klouzavosti 100 km/h. Testování proběhlo bez větších problémů a aplikace se ukázala jako použitelná navigace letounu.

Pilot se při dodržování vypočítané trajektorie cítil před přistávacím manévrem příliš nízko, proto i v některých případech byl nucen zapnout motor. Po poslední zatačce by pilot navrhl výšku letadla zvýšit až o 30 metrů. Tato situace nastala proto, že aplikace počítá se stále stejnou nejlepší klouzavostí za účelem co nejefektivnější využití výšky. Na obrázku 5.3 je situace vyššího rozpočtu dobře patrná. Pilot měl pocit nízké výšky proto vytvořil zatačku s nižším poloměrem a snažil se dostat do stejné pozice jako je zvyklý před přistávacím manévrem.



5. TESTOVÁNÍ

Pilot zavrhuje použití aplikace v případě výpadku motoru, protože pilot se podle něho nestihne soustředit na všechny údaje, startování motoru a nouzových procedur. Přesto vidí jako velké uplatnění aplikace při nouzové situaci zhoršené viditelnosti či při běžném přistání. Kompletní hodnocení aplikace od vojenského pilota Bc. Jana Hrušky je uvedeno v příloze C.

Závěr

V rámci diplomové práce byla vytvořena aplikace pro mobilní platformu Android, která usnadňuje pilotům ultralehkých letadel přistání v případě nouzové situace. Mobilní aplikace spočítá sestupnou trajektorii letu tak, aby bylo možné klouzavým letem přistát na nejbližším letišti. Aplikace zahrnuje do výpočtů nastavený vítr a dokáže navigovat pilota po vypočítané trajektorii až na přistávací dráhu. Aplikace se ukázala jako použitelná nejenom pro ULL, ale i u kluzáků a dalších typů sportovních létajících zařízení.

Trajektorii letu bylo nutné vypočítat na mobilním zařízení dostatečně rychle tak, aby nebyla ovlivněna bezpečnost přistání. Proto byla v rámci práce použita vlastní deterministická metoda odhadu, která je dostatečně přesná a rychlá. Vypočtenou trajektorii lze po správném nastavení parametrů snadno následovat letadlem i v režimu bezmotorového letu. Trajektorie v případě dostatečné výšky dodržuje pravidla bezpečného přistání. V případě nízké výšky letadla je spočítána trajektorie, která maximálně využívá výšku letadla. Použitá Databáze letišť od Patrika Sainera omezuje použití aplikace pouze na území České a Slovenské republiky.

Aplikace byla formálně navržena až na úroveň SW modulů a návrhů komunikačních API mezi nimi. Aplikace byla také otestována Unit testy, které byly navrženy především pro modul *EmergencyLanding*. Reálné testování aplikace ukázalo, že při běžném přistání jsou piloti zvyklí přistávat s vyšší výškou. Trajektorii letu by proto bylo vhodné počítat v závislosti na nebezpečnosti konkrétní situace. V případě velkého nebezpečí využívat maximální klouzavosti letadla i při dosednutí tak, jako je tomu teď. Naopak v případě běžného přistání s minimálním nebezpečím by bylo vhodné poslední úsek dosednutí letadla počítat s nižší klouzavostí.

Jako rozšíření aplikace se nabízí možnost dokončit implementaci navržené aplikace. Implementovat moduly *Weather* a *Sensors* a zvýšit tím přesnost výpočtu trajektorie. Další možností rozšíření je úprava GUI aplikace na 3D pohled, ve kterém by mohla být spočítaná trajektorie lépe viditelná. Jako další rozšíření výpočtu trajektorie se nabízí počítat trajektorii v závislosti

ZÁVĚR

na okolním terénu. Pro použitelnost vytvořeného výpočtu by bylo vhodné zaměřit se na integraci modulu *EmergencyLanding* do aplikací, které piloti ULL již využívají.

Literatura

- [1] Allen, G.: *Android 4.0, Průvodce programování mobilních aplikací*. Albatros Media a.s., 2013.
- [2] Bogan, L.: *Advanced Aerodynamics - Bluenose Soaring Club Ground School*. [Online; accessed 2-February-2014]. Dostupné z: http://www.bogan.ca/soaring/Advanced_Aerodynamics.PDF
- [3] Bussinesweek: *Google Buys Android for Its Mobile Arsenal*. [Online; accessed 5-April-2014]. Dostupné z: <http://www.webcitation.org/5wk7sIvVb>
- [4] Condor Club: *Import IGC file to Condor format - Condor Club*. [Online; accessed 6-January-2014]. Dostupné z: <http://www.condor-club.eu/igc2ftr/180/>
- [5] Condor Club Forum: *Using XCSoar for Android with Condor over Wifi: Setup Guide*. [Online; accessed 6-January-2014]. Dostupné z: <http://forum.condorsoaring.com/viewtopic.php?f=1&t=12538>
- [6] Daněk, M.: *Aerodynamika a mechanika letu pro piloty a techniky*. Naše vojsko, 1958.
- [7] Federazione degli Aeroclub VdA: *LK8000 society*. [Online; accessed 12-December-2013]. Dostupné z: <http://www.aecaosta.it/LK8000/LkAosta.htm>
- [8] FlightSource.ca: *Esterline CMC Electronics Displays Avionics at Heli-Expo*. [Online; accessed 15-April-2014]. Dostupné z: <http://www.flightsource.ca/blog/flightdeck/2010/02/20/esterline-cmc-electronics-displays-avionics-at-heli-expo/>
- [9] Glenn Baddeley: *GPS - NMEA sentence information*. [Online; accessed 20-October-2014]. Dostupné z: <http://aprs.gids.nl/nmea/>

- [10] Lengál, I.: iPad na palubě. *Pilot*, 7 2013: s. 12,13,14.
- [11] Letecká amatérská asociace: *Ultralehké letouny*. [Online; accessed 3-April-2014]. Dostupné z: <http://www.laacr.cz/stranky/odbornosti/ultralehke-letouny/default.aspx>
- [12] Letecká škola Piloták.cz: *Ultralehká letadla - Ultralight*. Dostupné z: <http://www.pilotak.cz/ultralehka-letadla/>
- [13] LK8000 society: *LK8000 Tactical Flight Computer*. [Online; accessed 12-December-2013]. Dostupné z: <http://www.lk8000.it/>
- [14] LK8000 society: *Software*. [Online; accessed 12-December-2013]. Dostupné z: <http://www.lk8000.it/download/software.html>
- [15] Meyer, A.: *Xavion with ADS B*. 2013, [Online; accessed 6-November-2013]. Dostupné z: <https://www.youtube.com/watch?v=3ZADo9F6N7I>
- [16] PGweb/XContest: *AAT úlohy v LK8000*. [Online; accessed 12-December-2013]. Dostupné z: <http://www.pgweb.cz/zpravodajstvi/clanek:aat-ulohy-v-lk8000>
- [17] Sainer, P.: Tablety a telefony pro leteckou navigaci. *Pilot*, 5 2013: s. 14,15,16,17.
- [18] Seattle Avionics: *Seattle Avionics - Products*. [Online; accessed 8-November-2013]. Dostupné z: <https://www.seattleavionics.com/Products.aspx>
- [19] SkyDemon: *SkyDemon, flight planning and in-flight GPS navigation software for PCs and Macs*. [Online; accessed 5-November-2013]. Dostupné z: <http://www.skydemon.aero/>
- [20] Ústav pro odborné zjišťování příčin leteckých nehod: *Zprávy o LN a Incidentech*. [Online; accessed 30-April-2014]. Dostupné z: http://www.uzpln.cz/cs/ln_incident
- [21] Střelcová, G.: *iOS vs Android*. Metodický portál RVP.CZ, [Online; accessed 5-April-2014]. Dostupné z: <http://spomocnik.rvp.cz/clanek/18205/IOS-VS-ANDROID.html>
- [22] Svět Androida: *Android vs iOS - kdo a v jaké zemi je oblíbenější?* [Online; accessed 5-April-2014]. Dostupné z: <http://www.svetandroida.cz/android-vs-ios-oblibenost-201403>
- [23] Šulc, B. D.: *Návrh softwarového modulu trenažéru přistání na padákovém kluzáku*. Diplomová práce, Vysoké učení technické v Brně, 2011.

-
- [24] Ventafridda, P.; Heynen, U.: *User Manual update*. LK8000 society, [Online; accessed 12-December-2013]. Dostupné z: http://www.bware.it/LK8000/LK8000_MANUAL_400.pdf
- [25] Vertical Power: *Products*. [Online; accessed 19-October-2013]. Dostupné z: <http://verticalpower.com/>
- [26] ČVUT FIT - katedra měření: *Přednáška - Flight management system*. Dostupné z: <http://measure.feld.cvut.cz/groups/lis/download/prednasky/EPS1/FMS.pdf>
- [27] Vyhnal, I. L.: *Četnosti EFIS v ULL - Laa*. 2014, laa Inspektor provozu ULL.
- [28] Wikipedia: *Android (operační systém)*. [Online; accessed 5-April-2014]. Dostupné z: http://cs.wikipedia.org/wiki/Android_%28opera%C4%8Dn%C3%AD_syst%C3%A9m%29
- [29] Wikipedia: *Haversine formula*. [Online; accessed 2-April-2014]. Dostupné z: http://en.wikipedia.org/wiki/Haversine_formula
- [30] Wikipedia: *iOS (Apple)*. [Online; accessed 5-April-2014]. Dostupné z: http://cs.wikipedia.org/wiki/IOS_%28Apple%29
- [31] Wikipedia: *Klouzavost*. [Online; accessed 3-April-2014]. Dostupné z: <http://cs.wikipedia.org/wiki/Klouzavost>
- [32] Wikipedia: *Kluzák*. [Online; accessed 3-April-2014]. Dostupné z: <http://cs.wikipedia.org/wiki/Kluzák>
- [33] Wikipedia: *NMEA 0183*. [Online; accessed 1-April-2014]. Dostupné z: http://en.wikipedia.org/wiki/NMEA_0183
- [34] X-Avionics, LLC: *Xavion Full Description*. [Online; accessed 5-January-2014]. Dostupné z: <http://xavion.com/>
- [35] Xample LLC: *Air Navigation Features / Xample*. [Online; accessed 20-November-2013]. Dostupné z: <http://www.xample.ch/air-navigation/air-navigation-features/>
- [36] Xample LLC: *See Air Navigation in action / Xample*. [Online; accessed 20-November-2013]. Dostupné z: <http://www.xample.ch/air-navigation/air-navigation-screenshots/>
- [37] XampleAirNav: *Air Nav 5 (new Features) - Air Navigation Pro*. 2012, [Online; accessed 6-November-2013]. Dostupné z: https://www.youtube.com/watch?v=DZda_BaJh3o

Seznam použitých zkratk

PNA	Personal Navigation Assistant
EFIS	Electronic Flight Information System
AGL	Above Ground Level
VFR	Visual Flight Rules
IFR	Instrument Flight Rules
IGC	International Gliding Commission
ASCII	American Standard Code for Information Interchange
TCP/IP	Transmission Control Protocol/Internet Protocol
NMEA	The National Marine Electronics Association
TAF	Terminal Aerodrome Forecast
NOTAM	Notice To Airmen
MTOW	Maximum Takeoff Weight
ETA	Estimated Time of Arrival
ETE	Estimated Time Enroute
VOR	VHF Omnidirectional Range
DME	Distance Measuring Equipment
FMS	Flight Management System
ULL	Ultra Lehké Letadlo
GPS	Global Positioning System

A. SEZNAM POUŽITÝCH ZKRATEK

SLZ Sportovní Létaající Zařízení

GUI Graphical User Interface

API Application Programming Interface

XML Extensible Markup Language

Obsah přiloženého CD

	readme.txt.....	stručný popis obsahu CD
	app.....	adresář s aplikací umožňující instalaci
	src	
	impl.....	zdrojové kódy implementace
	thesis.....	zdrojová forma práce ve formátu $\text{\LaTeX}$
	sim.....	data pro instalaci NMEA simulátoru
	test.....	výstupy z testů v průběhu implementace
	text.....	text práce
	thesis.pdf.....	text práce ve formátu PDF
	thesis.ps.....	text práce ve formátu PS

Zpráva z reálného testování

C.1 Prostředky a místo

Testování aplikace UL FMS 26.4.2014 na letišti Aeroklubu Tábor (LKTA). Cílem bylo zjistit funkčnost a koncepci aplikace v reálném prostředí letu. K testování byl zvolen motorový kluzák L-13SW Vivat, který svými letovými výkony odpovídá mnoha typům UL letounů a díky brzdícím klapkám umožňuje variabilní úpravu sestupové rychlosti. Provoz probíhal na dráze 30. Zvláštností této dráhy jsou 20 m vysoké stromy 500 m před prahem dráhy, které teoreticky zasahují do naprogramované sestupové osy. Aplikace běžela na mobilním telefonu Sony Xperia Tipo Dual umístěném na úchytu v kabině letounu.

C.2 Průběh testu

Před samotným testováním jsme zadali do aplikace parametry letounu: klouzavost 25, referenční rychlost 100 km/h a rezervu nad sestupovou cestou +30 m. Testovací let proběhl v dopoledních hodinách během vývoje kupovité oblačnosti se základnami ve výšce 500 m AAL (nad úrovní letiště). Z tohoto důvodu probíhalo testování pouze do této výšky. Po vzletu z dráhy 30 probíhalo stoupání z důvodu provozu 3 km JZ od letiště. V testovací výšce 350 m při letu směrem od letiště jsem tlačítkem „Recalculate landing“ započal s procedurou přistání podle programu. Během vteřiny se zobrazila požadovaná sestupová trať. Z důvodu relativně malé výšky a velké vzdálenosti program vypočítal pouze zatáčku o 180°, dlouhý rovný úsek do polohy „po větru“ a 3. a 4. zatáčku běžného okruhu. Výška nad sestupovou cestou byla +40 m přesně podle naprogramování. Udržoval jsem rychlost 100 km/h, vertikální rychlost byla 0,5-0,8 m/s. Při prvním přistání jsem striktně dodržoval parametry sestupu podle programu a zároveň porovnával průběh letu se zažitými postupy během běžného létání. Po dotočení finále dráhy 30 jsem pokračoval do výšky 20 m nad zem, kde jsem přistání přerušil a průletem nad letištěm zahájil druhou část testu. Průběh stoupání byl totožný s první částí, nicméně test jsem zahájil

až 500 m AAL. Tentokrát program spočítal mimo okruhu i 360° výhledovou zatáčku nad bodem přistání. Během celé procedury jsem se snažil dodržet poloměr této zatáčky, nicméně rezervu nad sestupovou rovinou jsem si upravil podle svých zvyklostí. V průběhu téměř celého testu jsem byl 50 m nad sestupovou cestou. Přistání proběhlo bez potíží, další testy jsme z časových důvodů již neudělali.

C.3 Hodnocení aplikace

Aplikace má velmi jednoduché a přehledné ovládání, intuitivní indikaci a pohodlné čtení. Podkladová mapa je dostatečně detailní a nabízí dostatek orientačních bodů. Rychlost aplikace je dostatečná, mírnou komplikací jsou sekundové intervaly příjmu GPS signálu. Při ukončení zatáčky často dochází k jejímu přetočení, jelikož správného směru dosáhne letoun dle aplikace až o 1 s později než ve skutečnosti. Nicméně tato vlastnost je běžná i u certifikovaných leteckých GPS navigací a piloti jsou k manipulaci s takovými přístroji cvičeni. Poloměry zatáček jsou správně vypočteny a při nízké rychlosti letu je lze snadno dodržovat. Indikace výšky nad sestupovou rovinou je poněkud nezvykle provedena barevnou škálou červená- zelená- červená. Pro přístrojové přistání používaný systém ILS má indikaci pomocí vertikálně orientované řady teček, které přesně určují velikost rezervy nad / pod sestupovou rovinou. Zobrazení pomocí škály je oproti tomu nejednoznačné, nicméně číselný údaj zobrazený pod ní tuto nevýhodu eliminuje. Toto zobrazení tedy považuji za dostatečné. Rezerva 40 m byla nedostačující, odpovídající hodnota by dle mého odhadu měla být 80 m během všech fází přistání kromě úseku na finále, kde by se měla postupně snižovat na cca 40 m.

C.4 Poznámky a návrhy ke zlepšení

Dle vyjádření autora je aplikace určena pro nouzové situace, například vysazení pohonné jednotky. Průběh letu je namodelován pro hodnotu ideální klouzavosti letounu bez rezerv pro zvýšené opadání v zatáčkách, turbulenci nebo vlivem pilotních chyb. K tomuto účelu slouží nastavitelná rezerva, kterou jsme během letu nastavili na 40 m. Tuto filozofii hodnotím jako nevýhodnou konkrétně pro případ vysazení, či nesprávné funkce jakéhokoli systému v letounu, zvláště pohonné jednotky. Pozornost pilota musí být v těchto případech věnována na udržení rychlosti, vybrání nouzové plochy, pokusu o opětovné nahození motoru a v případě neúspěchu na uzavření přívodu paliva, vypnutí elektrických okruhů a utažení poutacích pásů. Jako pilot si neumím v takových případech představit další pracovní zátěž ve formě aplikace, která mě po předem dané dráze dovede na zem. Pilotní výcvik i průběžná příprava během kariéry každého pilota jsou směřovány tak, aby byl schopen zvládat nouzové situace bez jakékoli aplikace. Naopak varuji před pokusem zavést ta-

kovou aplikaci do praxe v tomto kontextu, jelikož by mohlo dojít ke zvýšení počtu leteckých nehod vlivem pilotních chyb pramenících z kriticky vysokého pracovního zatížení posádky.

Obrovskou výhodou využití této aplikace naopak spatřuji v průběhu jiného druhu mimořádných situací. Současný pokrok v letecké avionice zpřístupnil pilotům UL letounů technologie, které byly donedávna k vidění pouze v kokpitech stíhaček a velkých dopravních letounů. Jedná se o zařízení EFIS, které je integrované do tzv. glass kokpitu. Jedná se o zobrazení parametrů letu a parametrů důležitých systémů letounu do jednoho uceleného zobrazení pomocí LCD displeje. Symbolika a ergonomie tohoto zobrazení je směřována k co nejintuitivnější, nejnázornější a nejpřesněji čitelné formě. Mimo základních parametrů jako je poloha vůči horizontu, letová a vertikální rychlost nebo výška, jsou do tohoto obrazu integrovaná i navigační data většinou získaná ze zařízení GPS.

UL letouny podle předpisů musejí létat za podmínek VFR (Visual Flight Rules). Ty jsou striktně dány – dohlednost 5 km, vzdálenost od oblačnosti 300 m vertikálně a 1500 m horizontálně. K měření těchto hodnot však na palubě letounů nejsou žádné přístroje a tak je hlavním hodnotícím kritériem zkušenost a odhad pilota. Při složitějších meteorologických podmínkách dochází čím dál častěji k vlétnutí letounu VFR do podmínek, ve kterých lze letět jen podle přístrojů (IFR – instrument flight rules). Pokud se do této situace dostane letoun nevybaveného umělým horizontem, nemá téměř šanci situaci bezpečně vyřešit. Pokud ale do ní vlétne např. UL vybavený přístrojem EFIS, poruší předpisy, přesto alespoň trochu zkušený pilot udrží kontrolu nad letounem. Při zachování klidu je schopen doletět nad blízké letiště podle GPS nebo podle pokynů řízení letového provozu, pokud jej kontaktuje. Problém však nastává ve chvíli, kdy se chce ve špatném počasí pokusit o přistání. Bez odpovídajícího přístrojového vybavení letiště a letounu je téměř nemožné se bezpečně dostat na správnou sestupovou rovinu v ose dráhy.

Aplikace UL FMS by v tomto případě mohla zachránit životy. Stálý profil klesání je příznivý pro pilota, který je pod značným tlakem a mohl by během měněního se profilu přistání udělat osudovou chybu. Ideální by byl postup, kdy by aplikace dovedla pilota 360° zatáčkou nad letištěm po 1. zatáčky levého okruhu na výšku přesně 400 m, z ní stálým mírným klesáním přes 2. zatáčku a polohu „po větru“ do 3. zatáčky ve výšce 200-240 m tak, aby od tohoto úseku stálým klesáním došlo k navedení letounu přes 4. zatáčku a finále do bodu 500 m před prahem dráhy ve výšce 50 m. V tomto bodě by měla aplikace vyprodukovat akustický či hlasový povel, při kterém by se pilot musel rozhodnout, zda vidí dráhu a může na ní přistát, nebo ji nevidí a musí postup na přistání zopakovat. Výše popsaná procedura kopíruje postupy používané při přístrojovém přiblížení na přistání a poskytuje pilotovi dostatek prostoru k řízení letounu a navigování zároveň. Popsaný postup platí pro letiště bez překážek, v případě, že v okolí letiště jsou, bylo by nutné výšky v jednotlivých úsecích upravit.

rtm.Bc. Jan Hruška – civilní a vojenský pilot s celkovým náletem 400 hod na větroních, UL a motorových letounech, v letech 2009-2013 studoval na Univerzitě obrany v Brně, obor Vojenský pilot

Instalační příručka

D.0.1 Instalace aplikace na Android zařízení

Pro instalaci aplikace je třeba mít nainstalované GoogleMaps. V případě, že na zařízení nejsou nainstalovány GoogleMaps, aplikace nabídne možnost doinstalace. Je třeba povolit všechny potřebné práva aplikace.

D.0.2 Instalace komunikace NMEA se simulátorem

Instalace NMEA komunikace se simulátorem je odzkoušena na Windows 7 64 bit s využitím plachtařského simulátoru Condor 1.1.5. Pro komunikaci je nutné správně nastavit TCP/IP komunikaci. Následující postup bude psán v bodech. Data pro instalaci jsou k nalezení na CD ve složce */src/sim*.

1. Je třeba nainstalovat *com0com*, všechny instalační možnosti je doporučeno nechat tak jak jsou původně.
2. Je třeba zkopírovat složku *hub4com* nejlépe na *c:/Program files*
3. Je třeba zkopírovat *com2tcp* - *zastupce* například na plochu
4. Je třeba připojit se telefonem přes wifi na stejnou síť jako je PC
5. Je třeba přidat IP adresu mobilního zařízení do spustitelné cesty v *com2tcp* - *zastupce*
6. Nyní je třeba spustit *com2tcp* - *zastupce*
7. Je třeba spustit Condor simulator a v něm nastavit *Options NMEA output true port CNCA0*
8. Nyní je třeba spustit aplikaci *UllFMS.apk* na mobilním zařízení v módu NMEA data a mít správně nastavený komunikační port

9. V Condor simulátoru je třeba spustit let na území České nebo Slovenské republiky.

Formáty používaných NMEA zpráv

V této příloze jsou popsány jednotlivé NMEA zprávy, které jsou použity v implementaci aplikace. Čísla pro každý parametr určují pořadí parametrů v NMEA zprávě oddělených čárkou.

E.0.3 GPGGA

\$GPGGA,hhmmss.ss,llll.ll,a,yyyy.yy,a,x,xx,x.x,x.x,M,x.x,M,x.x,xxxx

1 = UTC of position

2 = latitude of position

3 = N or S

4 = Longitude of position

5 = E or W

6 = GPS Quality indicator (0=no fix, 1=GPS fix, 2=Dif. GPS fix)

7 = number of satellites in use

8 = horizontal dilution of precision

9 = Antenna altitude above mean-sea-level

10 = units of antenna altitude, meters

11 = Geoidal separation

12 = units of geoidal separation, meters

13 = Age of Differential GPS data (seconds)

14 = Differential reference station ID

E.0.4 GPRMC

\$GPRMC,hhmmss.ss,A,llll.ll,a,yyyy.yy,a,x.x,x.x,ddmmyy,x.x,a*hh

- 1 = UTC of position fix
- 2 = Data status (V=navigation receiver warning)
- 3 = Latitude of fix
- 4 = N or S
- 5 = Longitude of fix
- 6 = E or W
- 7 = Speed over ground in knots
- 8 = Track made good in degrees True
- 9 = UT date
- 10 = Magnetic variation degrees
- 11 = E or W
- 12 = Checksum