

ČESKÉ VYSOKÉ UČENÍ TECHNICKÉ V PRAZE

FAKULTA DOPRAVNÍ

Ústav letecké dopravy

Diplomová práce

Bc. Lukáš Umlauf

Aplikace MLAT metody nad sítí low-cost ADS-B přijímačů

Praha 2015



K621..... **Ústav letecké dopravy**

ZADÁNÍ DIPLOMOVÉ PRÁCE (PROJEKTU, UMĚLECKÉHO DÍLA, UMĚLECKÉHO VÝKONU)

Jméno a příjmení studenta (včetně titulů):

Bc. Lukáš Umlauf

Kód studijního programu a studijní obor studenta:

N 3710 – PL – Provoz a řízení letecké dopravy

Název tématu (česky): **Aplikace MLAT metody nad sítí low-cost ADS-B
přijímačů**

Název tématu (anglicky): Application of MLAT Method over the Net of Low-Cost ADS
B Receivers

Zásady pro vypracování

Při zpracování diplomové práce se řiďte osnovou uvedenou v následujících bodech:

- Úvod
- Metody výpočtu polohy cíle pomocí multilaterační metody
- Návrh metody asociace zpráv z jednotlivých přijímacích stanic pro MLAT výpočet v reálném čase
- Návrh funkčního modelu - výpočetního modulu v systému MATLAB realizující multilaterační výpočet nad tokem z ADS-B přijímačů
- Závěr

Rozsah grafických prací: dle pokynů vedoucího diplomové práce

Rozsah průvodní zprávy: minimálně 55 stran textu (včetně obrázků, grafů a tabulek, které jsou součástí průvodní zprávy)

Seznam odborné literatury: NEVEN, W.H.L., T.J. QUILTER, R. WEEDON a R.A. HOGENDOORN. Wide Area Multilateration: Report on EATMP TRS 131/04.
Comparative Assessment of SSR vs. Wide Area Multilateration Eurocontrol. Edition 1.3. September 29, 2005
<http://www.mathworks.com/products/matlab/>

Vedoucí diplomové práce: **Ing. Stanislav Pleninger, Ph.D.**

Datum zadání diplomové práce: **31. července 2014**
(datum prvního zadání této práce, které musí být nejpozději 10 měsíců před datem prvního předpokládaného odevzdání této práce vyplývajícího ze standardní doby studia)

Datum odevzdání diplomové práce: **30. listopadu 2015**
a) datum prvního předpokládaného odevzdání práce vyplývající ze standardní doby studia a z doporučeného časového plánu studia
b) v případě odkladu odevzdání práce následující datum odevzdání práce vyplývající z doporučeného časového plánu studia



.....
doc. Ing. Stanislav Szabo, PhD. MBA
vedoucí
Ústavu letecké dopravy



.....
prof. Dr. Ing. Miroslav Svítek, dr. h. c.
děkan fakulty

Potvrzuji převzetí zadání diplomové práce.



.....
Bc. Lukáš Umlauf
jméno a podpis studenta

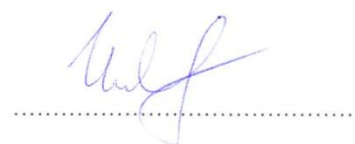
V Praze dne.....30. června 2015

Prohlášení

Prohlašuji, že jsem diplomovou práci vypracoval samostatně a že jsem uvedl veškeré informační zdroje v souladu s metodickým pokynem o etické přípravě vysokoškolských závěrečných prací.

Nemám závažný důvod proti užívání tohoto školního díla ve smyslu § 60 Zákona č.121/2000 Sb. o právu autorském, o právech souvisejících s právem autorským a o změně některých zákonů (autorský zákon).

V Praze dne 30.11.2015



Poděkování

Chci poděkovat vedoucímu diplomové práce, Ing. Stanislavu Pleningerovi, PhD., za cenné připomínky a rady, které mi poskytl při tvorbě této práce. Také všem, kteří mi jakkoli pomohli při řešení dané problematiky. Dále pak Ing. Martinovi Zachovi za spolupráci při školním projektu vytvoření nízkonákladového MLAT systému.

Abstrakt

Diplomová práce se zabývá problematikou vytvoření nízkonákladového MLAT systému. Cílem této práce bylo vytvořit modul, který umožní příjem a zpracování dat z různých přijímačů a následné spočítání polohy cíle. Pouze za pomoci programu pro spojení datových toků byl vytvořen kompletní skript v programu MATLAB. Práce se také zabývá přesností vytvořeného systému pomocí ADS-B zpráv a možností zlepšení výsledků tohoto systému.

Klíčová slova

MLAT, WAM, ADS-B, multilateration, Matlab, Instrument Control Toolbox, Mode S.

Abstract

This thesis deals with the creating the low-cost multilateration system. The main goal is to create the method which allows users to receive and to process the data from several receivers. Then we use the algorithm for calculating the 2D position of the aircraft. The whole script was made in the MATLAB program except the another part which joins the data flows into the one flow. The thesis also deals with the precision of this system thanks to the ADS-B position messages and with possibilities of enhancing this system.

Key words

MLAT, WAM, ADS-B, multilateration, Matlab, Instrument Control Toolbox, Mode S.

OBSAH

Abstrakt	5
Klíčová slova	5
Abstract.....	6
Key words.....	6
1 Úvod	10
2 Multilaterace	11
2.1. Princip.....	11
2.2. Uplatnění.....	12
2.3. Schéma MLAT systému	13
2.4. ADS-B	13
2.5. Vztah multilaterace a ADS-B	14
3 Metody výpočtu polohy cíle pomocí MLAT metody.....	15
3.1. Explicitní výpočet 2D polohy	15
3.1.1 Postup výpočtu	15
3.1.2 Výpočet.....	17
3.2. Nevýhody.....	20
3.2.1 Výška	20
3.2.2 Více možností řešení	20
3.3. Řešení výpočtu v programu Matlab.....	20
3.4. Výpočet 3D polohy (explicitně).....	22
3.5. Iterační metoda	27
3.5.1 Postup výpočtu	27
3.5.2 Nevýhody	28
4 Návrh metody asociace zpráv z jednotlivých přijímacích stanic	29
4.1. Prostředí MATLAB	29

4.1.1	Instrument Control Toolbox	29
4.1.2	Velikost bufferu	29
4.2	Sjednocení datových toků.....	29
4.2.1	Air funnel.....	29
4.2.2	Rozdílný čas zpracování signálů	32
4.3	Vyhledání shodných zpráv	34
4.3.1	Princip vyhledávání	34
4.3.2	Odražené signály	34
4.4.	Dekódování zpráv	35
4.4.1	Časová značka	35
4.4.2	ICAO adresa	36
4.4.3	Downlink formát	36
4.4.4	Obsah jednotlivých zpráv	37
5	Návrh funkčního modelu	39
5.1.	Přijímací stanice.....	39
5.1.1	Popis použitého zařízení.....	39
5.1.2	Pořizovací náklady	40
5.1.3	Webový server.....	41
5.1.4	Výstupní formát dat.....	42
5.1.5	Synchronizace hodin.....	43
5.2.	Rozmístění jednotlivých přijímačů	45
5.2.1	Koleje Strahov	45
5.2.2	Pankrác	46
5.2.3	Fakulta dopravní.....	47
5.2.4	Celkové rozmístění	48
5.3	Modul realizující příjem a výpočet.....	50
5.4	Aplikace MLAT metody na aktuální provoz.....	59

5.4.1	Příjem zpráv.....	59
5.4.2	Odchytky měření	59
5.4.3	Převod z kartézského systému zpět do WGS84	62
5.4.4	Další nepříznivé vlivy.....	63
5.4.5	Výstupy z programu MATLAB	64
5.4.6	Reálné odchytky MLAT systému.....	64
5.4.7	Schopnost programu zpracovávat data z reálného provozu	66
5.4.8	Jednoznačné určení polohy letounu.....	66
5.4.9	Integrita.....	67
5.4.10	Další možnosti pokračování na projektu	67
6	ZÁVĚR.....	69
	SEZNAM POUŽITÝCH ZDROJŮ.....	70
	SEZNAM POUŽITÝCH ZKRATEK	72
	PŘÍLOHY	74
	Funkce pro převod souřadnic	74
	Z WGS 84 do ECEF	74
	Z ECEF do ENU (East, North, Up).....	74
	Převod z ENU do WGS84.....	74
	Přílohy na CD	75

1 Úvod

V dnešní době existuje nemálo radarových systémů, které zjišťují polohu letounu. Většina z nich ale převyšuje svými náklady na výstavbu a provoz mnohamilionové částky. Takové systémy nelze vystavět zejména v obtížně dostupných lokalitách, horských oblastech nebo dokonce na vrcholcích hor. Oproti tomu MLAT systém je mnohem jednodušší na instalaci a mnohonásobně levnější než velké radary. Několik přijímačů je schopno pokrýt oblasti, kde dosud nebyla možnost sledovat letecký provoz. Na malých letištích, kde není dostatek finančních zdrojů, je MLAT systém velmi dobrou variantou pro sledování okolního provozu, protože je možné sledovat i menší letouny, které svou polohu nevysílají v ADS-B zprávách.

Multilaterační metody jsou známy již delší dobu. První WAM (Wide Area Multilateration) systém byl umístěn na ostravské letiště a do jeho okolí v roce 2003. Jedná se o první systém na světě, který byl nasazen do civilního provozu. Tento systém zde umístila společnost ERA se sídlem v Pardubicích. Během několika let společnost instalovala tyto systémy v různých státech světa včetně Velké Británie, Španělska, Číny, Nového Zélandu a Afriky. Největším WAM systémem na světě je soubor 55 stanic, který pokrývá celé území státu Namibie.

MLAT systémy mají na základě dlouhodobých analýz prokazatelně lepší výsledky než radar, a navíc je to velmi přesný nástroj pro určování výšky letounů.

Velkou výhodou této metody je, že není potřeba znát čas přenosu ze zdroje do přijímače. To znamená, že není potřeba přesné synchronizace hodin mezi zdrojem a přijímačem. Musí však být provedena synchronizace hodin všech přijímačů, se kterými budeme multilateraci provádět. Tato metoda nevyžaduje žádnou implementaci další techniky do letounu. V této práci se budu zabývat pasivní multilaterací a její aplikací nad sítí tří ADS-B přijímačů.

2 Multilaterace

2.1. Princip

Letoun emituje pulz elektromagnetické energie, který je vysílán všesměrově. Tento signál je detekován na přijímačích a následně je určena poloha letounu. Předpokládáme konstantní rychlost šíření tohoto pulzu v atmosféře.

Výpočet je založen na rozdílu časů přijatých signálů na jednotlivých stanovištích TDOA (Time Difference Of Arrival). Jelikož se vyslaný signál z letounu šíří rovnoměrně, dorazí k jednotlivým přijímačům se zpožděním. Tyto signály detekujeme na přijímači a následně můžeme spočítat polohu letounu z naměřených časů příjmu. Systém nezná absolutní čas letu signálu atmosférou. Může zjistit pouze čas příchodu na jednotlivé stanice, poté určit rozdíl těchto časů příchodu mezi zvolenými kombinacemi přijímačů a následně spočítat polohu.

Multilaterační systém může zpracovávat různá data vyslaná z vysílače letounu a různě je kombinovat dle potřeby pro dané účely.

MLAT systémy můžeme rozdělit na:

- a) **Pasivní** – systém pouze odposlouchává všechny okolní signály, spoléhá se pouze na zprávy vyslané letounem. Tento systém je výhodný proto, že není potřeba žádné povolení ani oprávnění pro použití. Další výhodou je, že nenarůstá počet dotazů v pásmu 1030MHz ani odpovědí v pásmu 1090MHz.

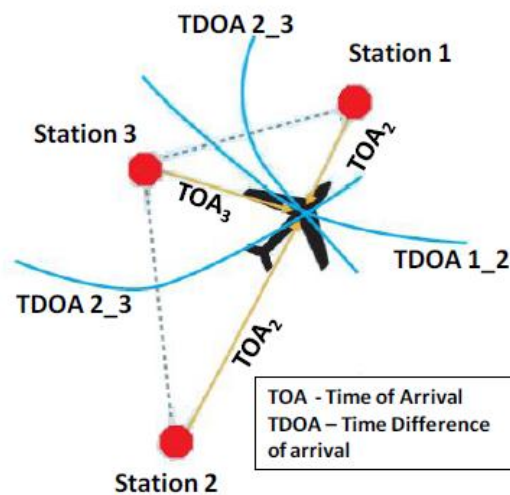
Pasivní MLAT systém může fungovat za předpokladu, že alespoň jedna z následujících možností platí:

- Letoun je vybaven odpovídačem Mode S.
- Letoun je vybaven odpovídačem Mode A/C a je v dosahu alespoň jednoho dotazovače.
- Letoun je vybaven odpovídačem Mode A/C a je v dosahu jiného letounu vybaveného systémem ACAS (Airborne Collision Avoidance System).

Proto je nejvýhodnější používat MLAT systémy například tam, kde se nachází:

- Oblast s větším provozem letadel vybavených ACAS.
- Oblasti s již vybudovanou infrastrukturou sekundárních radarů.
- Oblasti, kde je povinné použití Mode S.

b) **Aktivní** – systém aktivně dotazuje letouny v jeho dosahu. Je to prakticky stejný systém jako pasivní MLAT systém jen s tím rozdílem, že systém si dokáže vyžádat vlastní odpovědi. Dotazovač je v tomto případě mnohem jednodušší než u sekundárních radarů. Není potřeba rotující anténa, ale stačí všesměrová nebo sektorová anténa v závislosti na potřebě pokrytí. Pokrytí ale nedosahuje takových hodnot jako u MSSR.



Obrázek 1: Princip výpočtu polohy pomocí MLAT

2.2. Uplatnění

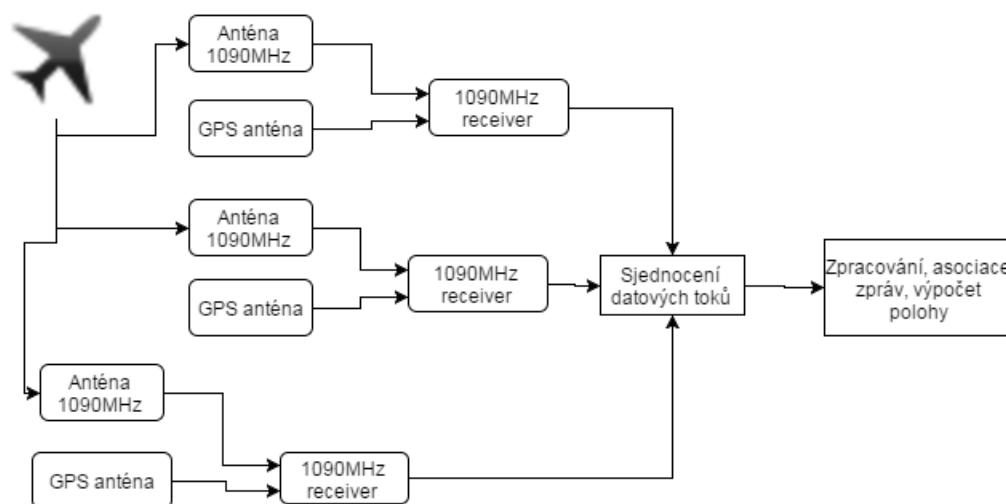
Multilaterace nachází uplatnění při sledování letounu ve všech fázích letu. Pro monitoring letounů na trati se používá Wide Area Multilateration (WAM). Nejlepší využití má zejména v členitém terénu. Jako příklad můžeme uvést rakouské město Innsbruck, kde kvůli členitému terénu nebylo vhodné použití SSR. Kolem města na okolních kopcích bylo nainstalováno devět přijímacích stanic a vytvořen WAM systém. Přesnost tohoto systému dosahuje 10 - 50 metrů. Dle rakouského řízení letového provozu dosahují roční náklady na tento systém 150 000 Eur. Při použití sekundárních radarů by tato částka vzrostla na 13 000 000 Eur [7]. MLAT systémy

jsou také přesný nástroj pro určování výšky. Pomocí pěti stanic je možné určit výšku s přesností dvacet pět metrů.[9]

MLAT systémy nacházejí uplatnění také při sledování vozidel a letounů na letištní ploše. Na větších letištích lze doplnit již instalovaný SMR (Surface Movement Radar), a tak zvýšit přesnost takového systému.

2.3. Schéma MLAT systému

Základní schéma systému, který jsme v rámci projektu vytvořili, vypadá následovně. Letoun vyšle impuls, který zachytí antény. Dále přijímač přidá časovou značku dle synchronizovaného GPS času. Informace putují přes síťové rozhraní do počítače, kde se datové toky spojí a je možné s nimi pracovat.



Obrázek 1: Jednoduché schéma našeho MLAT systému

2.4. ADS-B

Automatic Dependent Surveillance – Broadcast je systém, který bude hlavním zdrojem informací pro řízení letového provozu. Jeho zavádění je již v procesu dle strategií NextGen (Next Generation Air Transportation System) a SESAR (Single European Sky ATM Research). Jde o systém na palubě letounu, který získává svou polohu ze satelitní navigace GNSS a tyto údaje vysílá všesměrově do okolí bez ohledu

na to, zda je někdo přijímá. Tyto informace pak zachytí nejen řízení letového provozu, ale také okolní letadla, která jsou touto technologií vybavena. Hlavní charakteristiky tohoto systému jsou:

- Systém pracuje bez zásahu pilota.
- Navigační veličiny nejsou měřeny ale pouze získávány z GNSS.
- Slouží ke sledování letového prostoru.
- Signál je vysílán všesměrově pro široké okolí.

Pro náš systém jsme použili právě zprávy, které obsahují polohové informace. Je to proto, aby mohla být ověřena přesnost výpočtů a provedena následná opatření k optimalizaci celého systému.

2.5. **Vztah multilaterace a ADS-B**

I když jsou tyto dva systémy zcela odlišné prostředky pro monitorování leteckého provozu, neměli bychom je považovat za konkurenční. MLAT systémy představují vývojový mezikrok mezi SSR a využíváním ADS-B z toho důvodu, že MLAT systémy nevyžadují implementaci nové avioniky do letadel. Multilaterace bude plnit úlohu přímé podpory systému ADS-B, což je například:

- Porovnávání polohy získané pomocí ADS-B a MLAT a tím ověřovat přesnosti obou systémů.
- Kontrola integrity ADS-B, MLAT systémy by se mohly využívat ke shromažďování a analýze dat před tím, než začne docházet ke komplexnějším aplikacím..
- Nahrazení SSR pro letadla, jež nebudou vybavena ADS-B.

3 Metody výpočtu polohy cíle pomocí MLAT metody

3.1. Explicitní výpočet 2D polohy

Výpočet v rovině je založen na řešení soustavy dvou rovnic o dvou neznámých. Geometricky jde o průnik dvou větví hyperbol. Jelikož dostaneme dvě řešení soustavy, musíme vybrat pouze jedno. Obvykle je jedno řešení nesmyslné. I tak se objevují řešení, u nichž není poloha cíle jednoznačná, a proto musíme vybrat řešení, které je v takové poloze, jež odpovídá námi sledované oblasti.

Pro výpočet polohy ve dvou-dimenzionálním prostoru je nutné mít k dispozici minimálně tři přijímací stanice, které zachytí identický impuls vyslaný v jeden časový okamžik z jednoho zdroje. Na přijímač však dostáváme nepřetržitě signály z okolí, a proto je nutné asociovat jednotlivé signály, se kterými budeme výslednou polohu počítat.

3.1.1 Postup výpočtu

3.1.1.1 Převod souřadnic z WGS 84 do ECEF

Aby bylo možné výpočet provést, je nutné převést hodnoty souřadnic polohy přijímačů z geodetického systému WGS 84 do kartézského systému (ECEF-Earth Centered, Earth Fixed). K převodu nám poslouží následující konstanty a vzorce:

Délka hlavní poloosy: $a = 6\,378\,137m$

Délka vedlejší poloosy: $b = 6\,356\,752m$

Zploštění: $f = \frac{a-b}{a}$

Excentricita: $e = \sqrt{2f - f^2}$

Poloměr zakřivení: $N(\phi) = \frac{a}{\sqrt{1-e^2\sin^2\phi}}$

Zeměpisná šířka: ϕ

Zeměpisná délka: λ

Výška nad elipsoidem: h

Pro získání souřadnic (X, Y, Z) je nutné použít pro přepočítání tyto vzorce.

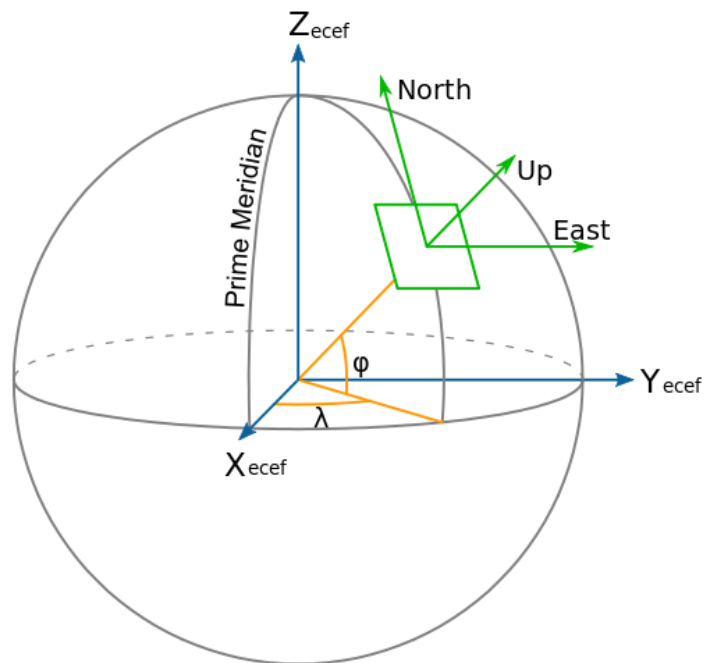
$$X = (N + h) \cos \phi \cos \lambda \quad 3.1$$

$$Y = (N + h) \cos \phi \sin \lambda \quad 3.2$$

$$Z = [N(1 - e^2) + h] \sin \phi \quad 3.3$$

3.1.1.2 Převod do lokálních souřadnic ENU

Tento krok není nezbytně nutný ve 3D multilateraci, protože pouze posune a otočí souřadnou soustavu do místa, které si zvolíme. Například střed této soustavy ztotožníme s polohou jedné naší přijímací stanice. Z vypočtených hodnot lze poté přehledněji určit polohu cíle vůči danému bodu, respektive vůči naší poloze. Ve 2D multilateraci je nutný proto, abychom promítli výslednou polohu na plochu. Pokud bychom nepoužili tuto transformaci, průmět by se nenacházel na správné pozici.



Obrázek 2: Transformace souřadného systému [13]

Pro převod bude potřeba znát souřadnice bodu, které budeme transformovat. Musí být převedeny již do kartézské soustavy $(X_{ECEF}, Y_{ECEF}, Z_{ECEF})$. Dále si zvolíme souřadnice referenčního bodu, do kterého střed soustavy posuneme, tedy zeměpisnou šířku, zeměpisnou délku a výšku. Převod probíhá následovně:

Definujeme si zeměpisnou šířku a délku zvoleného bodu jako referenční.

$$refLat = refLat * \frac{\pi}{180} \quad 3.4$$

$$refLong = refLong * \frac{\pi}{180} \quad 3.5$$

$$e = -\sin(refLong) * (X_{ECEF} - X_r) + \cos(refLat) * (Y_{ECEF} - Y_r) \quad 3.6$$

$$n = -\sin(refLat) * \cos(refLong) * (X_{ECEF} - X_r) - \sin(refLat) * \sin(refLong) * (Y_{ECEF} - Y_r) + \cos(refLat) * (Z_{ECEF} - Z_r) \quad 3.7$$

$$u = \cos(refLat) * \cos(refLong) * (X_{ECEF} - X_r) + \cos(refLat) * \sin(refLong) * (Y_{ECEF} - Y_r) + \sin(refLat) * (Z_{ECEF} - Z_r) \quad 3.8$$

Převodli jsme souřadný systém ECEF na systém se středem v námi zvoleném bodě a dostali jsme souřadnice ve formátu $e = east, n = north, u = up$.

3.1.2 Výpočet

Při výpočtu polohy ve 2D víme, že hledaný cíl bude ležet na průniku dvou hyperbol. Každá hyperbola má ohniska v referenčním přijímači a v jednom vedlejším přijímači. Definujeme si souřadnice polohy vedlejších přijímačů jako: $B_1 = (X_{B_1}, Y_{B_1})$, $B_2 = (X_{B_2}, Y_{B_2})$ a centrálního přijímače jako $C = (X_c, Y_c)$. Známe naměřené hodnoty časů t_1, t_2 a t_c . Dále h_1 a h_2 definujeme jako rozdíl vypočtených vzdáleností polohy cíle od vedlejšího a centrálního přijímače. Víme, že se signál šíří rychlostí světla, proto se $h_1 = (t_1 - t_c) * c_{světla}$ a $h_2 = (t_2 - t_c) * c_{světla}$. Pro polohu cíle zvolíme souřadnice $T = (x, y)$.

Pro tyto hodnoty platí rovnice:

$$h_1 = \sqrt{(x - x_{B_1})^2 + (y - y_{B_1})^2} - \sqrt{(x - x_c)^2 + (y - y_c)^2} \quad 3.9$$

$$h_2 = \sqrt{(x - x_{B_2})^2 + (y - y_{B_2})^2} - \sqrt{(x - x_c)^2 + (y - y_c)^2} \quad 3.10$$

Pro vyřešení soustavy je nutné několika operacemi rovnice upravit. V prvním kroku rovnice přepíšeme do tvaru:

$$h_1 + \sqrt{(x - x_c)^2 + (y - y_c)^2} = \sqrt{(x - x_{B_1})^2 + (y - y_{B_1})^2} \quad 3.11$$

$$h_2 + \sqrt{(x - x_c)^2 + (y - y_c)^2} = \sqrt{(x - x_{B_2})^2 + (y - y_{B_2})^2} \quad 3.12$$

Dále rovnice umocníme a roznásobíme:

$$2xx_c - x_c^2 + 2yy_c - y_c^2 - 2xx_{B_1} + x_{B_1}^2 - 2yy_{B_1} + y_{B_1}^2 - h_1^2 = 2h_1\sqrt{(x - x_c)^2 + (y - y_c)^2} \quad 3.13$$

$$2xx_c - x_c^2 + 2yy_c - y_c^2 - 2xx_{B_2} + x_{B_2}^2 - 2yy_{B_2} + y_{B_2}^2 - h_2^2 = 2h_2\sqrt{(x - x_c)^2 + (y - y_c)^2} \quad 3.14$$

Pro lepší orientaci ve výpočtu zavedeme konstanty:

$$L_1 = x_{B_1}^2 + y_{B_1}^2 - x_c^2 - y_c^2 \quad 3.15$$

$$L_2 = x_{B_2}^2 + y_{B_2}^2 - x_c^2 - y_c^2 \quad 3.16$$

Dosadíme a upravíme výrazy:

$$2x(x_c - x_{B_1}) + 2y(y_c - y_{B_1}) + L_1 - h_1^2 = 2h_1\sqrt{(x - x_c)^2 + (y - y_c)^2} \quad 3.17$$

$$2x(x_c - x_{B_2}) + 2y(y_c - y_{B_2}) + L_2 - h_2^2 = 2h_2\sqrt{(x - x_c)^2 + (y - y_c)^2} \quad 3.18$$

Rovnice od sebe odečteme a dostaneme:

$$x[h_2(x_c - x_{B_1}) + h_1(x_{B_2} - x_c)] + y[h_2(y_c - y_{B_1}) + h_1(y_{B_2} - y_c)] + \frac{(h_2L_1 - h_2h_1^2 - h_1L_2 - h_1h_2^2)}{2} = 0 \quad 3.19$$

Zavedeme další konstanty:

$$a = h_2(x_c - x_{B_1}) + h_1(x_{B_2} - x_c) \quad 3.20$$

$$b = h_2(y_c - y_{B_1}) + h_1(y_{B_2} - y_c) \quad 3.21$$

$$c = \frac{(h_2L_1 - h_2h_1^2 - h_1L_2 - h_1h_2^2)}{2} \quad 3.22$$

Po dosazení konstant dostaneme z rovnic hyperbol lineární rovnici:

$$ax + by + c = 0 \quad 3.23$$

Přepíšeme rovnici do směrnicového tvaru:

$$y = ex + f, \quad e = -\frac{a}{b}, \quad f = -\frac{c}{b} \quad 3.24$$

Rovnice 2.1.3.16 dosadíme do rovnice 2.1.3.9 a dostaneme:

$$2x(x_c - x_{B1}) + 2(ex + f)(y_c - y_{B1}) + L_1 - h_1^2 = 2h_1\sqrt{(x - x_c)^2 + (ex + f - y_c)^2} \quad 3.25$$

Rovnici upravíme a vytkneme x:

$$\begin{aligned} & 2x(x_c - x_{B1} + e(y_c - y_{B1})) + 2f(y_c - y_{B1}) + L_1 - h_1^2 = \\ & 2h_1\sqrt{x^2 - 2xx_c + x_c^2 + e^2x^2 + f^2 + y_c^2 + 2efx - 2exy_c - 2fy_c} \end{aligned} \quad 3.26$$

Zavedeme další konstanty:

$$M = x_c - x_{B1} + e(y_c - y_{B1}), \quad N = f(y_c - y_{B1}) + L_1 - h_1^2 \quad 3.27$$

Do rovnice dosadíme členy M a N a rovnici umocníme:

$$4M^2x^2 + 4MNx + N^2 = 4h_1^2(1 + e^2)x^2 + 8h_1^2(-x_c + ef - ey_c)x + 4h_1^2(x_c^2 + f^2 + y_c^2 - 2fy_c) \quad 3.28$$

Změníme členy na pravé straně na konstanty:

$$P = 4h_1^2(1 + e^2) \quad 3.29$$

$$Q = 8h_1^2(-x_c + ef - ey_c) \quad 3.30$$

$$R = 4h_1^2(x_c^2 + f^2 + y_c^2 - 2fy_c) \quad 3.31$$

Dostaneme rovnici:

$$4Mx^2 + 4MNx + N^2 = Px^2 + Qx + R \quad 3.32$$

Znovu přepíšeme konstanty:

$$A = 4M^2 - P \quad 3.33$$

$$B = 4MN - Q \quad 3.34$$

$$C = N^2 - R \quad 3.35$$

Při dosazení dostaneme kvadratickou rovnici:

$$Ax^2 + Bx + C = 0 \quad 3.36$$

Kořeny této rovnice jsou:

$$x_1 = \frac{-B + \sqrt{B^2 - 4AC}}{2A} \quad 3.37$$

$$x_2 = \frac{-B - \sqrt{B^2 - 4AC}}{2A} \quad 3.38$$

Kořeny dosadíme do rovnice 2.1.3.16 a dostaneme odpovídající hodnoty y :

$$y_1 = ex_1 + f, \quad y_2 = ex_2 + f \quad 3.39$$

3.2. Nevýhody

3.2.1 Výška

Pokud máme k dispozici pouze tři přijímače, můžeme zjistit polohu jenom ve 2D. Ve výpočtu také zanedbáváme rozdíl přímé vzdálenosti a vzdálenosti, která je rovna vzdálenosti při projekci dvou daných bodů na rovinu. Pokud bychom chtěli zjistit výšku, bylo by možné to zjistit ze zpráv Mode C, které obsahují barometrickou výšku daného letounu v daný okamžik.

Výška tedy také značně ovlivní výsledky. Pro potřeby transformace je nutné znát všechny tři souřadnice. Proto si musíme zvolit hladinu, do které budeme letouny usazovat. Tento krok nám výslednou polohu znehodnotí více, ale při použití tří přijímacích stanic je nutný.

3.2.2 Více možností řešení

Při výpočtu nám vyjdou dvě matematicky správná řešení. Pokud máme přijímače správně rozmístěny, mělo by být možné tyto varianty rozlišit v závislosti na sledované oblasti. Při 3D multilateraci vypočteme druhé řešení výsledné pozice se zápornou výškou, nebo naopak s výškou, která by nebyla reálná, a proto by měla být výsledná poloha jasná.

3.3. Řešení výpočtu v programu Matlab

Pro výpočet je nutné znát polohu přijímacích stanic, jež je převedena do kartézského systému a dále pak do lokálních souřadnic. Je nutné znát i výšku přijímačů pro

získání přesné transformace. Níže zvolené souřadnice značí polohu přijímačů v souřadnicích XYZ:

```
xc=P1_x;
yc=P1_y;
zc=P1_z;
```

```
xb1=P2_x;
yb1=P2_y;
zb1=P2_z;
```

```
xb2=P3_x;
yb2=P3_y;
zb2=P3_z;
```

Rozdíl vzdáleností mezi cílem a hlavním přijímačem a cílem a vedlejším přijímačem:

```
h1=((t2-t1))*c0;
```

```
h2=((t3-t1))*c0;
```

Samotný modul pro výpočet polohy. Odvození viz. Kapitola 2.1.2:

```
L1=(xb1^2)+(yb1^2)-(xc^2)-(yc^2);
```

```
L2=(xb2^2)+(yb2^2)-(xc^2)-(yc^2);
```

```
a=(h2*(xc-xb1)+(h1*(xb2-xc)));
```

```
b=(h2*(yc-yb1)+(h1*(yb2-yc)));
```

```
c=((h2*L1)-(h2*(h1^2))-(h1*L2)+(h1*(h2^2)))/2;
```

```
e=-(a/b);
```

```
f=-(c/b);
```

```
P=4*(h1^2)*(1+e^2);
```

```
Q=8*(h1^2)*(-xc+(e*f)-e*yc);
```

```
R=4*(h1^2)*(xc^2+f^2+yc^2-(2*f*yc));
```

```
M=xc-xb1+(e*(yc-yb1));
```

```
N=(2*f*(yc-yb1))+L1-(h1^2);
```

```
A=4*(M^2)-P;
```

```
B=(4*M*N)-Q;
```

```
C=(N^2)-R;
```

Dvojice kořenů, jeden z nich udává pozici námi sledovaného cíle:

```
x1=(-B+(sqrt((B^2)-4*A*C)))/(2*A);
```

```
x2=(-B-(sqrt((B^2)-4*A*C)))/(2*A);
```

```
y1=(e*x1)+f;
```

```
y2=(e*x2)+f;
```

Kořeny rovnice jsou opět v lokálních souřadnicích, takže je nutné je převést zpět do systému WGS84. Problém nastane zde, protože nelze převést pouze souřadnice X a Y. Proto musíme třetí souřadnici zvolit, čímž se nám zvýší odchylka výpočtu.

3.4. Výpočet 3D polohy (explicitně)

Pro 3-dimenzionální výpočtový mechanismus budeme vycházet z téměř totožných rovnic jako při výpočtu ve 2D. Provedeme pouze tu změnu, že přidáme třetí souřadnici. Cíl už se nebude nacházet na průniku dvou hyperbol, ale na průniku tří hyperboloidů. Musíme vyřešit následující soustavu rovnic:

Znovu si definujeme souřadnice polohy přijímačů jako: $B_i = (x_i, y_i, z_i)$, $B_j = (x_j, y_j, z_j)$, $B_k = (x_k, y_k, z_k)$, $B_l = (x_l, y_l, z_l)$. Dále hodnoty h pro které platí $h_{ab} = h_a - h_b$. Totožně jako v případě pro 2D výpočet (viz kap. 3.1.2) Pro polohu cíle zvolíme souřadnice $T = (x, y, z)$:

$$h_{ij} = \sqrt{(x - x_i)^2 + (y - y_i)^2 + (z - z_i)^2} - \sqrt{(x - x_j)^2 + (y - y_j)^2 + (z - z_j)^2} \quad (3.40)$$

$$h_{ik} = \sqrt{(x - x_i)^2 + (y - y_i)^2 + (z - z_i)^2} - \sqrt{(x - x_k)^2 + (y - y_k)^2 + (z - z_k)^2} \quad (3.41)$$

$$h_{kj} = \sqrt{(x - x_k)^2 + (y - y_k)^2 + (z - z_k)^2} - \sqrt{(x - x_j)^2 + (y - y_j)^2 + (z - z_j)^2} \quad (3.42)$$

$$h_{kl} = \sqrt{(x - x_k)^2 + (y - y_k)^2 + (z - z_k)^2} - \sqrt{(x - x_l)^2 + (y - y_l)^2 + (z - z_l)^2} \quad (3.43)$$

Následujících několik úprav provedu vzhledem k jejich podobnosti pouze pro první rovnici. Mění se jen dolní indexy u jednotlivých členů.

Upravíme rovnici do tvaru:

$$h_{ij} - \sqrt{(x_i - x)^2 + (y_i - y)^2 + (z_i - z)^2} = -\sqrt{(x_j - x)^2 + (y_j - y)^2 + (z_j - z)^2} \quad (3.44)$$

Dále rovnici umocníme:

$$h_{ij}^2 - 2h_{ij}\sqrt{(x_i - x)^2 + (y_i - y)^2 + (z_i - z)^2} + (x_i - x)^2 + (y_i - y)^2 + (z_i - z)^2 = \\ (x_j - x)^2 + (y_j - y)^2 + (z_j - z)^2 \quad (3.45)$$

Roznásobíme jednotlivé členy, eliminujeme x^2, y^2, z^2 a dostaneme:

$$h_{ij}^2 - 2h_{ij}\sqrt{(x_i - x)^2 + (y_i - y)^2 + (z_i - z)^2} + x_i^2 - 2xx_i + y_i^2 - 2yy_i + z_i^2 - 2zz_i = \\ x_j^2 - 2xx_j + y_j^2 - 2yy_j + z_j^2 - 2zz_j \quad (3.46)$$

Upravíme výraz na:

$$\sqrt{(x_i - x)^2 + (y_i - y)^2 + (z_i - z)^2} = [h_{ij}^2 + x_i^2 - x_j^2 + y_i^2 - y_j^2 + z_i^2 - z_j^2 + 2xx_j - 2xx_i - 2y_iy + 2y_jy - 2zz_i + 2zz_j]/2h_{ij} \quad (3.47)$$

Zjednodušíme výraz nahrazením $x_i - x_j$ za x_{ij} atd. Všechny zbývající rovnice budou vypadat následovně:

$$\sqrt{(x_i - x)^2 + (y_i - y)^2 + (z_i - z)^2} = [h_{ij}^2 + x_i^2 - x_j^2 + y_i^2 - y_j^2 + z_i^2 - z_j^2 + 2x_{ji}x + 2y_{ji}y + 2z_{ji}z]/2h_{ij} \quad (3.48)$$

$$\sqrt{(x_i - x)^2 + (y_i - y)^2 + (z_i - z)^2} = [h_{ik}^2 + x_i^2 - x_k^2 + y_i^2 - y_k^2 + z_i^2 - z_k^2 + 2x_{ki}x + 2y_{ki}y + 2z_{ki}z]/2h_{ik} \quad (3.49)$$

$$\sqrt{(x_k - x)^2 + (y_k - y)^2 + (z_k - z)^2} = [h_{kj}^2 + x_k^2 - x_j^2 + y_k^2 - y_j^2 + z_k^2 - z_j^2 + 2x_{jk}x + 2y_{jk}y + 2z_{jk}z]/2h_{ij} \quad (3.50)$$

$$\sqrt{(x_k - x)^2 + (y_k - y)^2 + (z_k - z)^2} = [h_{kl}^2 + x_k^2 - x_l^2 + y_k^2 - y_l^2 + z_k^2 - z_l^2 + 2x_{lk}x + 2y_{lk}y + 2z_{lk}z]/2h_{ij} \quad (3.51)$$

Rovnice jsou nyní v požadovaném tvaru. Po umocnění vzniknou rovnice protínajících se hyperboloidů. Pokud rovnice (3.12) a (3.13) odečteme, úpravami dostaneme (3.16):

$$\begin{aligned} & \frac{[h_{ij}^2 + x_i^2 - x_j^2 + y_i^2 - y_j^2 + z_i^2 - z_j^2 + 2x_{ji}x + 2y_{ji}y + 2z_{ji}z]}{2h_{ij}} \\ &= \frac{[h_{ik}^2 + x_i^2 - x_k^2 + y_i^2 - y_k^2 + z_i^2 - z_k^2 + 2x_{ki}x + 2y_{ki}y + 2z_{ki}z]}{2h_{ik}} \end{aligned} \quad (3.52)$$

Po úpravách dostaneme obecnou rovnici roviny $y = Ax + Bz + C$,

$$\begin{aligned} & x[h_{ij}x_{ki} - h_{ik}x_{ji}] + y[h_{ij}y_{ki} - h_{ik}y_{ji}] + z[h_{ij}z_{ki} - h_{ik}z_{ji}] \\ &= \frac{h_{ik}[h_{ij}^2 + x_i^2 - x_j^2 + y_i^2 - y_j^2 + z_i^2 - z_j^2]}{2} - \frac{h_{ij}[h_{ik}^2 + x_i^2 - x_k^2 + y_i^2 - y_k^2 + z_i^2 - z_k^2]}{2} \end{aligned} \quad (3.53)$$

kde

$$\begin{aligned}
A &= -\frac{h_{ij}x_{ki} - h_{ik}x_{ji}}{h_{ij}y_{ki} - h_{ik}y_{ji}} \\
B &= -\frac{h_{ij}z_{ki} - h_{ik}z_{ji}}{h_{ij}y_{ki} - h_{ik}y_{ji}} \\
C &= \frac{h_{ik}[h_{ij}^2 + x_i^2 - x_j^2 + y_i^2 - y_j^2 + z_i^2 - z_j^2] - h_{ij}[h_{ik}^2 + x_i^2 - x_k^2 + y_i^2 - y_k^2 + z_i^2 - z_k^2]}{2[h_{ij}y_{ki} - h_{ik}y_{ji}]}
\end{aligned} \tag{3.54}$$

Obdobně provedeme tuto operaci u rovnic (3.14) a (3.15) a dostaneme další rovnici roviny s vyjádřením $y = Dx + Ez + F$, kde

$$\begin{aligned}
D &= -\frac{h_{kj}x_{lk} - h_{kl}x_{jk}}{h_{kj}y_{lk} - h_{kl}y_{jk}} \\
E &= -\frac{h_{kl}z_{jk} - h_{kj}z_{lk}}{h_{kj}y_{lk} - h_{kl}y_{jk}} \\
F &= \frac{h_{kl}[h_{kj}^2 + x_k^2 - x_j^2 + y_k^2 - y_j^2 + z_k^2 - z_j^2] - h_{kj}[h_{kl}^2 + x_k^2 - x_l^2 + y_k^2 - y_l^2 + z_k^2 - z_l^2]}{2[h_{kj}y_{lk} - h_{kl}y_{jk}]}
\end{aligned} \tag{3.55}$$

Porovnáním výrazů rovnic ploch dostaneme rovnici:

$$Ax + Bz + C = Dx + Ez + F \tag{3.56}$$

Vyjádríme x :

$$x = Gz + H \tag{3.57}$$

kde

$$\begin{aligned}
G &= \frac{E - B}{A - D} \\
H &= \frac{F - C}{A - D}
\end{aligned} \tag{3.58}$$

Pokud hodnoty dosadíme zpět do rovnice $y = Ax + Bz + C$, dostaneme vyjádření hodnoty y jako:

$$y = A(Gz + H) + Bz + C \quad (3.59)$$

$$y = Iz + J \quad (3.60)$$

kde

$$I = AG + B$$

$$J = AH + C$$

(3.61)

Rovnice (3.21) a (3.24) vyjadřující hodnoty x a y dosadíme zpět do rovnice (3.12):

$$2h_{ik}\sqrt{(x_i - (Gz + H))^2 + (y_i - (Iz + J))^2 + (z_i - z)^2} = [h_{ik}^2 + x_i^2 - x_k^2 + y_i^2 - y_k^2 + z_i^2 - z_k^2 + 2x_{ki}(Gz + H) + 2y_{ki}(Iz + J) + 2z_{ki}z] \quad (3.62)$$

Výraz roznásobíme:

$$2h_{ik}\sqrt{(G^2z^2 - 2Gz(x_i - H) + (x_i - H)^2) + (I^2z^2 - 2Iz(y_i - J) + (y_i - J)^2) + (z^2 - 2zz_i - z_i^2)} = Lz + K \quad (3.63)$$

kde

$$K = h_{ik}^2 + x_i^2 - x_k^2 + y_i^2 - y_k^2 + z_i^2 - z_k^2 + 2x_{ki}H + 2y_{ki}J \quad (3.64)$$

$$L = 2[x_{ki}G + y_{ki}I + 2z_{ki}] \quad (3.65)$$

Po umocnění a upravení dostaneme:

$$4h_{ik}^2[(G^2z^2 - I^2z^2 + z^2 - 2Gz(x_i - H) - 2Iz(y_i - J) - 2zz_i + (x_i - H)^2 + (y_i - J)^2 + z_i] = L^2z^2 + 2KLz + K^2 \quad (3.66)$$

$$4h_{ik}^2[G^2 - I^2 + 1]z^2 - 8h_{ik}^2[G(x_i - H) - I(y_i - J) - z_i]z + 4h_{ik}^2[(x_i - H)^2 + (y_i - J)^2 + z_i] = L^2z^2 + 2KLz + K^2 \quad (3.67)$$

Odečteme pravou stranu od levé a získali jsme výraz:

$$Mz^2 - Nz + O = 0 \quad (3.68)$$

kde

$$M = 4h_{ik}^2[G^2 + I^2 + 1] - L^2 \quad (3.69)$$

$$N = 8h_{ik}^2 [G(x_i - H) - I(y_i - J) - z_i] + 2LK \quad (3.70)$$

$$O = 4h_{ik}^2 [(x_i - H)^2 + (y_i - J)^2 + z_i] + K^2 \quad (3.71)$$

Poté můžeme napsat řešení pro z jako:

$$z = \frac{N}{2M} \pm \sqrt{\left(\frac{N}{2M}\right)^2 - \frac{O}{M}} \quad (3.72)$$

Souřadnici „ z “ pak dosadíme zpět do rovnic 3.21 a 3.24, abychom získali souřadnice x a y .

3.5. Iterační metoda

Polohu cíle můžeme řešit i iteračně. Jde však o zdlouhavější metodu z výpočetního hlediska. Jedná se o opakování výpočetního procesu za účelem zpřesňování výsledku. Po určitém počtu opakování cyklu se výsledek nemění, nebo alespoň ne o hodnotu, která by měla na výpočet vliv a kterou můžeme zanedbat.

3.5.1 Postup výpočtu

Výpočet budeme realizovat v lokálním systému souřadnic ENU (viz kapitola 3.1.1). Pro názornost si zavedeme hodnoty pro počáteční polohu, od které se bude výpočet odvíjet: $K = (K_x, K_y, K_z)$, polohu n -tého přijímače $P_n = (P_{xn}, P_{yn}, P_{zn})$. Hodnota R nám udává skutečnou vzdálenost od počátečního bodu k přijímači. Pro n přijímačů bude třeba n rovnic.

$$R_n = \sqrt{(K_x - P_{xn})^2 + (K_y - P_{yn})^2 + (K_z - P_{zn})^2}$$

Dále zavedeme matici A . Matice bude nabývat rozměrů $(n - 1) * 3$, kde n je počet přijímačů. Minimální požadavek jsou však tři. Pro větší počet přijímačů plyne z posloupnosti členů další návaznost řádků.

$$A = \begin{bmatrix} \frac{K_x - P_{x1}}{R_1} - \frac{K_x - P_{x2}}{R_2} & \frac{K_y - P_{y1}}{R_1} - \frac{K_y - P_{y2}}{R_2} & \frac{K_z - P_{z1}}{R_1} - \frac{K_z - P_{z2}}{R_2} \\ \frac{K_x - P_{x1}}{R_1} - \frac{K_x - P_{x3}}{R_3} & \frac{K_y - P_{y1}}{R_1} - \frac{K_y - P_{y3}}{R_3} & \frac{K_z - P_{z1}}{R_1} - \frac{K_z - P_{z3}}{R_3} \\ \vdots & \vdots & \vdots \end{bmatrix}$$

V dalším kroku spočítáme matici B , která nám určí velikost odchylky od počáteční polohy K . Hodnota t_n značí čas příchodu na n -tou stanici. Hodnota c značí rychlost světla.

$$B = A / \begin{bmatrix} c * (t_2 - t_1) - R_1 + R_2 \\ c * (t_3 - t_1) - R_1 + R_3 \\ \vdots \end{bmatrix}$$

Vznikne nám tříprvková matice, jejíž příslušné hodnoty přičteme k souřadnicím počáteční polohy K. To znamená, že vznikne nová počáteční poloha K:

$$K = (K_x + B_{1,1}, K_y + B_{2,1}, K_z + B_{3,1})$$

Tento krok opakujeme, dokud se hodnota výsledku znatelně nemění.

3.5.2 Nevýhody

3.5.2.1 Počáteční poloha

Hlavní nevýhodou použití tohoto algoritmu je znalost alespoň přibližné polohy, od které bude moci výpočetní program začít počítat hledanou polohu. Pro výpočet již známé polohy z námi naměřeného souboru dat můžeme použít reálnou polohu znehodnocenou o chybu. Pro získání počáteční polohy můžeme použít výpočet pro získání 2D polohy a iterační metodou tento výpočet doplnit.

3.5.2.2 Vyšší náročnost na výpočet

Jelikož je tato metoda založena na stálém zpřesňování výsledku stálým opakováním cyklu, klade větší nároky na výkon počítače, a tím i na rychlost výpočtů.

3.5.2.3 Neznámá výška pro 2D výpočet

Jak bylo zmíněno výše, při použití tří přijímačů není možné získat výšku ani při použití iteračních algoritmů.

4 Návrh metody asociace zpráv z jednotlivých přijímacích stanic

Pro správné vypočítání polohy je nutné správně a efektivně vyhledat z každého přijímače zprávu, která náleží vyslanému signálu v jeden okamžik z daného letounu. Pro tuto činnost jsme zvolili program MATLAB, který nejenže zprávy vyhledá, ale také vypočte polohu cíle.

4.1. Prostředí MATLAB

4.1.1 Instrument Control Toolbox

Podmínkou pro přijímání zpráv je nutné mít v programu implementován Instrument Control toolbox. Jde o doplněk, který nám mimo jiné umožní kontrolovat a řídit různé typy zařízení přes protokoly TCP, TCP/IP a UDP. Obsahuje funkce pro přenášení dat přímo z našeho zařízení do prostředí Matlab. Tento toolbox není nutný, pokud je uživatel schopen vytvořit modul v jazyce Java, se kterým Matlab dokáže spolupracovat.

4.1.2 Velikost bufferu

Instrumental control toolbox také přináší možnost vytvořit si buffer pro příchozí data. Vzhledem k tomu, že již jeden buffer (viz níže) obsahuje program pro sjednocení zpráv do jednoho toku, není tato funkce úplně nutná. Primárně je však velikost bufferu nastavena na 32 768 bytů.

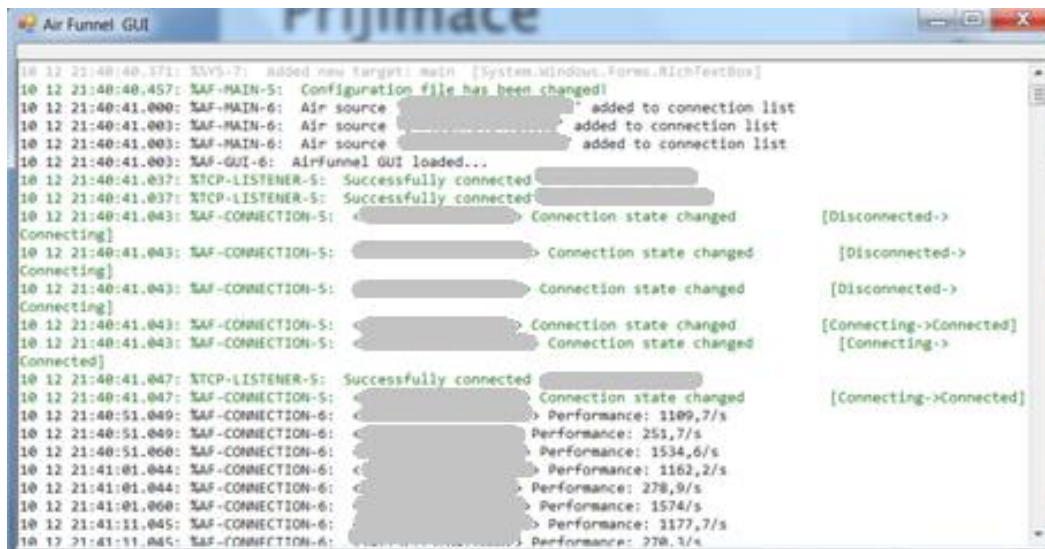
4.2 Sjednocení datových toků

Pro práci s větším počtem datových toků se program MATLAB jevil jako nevhodný. Z důvodu nemožnosti přijmout současně zprávy ze všech přijímačů bylo nutné sjednotit datové toky do jednoho. Proto také byla potřeba označit jednotlivé zprávy indexem, v našem případě IP adresou, jež jednoznačně určí, ze které stanice byla zpráva odeslána. S takto sjednocenými zprávami lze poté pracovat daleko jednodušeji.

4.2.1 Air funnel

Pro naše potřeby byl vytvořen program Air Funnel, který je schopen sjednotit zprávy z více toků, aby s nimi bylo možné dále pracovat. Program je napsán

v programovacím jazyce Visual Basic.NET. Dokáže sjednotit nespočet datových toků do jednoho. V našem případě postačí tři (v budoucnu lze rozšířit).



Obrázek 3 Ukázka připojení pomocí Air funnelu

4.2.1.1 Nesprávně přijaté zprávy

Při správném přijetí zprávy přijímačem je první znak zprávy označen znakem zavináče "@". Nesprávně přijaté zprávy mají jako první znak řetězce otazník "?". Takových zpráv se u přijímačů s dobrým umístěním nevyskytuje mnoho, ale pokud je přijímač z některé strany zastíněn, můžeme jich pozorovat více. Air funnel tyto zprávy automaticky maže, aby nebylo zbytečně ztíženo následné zpracování dat. V praxi lze naměřit u dobře položeného přijímače přibližně dvě až deset chybných zpráv z tisíce. U přijímače, který se nachází například v zastavěné oblasti a má zakrytý příjem z některého směru, se příjem chybných zpráv zvýší na hodnoty přesahující dvacet zpráv z tisíce.

```
@53407A2896938D72867258B3F47CB6B50965D682;  
@53407B27B2415D728672B20CC2;  
@53407B78E27E8D44D06899456E1EC0089ACE4A9D;  
?53407B9AC9D1360000000000000000000000000000;  
@534082C5D0D45D728672B20CC2;  
@53408351630D5D484F6CA47350;  
@534083A1B4D28D750293EA4478652B5C08F79BBE;
```

Obrázek 4: Chybně přijatá zpráva

4.2.1.2 Buffer zpráv

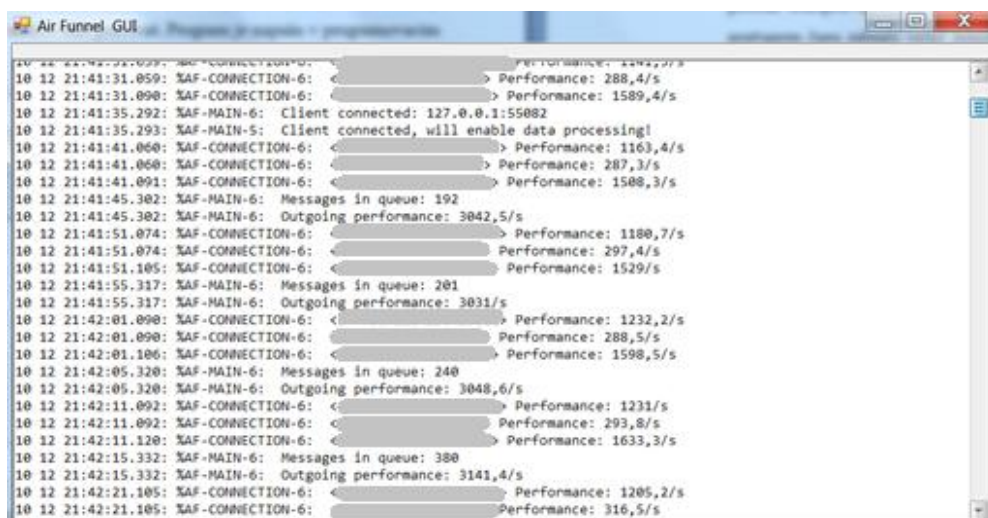
Samotný program Air Funnel má primárně nastavenou hodnotu pěti tisíc zpráv, které může ukládat do bufferu. Tuto hodnotu je možné nastavit na daleko vyšší

, ale pokud nebude počítač schopen zpracovávat daný tok, nemá smysl uchovávat více zpráv, které by postupem času zabraly velké množství operační paměti, až by nakonec stejně došlo k přetečení. Při odpojení klienta je buffer automaticky vymazán, aby se při připojení začalo pracovat s reálným datovým tokem a výpočetní program nebyl zatížen daty z bufferu.

Pokud nastane situace, že výpočetní program není schopen zpracovávat přijímaná data, Air funnel zobrazí počet zpráv, které čekají na odeslání. V případě překročení limitu uložených zpráv se začnou přijaté zprávy mazat, aby nedocházelo k neúnosnému narůstání objemu nezpracovaných dat.

4.2.1.3 Informace o příjmu a odchodu zpráv

Program poskytuje každých deset sekund od připojení na přijímač informace o počtu zpráv, které přijal. V případě, že není připojen žádný klient, program poskytne pouze informaci o přijatých zprávách, aniž by některou z nich uložil. Po připojení klienta se začnou zobrazovat informace o počtu zpráv, které čekají na odeslání, a také o tom, jak rychle je schopen cílový výpočetní program nebo klient zpracovávat daný tok zpráv. V našem případě se zobrazí na začátku operování s daty velký přijímací výkon, který se časem ustálí. Je to kvůli bufferu, s nímž pracuje také Matlab. Čím větší buffer je v Matlabu nastaven, tím delší čas na ustálení odesílací rychlosti je potřeba. Pokud je však datový tok nadále tak velký, že data nelze stíhat zpracovávat, program upozorní na stávající problém a nezpracované zprávy vymaže.



```
10 12 21:41:31.029: SAF-CONNECTION-6: <[redacted]> Performance: 1142,3/s
10 12 21:41:31.059: SAF-CONNECTION-6: <[redacted]> Performance: 288,4/s
10 12 21:41:31.090: SAF-CONNECTION-6: <[redacted]> Performance: 1589,4/s
10 12 21:41:35.292: SAF-MAIN-6: Client connected: 127.0.0.1:55882
10 12 21:41:35.293: SAF-MAIN-5: Client connected, will enable data processing!
10 12 21:41:41.060: SAF-CONNECTION-6: <[redacted]> Performance: 1163,4/s
10 12 21:41:41.060: SAF-CONNECTION-6: <[redacted]> Performance: 287,3/s
10 12 21:41:41.091: SAF-CONNECTION-6: <[redacted]> Performance: 1508,3/s
10 12 21:41:45.302: SAF-MAIN-6: Messages in queue: 192
10 12 21:41:45.302: SAF-MAIN-6: Outgoing performance: 3042,5/s
10 12 21:41:51.074: SAF-CONNECTION-6: <[redacted]> Performance: 1180,7/s
10 12 21:41:51.074: SAF-CONNECTION-6: <[redacted]> Performance: 297,4/s
10 12 21:41:51.105: SAF-CONNECTION-6: <[redacted]> Performance: 1529/s
10 12 21:41:55.317: SAF-MAIN-6: Messages in queue: 201
10 12 21:41:55.317: SAF-MAIN-6: Outgoing performance: 3031/s
10 12 21:42:01.090: SAF-CONNECTION-6: <[redacted]> Performance: 1232,2/s
10 12 21:42:01.090: SAF-CONNECTION-6: <[redacted]> Performance: 288,5/s
10 12 21:42:01.106: SAF-CONNECTION-6: <[redacted]> Performance: 1598,5/s
10 12 21:42:05.320: SAF-MAIN-6: Messages in queue: 240
10 12 21:42:05.320: SAF-MAIN-6: Outgoing performance: 3048,6/s
10 12 21:42:11.092: SAF-CONNECTION-6: <[redacted]> Performance: 1231/s
10 12 21:42:11.092: SAF-CONNECTION-6: <[redacted]> Performance: 293,8/s
10 12 21:42:11.120: SAF-CONNECTION-6: <[redacted]> Performance: 1633,3/s
10 12 21:42:15.332: SAF-MAIN-6: Messages in queue: 300
10 12 21:42:15.332: SAF-MAIN-6: Outgoing performance: 3141,4/s
10 12 21:42:21.105: SAF-CONNECTION-6: <[redacted]> Performance: 1205,2/s
10 12 21:42:21.105: SAF-CONNECTION-6: <[redacted]> Performance: 316,5/s
```

Obrázek 5: Ukázka ukládání zpráv do bufferu, zprávy čekající na odeslání a rychlost odebírání zpráv

4.2.1.4 Přiřazení IP adresy

Vyhledávání zpráv by bylo možné i přes neznalost stanice, z které byla zpráva odeslána. To bychom ale nemohli určit polohu letounu pomocí multilaterace. Dostali bychom mnoho kombinací poloh, z nichž lze polohu jen odhadovat. Před každou zprávou je proto programem připsána IP adresa přijímače, na kterém byla zpráva zachycena. Je to nejspíše nejjednodušší cesta, kdy se z konfiguračního XML dokumentu připiše pouze IP adresa jako jednoznačný identifikátor a nemusí se nic víc indexovat. Při změně stanoviště přijímače se pouze změní v nastavení adresa připojení a program začne pracovat.

4.2.1.5 Spouštění a konfigurační XML dokument

Ve složce AirFunnel / AirFunnel GUI / bin / Debug / se nachází spustitelný soubor aplikace **AirFunnel GUI.exe** a také konfigurační XML dokument **config.xml**. Při otevření konfiguračního dokumentu se nám naskytne možnost přidat další adresy přijímačů. V bloku `<airSources> </airSources>` můžeme spravovat a přidávat další zdroje připsáním nové adresy a portu. Nová adresa přijímače musí být vložena ve formátu např. `<source port="10002">XXX.XXX.XXX.XXX</source>`. V současném dokumentu je již vloženo několik adres, takže pro připsání další stačí zkopírovat některý ze stávajících řádků a pouze jej upravit. Dále máme možnost změnit vysílací port, který je primárně nastaven na hodnotu **10000**, a také kódování, které je nastaveno na hodnotu UTF-8. Tyto hodnoty však není třeba měnit.

Po spuštění aplikace **AirFunnel GUI.exe** začne aplikace slučovat zadané datové toky do jednoho. Odposlouchávání dat je možné na IP adrese **127.0.0.1**. Jde o logickou smyčku (loopback) datového toku, která vyjde z původního zařízení a je směřována zpět bez dalších úprav. Je využívána především k testování komunikačních tras nebo ke komunikaci vnitřních součástí počítače. Většina počítačů má nastavenou tuto smyčku právě na hodnotu 127.0.0.1, ale tato hodnota může být změněna v rozsahu 127.0.0.0 až 127.255.255.255. Nastavování a změna takových parametrů v systému počítače přesahuje potřeby v rámci řešení mé diplomové práce.

4.2.2 Rozdílný čas zpracování signálů

Je jasné, že pro správné vyhledávání a asociování určitého počtu zpráv je potřeba mít zprávy řazeny přesně podle časové značky. Pokud by zprávy nebyly seřazeny přesně podle časové značky, bylo by možné, že program vyhledá zprávy,

kteře sice odeslalo jedno letadlo, ale v jiný časový okamžik. Takto chybně určené zprávy způsobí velké nepřesnosti nebo úplně znehodnotí výsledek při zjišťování polohy letounu multilaterační metodou.

Každý přijímač pracuje s jiným datovým zatížením a také síť nezaručí to, aby se zprávy dostaly ke koncovému uživateli přesně podle toho, kdy byly přijímači přijaty. V ideálních podmínkách by to bylo teoreticky dosažitelné, ale jelikož každý přijímač pracuje v různých lokalitách a je zapojen do různých sítí, nelze se takovému problému vyhnout. Zprávy z jednoho přijímače přicházejí ve správném pořadí, ale při zapojení dalšího přijímače to v praxi vypadá tak, že zprávy přichází v určitých blocích. Velikost takových bloků záleží na množství letadel v krytí přijímače. Jejich správná pozice dle časové značky se může nacházet více než sto míst od pozice, než ve které byla přijata. Pokud by se s takovými daty nadále pracovalo, bylo by téměř nemožné určit, které zprávy patří shodnému signálu.

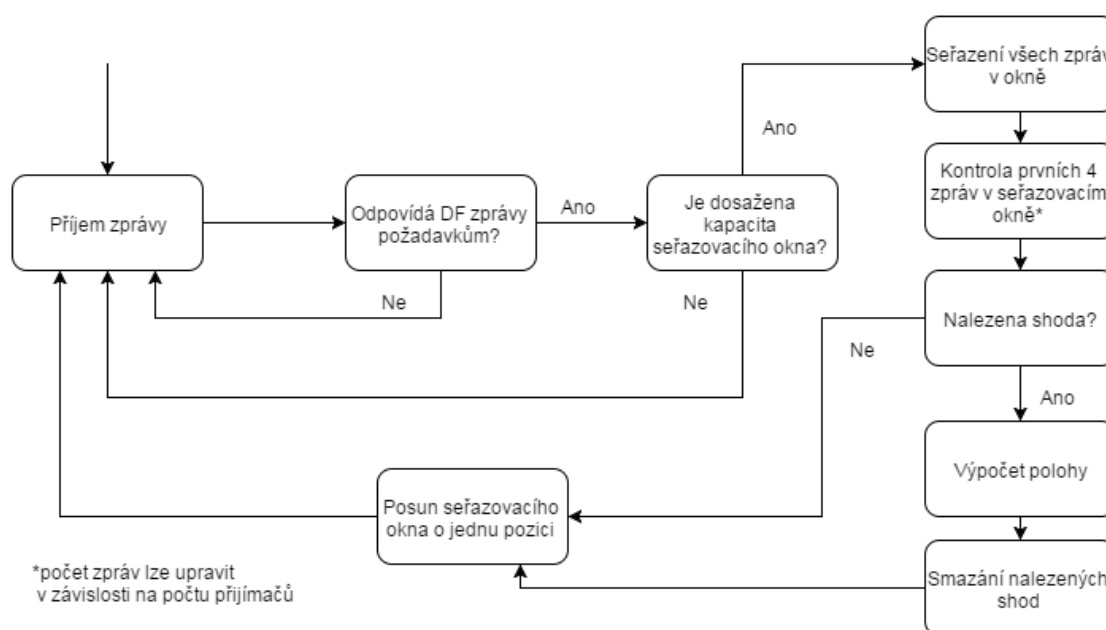
'IP adresa'		'Cas [s]'	'Cas [ns]'			
'147.		'16'	49364	248651937	147.	6>@301E6243BCE102E1971870B16E;
'147.		'16'	49364	248834125	147.	6>@301E624753945D4249B30E91F0;
'147.		'16'	49364	252020875	147.	6>@301E624AC7AD5D4249250953C5;
'147.		'16'	49364	255172156	147.	6>@301E625E88AE02E107130F282B;
'147.		'16'	49364	270397296	147.	6>@301E6279449DA030982C61C26C8D228AC90D754F;
'147.		'16'	49364	270813593	147.	6>@301E62A8D9475D4249250953C5;
'147.		'16'	49364	270813593	147.	6>@301E62C4E0B1A000163FA47A47357FFC0092488C;
'147.		'4'	49364	108968765	147	4>@301E5CE318D68D4B19069905A89B000783E70EB7;
'147.		'4'	49364	158013515	147	4>@301E5CEE7F3B5D40631F41856F;
'147.		'4'	49364	158431656	147	4>@301E5D5015C65D74072E6AC307;
'147.		'4'	49364	188038187	147	4>@301E5DAFC92A5D74072E6AC307;
'147.		'4'	49364	196860812	147	4>@301E5E12FC4E8D344548583594BD8CF6EA3A11C0;
'147.		'4'	49364	196860812	147	4>@301E5E191EAE8D4B190668C38168DAA96AAE06EC;
'147.		'4'	49364	270818609	147	4>@301E5E7B16935D74072E6AC307;
'147.		'4'	49364	294347250	147	4>@301E5EDED725D74072E6AC307;
'147.		'4'	49364	294765140	147	4>@301E5F42F4975D74072E6AC307;
'147.		'4'	49364	335450281	147	4>@301E5F7C4EE78D88801A60A9F49AE8A2D622A85F;
'147.		'4'	49364	335450281	147	4>@301E5F82FD188D88801A99118791080401625CAE;
'147.		'6'	49364	288616859	147	16>@301E62E25D66A000163FA03A3534200FFA4AB9C7;
'147.		'6'	49364	289576187	147	16>@301E63000275A000149810010090F528006840E7;
'147.		'6'	49364	291368734	147	16>@301E6316CD88A00017B0A42A1933A02400F4D957;
'147.		'6'	49364	296930484	147	16>@301E632A226320001A3045A92B;
'147.		'6'	49364	296930484	147	16>@301E634800528D4BA96F99159C14E0048648CA96;
'147.		'6'	49364	296930484	147	16>@301E6348E69BA70024324654B2FE8FBC663D766E;

Obrázek 6: Příjem zpráv v blocích (vlevo), časové rozmezí příjmu v daném bloku zpráv (vpravo). IP adresy byly záměrně smazány.

4.3 Vyhledání shodných zpráv

4.3.1 Princip vyhledávání

V prvním kroku modul spustí příjem zpráv. Po dosažení limitního počtu se musí tento blok zpráv nejprve seřadit dle časové značky, protože není možné docílit toho, aby zprávy přicházely již seřazeny z přijímačů. V takto seřazeném bloku zpráv o určité velikosti se spustí vyhledávání na prvních třech až pěti pozicích (dle nastavení programu). Pokud je nalezena shoda, program si z těchto zpráv načte časové značky a spočítá polohu. Dále následuje příjem další zprávy, která je seřazena do již existujícího bloku. Zpráva na první pozici se smaže. Systém znovu zkontroluje zprávy na prvních pozicích, a tak pokračuje až do docílení požadovaného počtu zpráv. Máme načteno několik zpráv v poli (přibližně sto) a algoritmus srovná pouze prvních n-zpráv (tři až pět v závislosti na konfiguraci) a pak se postupně v tomto poli posunuje. Je nutné tyto zprávy načítat dopředu právě kvůli příchodům opožděných zpráv.



Obrázek 7: Schéma vyhledání zpráv

4.3.2 Odražené signály

Občas nastane situace, že přijímač přijme hned po sobě dvě zprávy, které pochází z jednoho vyslaného impulzu. Takto odražené zprávy přijdou ve velmi krátkém intervalu. Protože je odražený signál vždy opožděn oproti přímému, dá se tato situace ve vyhledávání zpráv zohlednit. Pokud odražený signál přijde až poté, co už

jsou přijaty zprávy na všech třech stanovištích, pak nemá vliv na vyhledání daných shodných zpráv. Pokud přijde mezi nimi, algoritmus tuto kombinaci zpráv vyřadí. Pokud by se však stalo, že se zpráva odrazí i k dalšímu přijímači, soubor takových zpráv neprojde podmínkami vyhledávacího cyklu.

4.4. Dekódování zpráv

Pro naše potřeby bude nutné části přijatých zpráv dekodovat. Není nutné dekodovat úplně celé zprávy, protože pro multilateraci nám postačí pouze přesný čas příchodu signálu na přijímač, dále downlink formát zpráv, které nám usnadní vyhledávání mezi nimi, a nakonec 24bitová adresa, již má každé letadlo unikátní, a je možné jej díky ní vždy rozeznat. Zprávy přijaté po průchodu programem Air funnel vypadají následovně. Pro příklad jsou uvedeny zprávy DF11 a D17.

Krátký formát (Acquisition squitter):

XXX.XXX.XXX.XXX>@2549C085B1965D49D006E80D35;

Dlouhý formát (Extended squitter):

XXX.XXX.XXX.XXX>@456941FA5F598D3C646C58C3952B9B083E0611C7

4.4.1 Časová značka

Informace o čase, kdy byly konkrétní zprávy přijaty, by měly být co nejpřesnější. My máme k dispozici časovou značku, která je rozdělena na dvě části. První část nese informaci o sekundách od 00:00 UTC času a druhá část nese informaci o nanosekundách. Tato informace je zakódována do hexadecimální soustavy. Nejprve je nutné převést dvanáct číslic do binární soustavy. Dostaneme čtyřicet osm bitů, ve kterých prvních osmnáct bitů nese informaci o sekundách a dalších třicet bitů informaci o nanosekundách. Pro získání těchto hodnot pouze převedeme rozdělená binární čísla do desítkové soustavy:

[456941FA5F598D3C646C58C3952B9B083E0611C7](#)

Modrá část zprávy v sobě nese časovou značku. Po převedení do binární soustavy dostaneme číslo až o čtyřiceti osmi bitech. Jelikož potřebujeme právě čtyřicet osm

bitů, musí být nuly, které předchází první jedničky, také započítány. Jinak bychom nemohli čas správně převést:

'010001010110100101000001111110100101111101011001'

Po převedení prvních osmnácti bitů do desítkové soustavy dostaneme čas v sekundách od 00:00:00 UTC. V tomto případě je to 71 077 sekund. Po přepočtení dostaneme čas 19:44:17. Další část nese informaci o počtu nanosekund. Tato část je nejpodstatnější. Sekundy nás zajímají jen v případě, že zprávy přijmeme na rozhraní dvou sekund. Výše uvedená zpráva po převodu říká, že zpráva přišla v 19 hodin, 44 minut, 17 sekund a 33 185 628 nanosekund. Pro krátké zprávy je postup dekódování časové značky úplně stejný jako pro dlouhé zprávy.

4.4.2 ICAO adresa

Abychom mohli přiřadit zjištěné poloze letounu jeho identifikační číslo, je nutné ho také ze zpráv separovat. Čtyřicetibitovou adresu obsahují zprávy DF11 a DF17:

456941FA5F598D3C646C58C3952B9B083E0611C7

2549C0821C1B5D406AC33BC9C5

4.4.3 Downlink formát

Pro zjištění, o jaký typ zprávy se jedná, je nutné dekódovat downlink formát – formát sestupného spoje. Zprávy obsahují různé údaje, například polohu, ICAO adresu, rychlost, výšku a podobně.

Ve zprávách je downlink formát kódován hned jako první znak, ale přijímač přidá před zprávu časovou značku. Proto při aktivované možnosti přidávání časových značek před jednotlivé zprávy nalezneme DF na pozici 13 až 14. Jedná se ale pouze o prvních pět bitů. Následující tři bity nesou informaci o úrovni odpovídače tzv. Capability. Odpovídač první úrovně dokáže zajistit pouze sledování a nerozlišuje, zda se letadlo nachází na zemi nebo ve vzduchu. Odpovídač vyšších úrovní již tuto informaci dokáže poskytnout. V našem systému jsem se nezabýval rozlišením, zda je letadlo na zemi, nebo ve vzduchu, protože se pozemní zprávy z Letiště Václava Havla nedostanou ke všem přijímačům, tudíž se k samotné kalkulaci polohy také nedostanou.

V praxi by však neměl být problém ve výpočtu, zda se letoun nachází ve vzduchu, nebo na zemi.

456941FA5F598D3C646C58C3952B9B083E0611C7

2549C0821C1B5D406AC33BC9C5

Pomocí DF jsou odlišeny zprávy Mode S. Používají se následující typy:

- **DF 00** – Krátké sledování, vzduch-vzduch (ACAS)
- **DF 04** – Odpověď s údaji o výšce letounu
- **DF 05** – Odpověď s identifikačními údaji letounu
- **DF 11** – Acquisition squitter, odpověď na všeobecný dotaz
- **DF 16** – Dlouhé odpovědi, letadlo-letadlo (ACAS)
- **DF 17** – Extended squitter, rozšířená odpověď na všeobecný dotaz
- **DF 18** – TIS-B (např. vozidla na letištní ploše)
- **DF 19** – Vojenský extended squitter
- **DF 20** – Comm-B, odpověď výšky
- **DF 21** – Comm-B, odpověď identifikace
- **DF 22** – Pro vojenské účely
- **DF 24** – Comm-D (až 1280 bitů)

4.4.4 Obsah jednotlivých zpráv

Při provádění MLAT je nutné, abychom znali obsahy jednotlivých zpráv, jejich význam a počet bitů. Níže je tabulka všech informací, které jsou ve zprávách různých formátů obsaženy.

Downlink formát	DF (binárně)	Obsah zprávy
0	0 0000	VS, CC, SL, RI, AC, AP
,4	0 0100	FS, DR, UM, AC, AP
5	0 0101	FS, DR, UM, ID, AP
11	0 1011	CA, AA, PI
16	1 0000	VS, SL, RI, AC, MV, AP
17	1 0001	CA, AA, ME, PI
20	1 0100	FS, DR, UM, AC, MB, AP
21	1 0101	FS, DR, UM, ID, MB, AP
24	1 1xxx	KE, ND, MD, AP

- AA** – Aircraft Address, ICAO adresa letounu, použití v DF11 a DF17. (24 bitů)
- AC** – Altitude code, výška (13 bitů)
- AP** – Adresně paritní pole (24 bitů)
- CA** – Capability – úroveň odpovídače (3 bity)
- CC** – Crosslink capability (1 bit)
- DF** – Downlink formát - první pole 5-ti bitů u všech zpráv (5 bitů)
- DR** – Downlink requests (5 bitů)
- FS** – Flight status A/C (3 bity)
- ID** – Identifikace nastavována pilotem – 4096 možností (13 bitů)
- KE** – Defínuje obsah polí ND a MD Comm-D odpovědí (1 bit)
- MB** – Zpráva odeslána dotazovači, část Comm-B odpovědi (56 bitů)
- MD** – Comm-D pole (80 bitů)
- ME** – Extended squitter message – čistá zpráva DF 17 (56 bitů)
- MV** – Část dlouhé odpovědi při sledování „vzduch-vzduch“ (56 bitů)
- ND** – Poskytuje číslo segmentu pro Comm-D (4 bity)
- PI** – Parita (24 bitů)
- RI** – Informace pro TCAS – obsahuje informace o úrovni TCAS a o přibližné hodnotě rychlosti (4 bity)
- SL** – Sensitivity level TCAS (3 bity)
- UM** - Transmitter- responder status (6 bitů)
- VS** – Indikuje stav letounu na zemi (1), nebo ve vzduchu (0) (1bit)

5 Návrh funkčního modelu

5.1. Příjemací stanice

Ústav letecké dopravy Fakulty dopravní nám zajistil díky Studenstké grantové soutěži (SGS) tři ADS-B receivers. S takovým počtem přijímačů je možné monitorovat letecký provoz pouze ve 2D. Poslední přijímač byl pořízen v květnu 2015, a proto bylo možné v praxi začít testovat možnosti takového systému.

5.1.1 Popis použitého zařízení

5.1.1.1 Přijímač

Použili jsme ADS-B přijímače Radarcape s univerzálním vestavěným Linux systémem, který má schopnost pracovat plně sám bez nutnosti PC. Není nutný ani dodatečný software. Se síťovým a USB rozhraním je to nejvhodnější varianta ke sběru dat pro naše účely. Vestavěný GPS přijímač zaručuje dobrý základ právě pro multilateraci.

5.1.1.2 Anténa

Náš systém využívá pasivní anténu se střední velikostí sběru ve frekvenčním rozsahu 1090MHz. Tato anténa je vhodná pro venkovní instalaci při použití středně dlouhých propojovacích kabelů nebo kabelů s nízkou ztrátou.



Obrázek 8: Anténa a přijímač Radarcape [17]

5.1.2 Pořizovací náklady

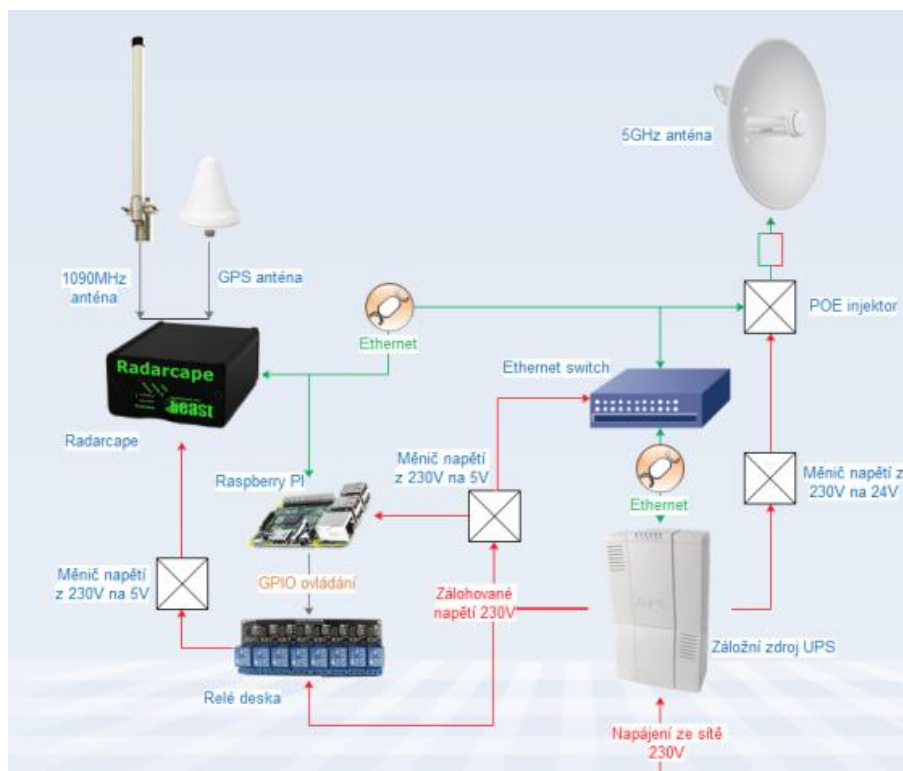
Profesionální MLAT systémy se dají pořídit v řádech stovek tisíc až milionů korun v závislosti na pokrytém území a využití. Vzhledem k tomu, že cílem mé práce je vytvořit low-cost systém, nemůžeme si dovolit používat zařízení za stovky tisíc. Je tedy nutné snížit náklady na minimum.

Náš systém využívá ke svému chodu možnosti, které poskytují střechy budov a vnitřních prostor. To znamená, že zapouzdření proti povětrnostním podmínkám u většiny přijímačů není třeba. V případě, že by bylo nutné vytvořit samostatně stojící přijímač, náklady se budou šplhat výš. Kalkulace takové jednotky se pohybuje okolo třiceti tisíc korun.

Produkt	Cena [Kč]
Radarcapex + GPS a 1090MHz anténa + adaptér	19.500
RaspberryPi2 + adaptér	1.600
Relé deska	350
Switch D-Link DGS-105/E	600
UBNT Powerbeam M5 včetně injektoru	2.400
APC Back-UPS HS 500VA	3.500
Stožár + spojovací materiál	1.000
Kabely a další příslušenství	2.000
Zapouzdření	2.000
Součet:	32950

Obrázek 9: Cenová kalkulace nové stanice [Zdroj: Zach M, *Návrh nízkonákladového MLAT, Praha, 2015*]

Celkové náklady na pořízení a konstrukci MLAT systému řešil ve své diplomové práci Ing. Martin Zach, který se mnou na tomto projektu spolupracoval.



Obrázek 10: Schéma celé pozemní stanice [Zdroj: Zach M, Návrh nízkonákladového MLAT, Praha, 2015]

5.1.3 Webový server

Při práci s přijímači je nutné znát jejich stav a mít možnost je nějakým způsobem konfigurovat. Pro tyto účely je nutné znát také adresy přijímačů. Ukázka adresy pro správu přijímačů vypadá následovně: <http://xxx.xxx.xxx.xxx/gps.html>.



Obrázek 11: Rozhraní webového serveru

V jednotlivých záložkách se nachází následující informace:

- Aircraft data

Jedná se o data z přijímače, která jsou zobrazena do tabulky. Obsahují základní informace o daném letounu - čas, ICAO adresu, číslo letu, souřadnice, výšku, rychlost, kategorii letounu, Squawk, vzdálenost a podobně.

- **Feeder**

Nastavování sdílení přijatých zpráv pro jiné instituce. Například pro FlightRadar24, Planeplotter, Planefinder a tak dále. Fakulta dopravní zatím tuto možnost sdílení dat mimo fakultu nevyužívá.

- **Status**

V této nabídce máme k dispozici přehled kompletního stavu přijímače. Je zde poloha, výška, UTC čas, teplota vnitřního modulu přijímače, detekované GPS satelity, chybové hlášení a také možnost restartovat GPS pozici. Vidíme také status o připojeních na daný port.

- **Configuration**

Pro hlavní nastavení přijímače slouží tato záložka. Je zde nastavení data feedingu pro již zmíněné instituce, nastavení výstupu do USB, změny jednotek, obnovovací frekvence a frekvence smazání letounů po určitém čase. Dále můžeme měnit mnoho věcí právě na výsledném výstupu. Tyto možnosti jsou popsány níže. Pro jakoukoli změnu na přijímači je potřeba znát heslo, které znají pouze oprávnění uživatelé.

5.1.4 Výstupní formát dat

Pomocí webového rozhraní lze na jednotlivých přijímačích nastavovat následující položky:

- Binární nebo hexadecimální formát datových zpráv. V našem případě je nutné, aby byl formát hexadecimální. Pro binární formáty zpráv by bylo nutné přeprogramovat celý skript právě pro tento formát.
- Použití časové značky. Máme na výběr, zda chceme, aby přijímač označil každou zprávu časovou značkou. Pro náš systém multilaterace je to nezbytné.
- Typ časové značky před jednotlivými zprávami (GPS nebo systémový čas). Náš systém využívá GPS synchronizaci, proto je nutné mít hodiny nastaveny dle GPS.
- Downlink formát přijímaných zpráv (všechny DF nebo pouze DF11, DF17 a DF18). Multilateraci můžeme provádět prakticky na všech formátech.

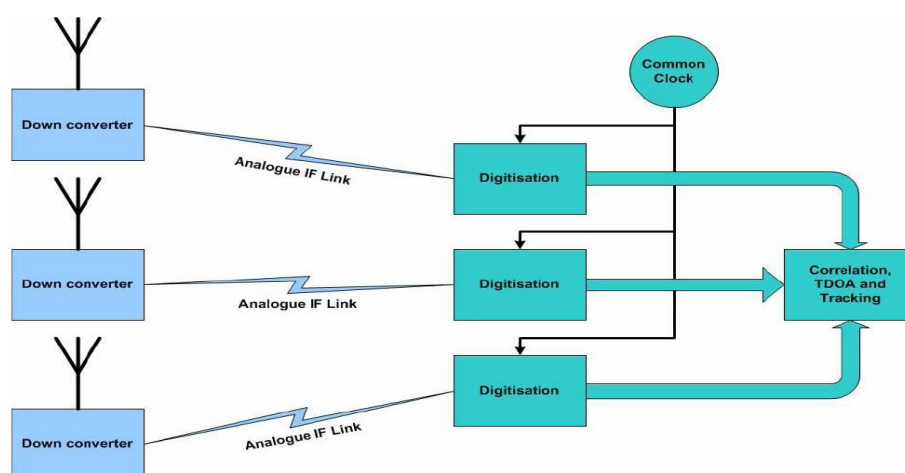
- CRC kontrola zpráv
- Příjem zpráv Módu A/C. Multilateraci lze provádět i na zprávách tohoto formátu. Tato práce se však takovou možností nezaobírá.

5.1.5 Synchronizace hodin

Správná synchronizace hodin přijímačů je nutná pro kalkulaci výsledné polohy a je jí možné docílit několika způsoby:

a) Systém společných hodin

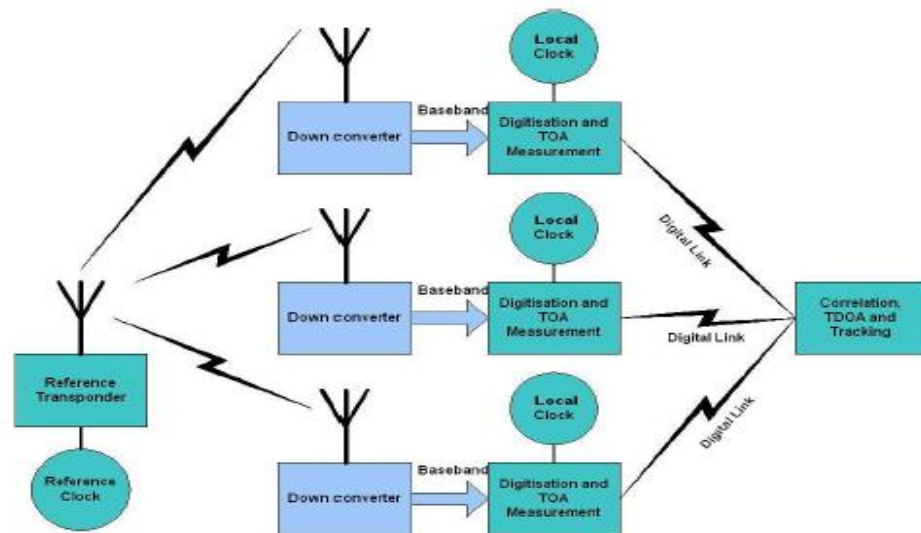
Systém, který využívá jednoduché přijímače pouze pro příjem signálu. Signál je dále veden analogově do místa zpracování, kde proběhne digitalizace a kde jsou umístěny centrální hodiny pro celý systém. Tento systém musí mít přesně zmapován analogový příjem signálu pro určení zpoždění z jednotlivých přijímačů.



Obrázek 12: Schéma synchronizace společných hodin [16]

b) Synchronizace pomocí lokálního vysílače

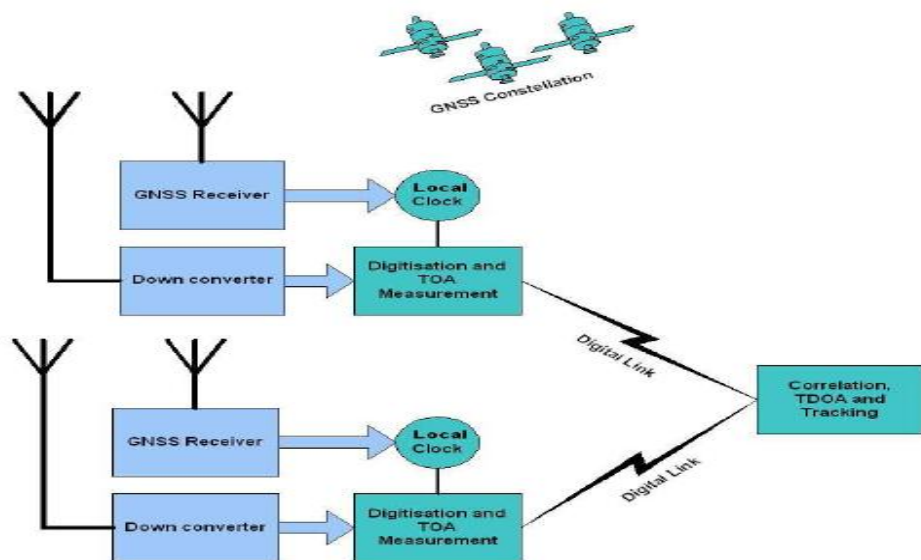
Systém, který pro synchronizaci hodin využívá vysílač, který synchronizuje hodiny na všech stanovištích příjmu. Tento systém se používá pouze pro krátké vzdálenosti mezi vysílačem a přijímačem. Na synchronizaci na větší vzdálenosti má dopad atmosféra. Způsobuje zpoždění signálu, které snižuje přesnost takového systému.



Obrázek 13: Schéma synchronizace pomocí vysílače [16]

c) GNSS synchronizace hodin

Pro náš systém je zvolena synchronizace pomocí GPS. Tato varianta zaručuje přesnost hodin v systému do 100ns UTC. Je to nejvhodnější varianta zejména pro nízkorozpočtové MLAT systémy.



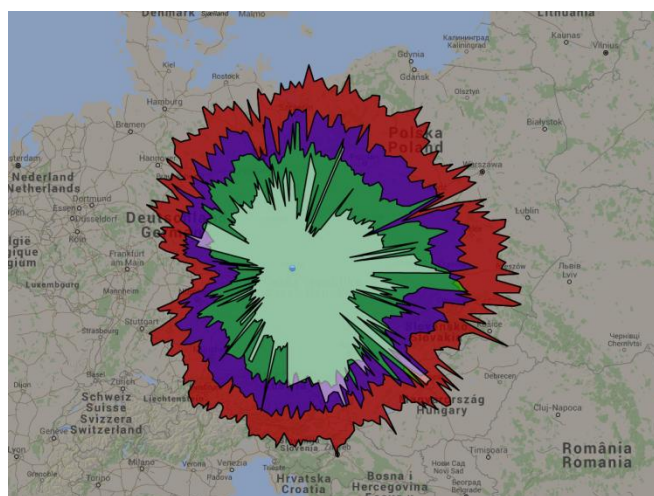
Obrázek 14: Schéma GNSS synchronizace [16]

5.2. Rozmístění jednotlivých přijímačů

Vzhledem k možnostem a účelu této práce byly přijímače rozmístěny pouze po dostupných lokalitách v Praze. Tato možnost je dostačující pouze pro sledování provozu v nejbližším okolí. Pro vzdálenější provoz nelze tento systém použít kvůli vysokým odchylkám vypočítané polohy od reálné polohy zjištěné z ADS-B zpráv, které nesou přesnou informaci o poloze. Je to způsobeno tím, že teoretické hyperboly (viz kapitola 3) se protínají pod velmi malým úhlem, a kvůli chybě hodin v našich přijímačích je výsledná poloha znehodnocena.

5.2.1 Koleje Strahov

První přijímač byl instalován na střechu strahovských kolejí. Je umístěn na bloku č.11. Je to strategická pozice pro příjem díky kopci, na kterém koleje stojí. Umožní kvalitní příjem zpráv ze širokého okolí. Pro náš systém je takový dosah poměrně zbytečný, protože je zatěžován velkým množstvím zpráv, které se nedají filtrovat. Pro jiné analýzy ADS-B zpráv je však velký dosah naopak podmínkou. Na obrázku níže je znázorněn dosah v jednotlivých výškových hladinách po 10 000 stopách. Světle zelená barva je výška od 0 stop do 10 000, zelená pak 10 000 až 20 000 stop. Fialová 20 000 až 30 000 stop a červená od 30 000 stop výše. Vidíme, že dosah v nejvyšších letových hladinách stačí na pokrytí téměř celé střední Evropy. Přijímač je umístěn v boxu, který navrhoval pan Martin Zach v rámci své diplomové práce pro potřeby projektu, tak aby box vydržel ztížené povětrnostní podmínky. Délka anténního kabelu dosahuje délky 7m a GPS kabelu 5m.



Obrázek 15: Dosah přijímače Strahov [14]



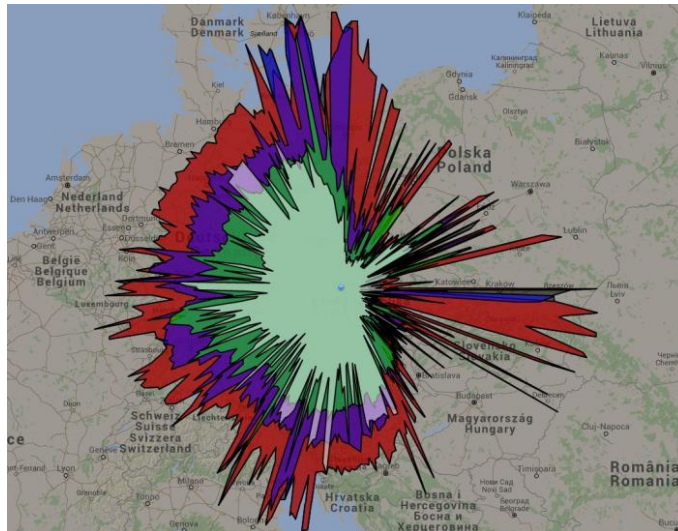
Obrázek 16: Zapouzdření přijímače na střeše strahovských kolejí [zdroj Ing. Martin Zach]

5.2.2 Pankrác

Tato lokalita je pro náš systém velmi vhodná díky instalaci na jednu z výškových budov v této oblasti. Pokud bychom však chtěli systém rozšířit a zvětšit oblast příjmu, bylo by vhodné buď přijímač na Strahově, nebo právě na Pankráci přemístit do vzdálenější lokality. Anténa přijímače na Pankráci je z jedné strany stíněna konstrukcí samotné budovy, proto můžeme vidět značný rozdíl v dosahu směrem na východ oproti přijímači na Strahově.

Samotný přijímač je umístěn v serverovně nacházející se několik metrů od antény. Dále je přijímač napojen na zásuvku připojenou do internetové sítě, která umožní vzdálený restart celého zařízení, pokud by nastal nějaký problém. Přijímač lze restartovat i samostatně, ale pokud nastane situace, že přijímač přestane komunikovat, je potřeba jej odpojit od elektrické sítě.

GPS i anténní kabel mají na této pozici mnohem delší kabely oproti ostatním přijímačům. GPS kabel měří 10m a kabel k anténě měří 18m. Značné rozdíly v délkách kabelů mají vliv na celkovou přesnost systému. Proto je nutné po trvalém usazení MLAT systému vytvořit korekce na jednotlivé přijímací stanice.



Obrázek 17: Dosah přijímače Pankrác [14]

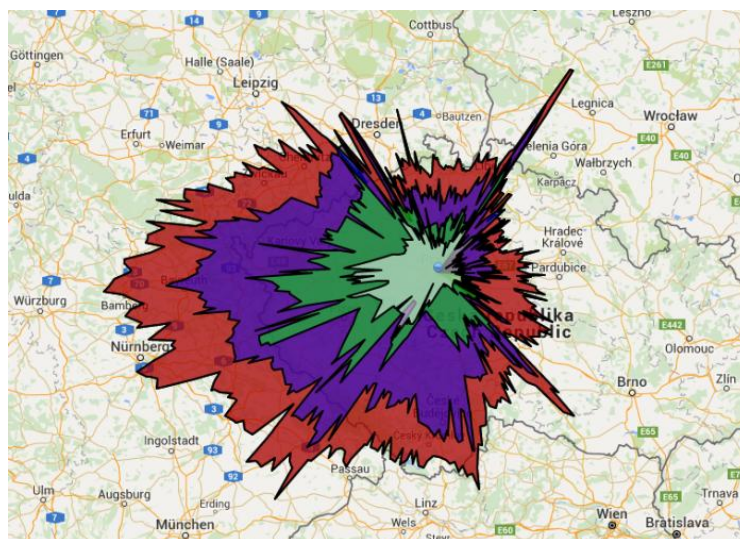


Obrázek 18: Elektrická zásuvka pro vzdálený restart přijímače kolejí [zdroj Ing. Martin Zach]

5.2.3 Fakulta dopravní

Tato lokalita byla zvolena pouze pro rozběhnutí celého systému do té doby, než se najde vhodná oblast pro instalaci. Přijímač je stíněn budovou fakulty, navíc je osazen v zastavěné oblasti a nachází se přibližně o 150 metrů níže než přijímače na Strahově a Pankráci. Proto je dosah značně omezen. Na obrázku vidíme, že dosah zejména v nízkých výškách je oproti zbylým přijímačům nevyhovující. Blízký provoz kolem Letiště Václava Havla nelze monitorovat vůbec, jelikož dosah přijímače v okolí letiště zabírá provoz přibližně nad 2000 – 4000 ft. v závislosti na poloze letounu. Tudíž vzletající i přistávající letadla najednou zmizí nebo se naopak objeví. Ve vyšších letových hladinách pokryje přijímač přibližně polovinu republiky. Pro potřeby multilaterace by dosah ve vyšších výškách byl dostatečný, ale vzhledem k rozmístění

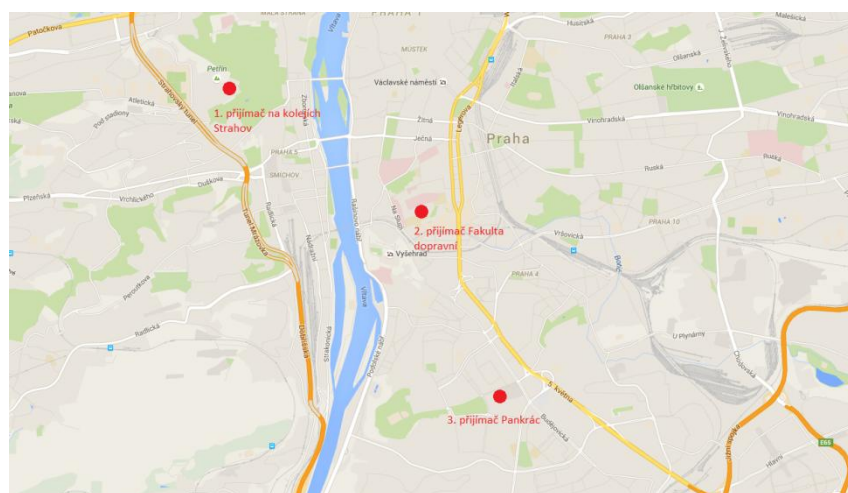
celého systému je nutné, aby byla zajištěna viditelnost zejména v nižších výškách, protože většina provozu v okolí Prahy směřuje z Letiště Václava Havla nebo právě na něj. Kvůli dočasnému umístění kabely dosahují délek 2m (GPS) a 0,5m (anténa).



Obrázek 19: Dosah přijímače na Fakultě dopravní [14]

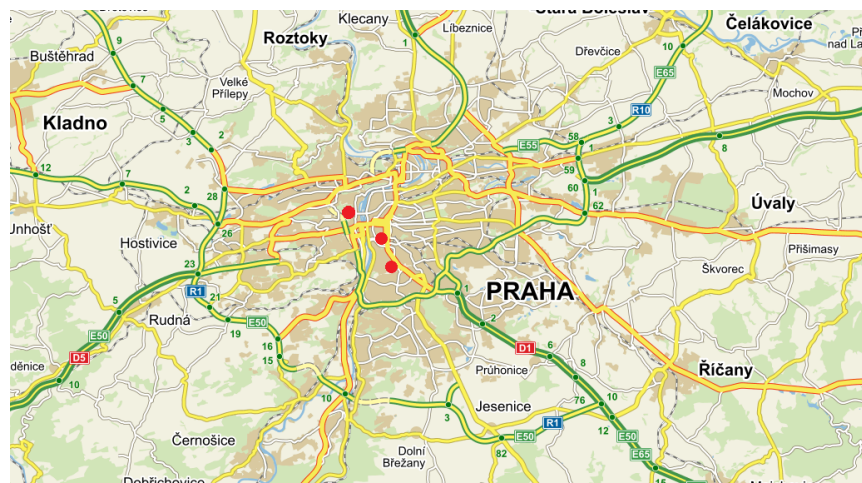
5.2.4 Celkové rozmístění

Z obrázku lze vyčíst, že rozmístění stanic není ideální ani z hlediska konkrétní pozice rozmístění. Přijímače jsou téměř v jedné linii, což nám také zvýší chybu vypočtené polohy. S rostoucí vzdáleností roste výrazně odchylka takového systému.



Obrázek 20: Umístění přijímačů v lokalitách Prahy [15]

Pro představu rozmístění ve větším měřítku nám poslouží následující obrázek:



Obrázek 21: Umístění přijímačů v Praze [15]

Pro lepší výsledky by bylo nutné rozmístit přijímače do přibližně rovnostranného trojúhelníku. Ideální by však bylo umístit všechny přijímače po okrajích Prahy, aby byla zaručena dostupnost pro náhlé opravy a kontroly. Pokrytí pro MLAT by se značně zvýšilo a odchylky značně snížily. Bylo by také možné snadněji odfiltrout nesprávné polohy, které algoritmus vytvoří.

5.3 Modul realizující příjem a výpočet

Níže následuje popis skriptu pro celý proces příjmu, asociace a výpočtu polohy:

```
close all
clear all

format long g
```

Instrument Control Toolbox nám umožní vytvořit připojení přes protokol TCP/IP. Pokud instalovaný Matlab nemá implementován tento toolbox, není možné skript spustit. Definujeme IP adresu a port. Adresa i port jsou nastaveny na smyčku (loopback) počítače.

```
t = tcpip('127.0.0.1', 10000);
```

Uurčíme pořadí přijímačů dle IP adres, abychom mohli jednoznačně určit, z kterého zpráva přišla. Adresy a porty na přijímačích se čas od času mění, proto je nutné pro správnost chodu zajistit, aby byly tyto hodnoty správně nastaveny. Pořadí je nutné pro nastavení hlavního a vedlejších přijímačů. IP adresy jsou uvedeny pouze pro názornost. Nejedná se o skutečné adresy přijímačů.

```
ip_1=147255255251; %147.255.255.251
ip_2=147255255252; %147.255.255.252
ip_3=147255255253; %147.255.255.253
```

Dále můžeme nastavit velikost vyrovnávací paměti. Lze ji upravit na libovolnou hodnotu, ale není nutné ji zvyšovat. Pokud by nestačil počítač data zpracovávat, hromadila by se zbytečně až do doby, kdy nějaký modul selže.. Tuto možnost využijeme v případě, že nemáme vytvořen buffer mimo prostředí Matlab. (Máme vytvořen ještě jeden slučovací program, který ale pro menší přehlednost nebyl použit.)

```
set(t, 'InputBufferSize', 32768);
```

Otevřeme připojení, na kterém jsou již sloučeny datové toky.

```
fopen(t);
```

Zvolíme počáteční hodnoty parametry příjmu a vyhledávání. Nejprve počet zpráv, poté zvolíme, mezi kolika zprávami se bude vyhledávat a pak rozhodneme, jak

velký blok zpráv se bude seřazovat pro vyhledávání. U těchto hodnot rozptylů bylo vyzorováno, že je zde určitý kompromis mezi rychlostí příjmu, zpracováním a vyhledávacím výkonem. Vše však záleží na aktuálním provozu.

```
n=30000;           %počet přijatých zpráv
rozptyl=4;        %počet zpráv mezi kterými se vyhledává
roz=100;         %rozptyl řazení časových značek
```

Definujeme proměnné, se kterými budeme pracovat později a pole, do kterého se budou zprávy ukládat:

```
pocet_shod=0; %pro informaci o počtu shod
k=0; %pro ukládání zpráv (viz níže)
L=0;
K1=cell(n/2,8); %definování pole pro ukládání zpráv
```

Spuštění počítačového času. Tento příkaz je pouze pro statistické účely:

```
tic;
```

Zde začíná cyklus pro příjem zpráv. Matlab přečte jeden řádek (jednu zprávu), převede ji na řetězec znaků, aby s ní byl schopen pracovat. Poté zprávu rozdělí na IP adresu a zprávu včetně časové značky. Tento cyklus se opakuje do té doby, než je dosažen počet přijatých zpráv, který byl nastaven v počátečních parametrech.

```
for i=1:n
% přečtení řádku z přijímače
tline = fgetl(t);
% převod na řetězec
char_msg = char(tline);
%nalezení @ pro rozdělení zprávy (různé IP mají různou délku)
zavinac = find(char_msg == '@');
IP=char_msg(1:zavinac-2); %pouze IP adresa
zprava=char_msg(zavinac+1:end-2); %zpráva od @ dále
```

V tomto případě provádíme multilateraci pro zprávy dlouhého formátu. Proto je přidána podmínka, aby měla zpráva čtyřicet znaků. Pokud bychom prováděli multilateraci pro jiné formáty (např. DF11), museli bychom podmínku upravit.

```
% podmínka délky řetězce pro dlouhé zprávy (40 znaků)
```

```
if length(zprava) == 40
```

Dále propustíme pouze zprávy DF17. Jak bylo zmíněno výše, typ zprávy je uveden hned za časovou značkou. Tato označení u DF17 nabývají hodnot 88, 89, 8A, 8B, 8C, 8D, 8E, 8F. Proto nemusíme dekódovat celou kombinaci do binární soustavy a poté srovnávat prvních pět bitů, ale stačí pouze podmínka pro první znak. Pokud bychom však dekodovali všech osm bitů, mohli bychom zjistit také úroveň odpovídáče. Tato hodnota je však pro nás zbytečná.

Při provádění multilaterace na zprávách DF11 je nutné tuto hodnotu změnit na 5. Také není nutné dekódovat celých osm bitů, protože hodnoty nabývají hodnot 58 až 5F. Pouze u zpráv typu DF20 a DF21 je nutné tyto kombinace ošetřit, protože oba formáty začínají písmenem „A“.

```
if zprava(13) == '8'
```

Poté převedeme do desítkové soustavy type code. Nachází se za ICAO adresou. Je to pouze prvních pět bitů, proto musíme vzít dva znaky. Tento krok je pouze pro ověření polohy z ADS-B zpráv které nesou informaci o poloze. Informací o poloze jsou označeny zprávy s type codem od 9 do 18. Při provádění MLAT na jiných zprávách se tento krok může celý vynechat.

```
hex_type = zprava(21:22); %type code  
dec_type=hex2dec(hex_type); %převod z hex do desítkové soust.  
bin_type=dec2bin(dec_type,8); %převod do 48bit binární soust.
```

```
type_code=bin2dec(bin_type(1:5)); % type code dle bitu
```

```
%separování zpráv které nesou informaci o poloze letadla  
if type_code >= 9 && type_code <= 18
```

```
%přičtení 1 pro zápis nové zprávy do pole  
k=k+1;
```

Dále změníme IP adresy z přijatých zpráv na číslo, abychom s nimi mohli pracovat. např 192.168.1.1 na 19216811 atd. Mohli jsme je změnit okamžitě, ale není potřeba zatěžovat program dalším krokem u zpráv, které neprojdou přes nastavené parametry.

```
%změna IP adresy na číslo ( např 192.168.1.1 na 19216811)  
IP_tecka=find(IP == '.');
```

```

IP_cislo=sprintf('%s%s%s%s' ,IP(1:IP_tecka(1)-1)
,IP(IP_tecka(1)+1:IP_tecka(2)-1)...
, IP(IP_tecka(2)+1:IP_tecka(3)-
1),IP(IP_tecka(3)+1:end) );
IP_cislo=str2double(IP_cislo);

```

V dalším kroku dekódujeme časovou značku příchozí zprávy. Je nutné ji nejprve převést do binární soustavy o 48 bitech a separovat část, která obsahuje sekundy a poté nanosekundy. Pokud není zaručeno, aby každá značka měla 48 bitů (pokud by nuly na začátku zprávy nebyly započítány), bude časová značka nesprávně dekódována.

```

hex_cas = zprava(1:12);%časová značka
dec_cas=hex2dec(hex_cas); %převod z hex. do desítkové soust.
bin_cas=dec2bin(dec_cas,48); %převod do 48bit binární soust.

bin_cas_sec=bin_cas(1:18); %separování části, která nese informaci o
sekundách
bin_cas_nano=bin_cas(19:48);%separování části která nese informaci o
nanosekundách

T_nano = bin2dec(bin_cas_nano); %převod nanosekund do desítkové
soustavy
T_sec = bin2dec(bin_cas_sec); %převod sekund do desítkové soustavy

cas=sprintf('%d.%d',T_sec, T_nano); %spojení sekund a nanosekund
cas=str2double(cas); %převedení předchozího spojení na číslo

```

Do vytvořeného pole začneme ukládat data potřebná pro vyhledávání a pro pozdější výpočet. Toto pole nám umožní pozdější kontrolu údajů po odpojení a následné úpravy.

```

%zápis zpráv do pole
K1{k,1}=zprava; %obsah ADS-B zprávy z přijímače
K1{k,2}=zprava(21:34); %cista zprava DF_17 (21:34)
K1{k,3}=IP_cislo; %IP adresa přijímače

K1{k,4}=T_sec;
K1{k,5}=T_nano;

```

Následuje cyklus pro vyhledání shodných zpráv. Protože zprávy nepřijdou uspořádané, je nutné nasbírat určitý počet zpráv. V našem případě to bude tak velké množství, jak velký bude rozptyl bloku pro seřazení. Tento parametr nastavujeme na začátku celého skriptu.

```

if k >= roz %podmínka pro zahájení vyhledávání jakmile bude načten určitý
počet zpráv (roz)

L=L+1; %slouží pro postupné posouvání v původním poli

```

Vytvoříme si druhé pole, které se nám bude seřazovat podle časových značek. Hodnoty se nakopírují, seřadí a vloží zase zpět do původního pole. Velikost tohoto pole je dána již zmíněnou hodnotou „roz“. Toto pole se posouvá s každou přijatou zprávou, která vyhovuje nastaveným filtrům, o jednu pozici dále v původním poli „K1“.

```
%vytvoření pole které se bude seřazovat
K2=K1(L:L+roz-1,1:end);
%řazení zpráv podle časových značek
K2=sortrows(K2,4:5);
%přepis seřazeného pole zpět
K1(L:L+roz-1,1:end)=K2;
```

Nyní vytvoříme pole, ve kterém se budou vyhledávat zprávy. Pro zprávy typu DF17 by se mohlo použít pole s již seřazenými zprávami, protože obsahují unikátní informaci o poloze, ale pro zprávy typu DF11 by se často stávalo, že se asociují zprávy, které k sobě nepatří. Navíc by varianta bez vytvoření dalšího pole byla pomalejší z důvodu prohledávání většího počtu zpráv. Toto pole si vybere pouze první čtyři hodnoty (v našem případě) a v nich se snaží najít shodné zprávy. Hodnota čtyř zpráv je pro tři příjemce více než dostačující. Případů, kdy přijde na některý příjemce zpráva v čase přesně mezi zbylými zprávami, je minimum. Odražených zpráv se v našem systému také nevyskytuje tolik, aby ovlivnily výslednou separaci.

```
%vytvoření pole pro vyhledávací cyklus
K3=K1(L:L+rozptyl-1,2);
```

Zahájení vyhledávání v definovaném poli.

```
for Z1=1:rozptyl
```

Porovnáme hodnoty v daném poli a nalezneme pozice, na kterých se nachází. Tento příkaz se jevil jako nejrychlejší varianta vyhledání shod. Při použití porovnání dat bez mezikroku, který definuje pole K3, by trvalo daleko déle najít shodné zprávy.

```
index = find(strcmp(K3,K3{Z1,1})); %nalezení stejných zpráv a
určení místa shod
```

Pokud je nalezena shoda tří zpráv a neobsahuje-li výsledek odražené zprávy, pustí podmínka cyklus dál a upraví se proměnné nutné pro orientaci programu v datech. Jinak cyklus pokračuje zpět na příjem zprávy.

```
if length(index) == 3 && K1{index(1)+L-1,3} ~= K1{index(2)+L-1,3}...
&& K1{index(1)+L-1,3} ~= K1{index(3)+L-1,3}...
```

```
&& K1{index(2)+L-1,3} ~= K1{index(3)+L-1,3}
```

```
%určení místa výsledku v celém poli přijatých zpráv  
vysledek=index+L-1;
```

```
%počítadlo shodných zpráv  
pocet_shod=pocet_shod+1;
```

Pro opakované vyhledávání již nalezených zpráv je nutné je vymazat. Nahradíme je náhodným číslem. Pokud bychom je nahradili třeba nulou, vyhledávací cyklus by tyto nuly znovu k sobě přiřadil.

```
%smazání výsledku pro opakované vyhledávání stejného výsledku  
K1{vysledek(1),2}=rand;  
K1{vysledek(2),2}=rand;  
K1{vysledek(3),2}=rand;
```

```
K3{index(1)}=rand;  
K3{index(2)}=rand;  
K3{index(3)}=rand;
```

Následuje převod a načtení všech časů příjmu u všech třech nalezených zpráv pro možnost vypočtení polohy cíle. Poté přiřadíme ICAO značku z nalezených zpráv.

```
%převod času na číslo
```

```
T_sec_1 = K1{vysledek(1),4}; %zápis sekund  
T_nano_1 = K1{vysledek(1),5}; %zápis nanosekund
```

```
cas_1=sprintf('%d.%d',T_sec_1, T_nano_1); %spojení sekund a nanosekund  
cas_1=str2double(cas_1); %převedení předchozího spojení na číslo
```

```
%=====
```

```
% to samé jako v předchozím kroku, jen pro druhou čas. značku
```

```
T_sec_2 = K1{vysledek(2),4};  
T_nano_2 = K1{vysledek(2),5};
```

```
cas_2=sprintf('%d.%d',T_sec_2, T_nano_2);  
cas_2=str2double(cas_2);
```

```
%=====
```

```
% to samé jako v předchozím kroku, jen pro třetí čas. značku
```

```
T_sec_3 = K1{vysledek(3),4};  
T_nano_3 = K1{vysledek(3),5};
```

```
cas_3=sprintf('%d.%d',T_sec_3, T_nano_3);  
cas_3=str2double(cas_3);
```

```
% ICAO značka v HEX soustavě
```

```
ICAO_hex_1= K1{vysledek(1),1}(15:20);
```


Musíme ošetřit to, aby se správně odečetly časy příchodu na daná stanoviště. A to v závislosti na centrální stanici a ne v závislosti na čase příjmu. Při prohození časů příjmu by výsledná pozice mohla být nesprávně spočítána, nebo by nemusela splnit všechny matematické podmínky (záporné číslo pod odmocninou) a výsledek by byl proto nesmyslný.

```

if K1{vysledek(1),3} == ip_1

    if K1{vysledek(2),3} == ip_2
        delta_1=T_nano_2 - T_nano_1;
        delta_2=T_nano_3 - T_nano_1;
    else
        delta_1=T_nano_3 - T_nano_1;
        delta_2=T_nano_2 - T_nano_1;
    end

elseif K1{vysledek(2),3} == ip_1

    if K1{vysledek(1),3} == ip_2
        delta_1=T_nano_1 - T_nano_2;
        delta_2=T_nano_3 - T_nano_2;
    else
        delta_1=T_nano_3 - T_nano_2;
        delta_2=T_nano_1 - T_nano_2;
    end

elseif K1{vysledek(3),3} == ip_1

    if K1{vysledek(1),3} == ip_2
        delta_1=T_nano_1 - T_nano_3;
        delta_2=T_nano_2 - T_nano_3;
    else
        delta_1=T_nano_2 - T_nano_3;
        delta_2=T_nano_1 - T_nano_3;
    end
end
end

```

Dále následuje funkce pro kontrolu vypočítané pozice a kontrolu vzdálenosti letadel od přijímačů. V našem případě od centra Prahy. Souřadnice jsou přepočítány do kartézského systému XXZ a poté převedeny do lokálních souřadnic pro lepší přehlednost.

%přesná poloha z ADS-B

```
[Rlat,Rlon]=fce_DF17_TC9_22_krom19_CPR_Air_decoder(K1{vysledek(1,1)}(13:40)); %13-40
```

%transformace souřadnic přibližnéhoho středu prahy (Jiráskův most)

```
WGS84_praha=[50.0757864 14.4117611 342];
```

```
ECEF_praha=WGS842ECEF(WGS84_praha);
```

```
[ee, nn,uu]=
```

```
xyz2enu(WGS84_praha(1),WGS84_praha(2),WGS84_praha(3),ECEF_praha(1),ECEF_praha(2),ECEF_praha(3));
```

```
ENU_praha=[ee,nn,uu];
```

%transformace polohy letounu

```
WGS84_acft=[Rlat Rlon 1000];
```

```
ECEF_acft=WGS842ECEF(WGS84_acft);
```

```
[ee, nn, uu]=
xyz2enu(WGS84_praha(1),WGS84_praha(2),WGS84_praha(3),ECEF_acft(1),ECEF_acft
(2),ECEF_acft(3));
ENU_acft=[ee, nn, uu];

%vzdálenost letounu od středu Prahy
hranice= abs(((ENU_praha(1)-ENU_acft(1))^2)+(ENU_praha(2)-
ENU_acft(2))^2)^(1/2);
```

Hranici, do které chceme provádět multilateraci, si můžeme nastavit. Ale pouze u zpráv DF17, které nesou informaci o poloze letounu. Pro zprávy jiných formátů by se musela hranice příjmu omezit nějakým způsobem na přijímači, jinak budeme počítat polohu letounů vzdálených i třeba stovky kilometrů daleko s neúnosnou chybou.

```
%podmínka pro vzdálenost letounu od prahy
if hranice < 30000
vyhovuje=vyhovuje+1;
```

Nastavíme si výšku, do které každé letadlo umístíme, a následně s danou výškou vypočteme polohu. Toto nastavení je nutné řešit z důvodu nutnosti navolení třetí souřadnice při 2D MLAT.

```
Ralt=6000; %nastaveni relativní výšky letadel (v metrech)
```

Teď již můžeme přejít k samotnému výpočtu polohy MLAT metodou. Funkce byla popsána výše a bude také zahrnuta v přílohách. Jsou zde další funkce pro přepočet souřadnic do kartézského systému a poté do lokálních souřadnic.

```
mlt_poloha=poloha_mlat_2(delta_1,delta_2,Rlat,Rlon,Ralt);
```

Porovnání velikosti odchylek. Předpokládáme, že ta, která je menší, patří našemu cíli. Při současném rozmístění nelze dobře separovat jedinou polohu z těch, které algoritmus vypočte.

```
if abs(mlt_poloha(1))<= abs(mlt_poloha(2))
    odchylka=mlt_poloha(1);
else
    odchylka=mlt_poloha(2);
end
fprintf(' %s , Odchylka %.f Vzdálenost %.f \n',ICAO_hex_1,
odchylka, hranice);

end
end
end
end
end
end
```

end

Po zvoleném počtu zpráv se cyklus zastaví a zobrazí se výsledky a statistiky příjmu. Poté se odpojí od navázaného spojení.

```
%zastavení počítání
c=toc;

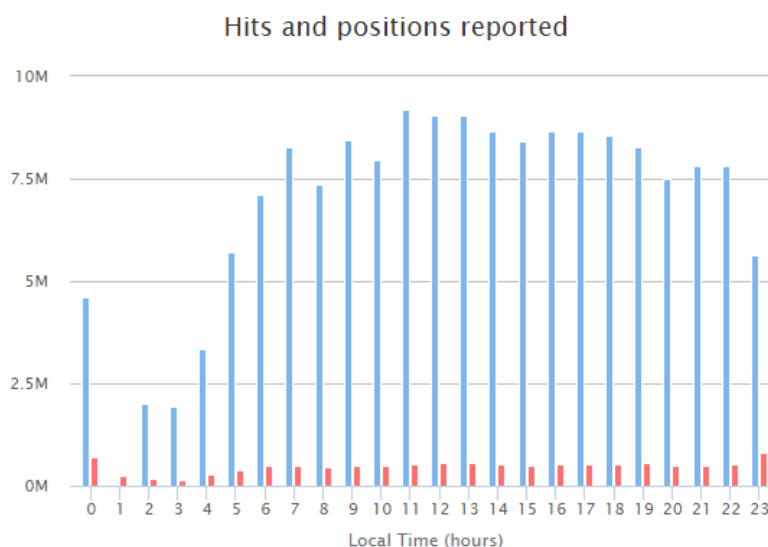
%Výstupní statistiky
fprintf('Příjem trval %f sec., počet zpráv = %d, počet shod = %d, %d
výpočtů, %f msg/sec.\n', c, n,pocet_shod,vyhovuje ,n/c);

% Odpojení od serveru
fclose(t);
```

5.4 Aplikace MLAT metody na aktuální provoz

5.4.1 Příjem zpráv

Statistika přijatých zpráv na našich přijímačích. Modře znázorněné hodnoty ukazují počet všech přijatých zpráv za hodinu a červené hodnoty značí zprávy, které obsahují polohu letounu. V našem systému jsem filtroval pouze tyto zprávy, aby bylo možné zkontrolovat velikost odchylky v reálném provozu. Při nasazení systému do provozu není tato kontrola potřeba.



Obrázek 22: Celkový počet přijatých zpráv a zpráv s polohovou značkou

Pro ověření slouží výše zmíněný skript, který dokáže dekódovat přesnou polohu letounu z jedné zprávy DF17. V MLAT systému však tato možnost odpadá a je potřeba se spolehnout pouze na něj.

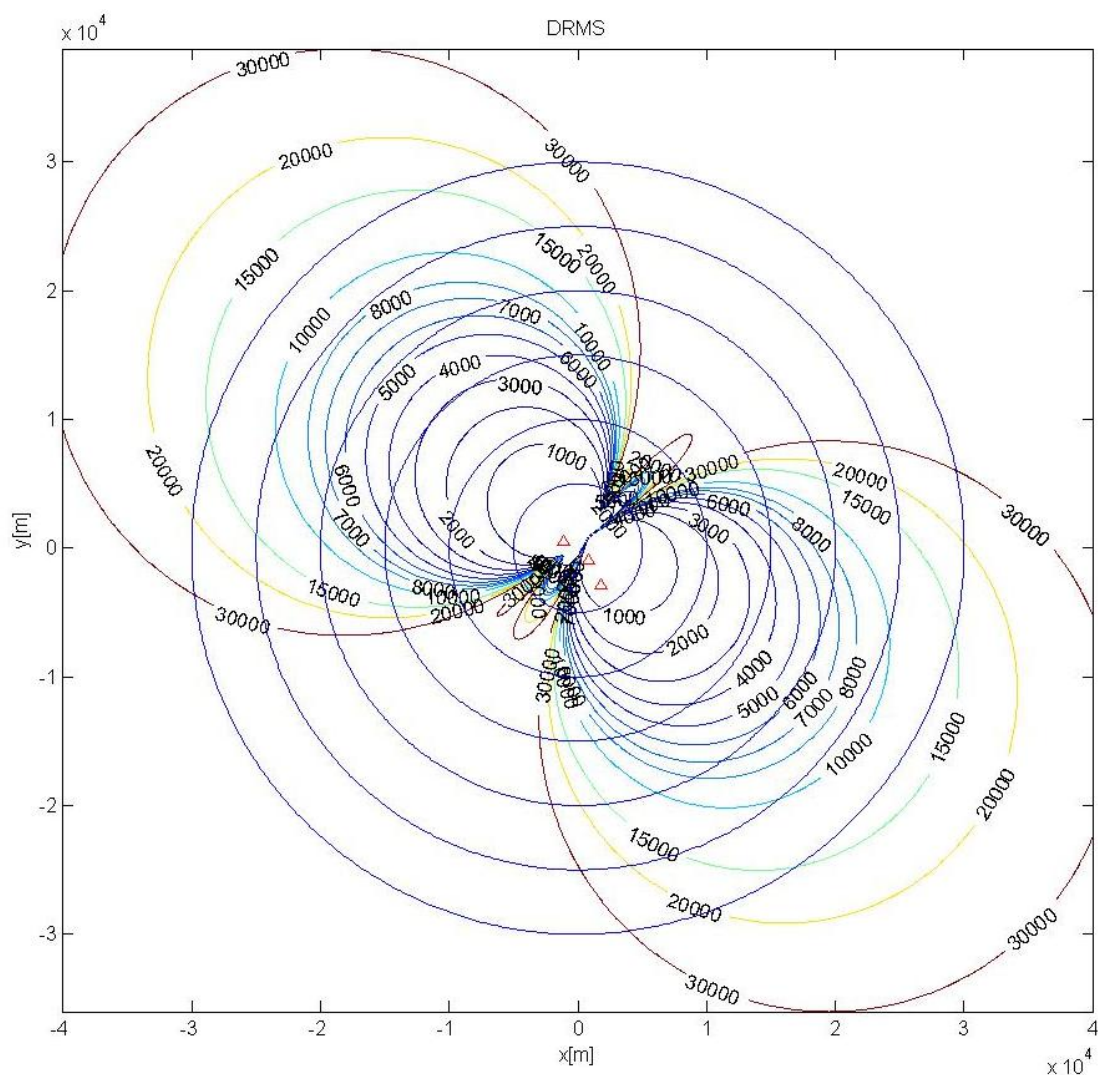
5.4.2 Odchylky měření

Pro stávající systém je největší problém rozmístění přijímacích stanic. Nachází se příliš blízko sebe a navíc v jedné linii. V takovém rozmístění nemůžeme předpokládat kvalitní výstup.

V následující části se budu snažit ukázat teoretické odchylky MLAT systému ve 2D v přesném rozmístění, jako má náš systém.

Pokud budeme uvažovat maximální odchylku GPS hodin maximálně 100ns, pak bude výpočetní chyba našeho systému vypadat následovně. Souběžné kruhy značí

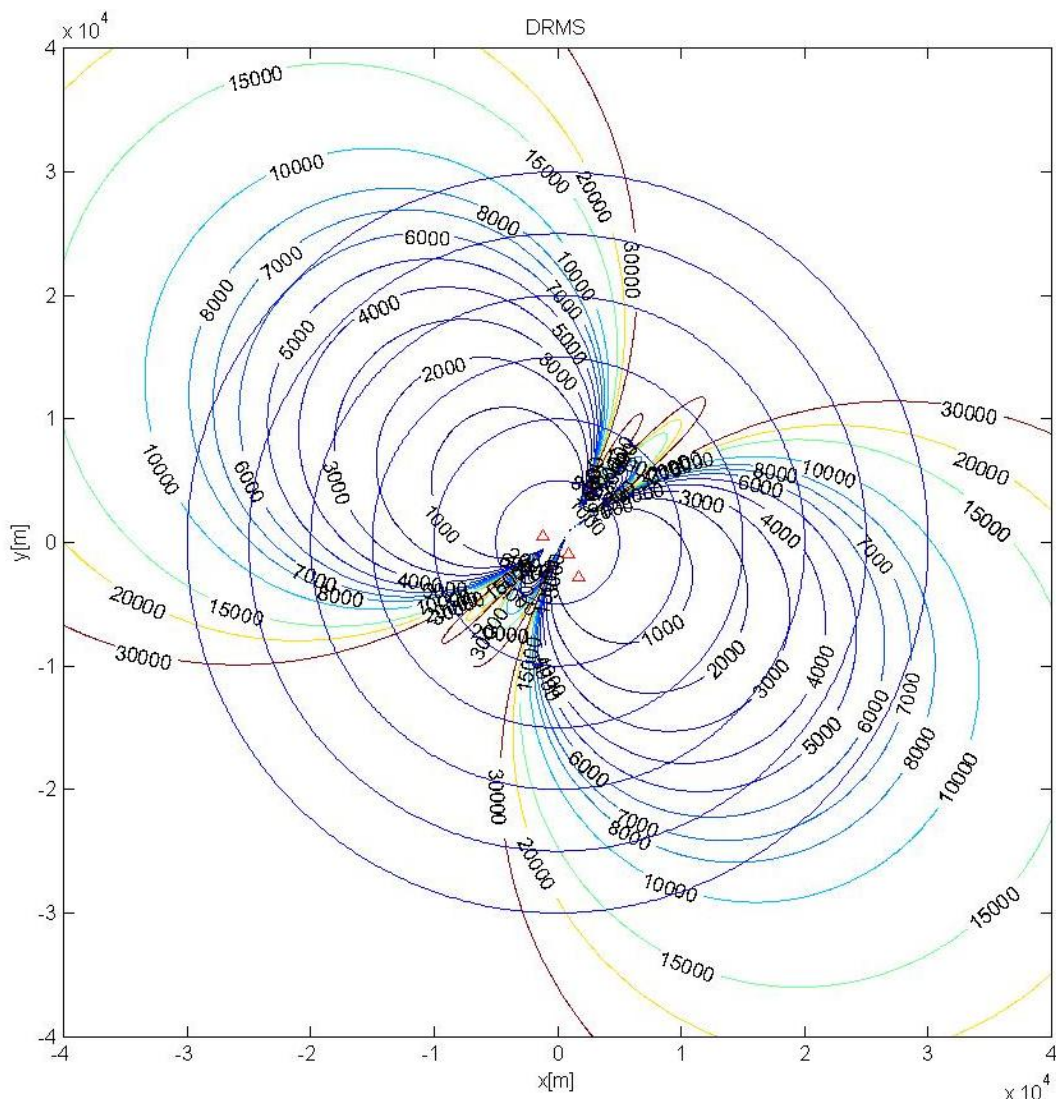
vzdálenost po násobcích pěti kilometrů. Červené trojúhelníky znázorňují reálnou polohu stanic. Vidíme, že nejmenší možná odchylka ve třiceti kilometrech je něco mezi patnácti až dvaceti kilometry. Tak velká odchylka je velmi neuspokojivá. Je několik možností, jak tuto odchylku snížit. První možností je zvýšit přesnost hodin. Jelikož ale není žádná možnost, jak zpřesnit hodiny v našem systému, je nutné rozmístit stanice dál od sebe.



Obrázek 23 :Odchylka systému při odchylce 100ns. Souběžné modré kruhy znázorňují vzdálenost po 5km

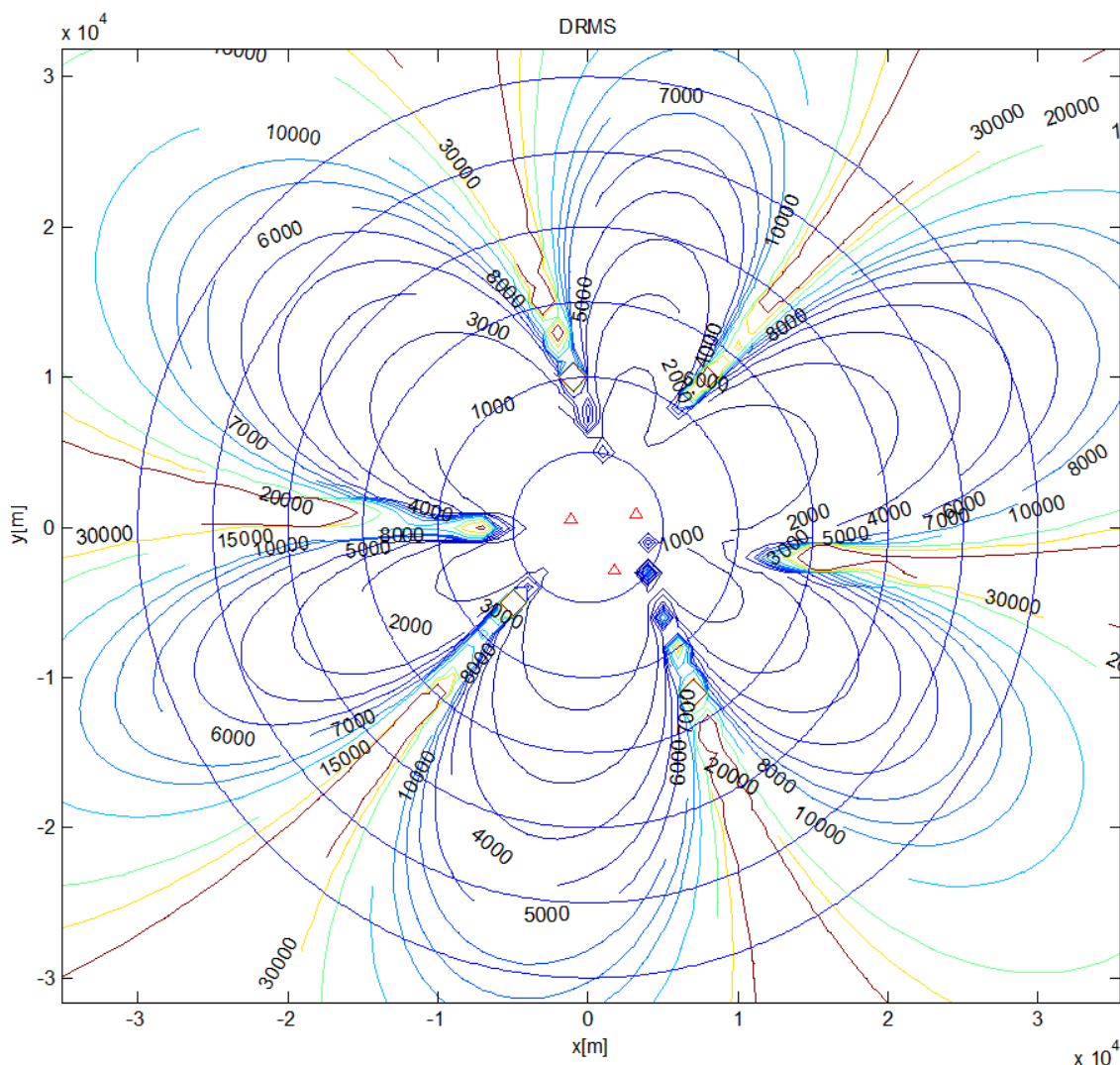
Přesnost hodin nemůžeme na našem systému nijak zjistit ani upravit. Můžeme pouze vycházet z předpokladů pro takový systém, kde jsou hodiny synchronizovány podle GPS. Odchylka by mohla být také menší. Pokud bychom počítali s maximální odchylkou, která by byla o polovinu menší, tedy 50ns, vypadala by výsledná odchylka

v závislosti na poloze následovně. Chyba by se viditelně snížila. Vidíme, že minimální možná odchylka ve vzdálenosti třiceti kilometrů od centra Prahy je přibližně pět kilometrů. Ta však také výrazně narůstá ve směru na severovýchod i jihozápad a znehodnocuje vypočtené pozice i v poměrné blízkosti cíle od přijímačů.



Obrázek 24: Odchylka systému při 50ns odchylce

Pokud bychom zachovali rozmístění stanic na Kolejích Strahov a na Pankráci a přemístili stanici z Fakulty dopravní do takové lokality, abychom dostali alespoň přibližně rovnostranný trojúhelník, vypadaly by odchylky výpočtu při zachování maximální chyby hodin 50ns takto. Lokalita je jen náhodně zvolena bez dalšího zkoumání vhodnosti umístění.



Obrázek 25: Odchylka při přemístění stanice na FD do jiné lokality tak, aby rozmístění vytvářelo trojúhelník

Odchylky vypočtené polohy od polohy skutečné by se značně snížily. Při vzdálenosti do pěti kilometrů od středu Prahy by odchylky neměly přesáhnout jeden kilometr. Pro vzdálenější cíle by však velmi záleželo na aktuální pozici konkrétního letadla. Je však zřejmé, že pro další vývoj a zdokonalování tohoto MLAT systému by bylo nejvhodnější stanici přemístit.

5.4.3 Převod z kartézského systému zpět do WGS84

Jak bylo zmíněno na začátku práce, pro výpočet je nutná transformace ze souřadného systému WGS 84 do systému ECEF a dále do lokálního systému ENU. Tato konverzace není zatížena chybou, která by zásadně ovlivnila výslednou polohu.

Bohužel, při provádění zpětné transformace se chybě nevyhneme, protože je nutné znát třetí souřadnici, kterou si musíme zvolit.

Na rovné ploše bychom mohli navolit hodnotu a ta by zůstávala po transformaci konstantní. Na kulové ploše se však výšková souřadnice nedá zvolit tak, aby po transformaci zůstala konstantní. Bylo nutné použít iterační algoritmus, který zajistí to, že se všechny výsledné polohy nachází v jedné výšce. Tento algoritmus jsme použili jenom pro transformaci, nikoliv pro samotný MLAT výpočet. Byla zvolena přibližná hodnota v souřadném systému ECEF a poté následoval převod.

Ošetření nastavené výšky bylo realizováno pomocí šesti opakování, které stačí i pro větší odchylky. Nastavená výška se liší od transformované o desítky centimetrů až několik metrů, což je přijatelné. Pokud bychom takový krok neprovedli a nechali nastavenou pouze jednu hodnotu, převedená hodnota nebude zaručovat držení se předvolené výškové hladiny.

Výpočet spočíval v tom, že jsme po prvním cyklu transformace zjistili, v jaké výšce se cíl nachází. V závislosti na této informaci jsme poté upravili původní hodnotu o rozdíl mezi požadovanou výškou a spočítanou výškou. Souřadnice se po následném opakování transformace posunuly a výška se ustálila u nastavené hodnoty. Pokud bychom tento krok opakovali vícekrát, dostali bychom se přesně na požadovanou hodnotu, ale na úkor rychlosti, která je nutná pro celý proces zpracování.

Tato transformace však nebyla použita pro kontrolu výsledků. Hodnoty odchylek byly zjišťovány v lokálním souřadném systému se středem mezi přijímači v centru Prahy.

5.4.4 Další nepříznivé vlivy

Odchylka je také způsobena různou délkou kabelů, jež způsobuje sice pravděpodobně malé odchylky, ale i proto vzniká chyba ve výpočtech. Vše bude možné řešit korekcemi na každý přijímač, jakmile bude pozice přijímačů ustálena a bude přesně zmapována kabeláž k přijímači.

Odchylka může být zvýšena i samotnou teplotou vodičů zejména v zimních měsících, kdy jsou u venkovních přijímačů kabely, nebo i celý přijímač, vystaveny nízké teplotě, zatímco u přijímačů umístěných v zástavbě jsou takovým teplotám

vystaveny pouze antény (současné umístění na Fakultě dopravní). Na teplotní vlivy však budou oproti délce kabelů mnohem hůře aplikovatelné korekce. Tyto odchylky by neměly však být nijak výrazné.

5.4.5 Výstupy z programu MATLAB

V momentální situaci je skript nastaven tak aby generoval pouze velikost odchylky, ICAO značku a vzdálenost letounu od Prahy. Nemá smysl, aby skript vypisoval hodnoty souřadnic jednotlivých letounů, protože si uživatel nepředstaví, v jaké pozici se letoun nachází. Pro přehlednost výstupů by se v budoucnu mohly výsledky importovat do například *google.maps.com* pomocí KLM souborů. Matlab takovou variantu umožňuje. Pro snadnější práci lze použít Mapping Toolbox, který umí vytvořit KLM soubor jednoduchým příkazem, nebo přímo KLM Toolbox.

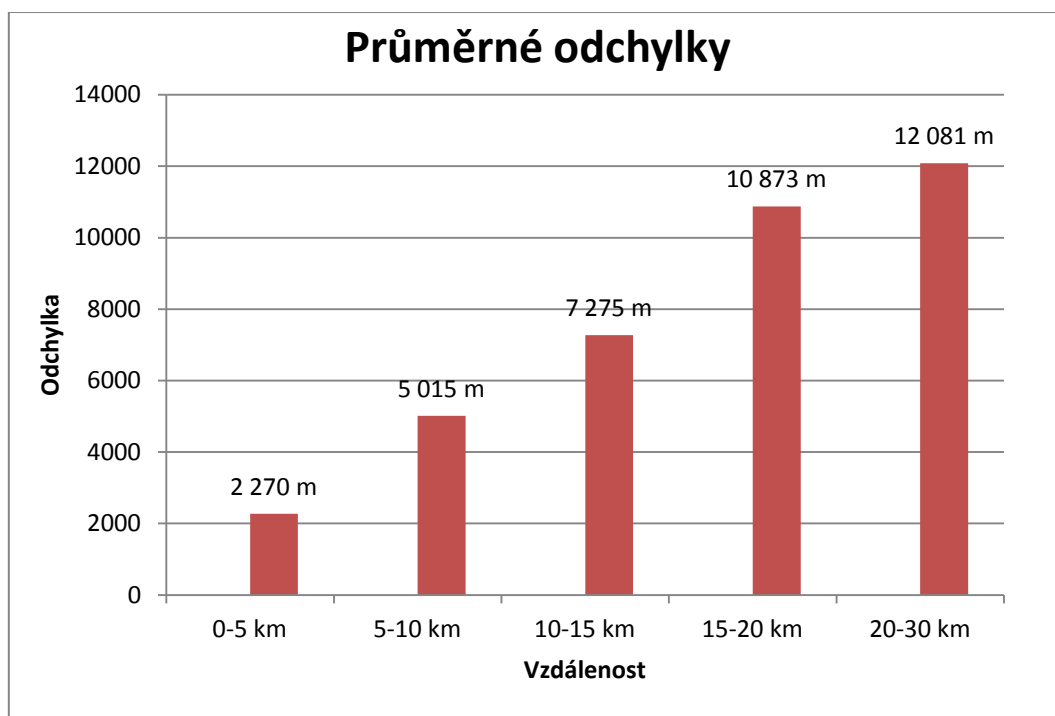
Pro takový výstup je potřeba znát všechny souřadnice. Při zpětné transformaci nám však vzniká chyba, která už není zanedbatelná. Záleží na zvolené výšce, kterou pro transformaci použijeme. Proto bude lepší použít grafické rozhraní, až budou k dispozici čtyři přijímače.

5.4.6 Reálné odchylky MLAT systému

Statistika byla vytvořena na souboru třiceti milionů zpráv, ze kterého jsme však filtrovali pouze letouny, jenž splňovaly kritéria pro kalkulaci. Musely být zachyceny zprávy na všech přijímačích a letoun se musel nacházet ve vzdálenosti do třiceti kilometrů od centra Prahy. V reálné situaci to vypadalo tak, že většina zpráv přicházela od přistávajících nebo vzlétajících letounů. Jak bylo zmíněno výše, zprávy byly přijímány pouze do určité výškové hranice, jelikož přijímač na Fakultě dopravní nebyl schopen pokrýt letouny v blízkosti země.

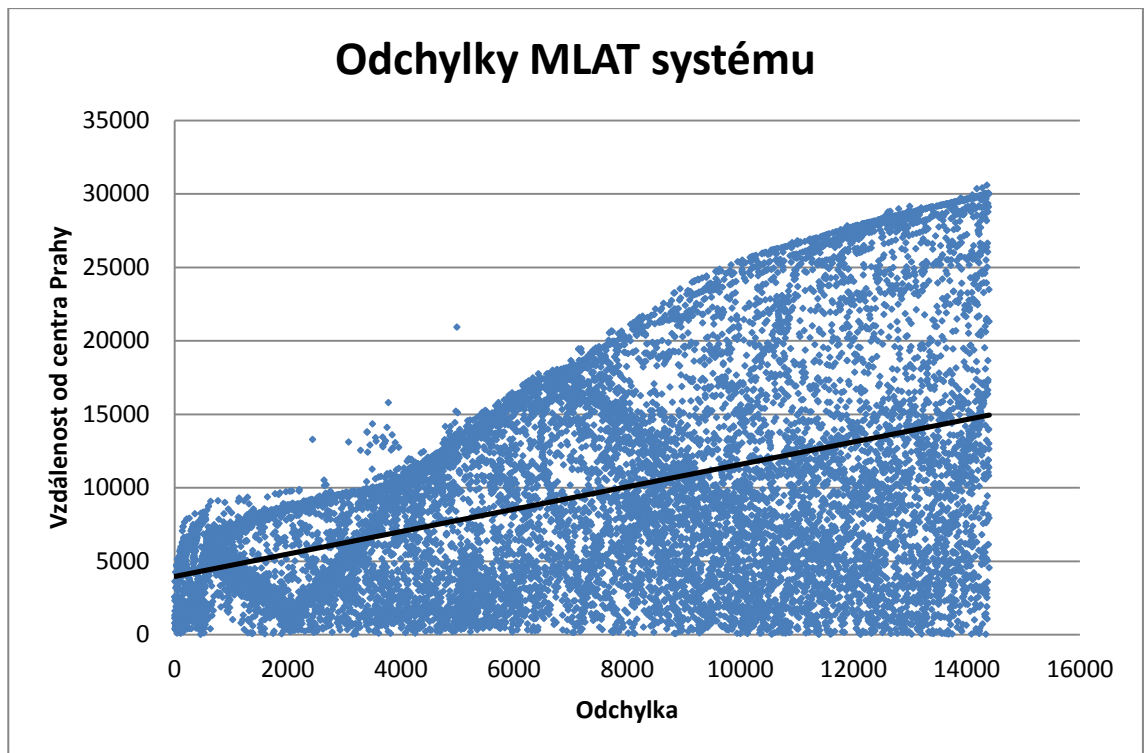
Letounů, které přelétaly nad Prahou do jiných destinací, byl pouze zlomek. Navíc se jejich výška pohybovala okolo deseti kilometrů, a tudíž vzrostla také chyba kvůli 2D multilateraci.

Pro vzdálenější letouny nemá smysl vytvářet statistiky, protože vypočtená odchylka se vzdáleností výrazně roste. Pokud by však škola měla možnost vytvořit WAM systém, bude naopak vítána možnost sledovat letouny mnohem více vzdálené. V tom případě by však musely být přijímače rozmístěny mnohem dále od sebe.



Obrázek 26: Velikost průměrné odchylky v závislosti na vzdálenosti od přijímače

Přesnost celého systému je velmi náchylná na prostor, ve kterém se letoun nachází. V předchozí kapitole lze vidět oblasti s velkou nepřesností výpočtu. V nadcházejícím grafu vidíme rozptyl jednotlivých odchylek v závislosti na vzdálenosti. Vidíme, že i odchylky v malých vzdálenostech od přijímače dosahují velkých hodnot. Pokrývají téměř celé spektrum odchylek na ose.



Obrázek 27: Závislost odchyly na vzdálenosti

5.4.7 Schopnost programu zpracovávat data z reálného provozu

Celý multilaterační modul je schopen zpracovávat přibližně 2000 zpráv za sekundu a provádět výpočet pro polohové zprávy DF17. Toto číslo je však velmi závislé na nastavení celého systému. Pokud omezíme délku seřazovacího pole na polovinu, zmenšíme rámec pro srovnávání zpráv, což znamená, že neporovnáváme celou informaci ve zprávě, ale pouze její část, jsme schopni rychlost zpracování zvýšit. Je však jisté, že program nebude schopen nalézt všechny zprávy, které patří jednomu signálu. Také se bude stávat, že při srovnávání pouze malé části zprávy systém nesprávně přiřadí zprávy k sobě a výsledná pozice bude znehodnocena.

Při malém provozu zejména v noci a brzkých ranních hodinách je systém schopen zpracovávat celý objem dat, jež přijímače zachytí. Nastává však situace, že se velmi málo letadel nachází v oblasti, nad kterou multilateraci provádíme. Kvůli velkému dosahu přijímačů je však systém stále zbytečně zahlcován.

5.4.8 Jednoznačné určení polohy letounu

V našem systému bylo předpokládáno, že se výsledná poloha bude dát eliminovat ze dvou výsledných poloh kvůli jedné nesmyslně vypočtené poloze, avšak

při 2D multilateraci není dostatečně dobře možné při enormně velkých odchylkách správně určit, která poloha je správná. Proto jsme zatím předpokládali, že správná poloha je ta, u které byla zjištěna menší odchylka.

Bylo by také možné vytvořit algoritmus, který se bude řídit podle pořadí příchodu impulzů na dané přijímače, a poté porovnat reálné možnosti pozice. V případě, že by se však letoun nacházel mezi stanicemi, by bylo obtížnější tuto variantu aplikovat. Pro náš systém bude velmi málo pravděpodobné, že se bude letoun nacházet v oblasti mezi přijímači.

5.4.9 **Integrita**

Multilaterační systém je ve své podstatě velmi spolehlivý systém. Výpočet lze negativně ovlivnit při správné konfiguraci systému jen velmi těžko. Jediná možnost, kdy by systém počítal nesprávně polohy letadel, nastane, pokud by selhala synchronizace hodin u některého z přijímačů, nebo u více z nich. Nesmělo by se však jednat o řády nanosekund. Posun synchronizace by musel být značný. Při takovém posunu už by měl být systém schopný rozpoznat, zda je vypočítaná poloha reálně možná. Zejména u těchto variant:

- Zda není v nesmyslné výšce nebo naopak pod zemí.
- Zda se nachází na území, které přijímače pokrývají.

MLAT systém je možné kontrolovat právě díky ADS-B zprávám. Je to velmi efektivní a velmi přesná metoda. Jedinou nevýhodou je, že se tato možnost nenaskytne u letounů, které svou polohu nevysílají.

Náš systém zatím nepracuje samostatně vzhledem k nedostatku přijímačů a nevhodnému rozmístění, proto je zatím nutné, aby byl propojen s ADS-B zprávami, které nesou informaci o poloze, a byl omezen pouze na tato letadla.

5.4.10 **Další možnosti pokračování na projektu**

Projekt školního MLAT systému rozhodně není v takové podobě, kdy by na něm nebylo co zdokonalovat a rozšiřovat. V první řadě je nutné najít umístění pro přijímač z fakulty, který svým umístěním zvedá odchylku výpočtů celého systému. Pokud se to povede, bude možné daleko přesněji určovat polohy letounů a provádět konečnou filtraci jejich skutečných poloh.

Pro další výzkum je nutné vlastnit čtvrtý přijímač, nebo mít alespoň přístup k datům z dalšího přijímače. V blízké době se předpokládá spolupráce s letištěm Praha Letňany. Pokud by letiště umožnilo použití přijímače pro potřeby fakulty, bylo by možné vytvořit modul, který by již zvládal mapovat letouny ve 3D.

Již vytvořený vyhledávací skript by stačilo poměrně jednoduše upravit. Stačil by pouze zvýšit rozptyl pole pro řazení zpráv dle časové značky, protože je pravděpodobné, že při připojení dalšího přijímače budou zprávy, které patří jednomu impulzu, přicházet s větším časovým rozestupem. Dále by se mohl zvýšit rozptyl vyhledávacího pole, i když by dostačoval i již zvolený. Samozřejmě by bylo přidat další adresu přijímače do programu, který všechny datové toky spojuje.

Nejdůležitějším článkem by byl samozřejmě výpočetní modul, který by se musel nahradit zcela novou funkcí. Bylo by zapotřebí pouze vytvořit funkci pro MLAT v prostoru. Odvození i výpočet byly již popsány výše. Poté by bylo možné změnit výstup skriptu, který by již nemusel řešit výšku, a dodával by do transformací z WGS 84 do ECEF a zpět všechny potřebné údaje.

V konečné fázi by bylo vhodné vytvořit skript, který by převáděl všechny výstupy do mapy. Samotný přijímač takovou možností disponuje, ale jde pouze o polohy určené díky ADS-B zprávám, nikoliv polohy určené multilaterací.

6 ZÁVĚR

Ve své práci jsem se zabýval samotnou aplikací MLAT metod nad sítí ADS-B receiverů. Zhodnotil jsem různé metody výpočtu, ale kvůli nedostatku přijímačů jsme se museli spokojit pouze s 2D multilaterací. I při použití low-cost přijímačů se cena za jednu přijímací stanici pohybuje okolo dvaceti až třiceti tisíc korun. Pro aplikaci 3D multilaterace je nutné mít čtyři takové stanice a to výslednou cenu takového systému značně navýší.

Cílem této diplomové práce bylo vytvořit modul, který bude zajišťovat vše potřebné pro výpočet polohy pomocí MLAT metody. Systém je zatím ve stádiu vývoje a bude jej potřeba stále zdokonalovat. Podařilo se nám však dosáhnout správného fungování systému v prostředí Matlab za použití dodatečného programu pro sjednocení datových toků. Přenos dat přes internet značně zpomaluje přenos paketů a způsobuje zdržení u doručení jednotlivých zpráv. Jako vhodnější varianta se jeví použití směrového bezdrátového doručení alespoň v lokalitách okolo Prahy. Nevýhodou této varianty by byly opět vyšší náklady.

Dosažené výsledky jsou však znehodnoceny právě vzhledem k samotnému rozmístění přijímacích stanic. Kvůli němu vznikají kolem přijímačů lokality, ve kterých není možné změřit polohu s přijatelnou odchylkou. Při přemístění alespoň jedné stanice budou výsledky mnohem uspokojivější.

Myslím si, že jsme vytvořili dobrý základ pro budoucí studenty, kteří se na daném projektu budou podílet. MLAT systém nabízí nespočet možností a kombinací, které je potřeba zmapovat. Cílem by měla být snaha se co nejvíce přiblížit profesionálním systémům.

SEZNAM POUŽITÝCH ZDROJŮ

- [1] STANSIFER, Ryan. *Exact Solution of a Three Dimensional Hyperbolic Positioning System*. Melbourne USA, 2011. Florida Institute of Technology.
- [2] PÉK, Vojtěch. *Využití informace z ADS-B v pasivních multilateračních systémech*. Plzeň, 2008. Diplomová práce. Západočeská univerzita v Plzni
- [3] *Mathworks, MATLAB* [online]. [cit. 2015-11-19]. Dostupné z: <http://www.mathworks.com/products/matlab/>
- [4] *Mathworks, Filechange* [online]. [cit. 2015-11-19]. Dostupné z: <http://www.mathworks.com/matlabcentral/fileexchange/>
- [5] *ERA a.s* [online]. [cit. 2015-11-19]. Dostupné z: www.era.cz
- [6] <http://www.atmonline.cz/news/2013/duben/020413b.htm>
- [7] GULDER, Alfred. AUSTRIA'S WIDE AREA MULTILATERATION. [online]. [cit. 2015-01-10]. Dostupné z: <http://www.acac.org.ma/ar/Workshop%20Presentation/WPSurveliancePPT08.pdf>
- [8] DIAMOND, Mellissa. *ADS-X* [online]. [cit. 5.1.2015]. Dostupný z: <http://era.aero/technology/ads-x/>
- [9] DRÁPAL, Stanislav. *VYUŽITÍ SSR MÓDU S PRO ŘÍZENÍ POHYBŮ LETADEL A VOZIDEL PO PLOŠE LETIŠTĚ*[online]. Brno, 2015, 2015-11-19 [cit. 2015-11-19]. Dostupné z: <https://dspace.vutbr.cz/handle/11012/40108>
- [10] *An Algebraic Solution to the Multilateration Problem* [online]. Abdelmoumen Norrdine, 2015, 2015-11-19 [cit. 2015-11-19]. Dostupné z: http://www.researchgate.net/publication/275027725_An_Algebraic_Solution_to_the_Multilateration_Problem
- [11] POTLURI, Sreeram. *HYPERBOLIC POSITION LOCATION ESTIMATOR WITH TDOAS FROM FOUR STATIONS*[online]. New Jersey, 2002, 2015-11-19 [cit. 2015-11-19]. Dostupné z: <http://archives.njit.edu/vol01/etd/2000s/2002/njit-etd2002-037/njit-etd2002-037.pdf>

- [12] BUCHER, RALPH. *A Synthesizable VHDL Model of the Exact Solution for Three-dimensional Hyperbolic Positioning System* [online]. New Jersey, 2001, 2015-11-19 [cit. 2015-11-19]. Dostupné z: <http://www.hindawi.com/journals/vlsi/2002/935925/abs/>
- [13] *ECEF_ENU_Longitude_Latitude_relationships* [online]. [cit. 2015-11-19]. Dostupné z: https://upload.wikimedia.org/wikipedia/commons/7/73/ECEF_ENU_Longitude_Latitude_relationships.svg
- [14] *Air Traffic View ATM k621* [online]. [cit. 2015-11-19]. Dostupné z: <http://atmlab.fd.cvut.cz/>
- [15] *Google maps* [online]. [cit. 2015-11-19]. Dostupné z: <https://www.google.cz/maps>
- [16] *Wide Area Multilateration* [online]. 2005, 2015-11-19 [cit. 2015-11-19]. Dostupné z: <https://www.eurocontrol.int/sites/default/files/publication/files/surveillance-report-wide-area-multilateration-200508.pdf>
- [17] *Radarcape* [online]. [cit. 2015-11-19]. Dostupné z: <http://www.modesbeast.com/>
- [18] ZACH, Martin. *Návrh nízkonákladového MLAT*. Praha, 2015.

SEZNAM POUŽITÝCH ZKRATEK

2D – Dvou-dimenzionální prostor

3D – Troj-dimenzionální prostor

ACAS – Airoborne Collision Avoidance System

ADS-B – Automatic Dependent Surveillance - Broadcast

ATM – Air Trafic Management, uspořádání letového provozu

CRC – Kontrolní kód při správném přenosu zprávy

DF – Downlink Format, formát sestupného spoje

ECEF – Earth Centered, Earth Fixed, souřadný systém s centrem v centru Země

ENU – East North Up, lokální souřadný systém

ERA – Společnost zabývající se vývojem MLAT systémů se sídlem v Pardubicích

EUROCONTROL – Evropská organizace pro bezpečnost a rozvoj letecké dopravy

Ft. – Feet , jednotka délky 1ft = 0.3048m

GNSS – Global Navigation Satelite System

GPS – Global Positioning System

ICAO – Interational Civil Aviation Organization

IP – Internet Protocol

KLM – Soubory zprostředkovávající zobrazení na mapovém podkladu

MATLAB – Matematická laboratoř, program, ve kterém byla zpracována tato práce

MLAT – Multilaterace

MSSR – Monopulzní sekundární radar

NextGen - Projekt pro optimalizaci a modernizaci ATM systémů v USA

SESAR – Single European Sky ATM research – Program EU pro optimalizaci řízení letového provozu

SMR – Surface Movement Radar, radar zaznamenávající pohyb po plochách letiště

SSR – Sekundární radar

TCP/IP – Internetový protokol

TDOA – Time Difference Of Arrival, rozdíl časů příchodu na jednotlivá stanoviště

UDP – User Datagram Protocol

USB – Universal Serial Bus, moderní způsob připojení periferií k počítači

UTC – Koordinovaný světový čas

WAM – Wide Area Multilateration

WGS84 – Celosvětově uznávaný geodetický systém

PŘÍLOHY

Funkce pro převod souřadnic

Z WGS 84 do ECEF

```
function[ECEF_position]=WGS84ECEF(position)
% převod z WGS84 do ECEF

fi=position(1,1)*pi./180; %zeměpisná šířka
lambda=position(1,2)*pi./180; %zeměpisná délka
h=position(1,3);% výška nad elipsoidem
a=6378137;
e=0.0818191908425;
N=a./ (sqrt(1-((e.^2.)*(sin(fi))^2)));
x=(N+h).*cos(fi).*cos(lambda);
y=(N+h).*cos(fi).*sin(lambda);
z=((1-e.^2).*N+h).*sin(fi);
ECEF_position=[x y z];
```

Z ECEF do ENU (East, North, Up)

```
% převod ECEF souřadnic do lokálních souřadnic
```

```
%refLat, refLong, refH, - ve WGS84
```

```
% X, Y, Z - v ECEF
```

```
function [e,n,u] = xyz2enu(refLat, refLong, refH, X, Y, Z)
```

```
[Xr,Yr,Zr] = wgs2ecef(refLat,refLong,refH);
```

```
refLat=refLat*pi./180;
```

```
refLong=refLong*pi./180;
```

```
e = -sin(refLong).* (X-Xr) + cos(refLong).* (Y-Yr);
```

```
n = -sin(refLat).*cos(refLong).* (X-Xr) -
```

```
sin(refLat).*sin(refLong).* (Y-Yr) + cos(refLat).* (Z-Zr);
```

```
u = cos(refLat).*cos(refLong).* (X-Xr) + cos(refLat).*sin(refLong).* (Y-
```

```
Yr) + sin(refLat).* (Z-Zr);
```

Převod z ENU do WGS84

```
[X,Y,Z] = enu2ecef(e,n,u,lat,lon,h,wgs84Ellipsoid );
```

```
WGS84 = ecef2lla([X Y Z]);
```

Pokud však neznáme třetí souřadnici a chceme znát polohu v systému WGS84, je nutné si souřadnici zvolit a použít následující skript. Hodnoty požadované výšky si můžeme nastavit.

```

XYZ =[X ,Y, Z];

pozadovana_vyska=6000;

for i=1:6
    if i>=2
        XYZ(3)= XYZ(3)-(WGS84(3) - pozadovana_vyska);
    end

WGS84 = ecef2lla([XYZ(1) XYZ(2) XYZ(3)])

end

```

Přílohy na CD

- Program pro sjednocení datových toků Air Funnel
- fce_DF17_TC9_22_krom19_CPR_Air_decoder.m - skript pro dekódování zpráv DF17
- Multilaterace.m - skript pro asociaci zpráv
- Poloha_mlat_2.m - skript pro výpočet polohy
- WGS842ECEF.m – skript pro transformaci z WGS84 do ECEF
- xyz2enu.m - skript pro transformaci z ECEF do lokálních souřadnic